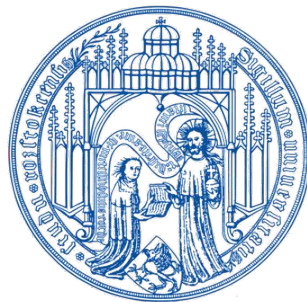


Distanzabhängige Interaktion in großen hochauflösenden Displayumgebungen

Dissertation
zur Erlangung des akademischen Grades
Doktor-Ingenieur (Dr.-Ing.)
der Fakultät für Informatik und Elektrotechnik
der Universität Rostock



vorgelegt von
Anke Lehmann

Rostock, 2015

Erster Gutachter: Prof. Dr. Sc. techn. Oliver Stadt, Lehrstuhl Visual Computing, Institut für Informatik, Universität Rostock

Zweiter Gutachter: Prof. Dr.-Ing. habil. Heidrun Schumann, Lehrstuhl Computergrafik, Institut für Informatik, Universität Rostock

Externer Gutachter: Prof. Dr. rer. nat. habil. Frank Steinicke, Human-Computer Interaction, Fachbereich Informatik, Universität Hamburg

Einreichung: 30. September 2014

Verteidigung: 17. Dezember 2014

Zusammenfassung

Mit zunehmender Verbreitung von großen hochauflösenden Displayumgebungen und der kostengünstigen Anschaffung der Systeme kommen diese in immer mehr Anwendungsgebieten zum Einsatz, z.B. bei der wissenschaftlichen Visualisierung, bei der Produktentwicklung in der Automobilindustrie oder als Geoinformationssystem. Große hochauflösende Displays (GHDs) bieten eine große Anzeigefläche mit hoher Darstellungsgenauigkeit. Sie ermöglichen es Benutzern Informationen auf eine neue Art und Weise wahrzunehmen und mit den Informationen anders zu interagieren, als es bei herkömmlichen Desktopumgebungen der Fall ist [AEYN11]. Dies hat Änderungen im Nutzerverhalten und den Interaktionsmöglichkeiten zur Folge. Eine natürliche Interaktion in GHD-Umgebungen bietet die physische Navigation, bei der Nutzer näher an das Display heran gehen, um detaillierte Informationen zu betrachten, bzw. sich von der Displaywand für eine Überblickdarstellung entfernen. Im Allgemeinen werden in GHD-Umgebungen Aufgaben nah oder entfernt vom Display ausgeführt, die eine unterschiedliche Präzision und Sensitivität der Nutzereingabe erfordern.

Das Ziel der vorliegenden Arbeit ist die Entwicklung von Methoden, die den Nutzer bei seiner (gegenwärtigen) Tätigkeit in großen Displayumgebungen unterstützen, indem die Visualisierung und die Interaktivität an den Nutzer, unter Berücksichtigung seines aktuellen Betrachtungsabstandes, angepasst werden.

Der vorgestellte Ansatz des Interaction Scaling (IS) verwendet die physische Navigation für die distanzabhängige Informationsdarstellung und für die distanzabhängige Präzision der Nutzerinteraktion in GHD-Umgebungen. Die resultierende Mappingmethode für IS kombiniert zwei Ansätze: die Berechnung eines distanzabhängigen Mappingfaktors und den automatischen oder manuellen Wechsel der Präzisionsstufen.

Die unterschiedlichen Einflussfaktoren für die distanzabhängige Interaktion in GHD-Umgebungen lassen sich in einen Parameterraum mit den Komponenten virtuelle Umgebung, physische Umgebung, Benutzer und Mapping einordnen. Mit Hilfe von Beispielszenarien wird die Gewichtung der Einflussfaktoren im Parameterraum diskutiert und welche Anpassung für die Mappingmethode daraus resultiert. Zur Realisierung des Konzeptes werden bekannte Navigations- und Manipulationstechniken angepasst und umgesetzt, sowie distanzabhängige und taskbasierte Mappingmethoden integriert und evaluiert.

Anhand von Nutzerstudien wird der Einfluss bisher wenig evaluierter subjektiver und objektiver Benutzereigenschaften auf Faktoren der physischen Umgebung und das Mapping untersucht. Mit Hilfe von iterativen Studien wird aufgezeigt, dass Interaction Scaling für 2D Manipulationstechniken die Nutzerperformanz verbessert, wenn die benötigte Präzision steigt. Des Weiteren müssen die Präferenzen der Benutzer bei der distanzabhängigen Interaktion berücksichtigt werden, z.B. die individuelle Distanz aus der sie in GHD-Umgebungen bevorzugt arbeiten.

Abstract

Nowadays, large high-resolution displays (LHRDs) are used in wide application areas like product engineering, geospatial imaging or scientific visualization. They combine a large physical display area with high pixel density. In addition, LHRDs changing the way the users perceive and interact with information as is known from small displays [AEYN11]. For instance, LHRDs allow users to perceive the global context of information from a distance and to explore detailed information at close-up range (physical navigation). In general, there are tasks performed in close-up range also at distant which required user input with different sensitivity and precision.

The goal of this thesis is the development of methods to assist the user during his current activity by adjusting the visualization and interactivity depending on the user's distance to the LHRD. The approach of interaction scaling (IS) uses physical navigation to provide distance-aware information visualization and precision of interaction with LHRDs. The mapping method of IS combines two approaches: the calculation of an distance-based mapping factor and the manual/automatic change of precision levels.

Physical environment, virtual environment, user, and mapping are components of the design-space to classify the various influence parameters of distance-aware interaction with LHRDs. We discuss the weighting of those parameters and there adjustment to the mapping methods by using scenarios. Realizing the introduced concept of IS we customize common navigation and manipulation techniques together with distance-aware and task-based mapping methods by integrating and evaluating. The influence of objective/subjective user properties is examined for parameters of the physical environment as well the mapping by user studies. Iterative studies have shown that the user performance is improved by applying interaction scaling on 2D manipulation techniques when the required precision is increasing. Furthermore, the user's preferences have to be consider in distance-aware interaction e.g., the user's preferred and individual interaction distance.

Keywords: Distance-Aware Interaction, Powerwall, Proximity, 2D Manipulation Technique, User Study

Teile dieser Arbeit wurden veröffentlicht in:

- Anke Lehmann, Oliver Staadt. Evaluation of Distance-Aware Bimanual Manipulation Techniques for Large High-Resolution Displays. In *Human-Computer Interaction – INTERACT 2015*, Volume 9298 of the series Lecture Notes in Computer Science, S. 436-454, Springer Berlin Heidelberg, 2015. (Kapitel 6.2)
- Christian Scheel, Falko Löffler, Anke Lehmann, Heidrun Schumann, Oliver Staadt. Dynamic Level of Detail for Tiled Large High-Resolution Displays. 11. Workshop GI VR/AR 2014, Shaker Verlag, S. 109–119, 2014. (Kapitel 3.3.1 und 5.1.2)
- Anke Lehmann, Oliver Staadt. Distance-Aware Bimanual Interaction for Large High-Resolution Displays. In *Computer Vision, Imaging and Computer Graphics. Theory and Application*, Volume 359 of the series Communications in Computer and Information Science, S. 97–111. Springer Berlin Heidelberg, 2013. (Kapitel 3.3.3 und 6.1.1)
- Axel Radloff, Anke Lehmann, Oliver Staadt, Heidrun Schumann. Smart Interaction Management: An Interaction Approach for Smart Meeting Rooms. In *Proceedings of the 8th International Conference on Intelligent Environments (IE'12)*, S. 228–235, 2012. IEEE Computer Society. (Kapitel 3.3.2)
- Anke Lehmann, Oliver Staadt. Distance-adapted 2D Manipulation Techniques for Large High-Resolution Display Environments. In *Proceedings of the International Conference on Computer Graphics Theory and Applications*, S. 387–394, 2012. SciTePress. (Kapitel 3.3.3 und 6.1.1.2)
- Anke Lehmann, Heidrun Schumann, Oliver Staadt, Christian Tominski. Physical Navigation to Support Graph Exploration on a Large High-Resolution Display. In *Proceedings of the 7th International Conference on Advances in Visual Computing - Volume Part I, ISVC'11*, S. 496–507, Berlin, Heidelberg, 2011. Springer-Verlag. (Kapitel 3.3.1 und 5.1.1)

Danksagung

Diese Dissertation wurde im Rahmen des Graduiertenkollegs MuSAMA (Multi-modal Smart Appliance Ensembles for Mobile Applications) durch die Deutsche Forschungsgemeinschaft (DFG) gefördert. Durch diese finanzielle Unterstützung konnte die Arbeit erst entstehen.

Mein besonderer Dank gilt meinem Betreuer Herrn Prof. Oliver Staadt, der die Arbeit in diesem interessanten Forschungsfeld ermöglicht hat und darüber hinaus durch Anregungen, klärende Diskussionen und wertvolle Hinweise zum Gelingen dieser Arbeit beitrug.

Außerdem möchte ich mich bei den beiden Gutachtern Frau Prof. Heidrun Schumann und Herrn Prof. Frank Steinicke für ihre Bereitschaft danken, diese Arbeit zu begutachten.

Einen besonderen Dank gilt meinen Kollegen Stephan Ohl und Malte Willert für die vielen konstruktiven Diskussionen und Anregungen, sowie den Kollegen an den Lehrstühlen Visual Computing und Computergrafik. Den Stipendiaten des Graduiertenkollegs MuSAMA, insbesondere Kristina, Till und Axel danke ich für die gute Arbeitsatmosphäre und den (fachlichen) Diskussionen am Kaffeeautomaten.

Lars Schwabe danke ich recht herzlich für seine Unterstützung bei den Experimenten. Allen Probanden danke ich für die Teilnahme an den Nutzerstudien und ihren vielfältigen Kommentare zu den Interaktionstechniken. Gleichzeitig bedanke ich mich bei allen Studenten, die mit ihren Abschlussarbeiten zum Gelingen dieser Arbeit beitrugen.

Ebenso möchte ich mich bei Christiane, Björn und Wolfgang bedanken für ihre Mühen beim Korrekturlesen, sowie ihrer moralischen und seelischen Unterstützung in den stressigen Phasen. Ein sehr herzlicher Dank gilt meiner Familie und meinen Freunden, die alle auf ihre Art und Weise zum Gelingen dieser Arbeit beigetragen haben.

Inhaltsverzeichnis

1	Einleitung	11
1.1	Zielsetzung der Arbeit	14
1.2	Ergebnisse der Arbeit	15
1.3	Struktur der Arbeit	17
2	Grundlagen und Stand der Forschung	19
2.1	Physiologische Eigenschaften des Menschen	19
2.1.1	Visuelle Wahrnehmung	20
2.1.2	Motorik	21
2.2	Evaluierung	27
2.2.1	Statistische Methoden	27
2.2.2	Evaluierungsmethoden	31
2.3	Eigenschaften von großen hochauflösenden Displayumgebungen	34
2.3.1	Displayauflösung und Displaygröße	34
2.3.2	Untersuchungen	36
2.4	Interaktion in großen hochauflösenden Displayumgebungen	40
2.4.1	Nutzerinteraktion	40
2.4.1.1	Eingabemodalität	40
2.4.1.2	Tracking	43
2.4.1.3	Interaktionsaufgaben	46
2.4.1.4	Mapping	50
2.4.2	Verwandte Arbeiten	55
2.4.2.1	Nutzereingabe nahe am Display	56
2.4.2.2	Nutzereingabe entfernt vom Display	56
2.4.2.3	Distanzabhängige Nutzereingabe	60
2.5	Anforderungen für Interaktion in großen hochauflösenden Displayumgebungen	65
3	Konzept der distanzabhängigen Interaktion	69
3.1	Parameterraum	70
3.1.1	Physische Umgebung	72
3.1.2	Virtuelle Umgebung	76
3.1.3	Benutzer	79
3.1.4	Mapping	82

3.2	3D Interaktion mit Interaction Scaling	88
3.2.1	Mappingfunktionen	90
3.3	Beispielszenarien	93
3.3.1	Navigation und Selektion mit Head-Tracking	94
3.3.2	Manipulation mit einhändiger Interaktion	99
3.3.3	Manipulation mit beidhändiger Interaktion	106
3.4	Zusammenfassung	110
4	Experimentalumgebung	113
4.1	Physischer Interaktionsraum	114
4.2	Trackingbereich	115
4.3	Software Frameworks	116
5	Umsetzung der distanzabhängigen Interaktion	119
5.1	Distanzabhängige Informationsdarstellung	119
5.1.1	Navigationstechniken für 2D Anwendung	119
5.1.2	Navigationstechnik für 3D Anwendung	123
5.2	Distanzabhängige Präzision der Manipulation	125
5.2.1	Testanwendung	126
5.2.2	Interaktionstechniken	128
5.2.3	Mapping	132
5.2.4	Designentscheidungen	136
6	Evaluierung der distanzabhängigen Interaktionstechniken	139
6.1	Pilotstudien	140
6.1.1	Beidhändige 2D Manipulationstechniken	140
6.1.1.1	Studie 1: symmetrische und asymmetrische Technik	141
6.1.1.2	Studie 2: asymmetrische Technik	146
6.1.2	Einhändige 2D Manipulationstechnik	152
6.1.2.1	Studie 4: absolute Nutzereingabe	152
6.2	Studie 3: symmetrische und asymmetrische Technik	157
6.2.1	Hypothesen	157
6.2.2	Design und Prozedur	158
6.2.3	Ergebnisse	163
6.2.3.1	Effizienz und Effektivität	164
6.2.3.2	Zufriedenheit und Arbeitsaufwand	167
6.2.3.3	Nutzerperformanz	169
6.2.4	Diskussion	175
6.3	Zusammenfassung	181
7	Schlussbetrachtung	185
	Ausblick	192

A Anhang	195
A.1 Eingabemodalitäten und Nutzerinteraktion	195
A.2 Fragebögen	196
A.2.1 Fragebogen der Studie 1	196
A.2.2 Fragebogen der Studie 2	199
A.2.3 Auszug PSSUQ, ASQ nach J.R. Lewis, 1995	201
A.2.4 Fragebögen der Studie 3	201
A.3 Ergebnisse der Evaluierungen	207
A.3.1 Auswertung mit SigmaPlot	207
A.3.2 Statistische Auswertungen der Studie 1	208
A.3.3 Ergebnisse und statistische Auswertungen der Studie 2	209
A.3.4 Ergebnisse und statistische Auswertungen der Studie 3	212
A.3.4.1 Repräsentative Laufprofile aus Studie 3	229
B Anhang	233
B.1 Kennwerte der deskriptiven Statistik	233
B.2 Kennwerte der Inferenzstatistik	235
Statistische Verfahren zur Durchführung der Signifikanzprüfung	237
Glossar	243
Abkürzungs- und Symbolverzeichnis	245
Verzeichnis der URLs	247
Abbildungs- und Tabellenverzeichnis	248
Literaturverzeichnis	253
Thesen	269

1. Einleitung

In den letzten Jahren haben sich die herkömmlichen Desktopsysteme von Einzelmonitor-Arbeitsplätzen über mehrere Monitore auf dem Schreibtisch bis hin zu komplexen Displayumgebungen entwickelt. Während sich bei den Standard Desktopumgebungen sehr gute Interaktionstechniken für jeden Anwendungsbereich herausbildeten, wurden für die komplexen Displayumgebungen (z.B. CAVE, Stereodisplays) neue Interaktionstechniken entwickelt. Heutzutage werden die Displays immer größer und nehmen ganze Räume ein [PPK13], was eine Herausforderung an die Interaktion des Nutzers mit solchen Systemen darstellt. Diese Displayumgebungen kombinieren die Vorteile einer großen Anzeigefläche mit hoher Darstellungsgenauigkeit (Pixeldichte) als sogenannte Displaywände.

Die Displayumgebung umfasst sowohl die Displays auf denen die Informationen dargestellt werden als auch den physischen Raum vor den Displays, in dem Personen die Informationen betrachten. Neben der Darstellung von Informationen ist auch die Interaktion mit den präsentierten Informationen in großen hochauflösenden Displayumgebungen ein wichtiger Aspekt. Denn um Informationen zu explorieren und zu verstehen, muss der Nutzer mit ihnen interagieren können [TC05, YKSJ07].

In diesem Zusammenhang sollte Interaktion in großen hochauflösenden Displayumgebungen folgende Eigenschaften unterstützen: Mobilität, Genauigkeit, schnelle Verfügbarkeit und eine einfache Bedienung, um Interaktionsaufgaben durchzuführen.

Der Betrachter kann einerseits aus der Entfernung viel Information auf der großen Anzeigefläche wahrnehmen. Andererseits kann er nahe an das Display herangehen, um sich Details anzusehen, ohne dabei den Kontext der gesamten dargestellten Informationen zu verlieren (Prinzip der Informationssuche: *overview first, zoom and filter, then details-on-demand* [Shn96]). Dies erfordert **Mobilität** bei der Interaktion.

Große hochauflösende Displayumgebungen werden immer populärer und kommen heutzutage vielseitig zum Einsatz, z.B. bei der wissenschaftlichen Visualisierung, bei der Produktentwicklung in der Automobilindustrie oder als Geoinformationssysteme. Der Vorteil für wissenschaftliche Visualisierung auf großen hochauflösenden Displays (GHD) ist die Präsentation von komplexen Daten mit hohem Detailgrad im Kontext der sie umgebenden Informationen. Diese detaillierten und komplexen Daten erfordern eine entsprechende **Genauigkeit** bei der Interaktion mit ihnen.

Für die Produktentwicklung bietet sich ein weiterer Vorteil, aufgrund der überdimensionalen Anzeigefläche, Produkte im Designstadium in Originalgröße darzustellen und so Dimensionen realistisch zu visualisieren [BFBK00]. Darüber hinaus werden diese Displayumgebungen für kollaboratives Arbeiten mit großen Datensätzen, wie hochauflösende Satellitenbilder genutzt. Beispielsweise wurden nach dem Erdbeben in Haiti 2010 die Zerstörungen auf der Insel anhand der aktuellen Satellitenbilder in den GHD-Umgebungen der University of California, Davis ausgewertet¹. Daher wird eine **schnelle Verfügbarkeit und einfache Bedienung** der Interaktion benötigt.

Da sich der Nutzer innerhalb solcher Umgebungen auf natürliche Weise bewegt, um Information aufzunehmen, sind stationäre Interaktionsgeräte (wie Maus und Tastatur) ungeeignet oder nur beschränkt einsetzbar. GHD-Umgebungen benötigen daher andere Interaktionsmethoden als Desktopsysteme. Hauptsächlich werden Interaktionstechniken aus dem Bereich 3D Interaktion für *Virtual Reality* (VR) verwendet und für große hochauflösende Displays angepasst. Typische Interaktionsaufgaben sind Selektion, Manipulation, Navigation, Symboleingabe und Systemkontrolle [BKLP04].

Es gibt mittlerweile eine Vielzahl von Interaktionstechniken und Anwendungen für große Displayumgebungen, welche gezielt die Vorteile solcher Umgebungen nutzen. Ball et al. [BNB07] haben in mehreren Studien gezeigt, dass der Einsatz der physischen Navigation sich positiv auf die Nutzerperformanz bei unterschiedlichen Navigationsaufgaben auswirkt und der virtuellen Navigation in GHD-Umgebungen vorgezogen wird. Bei der virtuellen Navigation ändert der Nutzer seine "virtuelle Blickrichtung"² auf die visuelle Darstellung durch Bewegungen des Eingabegerätes. Der Nutzer muss sich nicht zu dem Bildausschnitt hinbewegen, sondern seine Sicht wird angepasst. Bei der physischen Navigation ändert der Nutzer seine Blickrichtung durch physische Bewegungen, beispielsweise durch Augen- oder Kopfbewegungen, herumlaufen, sich ducken oder sitzen. Dies ist eine direkte und natürliche Interaktion und benötigt weniger kognitiven Aufwand für den Nutzer.

Dieses natürliche Verhalten der Nutzer beim Betrachten von virtuellen Objekten in GHD-Umgebungen sollte bei der Interaktion mit Visualisierungen berücksichtigt werden. Ein Beispiel: Befindet sich der Nutzer nah an dem hochauflösenden Display kann er nur einen begrenzten Ausschnitt der dargestellten Daten sehen, jedoch mit sehr hohem Detailgrad. Die Interaktion mit den detaillierten Daten sollte äquivalent erfolgen, d.h. der Nutzer kann sehr präzise mit den Daten im Sichtbereich interagieren. Im Gegensatz dazu erfolgt die Wahrnehmung der Datenrepräsentation aus der Entfernung für den Nutzer schnell und im allgemeineren Kontext. Daher sollte die Interaktion aus der Entfernung mit grober Präzision

¹ <http://phys.org/news200924198.html>

² Hier ist die virtuelle Kamera gemeint, welche die Blickrichtung des Nutzers in der virtuellen Umgebung repräsentiert.

und schnell erfolgen. Neben der Größe der Anzeigefläche in GHD-Umgebungen, die unterschiedliche Darstellungsgenauigkeit durch den Betrachtungsabstand bieten, kann auch der Betrachtungsabstand für die **Präzision der Interaktion** genutzt werden.

Buxton und Kollegen [BFBK00] konnten die Vorteile beim Einsatz von großen Displays in der Automobilindustrie zeigen: Einerseits können Objekte in Originalgröße dargestellt werden, was einen emotionalen Aspekt für die Kunden und Designer hat und andererseits die Kollaboration bei Diskussionen im Designprozess unterstützen. Neben dem reinen Darstellen von Informationen auf großen Displays sollte es Betrachtern auch möglich sein die dargestellten Bilder zu manipulieren, zu explorieren und zu annotieren. In ihrem Prototyp konnten Designer mit Hilfe der adaptierten Interaktionstechnik, welche die gleichen Bewegungsabläufe aus der realen Welt für die virtuelle Umgebung verwenden, die Umrisse von Automobilen wie gewohnt skizzieren.

Durch den Einsatz großer hochauflösender Displays können viel feinere Strukturen und Objekte visualisiert und nachbearbeitet werden, ohne die Visualisierung vergrößern zu müssen (Zoom). So können bei der Innenraumgestaltung von Automobilen GHDs eingesetzt werden, um dem Designer und dem Kunden einen realitätsnahen Eindruck der Dimensionen und Anordnungen von Objekten, z.B. dem Armaturenbrett, zu vermitteln. Mit Hilfe des GHD kann der Designer aus der Entfernung die Anordnung von einzelnen Komponenten schnell und grob nach Kundenwünschen platzieren. Anschließend geht der Designer näher an das Display heran, um die Feinjustierung vorzunehmen (z.B. Definition der exakten Abstände zwischen Servicesymbolen und Radioanzeige). Der Vorteil liegt darin, dass für den Designer die Grobplatzierung und Feinjustierung von Objekten innerhalb einer Umgebung stattfindet und er nicht zwischen Präsentationsmodus auf der Displaywand und Designmodus auf dem Arbeitsplatz-PC umschalten muss. Außerdem können damit Änderungswünsche während der Diskussion und Kollaboration in der GHD-Umgebung umgesetzt werden und die Rückmeldung erfolgt unmittelbar.

Zusammenfassend lässt sich feststellen, dass in großen hochauflösenden Displayumgebungen einerseits Aufgaben durchgeführt werden, in denen detaillierte Informationen bearbeitet werden und diese meist nah am Display stattfinden, und andererseits Aufgaben, die besser aus der Entfernung ausgeübt werden [NSS⁺06].

FAZIT: Die Herausforderung liegt darin, den aktuellen Abstand des Nutzers für den Wechsel zwischen den beiden Aufgabenbereichen, die oftmals eine unterschiedliche Präzision und Sensitivität der Nutzereingabe erfordern, zu nutzen. Das heißt, es werden Methoden benötigt, die den Nutzer bei seiner (gegenwärtigen) Tätigkeit in großen Displayumgebungen unterstützen, indem die Visualisierung und Interaktivität an den Nutzer angepasst wird.

1.1. Zielsetzung der Arbeit

Ausgehend von der Interaktionstechnik und der Anwendungsaufgabe werden unterschiedliche Präzisionsstufen in der Darstellung und in der Benutzerinteraktion benötigt. Dabei müssen die physische Umgebung (z.B. großer Interaktionsraum, Sensortechnologie) als auch die physiologischen Eigenschaften des Benutzers (z.B. Wahrnehmung, Motorik, Mobilität) berücksichtigt werden.

In dieser Arbeit wird folgende Fragestellung zu Interaktion in GHD-Umgebungen untersucht: *Wie muss distanzabhängige Interaktion für Fullscreen-Anwendungen in großen hochauflösenden Displayumgebungen gestaltet sein, damit die physische Navigation und die Präferenzen des Benutzers berücksichtigt werden?*

Zielsetzung der Arbeit ist die distanzabhängige Präzision der Interaktion (*Interaction Scaling*) in GHD-Umgebungen zu unterstützen. Dafür werden bekannte 3D Interaktionstechniken aus dem VR-Bereich verwendet (z.B. Ray-Casting, Drag-and-Drop, Head-Tracking) und bezüglich ihrer Eignung geprüft: *Wie müssen die Verhaltensweisen der 3D Interaktionstechnik an den neuen physischen Interaktionsraum des großen hochauflösenden Displays angepasst werden?*

Mit Hilfe zusätzlicher Sensoren im realen Raum (z.B. zur Bestimmung der Nutzerposition) und einem angepasstem Mapping (Interaktion im virtuellen Raum) soll die gewünschte Genauigkeit der Nutzerinteraktion aus unterschiedlichen Distanzen unterstützt werden, ohne Neuimplementierung der Anwendung. Die Herausforderung liegt vor allem in der Frage: *Welche Präzision wird in dem jeweiligen Nutzerszenario benötigt und wie direkt und unmittelbar muss die Nutzereingabe erfolgen?*

Die Nutzereingabe kann durch relative Bewegungen (z.B. Maus als Eingabegerät) oder durch absolute Bewegungen (z.B. Laserpointer als Eingabegerät) erfolgen. Dabei werden die physischen Bewegungen des Eingabegerätes auf virtuelle Bewegungen des Interaktionszeigers (Cursor) abgebildet. Werden relative Bewegungen verwendet, beobachtet der Nutzer nur den visuellen Cursor auf dem Display und die entsprechende Hand- oder Armbewegung muss nicht im Sichtfeld des Nutzers sein. In großen Displayumgebungen werden vorwiegend absolute Bewegungen als

Eingabemodalität verwendet und die eigene Armbewegung ist für den Nutzer dabei sichtbar. Jedoch ist bei der Verwendung von absoluten Cursorpositionen bei der freihändigen Interaktion das natürliche Handzittern zu beobachten und ein präzises Positionieren und Stillhalten des Cursors schwierig. Hier kann nur eingeschränkt mit relativen virtuellen Cursorbewegungen gearbeitet werden, da Armbewegung und Zeigerbewegung über die Zeit voneinander abweichen können und dies den Nutzer irritiert.

Die Thesen dieser Arbeit sind:

- Die physische Navigation bietet eine natürliche Nutzereingabe für die distanzabhängige Interaktion in GHD-Umgebungen.
- Die gewünschten Präzisionen der Interaktion und der Nutzereingabe erfordern differenzierte Mappingmethoden für die Interaktionstechniken.
- Die individuellen und subjektiven Eigenschaften der Nutzer haben einen starken Einfluss auf die Mappingmethode, so dass bei der distanzabhängigen Interaktion die geeignete Mappingmethode und der angepasste Mappingfaktor zwischen den Nutzern variiert.
- Bei der distanzabhängigen Interaktion müssen die Präferenzen der Nutzer berücksichtigt werden. Zum Beispiel haben Nutzer eine individuelle Distanz aus der sie in GHD-Umgebungen bevorzugt arbeiten.

1.2. Ergebnisse der Arbeit

Es wurde ein Parameterraum konzipiert, der distanzabhängige Interaktion in GHD-Umgebungen unterstützt. Abhängig davon welcher Faktor des Parameterraumes mehr gewichtet wird (Trackinggenauigkeit, Eingabemodalität, Nutzerpräferenz etc.) eignen sich unterschiedliche Mappingmethoden mit variierender Präzision.

BEITRAG: Der Parameterraum bietet abhängig von der Interaktionsaufgabe (z.B. Navigation, Manipulation) und den verfügbaren Freiheitsgraden der Nutzerinteraktion (oder des Interaktionsgerätes) ein distanzabhängiges Mapping von physischen Bewegungen (Körper, Gerät) im realen Interaktionsraum auf virtuelle Bewegungen (Cursor) im virtuellen Raum an.

Somit kann der Nutzer aus der Entfernung mit den virtuellen Inhalten interagieren, meist mit etwas gröberer Präzision aber schnellen Bewegungen, während er im Nahbereich mit sehr hoher Präzision, durch ausgedehnte Bewegungen agieren kann.

Die Einflussfaktoren des Parameterraumes lassen sich in vier Komponenten einordnen (siehe Abbildung 1.2.1): **physische Umgebung** (z.B. Displaykonfiguration,



Abbildung 1.2.1.: Die vier Komponenten des Parameterraumes mit beispielhaften Einflussfaktoren.

Eingabemodalität, Interaktionsraum), **virtuelle Umgebung** (z.B. virtueller Inhalt, Interaktionsaufgabe, Interaktionstechnik) und der **Benutzer** (z.B. Motorik und Mobilität, Präferenzen). Um die Nutzereingabe aus realen Bewegungen im physischen Raum auf Bewegungen im virtuellen Raum für die Interaktion mit der Anwendungen abzubilden wird eine weitere Komponente benötigt – **Mapping**³.

Basierend auf der Gewichtung der Einflussfaktoren kann, angepasst an die gewünschte Präzision der Interaktion und dem von der physischen und virtuellen Umgebung unterstützten Detailgrad, eine geeignete Mappingmethode empfohlen werden.

BEITRAG: Es werden bekannte Navigations- und Manipulationstechniken erweitert, welche die distanzabhängige Interaktion des konzipierten Parameterraumes verwenden. In Experimenten wird der Einfluss bisher wenig evaluierter Benutzereigenschaften auf Faktoren der physischen Umgebung und das Mapping untersucht. Aufbauend auf den Ergebnissen der Nutzerstudien werden Richtlinien für die distanzabhängige Interaktion in GHD-Umgebungen empfohlen.

Das hier vorgestellte Konzept zur distanzabhängigen Interaktion in GHD-Umgebungen wurde im speziellen für Visualisierungs- und *Fullscreen*-Anwendungen entwickelt, in denen hauptsächliche Navigations- und Manipulationsaufgaben ausgeführt werden. Bei *Fullscreen*-Anwendungen wird die gesamte Displayfläche von einer Anwendung verwendet und die Nutzerinteraktion erfolgt innerhalb der gegenwärtigen Anwendung. Die vorliegende Arbeit betrachtet keine erweiterten Desktopumgebungen, in denen typischerweise Interaktionsaufgaben wie Symboleingabe, Systemkontrolle und Modellierung durchgeführt werden. Ein Überblick über die Usability Herausforderungen in erweiterten Desktopumgebungen findet sich in [RCB⁺05, CRM⁺06], wo Probleme wie *losing the cursor, window and task management* und *GUI interaction* diskutiert werden. In dieser Arbeit wird nicht auf stereo-

³ In der Mensch-Computer-Interaktion wird die Abbildungsfunktion von physischer Eingabe auf virtuelle Ausgabe auch als Transition (oder Transition-Funktion) bezeichnet.

fähige, große hochauflösende Displays eingegangen. Das erarbeitete Konzept kann daher nur bedingt auf große Stereodisplays angewendet werden, da einige physiologische Aspekte (z.B. Stereo-Sehen) und technologische Aspekte (z.B. Stereo-Rendering) im Parameterraum nicht berücksichtigt werden. Beispielsweise beeinflusst die Objektdarstellung (Positive/Negative Parallaxe) die Berechnung des Selektionspunktes beim Ray-Casting, so dass diese Einflussfaktoren die Parameter für das Mapping anders gewichten.

Das Konzept zur distanzabhängigen Interaktion in GHD-Umgebungen bezieht sich auf die verfügbaren Freiheitsgrade der physischen Umgebung (Eingabe) und virtuellen Umgebung (Ausgabe), weshalb es auf verschiedene Eingabemodalitäten anwendbar ist.

1.3. Struktur der Arbeit

Nach diesem Einführungskapitel ist die vorliegende Arbeit wie folgt aufgebaut.

Kapitel 2.1 und 2.2 Grundlagen: Zunächst werden die physiologischen Eigenschaften von Nutzern erörtert, welche bei der Interaktion in diesen speziellen Displayumgebungen berücksichtigt werden müssen, wie die visuelle Wahrnehmung und Motorik des Menschen. Bei der Interaktion mit VR-Systemen sind Nutzerstudien unverzichtbar. Es werden einige Grundlagen der Statistik bei Nutzerstudien in VR kurz erläutert, um die Evaluierungen der vorausgehenden Forschungsarbeiten und der eigenen Studien einfacher nachvollziehen zu können.

Kapitel 2.3–2.5 Stand der Forschung: Anschließend werden die Eigenschaften von großen Displayumgebungen betrachtet, die Auswirkungen auf die Darstellung und die Interaktion mit Informationen haben. Es werden verwandte Arbeiten zu Interaktion in GHD-Umgebungen diskutiert und auf welchen Ergebnissen, innerhalb dieser Arbeit, aufgebaut wird. Hier wird insbesondere auf die Problematik der Abbildung von Eingabemodalität auf Ausgabefunktionalität eingegangen. Das Kapitel schließt mit einer Anforderungsanalyse zu distanzabhängiger Interaktion in großen hochauflösenden Displayumgebungen.

Kapitel 3 Konzept der distanzabhängigen Interaktion: Ausgehend von den untersuchten Arbeiten werden die verschiedenen Einflussfaktoren auf die Interaktion in großen Displayumgebungen aufgezeigt und in den konzipierten Parameterraum eingeordnet. Anschließend werden verschiedene Kombinationen der Parameter an Beispielszenarien für die distanzabhängige Informationsdarstellung und die distanzabhängige Präzision der Interaktion diskutiert.

Kapitel 4 Experimentalumgebung: Die verwendete Experimentalumgebung für die durchgeführten Evaluierungen wird in diesem Kapitel beschrieben. Dabei wird

auf die Einschränkungen der verwendeten hochauflösenden Displayumgebung eingegangen, die einige Designentscheidungen für die Nutzerstudien begründet.

Kapitel 5 Umsetzung der distanzabhängigen Interaktion: Die Realisierung des Konzeptes wird anhand der beschriebenen Beispielszenarien erläutert. Für die Umsetzung der distanzabhängigen Navigations- und Manipulationstechniken werden einige Faktoren des Parameterraumes durch die verwendeten Experimentalumgebung vorgegeben.

Kapitel 6 Evaluierung der distanzabhängigen Interaktion: In Pilotstudien wurden ausgesuchte Kombinationen der Faktoren im Parameterraum evaluiert. Der Fokus liegt auf der Anwendung des Konzeptes für Manipulationsaufgaben, die die distanzabhängigen Interaktionsgenauigkeiten unterstützen sollen und bisher nur unzureichend untersucht wurden. In einer größeren Folgestudie wurden zwei Manipulationstechniken mit einem automatischen und einem manuellen Wechsel der Interaktionsgenauigkeiten evaluiert. In dem Experiment wurde auf die physische Navigation (Mobilität und Motorik) bei Nutzern während der Manipulation eingegangen.

Kapitel 7 Schlussbetrachtung: In der Schlussbetrachtung werden die zuvor gestellten Forschungsfragen beantwortet und in wie weit das vorgestellte Konzept in anderen Displayumgebungen anwendbar ist. Außerdem werden die Grenzen des Parameterraumes aufgezeigt und im Ausblick Anregungen für weiterführende Arbeiten gegeben.

2. Grundlagen und Stand der Forschung

Große hochauflösende Displayumgebungen haben spezielle Anforderungen an die Interaktion, d.h. welche Eingabegeräte kann der Nutzer verwenden und welche Interaktionstechniken sind geeignet. Um die Eingabemodalitäten in großen Displayumgebungen zu diskutieren, müssen zunächst die physiologischen Eigenschaften von Nutzern erörtert werden. Hier wird auf die Wahrnehmung und Motorik des Menschen eingegangen, welche die physische Navigation und Interaktion von Nutzern in GHD-Umgebungen beeinflussen. In Form von Nutzerstudien wird die Gestaltung von interaktiven Systemen evaluiert. Daher wird ein kurzer Überblick über Evaluierungsmethoden in GHD-Umgebungen gegeben, welche in den eigenen Nutzerstudien angewendet wurden.

Anschließend werden die Eigenschaften von großen Displayumgebungen betrachtet, die Auswirkungen auf die Darstellung von Informationen und die Interaktion mit Informationen haben. Es werden verwandte Arbeiten diskutiert und bezüglich der verwendeten Interaktionsaufgabe und der Mappingmethode klassifiziert, um offene Punkte bei der Interaktion in GHD-Umgebungen aufzuzeigen.

Das Kapitel schließt mit einer Anforderungsanalyse zu Interaktion in großen hochauflösenden Displayumgebungen, welche den konzipierten Parameterraum für distanzabhängige Interaktion in GHD-Umgebungen einleitet.

2.1. Physiologische Eigenschaften des Menschen

Bei der Interaktion in großen hochauflösenden Displayumgebungen spielt die visuelle Wahrnehmung und die physische Navigation (Mobilität und Motorik) des Menschen eine wichtige Rolle. Je nach Abstand des Nutzers zum Display kann er unterschiedliche Detailstufen wahrnehmen (z.B. Objekte, Schriftzeichen oder einzelne Farbpunkte). Dies sollte bei der visuellen Darstellung berücksichtigt werden, um die verfügbare Displayfläche optimal zu nutzen. Da der Nutzer sich in GHD-Umgebungen frei bewegt, um Informationen in unterschiedlichen Detailstufen zu sehen, muss die Interaktion mit den Informationen aus verschiedenen Distanzen zum Display unterstützt werden. Dennoch sollte ein zu hoher körperlicher Aufwand für den Nutzer vermieden werden, welcher zu Ermüdung und Frustration führt.

2.1.1. Visuelle Wahrnehmung

Colin Ware hat in [War04] das menschliche Auge mit der Funktionsweise einer Fotokamera verglichen: Es enthält Äquivalente für die Linse, die Blende (Pupille) und einen Film (Retina); die Linse fokussiert ein kleines, umgekehrtes Bild der Welt auf der Retina; die Iris führt die Funktion einer variablen Blende aus und hilft dem Auge sich unterschiedlichen Lichtbedingungen anzupassen.

Das menschliche Sehen kann in foveales Sehen und peripheres Sehen unterschieden werden [Pre99]. Das foveale Sehen (oder zentrale Sehfeld) ist farbsensitiv, verfügt über hohe visuelle Sehschärfe und funktioniert am besten unter hoher Beleuchtungsstärke. Im Gegensatz dazu ist das periphere Sehen relativ farbungempfindlich, hat eine schlechte räumliche Sehschärfe und ist sensitiver bei niedriger Beleuchtungsstärke.

Die Sichtbereiche des linken und rechten Auges überlappen sich und bilden zusammen das Sehfeld, d.h. Objekte innerhalb des Sehfeldes sind für jedes Auge sichtbar allerdings aus unterschiedlichen Winkeln. Beide Augen zusammen bieten ein Sehfeld von etwas mehr als 180 Grad, wobei etwa 120 Grad binokular wahrgenommen werden [War04]. Das scharfe Sehen mit hoher Auflösung beschränkt sich jedoch auf einen kleinen Bereich des Sehfeldes (1-2 Grad) und die Auflösung nimmt außerhalb dieses Bereiches stark ab. Das Sehfeld wird durch Augenbewegung abgetastet, so dass ein größerer Sichtbereich vom Menschen als "scharf gesehen" wahrgenommen wird. In der Peripherie ist die Empfindlichkeit gegenüber beweglichen Objekten höher als bei statischen Objekten.

Der Sehwinkel (*visual angle*) gibt an wie viele sichtbare Objekte im Sehfeld auftauchen und die Sehschärfe (*visual acuity*) ist die Fähigkeit des Menschen Details wahrzunehmen [DFAB04]. Der Sehwinkel wird üblicherweise in Grad, Bogenminuten und Bogensekunden definiert¹. Beispielsweise hat ein 1 cm großes Objekt aus 57 cm Entfernung betrachtet einen Sehwinkel von ungefähr 1 Grad [War04]. Für die Berechnung des Sehwinkels θ mit der Betrachtungsdistanz d und der Objektgröße h kann die Gleichung 2.1.1 verwendet werden (vgl. Abbildung 2.1.1).

$$\theta = 2 \arctan \left(\frac{h}{2d} \right) \quad (2.1.1)$$

Die Sehschärfe wird mit Hilfe einer Sehprobentafel ermittelt, beispielsweise mit dem *Snellen-Index* bei einer Sehdistanz von sechs Metern.

Das Farbsehen ist die Fähigkeit Unterschiede in den Wellenlängen des Lichtes wahrzunehmen und wird mit einem *Ishihara chart* getestet. Dabei handelt es sich um ein Diagramm mit unterschiedlichen Farbpunkten, die so angeordnet sind, dass sie Zahlen bilden, die für "normale" und "farbblinde" Augen abweichen.

Ausgehend von der Sehschärfe kann die maximale Betrachtungsdistanz berechnet werden, bei der einzelne Objekte getrennt wahrgenommen werden. Beispiels-

¹ Eine Bogenminute entspricht 1/60 Grad und eine Bogensekunde ist 1/60 Bogenminute.

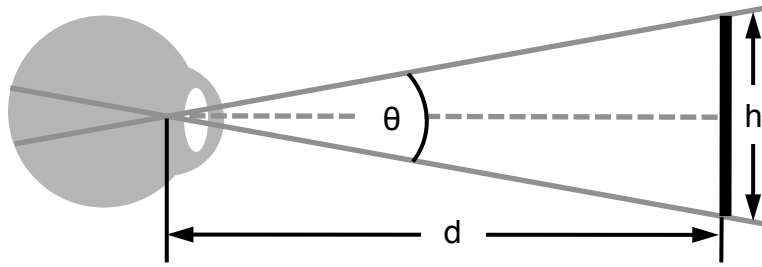


Abbildung 2.1.1.: Visualisierung der Berechnung des Sehwinkels eines Objektes (nach [War04]).

weise erfolgt die Berechnung des maximalen Betrachtungsdistanz d , um zwischen individuellen Pixeln mit dem Pixelabstand h unterscheiden zu können bei normaler Sehschärfe (100% Sehstärke, Sehwinkel $\theta = 1/60^\circ$) mit der Gleichung 2.1.2 [Kön10].

$$\text{Betrachtungsdistanz } d = \frac{\text{Pixelabstand } h}{2 \tan\left(\frac{\theta}{2}\right)} = \frac{\text{Pixelabstand } h}{2 \tan\left(\frac{1}{120}\right)} \quad (2.1.2)$$

Für die Lesbarkeit von lateinischen Schriftzeichen wird eine Zeichenhöhe von 20 Bogenminuten bis 22 Bogenminuten für die Darstellung auf einem Display empfohlen (nach [DIN12, ISO 9241-303:2012]).

2.1.2. Motorik

Neben der visuellen Wahrnehmung bewirkt auch die menschliche Motorik einige Einschränkungen für die Gestaltung und Verwendung der Interaktionsmöglichkeiten des Nutzers mit virtuellen Umgebungen.

Die Bewegungswissenschaft oder Bewegungslehre (*kinesiology*) untersucht die Anatomie, die Physiologie und die Mechanik der Bewegungen des menschlichen Körpers [Saf08]. Interessanterweise können sich Muskeln nur zusammenziehen, jedoch nicht "selbstständig" dehnen [Saf08]. Das heißt, die Dehnung eines Muskels (Antagonist) muss durch zusammenziehen des Gegenspielers (Agonist) erfolgen. Beispielsweise kann der Mensch die Wadenmuskulatur durch das Zusammenziehen der Schienbeinmuskulatur gezielt dehnen.

Im menschlichen Körper gibt es verschiedene Gelenkformen, deren Beweglichkeit anatomisch unterschiedlich beschränkt ist. Dies muss bei der Verwendung der Eingabemodalität (z.B. Armbewegungen) beachtet werden. Das Kugelgelenk (z.B. Schultergelenk) lässt einen großen Bewegungsradius zu, während das Scharniergelenk (z.B. Ellenbogengelenk) nur in einer Ebene vor- und zurückbewegt werden kann. Das Sattelgelenk (z.B. Wurzelgelenk der Daumen) lässt Bewegungen in zwei Freiheitsgraden zu mittels Beugung/Streckung und Abspreizen/Heranziehen. Die unterschiedlichen Bewegungen der Körperteile werden oft in Paaren bezeichnet; die Ausführung und die Gegenbewegung, welche den Körper oder die Körperteile

in die neutrale Position zurückführt [Saf08]. Typische Bewegungsformen sind beispielsweise Abduktion — Adduktion (Abspreizen/Heranziehen), Außenrotation — Innenrotation und Extension — Flexion (Streckung/ Beugung). Ein Überblick über die unterschiedlichen Bewegungen und Gelenkformen der menschlichen Motorik ist in [Saf08] zu finden.

Neben der anatomischen Betrachtung der Beweglichkeit des Menschen wird bei der ergonomischen Betrachtung der Bewegung zusätzlich der Komfort für den Nutzer berücksichtigt. Die Ergonomie liefert Einschränkungen für die Verwendung von Posen und Gesten, z.B. sollte das Verharren in festen Posen vermieden werden und die Gelenke sollten nicht zu weit von ihrer neutralen Position bewegt werden [NSMG03]. Nielsen et al. haben sechs Richtlinien für die Ergonomie von Bewegungen bei der Mensch-Computer-Interaktion identifiziert [NSMG03, Saf08]: (1) Vermeidung von Extremposition (z.B. Überdehnungen oder starke Dehnungen), (2) Vermeidung von Wiederholung, (3) Entspannung von Muskeln, (4) Verwendung von entspannten und neutralen Positionen, (5) Vermeidung von Verharren in festen Positionen und (6) Vermeidung von inneren und äußeren Druck auf die Gelenke.

Koordination der Finger

Weiterhin gibt es individuelle motorische Unterschiede zwischen den Nutzern. Als Beispiel dient hier die Bewegung der Finger: Für einige Personen ist es schwierig, ausgehend von einer geschlossenen Handfläche den kleinen Finger separat abzuspreizen. Meist wird der Ringfinger ungewollt mit abgespreizt. Diese Fingerbewegungen können zwischen der rechten und linken Hand eines Nutzers unter Umständen variieren. Untersuchungen zu ungewollter Bewegung von benachbarten Fingern wurde von Häger-Ross und Schieber [HRS00] durchgeführt. Sie erstellten einen *individuation index* für die einzelnen Finger; die Rangfolge beschreibt wie individuell einzelne Finger gezielt bewegt werden können ohne ungewolltes Mitbewegen benachbarter Finger. Der Daumen hat den höchsten *individuation index*, gefolgt vom Zeigefinger, dem kleinen Finger, Mittelfinger und dem Ringfinger. Der *individuation index* des Mittel-, Ring- und kleinen Fingers kann zwischen einzelnen Personen leicht variieren, ebenso zwischen der rechten und linken Hand einer Person. Die Autoren konnten keinen generellen Unterschied zwischen dominanter und nicht-dominanter Hand feststellen. Ingram et al. [IKHW08] führten ein ähnliches Experiment durch, bei dem natürliche Hand- und Fingerbewegungen im Alltagsleben untersucht wurden. Sie konnten zeigen, dass die Rangfolge für unabhängige Bewegung der Finger unter Laborbedingungen (nach [HRS00]) und bei Alltagsbewegungen gleich sind.

Die Kenntnisse über die motorische Steuerung individueller Finger ist für Interaktionsdesigner wichtig, da diese Information beispielsweise Einfluss auf die Anordnung von Buttons auf Eingabegeräten sowie den Einsatz von Gestensteuerung als Interaktionsform hat.

Hand-Augen-Koordination

Die physischen Eigenschaften eines Gerätes, wie das Fühlen, dass ein Button am Eingabegerät gedrückt wird, spielt ebenfalls beim Design von Eingabegeräten eine wichtige Rolle. Mit Hilfe der Sensormotorik, d.h. der innerlichen Wahrnehmung der Körperhaltung, -bewegung und Muskelverspannungen, können Nutzer fühlen wie sie ein Eingabegerät bewegen, ohne es dabei zu sehen [HW12]. Dieses Zusammenwirken von visueller Wahrnehmung und Motorik wird als Hand-Augen-Koordination des Nutzers bezeichnet. Um die Hand-Augen-Koordination zu unterstützen sollte daher das Mapping zwischen physischer Eingabe und visueller Ausgabe konsistent sein. Das heißt, bei Handbewegungen mit dem Eingabegerät im Uhrzeigersinn sollte die Rotation des visuellen Objektes auf dem Bildschirm ebenfalls im Uhrzeigersinn erfolgen [FWC84, PWF00]. Nutzer können jedoch auch einfache Abweichungen adaptieren; beispielsweise wenn der Nutzer eine Scrollbar abwärts bewegt, rollt der Text aufwärts [HW12]. Ein Störfaktor für die Hand-Augen-Koordination ist Latenz – die zeitliche Verzögerung zwischen Nutzereingabe und Sinneswahrnehmung des Nutzers. Latenz hat negative Auswirkungen auf die Nutzerperformanz, da es einen Widerspruch zwischen der internen Rückmeldung (z.B. körpereigene Reizwahrnehmung durch Steuerung des Eingabegerätes) und der externen Sinneswahrnehmung (z.B. visuelles Feedback vom System) gibt [BKLP04].

Koordination der Hände

Bei koordinierten und getrennten Aufgaben ist der Mensch fähig, die Objekte mit beiden Händen zu bedienen, z.B. Autofahren, Flugzeug fliegen oder ein Musikinstrument spielen [Jac97, BM86].

Die theoretische Grundlage für die meisten beidhändigen Interaktionstechniken liefert die Arbeit von Guiard, der die Arbeitsverteilung zwischen dominanter und nicht-dominanter Hand des Menschen untersucht hat [Gui87]. Er hat drei Klassen von beidhändiger Aktion beobachtet: (1) grundsätzliche einhändige Interaktion (z.B. Dartpfeile werfen), (2) beidhändige symmetrische Interaktion – wo beide Hände die gleiche Aktion synchron oder asynchron ausführen (z.B. Gewichtheben, Tastatureingabe) und (3) die beidhändige asymmetrische Interaktion – wo die Hände unterschiedliche aber aufeinander abgestimmte Aktionen durchführen. Für die asymmetrische Arbeitsteilung der Hände hat Guiard folgende drei Richtlinien für die beidhändige Interaktion vorgeschlagen [Gui87, BKLP04], die im Folgenden als Guiard's Framework bezeichnet werden:

- (1) Die nicht-dominante Hand bestimmt dynamisch das räumliche Bezugssystem für die Aktionen der dominanten Hand.
- (2) Mit der dominanten Hand lassen sich feingranulare präzise Interaktionen ausführen, während mit der nicht-dominanten Hand grobe Manipulationen durchgeführt werden.

(3) Die Manipulation wird durch die nicht-dominante Hand initiiert. Beispielsweise kontrolliert die nicht-dominante Hand die Lage des Notizzettels, um ein bequemes und effektives Schreiben mit der dominanten Hand zu ermöglichen.

Die Arbeitsteilung zwischen den Händen lässt sich bei der beidhändigen Manipulation jedoch nicht so einfach trennen, da die Interaktion mit den Händen ein dynamischer Prozess ist und der Nutzer bei komplexen Aufgaben zwischen der symmetrischen und asymmetrischen Manipulation hin und her wechselt [BKLP04].

Leganchuk et al. [LZB98] beobachteten, dass beidhändige Manipulation sowohl manuelle als auch kognitive Vorteile für die Mensch-Computer-Interaktion bietet. Da doppelt so viele Freiheitsgrade dem Nutzer gleichzeitig zur Verfügung stehen wächst die Zeit-Bewegungs-Effizienz bei der beidhändigen Interaktion, im Gegensatz zur einhändigen Interaktion. Kognitive Vorteile zeigen sich dadurch, dass die Beanspruchung des gedanklichen Komponierens sich verringert und die Aufgabe auf einem niedrigen mentalen Level visualisiert wird.

Hinckley et al. [HPPK98] konnten in Experimenten zeigen, dass eine hierarchische Struktur bei der beidhändigen Manipulation existiert; mit der dominanten Hand werden die Bewegungen ausgedrückt, relativ zur nicht-dominanten Hand (dem räumlichen Bezugssystem). Angelehnt an Guiard's räumliche Referenz-Richtlinie (siehe Guiard's Framework) konnten die Autoren die Hypothese widerlegen, dass die Hände parallele und unabhängige Eingabeströme sind, insbesondere bei präziser Manipulation. Die Verwendung beider Hände hilft den Nutzern, sich selbst in einem Interaktionsraum relativ zum Körper zu verorten. Außerdem wechselt die Reihenfolge der durchgeführten händischen Aufgaben, d.h. zweihändig kombinierte Aufgaben sind nicht das Gleiche wie die aufeinanderfolgende Kombination einhändiger Teilaufgaben. Diese Tatsache beeinflusst das Problemlöseverhalten der Nutzer, dass wiederum einen direkten Einfluss darauf hat, wie Nutzer über eine Aufgabe nachdenken [HPPK98].

Tremor

Bei der freihändigen Interaktion im realen Raum ist das natürliche Handzittern des Menschen ein Problem bei der präzisen Nutzereingabe. Der Mensch kann seine Hände nicht exakt still halten, weil seine Hände physiologisch bedingt stets kleine ungewollte Bewegungen ausführen (Tremor). Das physiologische Handzittern ist individuell stark ausgeprägt zwischen den Personen und variiert auch zeitweise bei jeder Person, des Weiteren nimmt es mit dem Alter zu [Lak10]. Dieser Tremor erschwert es Nutzern, eine präzise Manipulation mit freihändigen Eingabegeräten (z.B. Laserpointer) durchzuführen. Um beispielsweise ein Objekt zu selektieren ist es schwierig mit dem Gerät auf eine exakte Position zu zeigen und gleichzeitig einen Button auf dem Gerät zu drücken ohne dabei das Gerät ungewollt zu bewegen. Myers et al. [MBN⁺02] konnten in einer Studie mit unterschiedlichen La-

serpointern zeigen, dass es den Probanden nicht möglich war: (1) direkt mit dem Laserstrahl dahin zu zeigen, wo sie wollten; (2) den Laserstrahl still zu halten, wie sie wollten; (3) den Laserstrahl an der aktuellen Position per Tastendruck abzuschalten. Die Autoren stellten fest, dass es mindestens eine Sekunde dauerte bis der Laserstrahl an der gewünschten Stelle war.

Weitere physikalische Eigenschaften von Eingabegeräten, wie Gewicht, Form und Material, müssen aus ergonomischer Sicht bei der Interaktion berücksichtigt werden [FWC84, Zha98, HW12]. Einerseits können etwas schwerere Eingabegeräte das Handzittern reduzieren und damit eine höhere Manipulationspräzision ermöglichen, andererseits führen diese Geräte dann schneller zu Ermüdung und Überanstrengung des Nutzers.

Es gibt Interaktionstechniken, welche Filtertechniken oder *control-display-gain* Ansätze verwenden, die den Effekt des Tremors minimieren. Diese werden im Abschnitt 2.4.1.4 Mapping ausführlicher betrachtet.

Fitts' Law [Fit54] ist ein Modell für das menschliche psychomotorische Verhalten und wird für motorische Performanzbewertungen in der Mensch-Computer-Interaktion eingesetzt [Mac92, Seo05]. *Fitts' Law* prognostiziert die Bewegungszeit basierend auf der Bewegungsamplitude und der Breite des Zielbereiches. Card et al. [CEB78] haben gezeigt, dass das gleiche Modell mit ungefähr den gleichen Parametern sowohl Bewegungen mit der Maus als Eingabegerät als auch Bewegungen alleine mit der Hand abbildet. Das Versuchsparadigma wird darüber hinaus für den Vergleich und zur Optimierung von Zeigeräten (*pointing devices*) eingesetzt, indem gemessen wird, wie effektiv der Nutzer mit dem Zeigegerät ein Zielobjekt auf dem Bildschirm auswählen kann [HW12]. Es wurde beispielsweise zur Evaluierung von Interaktionstechniken wie *Dual-Precision Techniques* [NPBL11], *Adaptive Pointing* [KGDR09], *HybridPointing* [FVB06] oder *Semantic Pointing* [BGBL04] eingesetzt (siehe Abschnitt 2.4.2).

Guiard et al. [GBLB⁺04] stellen das Konzept des *multi-scale pointing* vor, bei dem neben dem herkömmlichen *cursor pointing* (nach *Fitts' Law*) auch das *view pointing* bei der motorischen Performanzbewertung berücksichtigt wird. Dies ist beispielsweise in großen Dokumenten notwendig, wenn das Zielobjekt außerhalb des Sichtbereiches liegt, muss zuerst der Sichtbereich bewegt werden um das Zielobjekt sichtbar zu machen (*view pointing*) und anschließend kann das Zielobjekt selektiert werden (*cursor pointing*). Kopper et al. [KBSM10] entwickelten ein *motoric model*, basierend auf *Fitts' Law*, bei dem die Distanz des Nutzers zum Display beim Zeigen (*pointing*) eingerechnet wird. Damit lassen sich Zeigetechniken aus der Entfernung besser untereinander und auch mit anderen Techniken (z.B. mausbasierte Techniken) vergleichen.

Nutzergruppen

Bei der Entwicklung von Eingabegeräten und Interaktionstechniken müssen Designer neben den physiologischen Limitierungen der einzelnen Nutzer auch folgende Aspekte der Nutzergruppen beachten, wie Händigkeit, Alter und Körpergröße [BKLP04]. Viele Eingabegeräte wurden (nur) für Rechtshänder entwickelt. Jedoch sollten asymmetrische beidhändige Interfaces für Rechts- und Linkshänder adaptierbar sein. Kinder benötigen oft ein anderes Interfacedesign als Erwachsene, weil sie physisch kleiner sind, eine kürzere Aufmerksamkeitsspanne haben und sich ihre mentalen Modelle über die Interaktion von denen der Erwachsenen unterscheiden. Ältere Nutzer benötigen möglicherweise die Textdarstellung in einer größeren Schriftgröße oder sie können unter Umständen keine schnellen Bewegungen beim Interagieren durchführen. Ebenso kann die Körpergröße bei einigen Interaktionstechniken eine Rolle spielen. Beispielsweise werden bei der *Go-Go*-Technik [PBWI96] naheliegende oder entfernte Objekte in der virtuellen Umgebung abhängig von der Streckung des Nutzerarmes manipuliert. Die Technik verwendet dabei einen festen Schwellenwert für alle Nutzer, um zwischen Nahbereich und Fernbereich zu unterscheiden. Somit kann es passieren, dass Nutzer mit kurzen Armen entfernte Objekte nicht manipulieren können.

2.2. Evaluierung

Für die vorliegende Arbeit ist die Evaluierung des entwickelten Interaktionskonzeptes, in Form von Nutzerstudien notwendig, da die physiologischen und die psychologischen Eigenschaften der Nutzer die Interaktionsmöglichkeiten wesentlich beeinflussen. In diesem Abschnitt werden ausgewählte Kennwerte der Statistik und die Datenerhebung bei Nutzerstudien im VR-Bereich kurz erläutert, die im Kapitel 6 verwendet werden. Die zugehörigen Definitionen der Kennwerte und die grundlegende Vorgehensweise der Signifikanzprüfung sind im Anhang B beschrieben. Für die Grundlagen der Evaluierung von Nutzerstudien wurde sich an dem Tutorial von Swan et al. [SAG09] orientiert. Für eine ausführliche Erläuterung der statistischen Methoden, sowie ihre Vor- und Nachteile, wird auf [Bor99b, RFHN06, Bel08] verwiesen.

2.2.1. Statistische Methoden

Statistik lässt sich in zwei Arten unterteilen, die deskriptive (oder beschreibende) Statistik und die Inferenzstatistik (oder schließende Statistik).

Bei der **deskriptiven Statistik** werden numerische und nicht-numerische Daten gesammelt und in Form von Zusammenfassungen, Tabellen und Graphen wiedergegeben.

“Wie können die empirischen Daten aufbereitet, dargestellt, zusammengefasst und strukturiert werden, so dass zentrale Merkmale sichtbar werden?“²

Die **Inferenzstatistik** analysiert und interpretiert die gesammelten Daten.

“Wie können auf Basis empirischer Daten (Wahrscheinlichkeits-) Aussagen über die Allgemeinheit getroffen werden?“²

Da man nicht die ganze Population (Grundgesamtheit) bezüglich eines Merkmals testen kann, wird eine **Stichprobe** aus der Population genommen. Mit Methoden der Inferenzstatistik wird das Ergebnis der Stichprobe auf die Population erweitert und die Glaubwürdigkeit der Ergebnisse gemessen. Im Gegensatz zur Stichprobe handelt es sich beim **Kollektiv** um eine Personengruppe, die nur beschrieben werden soll und für die sich die Frage der Repräsentativität hinsichtlich der Grundgesamtheit nicht stellt [Bor99b].

Daten: Die erhobenen Daten werden durch Merkmale und deren Ausprägungen beschrieben. Daten können in zwei Arten von Variablen unterschieden werden: qualitative und quantitative Daten.

² Zitat aus dem Foliensatz http://www.techfak.uni-bielefeld.de/~skopp/Lehre/MMI_WS1112/MMI_8.pdf

Skalenniveau	Statistische Kennwerte
Nominal	Modalwert
Ordinal	Modalwert, Median, Varianzbreite
Intervall, Verhältnis	Modalwert, Median, arithmetischer Mittelwert, Varianzbreite, mittlere Abweichung, Varianz und Streuung

Tabelle 2.2.1.: Übersicht der mathematischen Operationen für die Skalenniveaus.

Qualitative Daten sind nicht-numerische Daten (z.B. Haarfarbe und Wohnsitz). Quantitative Daten sind numerische Daten (z.B. Körpergröße, Alter), die sich in diskreten und kontinuierlichen Variablen unterteilen lassen. Diskrete quantitative Daten sind Ergebnisse einer Zählung (z.B. Anzahl der Tastendrucke), wohingegen kontinuierlich quantitative Daten Ergebnisse einer Messung sind (z.B. Bearbeitungszeit).

Skalenniveau: Des Weiteren werden die qualitativen und quantitativen Variablen nach ihrem Skalenniveau charakterisiert. Es wird in folgende Skalenniveaus unterschieden: Nominalskala (gleiche Merkmalsausprägung, gleiche Werte), Ordinalskala (größere Merkmalsausprägungen, größere Werte), Intervallskala (Größenordnung von Wertdifferenzen entspricht den Merkmalsunterschieden) und Verhältnisskala (Größenverhältnisse der Werte entspricht der Merkmalsausprägung). Abhängig von dem Skalenniveau sind unterschiedliche mathematische Operationen auf den Messwerten definiert (siehe Tabelle 2.2.1).

Die Variablen können unterschiedliche Funktionen in der Untersuchung haben. Grundsätzlich wird zwischen abhängigen Variablen (aV) und unabhängigen Variablen (uV) unterschieden.

Die unabhängigen Variablen beschreiben den Effekt, der untersucht werden soll. Daher werden die unterschiedlichen Ausprägungen der unabhängigen Variable im Experiment gezielt erzeugt und systematisch variiert.

Die abhängigen Variablen sind die gemessenen Variablen, deren Wirkung von den unabhängigen Variablen abhängt.

Die Störvariablen sind Faktoren, die nicht untersucht werden, aber das Experiment beeinflussen. Daher ist es wichtig diese Störvariablen vorherzusehen und zu kontrollieren.

Deskriptive Statistik

Die statistischen Kennwerte geben über spezielle Eigenschaften eines Datenkollektivs oder einer Merkmalsverteilung zusammenfassend Auskunft. Dabei wird zwischen den Kennwerten der zentralen Tendenz, die alle Messwerte einer Verteilung zusammenfassend repräsentieren, und den Kennwerten der Dispersion, die

Auskunft über die Variation der Messwerte geben, unterschieden. Bei der deskriptiven Statistik erfolgt keine Interpretation der Daten oder Hypothesen der Untersuchung.

Die Darstellung der **Häufigkeitsverteilung** durch Kennwerte ist eine der grundlegenden Techniken. Dazu werden eine kategoriale Variable festgelegt und die Anzahl der einzelnen Merkmalsausprägungen bestimmt.

Die Häufigkeiten werden als absolute, relative oder kumulierte Werte angegeben. Die grafische Darstellung der Häufigkeitsverteilung einer kategorialen Variable kann beispielsweise in einem Histogramm oder Balkendiagramm erfolgen.

Die gebräuchlichsten **Maße der zentralen Tendenz** sind: der Modalwert, der Medianwert (Median) und das arithmetische Mittel.

Zwei Verteilungen können sich hinsichtlich ihrer zentralen Tendenz ähneln, obwohl die einzelnen Werte aufgrund ihrer Streuung stark voneinander abweichen. Die **Kennwerte der Dispersion** beschreiben, wie stark die einzelnen Werte in einer Verteilung vom Mittelwert abweichen. Es werden üblicherweise folgende Kennwerte verwendet: Variationsbreite, mittlere Abweichung, Varianz und Standardabweichung.

Eine einfache grafische Darstellung der wesentlichen Kennwerte einer Verteilung bieten **Boxplots**, die einen schnellen Vergleich von Datensätzen ermöglichen. Ein Boxplot visualisiert für einen Datensatz den kleinsten und größten Wert (Minimum, Maximum) als sogenannte Whisker oder Antennen, den Median, sowie das untere Quartil (das 25. Perzentil) und das obere Quartil (das 75. Perzentil) als Box. Die Länge der Whisker wird auf das maximal 1,5-fache des Interquartilabstandes beschränkt, wobei die konkrete Darstellung der Länge an dem Datenwert, der noch innerhalb der Grenze liegt, festgelegt wird. Werte außerhalb der Whisker werden als Ausreißer im Diagramm dargestellt.

Inferenzstatistik

Basierend auf der Stichprobe kann mit Methoden der Inferenzstatistik geschlussfolgert werden, wie sich die Population bezüglich des gemessenen Parameters verhalten wird.

In VR-Nutzerstudien ist es schwierig, eine repräsentative Anzahl an Studienteilnehmern in einer Stichprobe zu akquirieren; oftmals ist nur einer begrenzten Anzahl an Personen der Zugang zu den Forschungslaboren möglich oder es werden Vorkenntnisse in der Benutzung bestimmter Techniken etc. benötigt. Daher werden meist sogenannte **Ad-hoc-Stichproben** erhoben, d.h. Personen schlicht nach Verfügbarkeit oder freiwilliger Teilnahme in die Untersuchung aufgenommen [BD06]. Die Ergebnisse sind daher weniger repräsentativ für die Grundgesamtheit. Basierend auf den Daten beziehungsweise Eigenschaften der Probanden erfolgt die Definition einer Grundgesamtheit bei Ad-hoc-Stichproben im Nachhinein.

Anhand der Kennwertverteilung der Stichprobe wird bestimmt, wie genau der Stichprobenwert den Populationswert wiedergibt. Die Streuung der Kennwertverteilung bestimmt wie gut ein einzelner Stichprobenwert (z.B. Stichprobenmittelwert \bar{x}) den unbekanntem Populationswert (z.B. Populationsmittelwert μ) schätzt. Diese Streuung der Kennwertverteilung wird als Standardfehler bezeichnet. Für die Mittelwerte aus Zufallsstichproben des Umfangs n einer Grundgesamtheit gilt, dass die Verteilung aller Stichprobenmittelwerte mit wachsendem Stichprobenumfang in eine Normalverteilung mit Mittelwert μ übergeht. Bereits bei einer Stichprobengröße $n \geq 30$ ist die Mittelwertverteilung für beliebige Verteilungen des Merkmals in der Population annähernd normalverteilt.

Bei den statistischen Aussagen werden empirisch ermittelte Kennwerte in Bezug zur Wahrscheinlichkeitsverteilung beurteilt, d.h. es wird von empirischen Daten auf unbekannte Populationskennwerte geschlossen. Beim **Signifikanztest** hingegen werden die Eigenschaften einer Population als Hypothese behauptet und anschließend geprüft, ob sich die Eigenschaften durch die Stichprobenergebnisse bestätigen lassen.

Hypothesen: Es wird zwischen zwei Varianten von Hypothesen unterschieden.

Die Unterschiedshypothesen postulieren Unterschiede zwischen Gruppen, während die Zusammenhangshypothesen Aussagen über Korrelationen zwischen Variablen treffen. Des Weiteren lassen sich Hypothesen als ungerichtet oder gerichtet und unspezifisch oder spezifisch charakterisieren.

Die Prüfung der Hypothese erfolgt durch den Vergleich zweier einander ausschließender Hypothesen H_1 und H_0 . Die **Alternativhypothese** H_1 beschreibt den vermuteten Unterschied zwischen den Gruppen beziehungsweise die vermutete Korrelation zwischen Variablen. Die **Nullhypothese** H_0 behauptet das Gegenteil, d.h. es gibt keinen Unterschied zwischen den Gruppen beziehungsweise keine Korrelation zwischen den Variablen. Die formulierten statistischen Hypothesen H_1 und H_0 geben Aussagen über den verwendeten statistischen Kennwert und dessen Ausprägung in der Population.

Bei dem Signifikanztest wird auf die Nullhypothese getestet. Daher muss bei akzeptierter Alternativhypothese die α -Fehlerwahrscheinlichkeit (Irrtumswahrscheinlichkeit) angegeben werden, die in Kauf genommen wird, wenn aufgrund der Ergebnisse die Nullhypothese verworfen wird. Für die Vergleichbarkeit und Qualität statistischer Untersuchungen wird eine Nullhypothese erst verworfen, wenn die Irrtumswahrscheinlichkeit $\leq 5\%$ (oder $\leq 1\%$) ist. Diese Schwellwerte werden als **Signifikanzniveau** (α -Niveau, α -Level) bezeichnet.

Unterschiedshypothesen werden im Allgemeinen mit Mittelwerts- oder Häufigkeitsvergleichen geprüft. Welches Verfahren für die statistische Signifikanzprüfung geeignet ist, hängt von verschiedenen Faktoren ab: dem Skalenniveau der Daten, den zu prüfenden statistischen Kennwerten und dem Design der Untersuchung.

In Kapitel 6 wird für die Signifikanzprüfung der Unterschiedshypothesen der t-Test oder die Varianzanalyse (*analysis of variance*, kurz ANOVA) verwendet. Für die Experimente wurde ein faktorielles Design der Untersuchung angewendet.

Beim faktoriellen Design wird eine Anzahl an unabhängigen Variablen festgelegt und deren Ausprägung (Bedingung, Level) gezielt im Experiment angewendet und deren Auswirkungen auf die abhängigen Variablen untersucht. Die einfachste Variante untersucht nur mehrere Bedingungen einer unabhängigen Variable, d.h. ein $n \times 1$ Design. Werden mehrere unabhängige Variablen evaluiert, dann handelt es sich um ein $n \times m$ Design, $n \times m \times p$ Design und so weiter. Die unabhängigen Variablen können auf zwei Arten variiert werden:

- (1) Jeder Proband sieht eine andere Ausprägung der Variable, d.h. er nimmt nur an einer Bedingung teil (*between-subject*);
- (2) Jeder Proband sieht alle Ausprägungen der Variable, d.h. er nimmt an allen Bedingungen teil (*within-subject*).

Beim *within-subject*-Design sind wichtige Störvariablen der Lerneffekt und die Ermüdung. Beim *between-subject*-Design können andere zeitbasierte Faktoren, wie Wetter oder Kalibrierungsfehler das VR-Experiment beeinflussen. Die Störvariablen können durch Randomisierung oder Ausbalancierung kontrolliert werden, indem die Bedingungen und deren Präsentationsreihenfolge für die Probanden unterschiedlich sind.

Für die statistische Auswertung wird beim *between-subject*-Design der t-Test für unabhängige Stichproben bzw. die Varianzanalyse (*independent ANOVA*) verwendet, während beim *within-subject*-Design der t-Test für abhängige Stichproben bzw. die Varianzanalyse mit Messwiederholungen (*repeated measurements ANOVA*) angewendet wird. Das allgemeine Vorgehen der Signifikanzprüfung mit Hilfe des t-Testes und der Varianzanalyse wird im Anhang B erläutert.

2.2.2. Evaluierungsmethoden

Evaluationen, insbesondere HCI-Studien³ bewerten die Usability und zeigen die Effekte der Interaktionstechnik auf die Nutzer. Mit Hilfe der Studien können Probleme und Hinweise für Verbesserungen aufgezeigt werden. Gängige Evaluationsmethoden für die Usability im VR sind Cognitive Walkthrough, heuristische Evaluation, formative Evaluation, summative Evaluation, Fragebögen, Interviews und Demos [BKLP04]. Im Rahmen dieser Arbeit wird die formative Evaluation verwendet, eine empirische Evaluationsmethode, die in verschiedenen Designstadien des Systems bzw. der Anwendung einsetzbar ist. Die formativen Evaluierungen reichen von der informalen Datenerhebung, bei der hauptsächlich qualitative Daten wie Nutzerkommentare und allgemeine Reaktionen gesammelt werden, bis zur

³ Studien im Bereich *human computer interaction* (HCI) bzw. Mensch-Computer-Interaktion untersuchen die benutzerfreundliche Gestaltung von interaktiven Systemen.

formalen und umfassenden Datenerhebung, bei der qualitative und quantitative Daten erfasst werden. Beispielsweise wird in einer Vorstudie eine Idee oder Hypothese mit einem kleineren Personenkreis untersucht, um das Design des Experimentes zu testen und mögliche Probleme oder Tendenzen aufzuzeigen. Anschließend wird das Experiment angepasst und in einer umfassenderen Benutzerstudie werden Daten aufgezeichnet, die für die statistische Auswertung und Bewertung der Interaktionstechnik erforderlich sind.

Um aussagekräftige Ergebnisse der Nutzerstudien zu Interaktion in GHD-Umgebungen zu erhalten muss im Vorfeld der Untersuchung folgende Frage geklärt werden:

Wie können die entwickelten Interaktionstechniken bewertet werden und welche Faktoren spielen eine Rolle?

Die Literaturrecherche hat gezeigt, dass die Interaktionstechniken durch die Messung objektiver Faktoren als auch durch die Bewertung subjektiver Faktoren evaluiert werden (vgl. [Hor06]). Boritz [Bor99a] nennt beispielsweise die Faktoren Genauigkeit, Geschwindigkeit, Prägnanz und Freude. Bei Zhai [Zha98] hingegen sind es Geschwindigkeit, Genauigkeit, Ermüdung und einfache Lernbarkeit. Während König [Kön10] die Kategorien Effektivität, Effizienz und Zufriedenheit beziehungsweise Nutzererfahrung wählt, die nach ISO 9241-11 die Usability bewerten.

Da die Usability nicht direkt gemessen werden kann, werden Maße genommen, die bestimmen wie (gut oder schlecht) ein bestimmter Nutzer ein bestimmtes Ziel effektiv, effizient und zufriedenstellend in einer definierten Umgebung erreicht [DIN99b, ISO 9241-11:1998]. Ein weiterer Faktor, der die Usability beeinflusst ist die Nutzerpräferenz, die oft als Indikator für die Effizienz gilt [NL94, FHH00]. Des Weiteren haben Ball et al. [BDN⁺06] festgestellt, dass Interaktionstechniken Eigenschaften wie geringer Lernaufwand (*easy to learn*), einfache Bedienbarkeit (*easy to use*) und Natürlichkeit (*natural usage*) haben sollten, damit sie effektiv sind.

Dementsprechend lassen sich folgende Faktoren für die Bewertung der Interaktionstechniken in Nutzerstudien identifizieren (angelehnt an [Kön10]):

- Effektivität (*effectiveness*): Genauigkeit und Präzision, Fehlerrate, effektive physische Navigation (Motorik und Mobilität);
- Effizienz (*efficiency*): Geschwindigkeit, Einarbeitungszeit, physiologische Aufwand (Wahrnehmung, Verarbeitung, Motorik);
- Zufriedenheit (*satisfaction*): Freude, Ermüdung, Präferenzen.

Die objektiven Faktoren können direkt oder indirekt während der Studie von beziehungsweise in der Experimentalumgebung aufgezeichnet werden (Testanwendung, weitere Aufzeichnungsgeräte). Die Geschwindigkeit ist meist als Bearbeitungszeit der Aufgabenstellung definiert und kann direkt von der Testanwendung gemessen werden. Die Geschwindigkeit, Genauigkeit und Fehlerrate werden als

Parameter für die Berechnung der Nutzerperformanz herangezogen. Aufgrund der physiologischen Eigenschaften des Menschen gibt es eine Abwägung zwischen Geschwindigkeit und Genauigkeit, die sich auf die Performanz auswirkt (vgl. Abschnitt 2.4.1.4).

Die physische Navigation kann beispielsweise mit Hilfe eines Trackingsystems aufgezeichnet werden und anschließend werden die motorischen und mobilen Bewegungsabläufe der Probanden miteinander verglichen.

Die subjektiven Faktoren, wie Zufriedenheit oder die Selbsteinschätzung des physiologischen Aufwandes können schwer gemessen werden. Mit Hilfe von Fragebögen werden daher Nutzermeinungen erfragt, aufgrund derer Rückschlüsse zur Zufriedenheit gezogen werden.

Die Fragebögen *Post-Study System Usability Questionnaire* (PSSUQ) und *After-Scenario Questionnaire* (ASQ) bieten standardisierte Messungen zur Zufriedenheit bei HCI-Studien [Lew93, Lew95]). Der PSSUQ umfasst insgesamt 19 Fragen, in denen die Nutzer ihre allgemeine Zufriedenheit, die Verwendbarkeit des Systems, die Informationsqualität und die Qualität des Interfaces bewerten. Lewis [Lew95] empfiehlt die Fragen zu erweitern, wenn die Evaluation es benötigt. Der ASQ umfasst insgesamt 3 Fragen zur allgemeinen Zufriedenheit mit einem System oder einer Interaktionstechnik. Die Fragebögen verwenden eine 7-Punkte Likert-Skala mit den Endpunkten *strongly agree* (Punktwertung 1) und *strongly disagree* (Punktwertung 7). Die Antworten einer Likert-Skala werden als ordinal- oder intervallskaliert betrachtet. Wenn eine Frage nicht beantwortet wurde, wird diese mit 0 bewertet und die Frage geht nicht in die Gesamtbewertung der Nutzerzufriedenheit für die Testperson ein. Die allgemeine Nutzerzufriedenheit mit einem System oder einer Interaktionstechnik wird dann als durchschnittliche Punktwertung aller Fragen berechnet. Für die Vorstudien und Benutzerstudie in Kapitel 6 wurden Fragen aus dem PSSUQ und ASQ entnommen und für die Experimente angepasst.

Andere Usability-Fragebögen, die in HCI-Studien eingesetzt werden sind beispielsweise *System Usability Scale* (SUS) [Bro96] und *Generic User Interface Questionnaire* (QUIS) [CDN88], die ähnliche Fragen wie Lewis [Lew95] verwenden.

Für die Selbsteinschätzung des physiologischen Aufwandes kann der *NASA Task Load Index* (NASA TLX) Fragebogen verwendet werden, der die Arbeitsbelastung während des Experimentes erfragt [Har06]. Hier bewerten Nutzer in einer 20-Punkte-Skala ihren Aufwand nach den Kategorien mentale/physische/zeitliche Anforderungen, allgemeine Performanz, Frustration und Anstrengung. Mit Hilfe des *NASA TLX Tool* können die Fragebögen ausgewertet werden.

2.3. Eigenschaften von großen hochauflösenden Displayumgebungen

In diesem Abschnitt werden die Vorteile für die Nutzerperformanz durch Erhöhen der Displaygröße und Auflösung diskutiert. Gleichzeitig muss berücksichtigt werden, welche physiologischen Einschränkungen (visuelle Wahrnehmung, Motorik) die Nutzerinteraktion in großen Displayumgebungen beeinflussen.

2.3.1. Displayauflösung und Displaygröße

Zunächst wird der Begriff "große hochauflösende Displays" (*large high-resolution displays*) im Rahmen dieser Arbeit kurz erläutert. Displays können nach ihrer physischen Größe (*small, large*) und nach ihrer Auflösung (*low-resolution, high-resolution*) klassifiziert werden (siehe Abbildung 2.3.2).

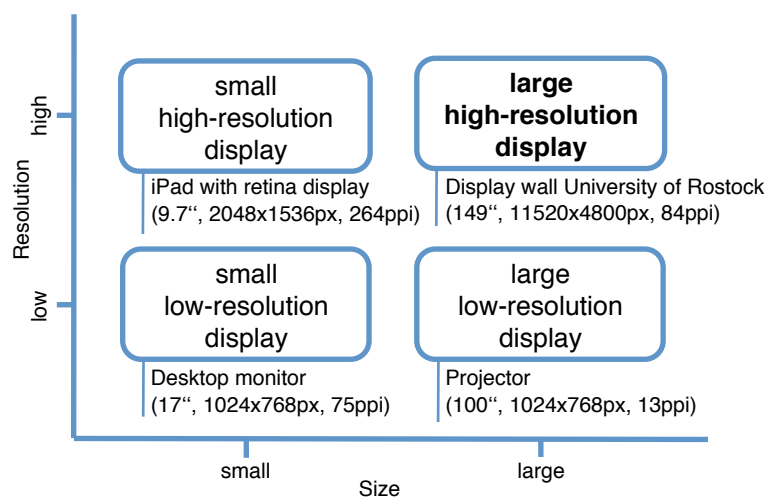


Abbildung 2.3.2.: Klassifizierung von Displays nach Displaygröße (Bild diagonale, *Size*) und Displayauflösung (Pixelanzahl, *Resolution*) mit Beispielen (angelehnt an [Kön10]).

Es ist relative einfach ein großes Display (*large display*) zu erzeugen – mittels einer weißen Wand und einem Projektor lässt sich eine physisch große Anzeigefläche projizieren. Dabei werden jedoch nur die Dimensionen der Anzeigefläche erhöht und nicht die Informationsmenge, die angezeigt wird (*large low-resolution display*). Wohingegen *large high-resolution displays* sowohl die Displaygröße erhöhen, als auch die Informationsmenge, die angezeigt werden kann [SADK⁺09].

Andrews et al. [AEYN11] erörtern, dass die Definition für große hochauflösende Displays einerseits aus technologischer Sicht erfolgen kann, d.h. diese Displays sind größer als traditionelle Displays, was aber eine subjektive Sicht ist. Andererseits kann die Definition über die Quantität der dargestellten Daten erfolgen. Andrews und Kollegen

[...] prefer to define the term 'large high-resolution display' as being a display that is human scale.

Damit sind Displays gemeint, deren physische Displaygröße und Displayauflösung (fast) dem Wahrnehmungsbereich und Einfluss des menschlichen Körpers entsprechen. Angelehnt an das Kriterium *human scale* bezeichnen wir als große hochauflösende Displays (GHDs) Displays mit großer Anzeigefläche und hoher Pixeldichte, die unter Umständen die visuelle Sehschärfe des Nutzers übersteigen können. Als große Anzeigefläche zählen einerseits Displays, die beispielsweise Personen und Objekte lebensgroß (*life-size*) darstellen können und andererseits Displays, die eine große Ausdehnung haben (*wall-size*). Bei diesen Displays ist es aus Nutzersicht nicht möglich nah am Display zu stehen und die komplette visuelle Darstellung zu sehen. Stattdessen muss der Nutzer einen gewissen Abstand einhalten, um die gesamte Darstellung "mit einem Blick" wahrnehmen zu können.

Weiterhin ermöglichen es GHD-Umgebungen, dass mehrere Personen gleichzeitig die Visualisierung auf dem Display betrachten können und somit die Kollaboration zwischen den Nutzern gefördert wird [BW00, GSW01].

Um die Displaygröße bei GHDs zu erweitern, gibt es aus technischer Sicht folgende zwei Ansätze (vgl. [NSS⁺06]):

Tiled LCD-Panels verwenden eine Reihe von LCD-Displays, die in einem variablen 2D Array angeordnet werden. LCD-Panels sind gegenüber Projektoren einfacher zu kalibrieren, preiswerter und benötigen weniger Platz (kein Projektionsabstand nötig). Jedoch befinden sich Monitorkanten (*Bezels*) zwischen den einzelnen LCDs innerhalb der gesamten Displayfläche, welche die visuelle Darstellung stören können [BBB10]. Die Bezels wurden aber in den letzten Jahren immer schmaler, so dass dieser Nachteil oft vernachlässigt werden kann.

Projector Arrays verwenden mehrere Projektoren, die sich an den Bildkanten überlagern und somit eine kontinuierliche Displayfläche ohne störenden Bezels erzeugen. Die Projektoren können, abhängig vom verfügbaren Platz variabel angeordnet werden. Der Vorteil einer Projektionswand gegenüber der Displaywand ist die Trennung zwischen Gerätegröße und Größe des dargestellten Bildes (Projektion). So kann ein kleiner Projektor sowohl ein großes visuelles Display als auch ein kleines visuelles Display projizieren. Die Nachteile sind der Kalibrierungsaufwand für die einzelnen Projektoren und deren Platzbedarf.

Großdisplays können unterschiedlich angeordnet sein und unterschiedliche Hardwarekonfigurationen haben (heterogene oder homogene Mehrdisplays). Die Anordnung der Displays kann planar, in Form einer Wand (z.B. Powerwall), oder bogenförmig, als Halbkreis (z.B. Panorama Screen) erfolgen. Weitere speziellen Anordnungen sind *L-shaped* Displays, würfelförmige (z.B. CAVE) oder kugelförmige (z.B. Dome) Displayumgebungen.

Die verfügbare Anzeigefläche und die Anordnung der Displays hat Einfluss auf die Usability und Performanz der Nutzer mit großen Displayumgebungen.

Als intuitive Interaktionsform in großen Displayumgebung spielt die physische Navigation eine immer größere Rolle.

Physische Navigation repräsentiert ein natürliches Verhalten von Nutzern, indem sie näher an die Displaywand heran gehen, um Details wahrnehmen zu können und sich von der Displaywand fortbewegen, um einen Überblick über die visuelle Darstellung zu erhalten. Die physische Navigation (Bewegung von Augen, Kopf, Körper) ist eine direkte und natürliche Interaktion und bedeutet weniger kognitive Belastung für den Nutzer.

Virtuelle Navigation passt den Blickwinkel auf die visuelle Darstellung an, indem der Nutzer sein Eingabegerät bewegt (z.B. *pan-and-zoom*). Somit muss der Nutzer sich nicht zu dem Bildausschnitt hinbewegen.

In den letzten Jahren gab es verschiedene empirische Untersuchungen zu Effizienz und Nutzen von großen hochauflösenden Displays, deren Ergebnisse im nächsten Abschnitt diskutiert werden.

2.3.2. Untersuchungen

Die Ergebnisse der Studien haben gezeigt, dass große hochauflösende Displayumgebungen die Nutzerperformanz bei Visualisierungs- und Manipulationsaufgaben positiv beeinflussen [CSR⁺03, TTLNS05, NBC06, SBY⁺06, BNB07] sowie kognitive Auswirkungen darauf haben, wie Nutzer Informationen wahrnehmen und mit diesen interagieren [TSPP01, TGSP06, BN08, SADK⁺09, AEN10].

Große Displays unterstützen die Produktivität, da sie das Wiedererkennungsgeächtnis und das periphere Bewusstsein des Nutzers verbessern [CSR⁺03, CRM⁺06]. Tan et al. [TSPP01] demonstrierten in einer Nutzerstudie, dass die Merkfähigkeit der Nutzer erhöht wurde, indem auf der zusätzlichen Displayfläche verschiedene Bilder in der Peripherie des Nutzers gezeigt wurden, um das Gedächtnis zu unterstützen.

Displaygröße

Bishop und Welch [BW00] erweiterten ihre Desktopumgebung im Büro, indem sie zwei Projektoren verwendeten anstatt des herkömmlichen Einzelmonitors. Sie stellten fest, dass es eine enorme Verbesserung im Arbeitsalltag (*everyday work*) gab, aufgrund verbesserter sozialer und technischer Interaktion, besserer Ergonomie und höherem Informationsgehalt. Die großen Displays unterstützten die Kollaboration zwischen Student und Dozent, da beide gleichzeitig auf Dokumente schauen konnten und jeder produktiv an der Diskussion teilnahm. Außerdem war der Nutzer aufgrund des großen Displays motiviert sich mehr zu bewegen (z.B. aufstehen

vom Bürostuhl, herumlaufen, Kopfbewegungen) anstatt unmittelbar vor dem Monitor zu sitzen und den Kopf nur minimal zu bewegen.

Eine ähnliche Studie mit mehreren LCD-Monitoren (3x3 LCD-Wand) wurde von Ball und North [BN05a] durchgeführt. Hier wurde das Nutzerverhalten bei *everyday tasks* über sechs Monate lang beobachtet. Ihre Beobachtungen zeigen bei der Nutzung großer hochauflösender Displayumgebungen sowohl Vorteile, z.B. verbesserte Nutzerperformanz bei Aufgabenwechseln oder Betrachten von großen Dokumenten, als auch Nachteile, wie Anpassungsphasen (Eingewöhnungszeit von einem auf mehrere Monitore und umgekehrt), Navigationsprobleme durch Verlieren des virtuellen Cursors und die physische Displaygröße benötigte gegebenenfalls mehr körperliche Beanspruchung. Die Bezels des GHDs wurden als natürliche Separation zwischen den Aufgaben und den Anwendungen genutzt, welche den Aufgabenwechsel unterstützten. Jedoch wurden Bilder oder Dokumente bei der Anzeige über mehrere Monitore durch die Bezels verzerrt, was zu Irritation und Frustration beim Nutzer führte.

Der Einfluss von Bezels bei unterschiedlichen Navigationsaufgaben (*visual search*, *tunnel steering*, *target selection*) in GHD-Umgebungen wurde von Bi et al. [BBB10] untersucht. Die Bezels beeinträchtigen weder die Fehlerrate noch die Zeit für die visuelle Suche oder stören bei der Selektion von Zielobjekten. Jedoch haben Bezels negative Auswirkungen auf die gezielte Pfadsteuerung (*straight-tunnel steering*) und auf die Genauigkeit der visuellen Suchen, wenn die Objekte über die Bezels aufgespalten werden.

Tyndiuk et al. [TTLNS05] untersuchten die Nutzerperformanz von perzeptiv-motorischen Aufgaben, wie Manipulation und *Travel* bei einem Desktopdisplay (15-Zoll TFT) und einem Großdisplay (3,3x2,8 m Projektion). Die Ergebnisse zeigen einen eindeutig positiven Effekt bei der Performanz von Manipulations- und Navigationsaufgaben bei der Visualisierung auf Großdisplays. Wobei nicht alle Personen gleich von der Nutzung großer Displays profitieren und der Performanzgewinn stark von der Art der Interaktionsaufgabe abhängt. Zum Beispiel profitieren interaktive Aufgaben, wie Manipulation mehr von der Verwendung großer Displays als passive Aufgaben, wie mentale Rotation.

Displayauflösung

Die Eigenschaften der menschlichen Wahrnehmung (siehe Abschnitt 2.1.1) wird zum Teil bei der Auflösung unterschiedlicher Displaybereiche genutzt. Beispielsweise werden in großen Displayumgebungen die Daten mit einer geringen Auflösung (*low-res*) dargestellt und nur im "Fokus" des menschlichen Sehens erfolgt eine hochauflösende Darstellung (*high-res*) der Daten auf dem Display [BGS01, SAKH06]. Beim *Foveated Rendering System* von Guenter et al. [GFD⁺12] werden unterschiedlich aufgelöste Bildebenen geschachtelt und überlappend gerendert, um eine 3D Szene an die Auflösung des menschlichen Auges angepasst darzustellen.

Isenberg et al. [IDW⁺13] kombinieren zwei unterschiedliche visuelle Repräsentationen in einer Informationsdarstellung (*hybrid-image visualization*), wobei eine Repräsentation aus der Entfernung und die andere Repräsentation nahe am Display wahrgenommen wird. Diese distanzabhängige Informationsdarstellung benötigt kein Personentracking.

Ball und North [BN05b] untersuchten die Nutzerperformanz bei grundlegenden Visualisierungs- und Navigationsaufgaben (z.B. Finden visueller Merkmale in großen 2D Räumen) bei hochauflösenden und niedrigauflösenden Displays. Bei fein-detaillierten Daten übertreffen die hochauflösenden Displays, die physische Navigation unterstützen, signifikant die kleineren Displays, die virtuelle Navigation (*pan-and-zoom*) verwenden. Die Autoren beobachteten im Nutzerverhalten mehr physische Navigation bei hochauflösenden Displays und mehr virtuelle Navigation bei den niedrigauflösenden Displays. Weiterhin konnte mehr Frustration im Umgang mit *pan-and-zoom* beobachtet werden, im Gegensatz zur physischen Navigation.

Displayanordnung

Ein wichtiger Aspekt in großen hochauflösenden Displayumgebungen ist der Sichtwinkel oder Sichtbereich des Nutzers, der durch die Anordnung der Displays beeinflusst wird. Shupp et al. [SBY⁺06] untersuchten die Nutzerperformanz bei unterschiedlichen GHD-Konfigurationen (4x3 und 8x3 LCD-Wand mit planarer und gewölbter Ausrichtung) bei Navigationsaufgaben (z.B. *multi-scale geospatial search*, *route tracing*) und verglichen diese mit einer Einzelmonitor-Konfiguration. Prinzipiell bevorzugten die Teilnehmer die physische Navigation anstatt der virtuellen Navigation. Verglichen mit der Arbeit am Einzelmonitor verbesserte der größere Sichtwinkel bei GHDs die Nutzerperformanz (z.B. Senkung der Nutzerfrustration und Verbesserung der Performanzzeit). Beim Vergleich der planaren LCD-Wand mit der gewölbten LCD-Wand (halbkreisförmig) konnten die Bearbeitungszeiten bei der gewölbten Konfiguration gesenkt werden. Grundsätzlich wurde die beste Nutzerperformanz mit der gewölbten 24-LCD Konfiguration erreicht. Jedoch sind die gewölbten Konfigurationen nur begrenzt für mehrere Benutzer kollaborativ verwendbar. In der Folgestudie von Ball et al. [BNB07] konnte ein Zusammenhang zwischen Displaygröße, Menge der physischen und virtuellen Navigation und der Taskperformanz des Nutzers identifiziert werden. Ein Anstieg der physischen Navigation in GHD-Umgebungen korreliert mit der Abnahme der virtuellen Navigation und verbessert die Nutzerperformanz bei Navigations- und Suchaufgaben.

Eine Zusammenfassung der unterschiedlichen Studien zu großen hochauflösenden Displays und physischer Navigation ist im Artikel von Andrews, Endert, Yost und North [AEYN11] zu finden.

In der Studie zur Informationsvisualisierung von Bezerianos und Isenberg [BI12] wurde die Wahrnehmung von visuellen Variablen (z.B. bei der Auswertung von

Diagrammen) auf großen hochauflösenden Displays untersucht. Die Studie hat gezeigt: Wenn Nutzer sehr nah an der Displaywand stehen, können visuelle Objekte aufgrund des Sichtwinkels des Nutzers größer eingeschätzt werden als diese tatsächlich sind. Wenn sich Nutzer frei bewegen können, erhöht sich die Genauigkeit bei der Unterscheidung der visuellen Variablen, jedoch wird etwas mehr Zeit für den Task benötigt. Basierend auf den Ergebnissen wurden einige Designentscheidungen für die Platzierung von Datenelementen und Charakteristiken für eine effektive physische Navigation bei der Informationsvisualisierung herausgearbeitet.

FAZIT: Große hochauflösende Displayumgebungen ermöglichen es Nutzern Informationen auf eine neue Art und Weise wahrzunehmen und mit diesen anders zu interagieren als bei herkömmlichen Desktopumgebungen [AEYN11], z.B. wird die physische Navigation der virtuellen Navigation vorgezogen. Dies hat Änderungen im Nutzerverhalten und bei den Interaktionsmöglichkeiten zur Folge, was wiederum bei der Informationsdarstellung und den verwendeten Interaktionstechniken in GHD-Umgebungen beachtet werden muss.

2.4. Interaktion in großen hochauflösenden Displayumgebungen

Im Anwendungsgebiet von großen hochauflösenden Displayumgebungen kann zwischen erweiterten Desktopumgebungen und Fullscreen-Anwendungen unterschieden werden. Bei erweiterten Desktopumgebungen wird die Arbeitsfläche eines Desktopsystems um zusätzliche Displays erweitert und Nutzerinteraktionen sind mit dem System und der Anwendung möglich. Bei Fullscreen-Anwendungen verwendet eine Anwendung die gesamte Displayarbeitsfläche und die Nutzerinteraktion erfolgt nur mit der Anwendung. Üblicherweise werden erweiterte Desktopumgebungen für Büro- und Analyseaufgaben (vgl. [CHBL05, EBZ⁺12]) und Fullscreen-Anwendungen für Informationsvisualisierungen (vgl. [BN07, AEYN11, NWP⁺11]) genutzt.

Die Portierung der Anwendungsgebiete von Kleindisplays (Single- oder Doppel-Display) auf Großdisplays resultiert in unterschiedlichen Problemen bei der Interaktion (siehe Abschnitt 2.3.1). Bei erweiterten Desktopumgebungen müssen die klassischen 2D Desktopmetaphern (z.B. WIMP – *window, icon, menu, pointing*) für große hochauflösende Displays angepasst werden. Bei Fullscreen-Anwendungen werden hauptsächlich Aufgaben wie Selektion von Datensätzen und Navigation innerhalb der Informationsvisualisierung durchgeführt, für welche andere Eingabemodalitäten als Maus und Tastatur umgesetzt werden müssen. Ein Überblick über angepasste und erweiterte Interaktionstechniken für große Displayumgebungen findet sich in [RCB⁺05] und [BBM⁺06].

2.4.1. Nutzerinteraktion

Die Interaktion eines Nutzers mit virtuellen Umgebungen umfasst im wesentlichen drei Komponenten: Eingabemodalität, Interaktionsaufgabe und Mapping. Die **Eingabemodalität** der Nutzerinteraktion (kurz Nutzereingabe) beschreibt welches Eingabegerät der Nutzer verwendet, um mit dem System oder der Anwendung zu interagieren und welche zusätzlichen Sensordaten (z.B. Positionstracking) für die Nutzereingabe verwendet werden. Die **Interaktionsaufgabe** beschreibt welche Aufgabe der Nutzer durchführt, indem er mit dem System oder der Anwendung interagiert. Das **Mapping** beschreibt wie die Nutzereingabe auf die gewünschte Interaktionsaufgabe abgebildet wird.

2.4.1.1. Eingabemodalität

Bei GHD-Umgebungen ist es notwendig, dass der Nutzer sich vor dem Display in unterschiedlichen Entfernungen (frei) bewegen kann um alle visualisierten Daten zu erfassen. Um mit der Datenvisualisierung oder dem System zu interagieren, werden verschiedenen Eingabemodalitäten in GHD-Umgebungen unterstützt, wie Eingabegeräte (Laserpointer, Datenhandschuh, Nintendo Wii Remote etc.), direkte

menschliche Eingabe (Gesten, Sprache, *brain input*) oder eine Kombination daraus (z.B. Gestensteuerung mit Wii Remote). Werden Eingabegeräte verwendet, müssen diese im realen dreidimensionalen Raum einsetzbar sein (ohne die Notwendigkeit einer 2D Ebene zum Ablegen des Gerätes), wobei die Geräte entweder Daten für den 2D Raum oder den 3D Raum erzeugen können. Wenn die Displaywand klein genug ist, so dass der Nutzer von einer festen Position alle repräsentierten Daten sehen kann, dann können auch stationäre Eingabegeräte verwendet werden (z.B. Maus und Tastatur, Joystick).

Eingabegeräte können nach unterschiedlichen Eigenschaften charakterisiert werden (vgl. [FWC84, Jac97, BKLP04, Kön10, HW12]). Für unseren konzipierten Parameterraum (siehe Abschnitt 3.1) sind folgenden Eigenschaften der Eingabegeräte relevant: Freiheitsgrade, Datenreports, Eingabeart, Benutzungsaufwand und physikalische Eigenschaften. Bei der Eingabemodalität, welche das Eingabegerät mit einem Tracking kombiniert ist außerdem der Trackingbereich und die Genauigkeit bedeutsam.

Freiheitsgrade: Ein Freiheitsgrad (*degree-of-freedom*, kurz DOF) ist eine bestimmte unabhängige Art, wie sich ein Körper im Raum bewegt [BKLP04]. Ein Tracker als Eingabegerät erzeugt beispielsweise drei Positionswerte und drei Orientierungswerte, womit insgesamt sechs Freiheitsgrade (6-DOF) bestimmt werden.

Datenreport: Die Datenreports sind der Eingabetyp und die Frequenz der Daten, die Eingabegeräte erzeugen [BKLP04]. Dabei kann zwischen diskreten und kontinuierlichen Datenreports oder einer Kombination aus beiden unterschieden werden. Die diskreten Eingabetypen erzeugen einzelne Datenwerte basierend auf einer Nutzerinteraktion, z.B. ein Button-Event bei Tastendruck auf der Tastatur. Kontinuierliche Eingabetypen erzeugen multiple Datenwerte als Reaktion auf Nutzeraktionen, z.B. Pixelkoordinaten bei Nutzereingabe via Touchpad. Die kontinuierlichen Datenreports können auch unabhängig von der gegenwärtigen Nutzeraktion erfolgen (z.B. bei Trackingsystemen). Um eine größere Bandbreite an gleichzeitiger Nutzerinteraktion zu ermöglichen werden in Eingabegeräten oft kontinuierliche und diskrete Eingabetypen kombiniert (z.B. Gyro-Maus, Spaceball).

Eingabeart: Bei der Eingabeart wird unterschieden zwischen absoluten und relativen Eingabegeräten, zwischen isotonischen und isometrischen Eingabegeräten, sowie zwischen direkten und indirekten Eingabegeräten. Absolute Eingabegeräte registrieren die Position einer Eingabe und geben diese an das System weiter, während relative Eingabegeräte nur die Änderung einer Position registrieren [Jac97, HW12]. Bei isotonischen Geräten (z.B. Maus) werden minimale Bewegungen registriert und verarbeitet, während bei isometrischen Geräten (z.B. Trackball) eine bestimmte Krafteinwirkung erfolgen

muss, um eine Änderung an das System weiterzugeben [Zha98]. Bei den direkten Eingabegeräten besteht eine unmittelbare Verbindung zwischen Nutzereingabe und Displayoberfläche. Bei indirekten Eingabegeräten erfolgt die Eingabe nicht im gleichen physischen Raum wie die Ausgabe [HW12].

Zum Beispiel ermöglichen *Touch-Screens* direkte Eingaben mit dem Finger oder Stift auf der Displayoberfläche, um das darunterliegende virtuelle Objekt auszuwählen. Bei der Verwendung einer Maus als indirektes Eingabegerät muss der Nutzer das Eingabegerät auf einer Oberfläche (z.B. Schreibtisch) bewegen um den Cursor auf einer anderen Oberfläche (Bildschirm) zu verschieben und ein virtuelles Objekt unter dem Cursor auswählen zu können [HW12]. Jedoch sind direkte Eingabegeräte nicht unbedingt einfacher zu bedienen als indirekte Eingabegeräte, da bei direkten Eingabegeräten durch das Eingabemedium (z.B. Finger, Hand, Stift) der visuelle Bereich auf dem Display, mit dem der Nutzer gegenwärtig interagiert, verdeckt wird. Indirekte Eingabegeräte eliminieren Verdeckungen des Bildschirms durch den Nutzer, aber sie benötigen mehr explizites Feedback für die Nutzereingabe, wie eine Repräsentation des Eingabegerätes (Cursor), des gewünschten Ziels auf dem Bildschirm (z.B. *mouse hover*) und des aktuellen Zustandes des Eingabegerätes (ob ein Button gedrückt ist oder nicht) [HW12]. Der Vorteil indirekter Eingabegeräte ist, dass sie besser auf großen Interaktionsflächen (z.B. GHD-Umgebungen) skalieren, weil sie weniger Körperbewegung benötigen als die direkte Eingabe und auch aus der Entfernung mit dem Display interagieren können.

Bei der Eingabeart müssen daher Einflüsse wie Ermüdung und Hand-Augen-Koordination berücksichtigt werden. Direkte Zeigetechniken beispielsweise minimieren zwar den mentalen Aufwand für die Hand-Augen-Koordination, aber der motorische Aufwand ist hoch, weil der Arm zum Display angehoben werden muss. Mit indirekten Zeigetechniken kann der motorische Benutzungsaufwand reduziert werden, aber sie benötigen einen erhöhten mentalen Aufwand zum Lernen der Hand-Augen-Koordination [FWC84]. Aufgrund dessen werden hybriden Techniken eingesetzt, welche direkte und indirekte Repräsentation mischen, z.B. indirekte Eingabe aus der Entfernung und direkte Eingabe nahe am Display (vgl. Abschnitt 2.4.2).

Benutzungsaufwand: Der Benutzungsaufwand eines Eingabegerätes beschreibt wie viel physische Interaktion nötig ist, um das Eingabegerät zu verwenden. Es kann zwischen aktiven und passiven Eingabegeräten unterschieden werden [BKLP04]. Die rein aktiven Eingabegeräte erfordern eine physische Aktion des Nutzers, damit Datenreports erzeugt werden. Diese Eingabegeräte können diskrete (z.B. Buttons) oder manuell kontinuierliche Eingabetypen (z.B. Slider) haben. Die rein passiven Eingabegeräte erzeugen kontinuierliche Datenreports und benötigen keine physische Aktion des Nutzers (z.B. Tracker). Diese Eingabegeräte können zwar vom Nutzer manipuliert werden, dass aber nicht zwingend für die Erzeugung von Daten notwendig ist. Solche

Eingabegeräte werden oft für Positionsbestimmungen im Raum verwendet, ohne dass die Anwendung explizit danach “fragen“ muss.

Physikalische Eigenschaften: Einige physikalische Eigenschaften der Eingabegeräte wie Form, Gewicht und Größe beeinflussen direkt den Komfort bei der Bedienbarkeit und damit den Benutzungsaufwand [FWC84, Zha98, HW12].

2.4.1.2. Tracking

In GHD-Umgebungen wird für die Interaktion oft die Position der Nutzer oder die Position der physischen Objekte im realen 3D Raum benötigt. Dazu werden Interaktionsgeräte verwendet, die von einem Trackingsystem im Raum erkannt und verfolgt werden (*Motion Tracking*). Bei den sogenannten *tracking devices* handelt es sich meist um rein passive Eingabegeräte, die kontinuierlich Informationen erzeugen. Für die Positions- und Bewegungserfassung von Objekten oder Nutzern können verschiedene Trackingtechnologien verwendet werden, z.B. magnetische, inertial, optische, mechanische, akustische oder Funkfrequenzsensoren. Eine Übersicht über die verschiedenen Trackingtechnologien findet sich in [ABW01, WF02]. Kritische Eigenschaften von *Motion Tracking* sind: Trackingbereich, Latenz, Jitter und Genauigkeit der Positions- und Orientierungsbestimmung [BKLP04].

An dieser Stelle werden nur Trackingtechnologien vorgestellt, die in GHD-Umgebungen genutzt werden und Mobilität bei der Nutzereingabe ermöglichen.

Inertial Tracking verwendet Trägheitsmessgeräte, wie Beschleunigungssensoren und Drehraten-Gyroskope. Die Messgeräte liefern sekundäre Messungen, d.h. das Gyroskop liefert die Winkelgeschwindigkeit und das lineare Akzelerometer liefert die lineare Beschleunigung, die anschließend integriert werden, um Positions- und Orientierungsdaten zu erhalten [BKLP04]. Die mobilen Sensoren sind klein und leicht, sie unterstützen den Tragekomfort und ermöglichen dem Nutzer, sich frei zu bewegen. Der Nachteil des Systems ist die Fehlerakkumulation der Sensoren (Drift).

Magnetisches Tracking verwendet einen Emitter (Sender eines niedrig-frequenten Magnetfeldes) und einen Empfänger (kleiner Sensor am Nutzer oder Objekt), der seine Position und Orientierung relativ zur magnetischen Quelle bestimmt [BKLP04]. Der verfügbare Trackingbereich kann nach *short-range* (bis ca. 1 m), *mid-range* (ca. 1-2 m) und *wide-range* (ab ca. 2 m) klassifiziert werden. Die kleineren Trackingbereiche verfügen über eine hohe Genauigkeit⁴, eignen sich aber weniger für große Displayumgebungen. Mit zunehmender Größe des Trackingbereiches nimmt die Genauigkeit der Bestimmung der Positions- und Orientierungsdaten ab⁵. Befinden sich jedoch ferromagnetische und an-

⁴ Der quadratische Mittelwert ist $< 1,4$ mm bei Position und $< 0,5^\circ$ bei Orientierung (z.B. Ascension trakSTAR <http://www.ascension-tech.com>).

⁵ Der quadratische Mittelwert bei *mid-range/wide-range* Tracking ist vom System abhängig, etwa $1,4 - 3,8$ mm bei Positionsdaten und $0,5^\circ$ bei Orientierungsdaten (z.B. Ascension trakSTAR).

dere leitende Objekte im Raum, können diese das magnetische Feld stören und die Genauigkeit reduzieren. Die meisten magnetischen Trackingsysteme verwenden kabelgebundene Sensoren womit die Mobilität des Nutzers auf deren Radius beschränkt ist.

Optisches Tracking verwendet Computer Vision Techniken und optische Sensoren, wie Kameras und Infrarot-Emitter. Bei den optischen Trackingsystemen kann zwischen *outside-in* und *inside-out* unterschieden werden [BKLP04]. Die *outside-in* Systeme befestigen die optischen Sensoren an festen Positionen in der Umgebung und das zu trackende Objekt ist mit aktiven oder passiven Markierungen gekennzeichnet (z.B. retroreflektive Marker oder farbige Handschuhe). Bei den *inside-out* Systemen sind die optischen Sensoren auf dem zu trackenden Objekt oder Nutzer befestigt und die Markierungen sind in der Umgebung platziert (z.B. *QR Code Scanning*). Durch die Verwendung von Markern und mobilen Kleinkameras (wie im Smartphone) sind die Mobilität und der Tragekomfort für den Nutzer gegeben. Die Nachteile der optischen Trackingsysteme sind der Aufbau (z.B. Kamerakalibrierung, geeignete Lichtverhältnisse und optimale Platzierung der Marker) und verdeckte Objekte. Die Kamera kann beispielsweise keine Informationen von Körperteilen der Nutzer aufnehmen, die von anderen verdeckt werden. Um das Verdeckungsproblem zu minimieren können mehrere Kameras mit unterschiedlichen Perspektiven verwendet werden, was aber die Komplexität des Trackingalgorithmus erhöht.

Funkbasiertes, mechanisches und akustisches Tracking werden in großen Displayumgebungen aus den folgenden Gründen selten eingesetzt:

Funkbasierte Trackingtechnologien unterstützen meist eine maximale Trackingauflösung im Zentimeterbereich (z.B. RFID, GPS). Mit einem hohen technischen Aufwand können auch Trackingbereiche mit einer Auflösung von wenigen Millimetern erreicht werden [WF02]. Manche funkbasierte Ansätze unterstützen zwar ein geräteloses Tracking von Personen, jedoch kann nur die Position von Personen bestimmt werden und nicht deren Orientierung (vgl. [WST13]).

Mechanische Trackingtechnologien verwenden ein mechanisches Gestänge, um Bewegungen zu erfassen, dass eine sehr hohe Genauigkeit beim *Motion Tracking* ermöglicht [BKLP04]. Diese Trackingtechnologie eignet sich für präzise und kleine Trackingbereiche, wie in der Medizin. Für Personentracking kann ein Exoskelett eingesetzt werden, welches aber den Tragekomfort erheblich einschränkt.

Akustisches Tracking ermöglicht einen großen Trackingbereich und ist toleranter bei Verdeckungen, da Schallwellen ihren Weg durch und um Objekte fin-

det [WF02]. Die Genauigkeit leidet jedoch, wenn sich akustisch reflektierende Oberflächen im Raum befinden und externe Geräusche (z.B. Klingeltöne) können Interferenzen im Trackingsignal verursachen [BKLP04].

Hybrid Tracking kombiniert zwei oder mehr Trackingtechnologien, um die Nachteile der einzelnen Systeme auszugleichen. Beim *ultrasonic-inertial* Tracking wird beispielsweise die inertial Komponente für die Bestimmung der Orientierung und die Ultraschall-Komponente für die Positionsberechnung verwendet. Außerdem werden die Informationen einer Komponente zur Verbesserung der Genauigkeit der anderen Komponente genutzt. Jedoch erhöht die Anzahl der Komponenten die Komplexität des hybriden Trackingsystems [BKLP04].

Eine Bewertung der Sensortechnologien für das *Motion Tracking* von Personen und Objekten in GHD-Umgebungen ist in Tabelle 2.4.2 zu finden.

Technologie	Vorteile (+) und Nachteile (-)	Eignung
Funkbasiertes Tracking	+ gerätefreies Tracking von Personen - Trackingauflösung im cm-Bereich - 3-DOF (nur Positionsbestimmung)	nein
Mechanisches Tracking	+ sehr präzises Tracking (mm-Bereich) + 6-DOF - mechanisches Gestänge für Sensorik	nein
Akustisches Tracking	+ großer Trackingbereich - <i>line-of-sight</i> -Bedingung - Ungenauigkeit durch akustische Reflexionen und Geräusche	nein
Magnetisches Tracking	+ großer Trackingbereich + präzises Tracking (mm-Bereich) + kleine Sensoren; 6-DOF - Kabellänge des Sensors begrenzt Trackingbereich - Ungenauigkeit durch ferromagnetische Objekte	ja
Inertial Tracking	+ großer Trackingbereich + präzises Tracking (mm-Bereich) + kleine Sensoren; 6-DOF - Fehlerakkumulation (Drift)	ja
Optisches Tracking	+ großer Trackingbereich + präzises Tracking (mm-Bereich) + kleine Sensoren (Marker, Kamera); 6-DOF - Verdeckungsproblem	ja
Hybrides Tracking	+ großer Trackingbereich + präzises Tracking (mm-Bereich) + kleine Sensoren; 6-DOF - Komplexität	ja

Tabelle 2.4.2.: Eignung der Sensortechnologien für das Tracking von Personen und Eingabegeräten für präzise Nutzerinteraktion in GHD-Umgebungen.

Neben der Verfolgung von expliziten Nutzerbewegungen können auch implizite motorische Bewegungen getrackt werden, wie z.B. die Augenbewegungen mit Hilfe von Eye-Tracking Systemen.

Eye-Tracking Systeme können als Evaluationstool oder als Eingabemodalität für Nutzerinteraktion mit der Anwendung verwendet werden (vgl. [Duc02]). Beim Eye-Tracking werden die Pupillen des Nutzers mittels Hornhautreflexion, die durch eine Kamera erkannt wird, getrackt [BKLP04]. Die Eye-Tracking Geräte können entweder vom Nutzer getragen werden oder im Display, auf das der Nutzer schaut, integriert werden. Nachteile der Systeme sind: Korrektes Eye-Tracking erfordert Kalibrierungsaufwand für jeden Nutzer und der Nutzer muss unter Umständen eine unförmige Kopfkonstruktion tragen, die den Tragekomfort beeinträchtigt. Die Verwendung von Kameras im Display ist zwar weniger aufdringlich für den Nutzer als eine Kopfkonstruktion, aber sie beschränken den verfügbaren Bewegungsbereich des Nutzers innerhalb dessen seine Augen getrackt werden können.

Daher wird oftmals ein Approximationsansatz mit Hilfe von Head-Tracking als Eingabemodalität verwendet.

Head-Tracking verfolgt die Kopfposition und -orientierung des Nutzers, um dessen Blickrichtung zu bestimmen, z.B. indem der Nutzer eine Mütze mit Markern trägt. Somit kann der Nutzer sich frei vor dem Display bewegen und es können Interaktionstechniken wie *gaze-directed steering* (z.B. [BDN⁺06]) oder *gaze-directed manipulation* (z.B. [AOSS05, FTS⁺10]) unterstützt werden. Jedoch ist Head-Tracking ungenauer als Eye-Tracking, insbesondere wenn der Nutzer den Kopf stillhält und die Augen hin-und-her bewegt. In diesem Fall kann keine blickrichtungsabhängige Änderung vom System erkannt werden.

Für den Einsatz von Head-Tracking in großen Displayumgebungen spielen weitere Faktoren eine wichtige Rolle: Ergonomie, Reichweite, kabellose Nutzung und Performanz [WF02]. Beispielsweise muss der Tracker (auf dem Kopf oder in der Hand) für den Nutzer über einen längeren Zeitraum tragbar und komfortabel sein. Große Displayumgebungen unterstützen oft Mehrbenutzer-Szenarien, daher sollte auf Kabel verzichtet werden, über welche die Nutzer stolpern oder sich gegenseitig in der Bewegungsfreiheit einschränken könnten. Unter Performanz fallen Eigenschaften wie geringe Latenz und geeignete Genauigkeit für Positionstracking, so dass schnell auf Positionsänderungen durch die virtuelle Umgebung reagiert werden kann und natürliches Kopf- oder Handzittern nicht als Positionsänderung verarbeitet wird.

2.4.1.3. Interaktionsaufgaben

Die grundlegenden 3D Interaktionsaufgaben in solchen Displayumgebungen sind nach Bowman et al. [BKLP04] Manipulation, Navigation, Systemkontrolle, Symbolingabe und Modellierung und sie werden wie folgt definiert.

Manipulation: Die Manipulationsaufgaben beschränken sich auf Objektveränderungen, welche die Form der Objekte erhalten. Dazu gehören Selektion (Iden-

tifizieren eines bestimmten Objektes aus einer Reihe verfügbarer Objekte), Positionierung (Ändern der 3D Position eines Objektes) und Rotation (Ändern der Orientierung eines Objektes). Anders als Bowman et al. zählen wir die Skalierung (Ändern der Objektgröße) ebenfalls zu den grundlegenden Manipulationsaufgaben, da bei dieser Objektdeformation⁶ die Form der Objekte erkennbar bleibt.

Navigation: Die Navigation kann allgemein in drei Kategorien klassifiziert werden: Erkunden, Suchen und Manövrieren. Ebenso kann die Navigation in eine motorische Komponente (*Travel*) und eine kognitive Komponente (*Wayfinding*) unterteilt werden. Beim *Travel* wird mittels Bewegung ein Ziel angesteuert indem der Nutzer die Position und Orientierung seines Standpunktes in der virtuellen Welt kontrolliert. *Wayfinding* umfasst Aufgaben für räumliches Wissen und Planung in der virtuellen Umgebung, wie beispielsweise die aktuelle Position innerhalb der Umgebung zu bestimmen, einen Pfad von der aktuellen Position zu einer Zielposition zu definieren und das Erstellen einer mentalen Karte der Umgebung.

Systemkontrolle: Dies sind Interaktionsaufgaben, bei denen entweder der Status des Systems oder der Modus der Interaktion geändert wird. Meistens erfolgt dies über Menüs (Widgets) oder Kommandos.

Symboleingabe: Dies umfasst die Eingabe, Bearbeitung und Hervorhebung von Buchstaben, Zahlen und anderen Zeichen oder Symbolen.

Modellierung: Bei der Interaktionsaufgabe Modellierung handelt es sich um eine komplexere Interaktionsaufgabe, bestehend aus dem Erzeugen geometrischer 3D Objekte und der Modifikation ihrer Eigenschaften. Hierzu gehören beispielsweise Objektdeformationen, welche die Objektform verändern.

Bei der Interaktion in virtuellen Umgebungen werden meist mehrere oder komplexe **Interaktionstasks** benötigt, die durch Interaktionstechniken umgesetzt werden. Die Selektion (das Auswählen von virtuellen Objekten) ist eine fundamentale Interaktionsaufgabe, die meist als initiale Nutzereingabe für weiterführende Tasks genutzt wird. Wie präzise der Nutzer interagieren kann, ist für die Interaktionstechnik wichtig: Soll beispielsweise eine pixelgenaue Selektion möglich sein oder werden alle Objekte innerhalb des Radius des Interaktionscursors selektiert.

Die Interaktionsaufgaben werden durch ihre Task-Parameter beschrieben. Am Beispiel der Manipulation und der Navigationsbewegung (*Travel*) werden die einzelnen Task-Parameter der Interaktionsaufgaben näher erläutert (vgl. Tabelle 2.4.3 und [Min95, BKLP04]).

⁶ Bei der Objektdeformation kann zwischen Größenänderungen (Manipulationsaufgabe) und Formänderungen (Modellierungsaufgabe) unterschieden werden.

Interaktionsaufgabe	Task-Parameter
Manipulation: Die visuellen Objekte werden verändert.	
Selektion	Entfernung und Richtung zum Zielobjekt, Größe des Zielobjektes, Objektdichte um das Zielobjekt, Anzahl der zu selektierenden Zielobjekte, Verdeckung des Zielobjektes
Positionierung	Entfernung und Richtung zur initialen Position, Entfernung und Richtung zur Zielposition, Verschiebungsdistanz, benötigte Präzision der Positionierung
Rotation	Entfernung zum Zielobjekt, initiale Orientierung, endgültige Orientierung, Rotationswinkel (Anzahl der Rotationen), Rotationszentrum, benötigte Präzision der Rotation
Skalierung	Entfernung zum Zielobjekt, initiale Skalierung, endgültige Skalierung, Skalierungsfaktor, Skalierungszentrum, benötigte Präzision der Skalierung
Navigation: Die Ansicht auf die visuellen Objekte wird verändert.	
Bewegung	Bewegungsrichtung (gezielte, gesteuerte, kontrollierte), Geschwindigkeitskontrolle (konstante Geschwindigkeit oder Beschleunigung), Bewegungsablauf (Eingabebedingung für Initiieren, Fortführen und Beenden der Navigation)

Tabelle 2.4.3.: Zerlegung der Manipulations- und Navigationsaufgaben in ihre Task-Parameter (nach [Min95, BKLP04]).

Selektion: Für die Selektion muss die Entfernung und Richtung zum Zielobjekt bestimmt werden, z.B. mittels *Ray-Casting*. Für die Entscheidung welches Objekt beim Auftreffen des Strahls als selektiert gilt, kann die Größe des Zielobjektes, die Objektdichte um das Zielobjekt oder die Anzahl der selektierbaren Zielobjekte verwendet werden. Erfolgt die Selektionsentscheidung z.B. über den Cursorradius, dann können entweder alle Objekte, die ganz oder teilweise vom Cursorradius überdeckt werden, als selektiert gelten oder nur das Objekt, welches dem Cursormittelpunkt am nächsten ist. Ein weiteres Entscheidungskriterium ist die Verdeckung von Zielobjekten, d.h. werden alle Zielobjekte unter dem Cursor selektiert, oder nur das oberste Zielobjekt bzw. kann der Nutzer die Tiefe des Selektionsstrahls bestimmen.

Positionierung: Für die Positionierung von Objekten muss die Interaktionstechnik die Entfernung und Richtung zur initialen und zur endgültigen Objektposition bestimmen können, z.B. indem ein Objekt selektiert wird (initiale Position), der Nutzer bewegt den visuellen Cursor und deselektiert das Objekt (endgültige Position). Die resultierende Verschiebungsdistanz kann entweder als Animation von der initialen zur endgültigen Position erfolgen (Berechnung des kürzesten Pfades) oder das selektierte Objekt bewegt sich mit dem Cursor mit bis es wieder deselektiert wird (Berechnung der Positionsänderungen). Ein weiterer Parameter ist die benötigte Präzision der Positionierung, z.B. können Objekte genau platziert werden oder Einrastungsmechanismen (*snapping*) werden verwendet.

Rotation: Äquivalent sind die Parameter für die Rotationstechniken: die Entfernung zum Zielobjekt, die initiale und endgültige Orientierung, die Anzahl der Rotationen und die Präzision der Rotation. Bei der Objektrotation kommt der Parameter Rotationszentrum hinzu, d.h. erfolgt die Rotation um den Objektmittelpunkt, den Objektschwerpunkt, den Koordinatenursprung der virtuellen Umgebung oder um eine beliebige Rotationsachse (z.B. der Selektionspunkt im virtuellen Objekt).

Skalierung: Für die Skalierungstechniken werden die äquivalenten Parameter für die Bestimmung der Entfernung zum Zielobjekt, die initiale und endgültige Skalierung und die benötigte Präzision der Skalierung verwendet. Die Objektskalierung kann entweder gleichmäßig in alle Objektdimensionen (uniformer Skalierungsfaktor) oder unterschiedlich für alle Objektdimensionen (nicht-uniforme Skalierungsfaktoren) ausgeführt werden. Das Zentrum für die Objektskalierung kann objektzentriert sein, d.h. es wird von der Objektmitte aus skaliert, oder nutzerdefiniert sein, d.h. das Skalierungszentrum wird durch die Nutzerinteraktion festgelegt (z.B. Skalierung an der Objektkante).

Bewegung: Für die Navigationsbewegungen in virtuellen Umgebungen werden die Task-Parameter Bewegungsrichtung, Geschwindigkeitskontrolle und Bewegungsablauf benötigt. Die Bestimmung der Bewegungsrichtung kann gezielt, gesteuert oder kontrolliert durchgeführt werden. Bei der gezielten Bewegungsrichtung wird die Zielposition über die Nutzereingabe bestimmt und die virtuelle Umgebung berechnet den kürzesten Pfad zum Ziel. Bei der gesteuerten und kontrollierten Bewegungsrichtung erfolgen kontinuierliche Positions- oder Orientierungsänderungen über die Nutzereingabe. Bei der gesteuerten Bewegungsrichtung erfolgt die Bewegung in Richtung der Orientierung der Nutzereingabe (z.B. *gaze-directed*, *torso-directed*), während bei der kontrollierten Bewegungsrichtung die Richtung durch das Eingabegerät bestimmt wird (z.B. Lenkrad, Schieberegler). Für die Geschwindigkeitskontrolle werden meist konstante Geschwindigkeiten oder Beschleunigungsfunktionen verwendet.

Bei einigen Navigationstechniken wird der Subtask Nutzerskalierung benötigt. Bei der Nutzerskalierung wird nur die wahrnehmbare Größe des Objektes verändert, nicht die tatsächliche Objektgröße in der virtuellen Umgebung (z.B. *zooming*). Der Skalierungsfaktor kann durch eine diskrete oder stufenlose Nutzereingabe angepasst werden (z.B. die Zoomstufen in Landkarten oder das Vergrößern von Vektorgrafiken).

Die Task-Parameter beeinflussen die Nutzerperformanz während der Ausführung der Interaktionsaufgabe [BKLP04] und je nach Zielsetzung lassen sich aus den Task-Parametern die Parameter für die verwendete Interaktionstechnik ableiten (siehe Abschnitt 2.4.2).

2.4.1.4. Mapping

Damit die Eingabemodalität der Nutzerinteraktion auf die gewünschte Interaktionsaufgabe abgebildet werden kann, wird ein Mapping von der realen Umgebung auf die virtuelle Umgebung (Anwendung oder System) benötigt. Eine Variante hierfür ist die Anwendung eines sogenannten *control-display-gain* (CD-Gain), d.h. ein "einheitsloser" Koeffizient (Faktor) bildet die Bewegungen des Eingabegerätes (*control space/motor space*) auf die Bewegungen des visuellen Cursors (*display space*) ab [CVBC08]. In Studien wurden unterschiedliche CD-Gains für verschiedene Interaktionsaufgaben untersucht, z.B. Selektion [MR94, Gra96, CVBC08], Rotation [PWF00, LK07], Manipulation [WM99] und Navigation [LFKZ01, GBLB⁺04].

Wie das Mapping der Nutzereingabe auf die virtuelle Interaktion umgesetzt wird, hängt von verschiedenen Kriterien ab, wie Isomorphie, Kinematik, Eingabeänderung und Projektion.

Isomorphie: Erfordert die Anwendung eine Abbildung der Realität, dann sollte eine strikte geometrische eins-zu-eins Korrespondenz zwischen Bewegungen in der physischen und der virtuellen Welt umgesetzt werden – wie beim isomorphen Ansatz. Der isomorphe Ansatz ist zwar natürlich für den Nutzer, aber mitunter sind nur eingeschränkte Bewegungen möglich, einerseits durch den begrenzten Trackingbereich des Eingabegerätes und andererseits durch anatomische Beschränkungen des Nutzers [PWF00]. Wenn Realismus nicht erforderlich ist, dann kann ein nicht-isomorpher Ansatz umgesetzt werden. Dieser ermöglicht Objektmanipulation anders als in der realen Welt und stattet den Nutzer mit sogenannten *magic virtual tools* aus [BKLP04]. Da beim nicht-isomorphen Ansatz die physischen Beschränkungen der Welt und des Menschen umgangen werden, kann dies effektiver für die Nutzerperformanz sein. Jedoch entsprechen die Bewegungen nicht der Realität und müssen daher vom Nutzer gelernt werden.

Kinematik: Das Kriterium Kinematik gibt an, welcher Bewegungsparameter der Eingabemodalität verwendet wird. Bei *position control* (*zero-order*) wird die Eingabe direkt auf die Position im virtuellen Raum gemappt, bei *rate control* (*first-order*) wird die Eingabe auf die Bewegungsgeschwindigkeit im virtuellen Raum gemappt und bei *acceleration control* (*second-order*) wird die Eingabe auf die Bewegungsbeschleunigung im virtuellen Raum gemappt [KCHI07].

Eingabeänderung: Die Eingabeänderung unterscheidet zwischen absoluten Werten und relativen Werten, ähnlich der Eingabeart von Eingabegeräten. Beim absoluten Mapping wird die absolute Veränderung des Eingabegerätes (z.B. Position/Orientierung) verwendet und entsprechend des CD-Gain verstärkt oder verringert, um die virtuelle Position/Orientierung des Cursors zu berechnen. Beim relativen Mapping wird die relative Veränderung des Eingabe-

begerätes verwendet und abhängig vom CD-Gain die Position/Orientierung des virtuellen Cursors berechnet.

Projektion: Eine andere Sicht zur Differenzierung der Mappingtechniken ist die Art der Projektion von Position/Orientierung der Nutzereingabe auf die virtuelle Umgebung (z.B. perspektivische oder orthogonale Projektion, Spiegelparadoxon).

In einer Studie haben Keijser et al. [KCHI07] unterschiedliche Arten der Projektion beim Mapping getestet. Sie verwendeten das Standard Mapping (*perspective mapping*), eine perspektivische Projektion der Eingabe vom physischen Raum in den virtuellen Raum und eine Normalprojektion (*orthographic mapping*). Weiterhin wurde für beide Mappingvarianten jeweils eine *mirrored* Technik angewendet, bei dem die Tiefenachse gespiegelt wird, d.h. wird das Eingabegerät zum Display hin bewegt, dann bewegt sich der virtuelle Cursor auf den Nutzer zu. Im Experiment wurden die vier Mappings (*perspective*, *orthographic*, *perspective mirrored*, *orthographic mirrored*) für Selektions- und Manipulationsaufgaben in einer 3D Szene verglichen (36-Zoll Display mit 1024x768 px). Die Ergebnisse zeigten, dass die Probanden persönliche Präferenzen hatten und das Standard Mapping nicht unbedingt bevorzugt wurde, wobei es einen signifikanten Unterschied zwischen den *mirrored* und *non-mirrored* Mappings gab. Die Autoren konnten zeigen, dass mit wenig visueller Unterstützung die Performanz beim *orthographic* Mapping nicht signifikant schlechter war und die Performanz bei den *mirrored* Mappings sich verbesserte, obwohl es keine visuelle Unterstützung für diese Mappings gab.

Die Qualität von 3D Interaktionstechniken messen sich daran, wie effektiv Objektmanipulation in virtuellen Umgebungen durchgeführt werden kann [PWF00]. Da die menschliche Physiologie den Nutzer in einigen Bewegungen einschränkt (vgl. Abschnitt 2.1.2), kann unter Umständen *Clutching* notwendig sein. Dies tritt auf, wenn der Nutzer ein virtuelles Objekt loslassen muss, um seine Hände oder Arme neu auszurichten, und anschließend das virtuelle Objekt wieder anfasst, um die Objektmanipulation fortzuführen. Solche Aktionen können beschwerlich und ermüdend für den Nutzer sein, wenn er bei der Objektmanipulation oft "nach greifen" muss und damit der Benutzungsaufwand für ihn erheblich erhöht wird.

Zum Beispiel steht dem Nutzer bei der Objektmanipulation durch Rotationsbewegungen mit dem Handgelenk nur ein kleiner Interaktionsraum zur Verfügung, den er sehr genau steuern kann. In diesem Fall eignen sich eher Verstärkungsfunktionen für den CD-Gain um die Nutzereingabe zu erleichtern.

Ausführliche Beschreibungen zu den Berechnungen des CD-Gain finden sich in [PWF00, KCHI07, CVBC08]. Im Allgemeinen wird entweder ein konstanter CD-Gain verwendet oder ein dynamischer CD-Gain, der durch die Nutzereingabe angepasst wird.

Konstanter CD-Gain

Wird ein konstanter vorberechneter Faktor für das Mapping verwendet, dann ist der CD-Gain für das verwendete Eingabegerät vordefiniert und in der Regel für alle Nutzer gleich.

Poupyrev et al. [PWF00] untersuchten die Nutzerperformanz einer nicht-isomorphen Rotationstechnik (Verstärkungsfaktor 1.8) mit einer isomorphen Rotationstechnik (1:1-Mapping) in einer herkömmlichen Desktopumgebung (17-Zoll Display mit 1280x1024 px). Mit der nicht-isomorphen Technik konnten die Probanden die gestellte Aufgabe 13% schneller durchführen als mit der isomorphen Rotationstechnik, ohne signifikanten Genauigkeitsverlust.

LaViola et al. [LK07] haben das Experiment erweitert und die Nutzerperformanz bei mehreren nicht-isomorphen Mappings (Verstärkungsfaktor von 2, 3 und 4) und einem isomorphen Mapping in einer großen Displayumgebung (Cave-basierte Umgebung mit je 1024x768 px pro Wand) verglichen. Das nicht-isomorphe Mapping mit Verstärkungsfaktor 3 wurde von den Probanden bevorzugt und erzielte die optimalen Ergebnisse bezüglich Genauigkeit und Bearbeitungszeit (15% schneller als isomorphes Mapping ohne signifikanten Genauigkeitsverlust). Die Autoren empfehlen daher eine Kombination von isomorphen und nicht-isomorphen Rotationstechniken in großen Displayumgebungen, insbesondere wenn die Wahrnehmung der virtuellen Objekte eng mit der physischen Umgebung übereinstimmt.

Das Experiment hat gezeigt, dass es unter Umständen Unterschiede bei der Verwendung der Techniken in den unterschiedlichen Umgebungen (Desktopdisplay, Großdisplays) gibt und die Mappingtechniken nur bedingt übertragbar sind.

Casiez et al. [CVBC08] untersuchten die Nutzerperformanz bei Selektionsaufgaben bei der Verwendung verschiedener konstanter CD-Gains und nicht-linearer CD-Gains abhängig von der Geschwindigkeit des Zeigergerätes (Maus) in einer Desktopumgebung (20-Zoll LCD mit 1600x1200 px und 100 dpi) und in einer großen Displayumgebung (1,7x1,4 m Display mit 25 dpi). Die konstanten CD-Gains wurden mittels eines Multiplikators variiert und die virtuelle Zeigerbewegung wurde unabhängig von der Bewegungscharakteristik des Eingabegerätes verstärkt. Die nicht-linearen CD-Gains passten die virtuelle Zeigerbewegung dynamisch an die Bewegungsgeschwindigkeit des Eingabegerätes an. In ihrem Experiment konnten die Autoren feststellen, dass ein niedriger CD-Gain negative Auswirkungen auf die Nutzerperformanz hat, während hohe CD-Gains vermehrt ein Überschreiten des Zielobjektes verursachte (hier war eine präzise Muskelkontrolle im kleinen Kontrollraum des Eingabegerätes notwendig). In der großen Displayumgebung war die Zeit für *Clutching* ein dominanter Faktor, der die gesamte Bewegungszeit im Versuchslauf erhöhte.

Casiez et al. verwendeten ein stationäres Eingabegerät, das einen kleinen Kontrollbereich unterstützte (ca. 37 cm) und nicht den verfügbaren Interaktionsraum des Nutzers ausnutzte (z.B. weitgreifende Armbewegungen). Dies führte zu vie-

len *Clutching*-Bewegungen, wenn ein kleiner CD-Gain verwendet wurde und dafür präzises Selektieren möglich war. Bei einem großem CD-Gain hingegen wurde wenig *Clutching* durchgeführt, dafür konnten die Nutzer nicht mehr präzise arbeiten.

Dynamischer CD-Gain

In GHD-Umgebungen müssen die Interaktionstechniken sowohl präzises Manipulieren oder Interagieren aus der Entfernung (ohne störendes Handzittern), als auch effizientes Bewegen von Objekten über große Entfernungen auf dem Display unterstützen. Dieser Kompromiss zwischen der Geschwindigkeit und der gewünschten Genauigkeit kann durch **Variieren des CD-Gain** während der Interaktion gelöst werden. Mit einem niedrigen CD-Gain (z.B. <1) ist die Bewegung des virtuellen Cursors im Darstellungsraum (*display space*) mehrfach niedriger als die tatsächlichen Bewegungen des Eingabegerätes durch den Nutzer im physischen Raum (*motor space*), was ein präzises Interagieren ermöglicht. Für das Überbrücken großer Entfernungen wird ein hoher CD-Gain (z.B. >1) verwendet, so dass kleine Bewegungen im physischen Raum in großen Bewegungen im Darstellungsraum resultieren.

König et al. [KGDR09] haben die Techniken zum Variieren des CD-Gain folgendermaßen unterschieden: *target-oriented*, *velocity-oriented* und *manual-switching*.

Die zielorientierten Techniken (*target-oriented*) verwenden eine Magnetismus-Metapher, indem der CD-Gain verkleinert wird, wenn der Cursor entweder das Zielobjekt betritt [CF03, AA07] oder wenn er sich dem Zielobjekt nähert [BGBL04, BCHE05, GB05] und wieder vergrößert wird, wenn er sich vom Zielobjekt entfernt. Beispielsweise wird bei der Selektionstechnik *Bubble Cursor* [GB05] der Cursorradius abhängig von den benachbarten Zielobjekten dynamisch skaliert, so dass ein Zielobjekt einfach selektiert werden kann ohne versehentlich mehrere naheliegende Objekte mit zu selektieren. Beim *Semantic Pointing* [BGBL04] erhalten Zielobjekte neben der visuellen Größe eine zusätzliche Größe für den Kontrollbereich. Die wichtigen Interaktionsbereiche (z.B. potentielle Zielobjekte) haben größere Kontrollbereiche, als visuell sichtbar, während unwichtige Bereiche (z.B. Leerraum) verkleinerte Kontrollbereiche haben. Somit wird je nach Interaktionsbereich die Genauigkeit oder die Geschwindigkeit unterstützt. Bei diesen Techniken ist ein semantisches Wissen über die virtuelle Umgebung und deren Zielobjekte notwendig und die Verarbeitung einer großen Anzahl von Zielobjekten kann problematisch sein [KGDR09].

Die geschwindigkeitsbasierten Ansätze (*velocity-oriented*) richten sich nach dem *optimized-submovement model* von Meyer et al. [MAK⁺88]; dies besagt, dass gezielte Bewegungen meist aus einer anfänglichen groben und schnellen Bewegung in Richtung des Zielobjektes besteht und anschließend eine langsamere korrigierende Bewegung erfolgt, um das Überschreiten oder Unterschreiten des Zielobjektes (*overshooting*, *undershooting*) zu kompensieren [Bal04]. Die Bewegungsgeschwin-

digkeit der Eingabemodalität im physischen Raum zeigt an, in welcher Phase der Bewegung der Nutzer gerade ist und welche Präzision oder Geschwindigkeit im Darstellungsraum dienlich sein könnte [KGDR09].

Diese Beschleunigungstechniken für Zeigergeräte werden beispielsweise beim Mausverhalten in Betriebssystemen umgesetzt (z.B. MacOS, Windows). Bei schnellen Gerätebewegungen wird angenommen, dass der Nutzer eine große Distanz überbrücken muss. Das heißt die Mausbewegungen können verstärkt werden, um das gewünschte Objekt schnell zu erreichen. Im Gegensatz dazu implizieren langsame Gerätebewegungen, dass das Zielobjekt nah ist und die Mausbewegungen sollten verlangsamt werden, um genaue Einstellungen zu unterstützen [CVBC08]. In Studien wurden unterschiedliche Beschleunigungsfunktionen untersucht, z.B. indem der CD-Gain mittels einer linearen oder nicht-linearen Beschleunigungsfunktion [CVBC08, KGDR09] angepasst wird oder durch Umschalten zwischen konstanten CD-Gain Faktoren [FK05].

Anstatt der Geschwindigkeit wird bei Ramos und Balakrishnan [RB05] Druck zur Erhöhung oder Verringerung des CD-Gain verwendet. Hier wird durch Druck auf eine berührungsempfindliche Oberfläche die Granularität der Manipulationsparameter (z.B. Zoomfaktor) angepasst. Bei schwachem Druck kann der Nutzer grob manipulieren (z.B. schnell in die virtuelle Szene rein oder heraus zoomen) und bei starkem Druck wird die Granularität des Parameterraumes erhöht (z.B. die Schrittweite des Zoomfaktors angepasst).

Bei den Techniken mit manuellem Umschalten (*manual-switching*) wechselt der Nutzer händisch zwischen absoluten und relativen Mappingtechniken, beispielsweise durch Tastendruck auf dem Eingabegerät [KSMB08, NPBL11] oder durch Gesten [MRB05, VB05, FVB06].

Große hochauflösende Displayumgebungen bieten, gegenüber herkömmlichen Desktopumgebungen und großen niedrigauflösenden Displayumgebungen, die Möglichkeit Informationen aus unterschiedlichen Distanzen zum Display zu betrachten. Die Betrachtungsdistanz kann aber nicht nur für die Auflösung der Informationsdarstellung (d.h. geringer oder hoher Detailgrad), sondern auch für die Auflösung der Interaktionen (d.h. grobe oder präzise Manipulation) verwendet werden. Die *distance-aware* Techniken lassen sich im wesentlichen in zwei Gruppen einteilen: zwei Distanzen (Nah- und Fernbereich) und variable Distanzen. Bei der Differenzierung zwischen Nah- und Fernbereich wird präzise Manipulation nah am Display unterstützt und aus der Entfernung kann der Nutzer schnell und grob interagieren [SBA07]. Andere Techniken unterstützen unterschiedliche vordefinierte Distanzen [CB03, VB04, PNB09, JSHKH13].

Im folgenden Abschnitt werden ausgewählte Interaktionstechniken für GHD-Umgebungen vorgestellt, dabei wird auf die verwendete Eingabemodalität und die Mappingmethode eingegangen. Eine schematische Übersicht ausgewählter

verwandter Arbeiten nach ihrer Interaktionsaufgabe und Mappingmethode ist in Abbildung 2.4.3 zu finden.

2.4.2. Verwandte Arbeiten

Herkömmliche 2D Desktopmetaphern (z.B. WIMP) und traditionelle Eingabegeräte, wie Maus und Tastatur sind in großen hochauflösenden Displayumgebungen nicht effektiv nutzbar. Hinzu kommt, dass in solchen Displayumgebungen besondere Probleme bezüglich ihrer Benutzbarkeit auftreten, die in Desktopumgebungen weniger relevant sind (vergleiche [NBC06]). Dazu zählt beispielsweise das Erreichen von entfernten Objekten. Das traditionelle Drag-and-Drop Paradigma funktioniert nicht, der Nutzer benötigt wesentlich mehr Mausbewegungen um ein Objekt über die große Anzeigefläche zu verschieben (*Clutching*). Um dem entgegenzuwirken, werden höhere Mausbeschleunigungen verwendet, die dem Nutzer das Verfolgen des Mauscursors jedoch erschweren. Ebenso ist es für den Nutzer schwierig einen relativ kleinen stationären Mauscursor auf der großen Displaywand zu finden. Ein weiteres Problem sind die Monitorrahmen bei gekachelten Displaywänden (Bezels). Die Displaykanten können vorteilhaft sein, um mehrere Aufgaben auf verschiedenen Displays zu organisieren [BN05a]. Die Bezels sind jedoch problematisch, wenn sich die Darstellung der virtuellen Objekte oder ein Bild über mehrere Displays erstreckt und deswegen eine visuelle Unterbrechung entsteht. Wenn der Cursor über die Displaykanten bewegt wird, entsteht ebenfalls eine Diskrepanz zwischen der tatsächlichen Bewegungsbahn des Cursors und dem vom Nutzer erwarteten Cursorpfad.

In den letzten Jahren wurden eine Reihe neuer VR Interfaces und 3D Interaktionstechniken für große Displayumgebungen entwickelt, z.B. *Mouse Ether*, *OneSpace* und *snapping and bumping* für die Handhabung der Bezels [RCB⁺05] oder *drag-and-pop* als Alternative zu Drag-and-Drop [BCC⁺03]. Ein Überblick über die Interaktionstechniken in GHD-Umgebungen ist in [RCB⁺05, BBM⁺06, NSS⁺06, Kha11] zu finden.

Wie bereits erwähnt werden bestimmte Aufgaben eher nahe am großen hochauflösenden Display ausgeführt und andere Aufgaben besser aus der Entfernung [NSS⁺06]. Ebenso benötigt die Selektion und Manipulation von kleinen oder physisch unerreichbaren Objekten unterschiedliche Interaktionstechniken. Das heißt, es gibt Aufgaben, die schnelle und ungenaue Aktionen erfordern (z.B. effizientes Bewegen von Objekten über große Entfernungen auf dem Display) als auch Aufgaben, die genaue und langsame Aktionen benötigen (z.B. präzise Positionierung von Objekten trotz natürlichem Handzittern). Die Herausforderung bei den Interaktionstechniken in GHD-Umgebungen liegt darin, wie zwischen den Aufgabenbereichen, die oftmals eine unterschiedliche Präzision oder Sensitivität der Nutzereingabe erfordern, gewechselt werden kann.

2.4.2.1. Nutzereingabe nahe am Display

In großen Displayumgebungen mit berührungsempfindlichen Oberflächen kann die Nutzereingabe direkt am Display erfolgen. Beispielsweise wird die Beschleunigung des Zeigers (Stift oder Touch-Eingabe) für eine schnelle oder genaue Platzierung von Icons auf großen virtuellen Desktopflächen verwendet (z.B. *push-and-pop* [CHBL05], *drag-and-pop* [BCC⁺03]) oder ein Finger kontrolliert die Cursorbewegungen, während ein Finger der anderen Hand die Cursorgeschwindigkeit manipuliert (*Dual Finger X-Menu* [BWB06]). Beim *HybridPointing* von Forlines et al. [FVB06] wurde eine Stiftinteraktion für große hochauflösende Displays umgesetzt. Das Umschalten zwischen absoluten und relativen Mapping erfolgt durch Tippgeste mit dem Eingabegerät. Für das relative Mapping (für schnelle oder präzise Interaktion) wird die Bewegung des virtuellen Cursors abhängig von der Geschwindigkeit des Interaktionsstiftes dynamisch durch den CD-Gain angepasst.

2.4.2.2. Nutzereingabe entfernt vom Display

Einige Arbeiten beschäftigen sich mit Interaktionstechniken, die den Nutzer aus unterschiedlichen Distanzen zum Display interagieren lassen. Bei Malik et al. [MRB05] sitzt der Nutzer in einiger Entfernung zum Display, so dass er die virtuelle Arbeitsfläche überblicken kann und er interagiert über Gesten mit dem System. Beispielsweise werden Gesten der linken Hand zur groben Positionierung von virtuellen Objekten verwendet und Gesten mit der rechten Hand für die Feinpositionierung. Andere Ansätze verwenden mobilen Endgeräten als Interaktionsgerät in großen Displayumgebungen (z.B. PDA, Smartphone, Tablet). Hier wird oft indirekte Manipulation mit dem Display unterstützt, indem ein selektierter Ausschnitt der virtuellen Umgebung auf dem Interaktionsgerät angezeigt wird (z.B. [PDW⁺11, JHKB10]) oder Zusatzinformationen über das Interaktionsgerät abgerufen werden können (z.B. [TMM⁺09]).

Grossman et al. [GBK⁺01] bieten eine beidhändige Interaktionstechnik für die Modellierung von 3D Modellen durch Zeichnen von 2D Profilkurven auf großen Displays. Neben der Darstellung der Objekte in Originalgröße soll es dem Designer möglich sein die dargestellten Objekte zu erstellen, zu manipulieren, zu explorieren und zu annotieren. In ihrem Prototypen [BFBK00, GBK⁺01] konnten Designer mittels *digital tape-drawing* Technik die Umrisse von Automobilen skizzieren. Dabei handelt es sich um eine Klebebandtechnik, mit der Designer gerade und geschwungene Linien exakter skizzieren können. Die Interaktionstechnik wurde von der realen Welt in die digitale Welt übertragen, indem der Nutzer mit dem Eingabegerät in der einen Hand das virtuelle Klebeband festhält und mit dem Eingabegerät der anderen Hand das Klebeband virtuell abrollt. Die Nutzung des Systems konnte von den Designern schnell erlernt werden, da die gleichen Bewegungsabläufe in der realen und virtuellen Umgebung angewendet wurden. Beim Skizzieren der 2D

Profilkurven ist Isomorphie erwünscht weshalb ein 1:1 Mapping der Bewegungsabläufe erfolgt.

Pointing

Damit ein Nutzer mit Objekten (z.B. GUI-Elemente, grafische Primitive, 3D Geometrien) in der virtuellen Umgebung interagieren kann, ist eine der grundlegenden Aktionen die Selektion von virtuellen Objekten. Oftmals werden *Ray-Casting* Techniken verwendet, bei denen ein imaginärer Strahl von dem Finger oder der Hand des Nutzers aus in die virtuelle Umgebung projiziert wird und wenn der Strahl ein Objekt trifft, dann wird dieses Objekt selektiert. Jedoch ist die Zeigepersistenz oft durch das natürliche Handzittern und die begrenzte Präzision der Handmotorik eingeschränkt (vergleiche Abschnitt 2.1.2). Als Eingabegerät werden meist Laserpointer verwendet, bei denen der virtuelle Cursor (z.B. [DC02, CP03, ATK⁺05]) oder der reale Punkt des Laserpointers (z.B. [SRH05b, KBSR07, KGDR09]) für den Nutzer nicht sichtbar ist, um visuelle Diskrepanzen zwischen realer Eingabe und visueller Repräsentation zu vermeiden.

Die *improved virtual pointer* Metapher (IVP) von Steinicke et al. [SRH05b, SRH05a] verwendet eine Ray-Casting Technik mit einer Art getrackten Laserpointer (6 DOF). Dabei wird der virtuelle Laserstrahl so abgelenkt, dass er das nächstliegende 3D Objekt zum Strahl auswählt, welches der Nutzer dann selektieren kann (gekrümmter Laserstrahl). Der virtuelle Cursor bleibt solange auf dem Objekt, bis ein anderes Objekt sich näher am virtuellen Laserstrahl befindet. Für die Manipulation verwenden sie ein 1:1 Mapping für die Objektrotation und für die Objektpositionierung eine lineare Mappingfunktion, welche die Translation skaliert. Dadurch lassen sich virtuelle Objekte im 3D Raum einfach erreichen und manipulieren.

Andujar und Argelaguet [AA07] verwenden virtuelle Fenster, welche über die virtuelle 3D Welt gelegt werden, um eine schnelle und genaue Selektion von 2D Widgets zu unterstützen (*anisomorphic ray-casting*). Wenn der Selektionsstrahl des Nutzers auf ein virtuelles Fenster zeigt, dann wird der CD-Gain zwischen der Orientierung der Nutzerhand und dem Selektionsstrahl angepasst. Damit rotiert der Selektionsstrahl langsamer als die Nutzerhand, dass den Tremor minimiert und die Selektionsgenauigkeit erhöht. Die Autoren verwendeten keine dynamische Anpassung des CD-Gain, sondern eine nutzerdefinierte Konstante, um die Individualität der Nutzer zu unterstützen.

Jota et al. [JPJ09] haben in einer Studie untersucht, wie die Entfernung des Nutzers zum großen Display die Interaktion beeinflusst. Sie wählten dazu die verbreiteten Metaphern *Grab*, *Point* und *Mouse*, die jeweils unterschiedliche Interaktionsdistanzen repräsentieren. Bei der *Grab* Metapher erfolgt die Interaktion sehr nahe am Display; der Nutzer geht zum virtuellen Objekt, greift dieses, bewegt sich zur neuen Position und lässt das Objekt wieder los. Die *Point* Metapher ist für die Interaktion aus mittlerer Distanz; um ein virtuelles Objekt zu bewegen, zeigt der Nutzer auf das Objekt, selektiert es, zeigt auf die neue Position und lässt das Objekt los.

Die *Mouse* Metapher funktioniert wie die herkömmliche Mausinteraktion, aber der Zeiger wird auf einer vertikalen Ebene bewegt. Diese Metapher wird nicht durch die Distanz des Nutzers zum Display beeinflusst und ist für entfernte Interaktion geeignet. Im Experiment mussten die Probanden ein Puzzle lösen, indem 2D Objekte in 2D Silhouetten einsortiert werden mussten, d.h. die Objekte mussten selektiert, bewegt und korrekt positioniert werden. Die drei Interaktionstechniken zur Selektion und Positionierung (*Grab*, *Point*, *Mouse*) wurden für jede Distanz (nah, mittel, fern) getestet und die Bearbeitungszeit gemessen, sowie alle Gesten und Nutzerbewegungen aufgezeichnet. Die schnellste Bearbeitungszeit wurde mit der *Point* Metapher erreicht, ungeachtet der Distanz und Benutzerbewegungen. Weiterhin wurde die *Point* Metapher von den meisten Probanden bevorzugt (10 von 21 Probanden, wobei 6 Probanden die *Mouse* Metapher und 5 Probanden die *Grab* Metapher bevorzugten).

Wie Studien gezeigt haben, kann die Nutzerperformanz verbessert werden, wenn die physische Navigation in GHD-Umgebungen unterstützt wird (vgl. Abschnitt 2.3.1). Dieser Aspekt wird in den folgenden genannten Arbeiten genutzt.

Eine gerätefreie Interaktion mit GHDs wurde von Stødle et al. [SBA07] entwickelt, welche einen hybriden Ansatz aus Computer Vision Techniken (2D Körperposition, Hände, Finger) und Geräuscherkennung (Finger schnippen) für die Lokalisation des Nutzers verwendet und gestenbasierte Interaktion aus der Entfernung und ganz nah am Display unterstützt.

Vogel und Balakrishnan [VB05] unterstützen die Interaktion mit virtuellen Objekten aus der Entfernung durch Gesten. Sie definierten verschiedene Handposen, mit denen der Nutzer explizit zwischen absolutem und relativem Modus wechseln kann. Im absoluten Modus wird ein konstanter CD-Gain verwendet und eine konventionelle Beschleunigungsfunktion für den CD-Gain im relativen Modus. Sie verglichen ihre 2-Modus Technik (*RayToRelative*) mit einer ausschließlich absoluten Selektionstechnik (*Ray-Casting*) und einer rein relativen Technik (*Relative*). In der Studie war der absolute Modus (*Ray-Casting*) signifikant schneller als die anderen beiden Techniken, insbesondere bei der Selektion von großen Zielobjekten und wenn *Clutching* benötigt wurde. Jedoch war eine signifikant höhere Fehlerrate beim absoluten Modus (mittlere Fehlerrate 22,5%), als beim relativen Modus (3,5%) oder beim *RayToRelative* (5,7%) das Ergebnis. Als guten Kompromiss zwischen Genauigkeit und Geschwindigkeit hat sich die Kombination von absolutem und relativem Modus bei der Interaktionstechnik von Vogel und Balakrishnan herausgestellt.

Bei der Technik *PRISM* von Frees et al. [FK05] wird der CD-Gain dynamisch zwischen der Nutzereingabe (Hand) und dem zu kontrollierenden Objekt in einer virtuellen 3D Umgebung angepasst. Sie gehen davon aus, dass Nutzer abhängig von ihrer Intention, Objekte mit unterschiedlicher Geschwindigkeit manipulieren. Daher verwenden sie die Handgeschwindigkeit des Nutzers, um zwischen präzisiertem und unpräzisiertem Modus implizit zu wechseln. Bei welcher Geschwindigkeit

der Nutzer im präzisen oder unpräzisen Modus ist wird durch eine individuelle Skalierungskonstante festgelegt. Ihre Evaluationsergebnisse zeigten eine klare Verbesserung bei präziser Positionierung von virtuellen 3D Objekten verglichen mit einem rein absoluten Mapping. Auch wenn diese Technik für *head-mounted-display* Umgebung entwickelt wurde, ist sie in großen Displayumgebungen anwendbar (z.B. [AA07]).

Um die Zeigeperformanz vor allem bei größeren Distanzen und hohen Bildschirmauflösungen zu verbessern haben König et al. das *Adaptive Pointing* entwickelt [KGDR09]. Die Technik ermöglicht es den CD-Gain bei absoluten Eingabegeräten implizit an das aktuelle Benutzerbedürfnis anzupassen. Die dynamische Anpassung des CD-Gain erfolgt abhängig von der Bewegungsgeschwindigkeit und dem aktuellen Offset⁷. Adaptive Pointing erscheint dem Nutzer als rein absolutes Mapping, aber der CD-Gain wird unbemerkt verringert, wenn eine genaue Präzision benötigt wird. Im Experiment wurde das Adaptive Pointing mit einer absoluten Zeigetechnik (Laserpointer als Eingabegerät) verglichen. Beim Adaptive Pointing konnte eine signifikante Performanzverbesserung bezüglich Bewegungszeit (19% schneller), Fehlerrate (63% geringer) und Nutzerzufriedenheit erreicht werden.

Nanchel et al. [NPBL11] haben verschiedene Techniken (*dual-precision techniques*) verglichen, die ein Umschalten zwischen grobem Positionierungsmodus und präzisen Selektionsmodus ermöglichen. Für den groben Zeigemodus (*Pointing*) wurde eine absolute Mappingtechnik (*RayCasting*) verwendet und für den präzisen Zeigemodus wurden relative Translations- oder Rotationsbewegungen verwendet. Zum Vergleich der *dual-precision techniques* wurde außerdem eine *velocity-oriented* Technik umgesetzt, die den CD-Gain dynamisch an die Geschwindigkeit des Eingabegerätes (kabellose Maus) anpasst. Die Autoren stellten fest, dass für die Selektion von kleinen Zielobjekten (< 30 mm) aus der Entfernung die *dual-mode* Techniken besser funktionieren als die klassischen Techniken. Die Kombination aus *RayCasting* mit Translations- oder Rotationsbewegungen des Eingabegerätes im präzisen Modus erzielte die bessere Performanz und die höhere Nutzerakzeptanz.

Kopper et al. [KSMB08] haben zwei Interaktionstechniken für präzises Zeigen in großen Displayumgebungen entwickelt, die auf dem Ray-Casting Prinzip beruhen und eine beidhändige Eingabemodalität (6-DOF Zeigegerät) unterstützt. Die Interaktionstechnik ARM (*Absolute and Relative Mapping*) verwendet die dominante Hand für Zeigeinteraktion mit absolutem Mapping, was schnelle und grobe Interaktion ermöglicht. Mit der nicht-dominanten Hand wird in den relativen Modus geschaltet, dass ein präziseres Interagieren ermöglicht. Das relative Mapping nimmt der Nutzer als "slow motion" des Cursors wahr. Bei der Interaktionstechnik ZELDA (*Zoom for Enhanced Large Display Acuity*) wird die dominante Hand, wie bei ARM zur Steuerung des Cursors verwendet und die nicht-dominante Hand kontrolliert das zusätzliche Zoom-Fenster bezüglich Position, Größe und Zoomfaktor.

⁷ Der Offset reduziert die Abweichung zwischen der Position im realen Raum und der Position im virtuellen Raum, die beim relativen Mapping auftritt (Drift-Effekt).

Das Zoom-Fenster ist eine vergrößerte Ansicht der virtuellen Objekte und erhöht nicht nur die Präzision für die Interaktion, sondern auch die visuelle Sehschärfe der virtuellen Objekte. In einer Nutzerstudie konnte gezeigt werden, dass bei Tasks, die eine hohe Präzision erfordern, beide Interaktionstechniken die Performanz verbessern verglichen mit dem herkömmlichen Ray-Casting. Jedoch beobachteten Kopper et al., dass die meisten Probanden eine feste Position zum Interagieren bevorzugten und nur nahe an das Display herangingen, wenn es absolut nötig war. Diese Beobachtungen widersprechen den Ergebnissen von Ball et al. [BNB07], dass Nutzer die physische Navigation der virtuellen Navigation in GHD-Umgebungen bevorzugen.

Navigation

Neben Interaktionstechniken für Selektion und Manipulation gibt es für die Navigationsaufgaben (z.B. *pan-and-zoom*) ebenfalls Techniken, die unterschiedliche Interaktionslevel benötigen. Beispielsweise können Nutzereingaben wie Beschleunigung und Position auf die Navigationsparameter abgebildet werden. Nancel et al. [NWP⁺11] untersuchten gestenbasierte Interaktion für Navigationsaufgaben, wie *pan-and-zoom*. Sie identifizierten dabei drei Einflussfaktoren für die Navigationsgesten in der Luft: (1) 1- oder 2-händige Eingabe, (2) Längs- oder Kreisbewegungen der Gesten, (3) Bewegungsbeschränkungen (*1D path* (Scrollrad), *2D surface* (berührungsempfindliche Oberfläche), *3D free* (Luft)). Diese Faktoren wurden kombiniert und in einer Studie in einer GHD-Umgebung untersucht. Die Gesten, die in der Luft ausgeführt wurden, waren weniger effizient und führten eher zu Ermüdung als die gerätebasierte Eingabe. Wurde die Bewegungsbeschränkung bei der Gesteneingabe verwendet, erhöhte dies die Genauigkeit. Überraschenderweise waren die linearen Gesten generell effizienter als die *Clutching*-freien kreisförmigen Gesten. Die Probanden bevorzugten in der Studie die zweihändigen Interaktionstechniken (8 von 12 Personen) und die Gesten mit Längsbewegungen (10 von 12 Personen).

Ball et al. [BDN⁺06] verwendeten die Kopforientierung als Eingabemodalität zur Navigation (*pan*) in virtuellen 2D Landkarten. Dabei wird die virtuelle Karte nach rechts verschoben wenn der Nutzer seinen Kopf nach links dreht und umgekehrt. Das Head-Tracking wurde mit einem Orientierungstracker in der Hand und der traditionellen Mausinteraktion (*click-and-drag*) als Navigationstechnik verglichen. Hierbei erzielte das Head-Tracking schlechte Performanzzeiten und Nutzerpräferenzen. Beim Head-Tracking waren die virtuelle Navigation und die Kopfbewegungen eng miteinander verbunden, dies erschwerte dem Nutzer, die virtuelle Karte zu verschieben, während er sie außerdem absuchte.

2.4.2.3. Distanzabhängige Nutzereingabe

Große hochauflösende Displayumgebungen bieten die Möglichkeit, Informationen aus unterschiedlichen Distanzen zum Display zu betrachten. Die Betrachtungsdi-

stanz kann aber nicht nur für die Auflösung der Informationsdarstellung genutzt werden, sondern auch für die Auflösung der Interaktionen (d.h. grobe oder präzise Manipulation) verwendet werden. Daher nutzen einige Arbeiten den aktuellen Abstand des Nutzers zum Display für die Nutzereingabe.

VisionWand von Cao und Balakrishnan [CB03] verwendet einen Stab als Eingabemodalität zur Interaktion mit großen Displays. Der passive Stab mit farbigen Endpunkten wird mit Hilfe von Computer Vision Techniken im physischen 3D Raum getrackt und die virtuellen Cursors werden mittels orthogonaler Projektion auf die Displaywand projiziert. Sie entwickelten eine Reihe von Gesten und Posen für die Interaktion mit Widgets. Aufgrund des sehr kleinen *motor space* des Stabes (-60 bis +60 Grad) wird bei der Manipulation kontinuierlicher Parameter (z.B. Farbwerte) der Displayabstand für die Granularität der Interaktion verwendet. Wird in einem größeren Abstand vom Display mit dem Stab interagiert, dann wird ein größeres Panel angezeigt und die Parametermanipulation ist schneller und gröber. Wenn der Stab näher am Display ist, wird ein kleineres Panel angezeigt und die Anpassung ist feiner und langsamer. Das Panel ist ein kreisförmiger Slider, der per Kippbewegungen des Stabes manipuliert wird.

Die Verwendung des Körperschattens des Nutzers als Eingabemodalität wurde von Shoemaker et al. [STKB10] vorgestellt. An den Körperschatten sind virtuelle Tools (z.B. virtuelle Medienbibliothek des Nutzers) gebunden, die dem Nutzer automatisch folgen und somit immer verfügbar sind. Eine direkte Interaktion mit der Anwendung erfolgt über den virtuellen Cursor, der an jede Schattenhand gebunden ist (Buttonevents werden durch zusätzliche Eingabegeräte in der Hand ausgelöst). Als Interaktionsaufgaben werden Navigation (*pan-and-zoom*), Symboleingabe durch Textannotation mittels Softkeyboard und Widget-Interaktion unterstützt. Um präzise Manipulation oder Selektion entfernter Objekte zu unterstützen, kann die Projektion des Nutzerschattens durch dynamische Positionierung der (virtuellen) Lichtquelle verändert werden. Shoemaker et al. verwenden den Abstand des Nutzers zum Display um die Lichtquelle zu steuern und somit den CD-Gain dynamisch zu kontrollieren. Trotz halb-transparentem Schatten wird der darunterliegende Displaybereich zum Teil verdeckt.

Vogel und Balakrishnan nutzen bei ihrem *interactive public ambient display* [VB04] die physische Navigation, um zwischen expliziter und impliziter Interaktion zu unterscheiden. Für die implizite Interaktion wird die Körperposition und -orientierung im physischen Raum getrackt, d.h. wenn der Nutzer an dem Display vorbei läuft, werden nur allgemeine Informationen angezeigt, bewegt er sich auf das Display zu werden detaillierte Informationen sichtbar. Der Interaktionsraum vor dem Display ist in unterschiedliche Zonen (*Ambient Display, Implicite Interaction, Subtle Interaction, Personal Interaction*) eingeteilt und je näher der Nutzer zum Display geht desto persönlichere Informationen werden angezeigt. Für die explizite Interaktion mit den Informationen werden einfache Handgesten (Manipulation aus der Entfer-

nung) und Touchscreen Eingaben (Interaktion mit den persönlichen Daten direkt am Bildschirm) unterstützt.

Peck, North und Bowman [PNB09] nutzen bei ihrer 3D Interaktionstechnik (*Multiscale Interaction*) die Vorteile der physischen Navigation, indem abhängig von der Distanz des Nutzers zum Display einerseits die Skalierung der Wahrnehmung und andererseits die Skalierung der Interaktion kontrolliert werden. Wenn der Nutzer seine physische Position relativ zum Display ändert, dann wird der *multiscale cursor* angepasst wodurch die Daten auf dem Display interaktiv neu skalieren. Es wurden Selektions- und Navigationstechniken umgesetzt, bei denen ein 3D Eingabegerät zur direkten Interaktion mit einem großen 2D Informationsraum auf dem GHD verwendet wird. Bei der kontinuierlichen Selektionstechnik wächst die Größe des 2D Cursors mit zunehmendem Abstand zum Display (exponentielles Mapping). Bei der hierarchischen Ebenen-Selektion ändert sich die Cursorgröße in diskreten Intervallen entsprechend der vordefinierten hierarchischen Struktur im Datenraum. Bei der geografischen Informationsvisualisierung wird z.B. die Selektionsskala an die verschiedenen topografischen Grenzen (z.B. *District, County, State, Country*) angepasst. Bei der Navigationstechnik wird die virtuelle Navigation (*pan-and-zoom*) durch eine automatische Rate-Anpassung der Navigationssteuerelemente basierend auf der relativen Position des Nutzers zum Display unterstützt. Die Objekte bewegen sich durch das Sichtfeld mit einer konstanten Geschwindigkeit ungeachtet der Entfernung zwischen Benutzer und Objekt. In einer Studie wurde die physische Navigationstechnik (*Multiscale Interaction*) mit einer expliziten Interaktionstechnik (manuelles Umschalten zwischen Skalenlevels über ein Menü auf dem PDA) und einer Lasso-Technik (konstante Cursorgröße im Detaillevel und der Nutzer musste höher skalierte Elemente mittels einer Zeichengeste auswählen) verglichen. Die Aufgabe bestand darin, ein hierarchisches Puzzle zu lösen. Die physische Navigationstechnik hat gezeigt, dass die Aufgabe konsistenter gelöst werden konnte, d.h. Nutzer ohne Erfahrung in GHD-Umgebungen konnten sich mit den erfahrenen Nutzern messen. Während bei den anderen beiden Interaktionstechniken eher Abweichungen oder Ausreißer (z.B. Anzahl beim Tauschen der Puzzleteile, Bearbeitungszeit) zwischen den Nutzern entstanden. Die Studie zeigt, dass die Skalierung der Interaktion das natürliche Verhalten des Nutzers in GHD-Umgebungen unterstützt.

Jakobsen et al. diskutieren in [JSHKH13], wie die räumliche Relation zwischen dem Nutzer und der Informationsdarstellung (Abstand, Orientierung, Bewegung, Position) für einzelne Visualisierungsaufgaben (z.B. *visualize, filter, select, coordinate*) in GHD-Umgebungen genutzt werden kann. In formativen Studien wurden drei Interaktionstechniken mit jeweils sechs Probanden getestet: Navigation durch physische Bewegung, Erkunden von koordinierten Ansichten durch Bewegung und Anpassung visueller Repräsentationen durch Abstand. Die Autoren konnten beobachten, dass die Interaktion mit der Informationsdarstellung durch räumliche Relation den Probanden eine physische Wahrnehmung für die abstrakten Daten

boten. Die Probanden empfanden, dass die Änderung von Zoomstufen oder der Repräsentation mit Hilfe des Betrachtungsabstandes den Benutzungsaufwand reduziert und eine flüssige Interaktion ermöglicht verglichen mit der Nutzereingabe per Gyro-Maus (als Referenzeingabe in den Studien verwendet). Die Skalierung der Datenansichten durch den Abstand funktionierte allerdings konträr zu den Erwartungen der Probanden. In der dritten Studie wurden die Nutzerbewegungen auf die Selektion von Attributwerten abgebildet. Einige Probanden konnten somit ihren Fokus auf die Datenansicht fixieren, während sie die Selektion durch Körperbewegung änderten. Jedoch wurde manchmal das Attribut ungewollt geändert, wenn der Nutzer sich bewegte (z.B. Brushing von koordinierten Ansichten).

In Abbildung 2.4.3 wurden ausgewählte Interaktionstechniken nach ihren Interaktionsaufgaben und dem verwendeten Mapping einsortiert. Vorrangig werden in dieser Arbeit Interaktionstechniken für große hochauflösende Displayumgebungen betrachtet, die sowohl präzise als auch grobe Nutzerinteraktion unterstützen und Interaktionstechniken, die dafür adaptierbar sind. Bei den Interaktionsaufgaben wurden die Menü-Interaktionen, wie Symboleingabe und Systemkontrolle unter Interaktion mit Widgets zusammengefasst.

		Mappingmethoden			
		zielorientiert (target-oriented)	geschwindigkeitsbasiert (velocity-oriented)	manuelles Umschalten (manual-switching)	distanzabhängig (distance-aware)
Interaktionsaufgaben	Selektion/ Positionierung	Grossman et al., 2005 [GB05] (Bubble Cursor) Blanch et al., 2004 [BGBL04] (Semantic Pointing) Steinicke et al., 2005 [SRH05a, SRH05b] (improved virtual pointer)	König et al., 2009 [KGDRO9] (Adaptive Pointing)	Vogel et al., 2005 [VB08] (RayToRelative) Forlines et al., 2006 [FVB06] (HybridPointing) Nancel et al., 2011 [NPBL11] (dual-precision)	Vogel et al., 2004 [VB04] (public ambient display) Peck et al., 2009 [PNB09] (Multiscale Interaction) Jakobsen et al., 2013 [JSHKH13] (visualization and proxemics) Stödle et al., 2007 [SBA07] (hybrid sound/vision-based)
	Widgets	Andujar et al., 2007 [AA07] (anisomorphic raycasting)	Ramos et al., 2005 [RB05] (Zliding) Collomb et al., 2005 [CHBL05] (push-and-pop)	Malik et al., 2005 [MRB05] (multi-finger gestures)	Cao et al., 2003 [CB03] (VisionWand) Vogel et al., 2004 [VB04] (public ambient display) Shoemaker et al., 2010 [STKB10] (body-based interfaces)
	Navigation		LaViola et al., 2001 [LFKZ01] (Multi-scale Navigation)		Peck et al., 2009 [PNB09] (Multiscale Interaction) Jakobsen et al., 2013 [JSHKH13] (visualization and proxemics)
	Manipulation		Frees et al., 2005 [FK05] (PRISM)	Kopper et al., 2008 [KSMB08] (ARM,ZELDA)	Interaction Scaling

Abbildung 2.4.3.: Schematische Einordnung ausgewählter verwandter Arbeiten zu Interaktion in großen Displayumgebungen nach Interaktionsaufgabe und der verwendeten Mappingmethode. Interaction Scaling ist der vorgestellte Ansatz als Beitrag zur Interaktion in GHD-Umgebungen mit Fokus auf Navigations- und Manipulationsaufgaben.

Die verschiedenen Interaktionstechniken lassen sich nach den Interaktionsaufgaben oder der Nutzereingabe einordnen. Typischen Interaktionsaufgaben in GHD-Umgebungen sind: (1) Selektion und Positionierung [CP03, VB05, FVB06, KSMB08, KGDR09, PNB09, NPBL11], (2) Widget-Interaktion [GSW01, CB03, CHBL05, MRB05, SRH05a, SBA07, STKB10], (3) Manipulation [CB03], (4) Modellierung [GBK⁺01] und (5) Navigation [BDN⁺06, PNB09, STKB10, NWP⁺11, JSHKH13].

Als Nutzereingabe werden beispielsweise: (1) gestenbasierte Techniken [MRB05, VB05, SBA07], (2) stiftbasierte Techniken [GSW01, CHBL05, FVB06], (3) Laserpointer [DC02, ATK⁺05, KGDR09], (4) getrackte Stäbe oder kabellose Mäuse [CB03, SRH05b, KSMB08, PNB09] verwendet, sowie (5) Kombinationen aus unterschiedlichen Eingabemodalitäten [CP03, VB04, BDN⁺06, STKB10, NWP⁺11, NPBL11].

2.5. Anforderungen für Interaktion in großen hochauflösenden Displayumgebungen

Für die Interaktion in großen hochauflösenden Displayumgebungen werden Methoden benötigt, die den Nutzer bei seiner (gegenwärtigen) Tätigkeit unterstützen, indem die Visualisierung und die Interaktivität an den Nutzer angepasst werden (**distanzabhängige Interaktion**). Eine Herausforderung stellt dabei der Wechsel zwischen den unterschiedlichen Interaktionsdistanzen dar, welche eine angepasste Präzision und Sensitivität der Nutzereingabe erfordern.

Die Literaturrecherche hat verschiedene Kriterien für die distanzabhängige Interaktion in GHD-Umgebungen aufgezeigt. Dabei wird zwischen den Anforderungen unterschieden, welche durch die Displaykonfiguration, die Eingabemodalität, das Mapping, die Interaktionsaufgabe und den Benutzer bedingt sind.

Displaykonfiguration

Große Displayumgebungen beeinflussen die Nutzerperformanz aufgrund ihrer physikalischen Eigenschaften (Hardware). Sie fördern die Performanz und Produktivität der Nutzer, wobei die Displayauflösung an die Wahrnehmung des Nutzers (d.h. foveales Sehen und peripheres Sehen) und an die Datendarstellung angepasst werden sollte. Beispielsweise sind hochauflösende Displays für die Darstellung von sehr detaillierten Daten geeigneter als Displays mit geringer Pixeldichte [BN05b]. Die Anordnung der Displays zu einer großen Displayumgebung hat Einfluss darauf ob Einzelbenutzer- oder Mehrbenutzer-Szenarien unterstützt werden. Das Vorhandensein von Bezels kann für bestimmte Aufgaben positiv genutzt werden (z.B. Taskseparation) oder hinderlich sein (z.B. verzerrte Darstellung). Wie Studien gezeigt haben, wird durch die physische Navigation die Nutzerperformanz in GHD-Umgebungen erhöht, vor allem bei Navigations- und Suchaufgaben in der Informationsvisualisierung [SBY⁺06, BNB07, BI12].

DISPLAYKONFIGURATION – Für die Interaktion in GHD-Umgebungen müssen folgende **Faktoren der Displaykonfiguration** beachtet werden: Displayfläche, Displayauflösung, Displayanordnung, Bezels und Unterstützung für die physische Navigation.

Eingabemodalität

Im Anwendungsgebiet von großen hochauflösenden Displayumgebungen kann zwischen erweiterten Desktopumgebungen und Fullscreen-Anwendungen unterschieden werden. Erweiterte Desktopumgebungen ergänzen die Arbeitsfläche eines Desktopsystems um zusätzliche Displays und die Nutzerinteraktion ist zwischen verschiedenen Anwendungen möglich. Hier werden hauptsächlich *everyday tasks* ausgeführt. Dazu müssen die Desktopmetaphern (z.B. Drag-and-Drop Ak-

tionen, Maus- und Tastatureingabe) angepasst werden, dass beispielsweise in den Arbeiten von Microsoft Research [BCC⁺03, RCB⁺05] und Stanford Interactive Mural [GSW01] erfolgte. Bei Fullscreen-Anwendungen wird die gesamte Displayfläche von einer Anwendung verwendet, in der die Nutzerinteraktion erfolgt. Für die Informationsvisualisierung in GHD-Umgebungen werden meist Fullscreen-Systeme verwendet, bei denen eine Maus- und Tastatureingabe von einer festen Nutzerposition aus negativen Einfluss auf die Nutzerperformanz hat und daher physische Navigation unterstützt werden sollte.

Um physische Navigation als natürliche Interaktion in GHD-Umgebungen zu ermöglichen, muss der Nutzer sich frei vor dem Display bewegen können.

EINGABEMODALITÄT – Somit ergeben sich folgende **Anforderungen für die Eingabemodalität**: mobil, schnell verfügbar, leicht anzuwenden und aus unterschiedlichen Entfernungen nutzbar.

Mapping

In GHD-Umgebungen interagiert der Nutzer aus unterschiedlichen Entfernungen und bewegt sein Eingabegerät im freien Raum vor dem Display (räumliche Interaktion). Dabei müssen unterschiedliche Genauigkeiten der Nutzerinteraktion (grob vs. präzise) abhängig von der Interaktionsaufgabe (Schnelligkeit vs. Genauigkeit) unterstützt werden. Einerseits soll eine Objektplatzierung schnell erfolgen, was aufgrund der menschlichen Physiologie nur mit grober Genauigkeit möglich ist. Andererseits ist eine genaue Objektplatzierung nur mit langsamen physiologischen Bewegungen möglich. Wie die 3D Interaktion des Nutzers im physischen Raum in den virtuellen 3D Darstellungsraum abgebildet werden kann, hängt von verschiedenen Parametern ab, wie Realismus, Kinematik, Eingabeart, Projektion und dynamischer Anpassung des CD-Gain (vergleiche Abschnitt 2.4.1.4).

Die Literaturrecherche hat gezeigt, dass oft *velocity-oriented* oder *manual-switching* für die dynamische Anpassung des CD-Gain verwendet wird. Eine weitere Möglichkeit sind *target-oriented* Ansätzen, bei denen jedoch ein semantisches Wissen über die virtuelle Umgebung notwendig ist, weshalb diese eher bei 2D GUIs eingesetzt werden. Es gibt kaum Ansätze, welche den Abstand des Benutzers zum Display nutzen (*distance-aware*). In der vorliegenden Arbeit wird ein *distance-aware* Mapping umgesetzt, das neben vordefinierten auch kontinuierliche Distanzen zur Anpassung der Interaktion unterstützt.

Ein distanzabhängiger Ansatz hat den Vorteil, dass das Konzept auch im *Smart Meeting Room* (SMR) [HK05, Thi10] anwendbar ist. Im SMR sitzen die Nutzer an unterschiedlichen Positionen im Raum (U-förmige oder O-förmige Tischplätze) und jeder Nutzer hat eine andere Entfernung zu den Großdisplays. Hier können positionsabhängig vordefinierte CD-Gains auf die Nutzereingabe angewendet werden. Ebenso ist ein *manual-switching* Mapping mit unterschiedlichen CD-Gains denkbar.

Im SMR sind *velocity-oriented* Ansätze aufwendig umsetzbar, da aufgrund der unterschiedlichen Eingabemodalitäten (z.B. Maus, Laserpointer, Smartphones, Touchpad) gerätespezifische Geschwindigkeits- oder Beschleunigungsberechnungen benötigt werden.

MAPPING – Für die Interaktion in GHD-Umgebungen sind die **Anforderungen für das Mapping** folgende: relative Unabhängigkeit von der Eingabemodalität, Berücksichtigung der gegenwärtigen Interaktionsdistanz und eine dynamische Anpassung des CD-Gain.

Interaktionsaufgabe

LaViola et al. [LK07] haben in ihrer Studie gezeigt, dass es unter Umständen Unterschiede bei der Anwendung der Interaktionstechniken in den verschiedenen Umgebungen gibt (herkömmliches Desktopsystem vs. große Displays) und die Mappingtechniken nur bedingt übertragbar sind. Für große hochauflösende Displayumgebungen müssen dementsprechend die Mappingtechniken für die Interaktionsaufgaben getestet werden, um allgemeine Richtlinien für eine ergonomische Interaktion ableiten zu können. In den letzten Jahren wurden zahlreiche Positionierungs- und Selektionstechniken für große Displayumgebungen entwickelt und in Studien untersucht (vergleiche Abschnitt 2.4.2). Die Literaturrecherche hat gezeigt, dass es noch wenige Untersuchungen zu Interaktionstechniken mit dynamischen CD-Gain gibt, die Navigations- und Manipulationsaufgaben unterstützen.

INTERAKTIONSAUFGABE – In dieser Arbeit werden für die Interaktion in GHD-Umgebungen folgende **Anforderungen an die Interaktionsaufgabe** gestellt: Unterstützung von Navigations- und Manipulationstechniken.

Benutzer

Die visuelle Wahrnehmung und die menschliche Motorik bewirken einige Einschränkungen für die Gestaltung und Verwendung von Interaktionstechniken in GHD-Umgebungen (siehe Abschnitt 2.1). Wie schon erwähnt sollte die unterschiedliche Wahrnehmung im fovealen und peripheren Sehen bei der virtuellen Darstellung berücksichtigt werden. Weiterhin sollte die Ergonomie bei der Eingabemodalität beachtet werden, um Ermüdung und Frustration bei den Nutzern zu minimieren (vergleiche ergonomische Richtlinien nach Nielsen et al. [NSMG03]). Es gibt beispielsweise individuelle motorische Eigenschaften bei der koordinierten Fingerbewegung, d.h. ein Tastendruck auf dem Eingabegerät sollte möglichst mit Daumen und/oder Zeigefinger erfolgen. Durch ein konsistentes Mapping zwischen physischer Eingabe und visueller Ausgabe wird die Hand-Augen-Koordination bei

der Nutzerinteraktion unterstützt. Bei einer beidhändigen Eingabemodalität sollte zwischen Aktionen mit der dominanten und nicht-dominanten Hand unterschieden werden, ebenso zwischen symmetrischen und asymmetrischen Bewegungen. Bei Manipulationsaufgaben bietet eine beidhändige Interaktion sowohl manuelle als auch kognitive Vorteile für den Nutzer [LZB98]. Jedoch erschwert das natürliche Handzittern ein präzises Manipulieren bei der räumlichen Interaktion. Darüber hinaus gibt es Unterschiede bei der Interaktion zwischen den Nutzergruppen, z.B. bei der Händigkeit (Rechts- oder Linkshänder), dem Alter (Kinder, Erwachsene oder Senioren) und der Körpergröße (unterschiedliche Armspanne zwischen kleinen und großen Personen). Folglich wird der Nutzer einerseits durch seine physiologischen Eigenschaften und andererseits durch seine individuellen Präferenzen bei der Interaktion beeinflusst.

BENUTZER – Es sind folgende **Anforderungen des Benutzers** an die Interaktion in GHD-Umgebungen zu berücksichtigen: Nutzergruppe, Wahrnehmung, unterschiedliche Koordinationsmöglichkeiten der Finger (Daumen/Zeigefinger vs. andere Finger) und Hände (dominante Hand vs. nicht-dominante Hand, Handpräferenz), Unterstützung der Hand-Augen-Koordination, natürliches Zittern, Ermüdung und Frustration.

Als Beitrag zur Interaktion in GHD-Umgebungen behandelt die vorliegende Arbeit weniger untersuchte Aspekte in diesem Forschungsbereich, indem (1) bekannte Navigations- und Manipulationstechniken umgesetzt werden, die *distance-aware* und *manual-switching* Mappingmethoden unterstützen, und (2) die Präferenzen der Nutzer bei der Interaktion in GHD-Umgebungen untersucht werden (siehe Abbildung 2.4.3 *Interaction Scaling*).

Informationsvisualisierung in GHD-Umgebungen, bei denen der Nutzer präzise mit den Daten interagieren soll, erfolgt üblicherweise unter Laborbedingungen (d.h. optimale Hardwarekonfiguration, bekannte Nutzergruppe). Die Interaktion in GHD-Umgebungen bezieht sich somit auf Einzelbenutzer- und begrenzte Mehrbenutzer-Szenarien (1 Nutzer bzw. 2-3 Nutzer) und es werden keine Displayumgebungen berücksichtigt, die unbekannte und große Nutzergruppen einschließen (z.B. *public display environments*).

Die Verwendung von stationären Eingabegeräten ermöglichen eine präzise Nutzerinteraktion, aber sie behindern die Mobilität der Nutzer, weshalb sie in dieser Arbeit nicht berücksichtigt werden. Weiterhin werden Gesten und Sprache als Eingabemodalität außer Acht gelassen; sie unterstützen zwar die Mobilität aber benötigen gegebenenfalls mehr Lernaufwand für den Nutzer.

3. Konzept der distanzabhängigen Interaktion

Zunächst wird die Interaktion in großen hochauflösenden Displayumgebungen in ihre Komponenten aufgeschlüsselt werden, um anschließend die Einflussfaktoren für die distanzabhängige Interaktion abzuleiten.

Die Nutzerinteraktion in GHD-Umgebungen findet in mehreren Interaktionsräumen statt (siehe Abbildung 3.0.1). Der Nutzer bewegt sich vor dem großen Display vor und zurück, um dargestellte Informationen mit unterschiedlichem Detailgrad zu betrachten (physische Navigation). Für die physische Navigation bewegt sich der Nutzer in seinem *mobility space* in Bezug zum Display (*display wall frame*). Neben seinen Laufbewegungen verwendet der Nutzer meist parallel ein Eingabegerät zur Interaktion mit der Anwendung, z.B. einen Laserpointer um Informationen auf dem Display auszuwählen. Diese physischen Aktionen finden im *motor space* des Nutzers statt in Bezug zu seinem räumlichen Standpunkt (*device/body frame*). Die Bewegungen und Aktionen des Nutzers im realen Raum, d.h. die physische Navigation (Mobilität und Motorik), werden im *physical interaction space* ausgeführt.

Aus Sicht der Anwendung erfolgt die Interaktion mit Hilfe eines virtuellen Cursors. Die Bewegungen des Cursors werden im *display space* durchgeführt, z.B. das Verschieben des Sichtfensters auf die dargestellten Informationen. Folglich werden Bewegungen und Aktionen innerhalb der Anwendung im *virtual interaction space* durchgeführt.

Der Interaktionsraum (*interaction space*) umfasst den physischen Interaktionsraum in der physischen Umgebung (*physical interaction space*) und den virtuellen Interaktionsraum der virtuellen Umgebung (*virtual interaction space*). Um die Nutzerinteraktion von der realen Welt in die virtuelle Welt übertragen zu können werden Transitionen benötigt. Dies erfolgt mit Hilfe von Abbildungsfunktionen (*Mapping*), welche die realen Bewegungen und Aktionen auf virtuelle Bewegungen und Aktionen abbilden. Da der Nutzer bei der Interaktion durch seine individuellen physiologischen Eigenschaften und Präferenzen geprägt ist, wird der Benutzer in dieser Arbeit als eigenständige Komponente betrachtet (vgl. Abschnitt 2.5).

Für die Interaktion in GHD-Umgebungen leiten sich daraus folgende vier Komponenten ab: (1) die **physische Umgebung**, (2) die **virtuelle Umgebung**, (3) der **Benutzer** und (4) das **Mapping**.

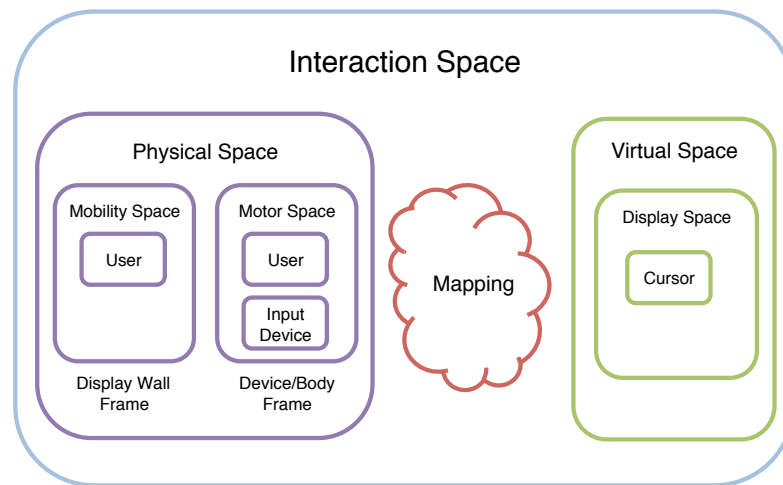


Abbildung 3.0.1.: Schematische Darstellung der verschiedenen Interaktionsräume in großen hochauflösenden Displayumgebungen.

3.1. Parameterraum

Bei der Interaktion mit Informationen in GHD-Umgebungen sollte die physische Navigation als natürliches Verhalten der Nutzer beim Betrachten von virtuellen Objekten berücksichtigt werden. Ein Beispiel: Befindet sich der Nutzer nah an dem hochauflösenden Display kann er nur einen begrenzten Ausschnitt der dargestellten Daten sehen, jedoch mit sehr hohem Detailgrad. Die Interaktion mit den detaillierten Daten sollte äquivalent erfolgen, d.h. der Nutzer kann sehr präzise mit den Daten im Sichtbereich interagieren. Im Gegensatz dazu erfolgt die Wahrnehmung der Datenrepräsentation aus der Entfernung für den Nutzer schnell und im allgemeineren Kontext. Daher sollte die Interaktion aus der Entfernung mit grober Präzision und schnell erfolgen. Neben der Größe der Anzeigefläche in GHD-Umgebungen, die unterschiedliche Darstellungsgenauigkeit durch den Betrachtungsabstand bieten, kann ebenfalls der Betrachtungsabstand für die Präzision der Interaktion genutzt werden.

Der Grad der Präzision der Nutzerinteraktion wird durch die folgenden Komponenten und deren Faktoren beeinflusst (vergleiche Abbildung 1.2.1):

Physische Umgebung Wo und womit findet die Interaktion in der physischen Umgebung statt? Die Komponente physische Umgebung umfasst die Einflussfaktoren aus der realen Welt. Dazu gehören beispielsweise die Dimensionen der Displaywand, welche Eingabemodalität zur Interaktion verwendet wird, wie der physische Interaktionsraum vor der Displaywand genutzt wird und damit verbunden, welche Sensortechnologie für die Erfassung der Nutzerinteraktion zur Verfügung steht.

Virtuelle Umgebung Wie wird in der virtuellen Umgebung interagiert? Die Komponente virtuelle Umgebung beinhaltet alle Einflussfaktoren in der virtuellen Welt oder der Anwendung mit der interagiert wird. Zur virtuellen Welt gehören die (interaktiven) virtuellen Inhalte, die verfügbaren Interaktionstechniken, die Interaktionsaufgaben und die Eigenschaften des virtuellen Interaktionsraumes (z.B. 2D, 3D, Verdeckung).

Benutzer Wer interagiert mit der virtuellen Umgebung innerhalb der physischen Umgebung? Der Benutzer als Komponente wird durch seine objektiven und subjektiven Eigenschaften in der Interaktion beeinflusst. Zu den objektiven Eigenschaften zählen die kognitiven, perzeptiven und motorischen Fähigkeiten des Nutzers (z.B. visuelle Wahrnehmung, physische Navigation, Koordinationsmöglichkeiten). Zu den subjektiven Eigenschaften zählen Präferenzen und Eigenheiten des Nutzers (z.B. Händigkeit, natürliche Zittern), Vorwissen und Erfahrungen (angelehnt an Thiede [Thi10]).

Mapping Welche Aktionen in der physikalischen Umgebung werden auf welche Aktionen in der virtuellen Umgebung abgebildet? Die Komponente Mapping bildet die Interaktion des Nutzers in der physischen Umgebung (Eingabe) auf die Interaktion in der virtuellen Umgebung (Ausgabe) ab. Die Mappingmethode ist abhängig davon, wie exakt die Eingabe auf die Ausgabe abgebildet werden soll, welche Einschränkungen in der physischen Umgebung oder beim Benutzer kompensiert werden sollen (z.B. Tremor), welche Präzision der Interaktion in der physischen Umgebung gegeben ist und welche Präzision in der virtuellen Umgebung unterstützt werden soll.

Die Einflussfaktoren der vier Komponenten stehen in enger Wechselbeziehung zueinander, wie beispielsweise die Displaygröße und die Darstellung des Informationsumfanges. Der Einfluss eines Parameters oder Subparameters einer Komponente auf Parameter einer anderen Komponente ist als Kreuz in der Tabelle 3.1.1 gekennzeichnet und ergibt sich aus der Literaturrecherche (vgl. Abschnitt 2.1, 2.3.2, 2.4.1). Die leeren Tabellenzellen wurden noch nicht untersucht oder sind keine sinnvollen Kombinationen.

Im Folgenden werden die Einflussfaktoren im mehrdimensionalen Parameterraum ausgehend von den gegebenen Faktoren diskutiert (vgl. Abbildung 3.1.5). Für die distanzabhängige Interaktion ist eine Gewichtung der Einflussfaktoren notwendig, die sich auf die geeignete Mappingmethode auswirkt. Dies wird an Beispielen im Abschnitt 3.3 beschrieben.

gegebenener Faktor		Physische Umgebung			Virtuelle Umgebung			Benutzer			Mapping				
		Einfluss auf Interaktionsraum	Eingabemodalität	Displayanordnung	Informationsdarstellung	Interaktionstechnik	Dimension d. Darstellung, Detailgrad	physische Navigation	Wahrnehmung	Nutzeranzahl	Isomorphie	Projektion	Kinematik	Cursorberechnung	Effekte (Drift, Clutching)
Physische Umgebung	Displaykonfiguration	x			x	x		x	x	x					
	- Displayfläche	x			x			x		x					
	- Displayauflösung	x			x										
	- Displayanordnung	x				x		x		x					
	- Bezels				x	x									
	Eingabemodalität	x			x			x			x		x	x	x
	Interaktionsraum							x				x		x	
Virtuelle Umgebung	Informationsdarstellung			x											
	- Inhalt			x			x								
	- Dimension d. Darstellung					x									
	- Detailgrad					x			x						
	Interaktionstask				x	x	x								
	Interaktionstechnik		x								x	x	x	x	
Benutzer	Objektive Eigenschaften		x		x	x									
	- Physische Navigation		x			x									x
	- Wahrnehmung	x			x			x							
	- Koordination				x	x				x					
	Subjektive Eigenschaften		x			x					x				
	Ergonomie		x			x		x			x				x

Tabelle 3.1.1.: Zusammenfassung der Einflussfaktoren einer Komponente auf Faktoren einer anderen Komponente im Parameterraum.

3.1.1. Physische Umgebung

Displaykonfiguration

Die Anforderungsanalyse hat ergeben, dass folgende Faktoren der Displaykonfiguration einen Einfluss auf die Interaktion in GHD-Umgebungen haben: Displayfläche, Displayauflösung, Displayanordnung und das Vorhandensein von Bezels.

Eine große zusammenhängende Displayfläche kann einerseits durch eine nahtlose Anzeigefläche ohne Bezels erzeugt werden und andererseits durch eine gekachelte Anzeigeflächen mit Bezels. Wenn **Bezels** vorhanden sind, muss dies bei der Darstellung der virtuellen Inhalte beachtet werden, um Verzerrung oder Verdeckung der virtuellen Objekte über und hinter den Bezels zu vermeiden. Beispielsweise kann virtueller Raum hinter den Bezels ergänzt werden. Die verdeckte Anzeigefläche wird daraufhin vom Nutzer mental ergänzt (ähnlich dem Schauen durch ein Gitterfenster). Andererseits können Bezels zur Taskseparation genutzt werden, das bei der Interaktionstechnik zu beachten ist.

Große **Displayflächen** fördern die Performanz und Produktivität von Nutzern, da sich die Displaygröße auf den Umfang der physischen Navigation des Nutzers auswirkt. Je größer die Ausdehnung der Anzeigefläche ist desto mehr muss der Nutzer sich bewegen (Mobilität), um die dargestellten Informationen in ihrer Gesamtheit und im Detailgrad wahrzunehmen.

Weiterhin spielt die **Displayanordnung** eine Rolle. Eine Anordnung von 24 Displays in der Ebene erzeugt beispielsweise eine breite Anzeigefläche an der mehrere Nutzer kollaborativ die Informationsvisualisierung diskutieren können (z.B. als 8x3 Displaywand). Die Breite der Anzeigefläche erfordert einen Mindestabstand von mehreren Metern vom Nutzer damit er die komplette Anzeigefläche wahrnehmen kann.

Wenn die GHD-Umgebung für ein spezielles Anwendungsgebiet installiert wird, kann die virtuelle Umgebung durchaus Einfluss auf die Displayanordnung und benötigte Displayfläche haben. Zum Beispiel kann bei der Darstellung von Automobilen in Originalgröße die Ausdehnung der Displayfläche in der Höhe begrenzt werden und die Anordnung der Displays kann eher in der Breite erfolgen. Die Neukonfiguration der Displayumgebung ist zeitaufwendig (z.B. Verkabelung, Bildschirmmontage, Softwarekonfiguration), weshalb die Displaykonfiguration lediglich bei der Installation oder in größeren Zeitabständen an den virtuellen Inhalt bzw. das Anwendungsgebiet angepasst wird.

Die verfügbare **Displayauflösung** beeinflusst wie hoch der Detailgrad der virtuellen Darstellung sein kann. Abhängig von der Displayauflösung gibt es einen maximalen Betrachtungsabstand, bei dem der Nutzer individuelle Pixel wahrnehmen kann (siehe Gleichung 2.1.1). Mit zunehmendem Abstand verschmelzen die Pixel für den Nutzer miteinander. Das heißt, abhängig von der Informationsdarstellung (z.B. Pixel Codierung oder virtuelle Objekte) und der Displayauflösung ist der geeignete Betrachtungsabstand in GHD-Umgebungen unterschiedlich.

Die hardware-spezifischen Eigenschaften der LCD-Displays oder Projektoren (Helligkeit, Sichtwinkel, Farbkontrast etc.) haben einen direkten Einfluss auf die physiologische Wahrnehmung der Nutzer. In Displaykonfigurationen werden vor allem LCD-Displays mit hohem Sichtwinkel (etwa 178 Grad) verwendet damit der Nutzer sehr nah an der Displaywand agieren kann. Bei Konfigurationen mit Projektoren muss darauf geachtet werden, dass der Nutzer die Projektion nicht verdeckt, indem z.B. Rückprojektionen verwendet werden.

Interaktionsraum

Die Ausdehnung der Displayfläche und der minimale und maximale Betrachtungsabstand des Nutzers definieren den physischen **Interaktionsraum** vor der Displaywand. Die Größe des Interaktionsraumes ist wiederum bedeutend für die Interaktionstechnik, mit welcher der Nutzer aus beliebigen Entfernungen interagieren kann. Die Dimensionen des Interaktionsraumes können durch die Anordnung der Displays verändert werden, z.B. verkleinert eine halbkreisförmige Dis-

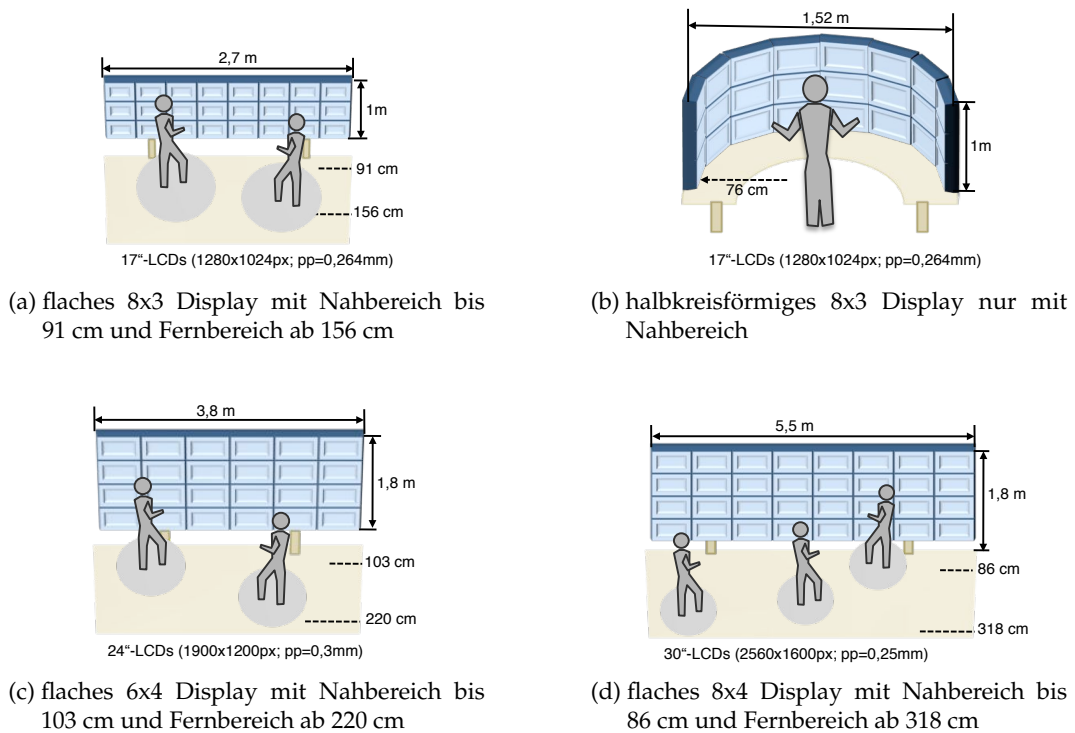


Abbildung 3.1.2.: Beispiele für Displaywände mit unterschiedlicher Displaygröße, -anordnung und -auflösung, den resultierenden Interaktionsräumen und Nutzeranzahl. Der Interaktionsraum ist in Nah- und Fernbereich unterteilt. Die Nutzeranzahl wird durch die Displaybreite und der persönlichen Distanz (graue Kreis) bestimmt.

playanordnung den Interaktionsraum. Unter Umständen kann nur ein Nutzer bei einer halbkreisförmigen Displaykonfiguration interagieren, dafür benötigt er jedoch weniger physischen Aufwand um Objekte zu selektieren (siehe Abbildung 3.1.2b).

In Abbildung 3.1.2 ist eine schematische Darstellung für unterschiedlicher Displaywände zu sehen, bei denen die Displayauflösung, Displayanordnung und die Displayanzahl variieren, und sich dementsprechend der physische Interaktionsraum und die Nutzeranzahl verändert. Die Beispiele wurden aus existierenden Arbeiten adaptiert (z.B. die 8x3 Displaywand von [SBY⁺06], die 6x4 Displaywand der Universität Rostock und die 8x4 Displaywand von [BI12]). Die Abbildung skizziert die Ausdehnung des Interaktionsraumes mit den Betrachtungsdistanzen für die Informationsdetails nahe am Display (Nahbereich nach Gleichung 3.1.1) und für den Informationsüberblick entfernt vom Display (Fernbereich nach Gleichung 3.1.2).

$$\text{Betrachtungsdistanz}_{\text{nah}} = \frac{\text{Pixelabstand pp}}{2 \tan\left(\frac{1}{120}\right)} \quad (3.1.1)$$

$$\text{Betrachtungsdistanz}_{\text{entfernt}} = \sqrt{\frac{(\text{Displaybreite})^2}{2 * (1 - \cos(120))}} \quad (3.1.2)$$

Aufgrund der Ausdehnung des Interaktionsraumes ist zu sehen, dass bei der halbkreisförmigen Anordnung nur ein Nutzer interagieren kann, während bei den anderen Konfigurationen zwei bis drei Nutzer agieren können, unter Berücksichtigung der persönlichen Distanz zwischen Personen (ca. 50 cm nach [Hal90]¹).

In einer Multi-Displayumgebung (z.B. Smart Meeting Room) bilden die Displays keine zusammenhängende Displayfläche. Die Displays sind meist heterogen, unterschiedlich zueinander orientiert und es befinden sich große Abstände zwischen den einzelnen Anzeigeflächen (z.B. Projektionsflächen an den Wänden und Laptops auf dem Konferenztisch). Hier eignen sich eher Aufgaben mit Taskseparation, z.B. die Diskussion über eine 2D Diagrammdarstellung anstatt die Navigation durch eine 3D Welt. Typischerweise werden Mehrbenutzer-Szenarien in Smart Meeting Rooms verwendet in denen die Nutzer oft feste Positionen einnehmen. Dies hat Einfluss auf die Ergonomie und physische Navigation der Nutzer, d.h. es finden eher Kopf- und Armbewegungen statt (viel Motorik), als dass die Personen herumlaufen (wenig Mobilität).

Eingabemodalität

Die Analyse hat gezeigt, dass folgende Anforderungen an die **Eingabemodalität** bestehen: mobil, schnell verfügbar, leicht anzuwenden, aus unterschiedlichen Entfernungen nutzbar und Unterstützung der physischen Navigation. Somit sind Eingabegeräte verwendbar, die im realen dreidimensionalen Raum genutzt werden können (ohne die Notwendigkeit einer 2D Ebene), wobei die Geräte Daten für den 2D Raum oder den 3D Raum erzeugen (Freiheitsgrade). Die Eingabegeräte werden oft in Kombination mit einem Trackingsystem verwendet, um die physische Navigation zu unterstützen.

Wir beschränken uns im Parameterraum auf optische Trackingsysteme, da diese eine hohe Genauigkeit (im mm-Bereich) ermöglichen, einen flexible Trackingbereich unterstützen und die benötigten Marker auf beliebigen Oberflächen bzw. Geräten anzubringen sind. Sie ermöglichen eine kontinuierliche Positions- und Orientierungsbestimmung von Objekten (bis zu 6-DOF). Die Anzahl der verfügbaren **Freiheitsgrade** hat Einfluss auf die Interaktionstechnik und die physische Navigation des Nutzers. Das **Trackingvolumen** begrenzt den physischen Interaktionsraum für die Nutzereingabe. Daher sollte der Interaktionsraum sowohl auf die Displaykonfiguration als auch auf die Eingabemodalität angepasst werden.

Bei der Eingabemodalität ist ein wichtiger Subparameter die **Auflösung der Eingabe**, d.h. mit welcher Genauigkeit werden Objekte im Interaktionsraum getrackt beziehungsweise mit welcher Genauigkeit wird eine Positions- oder Rate-Veränderung am Eingabegerät registriert (z.B. ein Tastendruck oder die Rotations-

¹ Basierend auf den unterschiedlichen physischen Distanzen innerhalb derer Personen miteinander interagieren, definiert Edward T. Hall [Hal90] die vier sozialen Zonen: (1) die vertrauliche Distanz (bis 18 in., 46 cm), (2) die persönliche Distanz (1,5–4 ft., 46–122 cm), (3) die soziale Distanz (4–12 ft., 122–366 cm) und (4) die öffentliche Distanz (ab 12 ft., 366 cm).

auslenkung). Die Auflösung hat Auswirkungen darauf, wie stark ein Gerät vom Nutzer zum Interagieren bewegt werden muss (Motorik). Bewegt der Nutzer zum Beispiel einen Nintendo Wii Remote Controller (Wiimote), der im Interaktionsraum getrackt wird, dann werden schon minimale Bewegungen vom Trackingsystem aufgrund der hohen Auflösung erkannt. Die internen Beschleunigungssensoren der Wiimote erzeugen hingegen erst eine Änderung, wenn der Nutzer eine schnelle Positionsänderung im Handgelenk durchführt.

Eine hohe Auflösung der Sensortechnologie kann von Vorteil sein, da sie kleinste Veränderungen registriert (z.B. genaue Kopfposition), aber auch zum Nachteil da ungewollte Veränderungen (z.B. das natürliche Handzittern) erkannt werden. Um die gewollten und ungewollten Positions- und Rate-Veränderungen der Eingabemodalität zu verstärken bzw. zu vermindern werden Filtertechniken und Mappingmethoden eingesetzt. Zusätzlich ist abzuwägen welche Nutzereingabe (**Eingabegerät**) auf welchen Subparameter der Interaktionstechnik abgebildet werden soll, und mit welcher Eingabeauflösung dies erfolgt.

FAZIT: Die **Displaykonfiguration** hat Einfluss auf: (1) den Interaktionsraum, (2) die Parameter der virtuellen Umgebung (Informationsrepräsentation, Interaktionstechnik) und (3) die Parameter der Komponente Benutzer (Mobilität, Motorik, Wahrnehmung, Nutzeranzahl). Die **Eingabemodalität** (Eingabegerät, Trackingvolumen, Auflösung, Freiheitsgrade) hat Einfluss auf (1) den Interaktionsraum in der physischen Umgebung und (2) die Interaktionstechnik in der virtuellen Umgebung. Der **Interaktionsraum** beeinflusst (1) die physische Navigation des Benutzers (Motorik und Mobilität) und (2) das benötigte Mapping.

3.1.2. Virtuelle Umgebung

Informationsdarstellung

In der Informationsvisualisierung sind **virtuelle Inhalte** beispielsweise die Darstellung von Temperaturskalen (1D Darstellung) und Landschaftskarten (2D Darstellung), die Visualisierung von Graphen (hierarchische 2D Darstellung bzw. 2,5D Darstellung) oder die Visualisierung von komplexen biologischen Strukturen (3D Darstellung). Informationen können in unterschiedlichen mehrdimensionalen Räumen dargestellt werden, wobei die Dimension der Information oftmals die **Dimension der Darstellung** bestimmt.

Der darstellbare **Informationsumfang** hängt von der vorhandenen Displayfläche ab. Ist eine große Displayfläche verfügbar, können viele Informationen dargestellt werden. Wenn die Displayfläche nicht ausreichend groß ist, kann nur ein Teil der Informationen dargestellt werden oder die Informationen müssen in einem höheren Abstraktionsgrad zusammengefasst werden (**Detailgrad**). Damit Informationen wie Textlabels für den Nutzer lesbar sind, sollte die Schriftgröße des Textes

an den Leseabstand des Nutzers angepasst werden. Außerdem sollte bei einer Displaykonfiguration mit Bezels die Textdarstellung nicht über die Bezels erfolgen, um den Lesefluss zu erhalten.

Bei Multi-Displayumgebungen sollten zusammenhängende Informationen auch örtlich beieinander angezeigt werden (z.B. auf einem Display). Vom Nutzer nicht wahrgenommene Displaybereiche erfordern keine visuelle Darstellung; somit kann die verfügbare Rechenleistung für das Rendern der sichtbaren Displaybereiche genutzt werden.

Um Informationen zu untersuchen und zu verstehen, muss der Nutzer mit ihnen interagieren können. Wie und in welchem Detailgrad der Nutzer interagieren kann, wird vom Interaktionstask bestimmt. Kann der Nutzer die Ansicht auf die Informationsvisualisierung verändern, z.B. ein 3D Terrain von unterschiedlichen Seiten betrachten oder Standpunkte innerhalb des Terrains auswählen, dann müssen Navigationsaufgaben unterstützt werden. Sollen virtuelle Objekte durch den Nutzer neu platziert oder sortiert werden, müssen Manipulationsaufgaben durchführbar sein. Wie in der Anforderungsanalyse diskutiert, beschränkt sich diese Arbeit auf Navigations- und Manipulationsaufgaben (siehe Abschnitt 2.5).

Interaktionstask und Interaktionstechnik

Die Interaktion erfolgt in virtuellen Umgebungen mit Hilfe mehrerer oder komplexer **Interaktionstasks** (siehe Abschnitt 2.4.1.3). Beispielsweise wird die Selektion als initiale Nutzereingabe für weiterführende Tasks verwendet. Die Kombination der Interaktionstasks wird durch die **Interaktionstechniken** umgesetzt.

Um die Objektselektion benutzerfreundlich zu gestalten, verwenden die Interaktionstechniken oft ein visuelles Highlighting, um die Selektion zu aktivieren, oder der Nutzer initiiert die Selektion durch einen Tastendruck am Eingabegerät. Aufgrund der Verwendung von Buttons für Modusänderungen kann der Nutzer explizit die Interaktion starten und beenden, das die Erkennung der Nutzerinteraktion in GHD-Umgebungen vereinfacht. Darüber hinaus muss für den Nutzer erkennbar sein wie die Objektmanipulation funktioniert, damit er die Interaktionstechnik versteht und nutzen kann (z.B. Animation der Positionsänderungen). Aus diesem Grund sollten die Interaktionstechniken für den Nutzer einfach zu lernen, sowie leicht und natürlich zu bedienen sein (vgl. Abschnitt 2.2.2).

Die Zerlegung der Navigations- und Manipulationsaufgaben in ihre Interaktionstasks und deren Task-Parameter ist in Tabelle 2.4.3 dargestellt. Je nach Zielsetzung der Nutzerinteraktion lassen sich somit aus den Task-Parametern die Parameter für die Interaktionstechnik ableiten. Beispielsweise können Körperbewegungen (Nutzereingabe) auf die Geschwindigkeitskontrolle (Task-Parameter) für Navigationsbewegungen (Interaktionstask) abgebildet werden. Solange der Nutzer seinen Oberkörper neigt, wird eine Navigationsbewegung mit konstanter Geschwindigkeit durchgeführt; die Geschwindigkeit wird beschleunigt je länger der Oberkörper vorwärts geneigt ist bzw. wieder verlangsamt, wenn der Nutzer sich zurücklehnt.

Der Bewegungsablauf bestimmt die Eingabebedingungen für das Starten, Fortführen und Beenden der Navigation.

Bekannte Interaktionstechniken für die Interaktionsaufgaben in GHD-Umgebungen sind: (1) Ray-Casting Techniken für die Cursorsteuerung zur Selektion und Manipulation von Objekten, (2) Head-Tracking für die Navigationsbewegungen und (3) Eingabegeräte mit Buttons für Modus Änderungen.

Wird Head-Tracking für die Navigationstechnik verwendet, kann die virtuelle Szene nach rechts/links oder oben/unten entsprechend der Änderungen der Kopfposition (Kopfnähe) verschoben werden. Die Dauer der Kopfnähe kann für die Verschiebung verwendet werden, d.h. solange der Kopf nach rechts geneigt ist, erfolgt die Navigationsbewegung nach rechts und endet, sobald der Kopf wieder gerade ausgerichtet ist (Ausgangsposition). Der Nutzer dreht natürlicherweise seinen Kopf beim Umherschauen. Diese Orientierungsänderungen (Kopfdrehung) kann ebenfalls für die Bildverschiebung beim Head-Tracking verwendet werden. Jedoch verursachen die Augenbewegungen des Nutzers beim Interagieren vor einer Displaywand minimale Kopfbewegungen, die zu ungewollten Änderungen in der virtuellen Umgebung führen können. Folglich ist Head-Tracking eher für einfache Interaktionsmechanismen als für feinmotorische Steuerungen geeignet. Wird als Eingabemodalität die Hand oder ein Eingabegerät in der Hand verwendet, kann die Orientierung der Eingabe für die Positionsänderung verwendet werden, da die visuelle Wahrnehmung und die Nutzereingabe entkoppelt und somit gleichzeitig ausführbar sind.

Folglich sollte bei der Interaktionstechnik beachtet werden, welche Freiheitsgrade der Nutzereingabe auf welchen Task-Parameter abgebildet werden und mit welcher Eingabeauflösung dies erfolgt.

FAZIT: Die **Informationsdarstellung** (virtueller Inhalt, Informationsumfang, Detailgrad) sollte an die physiologische Wahrnehmung des Nutzers und an die Displaykonfiguration angepasst werden. Der **Interaktionstask** (Selektion, Positionierung, Rotation, Skalierung, Bewegung) bestimmt (1) mit welchen virtuellen Inhalten und (2) in welchem Detailgrad der Nutzer interagieren kann. Die **Interaktionstechnik** ist abhängig vom Interaktionstask und der Eingabemodalität und beeinflusst die Nutzerergonomie. Die Freiheitsgrade der Eingabemodalität werden auf die Parameter der Interaktionstasks abgebildet. Dabei ist zu beachten welche Nutzereingabe auf welchen Task-Parameter abgebildet wird, damit der Nutzer schnell sein Ziel erreicht, ohne überanstrengt oder frustriert zu sein.

3.1.3. Benutzer

Objektive Eigenschaften

Die physiologische **Wahrnehmung** des Nutzers hat Einfluss auf den Umfang der Informationsdarstellung. Auch wenn eine große Displayfläche verfügbar ist um sehr viele Informationen darzustellen, kann die Informationsflut nur bedingt vom Nutzer wahrgenommen und verarbeitet werden (z.B. zu kleine Schriftgröße und zu viele Informationen). Hier eignet sich die Darstellung der Informationen in einer höheren Abstraktionsebene und der Nutzer ändert interaktiv den Detailgrad der Darstellung, z.B. durch das Herantreten an die Displaywand (vgl. Prinzip der Informationssuche [Shn96]). Wenn mehrere Nutzer kollaborativ vor der Displaywand agieren, müssen die individuellen Betrachtungsdistanzen beachtet werden. Das heißt, die Informationsdarstellung sollte für jeden Nutzer angepasst werden. Zum Beispiel kann der allgemeine Informationskontext auf der gesamten Displayfläche visualisiert werden (z.B. eine 2D Kartendarstellung), während distanzabhängig für jeden Nutzer individuelle Fokusbereiche mit detaillierten Informationen dargestellt werden (z.B. Beschriftungen innerhalb der Karte). Überlappen sich die Fokusbereiche zwischen den Nutzern, müssen Kompromisse bei der individuellen Informationsdarstellung gefunden werden (vgl. Abschnitt 3.3.1).

Die **Mobilität** und die **Motorik** des Nutzers (**physische Navigation**) haben einen wesentlichen Einfluss auf die nutzbaren Eingabemodalitäten in der physischen Umgebung und auf die geeignete Interaktionstechnik in der virtuellen Umgebung. Damit die Nutzerinteraktion durchführbar ist, müssen die motorischen Beschränkungen bei der Eingabemodalitäten berücksichtigt werden (vgl. Abschnitt 2.1.2). Zum Beispiel ermöglicht die Beweglichkeit des Handgelenkes eine sehr begrenzte Innenrotation, aber eine große Außenrotation. Das heißt, die linke Hand kann (mit Handrücken nach oben) nur um wenige Grad im Uhrzeigersinn gedreht werden, wohin gegen eine Drehung gegen den Uhrzeigersinn um fast 180 Grad möglich ist. Armbewegungen ermöglichen Drehungen von fast 360 Grad im Raum. Die großräumigen Bewegungen sind für den Nutzer jedoch körperlich anstrengender und führen schneller zu Ermüdung.

Die **Koordinationsmöglichkeiten der Finger** beschränken die Nutzung von Buttons auf den Eingabegeräten auf Daumen und Zeigefinger, da diese gezielt und getrennt voneinander bewegt werden können. Bei der **Koordination der Hände** muss zwischen dominanter und nicht-dominanter Hand unterschieden werden. Bei einhändiger Interaktion nimmt der Nutzer meist seine dominante Hand zum Interagieren. Bei der beidhändigen Interaktion wird meist die dominante Hand für genau (feinfühlig) Bewegungen benutzt, während die nicht-dominante Hand für gröbere Bewegungen und als Referenzframe verwendet wird.

Um die **Hand-Augen-Koordination** zu unterstützen sollten die Bewegungsrichtungen der realen Nutzereingabe und der virtuellen Objektmanipulation übereinstimmen (z.B. die Rotation im Uhrzeigersinn oder die Verschiebung nach rechts).

Sonst benötigt der Nutzer einen höheren mentalen Aufwand, um die Nutzerinteraktion zu lernen. Ein weiterer Punkt ist die Eingabeart, bei welcher der motorische und mentale Aufwand für die Nutzerinteraktion abgewogen werden muss. Indirekte Eingabegeräte eignen sich besonders für großen Displayoberflächen und können auch aus der Entfernung mit dem Display interagieren (vgl. Abschnitt 2.4.1.1). Direkte Selektionstechniken hingegen minimieren den mentalen Aufwand für die Hand-Augen-Koordination, aber ihr motorischer Aufwand ist größer. Bei indirekten Selektionstechniken ist der motorische Benutzungsaufwand geringer, jedoch wird ein erhöhter mentaler Aufwand zum Lernen der Hand-Augen-Koordination benötigt. Daher eignen sich hybride Selektionstechniken, die direkte und indirekte Repräsentationen mischen (vgl. Abschnitt 2.4.2).

Die Freiheitsgrade der Nutzereingabe werden auf die Task-Parameter der Interaktionstechnik abgebildet. Bei der Manipulationstechnik (z.B. Ray-Casting) werden zum Beispiel die Positionsänderungen des Eingabegerätes (3-DOF) zur Positionierung des virtuellen Objektes verwendet, während die Orientierungsänderungen des Gerätes (6-DOF) für die Drehung des virtuellen Objektes verwendet wird. Diese Art der Abbildung unterstützt die natürliche Bedienbarkeit der Interaktionstechnik, da sie der Objektmanipulation in der realen Welt nachempfunden ist.

Subjektive Eigenschaften

Die subjektiven Eigenschaften der Nutzer (z.B. Nutzergruppe, Körpergröße, Tremor) haben Einfluss auf die Interaktionstechnik. Wird beispielsweise eine Ray-Casting Technik zur Cursorsteuerung in GHD-Umgebungen verwendet, kann der Nutzer von einer beliebigen Betrachtungsdistanz jede Position auf der Displaywand erreichen. Jedoch ist das natürliche **Handzittern** im virtuellen Interaktionsstrahl sichtbar. Werden nur die Positionsdaten für die Berechnung des virtuellen Interaktionsstrahls genommen (orthogonale Projektion), ist der Tremor nicht sichtbar. Diese Variante erfordert aber mehr physische Navigation (Mobilität) vom Nutzer.

Allerdings kann der Nutzer aufgrund seiner **Körpergröße** (Armreichweite) gegebenenfalls nicht mehr alle Bereiche der Displayfläche erreichen. Zum Beispiel² kann ein 1,84 m großer Erwachsener virtuelle Objekte bis zu einer Höhe von ca. 2,15 m auf der Displaywand erreichen, während für einen mittelgroßen Erwachsenen mit einer Körpergröße von 1,65 m nur Displaybereiche bis ca. 1,95 m Höhe erreichbar sind und für ein Kind mit einer Körpergröße von 1,33 m sogar nur virtuelle Objekte in Höhe bis zu 1,53 m. Die Körpergröße wird als subjektive Eigenschaft anstatt als individuelle physische bzw. objektive Eigenschaft eingeordnet. Denn neben der Körpergröße eines Nutzers haben auch die Körperproportionen einen Einfluss auf die Interaktion (z.B. kurze/lange Arme). Das heißt, es muss ein Kompromiss zwischen der Nutzergruppe (z.B. Körpergröße), der Nutzereingabe (Freiheitsgrade und Mobilität) und der Ergonomie (störende Handzittern vs. mehr physische Navigation) für die Interaktionstechnik gefunden werden.

² Für die Beispiele wurden Personen mit den jeweiligen Körpergrößen vermessen (vgl. Tabelle A.1.1).

Nutzer können komplexe Interaktionsaufgaben, die symmetrische oder asymmetrische Bewegungsabläufe erfordern, sehr gut beidhändig ausführen. Hier spielt oftmals **das Vorwissen und die Erfahrung** des Nutzers eine Rolle. Ist er es gewöhnt, dass die linke und rechte Hand unterschiedliche Bewegungsabläufe ausführen, dann ist für ihn das Anwenden von asymmetrischen Bewegungen einfach. Ist er hingegen weniger mit solchen Bewegungen vertraut, dann eignen sich eher symmetrische Abläufe. Zum Beispiel kann die beidhändige uniforme Skalierung von virtuellen Objekten als asymmetrische oder symmetrische Bewegung durchgeführt werden. Bei der asymmetrischen Variante wird die nicht-dominante Hand als Referenzframe verwendet, d.h. sie hält das virtuelle Objekt fest und der Skalierungsfaktor wird durch die dominante Hand bestimmt (vgl. Abbildung 3.3.13b). Bei dem symmetrischen Bewegungsablauf bestimmen beide Hände den Skalierungsfaktor, indem der Abstand zwischen den Händen gleichmäßig vergrößert oder verkleinert wird (vgl. Abbildung 3.3.13a).

Die **kognitiven und perzeptiven Fähigkeiten** des Nutzers (Wahrnehmung, Hand-Augen-Koordination, Hand/Finger-Koordination), sowie die **ergonomische Nutzereingabe** beeinflussen die Interaktionstechniken. Damit die Interaktionstechnik in GHD-Umgebungen effektiv benutzbar ist, muss sie einen geringen Lernaufwand und eine einfache und natürliche Bedienbarkeit unterstützen (vgl. Abschnitt 2.2.2). Zum Beispiel eignen sich Eingabegeräte in den Händen der Nutzer zur Unterstützung der Sensormotorik bei der Interaktion (haptisches Feedback). Die Eingabegeräte sollten dabei ergonomisch zu bedienen sein und den Tragekomfort nicht beeinträchtigen.

Folglich beeinflussen mehrere Parameter und Subparameter der Komponente Benutzer die Interaktionstechnik. Alleine bei der Rotation virtueller Objekte (Interaktionstask) durch Rotationsbewegungen mit einer Hand (Eingabemodalität) sind vier Einflussfaktoren zu berücksichtigen: (1) die Motorik des Handgelenkes beschränkt die durchführbaren physischen Rotationsbewegungen; (2) die Händigkeit verursacht unterschiedliche durchführbare Rotationswinkel für Links- und Rechtshänder; (3) die Hand-Augen-Koordination erfordert eine virtuelle Rotationsrichtung im/gegen den Uhrzeigersinn; (4) die Ergonomie erfordert die Vermeidung von Überdehnungen oder Extrempositionen des Handgelenkes, sowie die Unterstützung entspannter und neutraler Handgelenkpositionen.

FAZIT: Die **objektiven Eigenschaften** des Nutzers (Motorik, Wahrnehmung, Koordinationsmöglichkeiten) beeinflussen (1) die Eingabemodalität in der physischen Umgebung, sowie (2) die Interaktionstechnik und (3) die Informationsdarstellung (Informationsumfang, Detailgrad) in der virtuellen Umgebung. Die **subjektiven Eigenschaften** (Körpergröße, Nutzergruppe, Händigkeit, Erfahrung) des Nutzers beeinflussen ebenfalls (1) die Eingabemodalität und (2) die Interaktionstechnik.

3.1.4. Mapping

Die Mappingmethode bildet die Eingabemodalität der Nutzerinteraktion in der physischen Umgebung auf die gewünschten Subparameter der Interaktionsaufgabe in der virtuellen Umgebung ab. Für die Mappingmethode wird ein Mappingfaktor (z.B. der Koeffizient CD-Gain) verwendet, der die Bewegungen des Eingabegerätes auf Bewegungen des Cursors in der virtuellen Umgebung abbildet und dabei verstärkt oder verringert. Der Mappingfaktor (MF) kann beispielsweise aus dem Verhältnis der Bewegungsstanz des Cursors zur Bewegungsstanz des Eingabegerätes berechnet werden (siehe Gleichung 3.1.3).

$$MF = \frac{\|\vec{t}_{cursor}\|}{\|\vec{t}_{eingabe}\|} \quad (3.1.3)$$

Isomorphie

Der Nutzer soll beispielsweise ein virtuelles Objekt um 360 Grad durch Rotationsbewegungen seines Handgelenkes drehen. Soll die Hand-Augen-Koordination unterstützt werden, dann ist Realismus wünschenswert, d.h. ein isomorpher Ansatz. Beim **isomorphen Ansatz** werden die physischen Bewegungen eins-zu-eins auf die virtuellen Bewegungen abgebildet (MF=1). In diesem Fall ist die Objektdrehung physiologisch bedingt nur um einige Grad möglich und der Nutzer muss mehrere Rotationsschritte durchführen, um das Objekt vollständig zu drehen (*Clutching*-Effekt). Soll die Interaktion hingegen mit weniger physischen Aufwand für den Nutzer durchführbar sein, dann wird ein nicht-isomorpher Ansatz gewählt, der keinen strikten Realismus verwendet (MF≠1). Beim **nicht-isomorphen Ansatz** kann die Objektmanipulation anders als in der realen Welt erfolgen und die physischen Beschränkungen des Menschen umgangen werden, jedoch muss der Nutzer die Handlungsbewegungen lernen (vgl. Abschnitt 2.4.1.4). Für die Objektmanipulation würde dies bedeuten, dass der Nutzer durch eine 120 Grad Drehung seines Handgelenkes das virtuelle Objekt um 360 Grad rotieren kann, indem die Manipulation um den Mappingfaktor drei verstärkt wird.

Cursorberechnung

In GHD-Umgebungen werden meist indirekte Selektionstechniken (z.B. Ray-Casting Techniken) eingesetzt, mit denen der Nutzer aus unterschiedlichen Entfernungen zum Display interagieren kann. Um die Hand-Augen-Koordination des Nutzers zwischen seiner physischen Eingabe im realen Raum und der visuellen Rückmeldung auf dem Display zu erleichtern werden visuelle Repräsentationen wie Cursor eingesetzt. Sie repräsentieren die physische Eingabe in der virtuellen Umgebung und die verwendete Mappingmethode berechnet abhängig von der Eingabemodalität, d.h. der **Veränderung der Eingabe**, die korrespondierenden Cursorbewegungen. Die **Cursorberechnung** kann durch relative, absolute oder hybride Cursorbewegungen erfolgen.

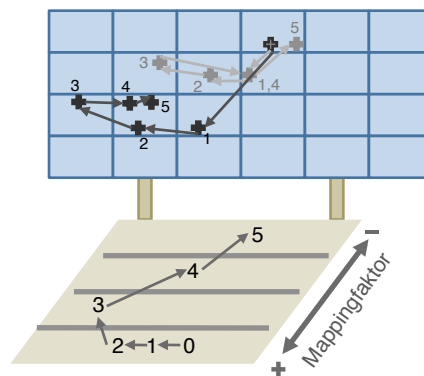


Abbildung 3.1.3.: Drift-Effekt – wahrnehmbare Abweichung zwischen physischer Eingabe (hellgraue Kreuze) und den entsprechenden relativen Cursorbewegungen (schwarze Kreuze) auf der Displaywand bei variierendem Mappingfaktor; die Zahlen auf dem Boden bilden die Trajektorie der physischen Navigation.

Bei der **absoluten Cursorberechnung** werden die realen Positionsdaten auf die virtuelle Sichtebene projiziert und durch absolute Cursorbewegungen dargestellt. Dabei werden die absoluten Veränderungen des Eingabegerätes, z.B. Position und Orientierung verwendet. Dies führt dazu, dass bei einer hohen Eingabeauflösung kleinste Positionsänderungen der Hand für den Nutzer sichtbar sind (z.B. Handzittern). Darüber hinaus ist es für den Nutzer schwierig, aufgrund der fehlenden Präzision pixelgenau auf dem Display zu selektieren. Jedoch ermöglicht die absolute Cursorberechnung eine komfortable Cursorsteuerung in GHD-Umgebungen für den Nutzer, da der Cursor sich dahin bewegt wohin der Nutzer zeigt.

Bei der **relativen Cursorberechnung** werden die relativen Positionsänderungen des Eingabegerätes auf die neue Cursorposition abgebildet. Die relativen Cursorbewegungen werden eingesetzt, wenn einerseits eine hohe Präzision der Interaktion gewünscht wird, z.B. indem große physische Eingabebewegungen in kleine virtuelle Cursorbewegungen resultieren (d.h. MF verringert), oder andererseits, wenn große Cursorbewegungen mit geringen physischen Eingabebewegungen erfolgen soll (d.h. MF verstärkt), die aber nur eine grobe Präzision erlauben. Bei absoluten Eingabegeräten kann, aufgrund der relativen Cursorberechnung die Hand-Augen-Koordination gestört werden. Je nachdem wie stark der Mappingfaktor verstärkt oder verringert wird, entsteht entweder sehr schnell oder erst nach einiger Zeit eine für den Nutzer sichtbarer Abweichung zwischen seiner physischen Eingabe und der visuellen Repräsentation (Drift-Effekt). Die Abbildung 3.1.3 veranschaulicht den **Drift-Effekt** bei variierendem Mappingfaktor, d.h. nah am Display wird der Mappingfaktor verringert und in der Entfernung verstärkt. Die hellgrauen Kreuze zeigen die projizierten absoluten Positionsdaten und die schwarzen Kreuze die für den Nutzer sichtbare Cursorposition. Der Nutzer sollte daher die Möglichkeit haben mit Hilfe der Interaktionstechnik den Drift zu korrigieren, z.B. indem die

Ausgangsposition für die relative Cursorberechnung auf die aktuelle absolute Cursorposition gesetzt wird.

In GHD-Umgebungen sollen sowohl schnelle und große Bewegungen als auch kleine und präzise Bewegungen ausgeführt werden, weshalb ein hybrider Ansatz der Cursorberechnung von Vorteil ist. Die Literaturrecherche hat gezeigt, dass für unpräzise Aktionen absolute Positionsdaten und für präzise Operationen relative Positionsdaten verwendet werden. Das heißt, bei der **hybriden Cursorberechnung** wird für Manipulationsoperationen, die eine geringe Präzision benötigen die absolute Cursorberechnung genutzt und bei Manipulationsoperationen mit einer hohen Präzision wird die relative Cursorberechnung verwendet. Der hybride Ansatz reduziert den Drift-Effekt, aufgrund des Wechsels zwischen absoluten und relativen Mapping bei der Cursorberechnung.

Projektion

Die Verwendung von Ray-Casting Techniken als Interaktionstechnik erfordert die Projektion der Positionsdaten im physischen Raum auf die Displayfläche, um die Cursorposition im virtuellen Raum zu bestimmen. In dieser Arbeit werden vorrangig GHD-Umgebungen betrachtet, in denen der physische Interaktionsraum in Breite und Höhe die gesamte Displayfläche und somit die 2D Bildebene der virtuellen Umgebung umfasst. Folglich ist keine Skalierung der Ausdehnung des Interaktionsraumes auf die Ausdehnung der virtuellen Umgebung notwendig. Daher sind folgende zwei Arten der Projektion anwendbar: die perspektivische Projektion und die orthogonale Projektion (Definitionen angelehnt an [KCHI07]). Die **perspektivische Projektion** ist die "Standard"-Projektion, dabei wird ausgehend von den Positionsdaten in der physischen Umgebung (6-DOF) unter Berücksichtigung der Orientierung ein imaginärer Strahl berechnet und wo dieser die Displayoberfläche schneidet, werden die zugehörigen virtuellen Positionsdaten projiziert (siehe Abbildung 3.1.4a). Die perspektivische Projektion ist für den Nutzer leicht verständlich, da der virtuelle Cursor da sichtbar ist wo er vom Nutzer erwartet wird. Weiterhin sind alle Bereiche der großen Displayfläche für den Nutzer allein durch die Änderung der Orientierung der Positionsdaten erreichbar. Jedoch wird das natürliche Handzittern auf die virtuelle Cursorposition projiziert, dass durch die Mappingfunktion wiederum reduziert werden muss.

Bei der **orthogonalen Projektion** handelt es sich um einen Spezialfall der Parallelprojektion, bei der die Positionsdaten im physischen Raum (3-DOF) im rechten Winkel auf die Displayfläche projiziert werden, ohne Berücksichtigung der Orientierung (siehe Abbildung 3.1.4b). Bei der orthogonalen Projektion wird nur die 3D Position im realen Raum verwendet, weshalb das natürliche Handzittern für den Nutzer nicht sichtbar ist. Im Gegensatz dazu benötigt der Nutzer mehr physische Navigation um Objekte auf der Displayfläche zu selektieren als bei der perspektivischen Projektion und unter Umständen sind nicht alle Bereiche der Displayfläche

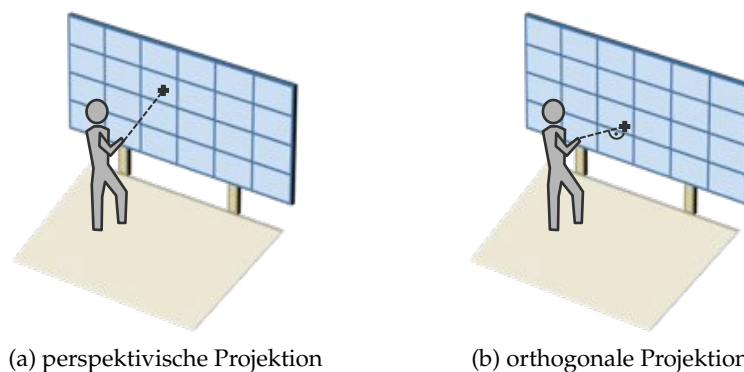


Abbildung 3.1.4.: Illustration der Ray-Casting Technik mit perspektivischer und orthogonaler Projektion.

für den Nutzer erreichbar (vgl. Abschnitt 3.1.3). Da die Orientierung nicht benötigt wird, ist ein Trackingansatz mit 3-DOF (bzw. 2,5-DOF) ausreichend.

Kinematik

Ein weiterer Parameter für die Mappingmethode ist die **Kinematik**. Sie gibt an, ob die Nutzereingabe auf Position (*position control*), Geschwindigkeit (*rate control*) oder Beschleunigung (*acceleration control*) im virtuellen Raum abgebildet wird. In GHD-Umgebungen wird die Nutzereingabe üblicherweise auf die Position oder die Geschwindigkeit der Subparameter der Interaktionstasks gemappt.

Mappingmethode

Die **Berechnung** des Mappingfaktors kann einerseits global erfolgen, d.h. ein konstanter vorberechneter Mappingfaktor wird verwendet. Andererseits kann sie lokal abhängig von der Nutzereingabe erfolgen, d.h. ein variabler Mappingfaktor innerhalb vordefinierter Grenzen wird verwendet. Wird eine Mappingfunktion mit konstanten vordefinierten Mappingfaktoren verwendet, sind die Parameter Projektion, Cursorberechnung, isomorpher Ansatz global für jeden Nutzer und die verwendete Eingabemodalität festgelegt. Bei einem variablen Mappingfaktor wird ein nicht-isomorpher Ansatz verwendet und zusätzlich die Kinematik berücksichtigt. In diesem Fall werden für die Berechnung meist lineare oder nicht-lineare Funktionen verwendet, die beispielsweise den Mappingfaktor erhöhen, wenn sich die Geschwindigkeit des Eingabegerätes erhöht.

Die Parameter für die Berechnung des Mappingfaktors ermöglichen es die Eingabegeräte zu kategorisieren (z.B. absolute oder relative Eingabeänderung) und somit Mappingmethoden zu verwenden, die relativ unabhängig von der Eingabemodalität sind. Weitere Anforderungen für das Mapping sind die Berücksichtigung der gegenwärtigen Interaktionsdistanz und ein dynamischer Mappingfaktor.

Die Herausforderung liegt darin, wie der **Wechsel** zwischen den verschiedenen Mappingfunktionen erfolgen soll, so dass einerseits eine feinfühligere Nutzereingabe

be (hohe Präzision der Interaktion) und andererseits eine schnelle Nutzereingabe (geringe Präzision) ermöglicht wird. Die Literaturrecherche hat gezeigt, dass in GHD-Umgebungen neben der Anpassung der Informationsdarstellung für die unterschiedlichen Betrachtungsdistanzen auch die Auflösung bzw. die Präzision der Interaktion an die unterschiedlichen Entfernungen angepasst werden kann. Existierende *distance-aware* Techniken unterscheiden zwischen Nah- und Fernbereich mit unterschiedlichen Mappingfaktoren: Präzise Manipulation wird nah am Display unterstützt und aus der Entfernung eine schnelle und ungenaue Manipulation oder es werden unterschiedlichen vordefinierten Distanzen verwendet, bei denen die Präzision mit zunehmenden Abstand zum Display verringert wird (siehe Abschnitt 2.4.1.4). Eine weitere Möglichkeit sind kontinuierliche Distanzen, d.h. die aktuelle Betrachtungs- und Interaktionsdistanz des Nutzers wirkt sich direkt auf den Mappingfaktor aus. Die distanzabhängigen Mappingfaktoren nutzen gezielt das natürliche Verhalten der physischen Navigation, so dass keine zusätzliche Nutzeraktion für den Wechsel der Mappingfaktoren benötigt wird. Um diesen Vorteil nutzen zu können, erfordert der Ansatz, dass der Nutzer sich bewegt.

Bei einer anderen Methode erfolgt der Wechsel auf Wunsch des Nutzers, d.h. er kann explizit zwischen den unterschiedlichen Mappingfaktoren umschalten (*manual-switching*). Damit kann der Nutzer aus jeder Distanz entweder mit hoher oder geringer Präzision interagieren, jedoch wird eine explizite Nutzereingabe benötigt.

In GHD-Umgebungen muss die Nutzereingabe auf die Interaktionstechnik abgebildet werden indem die Parameter Isomorphie, Projektion, Cursorberechnung und Kinematik bei der Berechnung des Mappingfaktors berücksichtigt werden. Davon abhängig, welcher Parameter mehr gewichtet wird, ergeben sich unterschiedliche Berechnungen für den Mappingfaktor. Bei Manipulations- und Navigationsaufgaben in GHD-Umgebungen werden unterschiedliche Präzisionen der Interaktion benötigt (z.B. pixelgenaue Selektion vs. schnelle Positionierung). Um dies zu ermöglichen sollten verschiedene Mappingfunktionen in einer Mappingmethode integriert sein, wobei das Umschalten zwischen den Funktionen entweder manuell oder distanzabhängig erfolgt.

FAZIT: Das **Mapping** wird beeinflusst von (1) den Interaktionsaufgaben in der virtuellen Umgebung, (2) der Eingabemodalität in der physischen Umgebung und (3) den Faktoren des Benutzers, insbesondere der Motorik, der Hand-Augen-Koordination und den subjektiven Eigenschaften (Körpergröße, Händigkeit, Tremor). Da die individuellen und subjektiven Eigenschaften der Nutzer einen starken Einfluss auf die Mappingmethode haben, ist es denkbar, dass bei der distanzabhängigen Interaktion in GHD-Umgebungen die geeignete Mappingmethode und der angepasste Mappingfaktor zwischen den Nutzern variiert.

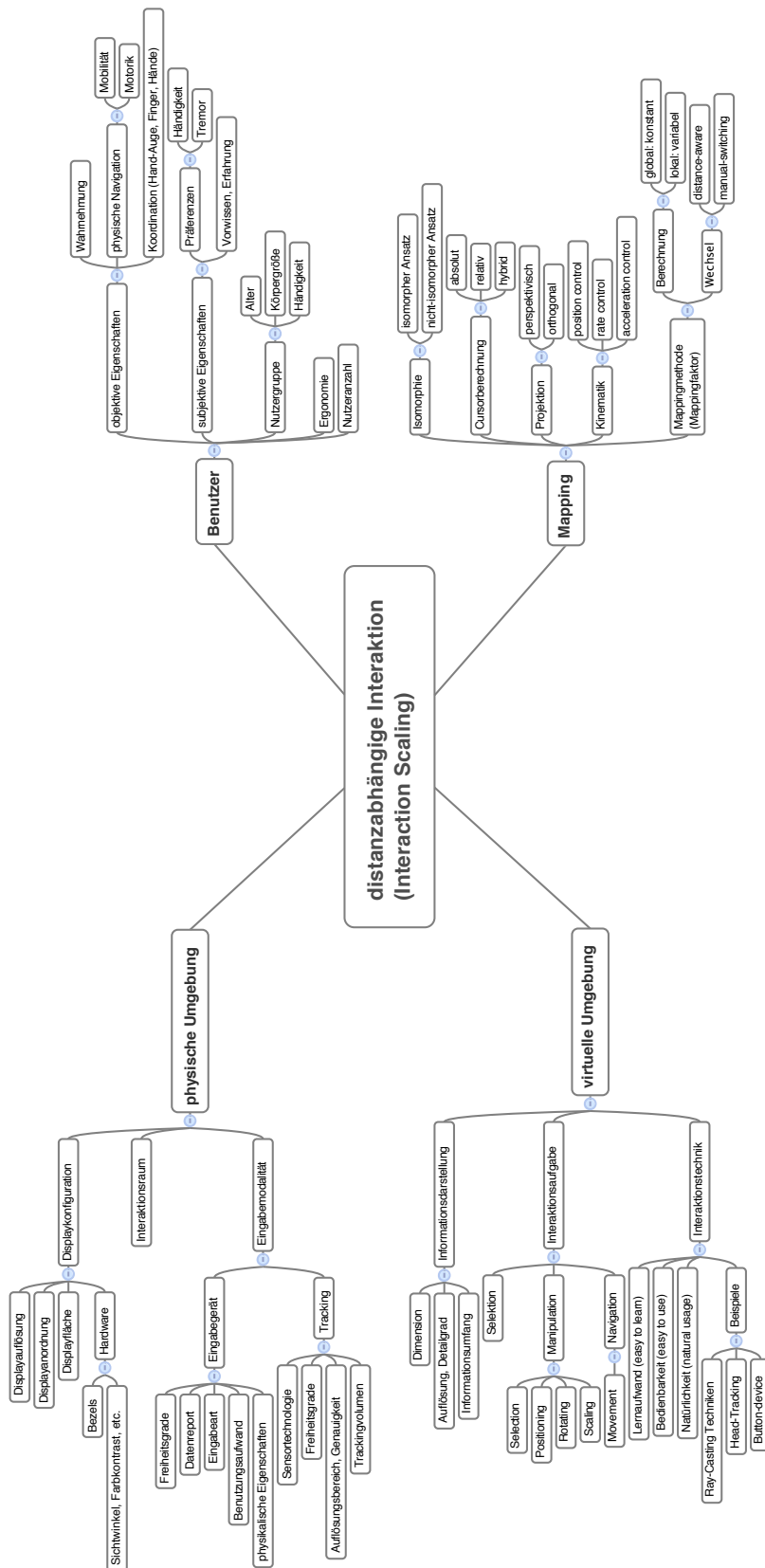


Abbildung 3.1.5.: Die Komponenten des Parameterraumes und ihre zugehörigen Einflussfaktoren für die distanzabhängige Interaktion.

3.2. 3D Interaktion mit Interaction Scaling

In großen hochauflösenden Displayumgebungen soll der Nutzer bei seiner gegenwärtigen Tätigkeit unterstützt werden, indem die Informationsdarstellung und die Interaktivität an den Nutzer angepasst werden. Am Beispiel von Fullscreen-Anwendungen für die Informationsvisualisierung wird gezeigt, dass einerseits nahe am Display detaillierte Informationen bearbeitet werden und andererseits aus der Entfernung mit Informationen interagiert wird. In GHD-Umgebungen bewegt sich der Nutzer frei vor dem Display und interagiert dabei im dreidimensionalen physischen Interaktionsraum – **3D Interaktion**. Die 3D Interaktion bezeichnet die räumliche Nutzereingabe durch physische Navigation vor einem großen hochauflösenden Display. Um mit der Informationsvisualisierung aus unterschiedlichen Betrachtungsdistanzen und mit unterschiedlichem Detailgrad zu interagieren, wird eine unterschiedliche Präzision und Sensitivität der Nutzereingabe erforderlich – **Interaction Scaling**. Interaction Scaling definiert einerseits die distanzabhängige Präzision der Nutzerinteraktion und andererseits die distanzabhängige Informationsdarstellung, basierend auf der physischen Navigation des Nutzers in GHD-Umgebungen.

Dazu wurde ein Parameterraum konzipiert, der die unterschiedlichen Einflussfaktoren in der GHD-Umgebung identifiziert und deren Beziehungen aufzeigt. Im vorherigen Abschnitt wurden die vier Komponenten (physische Umgebung, virtuelle Umgebung, Benutzer, Mapping) mit ihren Faktoren und deren Abhängigkeiten diskutiert (vgl. Abbildung 3.1.5).

Der Fokus der Arbeit und die Anforderungsanalyse geben einige Einschränkungen für die Einflussfaktoren vor:

- Displaykonfiguration: große hochauflösende Displays, ebene Displayanordnung, homogene Displaywand mit Bezels (GHD-Umgebung) und heterogene Displayumgebung mit Zwischenräumen (MD-Umgebung),
- Eingabemodalität: 3D Interaktion (mobile und räumliche Nutzereingabe), optisches Trackingsystem (hohe Präzision, flexibles Trackingvolumen), absolute und relative Eingabegeräte
- Interaktionsraum: vollständige Abdeckung der Ausdehnung der Displayfläche, Unterscheidung zwischen Nahbereich und Fernbereich
- Informationsdarstellung: Informationen mit unterschiedlichem Detailgrad (Anpassung an die Wahrnehmung)
- Interaktionstask: Navigation und Manipulation
- Interaktionstechnik: indirekte Selektions- und Manipulationstechniken, sowie *gaze-steering* Navigationstechniken

- Nutzeranzahl: kollaborative Interaktion an einem Display mit bis zu drei Nutzern
- Ergonomie: einhändige und beidhändige Manipulationsoperationen, blickrichtungsgesteuerte Selektion und Navigation

Die Fullscreen-Anwendung spezifiziert, welche Interaktionsaufgaben mit den virtuellen Inhalten durchgeführt werden sollen und mit welcher Eingabemodalität dies erfolgt. Somit lässt sich die Interaktionstechnik aus der Interaktionsaufgabe und der Eingabemodalität bestimmen.

Gängige Eingabegeräte in GHD-Umgebungen sind Laserpointer, getrackte Stäbe oder kabellose Mäuse und deren Kombination. In dieser Arbeit werden als Eingabemodalität folgende drei Beispiele verwendet (siehe Abbildung 3.2.6). Erstens ein Nintendo Wii Remote Controller³ (Wiimote) – ein mobiles Consumer-Eingabegerät mit Buttons, das für die räumliche Interaktion entwickelt wurde, relative Eingabeänderungen erfasst und gute physikalische Eigenschaften besitzt (haptisches Feedback, geeignet für Links- und Rechtshänder). Zweitens Head-Tracking – eine Baseballkappe mit optischen Markern, die durch ein kommerzielles optisches Trackingsystem erfasst wird und mit sehr hoher Genauigkeit die absolute Kopfposition und -orientierung des Nutzers im Raum erfasst. Drittens die Kombination der Wiimote mit einem Trackingsystem, so dass absolute Position- und Orientierungsänderungen eines Eingabegerätes mit Buttons möglich sind. Die Eigenschaften der Eingabegeräte sind in Tabelle 3.2.2 zusammengefasst. Die Positions- und Orientierungsdaten (6-DOF) werden als kontinuierliche Datenreports geliefert, während ein Tastendruck auf dem Eingabegerät (1-DOF) einen diskreten Datenreport liefert.

Die Parameter der Eingabegeräte werden auf die Taskparameter der Interaktionsaufgaben gemappt. Da eine pixelgenaue Selektion durch die Interaktionstech-



(a) Nintendo Wii Remote Controller



(b) Basecap mit Markern



(c) Controller mit Markern

Abbildung 3.2.6.: Verwendete Eingabegeräte für die Manipulation und Navigation.

³ <http://www.nintendo.com/wii/>

Parameter	Wiimote	Head-Tracker	getrackte Wiimote
Freiheitsgrade	6-DOF, 1-DOF	6-DOF	6-DOF, 1-DOF
Datenreports	kontinuierlich, diskret	kontinuierlich	kontinuierlich, diskret
Eingabeart	indirekt relativ	indirekt absolut	indirekt absolut
Benutzungsaufwand	aktiv	passiv	passiv und aktiv
Physikalische Eigenschaften	haptisches Feedback, handliches Gerät	kein haptisches Feedback, guter Tragekomfort mit Basecap	haptisches Feedback, handliches Gerät

Tabelle 3.2.2.: Parameter der verwendet Eingabegeräte: (a) Nintendo Wii Remote Controller (Wiimote), (b) Head-Tracker und (c) Button-device mit Tracking (getrackte Wiimote)

nik unterstützt werden soll, entfallen für die Selektion die Taskparameter Zielobjektgröße und Objektdichte um das Zielobjekt.

In dieser Arbeit werden die verbreiteten Interaktionstechniken Ray-Casting für die Manipulation und Head-Tracking für die Navigation eingesetzt. Die verwendeten Interaktionstechniken und Eingabegeräte geben ebenfalls Einschränkungen für das Mapping vor (z.B. Realismus, Cursorberechnung, Kinematik).

3.2.1. Mappingfunktionen

Im folgenden Abschnitt werden verschiedene Mappingfunktionen für die distanzabhängige Interaktion vorgestellt. Die Berechnung des Mappingfaktors kann global oder lokal erfolgen (vgl. Abschnitt 3.1.4). Bei einer **globalen** Berechnung wird ein konstanter vorberechneter Mappingfaktor verwendet (z.B. konstante Mappingfunktion). Bei einer **lokalen** Berechnung erfolgt die Bestimmung des Mappingfaktors abhängig von der Nutzereingabe, entweder geräteabhängig (z.B. Eingabegeschwindigkeit, *manual-switching*) oder distanzabhängig.

Statische oder konstante Mappingfunktion: Die statische Mappingfunktion bildet Bewegungen im realen Raum direkt auf die Cursorbewegungen im virtuellen Raum ab, entweder 1:1 oder 1:n Abbildung. Es findet keine distanzabhängige Anpassung der Präzision der Interaktion statt.

Für das Interaction Scaling soll die Präzision der Nutzereingabe oder die Informationsdarstellung, abhängig von der aktuellen Betrachtungsdistanz erfolgen. Dafür wird die Ausdehnung des physischen Interaktionsraumes genutzt, sowie dessen definierter Nahbereich (max. Betrachtungsdistanz für die Wahrnehmung individueller Pixel, Gleichung 3.1.1) und Fernbereich (min. Betrachtungsdistanz für die Wahrnehmung der gesamten Displayfläche, Gleichung 3.1.2).

Denkbare distanzabhängige Mappingfunktionen sind: (1) mit kontinuierlichem Mappingfaktor (linear, exponentiell), (2) mit vordefinierten festen Mappingfaktoren

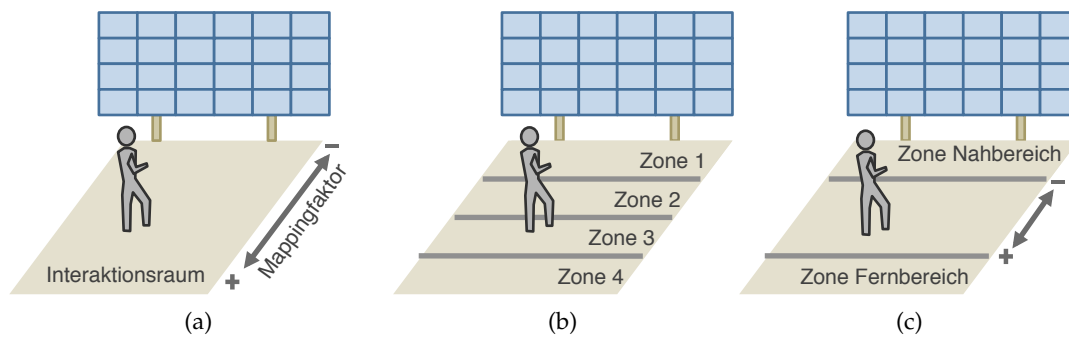


Abbildung 3.2.7.: Einteilung des physischen Interaktionsraumes für die Mappingfunktionen mit: (a) kontinuierlichem Mappingfaktor, (b) vordefinierten Mappingfaktoren, (c) vordefinierten und kontinuierlichen Mappingfaktoren.

ren (zonenartig) und hybride Mappingfunktionen mit (3) vordefinierten und kontinuierlichen Mappingfaktoren (siehe Abbildung 3.2.7).

Lineare Mappingfunktion: Bei der linearen Mappingfunktion wird ein linearer Faktor auf die physischen Bewegungen angewendet. Der Faktor wächst mit zunehmendem Abstand zur Displaywand. Im Nahbereich und mittlerer Distanz wird ein Mappingfaktor kleiner eins berechnet. Somit müssen größere reale Bewegungen erfolgen, um virtuell kleine Cursorbewegungen zu erreichen. Je kleiner der Mappingfaktor ist, desto höhere Präzision kann erreicht werden. Im Fernbereich ist der Mappingfaktor größer eins und die Präzision der Interaktion ist gröber, z.B. können die virtuellen Objekte mit kleineren realen Bewegungen um größere virtuelle Bewegungen manipuliert werden.

Exponentielle Mappingfunktion: Bei der exponentiellen Mappingfunktion wächst der Mappingfaktor kontinuierlich mit zunehmendem Abstand zur Displaywand. Im Nahbereich wird ein Mappingfaktor kleiner eins berechnet, der nur langsam wächst und somit eine sehr hohe Präzision der Interaktion ermöglicht. Mit mittlerer Entfernung wächst der Mappingfaktor schneller, so dass mit zunehmendem Abstand kleinere reale Bewegungen nötig sind, um die gleichen virtuellen Bewegungen zu erreichen. Im Fernbereich ist der Mappingfaktor größer eins und die Präzision der Interaktion wird gröber.

Zonen Mappingfunktion: Der Interaktionsraum wird in Zonen unterteilt. Die zonenartige Mappingfunktion verwendet vordefinierte Mappingfaktoren in den einzelnen Zonen. Der Mappingfaktor nimmt mit jeder Zone von der Displaywand entfernt zu. Wenn die Zonen sehr dicht beieinander liegen, kann ein Schritt vorwärts oder rückwärts zur Displaywand zu einem schnellen Wechsel des Mappingfaktors führen. Hier können Schwellbereiche zwischen den Zonen sinnvoll sein.

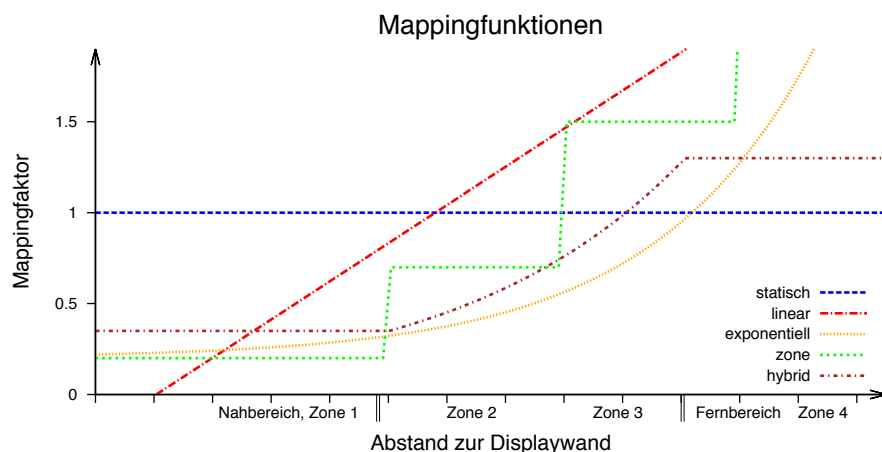


Abbildung 3.2.8.: Schematische Abbildung des Mappingfaktors abhängig von der Mappingfunktion und der Betrachtungsdistanz.

Hybride Mappingfunktion: Bei der hybriden Mappingfunktion wird der Interaktionsraum in drei Bereiche unterteilt. Im Nah- und Fernbereich werden vordefinierte Mappingfaktoren verwendet. Im dazwischen befindlichen Bereich wächst der Mappingfaktor kontinuierlich (z.B. linear, exponentiell) mit zunehmendem Abstand zur Displaywand.

Kontinuierliche distanzabhängige Mappingfunktionen sind für den Nutzer gut nachvollziehbar, d.h. der Nutzer bewegt sich hin zu oder weg von der Displaywand und dementsprechend nimmt die Präzision der Interaktion zu oder ab (siehe Abbildung 3.2.8). Bei konstanten distanzabhängigen Mappingfunktionen kann ein plötzliches Umschalten zwischen der Interaktionspräzision den Nutzer irritieren.

Kombination der Mappingfunktionen: In einigen Anwendungsfeldern können verschiedene Anforderungen für die Präzision existieren, z.B. soll mittels Handbewegungen die Objektskalierung um wenige Millimeter möglich sein, während die Objektplatzierung schneller und mit einer gröberen Genauigkeit erfolgen soll. Dies bedeutet für die Subtasks (z.B. Positionierung, Rotation, Skalierung), dass unterschiedliche Präzisionen für die gleiche Interaktionsdistanz benötigt werden. Hier wird beim Mapping neben der Interaktionsdistanz auch zwischen den Interaktionstasks für die Berechnung des Mappingfaktors unterschieden (siehe Abbildung 3.3.12a).

Nicht-distanzabhängige Mappingfunktionen: Alternativ zu den distanzabhängigen Mappingfunktionen kann die Anpassung der Präzision der Interaktion durch den Nutzer erfolgen. Eine herkömmliche Methode ist das *manual-switching*, bei denen der Nutzer zwischen einem Mappingfaktor für präzise Interaktion und einem Faktor für grobes und schnelles Interagieren umschalten kann. Diese Methode eignet sich insbesondere für Nutzer, welche die explizite Kontrolle über die Präzision der Interaktion benötigen, bedarf aber einer Umschaltfunktion als Nutzereingabe.

3.3. Beispielszenarien

Anhand von Beispielszenarien wird gezeigt, welche Mappingmethode basierend auf der Gewichtung der Einflussfaktoren, d.h. bei unterschiedlichen Randbedingungen und Einschränkungen der Anwendung, für das Interaction Scaling genutzt werden kann.

Es werden zwei hochauflösende Displayumgebungen betrachtet: eine homogene und eine heterogene Displayumgebung. Bei der homogenen Displayumgebung sind identische LCD Bildschirme zu einer zusammenhängenden planaren Displaywand angeordnet, die durch einen PC oder ein PC-Cluster angesteuert wird. Bei der heterogenen Displayumgebung, wie sie beispielsweise in Smart Meeting Rooms [HK05, Thi10] verwendet wird, handelt es sich um mehrere unterschiedliche Displays im Raum, die keine physisch zusammenhängende Displayfläche bilden. Solch eine Multi-Displayumgebung wird oftmals von einem Server angesteuert, der Visualisierungen von den persönlichen Geräten der Nutzer (Laptop, Tablet) entgegennimmt und auf die Großdisplays verteilt. In Tabelle 3.3.3 werden die charakteristischen Eigenschaften der beiden Displayumgebungen gegenübergestellt und welche Parameter für Interaction Scaling (IS) genutzt werden.

Als Interaktionsaufgaben werden die Navigation und die Manipulation diskutiert. Bei der Navigation mit Head-Tracking wird die physische Navigation, anstatt der virtuellen Navigation genutzt und der Detailgrad der Informationsdarstellung für 2D Daten und 3D Daten angepasst. Bei den Manipulationstechniken werden einhändige und beidhändige Nutzereingaben erörtert, die unterschiedliche Freiheitsgrade und Nutzergruppen adressieren. Insbesondere wird die Auswahl der verfügbaren Genauigkeit für die Objektmanipulation, abhängig vom Abstand des Nutzers zur Displaywand diskutiert. Es werden jeweils zwei Szenarien für jede Interaktionstechnik diskutiert, welche unterschiedliche Faktoren im Parameterraum festlegen, die sich auf die Mappingmethode auswirken (siehe Tabelle 3.4.5).

3. Konzept der distanzabhängigen Interaktion

Parameter	GHD-Umgebung	MD-Umgebung
physische Umgebung		
Displaykonfiguration	homogen	heterogen
– Anordnung	eben	unterschiedlich
– Auflösung	hochauflösend	hochauflösend, high-definition
– Fläche	zusammenhängend	getrennt
– Bezels	ja	nein, aber Zwischenräume
Tracking	optisches Tracking (IR), sehr hohe Genauigkeit (mm-Bereich)	z.T. optisches Tracking (RGB), gute Genauigkeit (cm-Bereich)
physischer Interaktionsraum	definiert durch Trackingvolumen und Displayfläche	nicht zusammenhängender Interaktionsraum
Eingabemodalität	Zeigergeräte, Head-Tracking	Maus-ähnliche Nutzereingabe
virtuelle Umgebung		
Informationsdarstellung	Berücksichtigung der Bezels	zusammenhängende Informationen auf einem Display
– Dimension	2D, 3D	2D
– Detailgrad	angepasst an variable Betrachtungsdistanz	angepasst für mehrere vordefinierte (feste) Betrachtungsdistanzen
Interaktionstask	Manipulation, Navigation	Manipulation
Interaktionstechnik	3D Interaktion, Head-Tracking	2D Interaktion, Maus-Adaption
Benutzer		
Nutzeranzahl	Einzelbenutzer, Mehrbenutzer (2-3)	Mehrbenutzer (≥ 2)
Wahrnehmung	variable Betrachtungsdistanz	feste Betrachtungsdistanz
physische Navigation	Motorik, Mobilität	Motorik
ergonom. Nutzereingabe	Kopf, 1-händig, 2-händig	1-händig
Handzittern	ja	nein
Interaktionspräzision	grob und sehr präzise	grob (Großdisplay), präzise (Laptopdisplay)
Mapping		
Isomorphie	nicht-isomorph	nicht-isomorph
Kinematik	position control	rate control
Eingabeänderung	absolute Eingabe	relative Eingabe
→ Cursorberechnung	hybrid, relativ	relativ
Projektion	orthogonal, perspektivisch	keine
Seiteneffekte	Drift, Clutching	Clutching

Tabelle 3.3.3.: Eigenschaften der verwendeten Displayumgebungen und deren Parameter für Interaction Scaling.

3.3.1. Navigation und Selektion mit Head-Tracking

Interaction Scaling erfolgt hier in Form der distanzabhängigen Informationsdarstellung, basierend auf der physischen Navigation des Nutzers in GHD-Umgebungen. Der Nutzer interagiert mit der Visualisierung indem er sich vor der Displaywand frei bewegt und somit wird der Detailgrad der Informationsdarstellung in der virtuellen Umgebung, abhängig vom Betrachtungsabstand ausgewählt. Eine Anpassung des Detailgrades kann beispielsweise erforderlich sein, wenn nicht ausreichend Displayfläche für die Visualisierung zur Verfügung steht oder das Rendern der virtuellen Objekte beschleunigt werden soll.

Im Folgenden wird diskutiert, wie Interaction Scaling für die Informationsdarstellung in einer 2D Anwendung (z.B. 2D Graphvisualisierung) und für das Rendern in einer 3D Anwendung (z.B. 3D Geländedarstellung) verwendbar ist. Dabei

gibt es folgende Einschränkungen für die Einflussfaktoren der physischen Umgebung, der virtuellen Umgebung und des Benutzers:

- Displaykonfiguration: flache LCD-Displaywand mit Bezels
- Eingabemodalität und Interaktionstechnik: Head-Tracking
- Informationsdarstellung: unterschiedliches Detailgrad, 2D oder 3D Objekte
- objektive Eigenschaften: Einsatz des fovealen und peripheren Sehens, Lesbarkeit und kognitive Verarbeitung
- Nutzeranzahl: ein bis zwei Nutzer im physischen Interaktionsraum

Die Interaktionstechnik Head-Tracking nutzt die physische Navigation des Nutzers in der physischen Umgebung zur Navigation in der virtuellen Umgebung. Dazu wird die absolute Kopfposition und -orientierung (6-DOF) des Nutzers im physischen Interaktionsraum getrackt. Die Parameter der Eingabemodalität werden wie folgt auf die Task-Parameter abgebildet (vgl. Tabelle 2.4.3). Die Blickrichtung bestimmt die Entfernung und Richtung zur Displaywand und somit zum Zielobjekt. Die Änderung der Nutzerposition (Betrachtungsdistanz) wird auf den Skalierungsfaktor des Subtasks Nutzerskalierung abgebildet (Zoom-Faktor). Die Kopforientierung approximiert die Blickrichtung des Nutzers und wird für die Selektion der virtuellen Inhalte genutzt.

Die verwendete Eingabemodalität ermöglicht eine Interaktionstechnik ohne die Nutzung der Hände, womit die Hände für feinmotorische Kontrollaufgaben nutzbar wären.

Interaction Scaling für 2D Anwendung

GHD-Umgebungen eignen sich für die Darstellung von großen Datenmengen. Jedoch ist es für den Nutzer schwierig zu viel dargestellte Informationen zu verarbeiten oder aus der Entfernung detaillierte oder sehr kleine Objekte zu erkennen. Die Verwendung von Interaction Scaling (IS) soll eine interaktive Exploration von Graphen durch physische Navigation und die Anpassung des Detailgrades der Informationsdarstellung an den Betrachtungsabstand ermöglichen.

Als Beispiel wird ein hierarchischer 2D Graph gewählt, d.h. ein *rooted tree* bei dem die Blattknoten Informationen mit der höchsten Detailgenauigkeit repräsentieren, während die Knoten (Nicht-Blattknoten) Abstraktionen ihrer korrespondierenden Kinderknoten sind. Es gibt zwei Möglichkeiten, wie Nutzer das Abstraktionslevel ändern können: Erstens indem der Nutzer global von einer Hierarchieebene zu einer anderen wechseln kann; hier werden alle Knoten des aktuellen Levels mit den Knoten des anderen Levels ersetzt. Zweitens der Nutzer kann Knoten aufklappen oder einklappen um das Abstraktionslevel lokal anzupassen, d.h. das Aufklappen eines Knotens ersetzt lokal den Knoten mit seinen Kinderknoten, was

zu mehr Detail und weniger Abstraktion führt. Die 2D Informationen werden als 2D Grafikprimitive (Kreise, Polygone, Kanten) mit Textlabels visualisiert.

Für die Interaktion soll die physische Navigation durch Head-Tracking als Eingabemodalität verwendet werden und die Wahrnehmung des Benutzers aus unterschiedlichen Betrachtungsdistanzen berücksichtigt werden. Dazu sollen die Informationsdarstellungen, insbesondere die Detailgrade und die Lesbarkeit der Labels, an die visuelle und kognitive Wahrnehmung des Nutzers angepasst werden. Aufgrund der Displaykonfiguration sollten keine Labels von den Bezels verdeckt oder auseinandergezerrt werden.

Ein einfacher Ansatz für Interaction Scaling ist den Interaktionsraum vor dem Display in die gleiche Anzahl von Zonen zu unterteilen wie diskrete Hierarchieebenen des Graphen vorhanden sind.

Für die globale Änderung des Abstraktionslevels eignet sich die Verwendung der Zonen Mappingfunktion: Je weiter weg der Nutzer vom Display ist desto höhere Abstraktionsebene wird angezeigt und die Labels mit einer lesbaren Schriftgröße für die Zone gerendert. Bleibt der Nutzer an der Grenze zwischen benachbarten Zonen stehen, kann der Wechsel zwischen den Abstraktionsebenen problematisch sein. Durch minimale Kopfbewegungen des Nutzers (Tremor) kann es zu einem ständigen Wechsel zwischen den angrenzenden Level kommen. Dadurch wird die Visualisierung stets neu gerendert, was für den Nutzer irritierend und störend ist. Durch das Einfügen sogenannter Schwellbereiche zwischen den Zonen können die ungewollten Kopfbewegungen kompensiert werden (siehe Abbildung 3.3.9a). Folglich werden die Abstraktionslevel erst gewechselt, wenn der Nutzer den Schwellbereich überschreitet, d.h. Umschalten in die jeweils höhere Ebene (rückwärts laufen) oder niedrigere Ebene (vorwärts laufen).

Die geeignete Zonentiefe ist abhängig von der Tiefe des physischen Interaktionsraumes. Dabei sollten die Zonen eine Mindesttiefe haben, damit physische Navigation möglich ist, d.h. der Nutzer explizit einen Schritt macht, der zum Levelwechsel führt und nicht nur eine Kopfneigung vor oder zurück erforderlich ist. Unregelmäßige Zonentiefen sind ebenfalls denkbar, wobei aufgrund der visuellen Wahrnehmung tiefe Zonen eher entfernt und kurze Zonen im Nahbereich geeignet sind (siehe Abbildung 3.3.9b).

Für die Navigation in Graphen bietet sich die Interaktionstechnik *focus-and-context* an, die in diesem Fall die physische Navigation als Nutzereingabe verwendet. Dazu wird die Zonen-Technik mit einer visuellen Linse erweitert. Das approximierte Sichtfeld des Nutzers wird anhand einer ovalen Linse auf der Displayfläche visualisiert (siehe Abbildung 3.3.9c). Die visuelle Linse wird für die Selektion von einzelnen Knoten verwendet. Somit werden die virtuellen Objekte innerhalb der Linse mit einer geringeren Abstraktionsebene, d.h. mit höherem Detailgrad angezeigt (Fokus als lokale Interaktion), als die umgebenden Informationen außerhalb der Linse (Kontext als globale Interaktion). Weiterhin wird der Radius der Linse an den aktuellen Betrachtungsabstand angepasst; mit zunehmenden Abstand zur Dis-

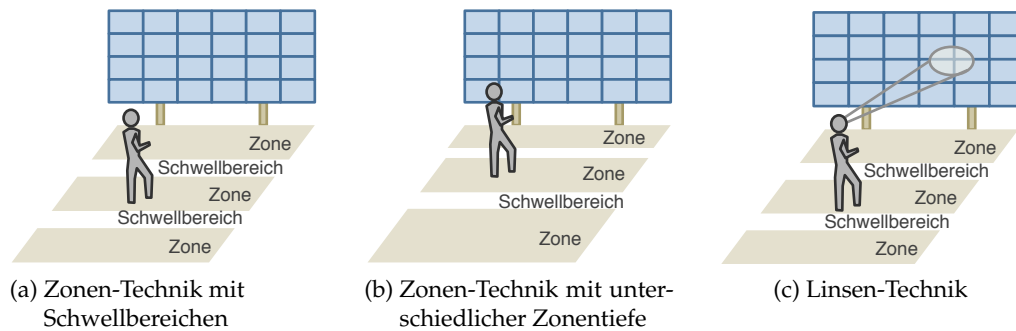


Abbildung 3.3.9.: Nutzung des Interaktionsraumes für die Navigationstechniken zur Exploration von Graphen.

playwand wird die Linse größer, um dem fovealen Sehen zu entsprechen. Damit können aus der Entfernung mehr Knoten selektiert werden (großer Selektionsradius) als nahe am Display (kleiner Selektionsradius).

Die Linsen-Technik als distanzabhängige Interaktion unterstützt somit die Anpassung des Detailgrades (Abstraktionsebene) und des Selektionsradius (Linsengröße).

Sollen die beschriebenen Navigationstechniken zwei Benutzer kollaborativ in der GHD-Umgebung unterstützen, dann ist zu klären welches Abstraktionslevel auf dem Display visualisiert werden soll. Dabei können drei Fälle unterschieden werden: (1) Beide Nutzer befinden sich in der selben Zone, dann wird das Abstraktionslevel global bestimmt. (2) Die Nutzer befinden sich in benachbarten Zonen, hier ist denkbar die zugehörigen Abstraktionslevel parallel anzuzeigen, sofern es die Anzeigefläche zulässt. (3) Die Nutzer befinden sich in entfernten Zonen, dann muss der kleinste gemeinsame Nenner gefunden werden.

Interaction Scaling für 3D Anwendung

Für das Rendering von komplexen Objekten (z.B. Geländedarstellung) werden *level-of-detail* (LOD) Algorithmen eingesetzt, bei denen unterschiedlich detaillierte Darstellungsformen der virtuellen Objekte erstellt werden, die abhängig von der verwendeten Fehlermetrik ausgewählt und dargestellt werden. Bei den blickpunktabhängigen LOD-Algorithmen für das Rendering von 3D Terrains ändert sich die Detailauflösung des Geländes abhängig vom Abstand des Betrachters (virtuelle Sichtposition) zum Objekt. Somit werden in der virtuellen Umgebung nahe liegende Objekte detaillierter dargestellt (viele Polygone), während entfernte Objekte approximiert dargestellt werden (wenig Polygone). In GHD-Umgebungen in denen der Nutzer seine physische Sichtposition ändert, kann die aktuelle Betrachtungsposition als Parameter für die LOD-Berechnung verwendet werden. Können zum Beispiel Bereiche der Displayfläche vom Nutzer nicht wahrgenommen werden, dann kann auf ein Objektrendering in diesen Displaybereichen verzichtet werden und die verfügbaren Ressourcen für die sichtbare Displayfläche genutzt werden. Somit

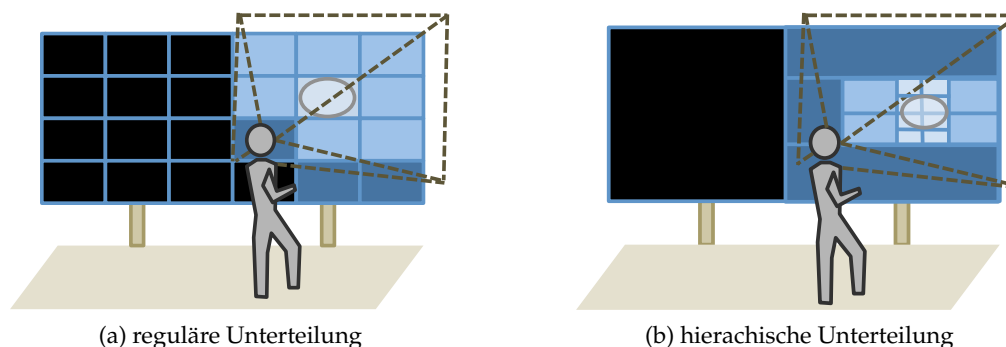


Abbildung 3.3.10.: Aktive Sichtzellen mit unterschiedlichem Detailgrad. Je dunkler die Sichtzelle desto geringer ist der Detailgrad.

müssen zwei Bedingungen für die LOD-Bestimmung der virtuellen Objekte berücksichtigt werden: der Abstand des Betrachters zum 3D Objekt und die Position des 3D Objektes auf der Displayfläche.

Mit Hilfe des Head-Tracking kann bestimmt werden, welche Bereiche der Displaywand der Nutzer betrachtet und für das Rendering genutzt werden sollten. Dazu wird die Displayfläche in Sichtzellen unterteilt. Entweder wird eine reguläre Unterteilung gewählt, z.B. bei gekachelten LCD-Wänden entspricht jedes Display einer Sichtzelle, oder es erfolgt eine hierarchische Unterteilung der Anzeigefläche, bei der Displaybereiche mit gleichen Eigenschaften zusammengefasst werden. Eine reguläre Einteilung ermöglicht das Vorberechnen von Displaybereichen, kann aber nicht flexibel an die Anwendung angepasst werden. Die hierarchische Unterteilung kann hingegen dynamisch für die Anwendung erfolgen, sie benötigt jedoch einen höheren Berechnungsaufwand.

Die Sichtzellen haben die Eigenschaft entweder aktiv zu sein, d.h. es werden Objekte gerendert oder sie sind inaktiv, d.h. es müssen keine Objekte gerendert werden. Die aktiven Sichtzellen nutzen die physiologische Wahrnehmung des Nutzers: Die Sichtzellen im Fokus des Nutzers rendern die Objekte mit einem hohem Detailgrad und je weiter die Sichtzellen vom Fokusbereich entfernt sind, desto geringer wird der verwendete Detailgrad. Abbildung 3.3.10 illustriert die Auswahl der Sichtzellen abhängig von der Nutzerposition: Die ovale Linse visualisiert das foveale Sehen und die Sichtzelle mit hohem Detailgrad (hellblau); die gepunkteten Linien visualisieren das periphere Sehfeld, in dem der Detailgrad der Sichtzellen mit zunehmender Entfernung vom Fokus abnimmt (mittelblau); die inaktiven Sichtzellen außerhalb der Peripherie rendern keine Geometrien (schwarz).

Da der Nutzer im peripheren Sehen empfindlich für bewegliche Objekte ist, sollten die Sichtzellen unmittelbar außerhalb des Sichtfeldes die virtuellen Objekte mit einer geringen Detailstufe rendern. Somit wird ein plötzliches An- und Abschalten der Sichtzellen in der Peripherie vermieden und die Verfeinerung der Geometrie flüssiger gestaltet.

Interagieren zwei Benutzer gleichzeitig mit der Anwendung, dann werden für jeden Benutzer die blickpunktabhängige Fehlermetrik und die zugehörigen aktiven Sichtzellen berechnet. Bei überlappenden aktiven Sichtzellen wird die Sichtzelle mit dem höheren Detailgrad gerendert. Wirkt sich das Rendering der Detailinformationen, aufgrund des höheren Berechnungsaufwandes, nachteilig auf die Interaktivität aus, dann sollten die Sichtzellen mit einem geringeren Detailgrad dargestellt werden. Denn für die visuelle Wahrnehmung sind Latenzen bzw. verzögerte Interaktivität für den Benutzer störender als ein geringerer Detailgrad.

Wird Head-Tracking wie beschrieben zur Navigation in 2D oder 3D Anwendungen verwendet, dann ergeben sich folgende Eigenschaften für die Mappingmethode. Die Nutzereingabe wird auf Positionsdaten (*position control*) mit perspektivischer Projektion abgebildet, um das foveale und periphere Sehen durch die approximierende Kopforientierung zu berücksichtigen. Wird nur die Betrachtungsdistanz und nicht der Fokusbereich des Nutzers benötigt, kann auch eine orthogonale Projektion verwendet werden. Die Betrachtungsdistanz wird für die Darstellung des Detailgrades genutzt, wobei der Tremor die Informationsdarstellung beeinflusst. Durch die Verwendung von Schwellbereichen oder aktiven Sichtzellen im erweiterten peripheren Sehfeld können die ungewollten Positionsänderungen für den Nutzer unbemerkt bleiben. Es wird ein nicht-isomorpher Ansatz für die Bestimmung des Zoom-Faktors verwendet, der diskrete vordefinierte oder kontinuierliche Mappingfaktoren verwendet.

Bei den verwendeten Navigationstechniken wird ein visueller Cursor im eigentlichen Sinne nicht verwendet (keine Cursorberechnung). Die Blickrichtung kann jedoch für die Selektion genutzt werden. Dazu wird eine virtuelle Linse visualisiert, die den Fokus des Nutzers simuliert und alle virtuellen Objekte innerhalb der Linse werden als selektiert markiert. In diesem Fall empfiehlt sich eine absolute Cursorberechnung für die visuelle Linse.

Die Umsetzung der Szenarien für Interaction Scaling in Form der distanzabhängigen Informationsdarstellung ist in Abschnitt 5.2 beschrieben. Die Beispiele wurden in Zusammenarbeit mit Studenten und Kollegen erarbeitet: die 2D Anwendung mit Thomas Gertz und Christian Tominski [Ger11, LSST11], die 3D Anwendung mit Falko Löffler und Steffen Girke [Gir13, SLL⁺14].

3.3.2. Manipulation mit einhändiger Interaktion

Interaction Scaling als distanzabhängige Präzision der Interaktion wird im Folgenden am Beispiel der einhändigen Interaktion für Manipulationsaufgaben gezeigt. Es werden zwei Interaktionstechniken gezeigt, die unterschiedliche Displaykonfigurationen, Eingabemodalitäten und Nutzergruppen adressieren.

Der Fokus liegt hier auf den unterschiedlichen Displaykonfigurationen und Nutzergruppen, welche die Nutzereingabe und somit die Mappingmethode gewichten. Als Displaykonfiguration wird eine heterogene Multi-Display-Umgebung mit

unterschiedlichen Displayanordnungen und eine homogene große hochauflösende Displayumgebung verwendet. Folgende Einschränkungen werden für die Einflussfaktoren vorgenommen:

- Displaykonfiguration: heterogene MD-Umgebung, flache homogene GHD-Umgebung
- Eingabemodalität: Controller mit relativer oder absoluter Eingabeart
- Informationsdarstellung: 2D Daten
- Interaktionstechnik: Maus-Adaption, Ray-Casting
- Ergonomie und Nutzergruppe: einhändige Nutzereingabe, Erwachsene oder Kinder
- Nutzeranzahl: ein bis mehrere Nutzer im physischen Interaktionsraum

Als Eingabegerät für die Manipulationsoperationen wird eine Wiimote verwendet mit der relativen Nutzereingaben oder in Kombination mit einem Trackingsystem absolute Nutzereingaben unterstützt werden (vgl. Tabelle 3.2.2). Die Wiimote wird im physischen Interaktionsraum bewegt und die Positionsänderungen oder Tastendrucke werden auf die Taskparameter abgebildet. Bei der einhändigen Interaktion werden die Subtasks Selektion, Positionierung, Skalierung und Rotation durch unterschiedliche motorische Bewegungsabläufe gesteuert. Die Interaktionsmethode ermöglicht dem Nutzer seine dominante Hand für die feinfühligere Nutzereingabe zu verwenden. Die horizontale Anordnung der Buttons auf der Wiimote ermöglicht eine Reihe von Modusänderungen per Tastendruck mit dem Daumen. Da eine räumliche Interaktion unterstützt wird, muss das natürliche Handzittern bei der Wiimote-Steuerung berücksichtigt werden.

Interaction Scaling für relative Nutzereingabe

Üblicherweise gibt es bei Diskussionsrunden im Smart Meeting Room (SMR) einen Moderator oder einen Vortragenden, der sich während der Präsentation vor den Großdisplays bewegt und mit der Informationsdarstellung interagiert. Darüber hinaus können auch die Zuhörer mit den Informationen interagieren, einmal lokal innerhalb ihrer persönlichen Geräte und ebenso global innerhalb der gemeinsamen Großdisplays (siehe Abbildung 3.3.11). Dabei ist zu berücksichtigen, dass sich die Nutzereingaben nicht gegenseitig behindern oder zu nicht nachvollziehbaren Interaktionsabläufen führen. Dies kann beispielsweise durch Rollenverteilung und Zugriffsrechten zwischen den Personen (aktiver Redner und passiver Zuhörer) geregelt werden. Im SMR ist der Betrachtungsabstand zu den einzelnen Großdisplays für jeden Zuhörer anders und die Nutzer agieren von einer definierten Sitzposition aus mit der Informationsdarstellung.

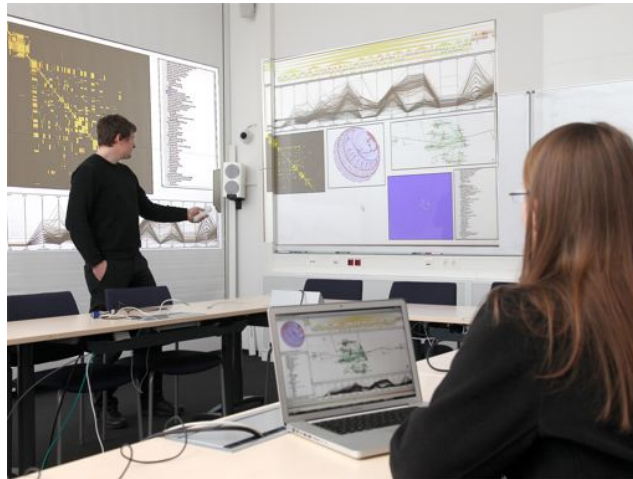


Abbildung 3.3.11.: Diskussionsrunde im Smart Meeting Room.

In der MD-Umgebung sind folgende Einflussfaktoren der Komponente physische Umgebung vorgegeben: (1) die Displayanordnung ist "räumlich" mit unterschiedlichen Displayanzeigen (Großdisplays an den Wänden und Laptopdisplays auf dem Tisch); (2) die verfügbare Displayfläche ist nicht zusammenhängend, sondern es befinden sich unterschiedliche physische Abstände zwischen den Displays und (3) der physische Interaktionsraum ist nicht zusammenhängend (z.B. lokale und globale Nutzereingabe).

Für die Komponente virtuelle Umgebung ergeben sich daraus folgende Einschränkungen: (1) zusammenhängende Informationen sollten innerhalb eines Displays angeordnet werden; (2) die Informationsdarstellung erfolgt für unterschiedliche vordefinierte Betrachtungspositionen (z.B. Präsentationen auf räumlich gegenüberliegenden Displays); (3) es werden 2D Inhalte angezeigt und (4) die Bedienung des Eingabegerätes mit einer Hand soll grundlegende Manipulationsoperationen unterstützen.

Resultierend aus den Einflussfaktoren der Komponente Benutzer soll die Eingabemodalität sowohl die Sitzposition, als auch das Herumlaufen unterstützen, z.B. indem die Nutzereingabe mittels einer *Air Mouse* oder *Wimote* erfolgt. Dabei handelt es sich um relative und indirekte Eingabegeräte mit Buttons, die in der Luft oder auf einer Ebene 2D Positionsveränderungen erzeugen. Sie ermöglichen es, bekannte Mausinteraktionen für GHD- und MD-Umgebungen anzupassen.

Die Anpassung der Informationsdarstellung erfolgt mit Hilfe des *Smart View Management*⁴ im SMR [RLS11], das sogenannte *Views* (z.B. Bilder, Visualisierungen) von unterschiedlichen persönlichen Geräten der Nutzer entgegen nimmt, interaktiv gruppiert und als kombinierte Anzeige auf einem Großdisplay anordnet. Somit werden Informationen zusammengehörig und für unterschiedliche Betrachtungspositionen auf mehrere Displays angezeigt. Das System ermöglicht, visuelle Infor-

⁴ Die Informationsdarstellung im Smart Meeting Room wurde im Rahmen der Dissertation von Axel Radloff [Rad14] im GRK MuSAMA bearbeitet.

mationen von unterschiedlichen Anwendungen zu kombinieren. Die Views können als 2D Objekte betrachtet werden, die der Nutzer manipulieren kann.

Dabei wird zwischen zwei Varianten der Manipulation unterschieden: (1) Interaktionen, die das Layout der Views verändern (anwendungsunabhängige Manipulationsoperationen) und (2) Interaktionen, die Inhalte innerhalb eines Views verändern (anwendungsabhängige Manipulationsoperationen). Die Auswahl der verfügbaren Manipulationsoperationen ist abhängig von der Nutzerposition und der Nutzerrolle im SMR. Angelehnt an die herkömmliche Mausinteraktion, wie diese für anwendungsabhängige Tasks am Laptop des Nutzers durchgeführt werden, können die Task-Parameter für die anwendungsunabhängige Interaktion mit der Wiimote als Eingabegerät folgendermaßen abgebildet werden (siehe Tabelle 3.3.4).

Ein View wird automatisch selektiert indem der Cursor auf den View bewegt wird, angelehnt an den *mouse hover* Effekt. Der selektierte View kann durch Aktivieren der Positionstaste und Armbewegungen bzw. relative Positionsänderungen des Eingabegerätes verschoben werden. Ein Loslassen der Positionstaste platziert den View an der aktuellen Position. Ein selektierter View kann durch Drücken einer der beiden Skalierungstasten in diskreten Schritten entsprechend vergrößert oder verkleinert werden. Eine alternative Eingabe für die Skalierung ist die Verwendung des Beschleunigungssensors des Eingabegerätes für eine kontinuierliche Skalierung, z.B. durch Kippen des Handgelenkes. Das System arrangiert gegebenenfalls die umliegenden Views auf einem Großdisplay automatisch, um die Anpassung der Informationsdarstellung zu erhalten. Rotation von Views macht in diesem Anwendungsszenario wenig Sinn und wird daher nicht betrachtet.

Die Subtasks werden getrennt ausgeführt und erfolgen aus Nutzersicht über unterschiedliche motorische Koordinationsmöglichkeiten: Das Aktivieren eines Subtasks erfolgt per Tastendruck mit dem Finger, die Positionierung wird über die Armbewegung gesteuert und die kontinuierliche Skalierung über das Kippen des Handgelenkes (vertikale Achse des Gerätes).

Die Modusänderung zwischen anwendungsabhängigen und anwendungsunabhängigen Nutzereingaben erfolgt über eine Umschalttaste. Die anwendungsabhängige Nutzereingabe erfordert und ermöglicht präzisere Manipulationsoperationen.

Das Eingabegerät liefert relative Positionsänderungen, die auf die Bewegungsgeschwindigkeit des Cursors abgebildet werden (*rate control*). Daher kann für die Mappingmethode eine relative Cursorberechnung verwendet werden, die keine Projektion erfordert und einen nicht-isomorphen Ansatz verwendet. Jedoch verursacht das relative Mapping viel Clutching, wenn Views über mehrere Displays bewegt werden sollen. In diesem Zusammenhang bietet Interaction Scaling basierend auf der Nutzerposition (sitzend, herumlaufend) eine angepasste Präzision der Interaktion. Erfolgt die Nutzereingabe vor einem Großdisplay werden meist anwendungsunabhängige Operationen ausgeführt, die schnell und grob erfolgen können, sowie anwendungsabhängige Manipulationsoperationen, die eine geringere Prä-

Task	Eingabegerät Maus	Eingabegerät Wiimote
View Selektion	Cursor per Mausbewegung auf View bewegt (<i>mouse hover</i>)	Cursor per Controllerbewegung auf View bewegt
View Positionierung	linke Maustaste gedrückt + Mausbewegung	A-Taste gedrückt + Positionsbewegungen
View Skalierung – diskret – kontinuierlich	Mausrad hoch/runter bewegen	Plus/Minus-Taste drücken
	mittlere Maustaste gedrückt + Mausbewegung entlang der vertikalen Achse	B-Taste gedrückt + Beschleunigung entlang der vertikalen Achse
Modusänderung	[insert]-Taste auf der Tastatur drücken	Home-Taste drücken

Tabelle 3.3.4.: Task-Parameter für relative Nutzereingabe mit unterschiedlichen Eingabegeräten.

zision benötigen (z.B. Selektion eines größeren Datensatzes). Hier eignet sich ein vordefinierter konstanter Mappingfaktor, der für alle Nutzer global gilt. Wohingegen am kleineren Laptopdisplay anwendungsabhängige Operationen durchgeführt werden, die eine hohe Präzision benötigen (z.B. pixelgenaue Selektion einer Datenzelle) und seltener anwendungsunabhängige Operationen. Für die präzise Manipulation kann somit jeder Nutzer seinen individuellen Mappingfaktor (für spezifische Geräte- oder Nutzereigenschaften) einstellen. Um den physischen Benutzungsaufwand beim Wechsel zwischen den Großdisplays gering zu halten, erfolgt eine initiale Cursorpositionierung mittig auf dem aktiven Display (der Nutzer schaut auf das Großdisplay oder steht vor diesem).

Eine Mehrbenutzerunterstützung kann dahingehend realisiert werden, indem Nutzereingaben sowohl mit der Maus am lokalen Gerät als auch mit der Wiimote im Raum erfolgen und lediglich gerätespezifische Tasten und unterschiedliche Präzisionen verwendet werden.

Die dargestellte Interaktionstechnik und deren Umsetzung ist in einer gemeinsamen Kooperation mit Axel Radloff entstanden und als *Smart Interaction Management* im Paper von Radloff et al. [RLSS12] näher beschrieben.

Interaction Scaling für absolute Nutzereingabe

In einer GHD-Umgebung sollen Kinder 2D Objekte auf der Displayfläche manipulieren können. Mit Hilfe des Interaction Scaling sollen sie in der Lage sein die virtuellen Objekte millimetergenau zu platzieren und zu drehen. Für die Interaktion wird eine getrackte Wiimote als Eingabegerät und eine Ray-Casting Technik verwendet. Dies ermöglicht eine natürliche Bedienbarkeit, indem Positions- und Orientierungsänderungen des Eingabegerätes auf Manipulationsbewegungen in der virtuellen Umgebung abgebildet werden (vgl. Abschnitt 3.1.3).

Die Nutzerinteraktion funktioniert folgendermaßen: 2D Objekte lassen sich selektieren, indem der virtuelle Cursor der Hand auf ein Objekt bewegt wird und

die Selektionstaste gedrückt wird. Wird die Selektionstaste losgelassen, so wird das Objekt an der aktuellen Position platziert. Das selektierte Objekt kann entsprechend der Hand- oder Armbewegung verschoben oder durch Handgelenkrotation gedreht werden. Als Rotationszentrum wird der Objektmittelpunkt verwendet, weshalb der virtuelle Cursor automatisch bei erfolgreicher Selektion auf den Objektmittelpunkt verschoben wird. Um die Hand-Augen-Koordination zu unterstützen, werden die Objekte entsprechend der physischen Bewegungen in die gleiche Richtung in der virtuellen Umgebung verschoben oder gedreht und die Änderungen erfolgen kontinuierlich.

Die Rotation und Positionierung kann gleichzeitig erfolgen, wobei die Trennung der Subtask für den Nutzer über unterschiedliche motorische Bewegungen stattfindet: Die Selektion erfolgt mit dem Finger, die Rotation wird über das Handgelenk gesteuert und die Positionierung über die Armbewegung. Zur Differenzierung der Objektskalierung von den anderen Subtasks kann eine Kombination der motorischen Bewegungen verwendet werden. Das Kippen des Handgelenkes initiiert die Skalierung und die Streckung des Armes bestimmt den kontinuierlichen Zoomfaktor.

Damit eine millimetergenaue Objektmanipulation durchführbar ist, werden die Einflussfaktoren für die Mappingmethode wie folgt berücksichtigt. Die Verwendung einer getrackten Wiimote ermöglicht eine absolute Nutzereingabe im physischen Interaktionsraum (6-DOF), die auf Positionsänderungen (*position control*) abgebildet werden. Damit alle Pixel auf der Displaywand (z.B. Displayhöhe von 3 m) für Kinder (Körpergröße unter 1,50 m) erreichbar sind, wird eine perspektivische Projektion beim Ray-Casting verwendet (vgl. Abschnitt 3.1.4).

Für die Manipulationsoperationen werden präzise und schnelle Bewegungen benötigt, weshalb sich eine hybride Cursorberechnung anbietet. Beim Bewegen des virtuellen Cursors über große Distanzen auf der Displayfläche sind schnelle und unpräzise Aktionen geeignet (absoluter Cursorberechnung). Erst wenn ein Objekt selektiert ist, werden präzise Manipulationsbewegungen benötigt (relative Cursorberechnung). Das absolute Mapping ermöglicht die Verwendung eines isomorphen Ansatzes, der das Finden des Cursors und schnelle Cursorbewegungen auf der Displaywand erleichtert. Das relative Mapping verwendet einen nicht-isomorphen Ansatz, welcher das natürliche Handzittern visuell reduziert. Der Wechsel zwischen absoluten und relativen Mapping erfolgt durch eine Nutzeraktion (Selektion bzw. Deselektion), so dass der Zusammenhang zwischen manipuliertem Objekt und den langsameren virtuellen Bewegungen (bzw. erhöhtem physischem Bewegungsaufwand) erhalten bleibt. Mit der hybriden Cursorberechnung werden die kognitiven und perzeptiven Fähigkeiten der Nutzer unterstützt.

Wie schon erwähnt, kann der Nutzer feinfühlig Rotationsbewegungen durchführen, jedoch sind präzise Verschiebungen durch Hand- oder Armbewegungen schwieriger. Daher eignet sich die Verwendung unterschiedlicher Mappingfakto-

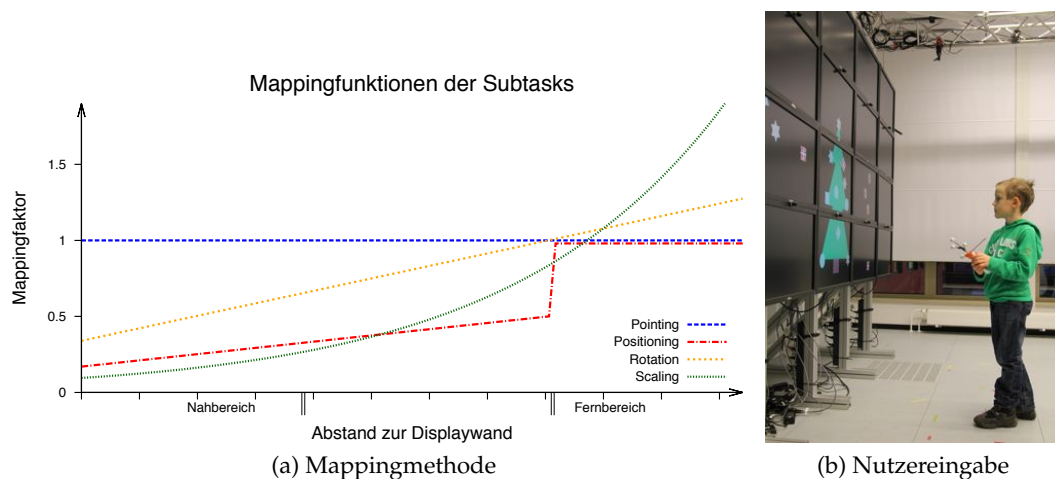


Abbildung 3.3.12.: Kombinierte Mappingfunktionen für die distanzabhängige Interaktion bei absoluter Nutzereingabe und Kindern als Nutzer.

ren für Positionierung und Rotation. Die Mappingfaktoren können global für alle Nutzer oder lokal für jeden Nutzer individuell vordefiniert oder distanzabhängig berechnet werden. Werden globale oder lokale vordefinierte Mappingfaktoren verwendet, müssen diese für jede Anwendung und Nutzergruppe in der GHD-Umgebung angepasst werden. Aufgrund der perspektivischen Projektion kann der Nutzer alle Objekte von einer beliebigen festen Betrachtungsposition aus erreichen. Lokale kontinuierliche Mappingfunktionen ermöglichen es, ein Intervall für die Mappingfaktoren zu bestimmen, so dass jeder Nutzer mittels physischer Navigation seinen Mappingfaktor individuell anpassen kann.

Eine distanzabhängige Berechnung des Mappingfaktors ist nur im Nahbereich des physischen Interaktionsraumes sinnvoll, wo eine hohe Präzision wahrnehmbar ist. Würde in der Entfernung ein kleiner Mappingfaktor angewendet werden, würden die Effekte von Clutching und Drift schneller für den Nutzer sichtbar.

In Abbildung 3.3.12a ist die Kombination der Mappingfunktionen für die vier Manipulationsaufgaben illustriert. Für die groben und schnellen Cursorbewegungen auf der Displaywand (*Pointing*) wird ein absolutes Mapping verwendet. Die Objektplatzierung (*Positioning*) wird durch den Tremor stark beeinflusst, weshalb ein kleiner Mappingfaktor angewendet wird. Dies gilt ebenfalls für die Objekskalierung (*Scaling*). Für die Objektdrehung (*Rotation*) wird ein größerer Mappingfaktor berechnet, da präzise, aber eingeschränkte Rotationsbewegungen möglich sind.

Die Umsetzung der Interaktionstechnik ist in Abschnitt 5.2 beschrieben und wurde in einer Pilotstudie mit fünf Kindern getestet (siehe Abschnitt 6.1.2).

3.3.3. Manipulation mit beidhändiger Interaktion

In diesem Abschnitt wird Interaction Scaling als distanzabhängige Manipulation bei beidhändiger Nutzerinteraktion in einer GHD-Umgebung diskutiert. Die Auswahl der verfügbaren Präzision der Objektmanipulation erfolgt abhängig vom Abstand des Nutzers zur Displaywand.

Nach Guiard [Gui87] gibt es drei Klassen, wie Nutzer mit beiden Händen agieren (vgl. Abschnitt 2.1.2). Die einhändige Interaktion (vgl. Beispiel 3.3.2) und die beidhändige symmetrische und asymmetrische Interaktion (vgl. Beispiel 3.3.3). Guiard's Framework besagt unter anderem, dass mit der dominanten Hand präzise Interaktionen durchführbar sind, während die nicht-dominante Hand für grobe Manipulationen genutzt wird. Dieser Aspekt wird bei der asymmetrischen Interaktion berücksichtigt.

Das Beispielszenario soll eine millimetergenaue Manipulation von 2D Objekten unterstützen, weshalb für jede Hand eine getrackte Wiimote als Eingabegeräte in Kombination mit einer Ray-Casting Technik verwendet wird. Die Eingabegeräte liefern absolute Positionsdaten (6-DOF) im physischen Interaktionsraum, die auf Positionsänderungen im virtuellen Raum abgebildet werden. Um das sichtbare Handzittern auf den virtuellen Cursorbewegungen zu minimieren, wird eine orthogonale Projektion verwendet. Die orthogonale Projektion erfordert einen ausreichend großen physischen Interaktionsraum, indem sich der Nutzer frei bewegt, um alle Objekte auf der Displaywand zu erreichen (vgl. Abschnitt 3.1.4).

Die physische Navigation in GHD-Umgebungen soll in dem Beispielszenario explizit genutzt werden. Befindet sich der Nutzer nahe am Display kann er nur einen begrenzten Ausschnitt der Visualisierung, allerdings mit sehr hohem Detailgrad wahrnehmen. Somit eignet sich eine sehr präzise Interaktion mit den Daten im Nahbereich des physischen Interaktionsraumes. Im Fernbereich hingegen erfolgt die Wahrnehmung der Datenrepräsentation für den Nutzer schnell und im allgemeinen Kontext. Daher sollte die Interaktion aus der Entfernung schnell und mit grober Präzision erfolgen. Für das Mapping wird ein nicht-isomorpher Ansatz verwendet, um die unterschiedlichen Präzisionen der Manipulationen zu unterstützen.

Folgende Einschränkungen werden für die Einflussfaktoren vorgenommen:

- physische Umgebung: flache LCD-Displaywand mit Bezels, getrackter Controller als Eingabegerät
- virtuelle Umgebung: präzise Manipulation von 2D Objekten, Ray-Casting Technik
- Nutzerinteraktion: beidhändige Nutzereingabe, symmetrische oder asymmetrische Interaktion, physische Navigation
- Nutzeranzahl: ein bis zwei Nutzer im physischen Interaktionsraum

- Mapping: nicht-isomorphe Ansatz, *position control*, orthogonale Projektion

Im Folgenden wird eine Manipulationstechnik für die symmetrische und asymmetrische Interaktion vorgestellt. Die Interaktionstechniken bilden die Nutzereingabe unterschiedlich auf die Task-Parameter der Manipulationsoperationen ab. Aufgrund der Verwendung von indirekten Zeigegeräten (Wiimotes) werden visuelle Repräsentationen (Cursors) der Hände zur Interaktion in der virtuellen Umgebung dargestellt. Bei der symmetrischen Interaktion wird ein dritter virtueller Cursor zwischen den visuellen Repräsentationen der Eingabegeräte dargestellt (Mittelpunkt-Cursor), der zur Selektion der 2D Objekte verwendet wird.

Mit der symmetrischen Interaktion werden die Manipulationstasks (Selektion, Positionierung, Rotation, Skalierung) gleichzeitig ausgeführt ohne den Interaktionsmodus zu wechseln. Bei der asymmetrischen Interaktion werden mit der dominanten Hand die Manipulationsaufgaben Selektion und Positionierung durchgeführt. Durch einen Wechsel des Interaktionsmodus kann Skalierung und Rotation mit der nicht-dominanten Hand durchgeführt werden.

Die Task-Parameter werden für die Manipulationstechniken folgendermaßen festgelegt (vgl. Tabelle 2.4.3). Bei der Selektion kann der aktuelle Selektionspunkt (a) auf den Objektmittelpunkt verschoben oder (b) als nutzerspezifischen Position festgelegt werden. Zur Unterstützung der Hand-Augen-Koordination ist das Rotationszentrum durch den Selektionspunkt bestimmt, d.h. es wird (a) um den Objektmittelpunkt oder (b) um die nutzerspezifische Rotationsachse rotiert. Es wird eine uniforme Skalierung verwendet, wobei das Skalierungszentrum aufgrund der Bewegungsabläufe der Manipulationstechniken unterschiedlich definiert ist. Bei der symmetrischen Interaktion wird eine objektzentrierte Skalierung verwendet, während bei der asymmetrischen Interaktion (a) eine objektzentrierte oder (b) eine nutzerspezifische Skalierung anwendbar ist. Hier ist das Skalierungszentrum ebenfalls durch den Selektionspunkt definiert, um den mentalen Aufwand für den Nutzer gering zu halten. Die entsprechenden Subparameter (z.B. Skalierungsfaktor, Rotationswinkel) werden aus den Bewegungsabläufen berechnet und als kontinuierliche Positions- und Orientierungsänderungen auf die virtuellen Objekte abgebildet.

Die Modusänderungen (Task starten oder beenden, Interaktionsmodus wechseln) erfolgen per Tastendruck mit dem Daumen auf dem Eingabegerät (Selektionstaste, Manipulationstaste).

Symmetrische Interaktion Für die Selektion wird der virtuelle Mittelpunkt-Cursor auf das Objekt bewegt und die Selektionstaste der dominanten Hand gedrückt (Manipulationstask startet). Um ein selektiertes Objekt zu positionieren, müssen die virtuellen Cursors der Eingabegeräte simultan in die gewünschte Richtung bewegt werden. Durch Vergrößern oder Verkleinern des Abstandes zwischen den Händen bzw. Eingabegeräten kann das Objekt uniform vergrößert oder verkleinert werden. Für die Rotation werden die Eingabegeräte ähnlich dem Steuern ei-

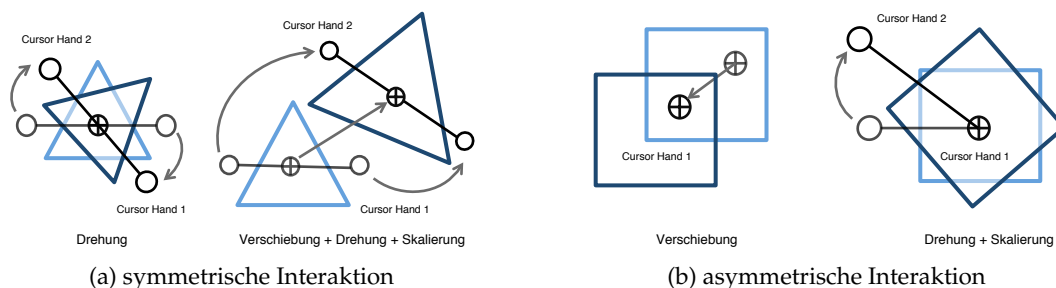


Abbildung 3.3.13.: Illustration der symmetrischen und asymmetrischen Manipulationstasks und der Trennung zwischen dominanter (*Cursor Hand 1*) und nicht-dominanter Hand (*Cursor Hand 2*).

nes Lenkrades bewegt (siehe Abbildung 3.3.13a). Wird die Selektionstaste losgelassen, wird das Objekt deselektiert und an der aktuellen Position des virtuellen Mittelpunkt-Cursors platziert.

Asymmetrische Interaktion Der virtuelle Cursor der dominanten Hand wird für die Selektion auf das Objekt bewegt und die Selektionstaste gedrückt. Das selektierte Objekt kann entsprechend der Bewegung der Hand verschoben werden. Wird die Selektionstaste losgelassen, wird das Objekt an der aktuellen Position platziert. Der Interaktionsmodus wird durch Drücken der Manipulationstaste mit der nicht-dominanten Hand gewechselt. Im SR-Modus (Objektskalierung und -rotation) wird der virtuelle Cursor der dominanten Hand fixiert und die Objektpositionierung deaktiviert. Das selektierte Objekt wird uniform verkleinert oder vergrößert, durch Verringern oder Vergrößern des Abstandes zwischen den Händen. Für die Objektrotation wird der Winkel zwischen den virtuellen Cursors verändert (siehe Abbildung 3.3.13b). Der SR-Modus wird beendet, sobald die Manipulationstaste losgelassen wird und der Nutzer kann das weiterhin selektierte Objekt verschieben. Wird die Selektionstaste im SR-Modus losgelassen, dann wird das Objekt deselektiert, an der aktuellen Position platziert und der SR-Modus abgeschaltet.

Interaction Scaling für symmetrische und asymmetrische Interaktion

Die Cursorberechnung ist abhängig davon welche Präzision der Interaktion in der virtuellen Umgebung gewünscht wird. Ist eine Genauigkeit im cm-Bereich erforderlich, eignet sich die absolute Cursorberechnung (schnell, geringer physischer Aufwand, keine Drift). Wird eine hohe Präzision, z.B. im mm-Bereich benötigt, muss eine relative Cursorberechnung verwendet werden (präzise, mehr physischer Aufwand, Drift-Effekt). Eine hybride Cursorberechnung eignet sich, wenn grobe und präzise Manipulationsoperationen erforderlich sind.

Bei der asymmetrischen Interaktion wird beispielsweise ein absolutes Mapping für die dominante Hand verwendet, weil mit der dominanten Hand feingesteuerte Manipulationsoperationen möglich sind (Selektion, Positionierung). Hingegen er-

fordert die Manipulation mit der nicht-dominanten Hand eine Unterstützung mittels relativem Mapping (Rotation, Skalierung). Bei der symmetrischen Interaktion ist beispielsweise eine hybride Cursorberechnung anwendbar. Das heißt, für die Selektion wird ein absolutes Mapping verwendet (grobe, schnelle Bewegungen) und für die präzisen Manipulationsoperationen (Positionierung, Rotation, Skalierung) wird ein relatives Mapping verwendet.

Da die physische Navigation für die distanzabhängige Präzision der Interaktion genutzt wird, können kontinuierliche, vordefinierte oder hybride Mappingfunktionen verwendet werden (vgl. Abbildung 3.2.7).

Die Zonen Mappingfunktion eignet sich wenn feste Präzisionen in der virtuellen Umgebung benötigt werden, z.B. nahe am Display ist eine Genauigkeit von ± 2 mm erforderlich und in mittlerer Distanz eine Genauigkeit von ± 5 cm. Bei einer kontinuierlichen Mappingfunktion kann jeder Nutzer seine Präzision innerhalb der Grenzen des Mappingfaktors durch seine physische Navigation steuern. Wobei es für den Nutzer nachvollziehbar sein muss, wie die Präzision der Interaktion ab- oder zunimmt. Die hybride Mappingfunktion kann die visuelle Wahrnehmung für die Anpassung der Präzision nutzen. Im Nahbereich kann der Nutzer individuelle Pixel unterscheiden, so kann beispielsweise eine pixelgenaue Selektion mittels eines sehr kleinen Mappingfaktors unterstützt werden. Im Fernbereich verschmelzen die Pixel oder visuellen Objekte für den Nutzer und die Genauigkeit wird entsprechend der Objektwahrnehmung festgelegt. Im Zwischenbereich kann der Nutzer den Mappingfaktor kontinuierlich variieren.

Die aktuelle Nutzerdistanz kann über den gemittelten Abstand der Hände oder über den Körperabstand zum Display bestimmt werden. Wird der Abstand der Hände verwendet, ist keine weitere Eingabemodalität nötig. Jedoch stimmt die Abstandsberechnung nicht mit der visuellen Wahrnehmung überein und eine Anpassung der Präzision erfolgt über die Motorik des Nutzers, d.h. die Bewegung der Arme relativ zum Oberkörper verändern den Mappingfaktor. Eine bessere Abstandsbestimmung ermöglicht das Kopf-Tracking, dass aber eine zusätzliche Eingabemodalität erfordert. In diesem Fall erfolgt eine Anpassung der Präzision durch die Mobilität des Nutzers.

In Nutzerstudien wurde Interaction Scaling für die symmetrische und asymmetrische Manipulationstechnik untersucht (siehe Abschnitt 6.1.1 und 6.2). Dabei wurden die beschriebenen Mappingmethoden für unterschiedliche Präzisionen der Interaktion (von 2,5 cm bis 1 mm) evaluiert. Weiterhin wurde untersucht, wie sich die subjektiven Eigenschaften der Probanden (z.B. Präferenz für symmetrische oder asymmetrische Interaktion, Tremor, bevorzugte Interaktionsdistanz) auf die Manipulationstechniken auswirkt.

Die beschriebenen Manipulationstechniken und die Pilotstudien wurden in [LS12] und [LS13] veröffentlicht.

3.4. Zusammenfassung

In großen hochauflösenden Displayumgebungen soll der Nutzer in seiner gegenwärtigen Tätigkeit unterstützt werden, indem Visualisierung und Interaktion an ihn angepasst werden. In GHD-Umgebungen werden verschiedene Aufgaben nah oder entfernt vom Display ausgeführt, die oftmals eine unterschiedliche Präzision und Sensitivität der Nutzereingabe erfordern. Das vorgestellte Interaction Scaling Konzept ermöglicht, den aktuellen Betrachtungsabstand des Nutzers zum Display für den Wechsel zwischen den Aufgabenbereichen zu nutzen. Dazu wird abhängig von der Interaktionsaufgabe (Navigation oder Manipulation) und den verfügbaren Freiheitsgraden der Eingabemodalität ein distanzabhängiges Mapping von physischen Bewegungen (Körper, Gerät) im realen Interaktionsraum auf Bewegungen (Cursor, Detailgrad) im virtuellen Raum angewendet.

Für die Interaktion in GHD-Umgebungen wurde der Parameterraum in die Komponenten physische Umgebung, virtuelle Umgebung, Benutzer und Mapping mit ihren Parametern aufgeschlüsselt. Des Weiteren wurde aufgezeigt, wie sich die einzelnen Parameter beeinflussen. Anhand von Beispielszenarien wurden die Kombination und Gewichtung ausgewählter Einflussfaktoren erläutert. Zusammenfassend ergeben sich folgende Schlussfolgerungen für die Mappingmethode (siehe Tabelle 3.4.5).

Interaction Scaling (IS) wird für virtuelle Umgebungen mit unterschiedlichen virtuellen Inhalten (2D und 3D) angewendet. Die Nutzereingabe erfolgt über die Kopfposition und -orientierung; die Mobilität und die visuelle Wahrnehmung des Nutzers in der GHD-Umgebung werden berücksichtigt. Auf diese Weise erfolgt IS als distanzabhängige Informationsdarstellung einerseits mit Hilfe von festen vordefinierten Mappingfaktoren für die Skalierung der Textlabels und der Auswahl der 2D Datenabstraktion und andererseits durch die Verwendung von kontinuierlichen Mappingfaktoren für die Berechnung der Fehlermetrik beim LOD-Rendering von 3D Geländedaten. Bei der 2D Graph-Exploration können mit Hilfe der Linsen-Technik unterschiedliche Präzisionen der Interaktion gleichzeitig genutzt werden, die lokale detaillierte Informationsdarstellung innerhalb der Linse (Fokus) und die globale abstrahierte Informationsdarstellung außerhalb der Linse (Kontext).

Bei der einhändigen Interaktion wird IS in unterschiedlichen physischen Umgebungen (MD-Umgebung, GHD-Umgebung) angewendet. Bei der Nutzereingabe in der MD-Umgebung sollen mausähnliche Nutzeraktionen im Raum und am persönlichen Gerät des Nutzers unterstützt werden. Es werden relative Nutzereingaben verwendet, die eine taskabhängige (d.h. anwendungsabhängige und anwendungsunabhängige Manipulationsoperationen) und distanzabhängige (d.h. globale Interaktion im Raum oder lokale Interaktion am persönlichen Gerät) Präzision der Interaktion unterstützen durch vordefinierte Mappingfaktoren. Die einhändige Manipulation in der GHD-Umgebung konzentriert sich auf Kinder als Nutzer (d.h. kleine Körpergröße, einfache Nutzereingabe). Somit wird IS für eine abso-

lute Nutzereingabe verwendet, bei der unterschiedliche Mappingfunktionen für die einzelnen Manipulationsaufgaben, abhängig von den Koordinationsmöglichkeiten der Finger, Hände und Arme, angewendet wird. Aufgrund der Nutzergruppe wird perspektivische Projektion mit hybrider Cursorberechnung in der Mappingmethode empfohlen, wobei das natürliche Handzittern teilweise sichtbar ist und der Drift-Effekt nicht zu vermeiden.

Bei der beidhändigen Interaktion wird IS für unterschiedliche motorische Koordinationsbewegungen (symmetrische und asymmetrische Interaktion) verwendet. In diesem Szenario soll das natürliche Handzittern für den Nutzer nicht sichtbar sein und die physische Navigation die präzisen Manipulationsoperationen unterstützen. Für die Mappingmethode wurde deshalb eine orthogonale Projektion verwendet und die Präzision der Interaktion durch kontinuierliche und hybride Mappingfunktionen an die Besonderheiten von Nah- und Fernbereich im physischen Interaktionsraum angepasst. Das heißt, es wird eine pixelgenaue Manipulation nahe am Display und schnelle ungenaue Manipulationen aus der Entfernung unterstützt.

In Pilotstudien wurde die distanzabhängige Interaktion ausgewählter Beispielszenarien mit Nutzern getestet. Eine umfassende Evaluierung des Interaction Scaling Ansatzes wurde am Beispiel der beidhändigen Interaktion durchgeführt (siehe Kapitel 6).

Zunächst werden in Kapitel 4 die verwendete Experimentalumgebung, einschließlich der verwendeten Setups vorgestellt und ausgewählte Details zur Umsetzung der Beispielszenarien beschrieben.

	<i>Head-Tracking</i>		<i>einhändige Interaktion</i>		<i>beidhändige Interaktion</i>	
	2D-Anwendung	3D-Anwendung	relative Nutzereingabe	absolute Nutzereingabe	symmetrische Interaktion	asymmetrische Interaktion
Physische Umgebung	Displaykonfiguration	GHD	MD	GHD	GHD	GHD
	Eingabemodalität (Eingabegerät)	Head-Tracker	Controller	getrackter Controller	getrackter Controller	getrackter Controller
	- Eingabeart	absolut	relativ	absolut	absolut	absolut
	- Freiheitsgrade	6-DOF	3-DOF	6-DOF	3-DOF	3-DOF
Virtuelle Umgebung	Informationsdarstellung	2D Objekte	2D Objekte	2D Objekte	2D Objekte	2D Objekte
	- Detailgrad	variierend	konstant	konstant	konstant	konstant
	Interaktionstasks	Selektion und Navigation	Manipulation	Manipulation	Manipulation	Manipulation
	Interaktionstechnik	Head-Tracking Fokus+Kontext	Maus-Adaption	Ray-Casting	Ray-Casting	Ray-Casting
Benutzer	objektive Eigenschaften	foveales und peripheres Sehen	Hand-Augen-Koordination	Hand-Augen-Koordination	symmetrische Armbewegung	asymmetrische Armbewegung
	- physische Navigation	Mobilität	Motorik	Motorik	Mobilität und Motorik	Mobilität und Motorik
	subjektive Eigenschaften (Nutzergruppe, Temor)		Erwachsene	Kinder (kleine Körpergröße)	Handzittern minimieren	Handzittern minimieren
	Nutzeranzahl	2	>2	2	1	1
Interaction Scaling	Ergonomie/Nutzereingabe	Kopfausrichtung	einhändig	einhändig	beidhändig, symmetrisch	beidhändig, asymmetrisch
	Informationsdarstellung/Präzision der Interaktion	distanzabhängige darstellung	task- und distanzabhängige Präzision der Manipulation	distanzabhängige Präzision der Manipulation	distanzabhängige Präzision der Manipulation	distanzabhängige Präzision der Manipulation
	Isomorphie	nicht-isomorph	nicht-isomorph	nicht-isomorph	nicht-isomorph	nicht-isomorph
	Projektion	perspektivisch	keine	perspektivisch	orthogonal	orthogonal
Mapping	Kinematik	position control	rate control	position control	position control	position control
	Cursorberechnung	keine, absolute (Linse)	relative	hybride	absolute, relative	hybride
	Mappingfunktion/-faktor	vordefinierte	vordefinierte	kombinierte	kontinuierliche, hybride	kontinuierliche, hybride

Tabelle 3.4.5.: Kombinationen der Einflussfaktoren in den Beispielszenarien und die resultierenden Parameter für die Mappingmethode. Die gewichteten Faktoren sind fett hervorgehoben.

4. Experimentalumgebung

Die Experimentalumgebung im Labor Visual Computing (VC-Lab) der Universität Rostock umfasst eine große hochauflösende Displaywand mit unterschiedlichen Sensortechnologien, die um und in der Displaywand installiert sind. Die hochauflösende Displaywand besteht aus 24 Breitbildmonitoren (Dell 2709W), die in einem 6 × 4 Gitter zu einer großen Anzeigefläche angeordnet sind. Jedes Display hat eine Auflösung von 1920 × 1200 Pixel und die Displaywand bildet somit eine Gesamtauflösung von 11520 × 4800 Pixel (55 Millionen Pixel). Die Displaywand wird mit Hilfe eines PC-Clusters gesteuert, indem jeweils vier Displays von einem Rendering-PC (*compute node*) angesteuert werden. In jedem *compute node* sind zwei Grafikkarten vom Typ Nvidia GeForce GTX 260 integriert, die jeweils zwei Displays ansteuern. Die sechs *compute nodes* und ein *head node* sind über ein Gigabit Ethernet zu einem Cluster verbunden. Der *head node* empfängt die Benutzereingaben und verteilt diese an das PC-Cluster.

Zum Detektieren von Benutzern und Geräten im physischen Interaktionsraum vor der Displaywand ist ein Infrarot-Trackingsystem der Firma NaturalPoint mit 12 Kameras installiert. Das verwendete Trackingsystem ermöglicht infrarot-reflektierende Marker zu erkennen. Mit Hilfe von Marker-Konstellationen, sogenannte *Trackables* oder *rigid bodies*, lassen sich die Position und Orientierung der Eingabegeräte oder Personen millimetergenau in Echtzeit detektieren (vgl. Abbildung 3.2.6).

Um den ungenutzten Platz zwischen den Bezels zu verwenden, sind in die Displaywand mehrere Mini-Kameras (PointGrey DragonFly2 mit einer optischen Auflösung von 150 lp/mm) integriert. Ebenso sind eine Reihe von PointGrey Flea2 Kameras um den physischen Interaktionsraum angebracht. Diese Videokameras ermöglichen es, Nutzer vor der Displaywand mit hoher Pixelauflösung zu filmen (vgl. Willert et al. [WOLS10]).

4.1. Physischer Interaktionsraum

Wie im Konzept beschrieben wird der physische Interaktionsraum durch die Displayauflösung, die Displayanordnung, die Displayfläche und die physiologische Wahrnehmung der Nutzer beeinflusst (vgl. Abschnitt 3.1.1). Die Eigenschaften der Displaykonfiguration der LCD-Wand¹ sind in Tabelle 4.1.1 zusammengefasst. Im Interaktionsraum wird zwischen **Nahbereich**, d.h. einzelne Pixel sind wahrnehmbar, und dem **Fernbereich**, d.h. die gesamte Displayfläche wird wahrgenommen, unterschieden. In einigen Fällen (z.B. Darstellung von Schriften oder kleinen Objekten) wird außerdem die **Schwellldistanz** berücksichtigt, bei der die Pixel, abhängig von der verfügbaren Displayauflösung, für den Nutzer miteinander verschmelzen.

Für die Displaywand im VC-Lab ergeben sich daraus folgende Betrachtungsabstände eines Nutzers (mit 100% Sehstärke, Sehwinkel 1/60 Grad) für den Nahbereich (nach Gleichung 3.1.1), den Fernbereich (nach Gleichung 3.1.2) und die Schwellldistanz.

$$\text{Nahbereich} = \frac{0,30 \text{ mm}}{2 \tan\left(\frac{1}{120}\right)} = 1031 \text{ mm} = 103 \text{ cm} \quad (4.1.1)$$

$$\text{Schwellldistanz} = \frac{2 * 0,30 \text{ mm}}{2 \tan\left(\frac{1}{120}\right)} = 2063 \text{ mm} = 206 \text{ cm} \quad (4.1.2)$$

$$\text{Fernbereich} = \sqrt{\frac{(3,80 \text{ m})^2}{2 * (1 - \cos(120))}} = 2,20 \text{ m} = 220 \text{ cm} \quad (4.1.3)$$

Das heißt, der Nutzer sollte maximal 1,00 m (41 inch) Abstand zur Displaywand haben, um noch einzelne Pixel und somit Informationen mit sehr hohem Detailgrad wahrnehmen zu können. Bei einem Betrachtungsabstand von etwa 2,10 m (81 inch) verschmelzen die Pixel für den Nutzer miteinander. Für den Informationsüberblick kann die gesamte Displaywand bei einem Betrachtungsabstand von etwa 2,20 m (86 inch) vom Nutzer wahrgenommen werden. Basierend auf diesen Betrachtungsabständen wurden Entscheidungen für die Nutzerstudien getroffen (z.B. Schwellwerte, Zonenbereiche).

Displayeigenschaft	LCD (Dell2709W)	LCD-Wand mit Bezel
Anordnung	Einzeldisplay	6 x 4 Matrix (24 LCDs)
Ausdehnung (BxH)	24,88 x 17,80 inch (63,20 x 45,20 cm)	149,3 x 71,20 inch (380 x 180 cm)
Auflösung	1920 x 1200 px	11520 x 4800 px (55 Mio. Pixel)
Anzeigefläche	22,90 x 14,31 inch (58,20 x 36,35 cm)	137,40 x 57,24 inch (349 x 145,40 cm)
Diagonale	27 inch	149 inch
Breite Bezel	1 inch	2 inch (benachbarte LCDs)
Pixel pro Inch (ppi)	83,86 px/in (33 px/cm)	
Pixelabstand (pp)	0,30 mm	

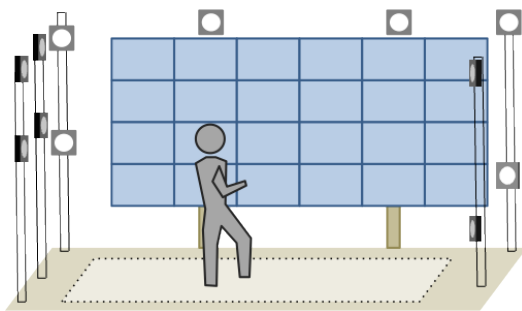
Tabelle 4.1.1.: Displaykonfiguration der verwendeten Displaywand im VC-Lab.

¹ Die tatsächliche Ausdehnung der Displaywand schwankt und verschiebt sich, aufgrund der Wärmeentwicklung der Displays minimal über die Zeit.

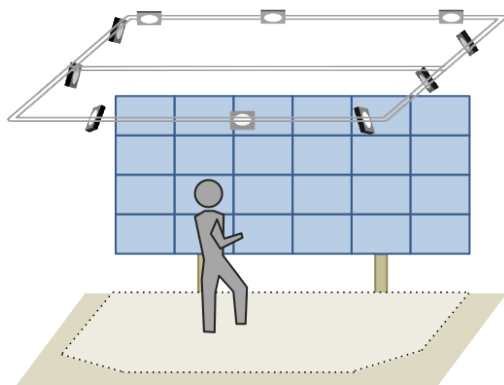
4.2. Trackingbereich

Die Abbildung 4.2.1 zeigt das Setup der Displaywand mit dem Trackingsystem, welches für die Umsetzung der Beispielszenarien verwendet wurde. Im ersten Setup, im Altbau wurden die Infrarot-Kameras (IR-Kameras) im Halbkreis an Haltestangen befestigt. Hier konnten Personen und Geräte ganz nahe an der Displaywand nicht detektiert werden, weil das Trackingvolumen die Displayfläche nicht erfassen konnte. Der Trackingbereich ist in Abbildung 4.2.1a auf dem Fußboden durch gelbe Klebestreifen markiert.

Daher wurde bei dem Umzug des VC-Labs in die neuen Laborräume im August 2011 das Trackingsystem neu ausgerichtet. Das hängende Traversensystem, an welchem die IR-Kameras montiert wurden, ermöglicht nun ein Trackingvolumen, welches die komplette LCD-Wand abdeckt, so dass der Nutzer bis an die Displays herantreten kann (siehe Abbildung 4.2.1b). Beim Ausrichten der IR-Kameras ließen sich die rechte und linke Ecke, entfernt von der Displaywand, nicht komplett abdecken, hier werden teilweise die Marker nicht zuverlässig erkannt.



(a) Trackingbereich mit IR-Kameras an Haltestangen



(b) Trackingbereich mit IR-Kameras am hängenden Traversensystem



Abbildung 4.2.1.: Illustration des Setups der Displaywand mit Trackingbereich im (a) Altbau und (b) Neubau.

4.3. Software Frameworks

Das PC-Cluster im VC-Lab wird mit dem Linux Betriebssystem Rocks² Version 5.2. (Setup im Altbau) bzw. Version 5.4. (Setup im Neubau) betrieben.

Die kommerzielle Trackingsoftware (NaturalPoint OptiTrack Tracking Tools³) läuft auf einem Tracking-PC mit Windows XP und verwendet synchronisierte IR-Kameras, die per USB mit dem Tracking-PC verbunden sind. Die Trackingsoftware berechnet die Position und Orientierung des Trackable im physischen Interaktionsraum (6-DOF), wobei die Positionsdaten in Millimetern und die Orientierung als Eulersche Winkel angegeben werden.

Für das Streaming der Trackingdaten von der Trackingsoftware zur Anwendung im PC-Cluster wird die Bibliothek *Virtual Reality Peripheral Network* (VRPN⁴) verwendet, die einen Server-Client-Mechanismus bietet [THS⁺01]. Die Trackingdaten werden als 3D Positionsdaten in Metern und die Rotation als Quaternion mit Hilfe des integrierten VRPN-Servers in das lokale Ethernet des VC-Labs übertragen. Über einen VRPN-Client können die Trackingdaten empfangen werden. Dieser Client läuft auf einem anderen PC im lokalen Netzwerk, in diesem Fall der *head node*.

Neben dem Tracking von Geräten und Personen als Nutzereingabe stehen ebenfalls zwei Nintendo Wii Remote Controller als Eingabegeräte zur Verfügung. Die Button-Events der Wiimotes werden per Bluetooth⁵ an den *head node* übertragen. Zur Bestimmung der Position und Orientierung der Wiimotes wird das IR-Trackingsystem genutzt und nicht die gerätespezifischen Funktionen (Beschleunigungssensoren, integrierte IR-Kamera), da diese zu ungenau für präzise Interaktionsaufgaben sind. Für das Beispielszenario der relativen Nutzereingabe können die Beschleunigungssensoren für die Cursorsteuerung genutzt werden, da eine größere Genauigkeit für die anwendungsunabhängigen Manipulationsoperationen verwendet wird (siehe Abschnitt 3.3.2).

Die softwarespezifischen Eigenschaften von PC-Clustern für die Ansteuerung von Displaywänden wird in Ni et al. [NSS⁺06] beschrieben. Unser Cluster verwendet ein Master-Slave-Prinzip für das Rendering, d.h. auf jedem *compute node* wird eine Instanz der Anwendung ausgeführt, der *head node* bearbeitet alle Benutzerinteraktionen und synchronisiert die Änderungszustände zwischen allen PCs.

Eine Übersicht der verschiedenen Kommunikations- und Datenkanäle der einzelnen Geräte im VC-Lab ist in Abbildung 4.3.2 dargestellt.

² <http://www.rocksclusters.org>

³ <http://www.naturalpoint.com/optitrack/products/tracking-tools/>

⁴ <http://www.cs.unc.edu/Research/vrpn/>

⁵ BlueZ package for linux: <http://www.bluez.org/>

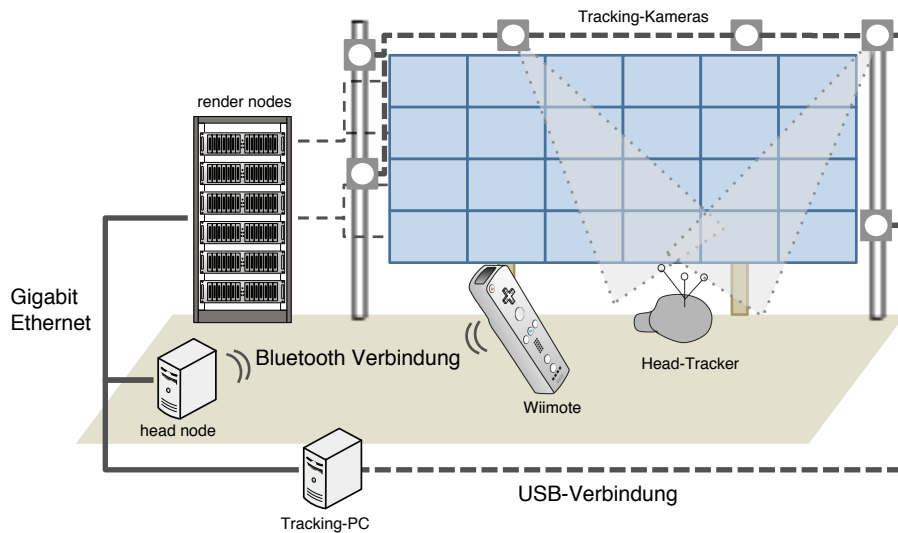


Abbildung 4.3.2.: Schematische Darstellung der Architektur im VC-Lab.

Rendering Frameworks

Für die Umsetzung der Beispielszenarien in der GHD-Umgebung wurden die C++ Frameworks *Cross-Platform Cluster Graphics Library* (kurz *cglX*⁶) [DK11] und *Vrui VR Toolkit* (kurz *Vrui*⁷) [Kre08] verwendet, welche die OpenGL Grafikbibliothek nutzen.

Um eine plausible 2D oder 3D Darstellung für den Nutzer auf der großen Displaywand zu erzeugen, ergänzen die Frameworks, basierend auf der Displaykonfiguration zusätzlich virtuellen Raum hinter die Bezels (siehe Abschnitt 3.1.2).

Vrui hat den Vorteil, dass nur eine Anwendung für verschiedene VR Setups (z.B. Desktop, CAVE, LCD-Wand) programmiert werden muss, da sämtliche Eingabeparameter (z.B. Interaktionsgeräte) und Ausgabeparameter (z.B. Displaykonfiguration) über eine Konfigurationsdatei beim Anwendungsstart eingelesen und konfiguriert werden. Das Framework verwendet einen sogenannten *tool layer*, der Parameter vom Eingabegerät auf semantische Events höherer Ebene (z.B. *selection*, *dragging*, *location*, *menu selection*) abbildet. Somit können die Eingabegeräte zur Laufzeit verändert werden. Das *Vrui* Framework unterstützt einen integrierten VRPN-Client und Wiimote-Tracker als abstrakte Eingabegeräte.

Das Framework *cglX* ermöglicht eine einfache Portierung von bestehenden C++/OpenGL Anwendungen auf die Displaywand, mit wenigen Änderungen am Programmcode. Jedoch muss der VRPN-Client selber in das Framework integriert werden.

Um die erfassten Positionsdaten im physischen Interaktionsraum auf Bewegungen in der virtuellen Umgebung, muss zunächst das Koordinatensystem des Trackingsystems in das Koordinatensystem der virtuellen Umgebung transformiert wer-

⁶ <http://vis.ucsd.edu/~cglx/>

⁷ <http://idav.ucdavis.edu/~okreylos/ResDev/Vrui/index.html>

den. Die verwendeten Frameworks nutzen ein rechtshändiges Koordinatensystem, während die Trackingsoftware ein linkshändiges Koordinatensystem verwendet. Der Koordinatenursprung der virtuellen Umgebung wird in der Konfigurationsdatei der jeweiligen Frameworks definiert und befindet sich standardmäßig in der Mitte der Displaywand. Zur Vereinfachung wird bei der Kalibrierung der Koordinatenursprung des Trackingvolumens auf der Mitte der Displaywand festgelegt. Die transformierten Trackingdaten aus dem physischen Interaktionsraum in das Weltkoordinatensystem der virtuellen Umgebung werden im Weiteren als **physische Koordinaten** (bzw. physische Positionsdaten) bezeichnet.

Die Beispielszenarien wurden hauptsächlich mit Vrui umgesetzt, da es Probleme bei der Synchronisierung der Displays mit cglX gab und die Anwendungen zur distanzabhängigen Manipulation speziell für die Nutzerstudien entwickelt wurden. In der Konfigurationsdatei von Vrui wird das angloamerikanische Maßsystem für die Anwendungen verwendet, da die Angaben für die Hardwarekonfiguration der Displays in Zoll angegeben werden. Daher werden in dieser Arbeit die Maßangaben für die Parameter größtenteils im angloamerikanischen System (z.B. inch, feet) angegeben und in das metrische System umgerechnet.

Für die Umsetzung der Beispiele wurde cglX 2.0 und die Vrui Versionen 1.0-068 bis 2.3-003 verwendet.

Im Rahmen einer Bachelorarbeit⁸ wurde eine Latenzmessung (nach Steed [Ste08]) für das Positionstracking in der GHD-Umgebung mit beiden Frameworks durchgeführt. Dazu wurde eine Pendelbewegung im physischen Interaktionsraum getrackt und als 2D Pendelbewegung in der virtuellen Umgebung animiert, dabei unterschieden sich die Latenzen zwischen den Frameworks nicht. Das heißt, beide Frameworks sind für unsere Studien gleichermaßen geeignet.

⁸Alexander Kluwe "Methoden zur Bestimmung der Latenz in virtuellen Umgebungen", Bachelorarbeit an der Universität Rostock, Mai 2011.

5. Umsetzung der distanzabhängigen Interaktion

5.1. Distanzabhängige Informationsdarstellung

Nachfolgend wird die Umsetzung der beiden Szenarien aus Abschnitt 3.3.1 beschrieben und wie die distanzabhängige Informationsdarstellung bestimmt wurde.

5.1.1. Navigationstechniken für 2D Anwendung

Mit Hilfe der physischen Navigation soll ein geclusterter 2D Graph exploriert werden, indem die 3D Position des Nutzers vor der Displaywand (Head-Tracking) zur virtuellen Navigation in der 2D Anwendung verwendet wird. Durch Ändern der aktuellen Betrachtungsdistanz kann der Nutzer die dargestellte Abstraktionsebene interaktiv ändern. Als Beispiel eines hierarchisch geclusterter Graphen wird die Hierarchie des *ACM computing classification systems* (ACM-Graph) verwendet, in dem Knoten den Textlabels der Kategorien entsprechen und Kanten zwischen den Knoten verbundene Kategorien anzeigen. Der ACM-Graph umfasst vier Abstraktionsebenen mit insgesamt 1473 Knoten und 464 Kanten.

Folgende Eigenschaften werden bei der distanzabhängigen Informationsdarstellung berücksichtigt:

- Informationsdarstellung: distanzabhängige Auswahl der Abstraktionsebenen und Größe der virtuellen Linse (Fokusbereich); Berücksichtigung der Bezels (Verzerrung, Verdeckung)
- Interaktionstechnik und Interaktionsraum: Zonen- und Linsen-Technik; 4 Abstraktionsebenen und Interaktionsraum mit 3 m Tiefe
- kognitive und perzeptive Eigenschaften: Lesbarkeit der Textlabels (Schriftgröße, Positionierung); Animation der Übergänge zwischen Abstraktionsebenen; Visualisierung der Cluster (konvexe Hüllen); Reduzierung des Tremor (Median-Berechnung der virtuellen Linse)
- Mappingfunktion: Zonen-Mapping (5 Zonen mit Zonentiefe 55 cm und Schwellbereich 20 cm)

In der Anwendung wird ein vorberechnetes Layout des Graphen verwendet. Anschließend wird eine automatische lokale Anpassung der Position der Graphknoten berechnet, um die Platzierung von Knoten hinter den Bezels zu vermeiden.



(a) Zonen-Technik



(b) Linsen-Technik mit hervorgehobener Linse

Abbildung 5.1.1.: Navigationstechniken für die Exploration des ACM-Graphen.

Der Graph verwendet beschriftete Knoten (Textlabels). Somit müssen folgende Eigenschaften der Textlabels beachtet werden: Lesbarkeit aus unterschiedlichen Betrachtungsdistanzen, keine Überlagerung und keine Verzerrung durch Bezels. Für die Platzierung der Labels wurde ein existierender Labeling Algorithmus von Luboschik et al. [LSC08] verwendet. Der Algorithmus unterstützt dynamische Schriftgrößen und verwendet sogenannte Konfliktpartikel an den Stellen der Anzeigefläche, wo keine Labels angezeigt werden sollen. Wir haben solche Konfliktpartikel zusätzlich an den Kanten der Bezels im virtuellen Raum hinzugefügt.

Interaktionstechniken und Mappingmethode

Für die physische Navigation zur Exploration von Graphen wurden die im Konzept beschriebenen Navigationstechniken Zonen- und Linsen-Technik implementiert (vgl. Abschnitt 3.3.1). Beide Techniken erlauben dem Nutzer die dargestellte Abstraktionsebene des Graphen, indem er sich vor der Wand bewegt, zu ändern und somit den Informationsumfang anzupassen. Die Zonentechnik entspricht einer stufenweisen globalen Anpassung der Abstraktionslevel (siehe Abbildung 5.1.1a). Die Linsentechnik hingegen führt eine lokale Anpassung der Abstraktionsebene durch (siehe Abbildung 5.1.1b). Übergänge von einem Level zu einem anderen werden animiert, um die mentale Karte der Umgebung beziehungsweise des Graphen für den Nutzer zu erhalten.

Der physische Interaktionsraum ist durch das Trackingvolumen begrenzt, d.h. der minimale und maximale Betrachtungsabstand ist 80 cm bzw. 355 cm (Setup Altbau). Somit ergibt sich eine Interaktionsdistanz von 2,75 m. Der ACM-Graph verwendet vier Abstraktionsebenen, wobei die unterste Ebene (Abstraktionslevel 0) den höchsten Detailgrad repräsentiert, d.h. insgesamt fünf Hierarchielevel umfasst. Der Interaktionsraum wird für die Zonen Mappingfunktion in fünf Zonen mit einer Tiefe von 55 cm unterteilt. Weiterhin werden Schwellbereiche (Tiefe 20 cm) zwischen den Zonen eingefügt. Abbildung 5.1.2 illustriert die Betrachtungsabstände

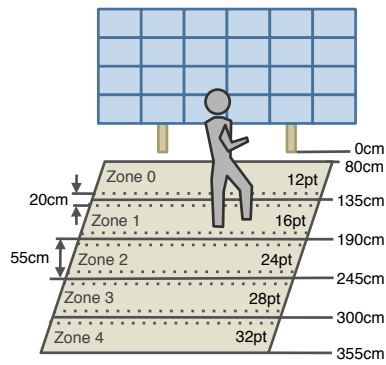


Abbildung 5.1.2.: Unterteilung des Interaktionsraumes für die Exploration des ACM-Graphen mit Zonen-Mapping und den zugehörigen Schriftgrößen.

für die einzelnen Zonen mit *Zone 0* für die unterste Ebene und *Zone 4* die höchste Abstraktionsebene des ACM-Graphen.

Für die Darstellung von Textlabels wird eine Zeichenhöhe von 20 Bogenminuten empfohlen (vgl. Abschnitt 2.1.1). Mit Hilfe der Gleichung 2.1.1 kann die Zeichenhöhe h der Textlabels für die einzelnen Zonen wie folgt berechnet werden, wobei die Schriftgröße s in Didot-Punkt¹ (pt) angegeben wird:

$$h = 2 \tan\left(\frac{\theta}{2}\right) * d$$

$$h = 2 \tan\left(\frac{20}{120}\right) * 800 \text{ mm} = 4,7 \text{ mm}$$

$$s = \frac{4,7}{0,376} [\text{pt}] = 12 \text{ pt}$$

Somit ergibt sich eine Schriftgröße von 12 pt im Nahbereich und bei einem Betrachtungsabstand von 3 m müsste die Schriftgröße 46 pt betragen. Um die Labels nahe an ihren korrespondierenden Knoten darzustellen, werden kleinere Schriftgrößen, die ebenso aus der Entfernung lesbar sind, verwendet. Für die Schriftdarstellung wird *Font Arial* im Bereich 12 pt bis 32 pt verwendet (vgl. Abbildung 5.1.2). Die Schriftgröße wurde für die jeweiligen Zonen festgelegt, je näher der Nutzer an die Displaywand herantritt, desto mehr Informationen werden angezeigt und die Schriftgröße der zugehörigen Textlabels verringert (vgl. Abbildung 5.1.3).

Für die Linsen-Technik wird außerdem die Größe der Linse anhand der aktuellen Nutzerdistanz angepasst, um den Umfang der dargestellten Informationen an den Fokusbereich anzugleichen. Aufgrund der visuellen Linse ist das natürliche Zittern des Nutzers wahrnehmbar, dass mit Hilfe eines Medianfilters der letzten 10 Positionsdaten reduziert wird.

Beim Mehrbenutzer-Szenario kann die Darstellung der Textlabels für zwei oder mehr Abstraktionsebenen, aufgrund der unterschiedlichen Schriftgrößen und An-

¹ 1 Didot-Punkt entspricht 0,376 mm (nach [DIN99a, DIN 16507-2:1999]).

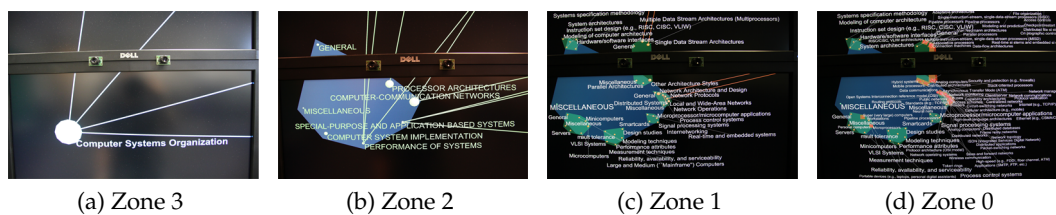


Abbildung 5.1.3.: Anpassung der Schriftgröße abhängig von der Nutzerdistanz (aus [Ger11]).

zahl der Labels, für die Nutzer schnell unübersichtlich werden. Zur Unterstützung der kognitiven und perzeptiven Wahrnehmung werden zusätzliche Visualisierungstechniken (z.B. Farbkodierung, Clustervisualisierung) und eine angepasste Linsen-Technik verwendet. Dazu werden die detaillierten Informationen (Textlabels) nur innerhalb der visuellen Linsen angezeigt und der Kontext wird mittels farbkodierten Grafikprimitiven dargestellt (siehe Abbildung 5.1.4). Beim Überlappen der Fokusbereiche muss darauf geachtet werden, dass sich die individuellen Informationsdarstellungen nicht behindern, indem zuerst die Textlabels in der Linse mit der niedrigeren Abstraktionsebene bzw. Schriftgröße berechnet werden.

Für das Rendering wurde cgIX und für die Schriftdarstellung die Bibliothek FTGL² verwendet.



Abbildung 5.1.4.: Verwendung der Linsen-Technik für zwei Benutzer (aus [Ger11]).

Pilotstudie

In einer Pilotstudie haben acht Informatikstudenten (7 männliche und 1 weibliche Testperson) die Anwendung mit den zwei Interaktionstechniken getestet. Die Teilnehmer sollten den Graphen durch physische Navigation vor der Displaywand etwa 10 Minuten erkunden. Sie konnten während des Testlaufes zwischen den Interaktionstechniken durch Drehen des Head-Trackers (Basecap mit Markern) wechseln. Anschließend wurden sie bezüglich der Benutzerfreundlichkeit, der Lesbar-

² <http://sourceforge.net/projects/ftgl/>

keit der Beschriftungen aus unterschiedlicher Entfernung und ihrer bevorzugten Interaktionstechnik befragt. Alle Testpersonen beschrieben das Head-Tracking als einfach zu benutzen und die Beschriftungen waren stets lesbar. Die Teilnehmer empfanden die Zonentechnik verglichen mit der Linsentechnik als einfacher zu benutzen, da hier keine unerwünschten Wechsel zwischen den Abstraktionslevels entstanden. Die Zonentechnik wurde aber als weniger interaktiv bezeichnet und die Testpersonen empfanden, dass zuviel Information in ihrem peripheren Sehfeld sichtbar war.

Die Linsentechnik wurde als "Eyecatcher" beschrieben und dass sie interaktiver als die Zonentechnik ist. Die Testpersonen mochten es, dass sie detaillierte Informationen innerhalb der Linse sahen, während der Kontext außerhalb der Linse erhalten blieb. Jedoch verursachten kleine Kopfbewegungen bei der Linsentechnik Interaktionen (auf- und einklappen von Knoten), was die Testpersonen als störend empfanden. Zusätzlich benötigt die Technik für jeden Nutzer eine Kalibrierung des Head-Trackers. Eine detaillierte Beschreibung der Pilotstudie ist in [Ger11] zu finden.

5.1.2. Navigationstechnik für 3D Anwendung

Im Beispielszenario der 3D Geländedarstellung soll die physische Betrachtungsposition des Nutzers für das Rendering genutzt werden, indem der Detailgrad der virtuellen Objekte (LOD-Level) vom Abstand des Betrachters und der Position auf dem Display abhängt (siehe Abbildung 5.1.5). Für das Rendering des Geländes wurde eine Implementierung angelehnt an Löffler et al. [LSS10], ein Chunked LOD Algorithmus [Ulr02] und das Framework Vrui verwendet (Setup Neubau). Es wurden zwei 3D Terrains mit 4096 x 4096 bzw. 8096 x 8096 Datenpunkten umgesetzt, die eine Quadtree Datenstruktur nutzen.

Für das Rendering wurden folgende Eigenschaften festgelegt:

- Informationsdarstellung: distanzabhängige Auswahl des LOD für die Sichtzellen
- Wahrnehmung: foveale und periphere Sehen bei der LOD-Bestimmung berücksichtigt
- Displaykonfiguration: reguläre Unterteilung der Sichtzellen
- Nutzerinteraktion: Detailgrad nur an das Sichtfeld (Abstand, Blickrichtung) angepasst
- Mappingmethode: perspektivische Projektion, Sichtzellen im erweiterten Sehfeld

Aufgrund der Displaykonfiguration empfiehlt sich eine reguläre Unterteilung der Displayfläche für die Sichtzellen. Jedes Display wird als einzelne Sichtzelle be-



Abbildung 5.1.5.: Rendering der unterschiedlichen LOD-Levels eines 3D Geländes für einen Benutzer (links) und zwei Benutzer (rechts).

handelt und abhängig von der Betrachtungsposition des Nutzers (6-DOF) als aktiv oder inaktiv geschaltet. Bei inaktiven Displays wird ein schwarzes Bild gerendert.

Für die Bestimmung des Detailgrades der aktiven Sichtzellen wird zuerst die Sichtzelle bzw. das Display bestimmt, auf welches der Nutzer schaut, basierend auf der approximierten Kopforientierung (Bestimmung des Fokuspunktes mit Head-Tracker). Dazu wird der Winkel vom Fokuspunkt zu den anderen Sichtzellen (Mittelpunkt auf Display) bestimmt. Je größer der Winkel γ ist, desto größer ist der Bildraumfehler³ BF für die LOD-Bestimmung; mit zunehmenden Abstand vom Fokuspunkt erhöht sich der Bildraumfehler um 2. Die Bestimmung erfolgt in diskreten Schritten von 15 Grad: $BF = \lceil d \rceil * 2^{(\gamma \bmod 15)}$ [Gir13].

Abbildung 5.1.6 illustriert die Auswahl der Sichtzellen abhängig von der Nutzerposition: Der Fokusbereich ist als ovale Linse visualisiert und umfasst die Sichtzelle mit hohem Detailgrad; der Detailgrad der Sichtzellen nimmt mit zunehmender Entfernung vom Fokus ab (vgl. Abschnitt 3.3.1).

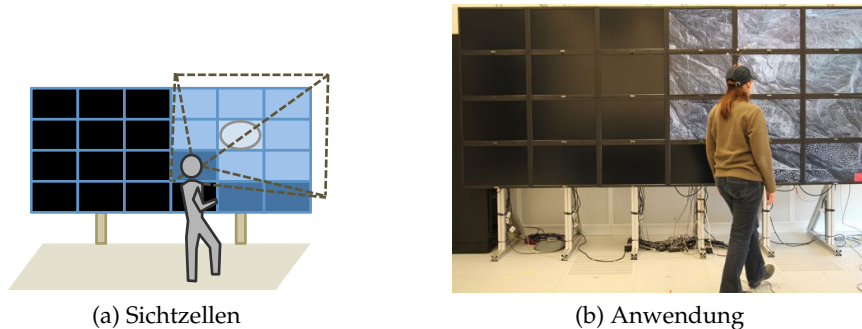


Abbildung 5.1.6.: Auswahl der Sichtzellen mit unterschiedlichem Detailgrad.

Bei mehreren Benutzern werden die LOD-Level für die Sichtzellen für jeden Benutzer berechnet und bei überlappenden Sichtzellen wird der höhere Detailgrad gerendert (siehe Abbildung 5.1.5). Die Bezels bieten eine natürliche Separation, so dass keine Lücken im Terrain aufgrund unterschiedlicher LOD-Level zwischen benachbarten Sichtzellen auftauchen.

³ Der Bildraumfehler bestimmt die Abweichung (in Pixeln) zwischen der Originalgeometrie und der dargestellten Approximation im Bildraum.

5.2. Distanzabhängige Präzision der Manipulation

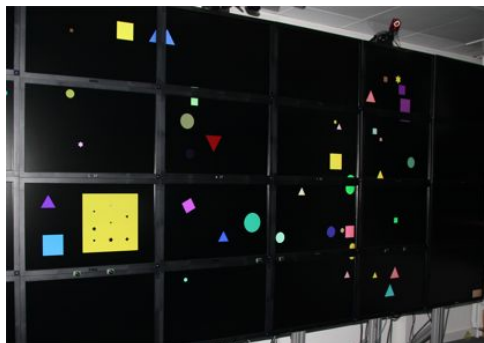
Der Fokus der Arbeit liegt auf der Anpassung der Interaktionspräzision für grundlegende Manipulationsaufgaben. Für die distanzabhängige Präzision der Interaktion wurden Ray-Casting Techniken für die einhändige und beidhändige Nutzereingabe umgesetzt. Die Interaktionstechniken sollen eine millimetergenaue Manipulation in der virtuellen Umgebung durch 3D Interaktion ermöglichen.

Für die Testanwendung wurde sich an einem Kinderspielzeug zum Training der motorischen Fähigkeiten bei Kleinkindern orientiert. Das Spielzeug motiviert Kinder die unterschiedlichen Bausteine in die Holzkiste einzusortieren, indem sie die passenden Formen auswählen und durch Drehen den Baustein in die zugehörige Form fallen lassen (siehe Abbildung 5.2.7a). Angelehnt daran sollen in der virtuellen Testanwendung unterschiedliche geometrische Figuren in die zugehörigen Silhouetten einsortiert werden (siehe Abbildung 5.2.7b). Ähnliche Sortieraufgaben wurden in verschiedenen Interaktionsstudien in VR-Umgebungen [KCHI07, HAM11] und GHD-Umgebungen [CHBL05, KSMB08, PNB09] verwendet.

Für die Nutzereingabe wurden die beschriebenen Manipulationstechniken aus Abschnitt 3.3.3 und 3.3.2 umgesetzt, mit Wiimotes als Eingabegeräte im physischen Interaktionsraum.



(a) reale Spielzeug



(b) virtuelle Testanwendung

Abbildung 5.2.7.: Übung zu feinmotorischer Manipulation.

Mit Hilfe der virtuellen Testanwendung werden ausgewählte Einflussfaktoren aus dem vorgestellten Parameterraum in Experimenten untersucht. Die Kombinationen der Einflussfaktoren für die umgesetzte einhändige und beidhändige Interaktion in GHD-Umgebungen sind in Tabelle 3.4.5 zusammengefasst. Die einfache Sortieraufgabe in der Testanwendung bietet die Möglichkeit, folgende Parameter zu variieren:

- virtuelle Umgebung: Interaktionstechnik und Task-Parameter, Präzision der Interaktion
- Mapping: Cursorberechnung, Mappingfunktion

Die Eigenschaften für die physische Umgebung sind durch die verwendeten 6-DOF Eingabegeräte (getrackte Wiimotes, Head-Tracker) und den physischen Interaktionsraum im VC-Lab (mit Trackingbereich Altbau oder Neubau) gegeben. In den Experimenten ist dabei von besonderem Interesse, wie der Benutzer aufgrund seiner subjektiven und objektiven Eigenschaften diese Parameter beeinflusst.

Die distanzabhängige Präzision der Manipulation wurde in zwei Pilotstudien (Studie 1, Studie 2) und einer umfassenden Studie (Studie 3) für die beidhändige 2D Manipulationstechnik und in einer Pilotstudie (Studie 4) für die einhändige Manipulationstechnik getestet (siehe Kapitel 6). Zwischen den Studien wurden die Task-Parameter der Interaktionsaufgabe, die Toleranzbereiche für die Sortierung und die Mappingfunktionen variiert, wobei Ergebnisse einer Studie in die Parameterfestlegung der nachfolgenden Studie einfließen. Die wichtigen Eigenschaften der vier Experimente sind in Tabelle 5.2.1 zusammengefasst.

In den Experimenten werden Synonyme für die beschriebenen 2D Manipulationstechniken verwendet:

- (1) beidhändige asymmetrische Interaktion: it1;
- (2) beidhändige symmetrische Interaktion: it2;
- (3) einhändige absolute Nutzereingabe: it3.

Parameter	Studie 1	Studie 2	Studie 3	Studie 4
Interaktionstechnik	it1, it2	it1	it1, it2	it3
Anzahl der Zielobjekte	3	8	8	9
Toleranzbereich (Größe, Position, Orientierung)	± 2,54 mm	± 1,5 mm (Größe), ± 2 mm (sonst)	± 2,54 mm (groß), ± 1 mm (klein) ¹	± 2,54 mm, ± 5 mm
Bestimmung der Nutzerdistanz	durchschnittlicher Abstand beider Wiimotes		Abstand Head-Tracker	Abstand der Wiimote
Trackingbereich (inkl. minD/maxD der Nutzerdistanz)	Altbau cm: [81; 230] inch: [32; 90]	Altbau cm: [51; 280] inch: [20; 110]	Neubau cm: [76; 280] inch: [30; 110]	Neubau cm: [51; 280] inch: [20; 110]
Cursorberechnung	relativ	hybrid	hybrid, absolut/relativ	hybrid
Mappingfunktion	linear, hybrid (linear)	linear, zone	linear, hybrid (exponentiell)	kombiniert, zone, linear, hybrid

¹ Studie 3 untersucht große und kleine Zielobjekte, die verschiedene Toleranzbereiche haben.

Tabelle 5.2.1.: Variation der Parameter für die Experimente mit den distanzabhängigen Manipulationstechniken.

5.2.1. Testanwendung

Für die Testanwendung wurden einfache geometrische 2D Primitiven (manipulierbare Objekte) und ein sogenannter Zielobjektcontainer (Targetcontainer oder Container) implementiert (siehe Abbildung 5.2.8). Als manipulierbare Objekte wurden die Primitiven Kreis, Quadrat, gleichseitiges Dreieck und sechseckiger Stern umgesetzt. Diese Objekte können selektiert, positioniert, gedreht und skaliert werden, während der Container nicht manipulierbar ist. Die Aufgabe ist die Objekte so zu manipulieren, dass sie in die zugehörigen Zielformen (Targets, Silhouetten) passen.

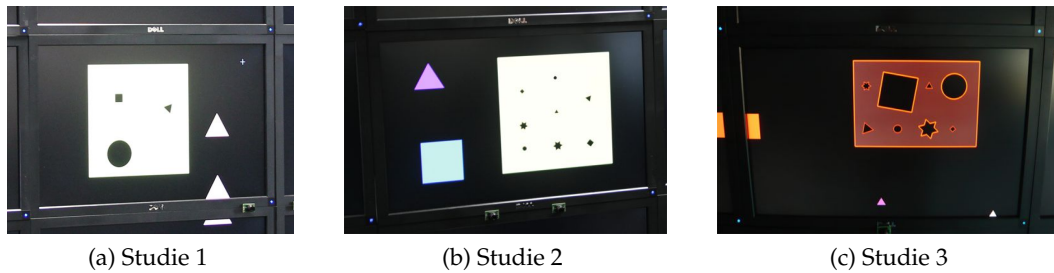


Abbildung 5.2.8.: Die verwendeten Targetcontainer in den Studien.

Es werden mehr manipulierbare Objekte als Targets dargestellt, damit der Nutzer sich zunächst einen Überblick über die Darstellung auf der Displaywand verschaffen muss. Die Objekte unterscheiden sich farblich gut (hoher Kontrast). Die Anzahl der Targets und zugehörigen Objekte ist identisch, so dass alle Objekte mit der Targetfarbe einsortiert werden müssen. Die initiale Positionierung der manipulierbaren Objekte erfolgt zufällig zum Programmstart, wobei kein Objekt hinter/über den Bezels und dem Container abgelegt wird⁴.

Die Überprüfung, ob ein Objekt korrekt einsortiert wurde, erfolgt erst wenn der Nutzer das manipulierte Objekt explizit über dem Target "loslässt", d.h. es deselektiert. Dann testet die Anwendung, ob die Bounding Box des Objektes den Container ganz oder teilweise schneidet; wenn dies der Fall ist, werden die Objekteigenschaften in folgender Reihenfolge getestet: Farbe, Form, Größe (Vergleich der Kantenlänge), Position (Vergleich des Objektmittelpunktes), Orientierung (Vergleich der Positionen der Eckpunkte). Hinsichtlich der Größe, Position und Orientierung kann das Objekt innerhalb eines definierten Toleranzbereiches abweichen. Die Toleranzwerte lagen in den Studien zwischen $\pm 2,54$ mm (0.1 inch) und ± 1 mm (0.04 inch). Ist das Objekt korrekt einsortiert, dann wird es grau eingefärbt (visuelles Feedback) und kann vom Nutzer nicht mehr selektiert werden. Objekte können nicht außerhalb der sichtbaren virtuellen Szene (Displayfläche) platziert werden. Wenn alle Objekte einsortiert sind, wird die Anwendung beendet.

Die Umsetzung der Testanwendung erfolgte mit dem Framework Vrui. Im Weltkoordinatensystem der virtuellen Umgebung entspricht die xz-Ebene der Displayfläche und die negative y-Achse dem Abstand zur Displaywand. Es wurden zwei Displaykonfigurationen für die Studien angelegt. In der Displaykonfiguration *wall with bezels* wird vom Framework virtueller Raum hinter den Bezels ergänzt. Damit die Objekte nicht hinter den Bezels verschwinden und unerreichbar werden, wird die minimale Objektgröße auf die doppelte Breite der Bezels (2 inch) beschränkt. In der Displaykonfiguration *wall without bezels* werden die Monitorkanten nicht in die Displayfläche integriert. In dieser Konfiguration kann die Objektgröße kleiner 2 inch sein und die Objekte werden über den Bezels verzerrt dargestellt. Die Ver-

⁴ Die initiale Positionierung der Objekte außerhalb des Targetcontainers funktionierte erst in der Studie 3 zuverlässig.

zerrung ist aufgrund der einfachen Aufgabenstellung für die Experimente jedoch nicht relevant. Die Studien 1, 2 und 3 verwenden die Displaykonfiguration *wall without bezels*.

Die Testanwendung wurde für die Studie 4 leicht abgeändert. Hier sollen Kinder die texturierten 2D Objekte (Weihnachtskugel, Lebkuchenstern, Geschenkbox) in die Platzhalter am Weihnachtsbaum (Zielcontainer mit 9 Targets) einsortieren. In der spielerischen Anwendung können zwei Spieler gleichzeitig jeweils einen Weihnachtsbaum schmücken (siehe Abbildung 5.2.9). Der linke Spieler (Wiimote 1) kann jedoch nur Objekte in den linken Weihnachtsbaum (Tree 1) einsortieren. Die Anzahl der manipulierbaren Objekte und Targets ist identisch. Die Studie 4 verwendet die Displaykonfiguration *wall with bezels*. Die Weihnachtsbäume wurden auf der Displaywall so angeordnet, dass eine optimale Erkennung beider Spieler im Trackingbereich möglich war.



Abbildung 5.2.9.: Spielerische Anwendung für das Experiment zur einhändigen Manipulationstechnik mit vergrößertem Zielcontainer (links).

5.2.2. Interaktionstechniken

Die Interaktionstechniken (it1, it2, it3) verwenden eine indirekte Nutzereingabe, weshalb zur Unterstützung der Hand-Augen-Koordination visuelle Cursor für jedes Eingabegerät dargestellt werden: ein visueller Cursor für die dominante Hand (C_{dH}) und für die nicht-dominante Hand (C_{ndH}), sowie ein Mittelpunkt-Cursor (C_{MP}) für die symmetrische Interaktionstechnik (siehe Abbildung 5.2.10).

Die Task-Parameter der Interaktionsaufgaben (Selektion, Positionierung, Rotation, Skalierung) wurden in den Experimenten folgendermaßen variiert (siehe Tabelle 5.2.2). Bei der einhändigen Interaktionstechnik erfolgen die Manipulationsoperationen ähnlich wie beim Holzspielzeug (Selektion, Positionierung, Rotation), es wird keine Skalierung unterstützt. Für die Selektion wurde in allen Studien die Entfernung und Richtung zum Zielobjekt durch Ray-Casting bestimmt, wobei Studie 4 eine perspektivische Projektion für den Selektionsstrahl verwendet und die anderen Studien eine orthogonale Projektion. Es wurde eine pixelgenaue Selektion umgesetzt, d.h. der Radius des Interaktionscursors entspricht einem Punkt ($S_r = 0$), der für die Selektionsentscheidung innerhalb des ausgewählten Objektes liegen muss. Es werden alle Objekte unter dem Cursor selektiert (Studie 1, Studie 2)

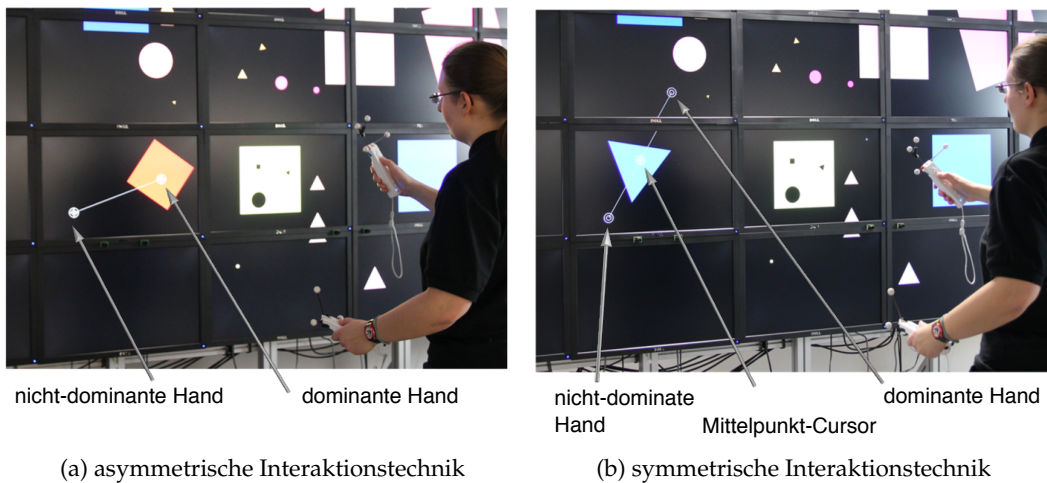


Abbildung 5.2.10.: Umsetzung der asymmetrischen und symmetrischen 2D Manipulationstechniken.

oder nur ein Objekt (Studie 3, Studie 4) und der Nutzer kann die Tiefe des Selektionsstrahls nicht bestimmen. Bei einer erfolgreichen Selektion wird der Cursor der dominanten Hand (it1, it3) bzw. der Mittelpunkt-Cursor (it2) in der Studie 3 und 4 in den Objektmittelpunkt verschoben, während in Studie 1 und 2 der Selektionspunkt nicht verschoben wird (nutzerspezifischer Selektionspunkt).

In der Testanwendung erfolgt die Aktualisierung der Positionsänderungen der physischen Eingabebewegungen auf die virtuellen Cursorbewegungen für jedes Frame. Die Berechnung der Manipulationsparameter (Verschiebungsdistanz, Rotationswinkel, Skalierungsfaktor) wird kontinuierlich durchgeführt und es wird eine genaue Objektplatzierung, -drehung und -skalierung unterstützt (ohne Snapping-Funktionalität oder vordefinierten Skalierungs-/Rotationschritten).

Die Verschiebungsdistanz für die Objektpositionierung wird aus der Positionsänderung der physischen Koordinaten der dominanten Hand (it1, it3) bzw. des Mittelpunkt-Cursors (it2) bestimmt, siehe Gleichung 5.2.10.

Im SR-Modus werden Bewegungen der dominanten Hand ignoriert (fixierter Cursor bei it1). Der Rotationswinkel für die Objektdrehung wird aus der Winkelveränderung der nicht-dominanten Hand (it1) bzw. beider Hände (it2) in der xz-Ebene berechnet, siehe Gleichung 5.2.12. In der Studie 4 wird der Rotationswinkel durch die Orientierungsänderung des Eingabegerätes in der y-Achse bestimmt. Der Selektionspunkt bestimmt das Rotationszentrum, d.h. es wird um den Objektmittelpunkt (Studie 3, Studie 4) oder die nutzerspezifische Rotationsachse (Studie 1, Studie 2) gedreht.

Der Skalierungsfaktor wird durch die Abstandsveränderung zwischen den physischen Koordinaten beider Hände berechnet (it1, it2), siehe Gleichung 5.2.11. Es wird eine objektzentrierte uniforme Skalierung durchgeführt.

In der Tabelle 5.2.2 ist die Kombination der Task-Parameter für die Interaktionsaufgaben innerhalb der Studien zusammengefasst.

Technik	beidhändig asymmetrisch (it1)			beidhändig symmetrisch (it2)		einhändig (it3)
Experiment	Studie 1	Studie 2	Studie 3	Studie 1	Studie 3	Studie 4
Selektion						
Entfernung und Richtung	Ray-Casting dH			Ray-Casting cursor	Mittelpunkt-	Ray-Casting dH
Selektionsentscheidung	Selektionspunkt C_{dH} im Objekt			Selektionspunkt C_{MP} im Objekt		Selektionspunkt C_{dH} im Objekt
Verdeckung	alle Objekte unter Cursor		unterste Obj. unter Cursor	alle Objekte unter Cursor	unterste Obj. unter Cursor	erster Treffer in Objektliste
Selektionspunkt	nutzerspezifisch		Objektmittelpunkt	nutzerspezifisch	Objektmittelpunkt	Objektmittelpunkt
Positionierung						
Start, Ende	Selektion/ Deselektion					
Verschiebungsdistanz	Positionsänderung dH (pP_{dH})			Positionsänderung pP_{MP}		Positionsänderung dH
Berechnung	kontinuierliche Positionsverschiebung					
Rotation						
Start, Ende	SR-Mode Start/Ende			Selektion/ Deselektion		
Rotationswinkel	Winkelveränderung xz-Ebene ndH			Winkelveränderung xz-Ebene zw. beiden Händen		Orientierungsänderung y-Achse dH
Rotationszentrum	Objektzentrum	nutzerspezifisch	Objektzentrum	Objektzentrum		Objektzentrum
Berechnung	kontinuierlicher Rotationswinkel					
Skalierung						
Start, Ende	SR-Mode Start/Ende			Selektion/ Deselektion		
Skalierungsfaktor	Abstand zw. beiden Händen					
Skalierungszentrum	objektzentriert; uniformer Skalierungsfaktor					
Berechnung	kontinuierlicher Skalierungsfaktor					

Tabelle 5.2.2.: Variation der Task-Parameter für die Interaktionsaufgaben in den Studien.

Cursordarstellung und Interaktionsmodi

Der aktuelle Interaktionsmodus wird über die Cursors visualisiert. Der Nutzer aktiviert die unterschiedlichen Interaktionsmodi per Tastendruck auf den Wiimotes mit dem Daumen oder dem Zeigefinger. Abbildung 5.2.11 zeigt die Cursordarstellung und Tastenbelegung für die unterschiedlichen Interaktionsmodi in den Studien 1, 2 und 3.

Bei der Cursordarstellung wird zwischen Leerlauf (Idle-Mode) und Manipulationsmodus (Drag-Mode, SR-Mode, DSR-Mode) unterschieden. Die Größe der Cursor wird an die Betrachtungsdistanz angepasst (siehe Gleichung 5.2.4). Bei it1 wird im SR-Mode eine zusätzliche Linie zwischen den Cursors angezeigt, um die Änderung des Skalierungsfaktors und des Rotationswinkels zu veranschaulichen. Die Cursordarstellung bei it3 ist analog zur Darstellung der dominanten Hand bei it1 (ohne SR-Mode).

Um den Drift-Effekt zwischen der physischen Wiimote-Steuerung und der virtuellen Cursordarstellung zu reduzieren, kann der Nutzer die virtuelle Cursorpositi-

on auf die zugehörige physische Position zurücksetzen (Cursor Reset). Der Cursor Reset erfolgt durch einmaliges Drücken der Resettaste. Damit das Zurücksetzen nicht während der Manipulationsoperationen durchgeführt wird (ggf. Verlust der mentalen Karte), wurde als Resettaste der Home-Button auf der Wimote der dominanten Hand gewählt ⁵.

In Studie 3 werden zwei Varianten des Wechsels zwischen absolutem und relativem Mapping getestet (manuelles und automatisches Umschalten). Beim manuellen Umschalten kann durch Drücken der Switchtaste (B-Button) der dominanten Hand von der absoluten in die relative Cursorberechnung gewechselt werden. Umgekehrt wird durch erneutes Drücken der Switchtaste wieder von der relativen in die absolute Cursorberechnung gewechselt. Hier wird eine Visualisierung des Manipulationsmodus mit absoluter Cursorberechnung und des Idle-Mode mit relativer Cursorberechnung benötigt (siehe Abbildung 5.2.11). Beim automatischen Umschalten wird eine hybride Cursorberechnung verwendet, d.h. es wird automatisch von der absoluten Cursorberechnung im Idle-Mode in die relative Cursorberechnung im Manipulationsmodus gewechselt und ein Drücken der Switchtaste wird von der Anwendung ignoriert.

Um die mentale Karte der Cursorpositionen zu erhalten werden beim Cursor Reset und beim Wechsel der Cursorberechnung die Cursorpositionen animiert.

Interaktion	asymmetrische Interaktion						symmetrische Interaktion								
	Studie 1		Studie 2		Studie 3		Studie 1			Studie 3					
	C_{ndH}	C_{dH}	C_{ndH}	C_{dH}	relativ C_{ndH} C_{dH}	absolut C_{ndH} C_{dH}	C_{ndH}	C_{MP}	C_{dH}	relativ C_{ndH} C_{MP} C_{dH}	absolut C_{ndH} C_{MP} C_{dH}				
Idle-Mode															
Drag-Mode (position)							DSR-Mode			DSR-Mode			DSR-Mode		
SR-Mode (scale/rotate)															
Buttons - Selektionstaste - Manipulations-taste - Resettaste - Switchtaste	dH: [A] und ndH: [A] dH: [B]	dH: [A] und ndH: [A] dH: [Home]	dH: [A] dH: [A] und ndH: [A] dH: [Home] dH: [B]				dH: [A] dH: [Home]			dH: [A] dH: [Home] dH: [B]					

Abbildung 5.2.11.: Tastenbelegung der Eingabegeräte und die Darstellung der unterschiedlichen Interaktionsmodi über die visuellen Cursors (Kreis, Kreuz): der dominanten Hand (C_{dH}), der nicht dominanten Hand (C_{ndH}) und dem Mittelpunkt-Cursor (C_{MP}).

⁵ In der Studie 1 konnte die Resettaste (B-Button) während der Manipulation aktiviert werden, dies führte zu unerwarteten Nebeneffekten im Experiment.

5.2.3. Mapping

Bei den distanzabhängigen Manipulationstechniken werden die physischen Handbewegungen mit der Wiimote auf virtuelle Cursorbewegungen in der Anwendung abgebildet, wobei der aktuelle Abstand des Nutzers zur Displaywand die Präzision der Manipulationsoperationen beeinflusst (siehe Abbildung 5.2.12). In der GHD-Umgebung im VC-Lab erfolgt das Mapping innerhalb der virtuellen Anwendung. Die Positions- und Orientierungsdaten im physischen Interaktionsraum werden vom Trackingsystem erkannt (6-DOF reale Positionsdaten, rP) und via VRPN an den VRPN-Client des Frameworks übertragen und in die virtuelle Umgebung transformiert (3-DOF physische Positionsdaten⁶, pP). In der Testanwendung werden die physischen Positionsdaten entsprechend der verwendeten Mappingmethode auf virtuelle Koordinaten auf der Displayfläche abgebildet (2-DOF virtuelle Positionsdaten⁷, vP). Die Parameter der Mappingmethode werden durch die Eingabemodalität, den physischen Interaktionsraum und die Testanwendung bestimmt. Es werden absolute Eingabegeräte verwendet (*position control*), deren Bewegungen orthogonal (Studie 1-3) oder perspektivisch (Studie 4) mittels Ray-Casting projiziert werden. In den Experimenten wurden unterschiedliche Kombinationen der Cursorberechnung und Mappingfunktion getestet (siehe Tabelle 5.2.1). Die Präzision der Interaktion erfolgt, indem der Mappingfaktor auf die Manipulationsparameter angewendet wird (Gleichung 5.2.13).

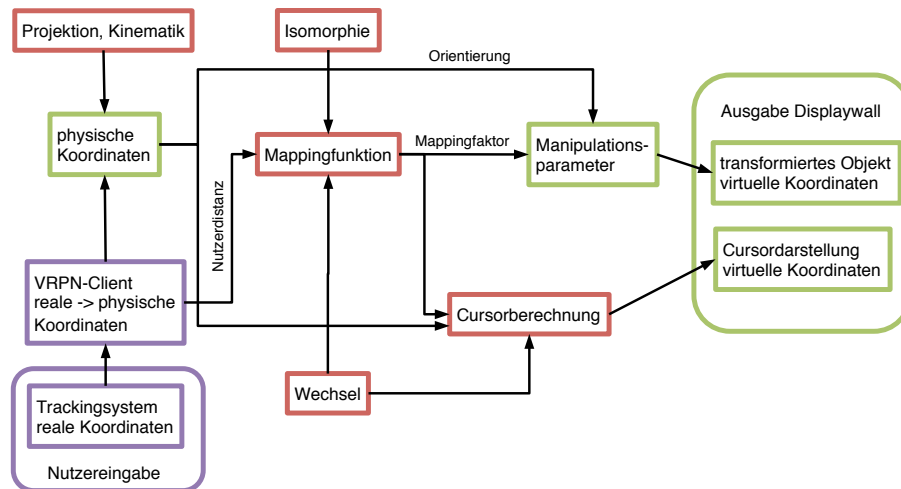


Abbildung 5.2.12.: Schematische Darstellung des Zusammenspiels der Parameter für die distanzabhängigen Manipulationstechniken.

⁶ Die Orientierung der Wiimote wird bei it3 in einem eigenen Orientierungsvektor gespeichert.

⁷ Es werden homogene Koordinaten verwendet, wobei die y-Koordinate von vP auf Null gesetzt ist.

Bestimmung der Nutzerdistanz

Für die Bestimmung der aktuellen Nutzerdistanz (*user-display-distance*, udd) wurde in Studie 1 und 2 der gemittelte Abstand beider Wiimotes zur Displaywand berechnet (Gleichung 5.2.1). Dadurch war der Mappingfaktor stärker von der Motorik als von der Mobilität des Nutzers abhängig. Deshalb wurde in Studie 3 die Kopfposition (Head-Tracker) zur Abstandsbestimmung verwendet, die eine gute Approximation zum tatsächlichen Betrachtungsabstand des Nutzers zur Displaywand bietet und die Mobilität stärker unterstützt (Gleichung 5.2.2). In Studie 4 wurde der Abstand der Wiimote zur Displaywand als Nutzerdistanz verwendet, da kein weiteres Eingabegerät verwendet werden sollte (Gleichung 5.2.3).

Die **Nutzerdistanz** wurde innerhalb des physischen Interaktionsraumes auf eine minimale und maximale Trackingdistanz D beschränkt ($udd \in [minD; maxD]$), abhängig vom verwendeten Setup (vgl. Tabelle 5.2.1):

$$\emptyset \text{ Wiimotes: } udd = (pP_{dH}[y] + pP_{ndH}[y]) * 0.5 \quad (5.2.1)$$

$$\text{Head-Tracker: } udd = pP_{head}[y] \quad (5.2.2)$$

$$\text{Wiimote: } udd = pP_{dH}[y] \quad (5.2.3)$$

Der **Cursorradius** wird an die Nutzerdistanz angepasst, wobei der Divisor durch Tests im Interaktionsraum ermittelt wurde:

$$C_{radius} = udd / 150.0 \text{ mit } C_{radius} \geq 0.2 \text{ inch} \quad (5.2.4)$$

Mappingfunktion

Die beschriebenen Mappingfunktionen aus Abschnitt 3.2.1 wurden für die Experimente folgendermaßen definiert. Der Mappingfaktor mf wird ebenfalls beschränkt ($mf \in [mf_{min}; mf_{max}]; mf, udd \in \mathbb{R}$).

Statisch: Die statische Mappingfunktion dient als Basisfunktion in den Experimenten. Hier werden die Bewegungen im realen Raum direkt auf die Cursorbewegungen im virtuellen Raum abgebildet (1:1 Mapping). Der Mappingfaktor ist konstant eins: $mf = 1.0$.

Zonenartig: Der vordefinierte feste Mappingfaktor nimmt mit jeder Zone (Zonentiefe k) entfernt von der Displaywand zu ($mf_{min} < mf_1 < mf_2 < mf_{max} \in \mathbb{R}$).

$$mf = f(udd) = \begin{cases} mf_{min}, & \text{wenn } udd \leq minD \\ mf_1, & \text{wenn } (minD + k) \geq udd > minD \\ mf_2, & \text{wenn } (minD + 2k) \geq udd > (minD + k) \\ mf_{max}, & \text{wenn } udd \geq maxD \end{cases} \quad (5.2.5)$$

Linear: Bei der linearen Mappingfunktion wächst der Mappingfaktor kontinuierlich mit zunehmenden Abstand zur Displaywand.

$$mf = f(udd) = m * udd + b; m, b \in \mathbb{R} \quad (5.2.6)$$

mit $m = \frac{mf_{max} - mf_{min}}{maxD - minD}$ und $b = mf_{max} - (m * maxD)$

Exponentiell: Bei der exponentiellen Mappingfunktion wächst der Faktor im Nahbereich langsamer an und im mittleren Entfernungsbereich schneller.

$$mf = f(udd) = a + b * e^{(udd*c)} \text{ mit } a, b, c \in \mathbb{R} \quad (5.2.7)$$

Hybrid: Die hybride Mappingfunktion verwendet im Nah- und Fernbereich (vgl. Abschnitt 4.1) einen konstanten Mappingfaktor und im Zwischenbereich einen kontinuierlichen (linearen oder exponentiellen) Mappingfaktor.

$$mf = \begin{cases} mf_{min}, & \text{wenn } udd \leq D_{Nahbereich} \\ mf_{max}, & \text{wenn } udd \geq D_{Fernbereich} \\ f(udd), & \text{Zwischenbereich} \end{cases} \quad (5.2.8)$$

Die Interaktionspräzision für die Tasks kann unterschiedlich sein (Kombination der Mappingfunktionen):

$$mf = \begin{cases} mf_T = f(udd)_T, & \text{für Translation} \\ mf_R = f(udd)_R, & \text{für Rotation} \\ mf_S = f(udd)_S, & \text{für Skalierung} \end{cases} \quad (5.2.9)$$

Wenn die Interaktionspräzision für alle Manipulationstasks gleich ist, dann gilt $mf = mf_T = mf_R = mf_S$.

Manipulationsparameter

Die Manipulationsparameter für die einzelnen Interaktionstasks werden aus der Nutzereingabe folgendermaßen berechnet. Für jeden Zeitpunkt i liegen die physischen Koordinaten für die Eingabegeräte vor $(pP_{i_{dH}}, pP_{i_{ndH}})$. Für die Bestimmung des Rotationswinkels (it1, it2) wird die Umrechnung von kartesischen Koordinaten $P(x, y)$ in Polarkoordinaten $P(r, \varphi)$ mit $\tan \varphi = \frac{y}{x}$ genutzt. Die (distanzabhängigen) Manipulationsparameter werden für die Objekttransformation verwendet, um die

Präzision der Manipulation anzupassen.

Positionierung (5.2.10)

$$translationVector = \overrightarrow{pP_{i_{dH}} pP_{(i-1)_{dH}}}$$

bzw.

$$pP_{i_{MP}} = (pP_{i_{dH}} + pP_{i_{ndH}}) * 0.5$$

$$translationVector = \overrightarrow{pP_{i_{MP}} pP_{(i-1)_{MP}}}$$

Skalierung (5.2.11)

$$V_i = \overrightarrow{pP_{i_{dH}} pP_{i_{ndH}}}$$

$$V_{i-1} = \overrightarrow{pP_{(i-1)_{dH}} pP_{(i-1)_{ndH}}}$$

$$scaleFactor = \frac{\|V_i\|}{\|V_{i-1}\|}$$

Rotation (5.2.12)

$$\varphi_{(i-1)} = \arctan\left(\frac{V_{(i-1)}[z]}{V_{(i-1)}[x]}\right)$$

$$\varphi_i = \arctan\left(\frac{V_i[z]}{V_i[x]}\right)$$

$$rotationAngle = \varphi_i - \varphi_{(i-1)}$$

Präzision der Manipulation (5.2.13)

$$translationVector' = translationVector * mf_T$$

$$scaleFactor' = (scaleFactor)^{mf_S}$$

$$rotationAngle' = rotationAngle * mf_R$$

Cursorberechnung

Die absolute Cursorberechnung verwendet einen statischen Mappingfaktor ($mf = (1;0;1)^T$; Gleichung 5.2.14). Bei der relativen Cursorberechnung wird die Cursorverschiebung (zum Zeitpunkt i) mit dem distanzabhängige Mappingfaktor multipliziert und als relative Bewegung auf die virtuelle Cursorposition abgebildet (Gleichung 5.2.15). Bei der hybriden Cursorberechnung wird im Idle-Mode die absolute Cursorberechnung verwendet und sobald ein Objekt erfolgreich selektiert ist, wird in die relative Cursorberechnung gewechselt, bis der Manipulationsmodus beendet wird.

absolut $vP_i = pP_i * (1;0;1)^T$ (5.2.14)

relativ $i = 1 : vP_i = pP_i * (1;0;1)^T$ (5.2.15)

$$i > 1 : vP_i = vP_{i-1} + (V_i * mf)$$

$$= vP_{i-1} + translationVector'$$

5.2.4. Designentscheidungen

Unser Experiment zu den distanzabhängigen Manipulationstechniken ist an die Aufgabenstellung von Jota et al. [JPJ09] angelehnt. Ihre Studie hat gezeigt, dass die *Point* Metapher für variable Distanzen die besseren Performanzergebnisse (Bearbeitungszeit, Nutzerpräferenz) geliefert hat (vgl. Abschnitt 2.4.2). Daher wurde für die Selektion und Positionierung eine Ray-Casting Technik umgesetzt.

Die Verwendung einer rein distanzabhängigen relativen Cursorberechnung (Studie 1) hat zu störendem Drift-Effekt geführt. Weiterhin hat Studie 1 gezeigt, dass für die Selektionsaufgabe ein absolutes Mapping in unserer GHD-Umgebung ausreicht und für die meisten Manipulationsaufgaben ein Mappingfaktor größer eins nicht zwingend notwendig ist. Angelehnt an die Studien von Vogel und Balakrishnan [VB05], die zeigen, dass eine Kombination von absoluten und relativen Selektionstechniken einen guten Kompromiss zwischen Geschwindigkeit und Genauigkeit bildet, wurde eine hybride Cursorberechnung für die nachfolgenden Studien verwendet.

In Studie 3 wurde die hybride Cursorberechnung um eine anwendungsspezifische Funktion erweitert, die die Nutzerfrustration reduzieren soll. Aufgrund der Aufgabenstellung und den Beobachtungen aus den Pilotstudien wird angenommen, dass der Nutzer weiterhin im Manipulationsmodus mit hoher Präzision arbeiten möchte, bis das Objekt korrekt manipuliert wurde, wenn er im Nahbereich ein Objekt innerhalb des Targetcontainers platziert. Somit wird nicht in das absolute Mapping beim Loslassen des Objektes gewechselt, sondern das relative Mapping bleibt aktiv (Idle-Mode). Jedoch müssen die virtuellen Cursorpositionen an die physischen Koordinaten beim Loslassen des Objektes angeglichen werden, um den Drift zu reduzieren. In das absolute Mapping wird bei erfolgreicher Einsortierung gewechselt oder der Nutzer verlässt explizit den Nahbereich und entfernt sich von der Displaywand.

Bei der 3D Interaktion ist das natürliche Handzittern aufgrund der orthogonalen Projektion bei den beidhändigen Interaktionstechniken für die Nutzer nicht wahrnehmbar gewesen. Bei der einhändigen Interaktionstechnik verursachte die perspektivische Projektion einen deutlich sichtbaren Tremor, der durch eine einfache Filterfunktion (gewichteter Mittelwert) gemildert wurde:

$$pP_i = \frac{(pP_{(i-1)} * 0.5) + pP_i}{1.5} \quad (5.2.16)$$

Um das Tracking zu stabilisieren und das Zittern zu reduzieren, wurden die Glättungsfaktoren für die Translation und Rotation der Trackables in der Trackingsoftware angepasst⁸.

Für die Positionierung ist eine Snapping-Funktionalität zwar für den Nutzer hilfreich, aber anwendungsabhängig. Beim verwendeten Interaction Scaling soll der

⁸ Smoothing Settings: Translation=0.1; Rotation=0.2

Nutzer gezielt bei der Genauigkeit der Manipulation unterstützt werden und die Untersuchungsergebnisse sollen auf andere Aufgaben übertragbar sein, weshalb das Snapping im Experiment nicht unterstützt wurde.

Die Mappingmethode wurde in die Testanwendung integriert, um die Parameter während der Ausführung einfach umschalten zu können. Alternativ kann die Präzision der Interaktion auch vor der eigentlichen Anwendung angepasst werden, indem die Berechnung der virtuellen Positionsdaten und die Präzision der Manipulation beispielsweise im VRPN-Client erfolgt. Dazu müssen die Modusänderungen vor der Anwendung abgefangen werden und in der entsprechenden Mappingmethode berücksichtigt werden. Bei der Transformation der realen Positionsdaten in die physischen Positionsdaten wird zusätzlich die Mappingmethode auf die physischen Koordinaten angewendet. Anschließend bildet die Anwendung die eingehenden Positionsdaten direkt auf die virtuellen Cursors ab bzw. berechnet die Manipulationsparameter aus den distanzabhängigen physischen Koordinaten.

6. Evaluierung der distanzabhängigen Interaktionstechniken

Die beschriebenen distanzabhängigen 2D Interaktionstechniken für GHD-Umgebungen aus Abschnitt 3.3.2 und 3.3.3 wurden in Nutzertests evaluiert. Um die variierende Präzision der Interaktion bei den Manipulationsaufgaben zu testen, wurde eine 2D Puzzleanwendung implementiert. In allen Studien ist die Aufgabenstellung, die 2D Objekte durch Manipulation in die zugehörigen Zielobjekte (Silhouetten) entsprechend der verwendeten Interaktionstechnik und mit Unterstützung des Interaction Scaling einzusortieren. Die Umsetzung der Interaktionstechniken und der Testanwendung wird ausführlich in Kapitel 5.2 beschrieben, sowie das Mapping für die distanzabhängigen Manipulationstechniken in Abschnitt 5.2.3 erläutert. Aus diesem Grund werden in den folgenden Abschnitten die Randbedingungen für die jeweilige Studie nur stichpunktartig aufgezählt. Des Weiteren wurden die Taskparameter zwischen den Studien variiert, die in Tabelle 5.2.2 zusammengefasst sind.

Im Rahmen der Arbeit wird untersucht, wie Interaction Scaling die Nutzerperformance bei Manipulationsaufgaben in GHD-Umgebungen unterstützen kann.

Für alle Nutzerstudien wurden Ad-hoc-Stichproben verwendet. Alle Testpersonen haben freiwillig und ohne Aufwandsentschädigung an den Studien teilgenommen. Die statistische Auswertung der Experimente erfolgte mit Hilfe der Statistiksoftware SigmaPlot¹. Die verwendeten Boxplots zeigen als Whisker das 10. und 90. Perzentil. Zusätzlich werden die Ausreißer (als Kreuze) und der Mittelwert (als gestrichelte Linie) im Boxplot dargestellt.

¹ <http://www.sigmaplot.com>

6.1. Pilotstudien

6.1.1. Beidhändige 2D Manipulationstechniken

Anhand von zwei Vorstudien wurden die asymmetrische Interaktionstechnik (it1) und die symmetrische Interaktionstechnik (it2) mit einem kleinen Personenkreis getestet. Auf diese Weise sollten die Parameter aufgedeckt werden, die das Konzept der distanzabhängigen Interaktion beeinflussen. Die Erkenntnisse der Vorstudien sind in das Experimentdesign der umfassenderen Benutzerstudie (Studie 3) eingeflossen.

Zu Beginn des Experimentes führten die Probanden ein 5-minütiges Tutorial zum Üben der Interaktionstechniken durch. Anschließend wurde das Experiment gestartet. Ein Durchlauf startet mit dem ersten selektierten Objekt und endet mit dem letzten korrekt einsortiertem Objekt. Am Ende des Experimentes hat jeder Proband Fragen zu seiner Person und seinen Präferenzen bezüglich der getesteten Interaktionstechniken und Mappingfunktionen (statisch, linear, hybrid, zone) beantwortet (Posttest-Fragebögen siehe Anhang A.2.1 und A.2.2).

Hypothesen

Wir behaupten, dass Interaction Scaling (IS) die Nutzerperformanz verbessert verglichen mit Manipulationsaufgaben, welche den aktuellen Betrachtungsabstand nicht berücksichtigen.

Im Rahmen der Vorstudien wurden folgende Unterschiedshypothesen getestet:

- Hypothese 1: Es gibt Unterschiede beim Lösen der Aufgabenstellung mit IS und ohne IS. In Studie 1 bezieht sich dies auf die Bearbeitungszeit bei beiden Interaktionstechniken (it1, it2), in Studie 2 auf die Bearbeitungszeit und die Selektionsanzahl bei it1.

$$H_0 : \mu_{\text{statisch}} = \mu_{IS} \quad \text{bzw.} \quad \mu_{\text{statisch}} = \mu_{\text{linear}} = \mu_{\text{hybrid/zone}}$$

$$H_1 : \mu_{\text{statisch}} \neq \mu_{IS} \quad \text{bzw.} \quad \mu_{\text{statisch}} \neq \mu_{\text{linear}} \quad \text{oder} \quad \mu_{\text{statisch}} \neq \mu_{\text{hybrid/zone}}$$

- Hypothese 2: Die Bearbeitungszeit ist kürzer mit einem distanzabhängigen Mappingfaktor (mit IS).

$$H_0 : \mu_{\text{statisch}} \leq \mu_{IS}$$

$$H_1 : \mu_{\text{statisch}} > \mu_{IS}$$

- Hypothese 3: Es gibt unterschiedliche Bearbeitungszeiten zwischen den Interaktionstechniken (Faktor A) und den Mappingfunktionen (Faktor B).

	H_0	H_1
$A :$	$\mu_{it1} = \mu_{it2}$	$\mu_{it1} \neq \mu_{it2}$
$B :$	$\mu_{statisch} = \mu_{linear} = \mu_{hybrid/zone}$	$\mu_{Bi} \neq \mu_{Bj}$
$AxB :$	$\mu_{AiBj} = \mu_{Ai} + \mu_{Bj} - \mu_{gesamt}$	$\mu_{AiBj} \neq \mu_{Ai} + \mu_{Bj} - \mu_{gesamt}$

- Hypothese 4: Mit Hilfe eines distanzabhängigen Mappingfaktors (mit IS) werden weniger Objektselektionen benötigt (nur Studie 2).

$$H_0 : \mu_{statisch} \leq \mu_{IS}$$

$$H_1 : \mu_{statisch} > \mu_{IS}$$

6.1.1.1. Studie 1: symmetrische und asymmetrische Interaktionstechnik

In Studie 1 wurden beide Interaktionstechniken mit zwei relativen kontinuierlichen Mappingfunktionen (mit IS) und einer absoluten Basisfunktion (ohne IS) getestet.

Design und Prozedur

Für Studie 1 gelten folgende Randbedingungen:

- physischer Interaktionsraum mit Trackingbereich Altbau ($udd \in [minD = 32; maxD = 90]$ inch)
- beidhändige Interaktionstechniken: asymmetrische Interaktionstechnik (it1), symmetrische Interaktionstechnik (it2)
- Berechnung der Nutzerdistanz als durchschnittlicher Abstand beider Wiimotes (nach Gleichung 5.2.1)
- Mappingfunktionen (siehe Abbildung 6.1.1)
 - statische Mappingfunktion (sm) ohne distanzabhängige Präzision $mf = 1.0$ (Basisfunktion)
 - lineare Mappingfunktion (lm) mit $mf = f(udd) = 0.047 * udd - 1.19$; $W = [0.3; 3.0]$ nach Gleichung 5.2.6
 - hybride Mappingfunktion (hm) mit konstantem Mappingfaktor im Nah- und Fernbereich von 20 inch (50 cm) Tiefe und linearem Mappingfaktor im Zwischenbereich (hybrid-lineare Mappingfunktion)

$$mf = \begin{cases} 0.3, & \text{Nahbereich } udd < 50 \text{ inch} \\ 0.106 * udd - 4.95, & \text{Zwischenbereich } W = [0.35; 3.0] \\ 3.0, & \text{Fernbereich } udd > 75 \text{ inch} \end{cases}$$

- Cursorberechnung: relative Cursorberechnung bei einem distanzabhängigem Mappingfaktor und absolute Cursorberechnung bei der Basisfunktion
- Testanwendung: Targetcontainer mit 3 Zielobjekten (Kreis, Quadrat, Dreieck) und Toleranzbereich ± 0.1 inch (2,54 mm) für Position, Größe und Orientierung (siehe Abbildung 5.2.8a)

Für die Skalierung war der maximale Mappingfaktor kleiner ($mf_{S_{max}} = 2.0$) als bei der Translation und Rotation, sowohl bei der linearen als auch bei der hybriden Mappingfunktion (siehe Abbildung 6.1.1). Der Fernbereich von 191 cm (75 inch) ist eine gute Approximation für die Kopfposition von ca. 2,20 m (siehe Gleichung 4.1.3) und der Nahbereich von 127 cm (50 inch) für eine Kopfposition zwischen Nah- und Fernbereich von ca. 1,50 m (nach Gleichung 4.1.1 und 4.1.2).

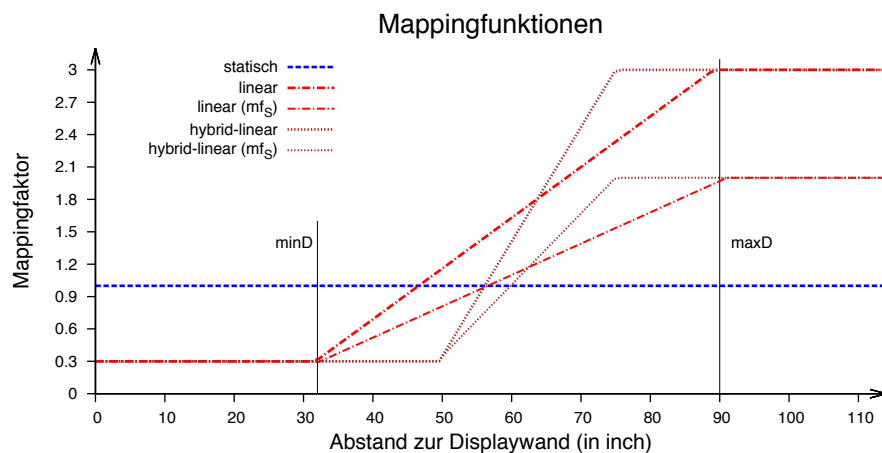


Abbildung 6.1.1.: Die verwendeten Mappingfunktionen in Studie 1.

Im Experiment waren die unabhängigen Variablen die Interaktionstechnik (it1, it2) und die Mappingfunktion (statisch, linear, hybrid-linear), die abhängigen Variablen die Bearbeitungszeit und die Anzahl der durchgeführten Cursor Resets. Die Bearbeitungszeit war die Dauer eines Durchlaufes, gemessen in Sekunden, die zum Lösen der Aufgabe von der ersten Objektselektion bis zum letzten einsortierten Objekt benötigt wurde. Es wurden zwei Durchläufe für jede Bedingung von allen Testpersonen durchgeführt, wobei die Reihenfolge der Bedingungen zwischen den Probanden randomisiert wurde (*within-subject-Design*). Jeder Proband führte somit 12 Bedingungen (2x3x2) im Experiment durch. Damit sich die Probanden für jeden Durchlauf einen Überblick über die Darstellung verschaffen mussten, wurde die Position des Targetcontainers zwischen den Bedingungen variiert, d.h. es gab drei vordefinierte gleichwertige Containerpositionen auf der Displaywand und innerhalb des Containers waren die Targets identisch. Außerdem wurden neben den drei Zielobjekten, welche die gleiche Farbe wie der Targetcontainer hatten, noch 24 bis 36 andersfarbige manipulierbare Objekte dargestellt. Die Objekte wurden zum Anwendungsstart zufällig platziert. Nach dem Einführungstutorial wurde das Ex-

periment gestartet. Der Proband hatte keinen Hinweis bekommen, welche Mappingfunktion im aktuellen Durchlauf verwendet wurde, außer der Wiederholung.

Es nahmen sechs männliche Mitarbeiter der Lehrstühle Computergrafik und Visual Computing an der Universität Rostock teil. Das Alter der Probanden lag zwischen 29 und 53 Jahren (durchschnittlich 37 Jahre) und ein Proband ist Linkshänder. Insgesamt wurden 72 Datensätze in Studie 1 aufgezeichnet.

Ergebnisse

Deskriptive Statistik Bei der asymmetrischen Interaktionstechnik (it1) waren die durchschnittliche Bearbeitungszeit mit der linearen Mappingfunktion ($\bar{x}_{lm}=38$ s, $AD_{lm}=10$ s) und der Basisfunktion ($\bar{x}_{sm}=37$ s, $AD_{sm}=7$ s) annähernd gleich, während die Probanden eine längere Bearbeitungszeit mit der hybriden-linearen Mappingfunktion ($\bar{x}_{hm}=56$ s, $AD_{hm}=29$ s) benötigten. Im Gegensatz dazu wurde bei der symmetrischen Interaktionstechnik (it2) die kürzeste durchschnittliche Bearbeitungszeit mit der Basisfunktion ($\bar{x}_{sm}=31$ s, $AD_{sm}=6$ s) erreicht. Die Testpersonen benötigten durchschnittlich länger mit der linearen Mappingfunktion ($\bar{x}_{lm}=45$ s, $AD_{lm}=12$ s) und der hybriden-linearen Mappingfunktion ($\bar{x}_{hm}=45$ s, $AD_{hm}=17$ s) bei it2. Beim Vergleich der beiden Interaktionstechniken ist die durchschnittliche Bearbeitungszeit bei it2 kleiner, vor allem mit der statischen und hybrid-linearen Mappingfunktion. Jedoch ist die Bearbeitungszeit mit der linearen Mappingfunktion durchschnittlich kürzer bei it1 als bei it2.

Wir konnten keinen Unterschied zwischen der Anzahl der Cursor Resets und den Interaktionstechniken bei der linearen Mappingfunktion beobachten ($\bar{x}_{lm}=1$ bei it1 und it2). Bei beiden Interaktionstechniken benötigte die hybride-lineare Mappingfunktion durchschnittlich die meisten Cursor Resets (it1: $\bar{x}_{hm}=4$ $Md_{hm}=3,5$ und it2: $\bar{x}_{hm}=2$ $Md_{hm}=1,5$). Bei einer höheren Bearbeitungszeit hat der Proband das Cursor Reset² auch öfters eingesetzt.

Die Zusammenfassung der benötigten Bearbeitungszeit und Cursor Resets für die verwendeten Interaktionstechniken und Mappingfunktionen ist anhand der Diagramme in Abbildung 6.1.2 dargestellt.

Darüber hinaus wurden die Probanden befragt, welche Interaktionstechnik sie bevorzugten und welche Kombination von Interaktionstechnik und Mappingfunktion (siehe Tabelle A.2.2 im Anhang). Alle Probanden konnten beide Interaktionstechniken benutzen. Drei Testpersonen verwendeten das Interaction Scaling gezielt bei der Interaktion und kein Proband war genervt vom Interaction Scaling. Drei Probanden bevorzugten die asymmetrische Interaktionstechnik und die anderen drei Probanden präferieren die symmetrische Interaktionstechnik. Die meisten Probanden bevorzugten die lineare Mappingfunktion gegenüber der statischen Mappingfunktion.

² Die Anzahl wird für die statische Mappingfunktion nicht berücksichtigt, da sie keinen Effekt auf die Nutzerinteraktion hatte, sondern nur ein Tastendruck war.

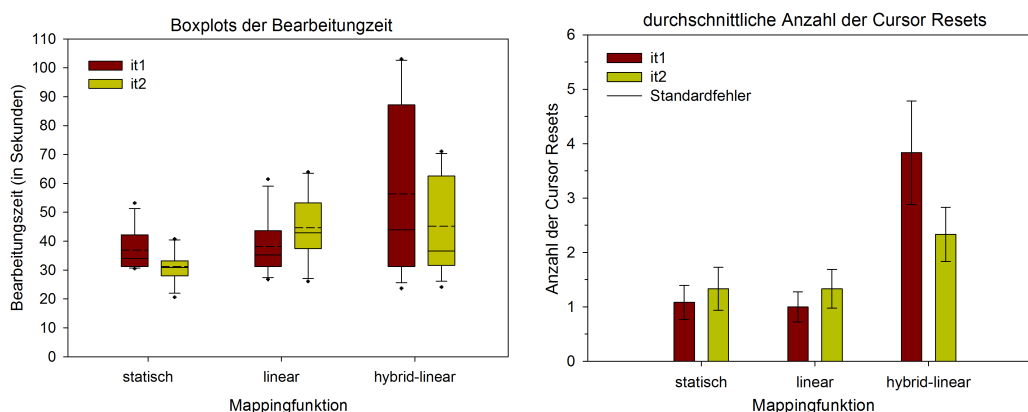


Abbildung 6.1.2.: Grafische Darstellung der Ergebnisse aus Studie 1.

Inferenzstatistik Die Unterschiede der Bearbeitungszeiten zwischen den Mappingfunktionen innerhalb der Interaktionstechnik wurde mit der einfaktoriellen Varianzanalyse mit Messwiederholung auf Signifikanz geprüft (Signifikanzniveau $\alpha = 5\%$). Es wurden bei beiden Interaktionstechniken signifikante Unterschiede zwischen den Mappingfunktionen festgestellt ($F(2, 22)_{it1}=4,68$ $p<0,05$ und $F(2, 22)_{it2}=5,63$ $p<0,05$). Mit Hilfe des Tukey-Tests wurde überprüft durch welche Treatmentstufen (d.h. Mappingfunktionen) die signifikanten Unterschiede entstehen.

Bei der asymmetrischen Interaktionstechnik ist der Unterschied zwischen der hybrid-linearen Mappingfunktion und der linearen bzw. statischen Mappingfunktion signifikant. Bei der symmetrischen Interaktionstechnik hingegen ist der Unterschied zwischen der statischen Mappingfunktion und den distanzabhängigen Mappingfunktionen signifikant.

Ein signifikanter Unterschied zwischen den Interaktionstechniken (A) und den Mappingfunktionen (B) konnte anhand der zweifaktoriellen Varianzanalyse mit Messwiederholungen ($\alpha = 5\%$) nicht festgestellt werden ($F_A(1, 5)=0,76$ n.s.; $F_B(2, 10)=4,09$ n.s.; $F_{A \times B}(2, 10)=1,44$ n.s.).

Im Anhang A.3.2 sind die statistischen Berichte zu finden.

STUDIE 1: Die Hypothese, dass ein Unterschied bezüglich der Bearbeitungszeit mit/ohne Interaction Scaling existiert, konnte für beide Interaktionstechniken mit der hybrid-linearen Mappingfunktion festgestellt werden – Hypothese 1 bestätigt. Jedoch ist der Unterschied anders als erwartet, da mit Hilfe der Basisfunktion die Aufgabe signifikant schneller gelöst werden konnte – Hypothese 2 nicht bestätigt. Ebenso konnte die Hypothese 3 nicht bestätigt werden.

Diskussion

Die Studie hat gezeigt, dass Interaction Scaling mit der verwendeten hybrid-linearen Mappingfunktion nicht geeignet war. Dafür gibt es drei Gründe. Der Mappingfaktor wächst im Zwischenbereich zu schnell, das für einige Probanden schwierig zu handhaben war. Der Nutzer muss nicht den gesamten Interaktionsraum zum Lösen der Aufgabe nutzen. Zusätzlich half eine Art Snapping-Funktion ($\pm 2,54$ mm Toleranz, d.h. 5 mm Snapping-Bereich) den Nutzern die Objekte einzusortieren. Daher war die Basisfunktion zum Lösen der Aufgabe verwendbar.

Weiterhin hat die Studie gezeigt, dass die Präferenz, welche Interaktionstechnik bevorzugt wird, nutzerabhängig ist. Die Probanden fanden die Idee des Interaction Scaling gut, nur waren in diesem Experiment die Nutzer ebenso gut ohne Interaction Scaling wie mit, da eine sehr hohe Präzision für das Lösen der Aufgabe nicht zwingend notwendig war.

Ferner sollten mit der Vorstudie die Effekte aufgedeckt werden, die die Anwendung von Interaction Scaling beeinflussen. Zum Beispiel wurden die Probanden nicht informiert, ob Interaction Scaling aktiv war oder nicht; dies führte teilweise zu Verwirrung oder anderen Erwartungshaltungen. Bei der statischen Mappingfunktion haben sich die Cursors erwartungsgemäß verhalten, so dass sich der Nutzer schnell darauf einstellen konnte. Dies lässt vermuten, dass das Interaction Scaling für Ungeübte nicht unbedingt intuitiv ist.

Für die Verwendung der hybrid-linearen Mappingfunktion war der Zwischenbereich zu schmal gewählt, denn mit nur einem Schritt vorwärts gelangten die Probanden vom unpräzisen Arbeiten zum sehr präzisen Arbeiten.

Ein überraschender Nebeneffekt der verwendeten Tastenbelegung auf der Wiimote war, dass die Resettaste, während der Manipulation gedrückt werden konnte. Dies führte dazu, dass ein selektierte Objekt mittels Cursoranimation verschoben werden konnte beziehungsweise der Nutzer hat die Wiimotes zum Container bewegt, die Resettaste gedrückt und sich somit physischen Aufwand gespart. Dies ist bei den häufigen Cursor Resets der hybrid-linearen Mappingfunktion sichtbar.

Im Fragebogen konnten die Probanden Probleme und Verbesserungsvorschläge benennen. Beispielsweise wünschten sich einige Probanden eine Art Feedback über die aktuelle Präzisionsstufe oder Hinweise, ob ein Schritt vorwärts/rückwärts empfehlenswert ist. Außerdem wurde eine Snapping-Funktionalität für das Einsortieren gewünscht. Diese Wünsche sind teilweise sehr anwendungsspezifisch (siehe Abschnitt 5.2.4 Designentscheidungen).

Unsere Schlussfolgerung aus der Vorstudie ist, dass die lineare Mappingfunktion geeignet ist, aber die hybrid-lineare Mappingfunktion mit dem verwendeten Setup nicht funktioniert.

6.1.1.2. Studie 2: asymmetrische Interaktionstechnik

Basierend auf den Ergebnissen der ersten Studie wurden die Anwendung und die Mappingfunktionen überarbeitet. Um die Dauer des Experimentes für jeden Probanden zu reduzieren, wurde in der Folgestudie nur die asymmetrische Interaktionstechnik mit zwei distanzabhängigen Mappingfunktionen (linear, zonenartig) und der Basisfunktion evaluiert.

Der Interaktionsraum wurde in vier Zonen unterteilt mit vordefinierten Mappingfaktoren. Bei einer Unterteilung in fünf Zonen wären die Zonen zu schmal. Für das lineare Mapping wurden der minimale und maximale Mappingfaktor angepasst. Weiterhin wurden einige Anwendungsparameter angepasst. Beispielsweise wurden kleinere Objekte und Targets verwendet, die aus der Entfernung noch sichtbar sind, so dass der Nutzer den gesamten Interaktionsraum für die Aufgabe benutzen muss. Außerdem wurde der Toleranzbereich für das Einsortieren auf 2 mm bzw. 1,5 mm verringert.

Design und Prozedur

Für Studie 2 gelten folgende Randbedingungen:

- physischer Interaktionsraum mit Trackingbereich Altbau
($udd \in [minD = 20; maxD = 100] \text{ inch}$)
- beidhändige Interaktionstechnik: asymmetrische Interaktionstechnik (it1)
- Berechnung der Nutzerdistanz als durchschnittlicher Abstand beider Wiimotes (nach Gleichung 5.2.1)
- Mappingfunktionen (siehe Abbildung 6.1.3)
 - statische Mappingfunktion (sm) ohne distanzabhängige Präzision $mf = 1.0$ (Basisfunktion)
 - lineare Mappingfunktion (lm) mit $mf = f(udd) = 0.0256 * udd - 0.311$; $W = [0.2; 2.5]$ nach Gleichung 5.2.6
 - Zonen Mappingfunktion (zm) mit konstantem Mappingfaktor in jeder Zone mit einer Zonentiefe von 22.5 inch (57 cm)

$$mf = \begin{cases} 0.3, & \text{Zone 1: } udd < 42.5 \text{ inch} \\ 0.7, & \text{Zone 2: } 42.5 \leq udd < 65 \text{ inch} \\ 1.5, & \text{Zone 3: } 65 \leq udd \leq 87.5 \text{ inch} \\ 2.5, & \text{Zone 4: } udd > 87.5 \text{ inch} \end{cases}$$

- Cursorberechnung: hybride Cursorberechnung bei einem distanzabhängigen Mappingfaktor und absolute Cursorberechnung bei der Basisfunktion

- Testanwendung: Targetcontainer mit 8 Zielobjekten (Kreis, Quadrat, Dreieck, Stern) und einem Toleranzbereich von ± 0.06 inch (1,5 mm) für Größe und ± 0.08 inch (2 mm) für Position und Orientierung (siehe Abbildung 5.2.8b)

Die Toleranzwerte wurden experimentell bestimmt, so dass die Nutzer die Objekte einsortieren können ohne dabei entmutigt zu werden. Für die Zone Mappingfunktion verwendeten wir vordefinierte Mappingfaktoren ungleich eins, so dass sie sich von den anderen Mappingfunktionen unterscheidet.

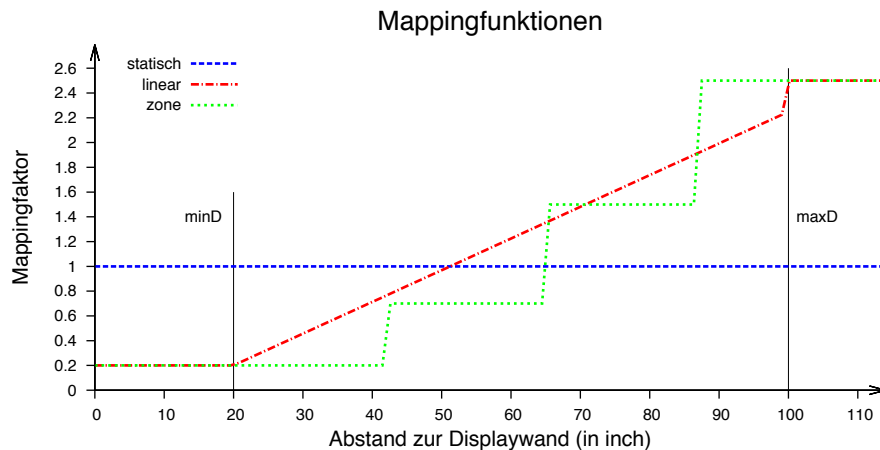


Abbildung 6.1.3.: Die verwendeten Mappingfunktionen in Studie 2.

In Studie 2 wurden die Hypothesen 1, 2 und 4 untersucht (siehe Abschnitt 6.1.1). Im Experiment war die unabhängige Variable die Mappingfunktion (statisch, linear, zone) und die abhängigen Variablen waren die Bearbeitungszeit, die Anzahl der durchgeführten Cursor Resets und die Anzahl der Selektionen. Die Reihenfolge der Bedingungen erfolgte ausbalanciert und jeder Teilnehmer führte zwei Durchläufe pro Bedingung durch (*within-subject-Design*). Nach dem Tutorial startete das Experiment. Der gelbe Targetcontainer wurde stets auf dem gleichen Display dargestellt und neben den acht gelben Objekte wurden noch weitere 44 andersfarbige manipulierbare Objekte dargestellt. Die Position, Größe und Orientierung der manipulierbaren Objekte wurde randomisiert. Die Anwendung hat die Bearbeitungszeit in Sekunden, die Anzahl der Objektselektionen und Cursor Resets aufgezeichnet. Für die Auswertung wurden nur die Selektionen der gelben Objekte berücksichtigt. Der Proband wurde vor jedem Durchlauf informiert, ob ein distanzabhängiger Mappingfaktor aktiv ist oder nicht, jedoch nicht welche Variante. Am Ende des Experimentes hat jeder Proband einen subjektiven Fragebogen ausgefüllt (siehe Anhang A.2.2).

An Studie 2 nahmen 14 Probanden teil, aber die Testergebnisse von vier Probanden waren unvollständig³. Bei der statistischen Auswertung wurden die fehlenden Datensätze von SigmaPlot berücksichtigt.

³ Bei drei Probanden fehlten die Daten von jeweils einer Wiederholung und bei einem Probanden wurde die Anzahl der Objektselektionen in 4 Durchläufen nicht aufgezeichnet.

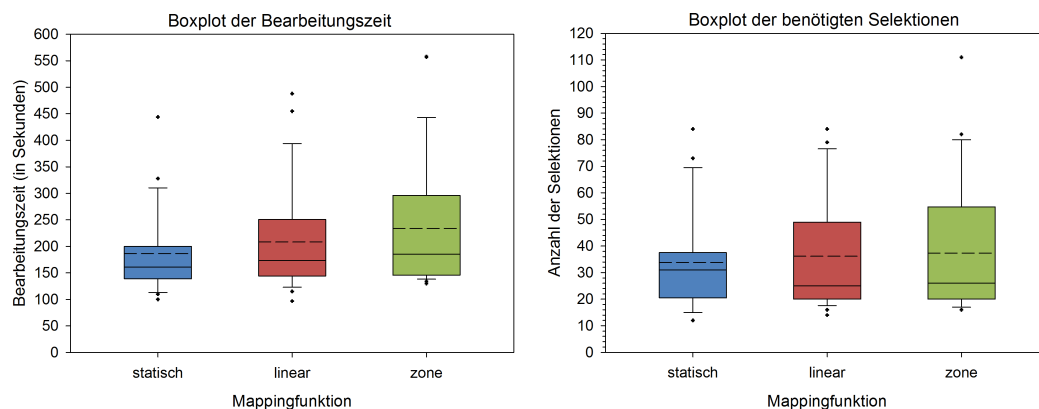


Abbildung 6.1.4.: Grafische Darstellung der Ergebnisse aus Studie 2.

Die Teilnehmer waren Studenten (50%) und Mitarbeiter (50%) der Fakultät Informatik und Elektrotechnik an der Universität Rostock. Es haben vier Frauen und zehn Männer teilgenommen, im Alter von 20 bis 39 Jahre (durchschnittliche Alter 27 Jahre). Die Körpergröße der Teilnehmer variierte zwischen 1,60 m und 1,91 m (durchschnittlich 1,76 m) und alle sind Rechtshänder. Die Teilnehmer hatten wenig oder keine Erfahrung bei der Interaktion in GHD-Umgebungen.

Ergebnisse

Die Berichte der statistischen Auswertung sind im Anhang A.3.3 zu finden und es wurde ein Signifikanzniveau von $\alpha = 5\%$ festgelegt.

Bearbeitungszeit Die durchschnittliche Bearbeitungszeit war mit der statischen Mappingfunktion am kürzesten ($\bar{x}_{sm}=186$ s, $AD_{sm}=79$ s). Die lineare Mappingfunktion und die Zonen Mappingfunktion benötigten durchschnittlich 20-50 Sekunden länger zum Lösen der Aufgabenstellung ($\bar{x}_{lm} =208$ s, $AD_{lm}=98$ s, $\bar{x}_{zm}=234$ s, $AD_{zm}=125$ s). Jedoch unterscheiden sich die Medianwerte nur um etwa 12-25 Sekunden ($Md_{sm}=161$ s, $Md_{lm}=174$ s, $Md_{zm}=186$ s). Zwischen den Mappingfunktionen konnte ein signifikanter Unterschied mit Hilfe der einfaktoriellen Varianzanalyse mit Messwiederholung festgestellt werden ($F(2, 51)=11$; $p<0,001$). Der Tukey-Test zeigt, dass es einen sehr signifikanten Unterschied zwischen der Zonen und der statischen Mappingfunktion gibt ($p<0,001$) und ebenfalls einen signifikanten Unterschied zwischen der Zonen und linearen Mappingfunktion ($p<0,05$).

Im Allgemeinen ist die Bearbeitungszeit zum Lösen der Aufgabe im ersten Durchlauf länger als in der Wiederholung, d.h. in der Wiederholung trat bei mehr als der Hälfte der Durchläufe eine Verbesserung ein. Bei der Basisfunktion waren die Unterschiede zwischen beiden Durchläufen statistisch signifikant ($t=2,34$; $p<0,05$). Hier haben sich 78% der Probanden um durchschnittlich 60 Sekunden verbessert. Bei den distanzabhängigen Mappingfunktionen konnte keine Signifikanz diesbezüglich festgestellt werden, jedoch haben sich 71% der Probanden um durchschnittlich

lich 43 Sekunden bei der linearen Mappingfunktion verbessert und 67% der Probanden um 50 Sekunden bei der Zonen Mappingfunktion.

Cursor Reset Bei der linearen Mappingfunktion wurden durchschnittlich weniger Cursor Resets benötigt als bei der Zonen Mappingfunktion ($\bar{x}_{lm}=4$, $AD_{lm}=3,4$ und $\bar{x}_{zm}=9$, $AD_{zm}=7,6$). Der Unterschied ist statistisch signifikant (Wilcoxon-Test $p<0,001$). Die Cursor Resets wurden für die statische Mappingfunktion nicht berücksichtigt².

Objektselektion Innerhalb eines Durchlaufes wurden 112 Datensätze für die Objektselektion aufgezeichnet (8 Objekte x 14 Probanden). Durchschnittlich haben die Testpersonen bei den distanzabhängigen Mappingfunktionen 2-3 Selektionen mehr benötigt als bei der Basisfunktion ($\bar{x}_{sm}=34$, $AD_{sm}=19$; $\bar{x}_{lm}=36$, $AD_{lm}=21$; $\bar{x}_{zm}=37$, $AD_{zm}=25$). Die Medianwerte verhalten sich umgekehrt, hier wurde mit IS weniger Selektionen benötigt ($Md_{sm}=31$, $Md_{lm}=25$, $Md_{zm}=26$), siehe Abbildung 6.1.4.

Wir konnten interessante Ergebnisse bezüglich der benötigten Objektselektionen identifizieren. Insbesondere ist ein Unterschied zwischen den distanzabhängigen Mappingfunktionen und der statischen Mappingfunktion im ersten Durchlauf erkennbar. Die Tabelle 6.1.1 zeigt, dass häufiger Objekte mit nur zwei Selektionen mit Hilfe des Interaction Scaling einsortiert wurden im Gegensatz zur Basisfunktion. Jedoch konnte keine statistische Signifikanz zwischen den Bedingungen festgestellt werden (beide Durchläufe: $\chi^2(2)=1,8$ n.s. bzw. erster Durchlauf: $F(2, 25)=0,97$ n.s.).

Die Tabelle zeigt ebenfalls den Unterschied zwischen den Objekttypen. Beispielsweise konnten die drehbaren Objekte, wie Quadrat, Dreieck und Stern, mit IS mit weniger Selektionen einsortiert werden als ohne IS.

Während des Experimentes beobachteten wir, dass die meisten Probanden aus der Entfernung ein Objekt näher an das zugehörige Ziel bewegten; anschließend näher an die Displaywand herangingen, um das Objekt einzupassen. Mit Hilfe dieser Strategie benötigten die Probanden folglich nur zwei Selektion mit IS zur korrekten Objektmanipulation. Benötigten die Probanden mehr als 5 Objektselektionen, dann war zu beobachten, dass sie ungeduldig waren oder sie weiterhin aus der Entfernung arbeiteten.

Es konnte ein Zusammenhang zwischen der Bearbeitungszeit und der Anzahl der benötigten Objektselektionen beobachtet werden. Ein Anstieg in der Bearbeitungszeit ist kombiniert mit einer höheren Anzahl an Objektselektionen. Manchmal hat die Testperson längere Zeit benötigt ein bestimmtes Objekt einzusortieren. Einer der Proband hat beispielsweise 16 Objektselektion für das Einsortieren eines Dreieckes benötigt und nur 2 oder 3 Selektionen für einen Stern oder ein Quadrat im gleichen Durchlauf.

Objekttyp	Absolute Häufigkeit												Relative Häufigkeit		
	Quadrate			Kreise			Dreiecke			Sterne			Objekte insgesamt		
	sm	lm	zm	sm	lm	zm	sm	lm	zm	sm	lm	zm	sm	lm	zm
s=1	2	2	4	3	4	1	1	0	2	4	7	4	8,9 %	11,6 %	10,5 %
s=2	6	9	6	10	9	10	9	13	9	5	6	6	26,8 %	33 %	29,8 %
s=3	5	8	2	5	5	6	5	2	1	4	3	4	17 %	16,1 %	12,5 %
s=4	6	0	4	3	1	1	3	2	1	3	2	3	13,4 %	4,5 %	8,7 %
s=5	2	2	3	1	1	2	5	2	3	1	2	1	8 %	6,2 %	8,7 %
s>5	7	7	7	6	8	6	5	9	10	11	8	8	25,9 %	28,6 %	29,8 %

Tabelle 6.1.1.: Häufigkeit der benötigten Selektionen für jeden Objekttyp und Mappingfunktion (statisch, linear, zonenartig) im ersten Durchlauf.

Fragebogen Am Ende des Experimentes wurden die Probanden bezüglich ihrer Zufriedenheit und der Nutzbarkeit, die Aufgabenstellung mit und ohne Interaction Scaling lösen zu können, befragt. Im Durchschnitt gaben die Probanden an, dass sie zufrieden waren die Aufgabe schnell und einfach zu lösen mit und ohne IS. Außerdem konnten sie die Interaktionstechnik schnell lernen und einfach bedienen. Hinsichtlich der Zufriedenheit gab es bei den Antworten keinen sichtbaren Unterschied zwischen den distanzabhängigen Mappingfunktionen und der Basisfunktion. Es haben neun Probanden die Interaktionstechnik mit IS bevorzugt, drei Probanden ohne IS und zwei Probanden gaben keine Präferenz an. Ein Proband hatte Probleme mit der Aufgabenstellung, hier konnte eine sehr hohe Frustration bei der Bearbeitung beobachtet werden und im Fragebogen wurde dementsprechend das Interaction Scaling abgelehnt. Der verwendete Fragebogen und die detaillierte Auswertung sind im Anhang A.2.2 zu finden.

Wir beobachteten, dass die Bearbeitungszeit in unserem Experiment ein weniger aussagekräftiger Faktor für die Nutzerperformanz ist, als die Anzahl der Objektselektionen, welche passender den physischen Aufwand und die Zufriedenheit wiedergibt. Bei der Wiederholung der Bedingung mit der statischen Mappingfunktion konnten ebenfalls mehr Objekte mit nur zwei Selektionen erfolgreich einsortiert werden (vgl. Tabelle A.3.4). Dies lässt vermuten, dass das Einsortieren ohne Interaction Scaling eine Frage der Übung ist. Weiterhin war zu beobachten, dass die Reihenfolge der Bedingungen einen Einfluss auf die Bearbeitungszeit hat. Die Aufgabenstellung erzeugt eine kurze Trainingskurve, d.h. je öfter der Proband das Puzzle übt umso schneller kann er die Aufgabe ohne Interaction Scaling lösen. Diese Beobachtung geben ebenfalls die Kommentare der Probanden wieder.

STUDIE 2: Es konnte ebenfalls ein signifikanter Unterschied zwischen einer distanzabhängigen Mappingfunktion und der statischen Mappingfunktion bezüglich der Bearbeitungszeit festgestellt werden – Hypothese 1 bestätigt. Hypothese 2 wurde nicht bestätigt, da die Bearbeitungszeit bei der statischen Mappingfunktion signifikant kürzer war als mit der Zonen Mappingfunktion bzw. keine Signifikanz zur linearen Mappingfunktion festgestellt werden konnte. In der Studie konnte kein signifikanter Unterschied bei der Anzahl der benötigten Objektselektionen zwischen den Bedingungen mit IS und ohne IS festgestellt werden – Hypothese 4 nicht bestätigt. Jedoch lassen die Ergebnisse vermuten, dass bei einem höheren Bedarf der Manipulationspräzision der physische Aufwand (z.B. motorische und mobile Bewegungen) mit Hilfe des Interaction Scaling reduziert wird.

Diskussion

Die Ergebnisse zeigen, dass ein Mappingfaktor von eins für den Fernbereich in unserem Setup gut geeignet ist, dies war ein Nachteil bei der Zonen Mappingfunktion mit $mf \neq 1$.

Bei der Zonen Mappingfunktion hatten viele Probanden Schwierigkeiten die Aufgabe zu lösen. Die Präzision der Manipulation war im Nahbereich (Zone 1) sehr hoch; die Probanden benötigten einen höheren physischen Aufwand um die Objekte zu manipulieren. Zusätzlich änderte sich der Mappingfaktor aufgrund der schmalen Zonen schnell. Bei der linearen Mappingfunktion wurde die kontinuierliche Anpassung von den Probanden weniger wahrgenommen bzw. es fiel ihnen leichter, ihre Bewegungen anzupassen.

Ein großes Problem beim Interaction Scaling ist der Drift zwischen der Position der Hände und den virtuellen Cursors, so dass die Nutzer teilweise eine unnatürliche Armhaltung zum Interagieren einnehmen oder die Cursorposition manuell zurücksetzen müssen. Dieser Drift-Effekt beeinflusst sehr stark die Nutzerperformance. Nutzer ohne Erfahrung in GHD/VR-Umgebungen waren sehr unzufrieden mit dem Drift-Effekt, vor allem beim Interaction Scaling mit der Zonen Mappingfunktion. Während die Nutzer mit Erfahrung in VR-Umgebungen den Drift-Effekt kompensieren konnten und das Cursor Reset gezielt einsetzten. Es war zu beobachten, dass das Cursor Reset für die Nutzer nicht unbedingt intuitiv ist.

Experimentelle Tests mit einigen Nutzern deutet an, dass die Bearbeitungszeit durch Üben mit Interaction Scaling deutlich verbessert wird. Im Wesentlichen können Nutzer mit einem starken Tremor vom Interaction Scaling profitieren. Diese Idee wird von den Kommentaren der Probanden unterstützt.

Zusätzlich wünschten sich einige Probanden, die automatische Anpassung der Präzision interaktiv deaktivieren zu können. Somit kann der Nutzer im Nahbereich ein großes Objekt schnell verkleinern um es in ein kleines Target einzupassen. Der

verwendete Ansatz des distanzabhängigen Mappingfaktors erfordert ein vor und zurück laufen vom Nutzer, um zügig arbeiten zu können. Jedoch war zu beobachten, dass einige Nutzer sich lieber seitwärts zur Displaywand beim Interagieren bewegen; hier würde eine manuelle Aktivierung der Präzision der Manipulation vermutlich geeigneter sein.

6.1.2. Einhändige 2D Manipulationstechnik

Bei der einhändigen 2D Manipulationstechnik wurde eine absolute Nutzereingabe mit perspektivischer Projektion umgesetzt (siehe Abschnitt 3.3.2). Die Interaktionstechnik (it3) sollte mit Kinder getestet werden, deshalb wurde eine einfache spielerische Anwendung für die Objektmanipulation implementiert (vgl. Abschnitt 5.2.1). Die Interaktionsaufgaben wurden durch unterschiedliche motorische Bewegungsabläufe gesteuert, d.h. die Selektion durch eine Zeigegeste und Tastendruck, die Positionierung durch Armbewegung und die Rotation mittels Handgelenkdrehung. Für die Subtasks Positionierung und Rotation wurde ein distanzabhängiger Mappingfaktor verwendet, um die Präzision der Interaktion zu erhöhen. Testweise wurden zwei verschiedene Toleranzbereiche für die Einpassung der Objekte verwendet. Ziel der explorativen Studie war es, einige Kinder mit der Anwendung spielen zu lassen und dabei zu beobachten, wie sie mit der distanzabhängigen Interaktionstechnik agierten.

6.1.2.1. Studie 4: absolute Nutzereingabe

Für Studie 4 gelten folgende Randbedingungen:

- physischer Interaktionsraum mit Trackingbereich Neubau ($udd \in [\min D = 20; \max D = 110] \text{ inch}$)
- Benutzer: Kinder als Nutzer und Mehrbenutzer-Szenario
- einhändige Interaktionstechniken (it3) mit distanzabhängigem Mappingfaktor
- Berechnung der Nutzerdistanz als Abstand der Wiimote (vgl. Gleichung 5.2.3)
- Mappingfunktionen (siehe Abbildung 6.1.5)
 - lineare Mappingfunktion mit $mf = f(udd) = 0.0067 * udd + 0.267$
 $W = [0.4; 1.0]$ nach Gleichung 5.2.6
 - Zonen Mappingfunktion mit drei unregelmäßigen Zonen (angelehnt an Ergebnisse der Studie 2 und Gleichung 4.1.2)

$$mf = \begin{cases} 0.4, & \text{Zone 1: } udd \leq 50 \text{ inch} \\ 0.6, & \text{Zone 2: } 50 < udd \leq 81 \text{ inch} \\ 1.0, & \text{Zone 3: } udd > 81 \text{ inch} \end{cases}$$

- hybride Mappingfunktion mit exponentiellem Mappingfaktor im Nahbereich und einem konstanten Mappingfaktor ab der Schwelldistanz sD (hybrid-exponentielle Mappingfunktion)

$$mf = \begin{cases} 0.2 + 0.02 * e^{(udd*0.046)}, & \text{Nahbereich } udd < 81 \text{ inch} \\ 1.0, & \text{sonst} \end{cases}$$

- Kombination der Mappingfunktionen, d.h. unterschiedliche Funktionen für die Subtasks Translation (T) und Rotation (R)

$$mf = \begin{cases} mf_T = 0.0041 * udd + 0.168, & udd \leq 81 \text{ inch}, W_T = [0.25; 0.6] \\ mf_R = mf_T * 2.0, & udd \leq 81 \text{ inch}, W_R = [0.5; 1.0] \\ mf_R = mf_T = 1.0, & udd > 81 \text{ inch} \end{cases}$$

- hybride Cursorberechnung: relative Cursorberechnung im Manipulationsmodus und absolute Cursorberechnung im Idle-Mode
- Tremor: Reduzierung mittels Glättungsfaktoren in der Trackingsoftware und einer Filterfunktion (siehe Abschnitt 5.2.4 und Gleichung 5.2.16)
- Testanwendung: Targetcontainer mit 9 Zielobjekten (Kreis, Quadrat, Stern) und für Position und Orientierung ein Toleranzbereich von ± 0.2 inch (5 mm) für die Quadrate und ± 0.1 inch (2,54 mm) für die Kreise und Sterne (siehe Abbildung 5.2.9)

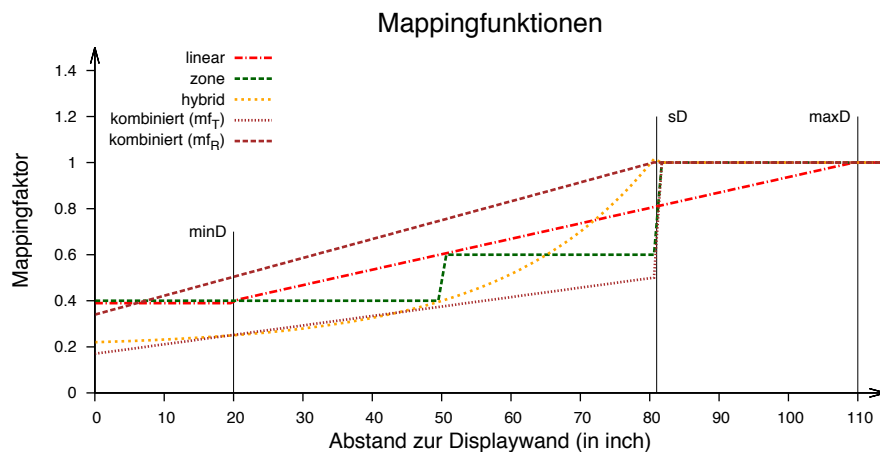


Abbildung 6.1.5.: Die verwendeten Mappingfunktionen in Studie 4.

Es haben fünf Kinder (4 Mädchen und 1 Junge) im Alter von 6 bis 8 Jahren und einer Körpergröße von 1,26 m bis 1,30 m an der Studie teilgenommen⁴. Den Kindern wurde die Aufgabenstellung kurz erläutert und vorgeführt, wie sie die Objekte in

⁴ Während der Studie war ein Elternteil der Kinder anwesend und teilweise waren die Eltern selbst vorab Probanden in Studie 3.

die vorgesehen Platzhalter einsortieren müssen. Ziel war es den Weihnachtsbaum zügig vollständig zu schmücken, wobei die Teilnehmer gegeneinander gespielt haben. Sobald ein Baum vollständig geschmückt war, endete der Durchlauf. Den Kindern wurde nicht erklärt wie das Mapping funktioniert, nur das sich die selektierten Objekte langsamer bewegen lassen, wenn sie nahe an die Displaywand herangehen. Es wurde keine Befragung der Teilnehmer bezüglich der Anwendung oder Interaktionstechnik vorgenommen, sondern nur Messwerte von der Anwendung aufgezeichnet und Beobachtungen durch den Experimentator vorgenommen. Da einige Durchläufe für die Datenaufzeichnung nötig waren, haben die Kinder zur Motivation ebenfalls gegen Erwachsene (Mitarbeiter am Lehrstuhl oder Eltern) gespielt.

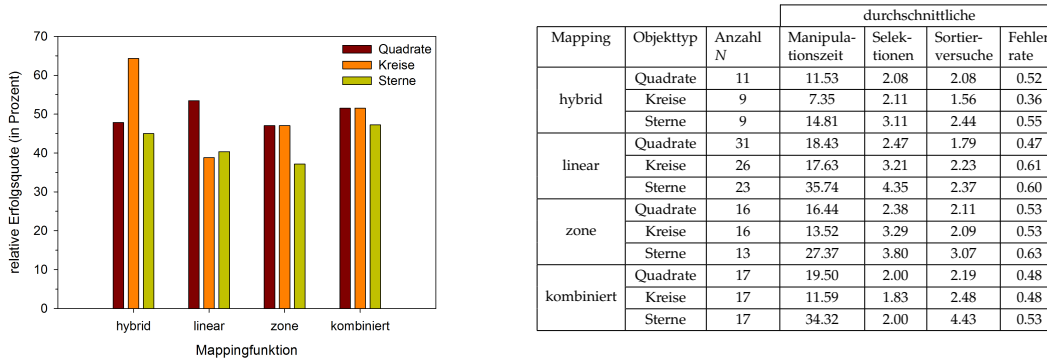
Für jeden Teilnehmer wurden die Anzahl der Objektselektionen, die Manipulationszeit pro Objekt, die Anzahl der Sortiersuche pro Objekt und die zugehörige Fehlerrate beim Einsortieren aufgezeichnet (siehe Erklärung in Studie 3, Abschnitt 6.2.2). Die Mappingfunktionen wurden zufällig variiert. Nach den Durchläufen mit den ersten zwei Kindern wurde noch die Kombination der Mappingfunktion als Variante in der Studie ergänzt.

Ergebnisse

Aufgrund eines Fehlers in der Anwendung waren die Datensätze der ersten drei Teilnehmer unvollständig, daher werden für die Auswertung nur die korrekt einsortierten Objekte berücksichtigt.

Grundsätzlich waren die Sterne am schwierigsten einzusortieren. Dies zeigt sich anhand der längeren durchschnittlichen Bearbeitungszeit von ca. 10–15 Sekunden, der höheren durchschnittlichen Selektionsanzahl und Sortiersuche gegenüber den Quadraten und Kreisen. Am einfachsten konnten die Kinder die Kreise einsortieren, hier wurden die durchschnittlich kürzesten Bearbeitungszeiten und die wenigsten Sortiersuche benötigt. Jedoch wurden durchschnittlich bei den Kreisen etwas mehr Selektionen als bei den Quadraten benötigt (siehe Abbildung 6.1.6b).

Als gute Mappingmethode hat sich die Kombination der Mappingfunktionen gezeigt. Hier konnten die Quadrate und Kreise mit 52% Erfolgsquote einsortiert werden, während die Sterne nur etwa 5% weniger erfolgreich waren. Ähnlich sehen die Ergebnisse mit der Zonen Mappingfunktion aus, die Quadrate und Kreise wurden mit 47% Erfolgsquote einsortiert und die Sterne mit 10% weniger (siehe Abbildung 6.1.6a). Bei der linearen und hybrid-exponentiellen Mappingfunktion variiert die Erfolgsquote zwischen den Objekttypen stärker. Mit Hilfe der linearen Mappingfunktion konnten die Kinder die Quadrate deutlich besser einsortieren (54% Erfolgsquote) im Gegensatz zu den Kreisen oder Sternen (39% bzw. 40% Erfolgsquote). Interessanterweise war die Manipulation der Kreise mit der hybrid-exponentiellen Mappingfunktion deutlich erfolgreicher (64%), als bei den Quadraten (48%) oder Sternen (45%). Diese Variante haben aber nur zwei Kinder getestet.



(a) durchschnittlichen relativen Erfolgsquote beim Einsortieren

(b) Zusammenfassung der Messwerte

Abbildung 6.1.6.: Ergebnisse nach Objekttyp und Mappingfunktion.

Im Allgemeinen haben die Kinder mit einem kleinen Mappingfaktor gearbeitet, da sie selten weiter als 59 inch (1,5 m) von der Displaywand entfernt waren.

Bei der Zonen Mappingfunktion arbeiteten die Teilnehmer nur in der Zone 1 mit einem konstanten Mappingfaktor (0.4). Bei der linearen Mappingfunktion agierten die Teilnehmer hauptsächlich im Nahbereich (20 bis 43 inch) und der Mappingfaktor variierte zwischen 0.4 und 0.56. Bei der hybrid-exponentiellen Mappingfunktion haben die zwei Kinder weiter entfernt agiert (35 bis 75 inch), so dass der Mappingfaktor im Bereich von 0.3 bis 0.8 lag. Bei der Kombination der Mappingfunktionen haben sich die Teilnehmer im Allgemeinen mehr in der Tiefe des Interaktionsraumes bewegt, z.B. von 14 bis 60 inch Interaktionsabstand (mit Mappingfaktor von 0.25 bis 0.41). Wobei hier zu berücksichtigen ist, dass es sich jeweils um den Abstand der Wiimote zur Displaywand handelt (siehe Randbedingungen Studie 4).

Die Kreise konnten mit dem kleinen Toleranzbereich sehr gut einsortiert werden, da nur die Positionierung angepasst werden musste. Musste die Objektpositionierung und -rotation angepasst werden, dann war dies für die Nutzer mit dem höheren Toleranzbereich einfacher, wobei sich die Anzahl der Sortierversuche nicht wesentlich unterschied.

Durch die perspektivische Projektion bei der einhändigen Interaktionstechnik konnten die Teilnehmer bequem von einer festen Position im Interaktionsraum die Objekte selektieren und manipulieren. Um das sichtbare Handzittern des Cursors im Idle-Mode zu reduzieren, haben alle Kinder die Wiimote mit beiden Händen festgehalten.

Interessanterweise war der Bewegungsablauf für die Objektrotation nicht für alle Kinder intuitiv. Ein Kind hat anfänglich die Wiimote geschüttelt, um das Objekt zu drehen, und andere Kinder drehten ihren Arm für die Objektrotation. Nachdem die Kinder jedoch das Prinzip der Rotation durch Drehung des Handgelenkes mit etwas Übung verstanden hatten, war es ihnen möglich die Objekte in einem Bewegungsablauf, beginnend mit der Selektion, korrekt einzusortieren. Zwei der Kinder

musste immer wieder die Hand ausschütteln wegen Ermüdungserscheinungen, da sie die Selektionstaste der Wiimote zu stark gedrückt hielten.

STUDIE 4: Alle Kinder hatten großen Spaß an der Anwendung und Interaktionstechnik und haben sich nicht an dem Wackeln der Cursor gestört. Grundsätzlich versuchten die Kinder, die Objekte mit einem Bewegungsablauf einzusortieren und stört sich nicht daran ihren Oberkörper dabei zu verrenken.

Für die verwendete einhändige absolute Nutzereingabe hat sich die Kombination der Mappingfunktionen, die hybrid-exponentielle und die Zonen Mappingfunktion als geeignet gezeigt. Da die Nutzer von einer festen Position im Nahbereich interagierten, sollte der Mappingfaktor nicht zu klein gewählt werden, um den Drift-Effekt zu reduzieren.

Aufgrund der kurzen Aufmerksamkeitsspanne der Kinder konnte die Anwendung nur fünf bis sieben mal pro Teilnehmer wiederholt werden. Hier wäre es ratsam gewesen verschiedene Targetcontainer als Abwechslung zu verwenden, z.B. einen Schneemann, bei welchem Nase, Augen und Knöpfe einsortiert werden.

6.2. Studie 3: symmetrische und asymmetrische Technik

Wir konnten in den Vorstudien folgendes beobachten: Je höher die erforderliche Präzision wird, desto eher unterstützt ein distanzabhängiger Mappingfaktor die Nutzerperformanz (vgl. Hypothese 2 in Studie 2). Die kontinuierlichen Mappingfunktionen haben sich für die distanzabhängigen Manipulationstechniken als geeignet erwiesen, so dass die lineare und hybride Mappingfunktion überarbeitet wurden. Für unser Setup ist ein Mappingfaktor von eins für die grobe Objektmanipulation ausreichend, somit kann der Nahbereich im Interaktionsraum vergrößert werden.

In der Aufgabenstellung wird zwischen Zielobjekten unterschieden, die mit grober Genauigkeit (Abweichung $\pm 2,5$ mm) und mit hoher Genauigkeit (Abweichung ± 1 mm) einsortiert werden. Damit wird eine Interaktion aus der Entfernung und im Nahbereich unterstützt und getestet. Die Zielobjekte, die eine grobe Genauigkeit beim Einsortieren erlauben, sind als größere Silhouetten mit stärkerer Konturlinie dargestellt als die Zielobjekte mit hoher Genauigkeit (siehe Abbildung 6.2.8).

Bei einem kleinen Mappingfaktor ist ein Cursor Reset erforderlich, jedoch haben die Vorstudien gezeigt, dass das Cursor Reset für einige Benutzer nicht intuitiv ist. Zusätzlich wünschten sich einige Nutzer eine manuelle Kontrolle der Präzision; vor allem die Nutzer, die ungern herumlaufen möchten. Daher wird in der Studie neben der hybriden Cursorberechnung (automatischer Cursorwechsel) auch ein manuelles Umschalten der Präzision getestet. Mit diesen zwei Varianten sollen die Präferenzen der Nutzer berücksichtigt und untersucht werden.

6.2.1. Hypothesen

Wir behaupten, dass mit Hilfe der distanzabhängigen Präzision der Interaktion die Nutzerperformanz für 2D Manipulationsaufgaben verbessert wird.

- Hypothese 1: Die Nutzer können die Aufgabe mit IS effizienter und effektiver bearbeiten, als ohne IS; dies gilt für beide Interaktionstechniken.
- Hypothese 2: Es gibt einen Unterschied beim Einsortieren der Zielobjekte zwischen großen Targets (*big*) und kleinen Targets (*small*).
 - a) Die Fehlerrate ist bei den kleinen Targets mit IS (*mIS*) geringer als ohne IS (*oIS*).
 - b) Die Fehlerrate unterscheidet sich bei den großen Targets nicht.

Targettyp	$H_1 : \mu_{big} \neq \mu_{small}$
kleine Targets	$H_1 : \mu_{mIS,small} < \mu_{oIS,small}$
große Targets	$H_1 : \mu_{mIS,big} = \mu_{oIS,big}$

- Hypothese 3: Es ist zu erwarten, dass bei it1 und it2 weniger Bearbeitungsschritte mit dem automatischen Cursorwechsel benötigt werden als beim manuellen Cursorwechsel ($H_1 : \mu_{auto} < \mu_{manu}$).
- Hypothese 4: Der physiologische Aufwand wird für den Nutzer mit Interaction Scaling verringert.
 - a) Der physiologische Aufwand ist bei it2 geringer als bei it1.
- Hypothese 5: Es werden weniger Sortiersuche benötigt, wenn der Nutzer die Vorteile des IS beim Laufverhalten oder der Taktik des Einsortierens anwendet.
 - a) Beim automatischen Cursorwechsel ist ein vorwärts/rückwärts Laufen bei beiden Interaktionstechniken vorteilhaft.
 - b) Beim manuellen Cursorwechsel wird ein Laufverhalten seitlich zur Displaywand im Nahbereich/Schwelldistanz unterstützt.

6.2.2. Design und Prozedur

Für Studie 3 gelten folgende Randbedingungen der Experimentalumgebung:

- physischer Interaktionsraum mit Trackingbereich Neubau
($udd \in [minD = 30; maxD = 110] \text{ inch}$)
- Berechnung der Nutzerdistanz durch den Kopfabstand zur Displaywand
(nach Gleichung 5.2.2)
- Testanwendung: Targetcontainer mit 8 Zielobjekten (Kreis, Quadrat, Dreieck, Stern) und einem Toleranzbereich von $\pm 0.1 \text{ inch}$ (2,5 mm) für große Targets und $\pm 0.04 \text{ inch}$ (1 mm⁵) für kleine Targets (siehe Abbildung 5.2.8c)

Neben den acht Zielobjekten werden noch weitere 32 manipulierbare Objekte dargestellt, damit der Nutzer sich einen Überblick verschaffen muss. Die Größe, Position und Orientierung der manipulierbaren Objekte ist randomisiert. Die Zielobjekte haben bei der Initialisierung nicht die Größe der Targets, d.h. sie müssen stets manipuliert werden. Es gibt von jedem Objekttyp gleich viele Objekte.

Die unabhängigen Variablen im Experiment sind Interaktionstechnik, Cursorwechsel und Mappingfunktion.

Interaktionstechnik Es wurde die symmetrische und asymmetrische Interaktionstechnik aus Abschnitt 3.3.3 verwendet, deren Nutzerinteraktion und Cursordarstellung in Abschnitt 5.2.2 erläutert wird. Um die Eingabemodalitäten im Experiment weitestgehend konstant zu halten, sind die Resettaste und Swichtaste der dominanten Hand bei beiden Interaktionstechniken identisch. Eine schematische Darstellung der verwendeten Wiimote Buttons für die Interaktionstechniken ist im Anhang Abbildung A.1.1 zu finden.

⁵ Es wurde experimentell getestet, dass die kleinen Targets mit der Basisfunktion einsortierbar sind.

Cursorwechsel Es werden zwei Varianten getestet: ein manuelles Umschalten (*manu*) und ein automatisches Umschalten (*auto*) zwischen absoluter/relativer Cursorberechnung.

- *manuell*: Die Testperson wechselt manuell zwischen absoluter und relativer Cursorberechnung mit Tastendruck (Switchtaste).
- *automatisch*: Beim automatischen Wechsel wird eine hybride Cursorberechnung verwendet (vgl. Abschnitt 5.2.3 – Cursorberechnung). Ist der Nutzer beim Beenden des Manipulationstasks im Nahbereich zur Displaywand (aktuelle Distanz \leq Nahdistanz), dann bleibt die relative Cursorberechnung im Idle-Mode aktiv und der Drift-Effekt wird automatisch reduziert (vgl. Abschnitt 5.2.4). Wenn der Nutzer im Idle-Mode den Nahbereich der Displaywand verlässt (aktuelle Distanz $>$ Nahdistanz), dann wird von der relativen in die absolute Cursorberechnung gewechselt. Die Nahdistanz ist 45 inch (115 cm).

Mappingfunktion Es werden zwei distanzabhängige Mappingfunktionen und eine statische Mappingfunktion getestet (siehe Abbildung 6.2.7):

- *statisch*: Die statische Mappingfunktion (*smf*) dient als Basisfunktion und verwendet keine distanzabhängige Präzision $mf = 1.0$.
- *linear*: Die lineare Mappingfunktion (*lmf*) berechnet einen kontinuierlichen Mappingfaktor, der mit zunehmendem Abstand zur Displaywand wächst (nach Gleichung 5.2.6): $mf = f(udd) = 0.015 * udd - 0.15; W = [0.3; 1.5]$.
- *exponentiell*: Bei der hybrid-exponentiellen Mappingfunktion (*emf*) wächst der Faktor im Nahbereich langsamer an und im mittleren Entfernungsbereich schneller (nach Gleichung 5.2.8 und 5.2.7, $W = [0.28; 1.0]$). Bei einer Distanz größer 81 inch bzw. 2 m (*sD*) bleibt der Mappingfaktor eins (vgl. Gleichung 4.1.2).

$$mf = \begin{cases} 0.28, & udd < 30 \text{ inch} \\ 0.2 + 0.02 * e^{(0.046 * udd)}, & udd \leq sD \\ 1.0, & udd > sD \end{cases}$$

Als abhängige Variablen werden objektive Messwerte (Geschwindigkeit, Fehler-rate, etc.) gemessen und subjektive Daten (Nutzerzufriedenheit, Arbeitsaufwand) gesammelt, um Aussagen über die Nutzerperformanz innerhalb der unterschiedlichen Bedingungen zu treffen. Folgende Daten werden während des Experimentes aufgezeichnet und für die Auswertung berechnet.

Selektion: Der Manipulationstask eines Objektes startet mit erfolgreicher Objekts-
elektion. Die Anwendung zählt die Anzahl der Selektionen pro Objekt.

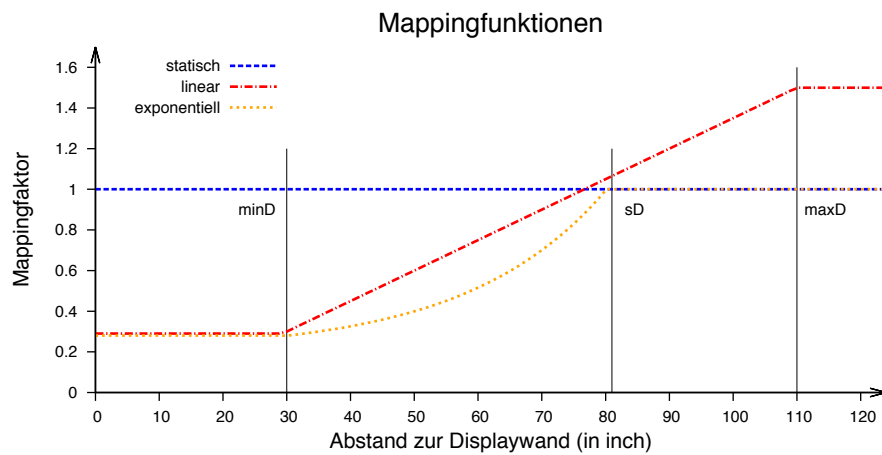


Abbildung 6.2.7.: Die verwendeten Mappingfunktionen in Studie 3.

Manipulationszeit: Sobald ein Objekt selektiert wurde, wird von der Anwendung die Manipulationszeit für das Objekt gemessen, bis das Objekt deselektiert wird. Wird das Objekt erneut bearbeitet, dann werden die Manipulationszeiten für das Objekt addiert. Die Manipulationszeit wird in Sekunden gemessen. Die gesamte Manipulationszeit für einen Durchlauf wird aus der Summe der acht manipulierten Objekte berechnet.

Cursor Resets: Die Anwendung zählt die Anzahl der durchgeführten Cursor Resets für jeden Durchlauf.

Sortiersuche: Für jedes Objekt wird die Anzahl der Sortiersuche mitgezählt, d.h. passt ein Objekt bezüglich Position, Orientierung und Größe in die Silhouette. Ein Sortiersuche erfolgt, wenn das Objekt teilweise oder ganz im Zielcontainer abgelegt wird. Die Pilotstudien haben gezeigt, dass die Probanden aufgrund der Aufgabenstellung die Objekte im Zielcontainer ablegen, auch bei einer Vorsortierung. Weiterhin wird für jeden Sortiersuche die Teststufe (Position, Orientierung, Größe), die fehlschlug und erfolgreich war, aufgezeichnet. Die relative Fehlerrate ergibt sich aus der durchschnittlichen Summe der fehlgeschlagenen Teststufen dividiert durch die Anzahl der Sortiersuche. Je weniger Sortiersuche pro Objekt benötigt werden, umso geringer ist die Fehlerrate für das Objekt.

Am Ende jedes Durchlaufes wird die Anzahl der durchgeführten Cursor Resets und für jedes Objekt die Manipulationszeit, die Anzahl der Selektionen und Sortiersuche gespeichert.

Nutzerperformanz: Die Nutzerperformanz ermittelt sich aus der Effektivität (Fehlerrate, physische Navigation), der Effizienz (Geschwindigkeit, Arbeitsaufwand) und der Zufriedenheit (vgl. Abschnitt 2.2.2).

Physische Navigation: In großen Displayumgebungen kann beobachtet werden, ob der Nutzer vor der Displaywand herumläuft oder eine feste Position ein-

nimmt und welche Entfernung er zum Interagieren bevorzugt (*Mobilität*). Bei der *Motorik* spielt die Anzahl der Tastendrucke (Button-Events) eine Rolle. Um Zusammenhänge zwischen der physischen Navigation der Probanden und den Bedingungen aufzuzeigen, werden die 3D Positionsdaten der Interaktionsgeräte (Basecap, Wiimotes) aufgezeichnet.

Button-Events: Die verschiedenen Button Ereignisse (Manipulationsoperation, Umschaltung) werden für den gesamten Durchlauf berechnet. Für die Bedingung manueller Cursorwechsel wird die Anzahl der Umschaltungen (Switch-taste) mitgezählt. Bei der symmetrischen Interaktionstechnik entspricht die Anzahl der Selektionen aller Objekte der Anzahl der Manipulationsoperationen (Drag-Scale-Rotate), da die Tasks simultan ausgeführt werden. Bei der asymmetrischen Interaktionstechnik wird die Anzahl der Manipulationsoperationen aus den Objektselektionen (Drag) und dem Aktivieren des SR-Mode (Scale-Rotate) berechnet.

Arbeitsaufwand und Zufriedenheit: Für die subjektive Bewertung der Zufriedenheit und des Arbeitsaufwandes werden standardisierte Fragebögen verwendet. Der Posttest-Fragebogen dient zur Bewertung der Nutzerzufriedenheit bezüglich der unterschiedlichen Methoden des Cursorwechsels. Weiterhin sollte jede Testperson subjektiv bewerten, bei welcher Variante des Cursorwechsels sie den Eindruck hatte die Aufgabe schneller zu lösen, weniger Fehler erzeugte und welche Variante sie insgesamt bevorzugt (siehe Anhang A.2.4). Mit dem *NASA Task Load Index* (NASA TLX) bewerten die Probanden die Arbeitsbelastung für die Varianten der Cursorwechsel in den Bereichen *perceptual*, *cognitive* und *motor*. Die Mappingfunktionen sind aus Nutzersicht sehr ähnlich, weshalb dazu keine Präferenzen oder die Arbeitsbelastung abgefragt wurden⁶.

Prozedur

Um ausreichend Daten für die Studie zu bekommen, wurde das Experiment von jedem Probanden mit der asymmetrischen Interaktionstechnik und der symmetrischen Interaktionstechnik durchgeführt. Die beiden Interaktionstechniken verwenden unterschiedliche Interaktionsmetaphern (und verschiedene Eingabemodalitäten), weshalb nur eine Technik pro Experimenttag von jedem Probanden durchgeführt wurde. Dadurch hat sich der Aufwand für den Probanden auf jeweils 45 Minuten in 2 Tagen beschränkt.

Die Reihenfolge der Bedingungen wurde zwischen den Probanden variiert (zufällige Permutation). Damit der mentale Aufwand nicht zu hoch ist, wurden Permutationen mit ständigem Cursorwechsel vermieden (z.B. manu-lmf, auto-lmf,

⁶ Im Experiment haben nur zwei Probanden wahrgenommen, dass die Mappingfunktionen unterschiedlich waren.

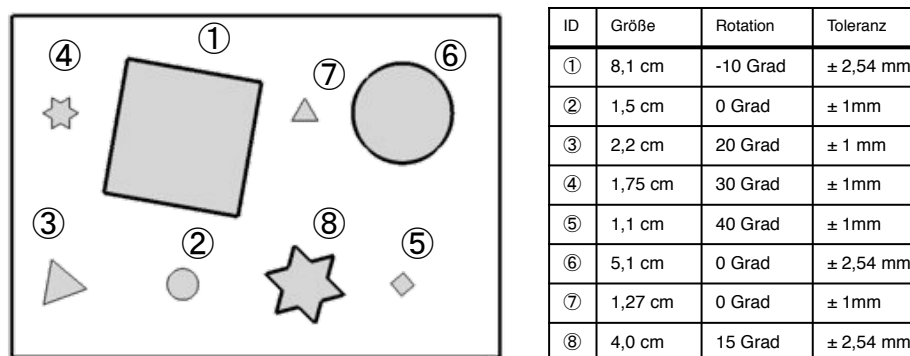


Abbildung 6.2.8.: Schematische Darstellung der Zielobjekte im Container mit den zugehörigen Eigenschaften.

manu-emf, auto-emf). Da die Position des Zielcontainers aufgrund der Bezels nicht genau auf der Mitte der Displaywand platziert werden kann, wurde die Containerposition (links/rechts von der Displaymitte) zwischen den Probanden gewechselt.

Um Informationen über die Vorkenntnisse der Testperson zu bekommen, füllte diese zunächst einen Pretest-Fragebogen aus. Anschließend wurde in einem ca. 10 min Tutorial⁷ die Anwendung erläutert und er/sie übte die Interaktionstechnik mit den verschiedenen Präzisionsfunktionen. Nachdem die Testperson sich mit der Anwendung vertraut gemacht hatte, begann das eigentliche Experiment. Hier musste die Testperson, durch Manipulation acht 2D Objekte (je 2 Sterne, Quadrate, Dreiecke und Kreise) in die zugehörigen gleichfarbigen Zielobjekte einsortieren.

Sobald der Nutzer das erste Objekt selektierte, wurde die Zeitmessung des Versuches gestartet. Die Testperson selektierte und manipulierte ein 2D Objekt entsprechend der verwendeten Interaktionstechnik. Der Versuch endete, wenn das letzte Objekt erfolgreich einsortiert wurde, dann wurde der Zeitmesser gestoppt und alle Messdaten des Versuches wurden von der Anwendung in eine csv-Datei geschrieben.

Der Versuch wurde für die unterschiedlichen Bedingungen wiederholt, wobei immer nur ein Durchlauf pro Bedingung und Testperson durchgeführt wurde. Die Testperson wurde informiert, welcher Cursorwechsel im Durchlauf verwendet wurde. Zum Vergleich führte jede Testperson als ersten und letzten Versuchslauf die Aufgabe ohne distanzabhängigen Mappingfaktor (Basisfunktion) durch. Nach den zwei Durchläufen des Cursorwechsels füllte die Testperson jeweils einen Fragebogen zur Arbeitsbelastung aus. Nachdem alle Versuchsläufe erfolgreich abgeschlossen waren, füllte die Testperson einen Posttest-Fragebogen für die Interaktionstechnik aus.

⁷ Die Aufgabenstellung im Tutorial ist ähnlich zum Versuchslauf, es werden vier Objekte (je eins pro Objekttyp) einsortiert und die Abweichungen zwischen Objekt und Silhouette sind etwas größer (± 0.15 inch und 0.08 inch)

Teilnehmer

Insgesamt haben 24 Probanden freiwillig an der Studie teilgenommen. Die Probanden wurden nicht für ihre Teilnahme am Experiment bezahlt. Von 2 Probanden konnten die Daten nicht verwendet werden, da hier Probleme auftauchten, und das Experiment abgebrochen wurde⁸.

Von den 22 Probanden waren 3 weiblich und 19 männlich, mit einem durchschnittlichen Alter von 30 Jahren (22–42 Jahren, Median 29 Jahre). Die Probanden waren Studenten (9 Teilnehmer) oder Mitarbeiter (13 Teilnehmer) der Fakultät Informatik- und Elektrotechnik, der Universität Rostock. Ein Proband war Linkshänder, alle anderen Rechtshänder. Die Probanden hatten entweder normale Sehschärfe oder trugen Sehhilfen (45% der Probanden). Bis auf 3 Probanden haben alle Teilnehmer schon mit der Wiimote interagiert (z.B. Wii Spiele oder als Eingabegerät). 7 Probanden hatten Erfahrung bei der Interaktion mit Displaywänden und 12 Probanden haben schon einmal mit der Wiimote als Interaktionsgerät (unabhängig von der Spielkonsole) gearbeitet.

Die Kombination der unabhängigen Variablen und der resultierenden Anzahl an Datensätzen ergibt sich wie folgt (*within-subject-Design*):

2	Cursorwechsel (manu, auto)	x
2	Mappingfunktion (lmf, emf)	x
2	Interaktionstechnik (it1, it2)	+
4	Basisfunktion (smf am Anfang und Ende je it)	=
12	Versuchsläufe pro Proband	x
22	Probanden	=
264	Datensätze	

6.2.3. Ergebnisse

Für die Signifikanzprüfung (Signifikanzniveau $\alpha = 5\%$) wurde die Varianzanalyse mit Messwiederholung und der t-Test für abhängige Stichproben verwendet bzw. die Rangvarianzanalyse nach Friedman. Für den Paarvergleich wurde der Student-Newman-Keuls-Test (SNK) und für die Überprüfung auf Normalverteilung der Daten die Shapiro-Wilk Methode verwendet. Die Berichte der statistischen Auswertung sind im Anhang A.3.4 beigefügt sowie eine zusammenfassende Darstellung der Ergebnisse der Signifikanztests in Tabelle A.3.6.

Die Versuchsläufe ohne distanzabhängige Präzision (Basisfunktion), die am Anfang und Ende des Experimentes durchgeführt wurden, werden zusammen betrachtet. Es wird das arithmetische Mittel über die Messwerte gebildet (Manipulationszeit, Sortiersuche, etc.), um eine geeignete Approximation der Bedingung

⁸ Die Teilnehmer konnten die Aufgabenstellung nicht erfolgreich lösen und während des Experimentes Verbesserungsvorschläge diskutierten wollten.

(ohne-smf), aufgrund der Erwartungen beim ersten Durchlauf und der Lernkurve, zu erhalten.

Angesichts der Ergebnisse der Vorstudien wird die gesamte Bearbeitungszeit zum Lösen der Aufgabe nicht berücksichtigt. Im Experiment wurde die Anzahl der Cursor Resets gemessen, jedoch wurde die Funktion von den Probanden bei beiden Interaktionstechniken sehr selten genutzt (94% der gesamten Datensätze haben null Cursor Resets). Daher werden die Cursor Resets nicht weiter betrachtet und nur bei den Button-Events berücksichtigt.

Die Anzahl der Selektionen und der zugehörigen Sortiersuche pro Objekt war bei 94% der Datensätze identisch. Bei den restlichen 6% war bis auf 5 Objekte, stets eine Selektion mehr als die Anzahl des Sortiersuches, so dass die Anzahl der Selektion nicht separat ausgewertet wird (nur bei Button-Events berücksichtigt).

6.2.3.1. Effizienz und Effektivität

Manipulationszeit Die durchschnittliche gesamte Manipulationszeit unterscheidet sich zwischen den Bedingungen innerhalb der Interaktionstechniken um 10-24 Sekunden. Bei it1 liegt sie zwischen 102-112 Sekunden und bei it2 zwischen 86-110 Sekunden, wobei die Bedingungen ohne-smf und auto-emf die längsten und manu-lmf die kürzesten Manipulationszeiten benötigten (vgl. Abbildung A.3.2). Es konnte ein statistisch signifikanter Unterschied bei it1 zwischen auto-emf und manu-lmf festgestellt werden und bei it2 zwischen ohne-smf und beiden manuellen Cursorwechseln ($F(4, 84)_{it1}=2,57$ $p=0,044$; $\chi^2(4)_{it2}=13,77$ $p=0,008$).

Betrachtet man die durchschnittliche Manipulationszeit der Zielobjekte nach ihrem Toleranzbereich, dann benötigten die großen Zielobjekte erwartungsgemäß eine kürzere Manipulationszeit als die kleinen Zielobjekte. Die durchschnittliche Manipulationszeit betrug bei den drei großen Targets etwa 11-12 Sekunden bei it1 und 8-10 Sekunden bei it2 (Md_{it1} zw. 10-12 s Md_{it2} zw. 7-10 s). Die Bedingung manu-emf benötigte jeweils die kürzeste Manipulationszeit (vgl. Abbildung 6.2.9). Bei beiden Interaktionstechniken konnte ein signifikanter Unterschied zwischen den Bedingungen festgestellt werden, jedoch nicht, durch welche Bedingungen dieser Unterschied hervorgerufen wird ($F(4, 84)_{it1,big}=2,51$ $p=0,048$ und $\chi^2(4)_{it2,big}=9,64$ $p=0,047$). Bei den fünf kleinen Zielobjekten hingegen variiert die durchschnittliche Manipulationszeit der Medianwerte bei den Interaktionstechniken zwischen 13-15 Sekunden (it1) bzw. 11-16 Sekunden (it2) und die Bedingungen auto-emf bei it1 und ohne-smf bei it2 benötigten die längsten Zeiten (vgl. Abbildung 6.2.9). Bei it2 ist der Unterschied zwischen den Manipulationszeiten ohne IS und mit IS signifikant ($\chi^2(4)_{it2,small}=16,2$ $p=0,003$), während bei it1 kein signifikanter Unterschied existiert.

Sortiersuche Bei beiden Interaktionstechniken unterscheidet sich die Anzahl der durchschnittlichen Sortiersuche pro Objekt zwischen den Bedingungen mit

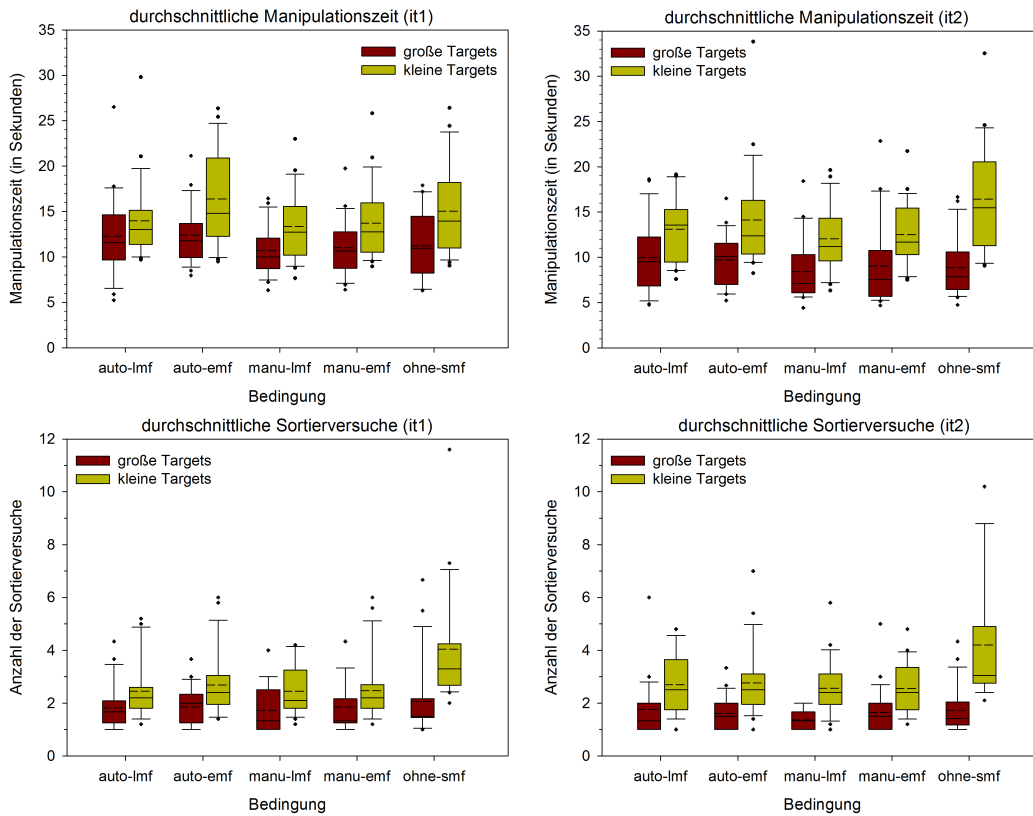


Abbildung 6.2.9.: Boxplots der durchschnittlichen Manipulationszeit und Anzahl der Sortierversuche für die großen und kleinen Targets entsprechend der Bedingungen und Interaktionstechniken.

und ohne IS (it1: $\bar{x}_{mIS}=2,25 \pm 0,091$ $\bar{x}_{oIS}=3,302 \pm 0,385$ bzw. it2: $\bar{x}_{mIS}=2,25 \pm 0,089$ $\bar{x}_{oIS}=3,27 \pm 0,325$). Außerdem ist ein Unterschied zwischen den großen und kleinen Targets bei allen Bedingungen erkennbar (vgl. Abbildung 6.2.9). Bezogen auf die kleinen Targets ist der Unterschied zwischen IS und ohne-smf sehr signifikant ($\chi^2(4)_{it1,small}=27,44$ und $\chi^2(4)_{it2,small}=19,02$ $p \leq 0,001$), bezüglich der großen Targets ist der Unterschied nicht signifikant ($\chi^2(4)_{it1,big}=2,44$ $p=0,66$ und $\chi^2(4)_{it2,big}=4,06$ $p=0,4$).

Fehlerrate Die geringste relative Fehlerrate wurde mit dem manuellen Cursorwechsel, sowohl bei den großen als auch bei den kleinen Zielobjekten erreicht (vgl. Abbildung 6.2.10). Interessanterweise war die Fehlerrate bei den kleinen Targets mit manu-emf geringer, während bei den großen Targets diese mit manu-lmf kleiner war (big: $\bar{x}_{it1,lmf}=13\%$, $\bar{x}_{it1,emf}=19\%$ und $\bar{x}_{it2,lmf}=13\%$, $\bar{x}_{it2,emf}=15\%$ bzw. small: $\bar{x}_{it1,lmf}=33\%$, $\bar{x}_{it1,emf}=31\%$ und $\bar{x}_{it2,lmf}=36\%$, $\bar{x}_{it2,emf}=35\%$). Die großen Targets konnten erwartungsgemäß mit der Basisfunktion sehr gut einsortiert werden, während die Fehlerrate bei den kleinen Targets sehr hoch war (big: $\bar{x}_{it1,smf}=18\%$, $\bar{x}_{it2,smf}=13\%$ und small: $\bar{x}_{it1,smf}=43\%$, $\bar{x}_{it2,smf}=46\%$). Bei it2 ist die Fehlerrate bei den kleinen Targets mit dem manuellen Cursorwechsel signifikant geringer als mit der Basisfunktion ($\chi^2(4)_{it2,small}=14,60$ $p=0,006$). Bei it1 erreichen, bis auf auto-emf, alle Bedin-

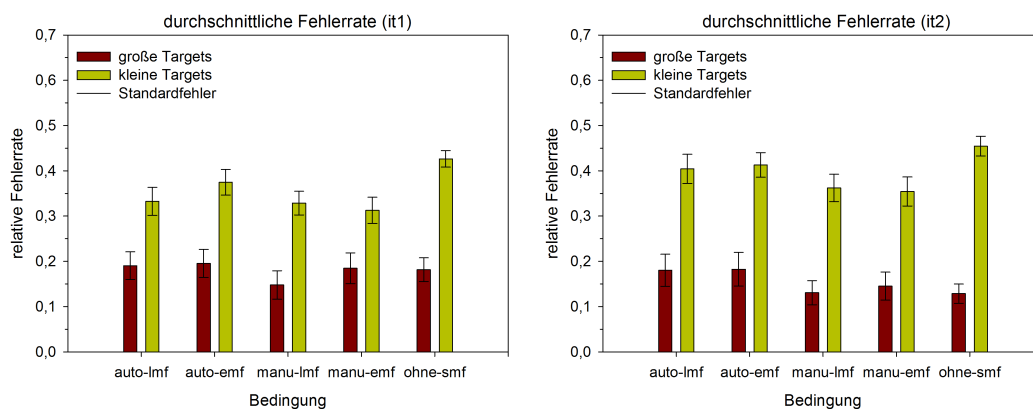


Abbildung 6.2.10.: Grafische Darstellung der durchschnittlichen relativen Fehlerrate inklusive Standardfehler abhängig von der Bedingung und dem Targettyp für it1 und it2.

gungen mit IS eine signifikant geringere Fehlerrate als ohne IS ($F(4, 84)_{it1,small}=5,12$ $p<0,001$).

Mit beiden Interaktionstechniken wurde eine zweifaktorielle Varianzanalyse mit Messwiederholung für die Fehlerrate durchgeführt, um die Wechselwirkung zwischen der Bedingung und dem Targettyp zu untersuchen. Es konnte eine signifikante Wechselwirkung bei it1 festgestellt werden ($F(4, 84)_{it1}=2,64$ $p=0,039$), d.h. der Unterschied zwischen den Targettypen ist bei jeder Bedingung sehr signifikant ($p<0,001$) und innerhalb der kleinen Targets signifikant zwischen ohne-smf und den Bedingungen manu-emf, manu-lmf und auto-lmf ($p\leq 0,01$). Bei it2 hingegen konnte nur ein sehr signifikanter Unterschied zwischen dem Haupteffekt Targettyp festgestellt werden ($F(1, 21)_{it2}=142,3$ $p<0,001$).

Insgesamt konnten bei it1 60% der kleinen Targets mit maximal zwei Sortierversuchen mit Hilfe des Interaction Scaling erfolgreich einsortiert werden und nur 45% ohne IS (vgl. Tabelle A.3.5). Beispielsweise benötigten die rotierten Targets Dreieck (ID 3) und Quadrat (ID 5) etwa $\frac{1}{3}$ weniger Sortierversuche mit IS als ohne IS (vgl. Abbildung 6.2.8). Bei it2 hingegen wurden mehr als 52% der kleinen Targets mit maximal zwei Sortierversuchen mit IS einsortiert und 49% ohne IS. Im Allgemeinen hatten die Probanden die meisten Schwierigkeiten die Zielobjekte mit der korrekten Größe und Orientierung (kleinen Dreiecken, kleiner Stern, große Box) einzupassen. Bei it2 wurden nur etwa 30% der kleinen Targets mit einer geringen Fehlerrate ($\leq 20\%$), sowohl mit der Basisfunktion als auch mit IS einsortiert. Insbesondere hatten die Probanden große Schwierigkeiten das kleine gedrehte Quadrat (ID 5) korrekt einzupassen, nur etwa 3 Probanden pro Bedingung gelang dies bei it2 mit der geringen Fehlerrate. Bei it1 hingegen konnte das gleiche Objekt von 6-11 Probanden mit der geringen Fehlerrate einsortiert werden. Bei it1 konnten weitgehend über 38% der kleinen Targets mit der geringen Fehlerrate mit IS einsortiert werden, während dies nur bei etwa 26% der kleinen Targets mit der Basisfunktion erreicht wurde. Eine Ausnahme bildet die Bedingung manu-emf, hier konnte die

Hälfte der kleinen Targets mit einer geringen Fehlerrate einsortiert werden (bei it1 57% und bei it2 50%; vergleiche Tabelle A.3.5).

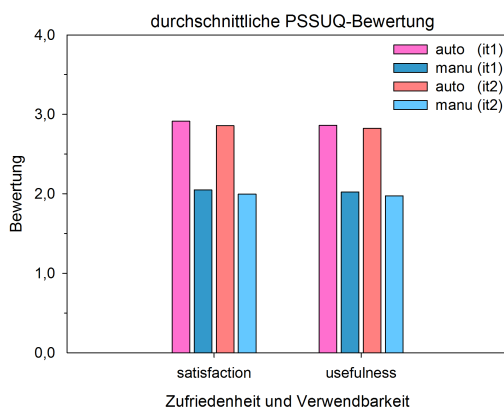
6.2.3.2. Zufriedenheit und Arbeitsaufwand

Mit Hilfe der Fragebögen sollten die Probanden ihre Arbeitsbelastung (NASA TLX), ihre Zufriedenheit (PSSUQ) und Präferenzen bezüglich der unterschiedlichen Methoden der Cursorwechsel und der verwendeten Interaktionstechniken bewerten. Die Vorgehensweise zur Auswertung der Fragebögen ist im Anhang A.2.4 beschrieben.

Zufriedenheit Die allgemeine Nutzerzufriedenheit war bei beiden Interaktionstechniken sehr gut, d.h. je zufriedener der Nutzer mit dem System oder der Technik ist, desto kleiner ist die gesamte Punktwertung (vgl. Abbildung 6.2.11a). Die Punktwertung zwischen den Interaktionstechniken ist gleich, während das manuelle Umschalten, von den Testpersonen etwas besser bewertet wurde, als das automatische Umschalten.

Präferenz Im Allgemeinen wurde das manuelle Umschalten bei beiden Interaktionstechniken bevorzugt (it1: 68%, it2: 77%), wobei bei it2 entweder auto oder manu bevorzugt wurden, während bei it1 4 Probanden beide Cursorwechsel präferierten. Insgesamt haben 7 Probanden (32%) bei it1 und 5 Probanden (23%) bei it2 den automatischen Cursorwechsel bevorzugt, und davon nur 3 Personen (14%) dies bei beiden Interaktionstechniken.

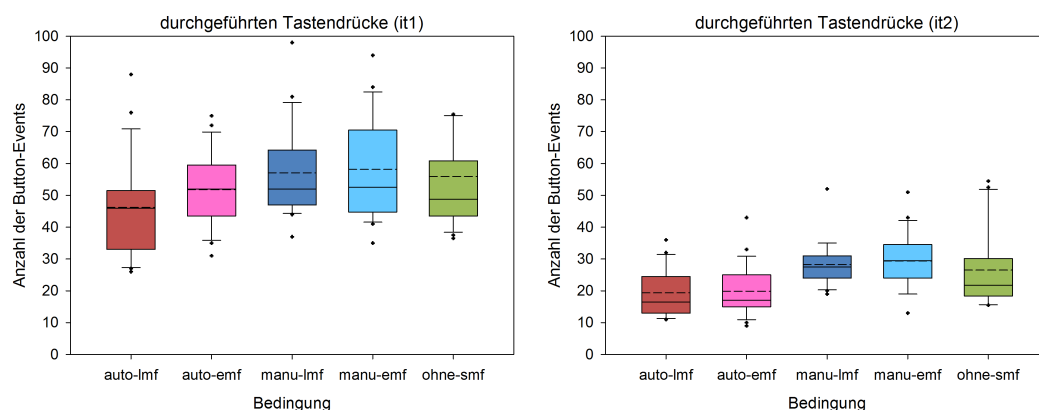
Bei der Selbsteinschätzung waren über die Hälfte der Probanden der Meinung mit dem manuellen Umschalten, die Aufgabe schneller (Schnelligkeit) und mit einer geringeren Fehlerrate (Genauigkeit) gelöst zu haben (siehe Tabelle 6.2.11b).



	Cursorwechsel	größere Schnelligkeit	höhere Genauigkeit	Präferenz
it1	automatisch	13,6 %	4,6 %	13,6 %
	manuell	59,1 %	54,5 %	68,2 %
	beide	27,3 %	40,9 %	18,2 %
it2	automatisch	18,2 %	18,2 %	22,7 %
	manuell	72,7 %	54,5 %	77,3 %
	beide	9,1 %	27,3 %	0 %

(a) Auswertung der PSSUQ-Fragen (von trifft voll zu (0) bis trifft überhaupt nicht zu (7)) (b) relative Häufigkeit der Antworten (N=22)

Abbildung 6.2.11.: Subjektive Bewertung der Interaktionstechniken und Varianten des Cursorwechsels im Posttest-Fragebogen.

Abbildung 6.2.12.: Boxplots der Anzahl der Button-Events pro Bedingung⁹.

Grundsätzlich wurden beide Interaktionstechniken ungefähr gleich bevorzugt, d.h. it2 von 45%, it1 von 41% und beide von 14% der Probanden. Dabei hat die Reihenfolge der Interaktionstechnik, mit der die Aufgabe gelöst wurde (Tag 1 vs. Tag 2), keinen Einfluss auf die Präferenz. Es haben 12 Probanden die Technik am 2. Tag bevorzugt, wobei die Hälfte it1 oder it2 angaben.

Button-Events Der motorische Aufwand ist bei der symmetrischen und asymmetrischen Interaktionstechnik design-bedingt unterschiedlich. Da it1 ein Umschalten in den SR-Mode für die Rotation und Skalierung erfordert, werden grundsätzlich mehr Manipulationsoperationen bzw. -schritte benötigt als bei it2. Beim manuellen Cursorwechsel erfordert das Umschalten zwischen der absoluten/relativen Cursorberechnung eine zusätzliche Manipulationsoperation im Vergleich zum automatischen Cursorwechsel oder der Basisfunktion. Folglich variiert die Anzahl der durchgeführten Button-Events zwischen den Bedingungen und den Interaktionstechniken (vgl. Abbildung 6.2.12). Bei it2 wurden die wenigsten Button-Events mit dem automatischen Cursorwechsel benötigt ($\bar{x}_{auto-lmf}=19$, $\bar{x}_{auto-emf}=20$), gefolgt von der Basisfunktion ($\bar{x}_{smf}=27$) und dem manuellen Cursorwechsel ($\bar{x}_{manu-lmf}=28$, $\bar{x}_{manu-emf}=29$). Gleichermaßen wurden die wenigsten Button-Events mit dem automatischen Cursorwechsel ($\bar{x}_{auto-lmf}=46$, $\bar{x}_{auto-emf}=52$) und die meisten Button-Events mit dem manuellen Cursorwechsel ($\bar{x}_{manu-lmf}=57$, $\bar{x}_{manu-emf}=58$) und der Basisfunktion ($\bar{x}_{smf}=56$) bei it1 erreicht. Der deutliche Unterschied zwischen automatischen und manuellem Cursorwechsel ist bei beiden Interaktionstechniken signifikant ($\chi^2(4)_{it1}=20,12$ $p<0,001$; $\chi^2(4)_{it2}=20,93$ $p<0,001$); ebenso unterscheiden sich der automatische Cursorwechsel und die Basisfunktion signifikant voneinander (siehe Paarvergleich mit SNK-Test im Anhang A.3.4).

Arbeitsbelastung Insgesamt wurden beide Interaktionstechniken bezüglich der gesamten Arbeitsbelastung (*workload*) von den Probanden gleich bewertet, wobei

⁹ Der extreme Ausreißer (140 Button-Events bei ohne-smf) bei it1 ist im Diagramm nicht dargestellt.

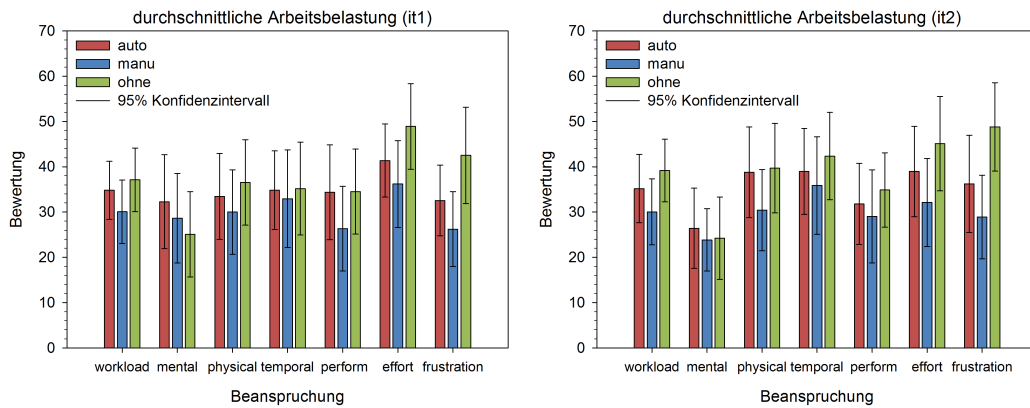


Abbildung 6.2.13.: Bewertung der durchschnittlichen Arbeitsbelastung für die unterschiedlichen Methoden des Cursorwechsels (ohne, auto, manu) innerhalb der einzelnen NASA TLX Skalen (von sehr niedrig (0) bis sehr hoch (100)) für die Interaktionstechniken.

die Arbeitsbelastung beim automatischen Cursorwechsel stets etwas höher angegeben wurde als beim manuellen Cursorwechsel (siehe Abbildung 6.2.13). Bei der Basisfunktion wurde die Frustration (*frustration*) und der Arbeitsaufwand (*effort*), sowohl bei it1 als auch bei it2, am höchsten eingestuft. Die Arbeitsbelastung wurde von den Probanden bei it2 etwas höher hinsichtlich der zeitlichen (*temporal*) und körperlichen (*physical*) Beanspruchung und der Frustration angegeben, insbesondere beim automatischen Cursorwechsel. Bei it1 wurde die geistige (*mental*) Beanspruchung, die Durchführung (*performance*) und der Arbeitsaufwand höher bewertet. Aufgrund der simultanen Manipulationsoperationen wird bei it2 ein erhöhter körperlicher Aufwand benötigt, der gleichzeitig den Frustrationsgrad erhöht (z.B. Ausgleichen einer ungewollten Skalierung bei gleichzeitiger Positionierung). Während bei it1 das Umschalten des Interaktionsmodus den Arbeitsaufwand und die geistige Beanspruchung erhöht, jedoch der Frustrationsgrad etwas geringer ausfällt (z.B. bleibt Positionierung beim Skalieren erhalten). Das manuelle Umschalten zwischen der Präzision der Interaktion erforderte aus Sicht der Probanden keine (erhöhte) körperliche oder geistige Beanspruchung bzw. Mühe.

6.2.3.3. Nutzerperformanz

Im Experiment soll die Nutzerperformanz der Manipulationstechniken mit IS und ohne IS bzw. den zwei Methoden des Cursorwechsels untersucht werden.

Für die Bestimmung der Nutzerperformanz wurden die Messwerte (1) der Effektivität und Effizienz (durchschnittliche Manipulationszeit, Sortiersuche und Fehlerrate), (2) des physiologischen Aufwandes (Anzahl der Button-Events, subjektive Arbeitsbelastung von *physical/mental/effort/frustration*) und (3) der Zufriedenheit (*TLX workload*) mit Hilfe der z-Transformation standardisiert. Je kleiner der z-Wert des Parameters ist, umso besser ist die Performanz (d.h. niedriger Wert entspricht guter Nutzerperformanz). Die Performanz der Teilaspekte Effizienz/Effek-

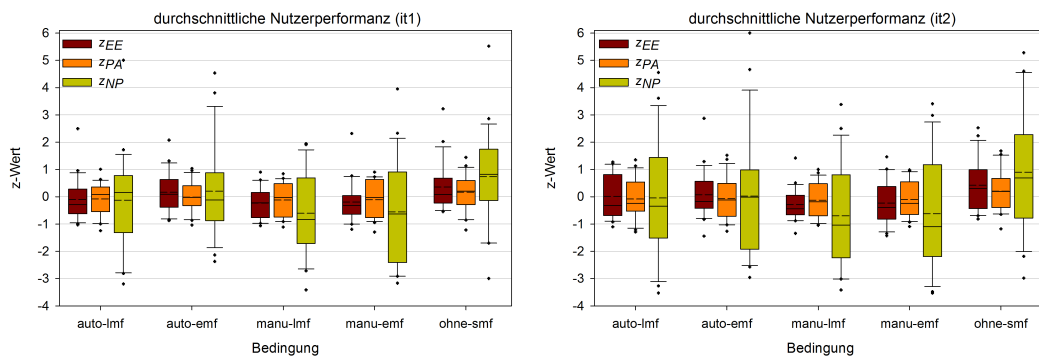


Abbildung 6.2.14.: Boxplots der durchschnittlichen Effizienz/Effektivität (z_{EE}), des durchschnittlichen physiologischen Aufwandes (z_{PA}) und der gesamten Nutzerperformanz (z_{NP}) pro Bedingung.

Effizienz/Effektivität (z_{EE}) und physiologischer Aufwand (z_{PA}) werden aus dem arithmetischen Mittel der zugehörigen z-Werte berechnet. Die TLX-Skalen der Arbeitsbelastung wurden aufgrund der Ergebnisse im Abschnitt 6.2.3.2 ausgewählt. Für die Zufriedenheit (z_{TLX}) wird der z-Wert der allgemeinen Arbeitsbelastung für die entsprechende Methode des Cursorwechsels verwendet. Da für die Basisfunktion keine PSSUQ-Bewertung erfasst wurde, wird diese bei der Nutzerperformanz nicht berücksichtigt. Die gesamte Nutzerperformanz (z_{NP}) einer Bedingung ist die Summe der z-Werte der Effizienz/Effektivität, dem physiologischen Aufwand und der Zufriedenheit (siehe Tabelle A.3.7).

Effizienz/Effektivität Die Effizienz/Effektivität unterscheidet sich zwischen den Bedingungen bei it1, d.h. mit IS waren die Probanden durchschnittlich effektiver und effizienter als ohne IS ($\chi^2(4)_{EE,it1}=12,04$ $p=0,017$). Das Gleiche konnte bei it2 festgestellt werden ($\chi^2(4)_{EE,it2}=13,35$ $p=0,01$), wobei zusätzlich ein signifikanter Unterschied zwischen dem automatischen Cursorwechsel und manu-lmf festgestellt wurde (siehe Abbildung 6.2.14).

Physiologischer Aufwand Bei beiden Interaktionstechniken war der physiologische Aufwand mit der Basisfunktion höher als beim manuellen oder automatischen Cursorwechsel (siehe Abbildung 6.2.14). Jedoch konnte nur eine Signifikanz bei it1 und nicht bei it2 festgestellt werden ($F(4, 84)_{PA,it1}=3,98$ $p=0,005$; $\chi^2(4)_{PA,it2}=7,38$ n.s.). Des Weiteren konnte kein Unterschied bezüglich des physiologischen Aufwandes zwischen it1 und it2 festgestellt werden, d.h. die erhöhte Anzahl der Button-Events bei it1 fällt nicht ins Gewicht ($F(1, 84)_{PA}=0,004$ n.s.).

Nutzerperformanz In Bezug auf die gesamte Nutzerperformanz konnte bei it1 und bei it2 mit Hilfe des Interaction Scaling die Performanz signifikant verbessert werden ($\chi^2(4)_{NP,it1}=19,82$ und $\chi^2(4)_{NP,it2}=20,18$ $p<0,001$).

physische Navigation (N=22)		it1	it2
Mobilität	vertikal	10	8
	horizontal	8	12
	Taktik	4	2
bevorzugte Distanz	nah	3	5
	nah-mittel	12	12
	mittel-entfernt	7	5
Motorik	wenig Wechsel	10	10
	viele Wechsel	12	12

(a) Anzahl der Teilnehmer zu den beobachteten Verhaltenskategorien

relative Häufigkeit (N=22) der tatsächlichen Ergebnisse (korrekten Einschätzung)			
	Cursorwechsel	Schnelligkeit	Genauigkeit
it1	automatisch	22,7 % (0/1)	13,6 % (0/3)
	manuell	63,6 % (7/12)	50,0% (8/13)
	beide	13,6 % (1/9)	36,4 % (5/6)
it2	automatisch	27,3 % (2/4)	18,2 % (2/4)
	manuell	59,1 % (8/12)	54,5 % (10/16)
	beide	13,6 % (1/6)	27,3 % (1/2)

(b) Auswertung der tatsächlichen objektiven Messwerte und der korrekten subjektiven Selbsteinschätzung in Klammern

Tabelle 6.2.2.: Häufigkeitsverteilung der Teilnehmer zu den beobachteten Verhaltenskategorien bei der physischer Navigation und die Auswertung der subjektiven Selbsteinschätzung mit den objektiven Messwerten bezüglich des Cursorwechsels.

Bei it1 wurde zusätzlich ein signifikanter Unterschied zwischen dem manuellen und automatischen Cursorwechsel festgestellt. Der Unterschied resultiert einerseits durch die schlechteren Performanzwerte bei der Effizienz/Effektivität und andererseits durch den höheren physiologischen Aufwand beim automatischen Cursorwechsel, insbesondere bei auto-emf (vgl. Abbildung 6.2.14).

Bei it2 hingegen ist zusätzlich ein signifikanter Unterschied zwischen dem automatischen Cursorwechsel und manu-lmf, sowie auto-lmf und manu-emf festzustellen. Der automatische Cursorwechsel erreichte durchschnittlich schlechtere Performanzwerte der Effizienz/Effektivität im Vergleich zum manuellen Cursorwechsel, während sich die Performanzwerte des physiologischen Aufwandes kaum unterschieden.

Es konnte kein Unterschied bezüglich der Methoden des Cursorwechsels zwischen it1 und it2 festgestellt werden, d.h. beide Interaktionstechniken sind entsprechend ihrer Vor- und Nachteile geeignet ($F(1,21)_{auto}=0,03$ $F(1,21)_{manu}=0,07$ $F(1,21)_{ohne}=0,01$ n.s.).

Bei der Selbsteinschätzung lagen bei it1 acht Probanden bezüglich der Genauigkeit und 13 Probanden bei der Schnelligkeit richtig, während bei it2 elf Probanden ihre Genauigkeit und 13 Probanden die Schnelligkeit korrekt einschätzen. Insgesamt haben mit dem manuellen Cursorwechsel 14 (it1) bzw. 13 Probanden (it2) die Aufgabe mit einer geringeren Fehlerrate bearbeitet, während fünf (it1) bzw. sechs Probanden (it2) dies mit dem automatischen Cursorwechsel erreichten. Die Aufgabenstellung konnte von elf (it1) bzw. zwölf Probanden (it2) mit dem manuellen Cursorwechsel schneller gelöst werden und von drei Probanden (it1, it2) mit dem automatischen Cursorwechsel. Bei den übrigen Probanden war jeweils kein Unterschied zwischen den beiden Methoden des Cursorwechsels bei der Schnelligkeit und Genauigkeit ersichtlich (siehe Tabelle 6.2.2b).

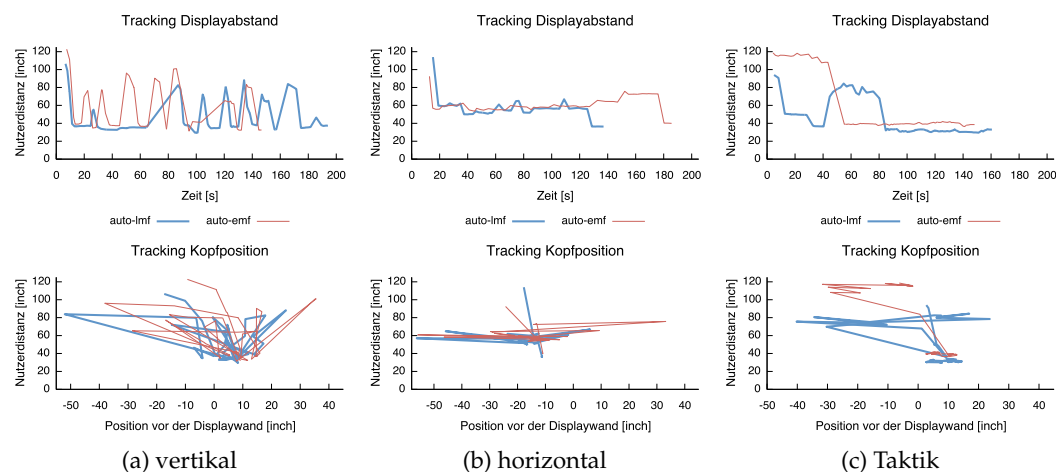


Abbildung 6.2.15.: Exemplarische Laufprofile für die beobachteten Verhaltensweisen bezüglich Mobilität.

Die Teilnehmer wurden bezüglich ihrer beobachteten physischen Navigation in unterschiedliche Kategorien eingeteilt (vgl. Tabelle 6.2.2a). Damit sollen Rückschlüsse zwischen physischer Navigation und geeignetem Interaction Scaling (Präzisionsfunktion, Cursorwechsel) für die Interaktionstechniken möglich sein. Repräsentative Laufprofile für die Kategorien sind im Anhang A.3.4.1 zu finden.

Mobilität: Im Wesentlichen konnten drei Verhaltensweisen bezüglich der Mobilität beim Lösen der Aufgabe beobachtet werden. Erstens, der Nutzer sortiert überwiegend jedes Objekt einzeln ein und er nutzt den gesamten verfügbaren Interaktionsraum, d.h. er läuft herum bzw. verwendet *vertikale Laufbewegungen* (siehe Abbildung 6.2.15a). Zweitens, der Nutzer bevorzugt eher einen konstanten Interaktionsabstand und er geht erst näher an das Display heran, wenn er nach den ersten Versuchen das Objekt nicht erfolgreich einsortieren konnte, d.h. er verwendet *horizontale Laufbewegungen* (siehe Abbildung 6.2.15b). Und drittens, der Nutzer verwendet eine bestimmte *Taktik*, indem er zunächst erst alle Objekte grob aus der Entfernung manipuliert und anschließend zur Feinjustierung an die Displaywand herangeht (siehe Abbildung 6.2.15c).

Bei it1 erreichten die Probanden mit vertikalen Laufbewegungen beim automatischen Cursorwechsel die besseren Performanzwerte bezüglich Effizienz/Effektivität, während beim manuellen Cursorwechsel die Personen mit horizontalen Laufbewegungen bzw. der Taktik effizienter und effektiver waren. Bei der Basisfunktion war die Taktik von Vorteil. Bei it2 waren grundsätzlich die Probanden mit vertikalen Laufbewegungen effizienter und effektiver, sowohl beim automatischen als auch beim manuellen Cursorwechsel. Da bei it2 die Taktik nur von zwei Probanden angewendet wurde, sind die Ergebnisse dazu nicht aussagekräftig. Es konnte kein Unterschied bezüglich der Anzahl der Sortiersuche und der Mobilität

bei it1 identifiziert werden, da die Probanden aller Kategorien durchschnittlich etwa 2,3 Sortierversuche benötigten ($F(2, 76)_{it1}=0,30$ n.s. bzw. $F(2, 19)_{auto,it1}=0,13$ n.s., $F(2, 19)_{manu,it1}=0,11$ n.s.). Bei it2 variierte die durchschnittliche Anzahl der Sortierversuche zwischen den Kategorien der Mobilität, aber nicht signifikant ($F(2, 76)_{it2}=0,46$ n.s. bzw. $F(2, 19)_{auto,it2}=1,12$ n.s., $F(2, 19)_{manu,it2}=1,66$ n.s.). Darüber hinaus wurde überprüft, ob ein Unterschied zwischen den Probanden mit horizontalen Laufbewegungen besteht, die im Nahbereich (nah, nah-mittel) bzw. der Schwelldistanz (mittel-entfernt) agierten. Dies war nicht der Fall, denn in allen Gruppen¹⁰ benötigten die Probanden durchschnittlich zwei Sortierversuche beim manuellen Cursorwechsel mit it1 und it2.

Insgesamt erreichten die Probanden, die häufig vor- und rückwärts liefen beim automatischen Cursorwechsel durchschnittlich die bessere Nutzerperformanz bei it1 und it2 (siehe Abbildung 6.2.16). Bei it2 galt dies auch beim manuellen Cursorwechsel. Es wurde eine signifikante Wechselwirkung zwischen Mobilität und Bedingung festgestellt ($F(8, 76)_{it2}=2,25$ $p=0,033$). Das heißt, innerhalb der Bedingung manu-lmf erreichten die Probanden mit horizontalen Laufbewegungen eine bessere Nutzerperformanz, als jene, die sich vertikal zur Displaywand bewegten. Bei den Probanden, die vertikalen Laufbewegungen verwendeten war die Nutzerperformanz mit IS signifikant besser als bei der Basisfunktion (vgl. Abbildung A.3.3).

Bei it1 unterscheidet sich die durchschnittlich gute Nutzerperformanz beim manuellen Cursorwechsel zwischen vertikalen und horizontalen Laufbewegungen nicht. Jedoch sind die Medianwerte bezüglich der Nutzerperformanz bei it1 anders als das arithmetische Mittel. Die Medianwerte zeigen, dass mit einer konstanten Interaktionsdistanz eine sehr gute Nutzerperformanz beim automatischen und manuellen Cursorwechsel erreicht wurde (vgl. Tabelle A.3.7). Dies lässt sich auf die subjektive Bewertung zurückführen, da diese Probanden im Allgemeinen eine geringe Arbeitsbelastung (z.B. körperliche Beanspruchung) für die Interaktionstechnik angaben.

Bevorzugte Distanz: Mit Hilfe der Laufprofile wurde der bevorzugte Displayabstand D des Probanden bestimmt, bei dem dieser hauptsächlich interagiert: *nah* ($D < 35$ inch); *nah-mittel* ($35 \geq D < 59$ inch); *mittel-entfernt* ($59 \geq D < 81$ inch)¹¹. Entfernte Distanzen ($D \geq 81$ inch) wurden selten von Probanden zum Einsortieren der Objekte genutzt.

Bei einer nah-mittel Distanz variiert der distanzabhängige Mappingfaktor von 0.4 – 0.7 (lmf) bzw. 0.3 – 0.5 (emf), was beim automatischen Cursorwechsel effizient und effektiv für die Probanden mit it1 war. Für das manuelle Umschalten konnten Probanden, die nah oder mittel-entfernt interagierten bessere z-Werte der Effektivität/Effizienz erreichen. Bei it2 waren die Probanden mit bevorzugter nah-mittel

¹⁰ Von den Probanden mit bevorzugten horizontalen Laufbewegungen haben bei it1 drei Probanden im Nahbereich/Schwelldistanz (nah, nah-mittel) gearbeitet und bei it2 sieben Probanden; die restlichen Probanden agierten jeweils bei einer mittel-entfernten Distanz.

¹¹ bzw. *nah* ($D < 90$ cm); *nah-mittel* ($90 \geq D < 150$ cm); *mittel-entfernt* ($150 \geq D < 200$ cm)

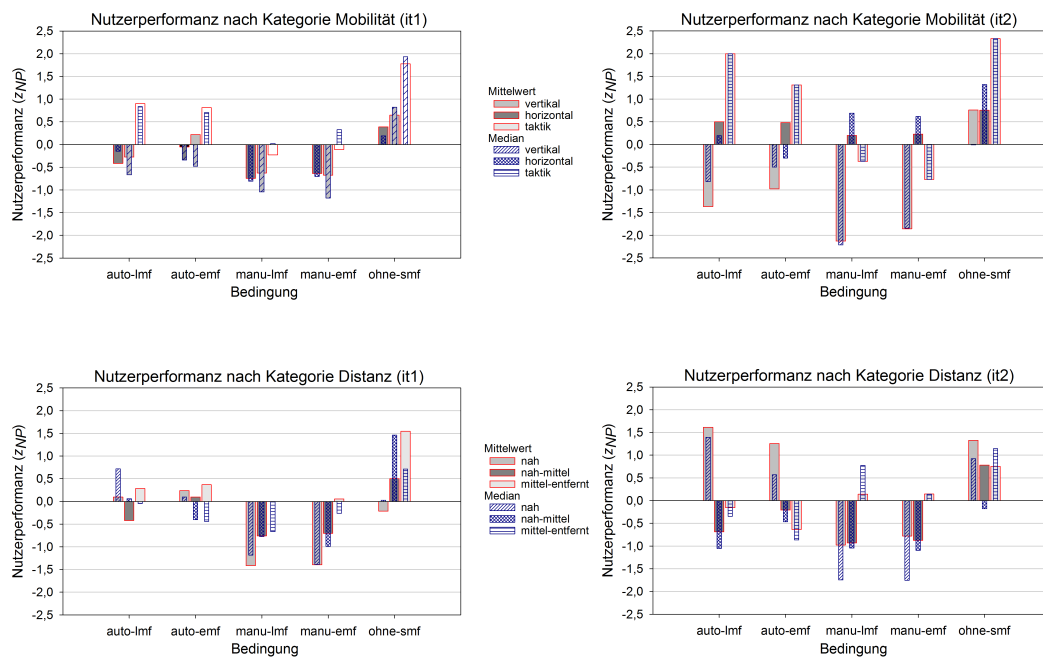


Abbildung 6.2.16.: Grafische Darstellung der mittleren und durchschnittlichen Nutzerperformanz entsprechend der Kategorien Mobilität und bevorzugter Distanz.

Distanz bei beiden Varianten des Cursorwechsels durchschnittlich effizienter und effektiver, mit Ausnahme von auto-emf. Hier erreichten die Probanden mit mittel-entfernter Distanz bessere Performanzwerte. Betrachtet man die Medianwerte von z_{EE} , dann ist erkennbar, dass sich beim manuellen Cursorwechsel die Performanzwerte nach bevorzugter Distanz bei it2 ähneln, wohingegen bei it1 die nah-mittel Distanz weniger effektiv/effizient war. Beim automatischen Cursorwechsel war bei beiden Interaktionstechniken bei der nah-mittel Distanz der lineare Mappingfaktor effizienter/effektiver, während bei mittel-entfernter Distanz nur ein sehr geringer Unterschied zwischen den Mappingfunktionen besteht (vgl. Tabelle A.3.7).

Im Allgemeinen war das Interagieren nah am Display mit automatischem Cursorwechsel weniger effektiv/effizient, was entweder aus der geringen Anzahl der Probanden oder dem erhöhten Nachgreifen beim Manipulieren (Clutching-Effekt) resultiert. Für die Basisfunktion sind die Performanzwerte der Effektivität/Effizienz bei it2 für alle Distanzen ähnlich, während bei it1 eine nah-mittel Distanz von Vorteil war (siehe Abbildung 6.2.16).

Durchschnittlich war die gesamte Nutzerperformanz bei bevorzugter nah-mittel Distanz bei it1 mit beiden Methoden des Cursorwechsels sehr gut. Beim manuellen Cursorwechsel erreichten die drei Probanden mit bevorzugter nah Distanz die kleinsten z-Werte, die jedoch auf die sehr geringe Beanspruchung beim physiologischen Aufwand zurückzuführen sind. Bei it1 zeigen die Medianwerte von z_{NP} , dass bei auto-emf und manu-lmf zwischen nah-mittel und mittel-entfernt die Nutzerperformanz gleich ist, wohingegen bei auto-lmf die mittel-entfernte Distanz

bzw. bei manu-emf die nah-mittel Distanz eine bessere Nutzerperformanz erzielen.

Bei it2 wurde eine sehr gute Nutzerperformanz mit dem manuellen Cursorwechsel von den Probanden erreicht, die eine nah oder nah-mittel Distanz bevorzugten. Beim automatischen Cursorwechsel war die Nutzerperformanz bei den Distanzen nah-mittel mit der linearen und mittel-entfernt mit der exponentiellen Mappingfunktion am besten. Die statistische Auswertung ergab, dass bei der Gruppe nah-mittel Distanz eine signifikant bessere Nutzerperformanz mit IS erreicht wurde als ohne IS, sowie bei der nah Distanz mit dem manuellen Cursorwechsel im Gegensatz zur Basisfunktion oder dem automatischen Cursorwechsel ($F(8, 76)_{it2}=2,42$ $p=0,022$). Jedoch ist die Voraussetzung der Normalverteilung bei der Varianzanalyse verletzt¹².

Die gesamte Nutzerperformanz war bei it1 und it2 mit der Basisfunktion wesentlich schlechter.

Motorik: Wenn die Anzahl der Umschaltungen¹³ beim manuellen Cursorwechsel zwischen absoluter/relativer Cursorberechnung maximal 20 für beide distanzabhängigen Mappingfunktionen ist, dann hat der Proband wenig Wechsel benötigt und somit einen geringeren motorischen Aufwand.

Die Verwendung von vielen oder wenigen Umschaltungen der Interaktionspräzision beim manuellen Cursorwechsel hat kaum einen Einfluss auf die Effizienz/Effektivität; bei it1 und it2 wurden geringe z-Werte erreicht. Ansonsten ist bei it1 erkennbar, dass ein häufiges Umschalten durchaus vorteilhaft ist. Das häufige Umschalten wirkt sich nicht auf den physiologischen Aufwand aus. Einerseits erreichten sowohl Probanden mit vielen Wechseln eine geringe Anzahl an Button-Events, als auch Probanden mit wenigen Wechseln eine hohe Anzahl an Button-Events benötigten. Andererseits wurde das Umschalten von den Probanden nicht als erhöhter physiologischer Aufwand wahrgenommen.

6.2.4. Diskussion

Im Experiment wurde untersucht ob Interaction Scaling einen positiven Einfluss auf die Performanz der Nutzer hat. Die Ergebnisse zeigen, dass bei der gewünschten sehr hohen Präzision Interaction Scaling die Nutzerperformanz bei den vorgestellten Manipulationstechniken deutlich verbessert. Neben dem signifikanten Unterschied zwischen der Nutzerperformanz mit IS und ohne IS bei beiden Interaktionstechniken, war zusätzlich der manuelle Cursorwechsel gegenüber dem automatischen Cursorwechsel signifikant besser bei der asymmetrischen Interaktionstechnik (it1). Bei der symmetrischen Interaktionstechnik (it2) war die Nut-

¹² SigmaPlot unterstützt keine zweifaktorielle Varianzanalyse an Rängen.

¹³ Der Schwellwert wurde als mindestens ein Cursorwechsel pro Objekt plus zwei zusätzlichen Wechsel als Toleranz für jeden Durchlauf festgelegt $((8 + 2) * 2 = 20)$.

zerperformanz beim manuellen Umschalten mit linearem Mappingfaktor signifikant besser als der automatische Cursorwechsel. Allerdings unterschieden sich die Interaktionstechniken bezüglich der Nutzerperformanz untereinander kaum (vgl. Abbildung 6.2.14). Es sind hauptsächlich die Präferenz oder das Verhalten die bestimmen, welche Technik und Mappingmethode für den jeweiligen Nutzer geeignet ist.

Die fehlende Anwendung der Cursor Resets im Experiment spricht für eine gute Cursorberechnung. Einerseits wurde auf die Präferenz der Nutzer mit der Möglichkeit des manuellen Umschaltens der Präzision eingegangen und andererseits beim automatischen Umschalten die hybride Cursorberechnung verbessert, indem der Drift im Nahbereich automatisch reduziert wird.

Hypothese 1 Das Experiment konnte zeigen, dass die Nutzer die Aufgabenstellung mit Hilfe von IS effizienter und effektiver bearbeiten konnten. Sowohl mit it1 als auch mit it2 benötigten die Versuchspersonen durchschnittlich kürzere Bearbeitungszeit, weniger Sortiersuche und eine geringere Fehlerrate mit beiden distanzabhängigen Mappingfunktionen in Kombination mit dem manuellen und automatischen Cursorwechsel, als mit der direkten 1:1 Mappingmethode – Hypothese 1 bestätigt. Dies lag hauptsächlich an der benötigten hohen Genauigkeit von ± 1 mm der kleinen Zielobjekte, die eine höhere Fehlerrate und mehr Selektionen bewirkten, während die großen Zielobjekte mit der Basisfunktion gut einsortierbar waren. Darüber hinaus waren die Probanden mit der Kombination manueller Cursorwechsel und linearer Mappingfunktion bei it2 signifikant effizienter und effektiver verglichen mit dem automatischen Cursorwechsel.

Hypothese 2 Mit Hilfe der unterschiedlichen Genauigkeit zwischen großen und kleinen Zielobjekten konnte gezeigt werden, dass die erforderliche Präzision einen Einfluss auf die Performanz hat. Es konnte ein signifikanter Unterschied beim Einsortieren zwischen den Targettypen groß und klein festgestellt werden, d.h. die großen Zielobjekte wurden schneller erfolgreich einsortiert – Hypothese 2 bestätigt. Die Fehlerrate beim Einsortieren der großen Zielobjekte unterschied sich zwischen den Bedingungen nicht signifikant – Hypothese 2b bestätigt. Bei den kleinen Zielobjekten hingegen war die Fehlerrate signifikant kleiner mit IS bei beiden Interaktionstechniken, mit Ausnahme von auto-emf bei it1 – Hypothese 2a bestätigt für it2 und abgelehnt für it1. Da bei auto-emf der Mappingfaktor beim Vorwärtslaufen schnell kleiner wird, erfordert dies für die Skalierung/Rotation ein Nachgreifen bei einigen Probanden, während das Objekt im Targetcontainer schon abgelegt wurde.

Außerdem stellte sich heraus, dass grundsätzlich die geringste Fehlerrate mit manuellem Cursorwechsel erreicht wurde, wobei für die großen Targets der lineare Mappingfaktor und für die kleinen Targets der exponentielle Mappingfaktor geeigneter waren. Mit der Kombination manu-emf konnte mindestens die Hälfte der kleinen Targets mit einer geringen Fehlerrate ($\leq 20\%$) erfolgreich einsortiert wer-

den. Im Allgemeinen konnten mehr Probanden mit it1 die kleinen Targets mit nur zwei Sortiersuchen mit IS erfolgreich einsortieren (15%), als dies der Fall bei it2 war (3%). Beim manuellen Cursorwechsel probierten die Probanden oftmals die Objekte zunächst mit absoluten Mapping einzusortieren und wechselten erst beim zweiten Versuch in das relative Mapping. Dies führt bei it2 aufgrund der simultanen Manipulationsoperationen dazu, dass beim Einsortieren von kleinen Targets mit absoluten Mapping das Objekt beim Loslassen unbewusst gedreht oder skaliert wird. Beim Umschalten in das relative Mapping berechnet die exponentielle Mappingfunktion einen sehr kleinen Mappingfaktor, mit dem die vorherige ungewollte Bewegung mit minimalen Aufwand korrigiert werden konnte.

Die meisten Schwierigkeiten hatten die Probanden mit dem korrekten Anpassen der Größe und Orientierung der Objekte. Dies zeigt sich insbesondere beim Einsortieren des kleinen rotierten Quadrates (Target 5, vgl. Abbildung 6.2.8), welches sich mit it2 schwierig gestaltete. Dabei handelt es sich um ein sehr kleines Objekt, das sich aufgrund der simultanen Bewegungsabläufe schwierig auf die erforderliche Größe skalieren und gleichzeitig rotieren lässt. Das Objekt erforderte von den Probanden eine mehrfache Skalierung, damit die Wimotes nicht aneinanderstießen.

Hypothese 3 Bei beiden Interaktionstechniken benötigten die Probanden mit dem automatischen Cursorwechsel signifikant weniger Button-Events als mit dem manuellen Cursorwechsel – Hypothese 3 bestätigt. Dies lag vor allem daran, dass die Mehrzahl der Probanden jedes Objekt einzelnen zunächst grob manipulierte (absolutes Mapping), dann die Feinjustierung vornahm zum Einsortieren (relative Mapping) und anschließend wieder in das absolute Mapping wechselte um das nächste Objekt zu selektieren. Nur etwa zwei bis vier Probanden haben die Taktik angewendet, die eine minimale Anzahl an Umschaltungen erforderte und sich somit kaum vom automatischen Cursorwechsel unterschied (z.B. Testperson 15 mit $x_{auto-lmf,it2}=32$ und $x_{manu-lmf,it2}=31$).

Hypothese 4 Die Anzahl der Button-Events wirkt sich auf den motorischen Aufwand der Nutzer aus, dass ein Kriterium für den physiologischen Aufwand darstellt. Außerdem spielten die subjektiven Arbeitsbelastungen, insbesondere die geistige und körperliche Beanspruchung und die Mühe eine Rolle. Das Experiment hat gezeigt, dass der physiologische Aufwand für den Nutzer mit Hilfe von Interaction Scaling bei it1 signifikant verringert werden konnte, jedoch nicht bei it2 – Hypothese 4 für it1 bestätigt und für it2 abgelehnt. Dies liegt vor allem daran, dass die Arbeitsbelastung beim manuellen Cursorwechsel etwas geringer als beim automatischen oder der Basisfunktion angegeben wurde, sich jedoch die Anzahl der Button-Events bei it2 zwischen der Basisfunktion und dem manuellen Cursorwechsel nur gering unterscheiden. Des Weiteren konnte kein Unterschied bezüglich des physiologischen Aufwandes zwischen it1 und it2 festgestellt werden – Hypothese 5a abgelehnt. Bei it1 wurde die geistige Beanspruchung, die Mühe und die allgemeine

Durchführung höher bewertet, was vor allem an der Trennung zwischen den Interaktionstasks Positionierung und Skalierung/Rotation liegt. Wohingegen bei it2 die simultanen Bewegungen ein einfaches und intuitives Interagieren ermöglichen. Da die subjektive Arbeitsbelastung beim physiologischen Aufwand stärker gewichtet wurde, fällt der design-bedingte große Unterschied der Anzahl der Button-Events zwischen it1 und it2 weniger stark ins Gewicht.

Zusätzlich hat die Bewertung der Performanz bei it1 aufgezeigt, dass der physiologische Aufwand bei Probanden mit vielen Umschaltungen beim manuellen Cursorwechsel geringer ist, denn nicht das Umschalten hat die vielen Button-Events hervorgerufen, sondern die Manipulation der Objekte im Allgemeinen (Selektion, SR-Mode). Daraus ergibt sich, dass der automatische Cursorwechsel beispielsweise dann dem manuellen Cursorwechsel vorzuziehen ist, wenn ein häufiges Umschalten über einen längeren Zeitraum erforderlich wäre und somit der physische Aufwand für den Nutzer spürbar steigt.

Grundsätzlich wurde der manuelle Cursorwechsel von den Probanden besser bewertet als der automatische Cursorwechsel (geringere Arbeitsbelastung, höhere Zufriedenheit und Präferenz). Dies lässt sich vor allem darin begründen, dass das manuelle Umschalten zwischen grober und feiner Präzision der Interaktion den meisten Probanden bekannt ist (z.B. Erfahrung, Vorwissen), während eine automatische Anpassung etwas Übung benötigt. Ebenso schätzten die Teilnehmer sich selbst so ein, die Aufgabe genauer und schneller mit manuellem Cursorwechsel bewältigt zu haben. Tatsächlich war dies auch bei etwa der Hälfte der Probanden der Fall, jedoch waren durchaus einige Probanden (23%) mit dem automatischen Cursorwechsel effektiver und effizienter als sie annahmen.

Im Rahmen des Experimentes wurde ebenfalls untersucht, ob sich Rückschlüsse aus dem Verhalten der Nutzer zu geeigneten Mappingmethoden, d.h. Methode des Cursorwechsels und Mappingfunktion treffen lassen. Dazu wurden die Ergebnisse der Teilnehmer in Bezug auf ihre Mobilität und bevorzugter Interaktionsdistanz untersucht. Im Allgemeinen erreichten die Probanden, die die Tiefe des Interaktionsraumes vor der Displaywand beim Interagieren nutzten (vertikale Laufbewegung) die bessere Nutzerperformanz beim automatischen Cursorwechsel. Die vertikalen Laufbewegungen waren bei it1 mit automatischem Cursorwechsel effizienter und effektiver, während beim manuellen Cursorwechsel die Probanden mit horizontalen Laufbewegungen zur Displaywand (konstanter Abstand) besser agierten. Die Taktik hat sich vor allem bei der Basisfunktion bewährt. Bei it2 hingegen waren die vertikalen Laufbewegungen sowohl beim automatischen als auch beim manuellen Cursorwechsel effizienter und effektiver, was sich ebenfalls in der gesamten Nutzerperformanz zeigte. Wenn Probanden horizontale Laufbewegungen bevorzugten, dann erreichten sie mit Hilfe der linearen Mappingfunktion ebenfalls sehr gute Performanzwerte.

Hypothese 5 Es konnte jedoch nicht nachgewiesen werden, dass die Probanden weniger Sortiersversuche aufgrund ihres Verhalten (Mobilität) benötigten. Das heißt weder vertikale Laufbewegungen beim automatischen Cursorwechsel noch horizontale Laufbewegungen beim manuellen Cursorwechsel sind (allein) entscheidend für die Anzahl der benötigten Sortiersversuche – Hypothesen 5, 5a und 5b abgelehnt.

Ein effizientes Verhalten bei der physischen Navigation bedarf Erfahrung mit der Interaktion in GHD-Umgebungen. Beispielsweise hat sich gezeigt, dass die Taktik keine offensichtliche Strategie beim Lösen der Aufgabenstellung darstellte. Bei it2 haben zwei Probanden die Taktik bezüglich der Mobilität angewendet und diese am zweiten Tag bei it1 ebenso, wobei hier zusätzlich zwei weitere Probanden hinzukamen. Dies lässt vermuten, dass diese Strategie aufgrund der Aufgabenstellung angewendet wurde und den Probanden wenig herumlaufen ermöglichte bzw. eine gewisse Effizienz.

Hinsichtlich des individuellen Abstandes zur Displaywand wurden drei Interaktionsbereiche identifiziert. Etwa die Hälfte der Probanden haben hauptsächlich die nah-mittel Distanz zum Interagieren genutzt, während die restlichen Probanden bei it2 entweder sehr nah oder mittel-entfernt agierten und bei it1 eher eine mittel-entfernte Distanz nutzten. Bei bevorzugter nah-mittel Distanz waren die Probanden mit automatischen Cursorwechsel effektiver und effizienter. Für die anderen Distanzen bot sich eher ein manueller Cursorwechsel an, da vor allem sehr nah am Display durch die Grobmanipulation der Clutching-Effekt reduziert wurde. Bei mittel-entfernter Distanz war mit auto-lmf und bei nah-mittel Distanz mit manu-emf die Nutzerperformanz insgesamt sehr gut. Bei it2 hat sich gezeigt, dass im Nahbereich mit manuellem Cursorwechsel die Nutzerperformanz besser war. Für den automatischen Cursorwechsel war bei nah-mittel Distanz die lineare Mappingfunktion und bei mittel-entfernt Distanz die exponentielle Mappingfunktion von Vorteil. Insbesondere bei bevorzugter mittel-entfernt Distanz haben die Probanden eher entfernt gearbeitet (> 81 inch), so dass ein 1:1 Mapping stattfand und mit nur einen Schritt vorwärts erreichten sie einen wesentlich kleineren Mappingfaktor (z.B. 0.6 bei 65 inch), der wahrscheinlich ausreichend war.

Laut Abschnitt 2.2.2 ist ein Kriterium der Effektivität die effektive physische Navigation (Motorik und Mobilität). Dies konnte im Experiment nicht direkt gemessen werden, da noch Erkenntnisse fehlen, die eine gute physische Navigation abschätzen. Mit Hilfe der hier durchgeführten Studien konnten bestimmte Verhaltensweisen beobachtet werden, die nicht nur individuelles Verhalten bei der Interaktion in GHD-Umgebungen reflektieren. Ein weiterer Aspekt ist der physiologische Aufwand als Kriterium für die Effizienz. In unserem Experiment wurde dieser hauptsächlich über die subjektive Bewertung der Arbeitsbelastung gemessen. Um eine möglichst unabhängige Beurteilung der Performanz für die Interaktionstechniken und der Kombination zwischen Mappingfunktion und Cursorwechsel bezüglich der Effizienz und Effektivität vornehmen zu können, wurden die objek-

tiven Messwerte der beiden Kriterien zusammengefasst und der Aspekt des physiologischen Aufwandes getrennt betrachtet.

Aufgrund eines Designfehlers im Experiment konnte die Bewertung der Zufriedenheit und Nützlichkeit (PSSUQ) nicht berücksichtigt werden. Das Kriterium Zufriedenheit wird bei der Nutzerperformanz nur indirekt über die Frustration bzw. die gesamte Arbeitsbelastung bestimmt.

Die Abbildung 6.2.14 verdeutlicht, dass die subjektive Bewertung der Arbeitsbelastung zwischen den Probanden stark variiert, so dass stets die objektive Performanz der Interaktionstechnik, als auch die subjektive Performanz beachtet werden muss. Zum Beispiel erreichten für it2 mit manu-lmf zwei Testpersonen (tp1, tp2) ähnliche z_{EE} -Werte ($z_{EE,tp1} = -0,7$ und $z_{EE,tp2} = -0,6$), aber sie gaben unterschiedliche Arbeitsbelastungen an ($z_{TLX,tp1} = 0,9$ und $z_{TLX,tp2} = -1,5$), so dass die gesamte Nutzerperformanz zwischen ihnen stark variiert ($z_{NP,tp1} = 0,8$ und $z_{NP,tp2} = -3,0$).

Da sich diese Verhaltensweisen zu physischer Navigation im Vorfeld des Experimentes schwer abschätzen ließen, erfolgte die zugehörige Datenerfassung unzureichend oder die Kategorien wurden nicht optimal gewählt. Beispielsweise war die Anzahl der Umschaltungen beim manuellen Cursorwechsel keine geeignete Kategorie für den motorischen Aufwand. Darüber hinaus müsste die Einteilung des bevorzugten Interaktionsabstandes detaillierter erfolgen und für die Methoden des Cursorwechsels getrennt betrachtet werden.

Weiterhin hatten große Testpersonen aufgrund der orthogonalen Projektion den Vorteil, dass sie sich wegen ihrer größeren Armreichweite weniger im Interaktionsraum bewegen mussten, um entfernte Objekte zu selektieren, während sich kleine Testpersonen weiter weg von der Displaywand entfernen mussten. Außerdem war zu beobachten, wenn der Proband eine feste Position bevorzugte, er bei der Objektselektion seinen Arm möglichst weit ausstreckte, anstatt einen Schritt zurück oder zur Seite zu treten. Diese Aspekte haben Einfluss auf den motorischen Aufwand, was in dieser Studie nicht berücksichtigt wurde.

Es ist von der Interaktionstechnik abhängig, ob das automatische oder das manuelle Umschalten bevorzugt wird und nicht personenabhängig. Während die Präferenz für die asymmetrische oder symmetrische Interaktionstechnik personenabhängig ist. Die simultanen Manipulationsoperationen bei it2 sind intuitiv und schnell zu lernen, erschweren jedoch gleichzeitig die präzise Skalierung und Positionierung. Wohingegen Probanden, die die Trennung von Interaktionsaufgaben gewöhnt sind, die Separierung zwischen der Positionierung und Skalierung/Rotation bei it1 als Vorteil empfanden. Dies führt dazu, dass beispielsweise das automatische Umschalten bei it2 stärker präferiert wird als bei it1.

STUDIE 3: Grundsätzlich hat sich gezeigt, dass bei beiden Interaktionstechniken der automatische Cursorwechsel mit linearem Mappingfaktor und der manuellen Cursorwechsel mit exponentiellem Mappingfaktor die besseren Performanzergebnisse lieferten. Da beim manuellen Umschalten der Präzision nur noch eine minimale Anpassung nach der Grobjustierung erfolgen muss, ist ein kleinerer Mappingfaktor effizienter. Beim automatischen Cursorwechsel erfolgt eine kontinuierliche Anpassung, somit ist ein größerer Mappingfaktor geeigneter, um die motorischen Bewegungen effektiver einzusetzen (Clutching).

6.3. Zusammenfassung

Damit Interaction Scaling mit distanzabhängigem Mappingfaktor funktioniert, müssen die Nutzer wissen und nachvollziehen können, wie die Präzision der Interaktion durch ihren Displayabstand angepasst wird (vgl. Studie 1).

Die automatische distanzabhängige Anpassung der Präzision der Manipulation ruft die Seiteneffekte Clutching und Drift hervor. Eine Möglichkeit zur Reduzierung des Drift-Effektes ist ein Zurücksetzen der virtuellen Cursorpositionen auf die tatsächlichen korrespondierenden realen Positionen (Cursor Reset). Jedoch ist diese Aktion nicht unbedingt intuitiv für die Nutzer (siehe Studie 1 und 2). Demzufolge wurde auf die Präferenz von Nutzern eingegangen, indem ein interaktives Abschalten der distanzabhängigen Präzision umgesetzt wurde (manueller Cursorwechsel). Eine weitere Methode zur Reduzierung des Drift-Effektes hat sich durch die Verwendung der hybriden Cursorberechnung gezeigt, indem eine absolute Cursorberechnung verwendet wird, wenn keine Manipulationsaktion stattfindet und eine relative Cursorberechnung, wenn ein Manipulationstask ausgeführt wird. Das heißt, eine taskbasierte Präzision der Interaktion ist sinnvoll.

Anhand der iterativen Studien konnte aufgezeigt werden, dass Interaction Scaling zunehmend die Nutzerperformanz verbessert, wenn die benötigte Präzision steigt.

Wird eine geringe Präzision der Interaktion benötigt (z.B. 5 mm), dann war bei beiden Manipulationstechniken ein absolutes Mapping ausreichend, da der Tremor durch die orthogonale Projektion ausreichend reduziert wird, kein Drift-Effekt auftritt und ein schnelles Arbeiten unterstützt wurde.

Wird die Präzision erhöht, dann wird ein relatives Mapping notwendig, um die motorischen Einschränkungen bei der 3D Interaktion zu kompensieren. Die Kombination des automatischen Cursorwechsels (hybride Cursorberechnung) mit einer distanzabhängigen linearen Mappingfunktion ist für eine mittlere Genauigkeit (4 bis 2 mm) geeignet. Da einerseits ungewollte Bewegungen, z.B. durch Loslassen der Selektionstaste, vermindert werden und andererseits ein linear veränderlicher

Mappingfaktor durch vor und zurück laufen für die Nutzer leicht nachvollziehbar ist.

Für eine sehr hohe Präzision (2 bis 1 mm) eignet sich der manuelle Cursorwechsel (absolute/relative Cursorberechnung) in Kombination mit einer distanzabhängigen exponentiellen Mappingfunktion. Der Nutzer kann den Mappingfaktor durch geringe vertikale Bewegungen interaktiv variieren.

In einer Folgestudie sollte untersucht werden, wie mehrere konstante Mappingfaktoren (z.B. Zonen Mappingfunktion) in Kombination mit dem manuellen Umschalten funktionieren.

Die subjektiven Eigenschaften (z.B. Präferenzen, Erfahrung) haben einen Einfluss auf die Nutzerperformanz, die aber unterschiedlich stark von den Nutzern gewichtet werden (siehe physiologischer Aufwand in Studie 3).

Die symmetrische Interaktionstechnik (it2) empfanden viele Nutzer als intuitiver, d.h. natürliche und einfache Handhabung, weniger geistig anspruchsvoll. Allerdings kommentierten einige Nutzer, dass sie mit der asymmetrischen Interaktionstechnik (it1) nach einer Eingewöhnungszeit sogar leichter zurechtkamen. Dies lag hauptsächlich daran, dass die Trennung der Interaktionstasks zwischen Selektion/Positionierung und Rotation/Skalierung eine präzisere Manipulation ermöglichte und sie die Technik auf Dauer als weniger körperlich anstrengend empfanden.

Die asymmetrische Interaktionstechnik verwendet entgegen Guiard's Framework [Gui87] die dominante Hand als räumliches Bezugssystem der nicht-dominanten Hand für die Skalierung/Rotation (vgl. Abschnitt 2.1.2). Vor dem Hintergrund, dass die dominante Hand für den Selektionstask verwendet wurde und somit die Manipulationsoperationen initiierte. Der Selektionstask erforderte eine feingranulare und präzise Interaktion aufgrund des absoluten Mappings, was als bedeutender festgelegt wurde. In einer Folgestudie sollte geklärt werden, ob die angepasste asymmetrische Interaktionstechnik nach Guiard's Framework effizienter ist als die verwendete Manipulationstechnik.

Bezüglich der Methode des Wechsels zwischen präziser und grober Interaktionengenauigkeit wurde der manuelle Cursorwechsel von den meisten Probanden bevorzugt, da die Nutzer das automatische Umschalten als Bevormundung empfanden. Des Weiteren mussten sich die Nutzer beim manuellen Cursorwechsel weniger bewegen. Im Gegensatz dazu bevorzugten einige Nutzer das automatische Umschalten, da sie es als einfacher empfanden auf das Systemverhalten durch physische Navigation zu reagieren, anstatt einen "weiteren Knopf zu lernen". Vor allem bei der asymmetrischen Interaktionstechnik war das automatische Umschalten hilfreich, da die Hände schon unterschiedliche Interaktionstasks durchführten und das Drücken von drei Tasten gleichzeitig eine hohe Konzentration erforderte.

Während der Studie 3 neigte etwa die Hälfte der Probanden dazu, sich während der Aufgabe mit dem Experimentator zu unterhalten. Dies lässt auf den geringen

mentalen Aufwand durch die physische Navigation schließen. Im Vergleich zu den Pilotstudien, wo der ein oder andere Proband Verspannungen im Schulterbereich aufgrund der ungewohnten 3D Interaktion spürte, war dies in Studie 3 nicht der Fall. Dies spricht einerseits für die verbesserte Cursorberechnung, als auch für die angepassten Mappingfunktionen.

Grundsätzlich haben die Studien gezeigt, dass die physische Navigation (Motorik und Mobilität) von den Nutzern unterschiedlich eingesetzt wird und die bevorzugte Interaktionsdistanz variiert.

Der automatische Cursorwechsel verlangte häufiges vor und zurück laufen von den Nutzern. Dies war aus Nutzersicht für die einfache Aufgabenstellung der Testanwendung nicht effizient. Daher sollte Interaction Scaling mit automatischem Cursorwechsel mit einer anderen Aufgabenstellung untersucht werden, bei der vertikale Bewegungen effizienter sind, z.B. in dem ein Knoten, in einem hierarchischen Graphen (vgl. Abschnitt 5.1.1) oder ein hierarchisches Puzzle (ähnlich [PNB09]) manipuliert werden soll.

In Studie 4 wurde aufgrund der perspektivischen Projektion von den Nutzern eine feste Position im Interaktionsraum bevorzugt. Daher ist die Anpassung der Präzision über die Betrachtungsdistanz nur bedingt effektiv. Hier sind voraussichtlich geschwindigkeitsabhängige oder stufenartige manuelle Umschaltungen der Interaktionspräzision geeigneter.

Die Verwendung der Wiimote als Eingabegerät ermöglicht es, Vibrationen als haptisches Feedback zu nutzen. Dies wurde in unseren Studien nicht berücksichtigt, um die Untersuchungsergebnisse auf unterschiedliche getrackte Eingabegeräte mit Buttons übertragen zu können. Jedoch bietet die Vibrationsfunktion die Möglichkeit, den Nutzer bei der distanzabhängigen Interaktion zu unterstützen, indem ein kurzes Vibrationssignal dem Nutzer den Wechsel der Interaktionsgenauigkeit beim Zonen Mapping anzeigt.

Um den physischen Interaktionsraum vor der Displaywand optimal nutzen zu können, muss die Markererkennung zuverlässig funktionieren. Im Setup Trackingbereich Altbau war eine Markererkennung sehr nahe an der Displaywand nicht möglich, so dass für die Interaktion ein Mindestabstand von ca. 45 cm notwendig war. Diese Einschränkung wurde mit dem Setup Trackingbereich Neubau behoben. Jedoch wurden während der Studien die Marker teilweise an den Ecken im Fernbereich nicht zuverlässig erkannt (vgl. Abschnitt 4.2). Dies hat einige Probanden in Studie 3, die weit zurück gelaufen sind, bei der Interaktion gestört, da die Cursors kurzzeitig eingefroren wurden. Die Installation zusätzlicher Kameras könnte die Probleme vermindern und eine zuverlässige Markererkennung im gesamten physischen Interaktionsraum ermöglichen, dies ist insbesondere für Mehrbenutzer-Anwendungen erforderlich.

In den durchgeführten Studien wurde Fitts' Law nicht berücksichtigt, da die Aufgabenstellung auf unterschiedliche Vorgehensweise mit den verwendeten Interak-

tionstechniken durchführbar war. Es war nicht vorhersagbar, wie die Teilnehmer die Aufgabe bearbeiten und sich bezüglich ihrer physischen Navigation verhalten (z.B. Strategien der Mobilität).

Die Verwendung von Ad-hoc-Stichproben in den Studien führt unter Umständen dazu, dass signifikante Unterschiede zwischen den Testbedingungen nicht aufgezeigt wurden, da kleine Stichprobengrößen, die z.T. hohe Varianzen in den Bedingungen zeigten, verwendet wurden. Es haben viele Probanden im Rahmen der Studien erstmalig mit einem GHD interagiert. Außerdem können die verwendeten Interaktionstechniken als schnell verfügbar und einfach zu bedienen bewertet werden, da die Studienteilnehmer die Aufgaben erfolgreich bewältigten.

7. Schlussbetrachtung

Große hochauflösende Displayumgebungen ermöglichen es Nutzern, Informationen auf eine neue Art und Weise wahrzunehmen und mit diesen anders zu interagieren als bei herkömmlichen Desktopumgebungen [AEYN11]. Dies hat Änderungen im Nutzerverhalten und bei den Interaktionsmöglichkeiten zur Folge, was wiederum bei der Informationsdarstellung und den verwendeten Interaktionstechniken in GHD-Umgebungen zu beachten ist.

Eine natürliche Interaktion mit der Informationsdarstellung bietet die physische Navigation, bei der Nutzer näher an das Display heran gehen, um detaillierte Informationen zu betrachten, bzw. sich von der Displaywand entfernen, um die Informationen in ihrem Kontext zu betrachten oder eine Überblickdarstellung zu erhalten. Demzufolge werden einerseits Aufgaben nah am Display mit detaillierten Informationen und andererseits Aufgaben, die besser aus der Entfernung ausgeübt werden durchgeführt [NBC06]. Diese beiden Aufgabenbereiche erfordern jedoch eine unterschiedliche Präzision und Sensitivität der Nutzereingabe.

Indem die großen hochauflösenden Displays unterschiedliche Darstellungsgenauigkeit durch den Betrachtungsabstand bieten, kann dieser Displayabstand außerdem für die Präzision der Interaktion mit der Visualisierung genutzt werden. Dabei haben wir untersucht, wie der aktuelle Abstand des Nutzers zur Displaywand für den Wechsel zwischen den unterschiedlichen Aufgabenbereichen genutzt werden kann.

Das Ziel der vorliegenden Arbeit war die Entwicklung von Methoden, die den Nutzer bei seiner (gegenwärtigen) Tätigkeit in großen Displayumgebungen unterstützen, indem die Visualisierung und Interaktivität an den Nutzer unter Berücksichtigung seines aktuellen Betrachtungsabstandes angepasst wird. In diesem Zusammenhang wurden drei Forschungsfragen gestellt und beantwortet (vgl. Abschnitt 1.1).

Um den Nutzer in intelligenten Umgebungen zu unterstützen, muss das Verhalten von Personen vorhersehbar sein, damit eine Unterstützung und keine Beeinträchtigung für den Nutzer entsteht. Erste Nutzerstudien zeigten, dass die physische Navigation bei Visualisierungs- und Navigationsaufgaben in GHD-Umgebungen die Nutzerperformanz verbessert [BN05b, BI12]. Manipulationsaufgaben, welche die physische Navigation der Nutzer vor der Displaywand unterstützen, wurden in der Forschung bisher wenig untersucht, insbesondere mit Hinblick auf die individuellen Eigenschaften der Nutzer (vgl. Abschnitt 2.4.2 und 2.5).

Hauptsächlich werden 3D Interaktionstechniken aus dem VR-Bereich für die Interaktion in GHD-Umgebungen adaptiert und erweitert.

Wie müssen die Verhaltensweisen der 3D Interaktionstechnik an den neuen physischen Interaktionsraum des großen hochauflösenden Displays angepasst werden?

In einem ersten Schritt wurde eine Analyse der Einflussfaktoren, die eine Rolle bei der Interaktion in GHD-Umgebungen spielen, vorgenommen. Die unterschiedlichen Einflussfaktoren lassen sich in einen Parameterraum mit vier Komponenten (physische Umgebung, virtuelle Umgebung, Benutzer und Mapping) einordnen. Daraus resultierend müssen die 3D Interaktionstechniken an die verfügbare Displaykonfiguration, u.a. an die Displayfläche, die Displayauflösung, die Displayanordnung und Bezels angepasst werden. Um die physische Navigation zu unterstützen, muss die Eingabemodalität mobil, schnell verfügbar, leicht anzuwenden und aus unterschiedlichen Entfernungen nutzbar sein. Des Weiteren sind die objektiven und subjektiven Eigenschaften der Benutzer bei der räumlichen Nutzereingabe zu beachten, z.B. visuelle Wahrnehmung, Tremor, Ermüdung, Nutzergruppe. Für die Abbildung der physischen Nutzereingabe auf die virtuelle Cursorbewegung sollte die Mappingmethode relativ unabhängig von der Eingabemodalität sein, die gegenwärtigen Interaktionsdistanz berücksichtigen und eine dynamische Anpassung des CD-Gain verwenden.

Der vorgestellte Ansatz des Interaction Scaling (IS) verwendet die physische Navigation für die distanzabhängige Informationsdarstellung und für die distanzabhängige Präzision der Nutzerinteraktion in GHD-Umgebungen. Die resultierende Mappingmethode für IS kombiniert zwei Ansätze: (1) die Berechnung des Mappingfaktors, welche die physischen Eingabebewegungen auf die virtuellen Ausgabebewegungen anpasst und (2) der Wechsel der Präzisionsstufen zwischen präziser und grober Genauigkeit. Bei der Berechnung des Mappingfaktors (CD-Gain) werden die Eigenschaften der Eingabemodalität wie Kinematik, Projektion und Isomorphie einbezogen.

Berechnung Mappingfaktor: Die Anpassung des Mappingfaktors kann global als konstanter vorberechneter Mappingfaktor erfolgen, oder lokal abhängig von der Nutzereingabe (geräte- oder distanzabhängig). Für Interaction Scaling wurden verschiedene distanzabhängige Mappingfunktionen mit kontinuierlichem Mappingfaktor (linear, exponentiell), mit vordefinierten festen Mappingfaktor (zonenartig) und hybride Mappingfunktionen untersucht.

Wechsel der Präzisionsstufen: Für das Umschalten zwischen den Präzisionsstufen wurden taskabhängige (relative und hybride Cursorberechnung) und geräteabhängige (*manual-switching*) Varianten untersucht. Für IS wird die taskbasierte (automatische) Anpassung der Präzisionsstufen verwendet.

Im Rahmen dieser Arbeit wurde sich auf Navigations- und Manipulationstechniken konzentriert. Mit Hilfe des Parameterraumes wurden der Head-Tracker für die

Navigation und der getrackte Nintendo Wii Remote Controller für die Manipulation als geeignetes Eingabegerät identifiziert. Die Eingabegeräte wurden in Kombination mit einem optischen Trackingsystem verwendet, um eine millimetergenaue Bewegungserfassung der Nutzereingabe zu ermöglichen. Anhand von Beispielszenarien wurden die Gewichtung der Einflussfaktoren diskutiert und welche Anpassung für die Mappingmethode daraus resultiert (vgl. Tabelle 3.4.5).

Welche Präzision wird in dem jeweiligen Nutzerszenario benötigt und wie direkt und unmittelbar muss die Nutzereingabe erfolgen?

Navigation Als Szenarien für die distanzabhängige Informationsvisualisierung wurde die Exploration eines 2D Graphen und die 3D Terrainvisualisierung gewählt. Als Navigationstechnik wurde jeweils Head-Tracking verwendet, wobei insbesondere das foveale und periphere Sehen bei der Informationsdarstellung zu berücksichtigen war. Für die 3D Terrainvisualisierung wurde neben der virtuellen Betrachtungsposition (Kamera) auch der reale Betrachtungsabstand (Benutzer) für die LOD-Berechnung der virtuellen Objekte genutzt. Dazu werden die Displaybereiche mit unterschiedlichem Detailgrad gerendert, die der aktuellen visuellen Wahrnehmung des Betrachters entsprechen (Sichtzellen). Für die Exploration eines hierarchischen 2D-Graphen wurden die Abstraktionsebenen auf den physischen Interaktionsraum abgebildet (distanzabhängiger Zoomfaktor), so dass der Detailgrad zur Displaywand hin zunimmt und der Umfang der Information und die Schriftgröße interaktiv an die Nutzerposition angepasst wird. Neben dem globalen Umschalten der Abstraktionsebenen (Zonen Mapping für Kontext der Informationsdarstellung) wurde eine zusätzliche lokale Umschaltung durch eine visuelle Linse ergänzt (absolute Cursorberechnung für Fokus der Informationsdarstellung). Die Nutzereingabe wurde auf Positionsdaten mit perspektivischer Projektion abgebildet (direkte Interaktion mit Approximation der Blickrichtung). Um ein ungewolltes Umschalten zwischen den Detailstufen der Informationsdarstellung aufgrund des Tremor zu reduzieren, wurden Zonen mit Schwellbereichen verwendet beziehungsweise die aktiven Sichtzellen im peripheren Sehfeld erweitert.

Einhändige Manipulation In den Beispielszenarien mit einhändiger Nutzereingabe wurde die Präzision der Manipulation an die gegenwärtige Aufgabe des Nutzers angepasst. Bei der Interaktion im Smart Meeting Room werden anwendungsunabhängige und anwendungsabhängige Manipulationsoperationen ausgeführt. In diesem Szenario wurde eine mausähnliche Interaktionstechnik gewünscht, welche die Rolle des Präsentators und des Zuhörers, sowie die heterogene Displayanordnung berücksichtigt. Daher wurde eine relative Nutzereingabe mit der Maus am persönlichen Gerät des Zuhörers (anwendungsabhängig) und der Wiimote im Raum durch den Präsentator (anwendungsunabhängig) umgesetzt (indirekte In-

teraktion). Interaction Scaling unterstützt die anwendungsunabhängige Manipulation, indem ein distanzabhängiger Mappingfaktor für die Interaktion mit geringerer Präzision auf dem Großdisplay verwendet wird und die anwendungsabhängige Manipulation mit Hilfe eines lokalen geräte-spezifischen Mappingfaktors für die präzise Interaktion am persönlichen Gerät.

Für das Szenario der einhändigen Manipulationstechnik in GHD-Umgebungen wurde der Fokus auf die Nutzergruppe Kinder (kleine Körpergröße) gesetzt. Diese Nutzergruppe erfordert eine absolute Nutzereingabe, welche ein schnelles und grobes Interagieren über die gesamte Displayfläche ermöglicht (direkte Interaktion). Jedoch ist das natürliche Handzittern durch die absolute Cursorberechnung mit perspektivischer Projektion sichtbar. Die Pilotstudie zeigte, dass bei der gewählten Präzision der Manipulation der Tremor tolerierbar war. Hier wurden unterschiedliche distanzabhängige Mappingfunktionen getestet, wobei die Subtask-abhängigen Mappingfaktoren (Kombination der Mappingfunktionen), aufgrund der vorwiegend konstanten Interaktionsposition vielversprechend waren.

Beidhändige Manipulation Für die beidhändige Manipulation von 2D Objekten wurden eine symmetrische und eine asymmetrische Manipulationstechnik umgesetzt, die eine sehr hohe Präzision der Interaktion unterstützen. Aufgrund der Displaykonfiguration und zur Reduzierung des Tremors verwenden die Interaktionstechniken eine orthogonale Projektion der Positionsdaten (direkte Interaktion). Im Rahmen der Nutzerstudien wurden verschiedene Mappingmethoden für Interaction Scaling getestet, d.h. diverse distanzabhängige Mappingfunktionen mit unterschiedlichen Varianten der Cursorberechnung (für den Wechsel der Präzisionsstufen). Für die distanzabhängige Manipulationspräzision eignet sich die hybride Cursorberechnung mit kontinuierlichem Mappingfaktor für beide Interaktionstechniken. Da die verwendeten Manipulationstechniken unterschiedliche Metaphern der Interaktion verwenden (Trennung zwischen Selektion/Positionierung und Rotation/Skalierung bzw. simultane Manipulationsoperationen) ist ihre Verwendung einerseits von der gewünschten Interaktionsgenauigkeit und andererseits von subjektiven Nutzereigenschaften abhängig. Eine schnelle und intuitive Manipulation der 2D Objekte ermöglicht die symmetrische Interaktionstechnik und in Kombination mit dem manuellen Umschalten der Präzisionsstufen vor allem für unerfahrene Nutzer zur Interaktion in GHD-Umgebungen geeignet. Eine feinmotorische und präzisere Manipulation wird durch die asymmetrische Interaktionstechnik unterstützt, die von Nutzern eine gewisse Erfahrung mit asynchronen Bewegungsabläufen erfordert. In Kombination mit der automatischen Anpassung der Präzisionsstufen konnten einige Nutzer aktiv von der intelligenten Umgebung bei ihrer gegenwärtigen Tätigkeit, d.h. der Manipulation von 2D Objekten unterstützt werden.

Mit Hilfe der Studien konnten die Einflussfaktoren der subjektiven und objektiven Eigenschaften der Benutzer auf Faktoren der physischen Umgebung und des Mappings untersucht werden, die bisher kaum untersucht wurden (siehe Tabelle 3.1.1 und Tabelle 7.0.1). Von besonderem Interesse war dabei, wie sich die subjektiven Nutzereigenschaften auf die physische Navigation und damit auf die Nutzung des Interaktionsraumes für die distanzabhängige Interaktion auswirken.

gegebener Faktor		Physische Umgebung			Virtuelle Umgebung			Benutzer			Mapping			
		Einfluss auf Interaktionsraum	Eingabemodalität	Displayanordnung	Informationsdarstellung	Interaktionstechnik	Dimension d. Darstellung, Detailgrad	physische Navigation	Wahrnehmung	Nutzeranzahl	Isomorphie	Projektion	Kinematik	Cursorberechnung
Benutzer	Objektive Eigenschaften		x		x	x								
	– Physische Navigation		x			x							x	x
	– Wahrnehmung	x			x			x						
	– Koordination				x	x				x			x	
	Subjektive Eigenschaften	x	x			x		x			x		x	
	Ergonomie		x			x		x			x			x

Tabelle 7.0.1.: Ergänzung der Einflussfaktoren aus Tabelle 3.1.1 basierend auf den durchgeführten Studien (grau markiert).

Wie muss distanzabhängige Interaktion für Fullscreen-Anwendungen in großen hochauflösenden Displayumgebungen gestaltet sein, damit die physische Navigation und die Präferenzen des Benutzers berücksichtigt werden?

Die physische Navigation wurde für die Anpassung der Präzision der Interaktion und der Informationsdarstellung genutzt und in die Interaktionstechnik integriert. Aufgrund unserer Ergebnisse empfehlen wir folgende Richtlinien für die distanzabhängige Interaktion in GHD-Umgebungen.

Für die Definition von Interaction Scaling müssen neben den Parametern des physischen Interaktionsraumes (Eingabegerät, Ausdehnung Interaktionsraum, etc.) und den objektiven Eigenschaften des Nutzers (Größe, physiologische Einschränkungen) auch die individuellen subjektiven Eigenschaften für eine effiziente und effektive Nutzerperformanz berücksichtigt werden.

Grundsätzlich ist die distanzabhängige Berechnung des Mappingfaktors angebracht, jedoch sollte die Berechnung nicht über alle Interaktionsdistanzen gleich erfolgen. Einerseits müssen die bevorzugte Interaktionsdistanz des Nutzers und andererseits die erforderliche Mobilität aufgrund der Interaktionsaufgaben berücksichtig-

sichtigt werden. Die individuellen Nutzerpräferenzen erfordern beispielsweise eine andere Anpassung der Präzision im Nahbereich als im Fernbereich. Im Nahbereich muss die Möglichkeit der groben Präzision geben sein, z.B. durch manuelles Umschalten der Präzisionsstufen. Wenn der Nutzer bevorzugt entfernt arbeitet, dann sollte der Nutzer mit wenigen Schritten zur Displaywand eine hohe Präzisionsstufe erreichen, z.B. durch Verwendung einer exponentiellen Mappingfunktion.

Die Idee des Interaction Scaling mit taskbasierten automatischen Wechsel zwischen grober und präziser Interaktionsgenauigkeit ist in GHD-Umgebungen nur bedingt einsetzbar. In intelligenten Umgebungen sollte ein semi-automatischer distanzabhängiger Ansatz verwendet werden, so dass der Nutzer die Kontrolle über die Interaktionsgenauigkeit hat (manueller Wechsel), aber diese auf Wunsch des Nutzers durch das System bzw. die intelligente Umgebung übernommen wird (automatischer Wechsel). Je höher die erforderliche Präzision der Interaktion ist, umso wichtiger ist die Kontrolle für den Nutzer, damit er verlässlich arbeiten kann (d.h. geringe Frustration, erwartungsgemäßes Systemverhalten). Bei den vorgestellten Szenarien zur Navigation mit IS war die Präzision der Interaktion geringer und die Anpassung der Informationsdarstellung war aus Nutzersicht nachvollziehbar. Während in den Studien zu den beidhändigen Manipulationstechniken einige Nutzer Schwierigkeiten mit IS beim Interagieren hatten.

Wenn die gegenwärtige Tätigkeit in der GHD-Umgebung ein häufiges vor und zurück laufen vom Nutzer erfordert, dann ist ein automatisches Umschalten der distanzabhängigen Präzision dem manuellen Umschalten vorzuziehen. In Studie 3 konnte gezeigt werden, dass die Nutzerperformanz beim automatischen Wechsel deutlich besser war als beim manuellen Wechsel.

Die vorliegende Arbeit knüpft an die Dissertation von König [Kön10] an und fokussiert auf den Benutzer in der GHD-Umgebung. Daher wurden ausgewählte Parameter aus seinem Design-space für die Interaktionstechniken übernommen und um die physiologischen Aspekte für den Benutzer ergänzt. Da unser Parameterraum mit unbekanntem Interaktionstechniken arbeitet, wurden andere Parameter für das Mapping betrachtet als dies bei König der Fall ist (z.B. Isomorphie, Projektion).

Ein wesentlicher Aspekt des Parameterraumes ist die Adaption auf unterschiedliche Displayumgebungen (z.B. homogene und heterogene große Displays). Dies wurde mit dem Ansatz des Smart Interaction Management [RLSS12] im Smart Meeting Room (SMR) gezeigt. Der Ansatz kombiniert die Aspekte der Informationsbereitstellung von und deren Anpassung auf unterschiedliche Geräten (vgl. [Rad14]) mit den Nutzereingaben von verschiedenen Eingabegeräten und variierenden Eingabedistanzen (IS). Bei der Informationsdarstellung im SMR sollten die objektiven und subjektiven Eigenschaften der Nutzer berücksichtigt werden (vgl. [Thi10]). Dies gilt ebenfalls für die Interaktion im SMR und wird durch den vorgestellten Parameterraum in dieser Arbeit abgedeckt.

Bei hochauflösenden kleinen Displays (z.B. 42-Zoll Monitor mit 3840x2160 px) würde der Nutzer eher eine feste Interaktionsdistanz verwenden (keine Mobilität). In diesem Fall würde für IS der motorische Aspekt der physischen Navigation genutzt werden, d.h. die Anpassung des distanzabhängigen Mappingfaktors erfolgt über die Armstreckung, ähnlich wie Poupyrev et al. [PBWI96].

Beim Einsatz von großen hochauflösenden Displays als erweiterte Desktopumgebung werden üblicherweise Büro- und Analyseaufgaben von einer sitzenden konstanten Position aus durchgeführt [CHBL05, EBZ⁺12]. In diesem Anwendungsgebiet eignet sich eher eine geräteabhängige und weniger eine distanzabhängige Mappingfunktion. In diesem Fall würde eine globale Berechnung des Mappingfaktors für die verwendete Displaykonfiguration erfolgen, die durch eine zusätzliche lokale Anpassung aufgrund des verwendeten Eingabegerätes oder der Nutzerpräferenz ergänzt wird, angelehnt an Smart Interaction Management.

Unsere Studien wurden in einer GHD-Umgebung durchgeführt, die keinen Mappingfaktor größer eins erfordert. Darüber hinaus war es für Erwachsene möglich mit Hilfe der beidhändigen Interaktionstechniken jedes Objekt auf der Displaywand zu selektieren. Wird die Displayfläche jedoch wesentlich größer, führt dies aufgrund der distanzabhängige Mappingmethode mit orthogonaler Projektion zu einem deutlich höheren physiologischen Aufwand. In diesem Fall eignet sich eine Mappingmethode mit perspektivischer Projektion, die jedoch den Tremor sichtbar macht, was vor allem bei der symmetrischen Manipulationstechnik störend sein kann. Hier wäre eine Kombination der Projektionen in der Mappingmethode denkbar, indem beispielsweise die perspektivische Projektion für die Selektion und die orthogonale Projektion für die anderen Manipulationstasks verwendet wird.

Die verwendete Displaywand bietet die Möglichkeit, Mini-Kameras zwischen den Bezels zu integrieren. Diese Kameras können für die Bestimmung des Nutzerabstandes vor der Displaywand verwendet werden, wenn beispielsweise keine Blickrichtung oder schnell wechselnde exakte Abstandsposition benötigt werden, da die Abstandsbestimmung mittels Kamera-Matrix im Cluster zusätzlichen Berechnungsaufwand benötigt (abhängig von der verwendeten Kameraauflösung). Im Szenario 3D Terrainvisualisierung kann die Bestimmung der aktiven Displays abhängig von der Nutzerposition und -orientierung somit als geräteloses Head-Tracking erfolgen. Da hier die Blickrichtung des Betrachters zum Displaymittelpunkt berechnet wird, ist eine Genauigkeit im cm-Bereich ausreichend.

FAZIT: Als Beitrag zur Interaktion in GHD-Umgebungen wurde in dieser Arbeit die bisher weniger untersuchten Aspekte in diesem Forschungsbereich behandelt, indem (1) bekannte Navigations- und Manipulationstechniken umgesetzt wurden, die *distance-aware* und *manual-switching* Mappingmethoden unterstützen (siehe Abbildung 2.4.3 *Interaction Scaling*), und (2) die damit verbundenen Präferenzen der Nutzer bei der Interaktion untersucht wurden (siehe Tabelle 7.0.1).

Allerdings besteht noch ein erheblicher Bedarf an weiteren Forschungsarbeiten bzw. wurden neue Fragestellungen mit dieser Arbeit aufgeworfen. Offene Aspekte werden im folgenden Abschnitt zusammengefasst.

Ausblick

Unsere Studien haben gezeigt, dass es Probanden gibt, die eine feste Position bei der Interaktion in GHD-Umgebungen bevorzugen, als auch Probanden, die grundsätzlich die physische Navigation verwenden. Einerseits stimmen wir den Untersuchungsergebnissen von Kopper et al. [KSMB08] zu, die besagen, dass die meisten Probanden eine feste Position zum Interagieren bevorzugten und nur nahe an das Display herangingen, wenn es absolut nötig war. Andererseits deckt es sich teilweise mit den Studienergebnissen von Ball et al. [BNB07], dass Nutzer die physische Navigation der virtuellen Navigation in GHD-Umgebungen vorziehen.

Jedoch wurde in unseren Studien und Beispielszenarien die virtuelle Navigation nicht unterstützt. Daher sollte in weiterführenden Arbeiten untersucht werden, ob die physische Navigation wirklich effizienter als die virtuelle Navigation ist [Kön10], z.B. in Kombination mit komplexeren Manipulationsaufgaben.

Um den Nutzer in intelligenten Umgebungen zu unterstützen, z.B. durch Automatismen oder Intentionserkennung, sind Profile der Nutzertypen erforderlich. Dazu müssen folgende Fragen beantwortet werden: Kann aufgrund der gemessenen Daten der Nutzertyp und dessen Nutzerpräferenzen vorhergesagt werden? Lassen sich die beobachteten Eigenschaften zu allgemeinen Profilen clustern?

Im Rahmen unserer Studien konnten wir diese Fragen nicht beantworten, da die gewählten Verhaltenskategorien zu ungenau oder teilweise unbekannt waren. Hier empfiehlt sich, aufbauend auf unseren ersten Verhaltensbeobachtungen, in Folgestudien die Kategorien detaillierter zu wählen und das Nutzerverhalten genauer zu evaluieren (z.B. bevorzugte Interaktionsdistanz).

In der vorliegenden Arbeit wurde primär die Manipulation mit einfachen 2D Objekten betrachtet. Es wurde nicht untersucht, welche zusätzlichen Einflussfaktoren bei der Interaktion mit 3D Objekten bzw. im virtuellen 3D Raum eventuell zu berücksichtigen sind. Denkbare wäre die Displaydistanz für die Bestimmung der Tiefe des Selektionsstrahls zu verwenden, z.B. in Kombination mit dem Zonen Mapping. Ein anderer Ansatz wäre beispielsweise die Kombination des Interaction Scaling mit der *improved virtual pointer* Metapher von Steinicke et al. [SRH05a] für die 3D Objektmanipulation, indem ein zusätzlicher distanzabhängiger Mappingfaktor für die präzise Positionierung von entfernten Objekten genutzt wird, ohne dass der Nutzer die Objekte näher heranholen muss.

Große Stereodisplays bieten die Möglichkeit, virtuelle 3D Objekte sehr realistisch darzustellen. Für diese Displaytypen müssen weitere Einflussfaktoren für die Interaktion berücksichtigt werden, z.B. Objektdarstellung (Positive/Null/Negative Par-

allaxe), Störungen beim Fokussieren zwischen Eingabegerät und virtuellem Objekt, Distanz zwischen 3D Objekt und Display bei berührungsempfindlichen Displays. Typische 3D Interaktionstechniken mit Stereodisplays sind beispielsweise Gesten [KAHF05, BSS13] oder getrackte Eingabegeräte [DB10]. Hier muss in weiteren Forschungsarbeiten untersucht werden, ob eine distanzabhängige Interaktion mit Stereodisplays möglich ist.

Der Aspekt von mehreren Benutzern bei der Interaktion in GHD-Umgebungen wurde in dieser Arbeit mit zwei Benutzern betrachtet. Hier ist in weiteren Arbeiten zu klären, wie beispielsweise die beidhändig distanzabhängigen Interaktionstechniken mit mehreren Benutzern im physischen Interaktionsraum funktioniert. Dabei müssen sowohl technische Herausforderungen (z.B. verlässliche eindeutige Markerkennung) als auch soziale Herausforderungen (z.B. effektive Zusammenarbeit, soziale Interaktionsabstände) bewältigt werden.

A. Anhang

A.1. Eingabemodalitäten und Nutzerinteraktion

Nutzergruppe	Körpergröße	Handgröße ¹	max. Höhe auf GHD ²	Arbeitshöhe für GHD
Erwachsener (männl.)	184cm	20cm	235cm	235 – 20 = 215 [cm]
Erwachsener (weibl.)	165cm	17cm	211cm	211 – 17 = 194 [cm]
Kind (männl.)	133cm	14cm	167cm	167 – 14 = 153 [cm]

¹ Länge der Handfläche vom Mittelfinger zur Handwurzel

² erreichbare Höhe mit ausgestrecktem Arm nach oben

Tabelle A.1.1.: Beispielberechnung der Testpersonen für unterschiedliche Körpergrößen in Abschnitt 3.1.3.



Abbildung A.1.1.: Tastenbelegung für die Interaktionsmodi in den Experimenten (siehe Abschnitt 5.2.2).

A.2. Fragebögen

In den Publikationen [LS12, LS13] und den Experimenten wurden folgende Synonyme für die beschriebenen 2D Manipulationstechniken verwendet:

- (1) asymmetrische Interaktion: ribbon, separated-cursors (2 Cursor), it1;
- (2) symmetrische Interaktion: wheel, connected-cursors (3 Cursor), it2;
- (3) einhändige absolute Nutzereingabe: it3.

A.2.1. Fragebogen der Studie 1

Für die Studie 1 wurde von jedem Probanden der Fragebogen am Ende des Experimentes ausgefüllt (vgl. Abschnitt 6.1.1.1). Im Experiment wurden die asymmetrische Interaktionstechnik (Ribbon-Mode, it1) und die symmetrische Interaktionstechnik (Wheel-Mode, it2) mit dem absolutem Mappingfaktor (1:1), dem linearen Mappingfaktor (linear) und dem hybriden Mappingfaktor (Zonen) getestet. Die Mappingmethode wurde in Studie 1 "Zonen" genannt, da sie in Nah-, Fern- und Zwischenbereich unterscheidet; sie ist nicht zu verwechseln mit der zonenartigen Mappingfunktion in Studie 2. Die Tabelle A.2.2 zeigt die Auswertung der Fragebögen der sechs Probanden im Experiment. Ein konkretes 3D Modellierungswerkzeug haben Proband 2 (3ds max) und Proband 4 (Blender) genannt. Proband 3 präferierte keine Interaktionstechnik bezüglich der Handhabung.

Proband	1	2	3	4	5	6
5-Punkt Likert-Skala: 1 = trifft im geringen Masse zu; 3 = neutral; 5 = trifft in hohem Masse zu						
Ich habe Erfahrung mit der Interaktion in großen Displayumgebungen.	4	2	1	4	3	1
Ich habe Erfahrung mit 3D Modellierungswerkzeugen (z.B. Blender, Maya,...).	3	4	1	3	1	1
Ich konnte die Interaktionstechniken recht schnell bedienen.	5	4	5	4	4	4
Ribbon-Mode: Ich habe das Interaction Scaling von realen zu virtuellen Bewegungen bewusst eingesetzt.	4	3	3	4	3	4
Wheel-Mode: Ich habe das Interaction Scaling von realen zu virtuellen Bewegungen bewusst eingesetzt.	4	3	3	4	3	4
Ribbon-Mode: Ich fand das Interaction Scaling störend.	3	2	1	3	3	3
Wheel-Mode: Ich fand das Interaction Scaling störend.	3	2	1	3	3	3
Ranking: 1 = am Besten gefallen; 3 = am wenigsten gefallen						
Mir hat die Interaktionstechnik von der Handhabung besser gefallen:	it2	it1	-	it2	it1	it2
Ribbon-Mode: Ranking des Mappings von realen auf virtuelle Bewegungen bei:						
1:1	1	2	2	2	3	1
linear	2	1	1	1	1	2
Zonen (hybrid)	3	3	3	3	2	3
Wheel-Mode: Ranking des Mappings von realen auf virtuelle Bewegungen bei:						
1:1	1	2	2	2	3	1
linear	2	1	1	1	1	2
Zonen (hybrid)	3	3	3	3	2	3
Welche Kombination bevorzugst du:	d	b	b, e	e	b	d

Tabelle A.2.2.: Auswertung der Fragebögen aus Studie 1.

Fragebogen

Testperson (Nr.): Alter: Geschlecht (m/w): Lehrstuhl/Fachgebiet:

Rechtshänder/ Linkshänder

Farbenblindheit/-beeinträchtigung:

Ich habe Erfahrung mit der Interaktion in großen Displayumgebungen.	trifft im geringen Masse zu [1]	[2]	[3]	[4]	trifft in hohem Masse zu [5]
Ich habe Erfahrung mit 3D Modellierungswerkzeugen (z.B. Blender, Maya,...).	trifft im geringen Masse zu [1]	[2]	[3]	[4]	trifft in hohem Masse zu [5]
Erfahrung mit welchem 3D Modellierungswerkzeug:					
Ich konnte die Interaktionstechniken recht schnell bedienen.	trifft im geringen Masse zu [1]	[2]	[3]	[4]	trifft in hohem Masse zu [5]
Ribbon-Mode: Ich habe das Interaction Scaling von realen zu virtuellen Bewegungen bewusst eingesetzt.	trifft im geringen Masse zu [1]	[2]	[3]	[4]	trifft in hohem Masse zu [5]
Wheel-Mode: Ich habe das Interaction Scaling von realen zu virtuellen Bewegungen bewusst eingesetzt.	trifft im geringen Masse zu [1]	[2]	[3]	[4]	trifft in hohem Masse zu [5]
Ribbon-Mode: Ich fand das Interaction Scaling störend.	trifft im geringen Masse zu [1]	[2]	[3]	[4]	trifft in hohem Masse zu [5]
Wheel-Mode: Ich fand das Interaction Scaling störend.	trifft im geringen Masse zu [1]	[2]	[3]	[4]	trifft in hohem Masse zu [5]

Mir hat die Interaktionstechnik von der Handhabung besser gefallen:

- Ribbon
 Wheel

Ranking des Mappings von realen (Hand) auf virtuelle (Cursor) Bewegungen bei Ribbon-Mode: von 1 bis 3 bewerten (1 = am besten gefallen)

- 1:1
 lineare
 Zonen

Ranking des Mappings von realen (Hand) auf virtuelle (Cursor) Bewegungen bei Wheel-Mode: von 1 bis 3 bewerten (1 = am besten gefallen)

- 1:1
 lineare
 Zonen

A.2.2. Fragebogen der Studie 2

Folgender Fragebogen wurde in Studie 2 verwendet (vgl. Abschnitt 6.1.1.2). Da den Probanden während der Studie nur mitgeteilt wurde, ob ein distanzabhängiger Mappingfaktor aktiv ist oder nicht, wurde in dem Fragebogen nur nach den Bedingungen mit Interaction Scaling (lineare/Zonen Mappingfunktion) und ohne Interaction Scaling (statische Mappingfunktion) gefragt. Die Fragen zur Zufriedenheit sind aus dem PSSUQ [Lew93] adaptiert (siehe Anhang A.2.3). In Tabelle A.2.3 ist die Auswertung der 14 Fragebögen zusammengefasst.

Average age:	27 years (range: 20–39 years)
Average size:	176 cm (range: 160–191cm)
Right hander:	14 (100%)
Gender:	10 male (71%), 4 female (29%)
Profession:	7 students (50%), 7 colleagues (50%)
VR experience:	6x never, 5x seldom, 2x occasionally, 1x frequently

Question Subject	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Overall, I am satisfied with how easy it is to use the puzzle application.	4	5	4	2	4	4	5	-	4	3	4	5	2	5
It was easy to learn to use the puzzle application with interaction scaling.	5	4	4	1	5	2	5	5	4	4	5	5	4	4
I was able to complete the tasks and scenarios quickly using interaction scaling.	4	4	3	1	4	2	5	4	4	4	4	4	3	4
Overall, I am satisfied with the ease of completing the tasks in this scenario without interaction scaling.	4	4	4	5	4	5	3	3	5	3	3	3	3	5
Overall, I am satisfied with the amount of time it took to complete the tasks in this scenario without interaction scaling.	5	4	3	5	5	4	4	3	5	3	4	3	2	4
Overall, I am satisfied with the ease of completing the tasks in this scenario with interaction scaling.	5	4	3	2	4	2	5	4	4	3	4	5	3	4
Overall, I am satisfied with the amount of time it took to complete the tasks in this scenario with interaction scaling.	5	4	4	2	5	3	4	4	5	3	3	5	2	4
Preferences:														
Yes, I prefer the scenario with interaction scaling.	x		x		x		x	x		x	x	x		x
No, I prefer the scenario without interaction scaling.				x		x			x					
I have no preference.		x											x	

Tabelle A.2.3.: Auswertung der Fragebögen aus Studie 2.

Experiment Puzzle Solver – Questionnaire

subject ID:		gender:	male <input type="checkbox"/>	female <input type="checkbox"/>
age:		right hander:	yes <input type="checkbox"/>	no <input type="checkbox"/>
body size/height:		<input type="checkbox"/> Never <input type="checkbox"/> Seldom <input type="checkbox"/> Occasionally <input type="checkbox"/> Frequently		
Do you ever work with VR environments (e.g., Cave, display wall, head-mounted display)?				

Instructions: Please indicate your level of agreement or disagreement with the statements below.
 1 = Strongly Disagree, 3 = No Opinion, 5 = Strongly Agree

	strongly disagree				strongly agree
Overall, I am satisfied with how easy it is to use the puzzle application.	①	②	③	④	⑤
It was easy to learn to use the puzzle application with interaction scaling.	①	②	③	④	⑤
I was able to complete the tasks and scenarios quickly using interaction scaling.	①	②	③	④	⑤
Overall, I am satisfied with the ease of completing the tasks in this scenario without interaction scaling.	①	②	③	④	⑤
Overall, I am satisfied with the amount of time it took to complete the tasks in this scenario without interaction scaling.	①	②	③	④	⑤
Overall, I am satisfied with the ease of completing the tasks in this scenario with interaction scaling.	①	②	③	④	⑤
Overall, I am satisfied with the amount of time it took to complete the tasks in this scenario with interaction scaling.	①	②	③	④	⑤

1. What did you *like* the most about using the puzzle application with interaction scaling?

2. What did you *dislike* the most about using the puzzle application with interaction scaling?

3. Do you prefer the scenario with interaction scaling?

- Yes, I prefer the scenario with interaction scaling.
- No, I prefer the scenario without interaction scaling.
- I have no preference.

A.2.3. Auszug PSSUQ, ASQ nach J.R. Lewis, 1995

Im Folgenden sind die Fragen aus dem originalen PSSUQ und ASQ aufgelistet, die als Grundlage für den Post-Test Fragebogen in der Studie 3 verwendet wurden [Lew95][Lew93].

PSSUQ (Frage 1-8 und 19) und ASQ (Frage 2):

1. Overall, I am satisfied with how easy it is to use this system.
2. It was simple to use this system.
3. I could effectively complete the tasks and scenarios using this system.
4. I was able to complete the tasks and scenarios quickly using this system.
5. I was able to efficiently complete the tasks and scenarios using this system.
6. I felt comfortable using this system.
7. It was easy to learn to use this system.
8. I believe I could become productive quickly using this system.
19. Overall, I am satisfied with this system.
2. Overall, I am satisfied with the amount of time it took to complete this task.

Die Fragen werden mit einer 7-Punkt Likert-Skala bewertet:

STRONGLY AGREE 1 2 3 4 5 6 7 STRONGLY DISAGREE

COMMENTS:

A.2.4. Fragebögen der Studie 3

In der Studie 3 wurde von jedem Probanden vor dem Experiment der "Pre-Test Fragebogen" ausgefüllt, nach jedem Szenario der "NASA Task Load Index Questionnaire" nach [Har06] und abschließend der "Post-Test Fragebogen" im Experiment ausgefüllt (vgl. Abschnitt 6.2).

Die Testpersonen wurden gebeten nach dem Experiment einen Fragebogen zur Zufriedenheit und Benutzbarkeit der verwendeten Arten des Cursorwechsels (automatisch, manuell) auszufüllen. Dieser Posttest-Fragebogen wurde von jedem Probanden für jede Interaktionstechnik getrennt ausgefüllt.

Für die Bewertung der Nutzerzufriedenheit wurden Fragen aus dem PSSUQ von Lewis [Lew95] verwendet (vgl. Abschnitt 2.2). Da wir Interaktionstechniken getestet haben, waren von den 19 Fragen nur die Fragen zu *system usefulness* (Fragen 1-8 und Frage 19) anwendbar, d.h. Fragen zu *completion of work* und *ease of learning* (siehe Anhang A.2.3). Lewis [Lew95] empfiehlt, die Fragen zu erweitern, wenn die Evaluation es benötigt. Da der Zeitfaktor als subjektive Zufriedenheitsbewertung bei 3D Interaktionstechniken eine Rolle spielt, wurden die Frage 2 aus dem ASQ von Lewis [Lew95] verwendet und auf die eigene Studie angepasst. Der ASQ alleine konnte nicht für die Bewertung der Nutzerzufriedenheit verwendet werden, da drei Fragen zu wenige Bewertungskriterien für die evaluierten Interaktionstechniken sind. Einerseits ist die Frage 1 im ASQ äquivalent zur Frage 1 im PSSUQ

und andererseits konnte die Frage 3 aus dem ASQ nicht verwendet werden, da im eigenen Experiment keine Dokumentation zur Verfügung stand und auch nicht Bestandteil des Experimentes war.

Es wurde eine 7-Punkte Skala verwendet mit den Endpunkten *strongly agree* (Punktwertung 1) und *strongly disagree* (Punktwertung 7). Wenn eine Frage nicht beantwortet wurde, wird diese mit 0 bewertet und die Frage geht nicht in die Gesamtbewertung der Nutzerzufriedenheit für die Testperson ein. Bei einer Testperson kam es vor, dass zwei Fragen nicht beantwortet wurden. Die allgemeine Nutzerzufriedenheit zu einer Interaktionstechnik wird als durchschnittliche Punktwertung aller Fragen berechnet.

Die Auswertung der NASA TLX Fragebögen erfolgte mit der zugehörigen Software¹, dabei werden die Werte der einzelnen Skalen normalisiert (Wert zwischen 0 bis 100). Es wurde keine Gewichtung der Skalen vorgenommen, sondern ein *Raw TLX* [Har06] verwendet.

¹<http://www.playgraph.com/2009/bits/nasa-tlx>

Pre-Test Fragebogen

Subject #ID: _____

Alter: _____

Geschlecht: männlich weiblich

Körpergröße (in m): _____

Rechtshänder /Linkshänder: R LBeeinträchtigung im Sehvermögen: _____
(Sehhilfe, Farbblindheit/beeinträchtigung)

Art der Beeinträchtigung im Sehvermögen:

- Kurzsichtigkeit Dioptrien: ca. _____
- Weitsichtigkeit Dioptrien: ca. _____
- Hornhautverkrümmung
- Sonstiges: _____

Momentane Tätigkeit: _____
(bei Student den Studiengang bzw. bei Angestellter den Lehrstuhl angeben)

Haben Sie jemals mit VR-Umgebungen gearbeitet (z.B. Head-mounted Displays, Cave, Displaywand, Stereodisplay)?

- Niemals Selten Gelegentlich Häufig

In welchen der folgenden Displayumgebungen haben Sie schon gearbeitet?
(mehrere Antworten sind möglich)

- Multi-Displays(2-4 Monitore) LCD-Wand (mehr als 4 LCDs)
- Projektor (Beamer) Rückprojektionswand
- Smart Room (mehrere Projektoren) Multitouch Display (kein SmartPhone)

Welche der folgenden Eingabegeräte haben Sie schon benutzt?
(mehrere Antworten sind möglich)

- Laserpointer Spielkonsole Nintendo Wii
- Gyro-Maus Spielkonsole Kinect
- Nintendo Wii Remote (nicht als Spielkonsole)

NASA Task Load Index (TLX) Questionnaire

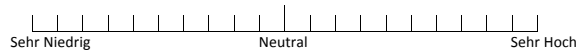
Subject #ID: _____ Interaktionstechnik: A B Test: 1 2 3

Mit diesem Fragebogen möchte ich herausfinden, wie anstrengend oder belastend Sie das Bearbeiten der Aufgabe (Lösen des Puzzle) empfunden haben.
Im ersten Teil des Fragebogens sollen Sie ankreuzen, wie anstrengend Sie den Task in mehrerer Hinsicht empfunden haben. Kreuzen Sie bitte die für Sie zutreffende Anstrengungsstärke an.

Mentale Anforderung (geistige Beanspruchung)

Unter geistiger Beanspruchung versteht man das Ausmaß, in dem Denk- und Verstehensbemühungen notwendig waren, um die Aufgabe zu lösen (z.B. Denken, Entscheiden, Erinnern, etc.)

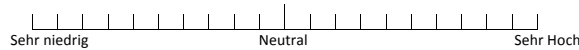
Wie geistig anspruchsvoll war die Aufgabe?



Physische Anforderung (Körperliche Beanspruchung)

Mit körperlicher Anforderung ist gemeint, wie viel körperliche Aktivität nötig war, um die Aufgabe zu lösen.

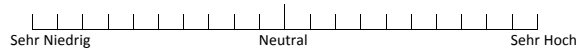
Wie körperlich anstrengend war die Aufgabe?



Zeitliche Anforderung (Zeitdruck)

Mit Zeitdruck ist die Geschwindigkeit gemeint, in der die Aufgabe ausgeführt werden musste (langsam oder schnell, ausreichend Zeit zum Fertigwerden oder zu wenig Zeit).

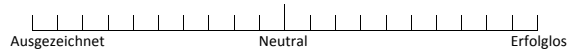
Wie eilig oder hastig war das Tempo der Aufgabe?



Durchführung (Performanz, Leistungsdruck)

Die Performanz (Leistungsdruck) bezieht sich darauf, inwieweit Sie denken, die Ziele der Aufgabe erreicht zu haben, wie zufrieden Sie mit Ihrer Leistung waren.

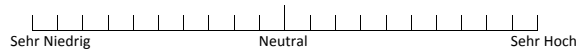
Wie erfolgreich waren Sie beim Erreichen der Aufgabe?



Arbeitsaufwand (Mühe)

Der Arbeitsaufwand (Mühe) bezieht sich darauf, wie hart man arbeiten muss, um die Ziele der Aufgabe zu erreichen.

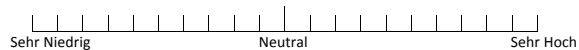
Wie schwer mussten Sie arbeiten, um Ihr Anforderungsniveau zu schaffen?



Frustration

Die Frustration bezieht sich darauf, wie unsicher, entmutigt, verwirrt, gestresst oder genervt man sich während der Erfüllung der Aufgabe gefühlt hat.

Wie unsicher, entmutigt, irritiert, angespannt und verärgert waren Sie?



Post-Test Fragebogen

Subject #ID: _____

Interaktionstechnik: A B

Instructions: Please indicate your level of agreement or disagreement with the statements below.

	strongly agree							strongly disagree
1. Overall, I am satisfied with how easy it is to use the automatic cursor switching.	1	2	3	4	5	6	7	
2. It is simple to use the automatic cursor switching.	1	2	3	4	5	6	7	
3. I could effectively complete the tasks and scenarios using automatic cursor switching.	1	2	3	4	5	6	7	
4. I was able to complete the tasks and scenarios quickly using automatic cursor switching.	1	2	3	4	5	6	7	
5. I was able to efficiently complete the tasks and scenarios using automatic cursor switching.	1	2	3	4	5	6	7	
6. I felt comfortable using automatic cursor switching.	1	2	3	4	5	6	7	
7. It was easy to learn to use automatic cursor switching.	1	2	3	4	5	6	7	
8. I believe I could become productive quickly using automatic cursor switching.	1	2	3	4	5	6	7	
9. Overall, I am satisfied with the amount of time it took to complete the tasks using automatic cursor switching.	1	2	3	4	5	6	7	
10. Overall, I am satisfied with the automatic cursor switching.	1	2	3	4	5	6	7	

Comments:

	strongly agree							strongly disagree
1. Overall, I am satisfied with how easy it is to use the manual cursor switching.	1	2	3	4	5	6	7	
2. It is simple to use the manual cursor switching.	1	2	3	4	5	6	7	
3. I could effectively complete the tasks and scenarios using manual cursor switching.	1	2	3	4	5	6	7	
4. I was able to complete the tasks and scenarios quickly using manual cursor switching.	1	2	3	4	5	6	7	
5. I was able to efficiently complete the tasks and scenarios using manual cursor switching.	1	2	3	4	5	6	7	
6. I felt comfortable using manual cursor switching.	1	2	3	4	5	6	7	
7. It was easy to learn to use manual cursor switching.	1	2	3	4	5	6	7	
8. I believe I could become productive quickly using manual cursor switching.	1	2	3	4	5	6	7	
9. Overall, I am satisfied with the amount of time it took to complete the tasks using manual cursor switching.	1	2	3	4	5	6	7	
10. Overall, I am satisfied with the manual cursor switching.	1	2	3	4	5	6	7	

A.3. Ergebnisse der Evaluierungen

Für die Signifikanzprüfung der Varianzanalysen wurden folgende kritische Werte benötigt (aus [RFHN06]).

$\alpha = 5\%$	$\alpha = 1\%$	$\alpha = 5\%$	$\alpha = 1\%$	$\alpha = 5\%$	$\alpha = 1\%$
F(1,21)= 4,32	F(1,21)= 8,02	q(2,84)= 2,79	q(2,84)= 3,74	$\chi^2(2)= 5,99$	$\chi^2(2)= 9,21$
F(1,84)= 3,97	F(1,84)= 6,99	q(4,20)= 3,96	q(4,20)= 5,02	$\chi^2(4)= 9,49$	$\chi^2(4)= 13,28$
F(2,19)= 3,52	F(2,19)= 5,93	q(4,21)= 3,94	q(4,21)= 4,99		
F(2,22)= 3,44	F(2,22)= 5,72	q(5,20)= 4,23	q(5,20)= 5,29		
F(2,25)= 3,39	F(2,25)= 5,57	q(5,21)= 4,21	q(5,21)= 5,26		
F(2,26)= 3,38	F(2,26)= 5,53	q(5,84)= 3,94	q(5,84)= 4,78		
F(2,45)= 3,21	F(2,45)= 5,13				
F(2,51)= 3,19	F(2,51)= 5,07				
F(2,76)= 3,13	F(2,76)= 4,93				
F(4,84)= 2,50	F(4,84)= 3,58				

A.3.1. Auswertung mit SigmaPlot

Die statistische Auswertung der Studien erfolgte mit der Software SigmaPlot. Für die Signifikanzprüfung ($\alpha = 5\%$) wurden der t-Test für abhängige Stichproben (*paired t-test*), die einfaktorielle Varianzanalyse mit Messwiederholung (*one-way RM ANOVA*) und die zweifaktorielle Varianzanalyse mit Messwiederholung (*two-way RM ANOVA*) verwendet. Wurde eine Voraussetzung des parametrischen Verfahrens verletzt, dann wurde automatisch das entsprechende nicht parametrische Verfahren für die Signifikanzprüfung durchgeführt, d.h. der Wilcoxon-Test (*Wilcoxon Signed Rank Test*) für zwei abhängige Stichproben und die Friedman Varianzanalyse an Rängen mit wiederholten Messungen (*Friedman RM ANOVA on Ranks*). SigmaPlot unterstützt keine Friedman zweifaktorielle Varianzanalyse an Rängen und bietet kein Korrekturverfahren bei Verletzung der Sphärizität. Für die Überprüfung der Normalität bzw. gleicher Varianz wird eine Irrtumswahrscheinlichkeit von 5% festgelegt. Die fehlenden Datensätze für eine Bedingung werden in den Berichten als *Missing* gekennzeichnet.

Für die Testberichte verwendet SigmaPlot einige Einschränkungen:

Note: The multiple comparisons on ranks do not include an adjustment for ties.

A result of "Do Not Test" occurs for a comparison when no significant difference is found between two means that enclose that comparison. For example, if you had four means sorted in order, and found no difference between means 4 vs. 2, then you would not test 4 vs. 3 and 3 vs. 2, but still test 4 vs. 1 and 3 vs. 1 (4 vs. 3 and 3 vs. 2 are enclosed by 4 vs. 2: 4 3 2 1). Note that not testing the enclosed means is a procedural rule, and a result of Do Not Test should be treated as if there is no significant difference between the means, even though one may appear to exist.

A.3.2. Statistische Auswertungen der Studie 1

Nachfolgend sind die SigmaPlot Berichte der statistischen Auswertung beigefügt: (1) die deskriptive Statistik für die abhängigen Variablen Anzahl der Cursor Resets (*count of cursor resets*) für it1 und it2; für die abhängige Variable Bearbeitungszeit (*completion time*) (2) die einfaktorielle – und (3) die zweifaktorielle Varianzanalyse mit Messwiederholung.

(1) Descriptive Statistics

Data source: *count of cursor resets (it1) in study-1*

Column	Size	Missing	Mean	Std Dev	Std. Error	C.I. of Mean
count static	12	0	1,083	1,084	0,313	0,689
count linear	12	0	1,000	0,953	0,275	0,606
count hybrid-linear	12	0	3,833	3,298	0,952	2,096
Column	Range	Max	Min	Median	25%	75%
count static	3,000	3,000	0,000	1,000	0,000	1,500
count linear	3,000	3,000	0,000	1,000	0,000	1,500
count hybrid-linear	10,000	10,000	0,000	3,500	1,000	6,500

Data source: *count of cursor resets (it2) in study-1*

Column	Size	Missing	Mean	Std Dev	Std. Error	C.I. of Mean
count static	12	0	1,333	1,371	0,396	0,871
count linear	12	0	1,333	1,231	0,355	0,782
count hybrid-linear	12	0	2,333	1,723	0,497	1,095
Column	Range	Max	Min	Median	25%	75%
count static	4,000	4,000	0,000	1,000	0,000	2,500
count linear	4,000	4,000	0,000	1,000	0,500	2,000
count hybrid-linear	5,000	6,000	1,000	1,500	1,000	3,000

(2) One Way Repeated Measures Analysis of Variance

Data source: *completion time (it1) in study-1*

Normality Test (Shapiro-Wilk): Passed (P = 0,448); **Equal Variance Test:** Passed (P = 0,332)

Treatment Name	N	Missing	Mean	Std Dev	SEM
time static	12	0	36,853	7,352	2,122
time linear	12	0	38,140	10,410	3,005
time hybrid-linear	12	0	56,405	29,169	8,420
Source of Variation	DF	SS	MS	F	P
Between Subjects	11	4399,775	399,980		
Between Treatments	2	2870,134	1435,067	4,680	0,020
Residual	22	6746,240	306,647		
Total	35	14016,149			

The differences in the mean values among the treatment groups are greater than would be expected by chance; there is a statistically significant difference (P = 0,020). To isolate the group or groups that differ from the others use a multiple comparison procedure.

Power of performed test with alpha = 0,050: 0,612

All Pairwise Multiple Comparison Procedures (Tukey Test):

Comparison	Diff of Means	p	q	P	P<0,05
time hybrid-linear vs. time static	19,552	3	3,868	0,031	Yes
time hybrid-linear vs. time linear	18,265	3	3,613	0,046	Yes
time linear vs. time static	1,287	3	0,255	0,982	No

Data source: *completion time (it2) in study-1*

Normality Test (Shapiro-Wilk): Passed (P = 0,920); **Equal Variance Test:** Passed (P = 0,120)

Treatment Name	N	Missing	Mean	Std Dev	SEM
time static	12	0	31,179	5,599	1,616
time linear	12	0	44,624	11,755	3,393
time hybrid-linear	12	0	45,165	16,695	4,819
Source of Variation	DF	SS	MS	F	P
Between Subjects	11	1989,330	180,848		
Between Treatments	2	1506,656	753,328	5,634	0,011
Residual	22	2941,475	133,703		
Total	35	6437,462			

The differences in the mean values among the treatment groups are greater than would be expected by chance; there is a statistically significant difference (P = 0,011). To isolate the group or groups that differ from the others

use a multiple comparison procedure.

Power of performed test with $\alpha = 0,050$: 0,725

All Pairwise Multiple Comparison Procedures (Tukey Test):

Comparison	Diff of Means	p	q	P	P<0,05
time hybrid-linear vs. time static	13,986	3	4,190	0,019	Yes
time hybrid-linear vs. time linear	0,541	3	0,162	0,993	No
time linear vs. time static	13,445	3	4,028	0,024	Yes

(3) Two Way Repeated Measures ANOVA (Two Factor Repetition)

Data source: *it1 and it2 in study-1*

Balanced Design (IT={it1, it2}, MF={static, linear, hybrid-linear})

Dependent Variable: mean completion time

Normality Test (Shapiro-Wilk): Passed ($P = 0,358$); Equal Variance Test: Passed ($P = 0,816$)

Source of Variation	DF	SS	MS	F	P
Subject	5	1569,811	313,962		
IT	1	108,785	108,785	0,762	0,423
IT x Subject	5	713,981	142,796		
MF	2	1695,445	847,723	4,092	0,050
MF x Subject	10	2071,459	207,146		
IT x MF	2	492,950	246,475	1,436	0,283
Residual	10	1716,389	171,639		
Total	35	8368,821	239,109		

The difference in the mean values among the different levels of IT is not great enough to exclude the possibility that the difference is just due to random sampling variability after allowing for the effects of differences in MF. There is not a statistically significant difference ($P = 0,423$).

The difference in the mean values among the different levels of MF is not great enough to exclude the possibility that the difference is just due to random sampling variability after allowing for the effects of differences in IT. There is not a statistically significant difference ($P = 0,050$).

The effect of different levels of IT does not depend on what level of MF is present. There is not a statistically significant interaction between IT and MF ($P = 0,283$).

Power of performed test with $\alpha = 0,0500$: for IT : 0,0500

Power of performed test with $\alpha = 0,0500$: for MF : 0,460

Power of performed test with $\alpha = 0,0500$: for IT x MF : 0,100

A.3.3. Ergebnisse und statistische Auswertungen der Studie 2

1. Durchlauf	Absolute Häufigkeit ($N_{sm} = N_{lm} = 28, N_{zm} = 26$)												Relative Häufigkeit		
	Quadrate			Kreise			Dreiecke			Sterne			Objekte insgesamt		
Objekttyp	sm	lm	zm	sm	lm	zm	sm	lm	zm	sm	lm	zm	sm	lm	zm
Selektionen (s)	sm	lm	zm	sm	lm	zm	sm	lm	zm	sm	lm	zm	sm	lm	zm
s=1	2	2	4	3	4	1	1	0	2	4	7	4	8,9 %	11,6 %	10,5 %
s=2	6	9	6	10	9	10	9	13	9	5	6	6	26,8 %	33 %	29,8 %
s=3	5	8	2	5	5	6	5	2	1	4	3	4	17,0 %	16,1 %	12,5 %
s=4	6	0	4	3	1	1	3	2	1	3	2	3	13,4 %	4,5 %	8,7 %
s=5	2	2	3	1	1	2	5	2	3	1	2	1	8,0 %	6,2 %	8,7 %
s>5	7	7	7	6	8	6	5	9	10	11	8	8	25,9 %	28,6 %	29,8 %

2. Durchlauf	Absolute Häufigkeit ($N_{sm} = 24, N_{lm} = 26, N_{zm} = 22$)												Relative Häufigkeit		
	Quadrate			Kreise			Dreiecke			Sterne			Objekte insgesamt		
Objekttyp	sm	lm	zm	sm	lm	zm	sm	lm	zm	sm	lm	zm	sm	lm	zm
Selektionen (s)	sm	lm	zm	sm	lm	zm	sm	lm	zm	sm	lm	zm	sm	lm	zm
s=1	4	6	2	6	3	4	4	5	1	3	4	4	17,7 %	17,3 %	12,5 %
s=2	6	6	9	6	8	10	5	6	5	6	4	6	24,0 %	23,1 %	34,1 %
s=3	4	3	3	4	8	2	10	5	6	5	2	5	24,0 %	17,3 %	18,1 %
s=4	5	4	3	1	3	1	1	2	3	0	3	0	7,3 %	11,5 %	8,0 %
s=5	1	0	1	1	0	1	1	3	1	4	3	3	7,3 %	5,8 %	6,8 %
s>5	4	7	4	6	4	4	3	5	6	6	10	4	19,7 %	25,0 %	20,5 %

Tabelle A.3.4.: Häufigkeit der benötigten Selektionen für jeden Objekttyp und Mappingfunktion (statisch (sm), linear (lm), zonenartig (zm)) im ersten und zweiten Durchlauf.

Nachfolgend sind die Berichte der statistischen Auswertung mit SigmaPlot beigefügt: (1) die deskriptive Statistik für Anzahl der Cursor Resets (*count of cursor resets*) für it1; (2) die einfaktorische Varianzanalyse für Bearbeitungszeit (*completion time*) und Objektselektionen (*sum selections*); (3) die t-Tests für abhängige Stichproben für die Bearbeitungszeit im ersten und zweiten Durchlauf (*run-1, run-2*) innerhalb jede Bedingung und die Anzahl der Cursor Rests zwischen den Bedingungen.

(1) Descriptive Statistics

Data source: *count of cursor resets in study-2*

Column	Size	Missing	Mean	Std Dev	Std. Error	C.I. of Mean
static	28	1	2,000	2,449	0,471	0,969
linear	28	0	3,964	3,361	0,635	1,303
zone	28	2	9,385	7,595	1,490	3,068

Column	Range	Max	Min	Median	25%	75%
static	11,000	11,000	0,000	1,000	1,000	2,750
linear	16,000	16,000	0,000	3,500	2,000	5,000
zone	34,000	35,000	1,000	7,500	4,000	13,000

(2) One Way Repeated Measures Analysis of Variance

Data source: *completion time in study-2*

Normality Test (Shapiro-Wilk): Passed (P = 0,076); **Equal Variance Test:** Passed (P = 0,110)

Treatment Name	N	Missing	Mean	Std Dev	SEM
static	28	1	186,296	78,552	15,117
linear	28	0	208,357	97,496	18,425
zone	28	2	234,000	124,462	24,409

Source of Variation	DF	SS	MS	F	P
Between Subjects	27	694296,812	25714,697		
Between Treatments	2	47493,087	23746,543	11,004	<0,001
Residual	51	110053,247	2157,907		
Total	80	834524,543	10431,557		

The differences in the mean values among the treatment groups are greater than would be expected by chance; there is a statistically significant difference (P = <0,001). To isolate the group or groups that differ from the others use a multiple comparison procedure.

Power of performed test with alpha = 0,050: 0,987

All Pairwise Multiple Comparison Procedures (Tukey Test):

Comparison	Diff of Means	p	q	P	P<0,05
zone vs. static	60,527	3	6,617	<0,001	Yes
zone vs. linear	35,408	3	3,922	0,021	Yes
linear vs. static	25,119	3	2,822	0,124	No

Data source: *sum selections in study-2*

Normality Test (Shapiro-Wilk): Failed (P < 0,050); **Equal Variance Test:** Passed (P = 0,341)

Friedman Repeated Measures Analysis of Variance on Ranks

Group	N	Missing	Median	25%	75%
static	28	2	31,000	20,500	37,500
linear	28	1	25,000	20,000	49,000
zone	28	4	26,000	20,000	54,750

Chi-square= 1,800 with 2 degrees of freedom. (P = 0,407)

Data source: *sum selections in study-2 (run-1)*

Normality Test (Shapiro-Wilk): Passed (P = 0,234); **Equal Variance Test:** Passed (P = 0,560)

Treatment Name	N	Missing	Mean	Std Dev	SEM
static	14	0	37,357	19,998	5,345
linear	14	0	36,786	21,962	5,870
zone	14	1	41,846	28,815	7,992

Source of Variation	DF	SS	MS	F	P
Between Subjects	13	19513,165	1501,013		
Between Treatments	2	149,068	74,534	0,970	0,393
Residual	25	1920,099	76,804		
Total	40	21637,951	540,949		

(3) Paired t-test

Data source: completion time in study-2

Dependent Variable: completion time (static mapping)

Normality Test (Shapiro-Wilk): Passed (P = 0,425)

Treatment Name	N	Missing	Mean	Std Dev	SEM
run-1	13	0	185,385	59,459	16,491
run-2	13	1	154,000	41,547	11,994
Difference	13	1	35,500	52,549	15,169

t = 2,340 with 11 degrees of freedom. (P = 0,039)

95 percent confidence interval for difference of means: 2,112 to 68,888

The change that occurred with the treatment is greater than would be expected by chance; there is a statistically significant change (P = 0,039).

Power of performed test with alpha = 0,050: 0,481

Data source: completion time in study-2

Dependent Variable: completion time (linear mapping)

Normality Test (Shapiro-Wilk): Passed (P = 0,116)

Treatment Name	N	Missing	Mean	Std Dev	SEM
run-1	13	0	196,000	78,853	21,870
run-2	13	0	180,231	50,607	14,036
Difference	13	0	15,769	59,421	16,480

t = 0,957 with 12 degrees of freedom. (P = 0,358)

95 percent confidence interval for difference of means: -20,139 to 51,677

Data source: completion time in study-2

Dependent Variable: completion time (zone mapping)

Normality Test (Shapiro-Wilk): Failed (P < 0,050); Wilcoxon Signed Rank Test

Group	N	Missing	Median	25%	75%
run-1	13	0	182,000	153,500	311,000
run-2	13	1	171,000	140,750	215,500

W = -18,000 T+ = 30,000 T- = -48,000

Z-Statistic (based on positive ranks) = -0,706 ; P(est.) = 0,505 P(exact) = 0,519

Data source: count of cursor resets in study-2

Normality Test (Shapiro-Wilk): Failed (P < 0,050); Wilcoxon Signed Rank Test

Group	N	Missing	Median	25%	75%
linear	28	0	3,500	2,000	5,000
zone	28	2	7,500	3,750	13,000

W = 222,000 T+ = 237,500 T- = -15,500

Z-Statistic (based on positive ranks) = 3,609 ; (P = <0,001)

The change that occurred with the treatment is greater than would be expected by chance; there is a statistically significant difference (P = <0,001).

A.3.4. Ergebnisse und statistische Auswertungen der Studie 3

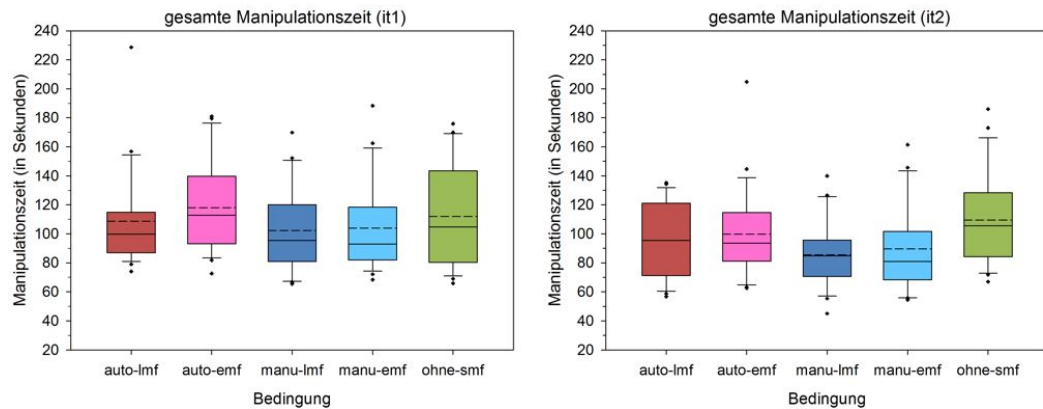


Abbildung A.3.2.: Boxplots der durchschnittlichen gesamte Manipulationszeit für die Bedingungen und Interaktionstechniken.

		absolute Häufigkeit							relative Häufigkeit		
Target (ID)	Bedingung	Quadrat (1) ¹	Kreis (2) ²	Dreieck (3) ²	Stern (4) ²	Quadrat (5) ²	Kreis (6) ¹	Dreieck (7) ²	Stern (8) ¹	\bar{x}_{big}	\bar{x}_{small}
		Sortiersuche ≤ 2 (N=22)							N=66	N=110	
it1	auto-lmf	17	18	13	10	14	21	15	17	83,3 %	63,6 %
	auto-emf	19	18	10	11	13	19	13	16	81,8 %	59,1 %
	manu-lmf	20	15	13	12	12	21	14	16	86,4 %	60,0 %
	manu-emf	20	19	12	13	12	21	14	12	80,3 %	63,6 %
	ohne-smf	14,5	12	7,5	11	8	21	10,5	17,5	80,3 %	44,5 %
it2	auto-lmf	17	17	13	13	7	21	12	19	86,4%	56,4%
	auto-emf	18	16	15	10	5	21	11	19	87,9%	51,8%
	manu-lmf	22	17	10	14	11	22	11	19	95,5%	57,3%
	manu-emf	19	18	13	14	6	22	13	19	89,4%	58,2%
	ohne-smf	19	16	7,5	9	8,5	22	12,5	18	89,4%	48,6%
Bedingung		Fehlerrate $\leq 20\%$ (N=22)							N=66	N=110	
it1	auto-lmf	11	16	6	5	10	17	10	15	65,2%	42,7%
	auto-emf	13	13	6	8	9	17	6	11	62,1%	38,2%
	manu-lmf	16	14	7	9	7	20	9	15	77,3%	41,8%
	manu-emf	17	19	19	7	9	19	9	12	72,7%	57,3%
	ohne-smf	12,5	11	3,5	6,5	5	20	3	15	72,0%	26,4%
it2	auto-lmf	15	12	5	7	3	18	7	15	72,7%	30,9%
	auto-emf	15	12	8	5	3	18	7	13	69,7%	31,8%
	manu-lmf	18	12	5	10	5	19	4	14	77,3%	32,7%
	manu-emf	17	16	16	12	3	17	8	14	72,7%	50,0%
	ohne-smf	17	12	4,5	6	2,5	20	8	15	78,8%	30,0%

¹ Targettyp *big* (Toleranzbereich $\pm 2,54$ mm), ² Targettyp *small* (Toleranzbereich ± 1 mm)

Tabelle A.3.5.: Häufigkeitsverteilung bei geringer Anzahl der Sortiersuche (≤ 2) und geringer Fehlerrate ($\leq 20\%$).

Nachfolgend sind die Berichte der statistischen Auswertung mit SigmaPlot beigefügt: für jede Interaktionstechnik (it1, it2) die einfaktorische Varianzanalyse für (1) Bearbeitungszeit (*task manipulation time*, *mean manipulation time big/small targets*), Sortiersuche (*mean sorting tests big/small targets*), Fehlerrate (*mean error rate big/small targets*) und Anzahl der Button-Events (*button events*); (2) die durchschnittliche Performanz bezüglich Effizienz/Effektivität (z_{EE}), physiologischer Aufwand (z_{PA}) und Nutzerperformanz (z_{NP}); die zweifaktorielle Varianzanalyse für (3) den Vergleich der Fehlerraten zwischen den Targettypen, den Vergleich des physiologischen Aufwandes und der Nutzerperformanz zwischen it1 und it2; (4) die Anzahl

der Sortierversuche und die Nutzerperformanz bezüglich der Kategorien Mobilität und bevorzugter Distanz.

(1) One Way Repeated Measures Analysis of Variance

Data source: task manipulation time (it1) in study-3

Normality Test (Shapiro-Wilk): Passed (P = 0,107); Equal Variance Test: Passed (P = 0,246)

Treatment Name	N	Missing	Mean	Std Dev	SEM
auto-lmf	22	0	108,616	33,683	7,181
auto-emf	22	0	118,098	31,077	6,626
manu-lmf	22	0	102,292	28,534	6,083
manu-emf	22	0	104,009	31,392	6,693
ohne-smf	22	0	112,083	35,279	7,522

Source of Variation	DF	SS	MS	F	P
Between Subjects	21	78876,908	3756,043		
Between Treatments	4	3571,279	892,820	2,572	0,044
Residual	84	29159,346	347,135		
Total	109	111607,533			

Power of performed test with alpha = 0,050: 0,458

All Pairwise Multiple Comparison Procedures (Student-Newman-Keuls Method):

Comparison	Diff of Means	p	q	P	P<0,05
auto-emf vs. manu-lmf	15,806	5	3,979	0,047	Yes
auto-emf vs. manu-emf	14,089	4	3,547	0,066	No
auto-emf vs. auto-lmf	9,482	3	2,387	0,216	Do Not Test
auto-emf vs. ohne-smf	6,015	2	1,514	0,287	Do Not Test
ohne-smf vs. manu-lmf	9,791	4	2,465	0,308	No
ohne-smf vs. manu-emf	8,074	3	2,033	0,327	Do Not Test
ohne-smf vs. auto-lmf	3,467	2	0,873	0,539	Do Not Test
auto-lmf vs. manu-lmf	6,324	3	1,592	0,501	Do Not Test
auto-lmf vs. manu-emf	4,607	2	1,160	0,415	Do Not Test
manu-emf vs. manu-lmf	1,717	2	0,432	0,761	Do Not Test

Data source: mean manipulation time big targets (it1) in study-3

Normality Test (Shapiro-Wilk): Passed (P = 0,112); Equal Variance Test: Passed (P = 0,670)

Treatment Name	N	Missing	Mean	Std Dev	SEM
auto-lmf-big	22	0	12,281	4,617	0,984
auto-emf-big	22	0	12,406	3,109	0,663
manu-lmf-big	22	0	10,677	2,739	0,584
manu-emf-big	22	0	11,011	3,157	0,673
ohne-smf-big	22	0	11,208	3,646	0,777

Source of Variation	DF	SS	MS	F	P
Between Subjects	21	849,204	40,438		
Between Treatments	4	53,505	13,376	2,512	0,048
Residual	84	447,259	5,325		
Total	109	1349,967			

Power of performed test with alpha = 0,050: 0,442

All Pairwise Multiple Comparison Procedures (Student-Newman-Keuls Method):

Comparison	Diff of Means	p	q	P	P<0,05
auto-emf-big vs. manu-lmf-big	1,729	5	3,516	0,104	No
auto-emf-big vs. manu-emf-big	1,395	4	2,837	0,194	Do Not Test
auto-emf-big vs. ohne-smf-big	1,198	3	2,436	0,203	Do Not Test
auto-emf-big vs. auto-lmf-big	0,125	2	0,255	0,858	Do Not Test
auto-lmf-big vs. manu-lmf-big	1,604	4	3,261	0,105	Do Not Test
auto-lmf-big vs. manu-emf-big	1,270	3	2,582	0,168	Do Not Test
auto-lmf-big vs. ohne-smf-big	1,073	2	2,181	0,127	Do Not Test
ohne-smf-big vs. manu-lmf-big	0,531	3	1,080	0,726	Do Not Test
ohne-smf-big vs. manu-emf-big	0,197	2	0,401	0,778	Do Not Test
manu-emf-big vs. manu-lmf-big	0,334	2	0,679	0,633	Do Not Test

Data source: mean manipulation time small targets (it1) in study-3

Normality Test (Shapiro-Wilk): Passed (P = 0,109); Equal Variance Test: Failed (P < 0,050)

Friedman RM ANOVA on Ranks begun

Group	N	Missing	Median	25%	75%
auto-lmf-small	22	0	13,017	11,387	15,120
auto-emf-small	22	0	14,809	12,265	20,895
manu-lmf-small	22	0	12,722	10,212	15,558
manu-emf-small	22	0	12,766	10,539	15,935
ohne-smf-small	22	0	13,935	10,979	18,203

Chi-square= 8,155 with 4 degrees of freedom. (P = 0,086)

Data source: task manipulation time (it2) in study-3

Normality Test (Shapiro-Wilk): Failed ($P < 0,050$); Friedman RM ANOVA on Ranks begun

Group	N	Missing	Median	25%	75%
auto-lmf	22	0	95,379	71,297	121,139
auto-emf	22	0	93,470	81,158	114,707
manu-lmf	22	0	84,830	70,554	95,672
manu-emf	22	0	80,901	68,262	101,654
ohne-smf	22	0	105,576	84,221	128,237

Chi-square= 13,768 with 4 degrees of freedom. ($P = 0,008$)

All Pairwise Multiple Comparison Procedures (Student-Newman-Keuls Method):

Comparison	Diff of Ranks	q	P<0,05
ohne-smf vs. manu-lmf	35,000	4,719	Yes
ohne-smf vs. manu-emf	29,500	4,872	Yes
ohne-smf vs. auto-emf	15,500	3,305	No
ohne-smf vs. auto-lmf	15,000	4,523	Do Not Test
auto-lmf vs. manu-lmf	20,000	3,303	No
auto-lmf vs. manu-emf	14,500	3,091	Do Not Test
auto-lmf vs. auto-emf	0,500	0,151	Do Not Test
auto-emf vs. manu-lmf	19,500	4,157	Do Not Test
auto-emf vs. manu-emf	14,000	4,221	Do Not Test
manu-emf vs. manu-lmf	5,500	1,658	Do Not Test

Data source: mean manipulation time big targets (it2) in study-3

Normality Test (Shapiro-Wilk): Failed ($P < 0,050$); Friedman RM ANOVA on Ranks begun

Group	N	Missing	Median	25%	75%
auto-lmf-big	22	0	9,502	6,838	12,259
auto-emf-big	22	0	10,107	7,005	11,542
manu-lmf-big	22	0	7,118	6,094	10,314
manu-emf-big	22	0	7,522	5,696	10,754
ohne-smf-big	22	0	7,856	6,467	10,608

Chi-square= 9,636 with 4 degrees of freedom. ($P = 0,047$)

All Pairwise Multiple Comparison Procedures (Student-Newman-Keuls Method):

Comparison	Diff of Ranks	q	P<0,05
auto-emf-big vs. manu-emf-big	23,000	3,101	No
auto-emf-big vs. ohne-smf-big	22,000	3,633	Do Not Test
auto-emf-big vs. manu-lmf-big	19,000	4,051	Do Not Test
auto-emf-big vs. auto-lmf-big	1,000	0,302	Do Not Test
auto-lmf-big vs. manu-emf-big	22,000	3,633	Do Not Test
auto-lmf-big vs. ohne-smf-big	21,000	4,477	Do Not Test
auto-lmf-big vs. manu-lmf-big	18,000	5,427	Do Not Test
manu-lmf-big vs. manu-emf-big	4,000	0,853	Do Not Test
manu-lmf-big vs. ohne-smf-big	3,000	0,905	Do Not Test
ohne-smf-big vs. manu-emf-big	1,000	0,302	Do Not Test

Data source: mean manipulation time small targets (it2) in study-3

Normality Test (Shapiro-Wilk): Failed ($P < 0,050$); Friedman RM ANOVA on Ranks begun

Group	N	Missing	Median	25%	75%
auto-lmf-small	22	0	13,576	9,468	15,282
auto-emf-small	22	0	12,377	10,379	16,289
manu-lmf-small	22	0	11,195	9,597	14,321
manu-emf-small	22	0	11,674	10,312	15,452
ohne-smf-small	22	0	15,468	11,291	20,543

Chi-square= 16,218 with 4 degrees of freedom. ($P = 0,003$)

All Pairwise Multiple Comparison Procedures (Student-Newman-Keuls Method):

Comparison	Diff of Ranks	q	P<0,05
ohne-smf-small vs. manu-lmf-small	40,000	5,394	Yes
ohne-smf-small vs. manu-emf-small	31,000	5,119	Yes
ohne-smf-small vs. auto-lmf-small	24,000	5,117	Yes
ohne-smf-small vs. auto-emf-small	20,000	6,030	Yes
auto-emf-small vs. manu-lmf-small	20,000	3,303	No
auto-emf-small vs. manu-emf-small	11,000	2,345	Do Not Test
auto-emf-small vs. auto-lmf-small	4,000	1,206	Do Not Test
auto-lmf-small vs. manu-lmf-small	16,000	3,411	Do Not Test
auto-lmf-small vs. manu-emf-small	7,000	2,111	Do Not Test
manu-emf-small vs. manu-lmf-small	9,000	2,714	Do Not Test

Data source: mean sorting tests big targets (it1) in study-3

Normality Test (Shapiro-Wilk): Failed ($P < 0,050$); Friedman RM ANOVA on Ranks begun

Group	N	Missing	Median	25%	75%
auto-lmf-big	22	0	1,667	1,250	2,083
auto-emf-big	22	0	2,000	1,250	2,333
manu-lmf-big	22	0	1,333	1,000	2,500
manu-emf-big	22	0	1,333	1,250	2,167
ohne-smf-big	22	0	1,500	1,458	2,167

Chi-square= 2,438 with 4 degrees of freedom. ($P = 0,656$)

Data source: mean sorting tests small targets (it1) in study-3

Normality Test (Shapiro-Wilk): Failed ($P < 0,050$); Friedman RM ANOVA on Ranks begun

Group	N	Missing	Median	25%	75%
auto-lmf-small	22	0	2,200	1,800	2,600
auto-emf-small	22	0	2,400	1,950	3,050
manu-lmf-small	22	0	2,100	1,800	3,250
manu-emf-small	22	0	2,200	1,800	2,700
ohne-smf-small	22	0	3,300	2,675	4,250

Chi-square= 27,435 with 4 degrees of freedom. ($P = <0,001$)

All Pairwise Multiple Comparison Procedures (Student-Newman-Keuls Method):

Comparison	Diff of Ranks	q	P<0,05
ohne-smf-small vs. manu-emf-small	46,000	6,203	Yes
ohne-smf-small vs. auto-lmf-small	44,500	7,349	Yes
ohne-smf-small vs. manu-lmf-small	40,500	8,635	Yes
ohne-smf-small vs. auto-emf-small	29,000	8,744	Yes
auto-emf-small vs. manu-emf-small	17,000	2,807	No
auto-emf-small vs. auto-lmf-small	15,500	3,305	Do Not Test
auto-emf-small vs. manu-lmf-small	11,500	3,467	Do Not Test
manu-lmf-small vs. manu-emf-small	5,500	1,173	Do Not Test
manu-lmf-small vs. auto-lmf-small	4,000	1,206	Do Not Test
auto-lmf-small vs. manu-emf-small	1,500	0,452	Do Not Test

Data source: mean sorting tests big targets (it2) in study-3

Normality Test (Shapiro-Wilk): Failed ($P < 0,050$); Friedman RM ANOVA on Ranks begun

Group	N	Missing	Median	25%	75%
auto-lmf-big	22	0	1,333	1,000	2,000
auto-emf-big	22	0	1,500	1,000	2,000
manu-lmf-big	22	0	1,333	1,000	1,667
manu-emf-big	22	0	1,500	1,000	2,000
ohne-smf-big	22	0	1,417	1,167	2,042

Chi-square= 4,062 with 4 degrees of freedom. ($P = 0,398$)

Data source: mean sorting tests small targets (it2) in study-3

Normality Test (Shapiro-Wilk): Failed ($P < 0,050$); Friedman RM ANOVA on Ranks begun

Group	N	Missing	Median	25%	75%
auto-lmf-small	22	0	2,500	1,750	3,650
auto-emf-small	22	0	2,500	1,950	3,100
manu-lmf-small	22	0	2,400	1,950	3,100
manu-emf-small	22	0	2,400	1,750	3,350
ohne-smf-small	22	0	3,050	2,750	4,900

Chi-square= 19,017 with 4 degrees of freedom. ($P = <0,001$)

All Pairwise Multiple Comparison Procedures (Student-Newman-Keuls Method):

Comparison	Diff of Ranks	q	P<0,05
ohne-smf-small vs. manu-emf-small	39,500	5,326	Yes
ohne-smf-small vs. manu-lmf-small	37,500	6,193	Yes
ohne-smf-small vs. auto-emf-small	30,000	6,396	Yes
ohne-smf-small vs. auto-lmf-small	28,000	8,442	Yes
auto-lmf-small vs. manu-emf-small	11,500	1,899	No
auto-lmf-small vs. manu-lmf-small	9,500	2,025	Do Not Test
auto-lmf-small vs. auto-emf-small	2,000	0,603	Do Not Test
auto-emf-small vs. manu-emf-small	9,500	2,025	Do Not Test
auto-emf-small vs. manu-lmf-small	7,500	2,261	Do Not Test
manu-lmf-small vs. manu-emf-small	2,000	0,603	Do Not Test

A. Anhang

Data source: error rate small targets (it1) in study-3

Normality Test (Shapiro-Wilk): Passed ($P = 0,227$); **Equal Variance Test:** Passed ($P = 0,783$)

Source of Variation	DF	SS	MS	F	P
Between Subjects	21	0,911	0,0434		
Between Treatments	4	0,186	0,0466	5,115	<0,001
Residual	84	0,765	0,00911		
Total	109	1,863			

Power of performed test with $\alpha = 0,050$: 0,912

All Pairwise Multiple Comparison Procedures (Student-Newman-Keuls Method):

Comparison	Diff of Means	p	q	P	P<0,05
ohne-smf-small vs. manu-emf-small	0,114	5	5,586	0,002	Yes
ohne-smf-small vs. manu-lmf-small	0,0977	4	4,799	0,006	Yes
ohne-smf-small vs. auto-lmf-small	0,0940	3	4,620	0,005	Yes
ohne-smf-small vs. auto-emf-small	0,0517	2	2,543	0,076	No
auto-emf-small vs. manu-emf-small	0,0619	4	3,043	0,146	No
auto-emf-small vs. manu-lmf-small	0,0459	3	2,256	0,253	Do Not Test
auto-emf-small vs. auto-lmf-small	0,0423	2	2,077	0,146	Do Not Test
auto-lmf-small vs. manu-emf-small	0,0197	3	0,966	0,774	Do Not Test
auto-lmf-small vs. manu-lmf-small	0,00365	2	0,179	0,900	Do Not Test
manu-lmf-small vs. manu-emf-small	0,0160	2	0,787	0,579	Do Not Test

Data source: error rate small targets (it2) in study-3

Normality Test (Shapiro-Wilk): Failed ($P < 0,050$); Friedman RM ANOVA on Ranks begun

Group	N	Missing	Median	25%	75%
auto-lmf-small	22	0	0,397	0,302	0,539
auto-emf-small	22	0	0,460	0,353	0,478
manu-lmf-small	22	0	0,372	0,265	0,486
manu-emf-small	22	0	0,347	0,282	0,480
ohne-smf-small	22	0	0,447	0,382	0,536

Chi-square= 14,603 with 4 degrees of freedom. ($P = 0,006$)

All Pairwise Multiple Comparison Procedures (Student-Newman-Keuls Method):

Comparison	Diff of Ranks	q	P<0,05
ohne-smf-small vs. manu-emf-small	31,500	4,247	Yes
ohne-smf-small vs. manu-lmf-small	31,500	5,202	Yes
ohne-smf-small vs. auto-emf-small	14,000	2,985	No
ohne-smf-small vs. auto-lmf-small	8,000	2,412	Do Not Test
auto-lmf-small vs. manu-emf-small	23,500	3,881	Yes
auto-lmf-small vs. manu-lmf-small	23,500	5,010	Yes
auto-lmf-small vs. auto-emf-small	6,000	1,809	Do Not Test
auto-emf-small vs. manu-emf-small	17,500	3,731	Yes
auto-emf-small vs. manu-lmf-small	17,500	5,276	Yes
manu-lmf-small vs. manu-emf-small	0,000	0,000	No

Data source: button events (it1) in study-3

Normality Test (Shapiro-Wilk): Failed ($P < 0,050$); Friedman RM ANOVA on Ranks begun

Group	N	Missing	Median	25%	75%
auto-lmf	22	0	46,000	33,000	51,500
auto-emf	22	0	52,000	43,500	59,500
manu-lmf	22	0	52,000	47,000	64,250
manu-emf	22	0	52,500	44,750	70,500
ohne-smf	22	0	48,750	43,500	60,875

Chi-square= 20,120 with 4 degrees of freedom. ($P = <0,001$)

All Pairwise Multiple Comparison Procedures (Student-Newman-Keuls Method):

Comparison	Diff of Ranks	q	P<0,05
manu-emf vs. auto-lmf	40,500	5,461	Yes
manu-emf vs. auto-emf	26,500	4,376	Yes
manu-emf vs. ohne-smf	13,000	2,772	No
manu-emf vs. manu-lmf	5,000	1,508	Do Not Test
manu-lmf vs. auto-lmf	35,500	5,863	Yes
manu-lmf vs. auto-emf	21,500	4,584	Yes
manu-lmf vs. ohne-smf	8,000	2,412	Do Not Test
ohne-smf vs. auto-lmf	27,500	5,863	Yes
ohne-smf vs. auto-emf	13,500	4,070	Yes
auto-emf vs. auto-lmf	14,000	4,221	Yes

Data source: button events (it2) in study-3

Normality Test (Shapiro-Wilk): Failed ($P < 0,050$); Friedman RM ANOVA on Ranks begun

Group	N	Missing	Median	25%	75%
auto-lmf	22	0	16,500	13,000	24,500
auto-emf	22	0	17,000	15,000	25,000
manu-lmf	22	0	27,500	24,000	31,000
manu-emf	22	0	29,500	24,000	34,500
ohne-smf	22	0	21,750	18,375	30,125

Chi-square= 20,931 with 4 degrees of freedom. ($P = <0,001$)

All Pairwise Multiple Comparison Procedures (Student-Newman-Keuls Method):

Comparison	Diff of Ranks	q	P<0,05
manu-emf vs. auto-emf	38,000	5,124	Yes
manu-emf vs. auto-lmf	37,500	6,193	Yes
manu-emf vs. ohne-smf	20,000	4,264	Yes
manu-emf vs. manu-lmf	9,500	2,864	Yes
manu-lmf vs. auto-emf	28,500	4,707	Yes
manu-lmf vs. auto-lmf	28,000	5,970	Yes
manu-lmf vs. ohne-smf	10,500	3,166	Yes
ohne-smf vs. auto-emf	18,000	3,838	Yes
ohne-smf vs. auto-lmf	17,500	5,276	Yes
auto-lmf vs. auto-emf	0,500	0,151	No

(2) One Way RM ANOVA for user performance

Data source: efficiency/effectiveness z_{EE} (it1) in study-3

Normality Test (Shapiro-Wilk): Failed ($P < 0,050$); Friedman RM ANOVA on Ranks begun

Group	N	Missing	Median	25%	75%
auto-lmf	22	0	-0,288	-0,619	0,282
auto-emf	22	0	0,0938	-0,381	0,630
manu-lmf	22	0	-0,220	-0,761	0,156
manu-emf	22	0	-0,313	-0,644	0,0441
ohne-smf	22	0	0,0813	-0,228	0,681

Chi-square= 12,036 with 4 degrees of freedom. ($P = 0,017$)

All Pairwise Multiple Comparison Procedures (Student-Newman-Keuls Method):

Comparison	Diff of Ranks	q	P<0,05
ohne-smf vs. manu-emf	32,500	4,382	Yes
ohne-smf vs. manu-lmf	27,500	4,541	Yes
ohne-smf vs. auto-lmf	18,000	3,838	Yes
ohne-smf vs. auto-emf	12,000	3,618	Yes
auto-emf vs. manu-emf	20,500	3,385	No
auto-emf vs. manu-lmf	15,500	3,305	Do Not Test
auto-emf vs. auto-lmf	6,000	1,809	Do Not Test
auto-lmf vs. manu-emf	14,500	3,091	Do Not Test
auto-lmf vs. manu-lmf	9,500	2,864	Do Not Test
manu-lmf vs. manu-emf	5,000	1,508	Do Not Test

Data source: efficiency/effectiveness z_{EE} (it2) in study-3

Normality Test (Shapiro-Wilk): Failed ($P < 0,050$); Friedman RM ANOVA on Ranks begun

Group	N	Missing	Median	25%	75%
auto-lmf	22	0	-0,322	-0,688	0,817
auto-emf	22	0	-0,178	-0,423	0,563
manu-lmf	22	0	-0,443	-0,662	0,0554
manu-emf	22	0	-0,396	-0,821	0,375
ohne-smf	22	0	0,299	-0,434	0,989

Chi-square= 13,345 with 4 degrees of freedom. ($P = 0,010$)

All Pairwise Multiple Comparison Procedures (Student-Newman-Keuls Method):

Comparison	Diff of Ranks	q	P<0,05
ohne-smf vs. manu-lmf	35,000	4,719	Yes
ohne-smf vs. manu-emf	27,000	4,459	Yes
ohne-smf vs. auto-emf	16,000	3,411	Yes
ohne-smf vs. auto-lmf	12,000	3,618	Yes
auto-lmf vs. manu-lmf	23,000	3,798	Yes
auto-lmf vs. manu-emf	15,000	3,198	No
auto-lmf vs. auto-emf	4,000	1,206	Do Not Test
auto-emf vs. manu-lmf	19,000	4,051	Yes
auto-emf vs. manu-emf	11,000	3,317	Do Not Test
manu-emf vs. manu-lmf	8,000	2,412	No

A. Anhang

Data source: physiological effort z_{PA} (it1) in study-3

Normality Test (Shapiro-Wilk): Passed ($P = 0,124$); **Equal Variance Test:** Passed ($P = 0,060$)

Source of Variation	DF	SS	MS	F	P
Between Subjects	21	34,873	1,661		
Between Treatments	4	1,267	0,317	3,977	0,005
Residual	84	6,691	0,0797		
Total	109	42,832			

Power of performed test with $\alpha = 0,050$: 0,776

All Pairwise Multiple Comparison Procedures (Student-Newman-Keuls Method):

Comparison	Diff of Means	p	q	P	P<0,05
ohne-smf vs. manu-lmf	0,292	5	4,859	0,008	Yes
ohne-smf vs. manu-emf	0,279	4	4,638	0,008	Yes
ohne-smf vs. auto-lmf	0,250	3	4,153	0,012	Yes
ohne-smf vs. auto-emf	0,182	2	3,022	0,036	Yes
auto-emf vs. manu-lmf	0,111	4	1,837	0,566	No
auto-emf vs. manu-emf	0,0972	3	1,616	0,491	Do Not Test
auto-emf vs. auto-lmf	0,0680	2	1,131	0,426	Do Not Test
auto-lmf vs. manu-lmf	0,0425	3	0,706	0,872	Do Not Test
auto-lmf vs. manu-emf	0,0292	2	0,485	0,732	Do Not Test
manu-emf vs. manu-lmf	0,0133	2	0,221	0,876	Do Not Test

Data source: physiological effort z_{PA} (it2) in study-3

Normality Test (Shapiro-Wilk): Failed ($P < 0,050$); Friedman RM ANOVA on Ranks begun

Group	N	Missing	Median	25%	75%
auto-lmf	22	0	-0,240	-0,529	0,543
auto-emf	22	0	-0,116	-0,713	0,488
manu-lmf	22	0	-0,182	-0,699	0,479
manu-emf	22	0	-0,240	-0,646	0,544
ohne-smf	22	0	0,193	-0,396	0,674

Chi-square= 7,379 with 4 degrees of freedom. ($P = 0,117$)

Data source: user performance z_{NP} (it1) in study-3

Normality Test (Shapiro-Wilk): Failed ($P < 0,050$); Friedman RM ANOVA on Ranks begun

Group	N	Missing	Median	25%	75%
auto-lmf	22	0	0,161	-1,319	0,779
auto-emf	22	0	-0,113	-0,875	0,882
manu-lmf	22	0	-0,832	-1,710	0,695
manu-emf	22	0	-0,626	-2,410	0,911
ohne-smf	22	0	0,823	-0,132	1,747

Chi-square= 19,818 with 4 degrees of freedom. ($P = <0,001$)

All Pairwise Multiple Comparison Procedures (Student-Newman-Keuls Method):

Comparison	Diff of Ranks	q	P<0,05
ohne-smf vs. manu-lmf	41,000	5,528	Yes
ohne-smf vs. manu-emf	38,000	6,275	Yes
ohne-smf vs. auto-lmf	22,000	4,690	Yes
ohne-smf vs. auto-emf	19,000	5,729	Yes
auto-emf vs. manu-lmf	22,000	3,633	Yes
auto-emf vs. manu-emf	19,000	4,051	Yes
auto-emf vs. auto-lmf	3,000	0,905	No
auto-lmf vs. manu-lmf	19,000	4,051	Yes
auto-lmf vs. manu-emf	16,000	4,824	Yes
manu-emf vs. manu-lmf	3,000	0,905	No

Data source: user performance z_{NP} (it2) in study-3

Normality Test (Shapiro-Wilk): Failed ($P < 0,050$); Friedman RM ANOVA on Ranks begun

Group	N	Missing	Median	25%	75%
auto-lmf	22	0	-0,344	-1,511	1,441
auto-emf	22	0	-0,0117	-1,924	0,986
manu-lmf	22	0	-1,043	-2,235	0,805
manu-emf	22	0	-1,094	-2,192	1,177
ohne-smf	22	0	0,684	-0,783	2,279

Chi-square= 20,182 with 4 degrees of freedom. ($P = <0,001$)

All Pairwise Multiple Comparison Procedures (Student-Newman-Keuls Method):

Comparison	Diff of Ranks	q	P<0,05
ohne-smf vs. manu-lmf	43,000	5,798	Yes
ohne-smf vs. manu-emf	36,000	5,945	Yes
ohne-smf vs. auto-emf	27,000	5,756	Yes
ohne-smf vs. auto-lmf	19,000	5,729	Yes
auto-lmf vs. manu-lmf	24,000	3,963	Yes
auto-lmf vs. manu-emf	17,000	3,624	Yes
auto-lmf vs. auto-emf	8,000	2,412	No
auto-emf vs. manu-lmf	16,000	3,411	Yes
auto-emf vs. manu-emf	9,000	2,714	No
manu-emf vs. manu-lmf	7,000	2,111	No

(3) Two Way Repeated Measures ANOVA (Two Factor Repetition)

Data source: error rate and target type (it1) in study-3

Balanced Design (Target type={big,small}, Condition={auto-lmf, auto-emf, manu-lmf, manu-emf, ohne-smf})

Dependent Variable: error rate

Normality Test (Shapiro-Wilk): Passed (P = 0,781); Equal Variance Test: Passed (P = 0,633)

Source of Variation	DF	SS	MS	F	P
Subject	21	1,644	0,0783		
Condition	4	0,127	0,0317	2,204	0,075
Condition x Subject	84	1,207	0,0144		
Target type	1	1,684	1,684	125,925	<0,001
Target type x Subject	21	0,281	0,0134		
Condition x Target type	4	0,0908	0,0227	2,643	0,039
Residual	84	0,721	0,00859		
Total	219	5,754	0,0263		

Main effects cannot be properly interpreted if significant interaction is determined. This is because the size of a factor's effect depends upon the level of the other factor. The effect of different levels of Condition depends on what level of Target type is present. There is a statistically significant interaction between Condition and Target type (P = 0,039).

Power of performed test with alpha = 0,0500: for Condition : 0,355

Power of performed test with alpha = 0,0500: for Target type : 1,000

Power of performed test with alpha = 0,0500: for Condition x Target type : 0,478

All Pairwise Multiple Comparison Procedures (Student-Newman-Keuls Method):

Comparisons for factor: Target type within Condition

Condition	Comparison	Diff of Means	p	q	P	P<0,05
auto-lmf	small vs. big	0,142	2	6,810	<0,001	Yes
auto-emf	small vs. big	0,179	2	8,605	<0,001	Yes
manu-lmf	small vs. big	0,181	2	8,688	<0,001	Yes
manu-emf	small vs. big	0,128	2	6,144	<0,001	Yes
ohne-smf	small vs. big	0,245	2	11,759	<0,001	Yes

Comparisons for factor: Condition within big

Comparison	Diff of Means	p	q	P	P<0,05
auto-emf vs. manu-lmf	0,0476	5	2,085	0,579	No
auto-emf vs. ohne-smf	0,0139	4	0,610	0,973	Do Not Test
auto-emf vs. manu-emf	0,0107	3	0,468	0,942	Do Not Test
auto-emf vs. auto-lmf	0,00489	2	0,214	0,880	Do Not Test
auto-lmf vs. manu-lmf	0,0427	4	1,872	0,548	Do Not Test
auto-lmf vs. ohne-smf	0,00905	3	0,396	0,958	Do Not Test
auto-lmf vs. manu-emf	0,00579	2	0,254	0,858	Do Not Test
manu-emf vs. manu-lmf	0,0369	3	1,618	0,487	Do Not Test
manu-emf vs. ohne-smf	0,00326	2	0,143	0,920	Do Not Test
ohne-smf vs. manu-lmf	0,0337	2	1,475	0,297	Do Not Test

Comparisons for factor: Condition within small

Comparison	Diff of Means	p	q	P	P<0,05
ohne-smf vs. manu-emf	0,114	5	4,977	0,004	Yes
ohne-smf vs. manu-lmf	0,0977	4	4,276	0,013	Yes
ohne-smf vs. auto-lmf	0,0940	3	4,116	0,010	Yes
ohne-smf vs. auto-emf	0,0517	2	2,266	0,109	No
auto-emf vs. manu-emf	0,0619	4	2,711	0,221	No
auto-emf vs. manu-lmf	0,0459	3	2,010	0,330	Do Not Test
auto-emf vs. auto-lmf	0,0423	2	1,850	0,191	Do Not Test
auto-lmf vs. manu-emf	0,0197	3	0,861	0,815	Do Not Test
auto-lmf vs. manu-lmf	0,00365	2	0,160	0,910	Do Not Test
manu-lmf vs. manu-emf	0,0160	2	0,701	0,620	Do Not Test

Data source: error rate and target type (it2) in study-3

Balanced Design (Target type={big,small}, Condition={auto-lmf, auto-emf, manu-lmf, manu-emf, ohne-smf})

Dependent Variable: error rate

Normality Test (Shapiro-Wilk): Passed (P = 0,375); **Equal Variance Test:** Passed (P = 0,429)

Source of Variation	DF	SS	MS	F	P
Subject	21	1,588	0,0756		
Condition	4	0,112	0,0279	2,003	0,102
Condition x Subject	84	1,172	0,0140		
Target type	1	3,277	3,277	142,302	<0,001
Target type x Subject	21	0,484	0,0230		
Condition x Target type	4	0,0953	0,0238	2,270	0,068
Residual	84	0,882	0,0105		
Total	219	7,610	0,0348		

The difference in the mean values among the different levels of Condition is not great enough to exclude the possibility that the difference is just due to random sampling variability after allowing for the effects of differences in Target type. There is not a statistically significant difference (P = 0,102). The difference in the mean values among the different levels of Target type is greater than would be expected by chance after allowing for effects of differences in Condition. There is a statistically significant difference (P = <0,001). To isolate which group(s) differ from the others use a multiple comparison procedure. The effect of different levels of Condition does not depend on what level of Target type is present. There is not a statistically significant interaction between Condition and Target type (P = 0,068).

Power of performed test with alpha = 0,0500: for Condition : 0,297

Power of performed test with alpha = 0,0500: for Target type : 1,000

Power of performed test with alpha = 0,0500: for Condition x Target type : 0,373

All Pairwise Multiple Comparison Procedures (Student-Newman-Keuls Method):

Comparisons for factor: **Target type**

Comparison	Diff of Means	p	q	P	P<0,05
small vs. big	0,244	2	16,870	<0,001	Yes

Data source: physiological effort z_{PA} it1 and it2 in study-3

Balanced Design (IT={it1, it2}, MF={auto-lmf, auto-emf, manu-lmf, manu-emf, ohne-smf})

Dependent Variable: physiological effort z_{PA}

Normality Test (Shapiro-Wilk): Failed (P < 0,050); **Equal Variance Test:** Failed (P < 0,050)

Source of Variation	DF	SS	MS	F	P
Subject	21	70,217	3,344		
IT	1	0,00150	0,00150	0,00416	0,949
IT x Subject	21	7,574	0,361		
MF	4	2,893	0,723	4,997	0,001
MF x Subject	84	12,156	0,145		
IT x MF	4	0,0589	0,0147	0,195	0,940
Residual	84	6,341	0,0755		
Total	219	99,240	0,453		

The difference in the mean values among the different levels of IT is not great enough to exclude the possibility that the difference is just due to random sampling variability after allowing for the effects of differences in MF. There is not a statistically significant difference (P = 0,949). The difference in the mean values among the different levels of MF is greater than would be expected by chance after allowing for effects of differences in IT. There is a statistically significant difference (P = 0,001). To isolate which group(s) differ from the others use a multiple comparison procedure. The effect of different levels of IT does not depend on what level of MF is present. There is not a statistically significant interaction between IT and MF (P = 0,940).

Power of performed test with alpha = 0,0500: for IT : 0,0500

Power of performed test with alpha = 0,0500: for MF : 0,902

Power of performed test with alpha = 0,0500: for IT x MF : 0,0500

All Pairwise Multiple Comparison Procedures (Student-Newman-Keuls Method):

Comparisons for factor: **MF**

Comparison	Diff of Means	p	q	P	P<0,05
ohne-smf vs. manu-lmf	0,314	5	5,479	0,002	Yes
ohne-smf vs. manu-emf	0,295	4	5,150	0,003	Yes
ohne-smf vs. auto-lmf	0,272	3	4,738	0,004	Yes
ohne-smf vs. auto-emf	0,232	2	4,054	0,005	Yes
auto-emf vs. manu-lmf	0,0817	4	1,425	0,746	No
auto-emf vs. manu-emf	0,0628	3	1,096	0,720	Do Not Test
auto-emf vs. auto-lmf	0,0392	2	0,683	0,630	Do Not Test
auto-lmf vs. manu-lmf	0,0425	3	0,741	0,860	Do Not Test
auto-lmf vs. manu-emf	0,0236	2	0,412	0,772	Do Not Test
manu-emf vs. manu-lmf	0,0189	2	0,329	0,817	Do Not Test

Data source: user performance z_{NP} it1 and it2 in study-3

Balanced Design (IT={it1, it2}, CS-a={auto-lmf, auto-emf})

Dependent Variable: user performance z_{NP}

Normality Test (Shapiro-Wilk): Passed (P = 0,158); **Equal Variance Test:** Passed (P = 1,000)

Source of Variation	DF	SS	MS	F	P
Subject	21	279,936	13,330		
IT	1	0,0500	0,0500	0,0270	0,871
IT x Subject	21	38,862	1,851		
CS-a	1	0,878	0,878	2,137	0,159
CS-a x Subject	21	8,631	0,411		
IT x CS-a	1	0,369	0,369	0,813	0,377
Residual	21	9,519	0,453		
Total	87	338,246	3,888		

Data source: user performance z_{NP} it1 and it2 in study-3

Balanced Design (IT={it1,it2}, CS-m={manu-lmf, manu-emf})

Dependent Variable: user performance z_{NP}

Normality Test (Shapiro-Wilk): Passed (P = 0,239); **Equal Variance Test:** Passed (P = 1,000)

Source of Variation	DF	SS	MS	F	P
Subject	21	246,486	11,737		
IT	1	0,136	0,136	0,0681	0,797
IT x Subject	21	41,776	1,989		
CS-m	1	0,0900	0,0900	0,459	0,505
CS-m x Subject	21	4,118	0,196		
IT x CS-m	1	0,00361	0,00361	0,0123	0,913
Residual	21	6,159	0,293		
Total	87	298,769	3,434		

Data source: user performance z_{NP} it1 and it2 in study-3

One Way Repeated Measures Analysis of Variance

Dependent Variable: user performance z_{NP} with condition "ohne-smf"

Normality Test (Shapiro-Wilk): Passed (P = 0,626); **Equal Variance Test:** Passed (P = 1,000)

Source of Variation	DF	SS	MS	F	P
Between Subjects	21	140,733	6,702		
Between Treatments	1	0,295	0,295	0,176	0,680
Residual	21	35,279	1,680		
Total	43	176,307			

(4) Two Way Repeated Measures ANOVA (One Factor Repetition)

Data source: sorting tests and mobility (it1) in study-3

General Linear Model (Mobility={vertical, horizontal, tactical}, Condition={auto-lmf, ..., ohne-smf})

Dependent Variable: count of sorting tests

Normality Test (Shapiro-Wilk): Failed (P < 0,050); **Equal Variance Test:** Passed (P = 0,074)

Source of Variation	DF	SS	MS	F	P
Mobility	2	1,917	0,959	0,297	0,746
Subject(Mobility)	19	61,338	3,228		
Condition	4	12,614	3,154	3,888	0,006
Mobility x Condition	8	6,116	0,765	0,943	0,487
Residual	76	61,650	0,811		
Total	109	150,963	1,385		

Data source: sorting tests and mobility (it1) in study-3

General Linear Model (Mobility={vertical, horizontal, tactical}, CS-a={auto-lmf, auto-emf})

Dependent Variable: count of sorting tests

Normality Test (Shapiro-Wilk): Passed (P = 0,536); **Equal Variance Test:** Passed (P = 0,134)

Source of Variation	DF	SS	MS	F	P
Mobility	2	0,298	0,149	0,125	0,884
Subject(Mobility)	19	22,695	1,194		
CS-a	1	0,0967	0,0967	0,199	0,661
Mobility x CS-a	2	0,355	0,177	0,364	0,699
Residual	19	9,242	0,486		
Total	43	32,868	0,764		

Data source: sorting tests and mobility (it1) in study-3

General Linear Model (Mobility={vertical, horizontal, tactical}, CS-m={manu-lmf, manu-emf})

Dependent Variable: count of sorting tests

Normality Test (Shapiro-Wilk): Passed (P = 0,336); **Equal Variance Test:** Passed (P = 0,536)

Source of Variation	DF	SS	MS	F	P
Mobility	2	0,274	0,137	0,113	0,894
Subject(Mobility)	19	23,110	1,216		
CS-m	1	0,00174	0,00174	0,00518	0,943
Mobility x CS-m	2	0,135	0,0673	0,201	0,820
Residual	19	6,377	0,336		
Total	43	29,930	0,696		

Data source: sorting tests and mobility (it1) in study-3

General Linear Model (Mobility2={vertical, horizontal mittel-entfernt, horizontal other, tactical},

CS-m={manu-lmf, manu-emf})

Dependent Variable: count of sorting tests

Normality Test (Shapiro-Wilk): Passed (P = 0,318); **Equal Variance Test:** Passed (P = 0,165)

Source of Variation	DF	SS	MS	F	P
Mobility2	3	2,046	0,682	0,575	0,639
Subject(Mobility2)	18	21,337	1,185		
CS-m	1	0,000553	0,000553	0,00161	0,968
Mobility2 x CS-m	3	0,346	0,115	0,337	0,799
Residual	18	6,165	0,343		
Total	43	29,930	0,696		

Data source: sorting tests and mobility (it2) in study-3

General Linear Model (Mobility={vertical, horizontal, tactical}, Condition={auto-lmf, ..., ohne-smf})

Dependent Variable: count of sorting tests

Normality Test (Shapiro-Wilk): Failed (P < 0,050); **Equal Variance Test:** Passed (P = 0,164)

Source of Variation	DF	SS	MS	F	P
Mobility	2	2,236	1,118	0,458	0,640
Subject(Mobility)	19	46,432	2,444		
Condition	4	8,891	2,223	3,199	0,018
Mobility x Condition	8	6,719	0,840	1,209	0,305
Residual	76	52,805	0,695		
Total	109	127,228	1,167		

Data source: sorting tests and mobility (it2) in study-3

General Linear Model (Mobility={vertical, horizontal, tactical}, CS-a={auto-lmf, auto-emf})

Dependent Variable: count of sorting tests

Normality Test (Shapiro-Wilk): Failed (P < 0,050); **Equal Variance Test:** Passed (P = 0,867)

Source of Variation	DF	SS	MS	F	P
Mobility	2	2,740	1,370	1,115	0,348
Subject(Mobility)	19	23,348	1,229		
CS-a	1	0,173	0,173	0,430	0,520
mobility x CS-a	2	0,516	0,258	0,643	0,537
Residual	19	7,629	0,402		
Total	43	34,237	0,796		

Data source: sorting tests and mobility (it2) in study-3

General Linear Model (Mobility={vertical, horizontal, tactical}, CS-m={manu-lmf, manu-emf})

Dependent Variable: count of sorting tests

Normality Test (Shapiro-Wilk): Passed (P = 0,486); **Equal Variance Test:** Passed (P = 0,996)

Source of Variation	DF	SS	MS	F	P
Mobility	2	3,221	1,611	1,662	0,216
Subject(Mobility)	19	18,412	0,969		
CS-m	1	4,894E-033	4,894E-033	2,901E-032	1,000
Mobility x CS-m	2	0,227	0,113	0,672	0,522
Residual	19	3,205	0,169		
Total	43	25,180	0,586		

Data source: mobility and sorting (it2) in study-3

General Linear Model (Mobility2={vertical, horizontal mittel-entfernt, horizontal other, tactical},

CS-m={manu-lmf, manu-emf})

Dependent Variable: count of sorting tests

Normality Test (Shapiro-Wilk): Passed (P = 0,470); **Equal Variance Test:** Passed (P = 0,806)

Source of Variation	DF	SS	MS	F	P
Mobility2	3	3,222	1,074	1,050	0,395
Subject(Mobility2)	18	18,412	1,023		
CS-m	1	0,00284	0,00284	0,0160	0,901
Mobility2 x CS-m	3	0,247	0,0823	0,465	0,710
Residual	18	3,185	0,177		
Total	43	25,180	0,586		

Data source: z_{NP} and mobility (it1) in study-3

General Linear Model (Mobility={vertical, horizontal, tactical}, Condition={auto-lmf, ..., ohne-smf})

Dependent Variable: user performance z_{NP}

Normality Test (Shapiro-Wilk): Failed ($P < 0,050$); **Equal Variance Test:** Passed ($P = 0,526$)

Source of Variation	DF	SS	MS	F	P
Mobility	2	12,515	6,257	0,490	0,620
Subject(Mobility)	19	242,434	12,760		
Condition	4	27,999	7,000	7,322	<0,001
Mobility x Condition	8	2,217	0,277	0,290	0,967
Residual	76	72,659	0,956		
Total	109	357,346	3,278		

Data source: z_{NP} and distance (it1) in study-3

General Linear Model (Distance={nah, nah-mittel, mittel-entfernt}, Condition={auto-lmf, ..., ohne-smf})

Dependent Variable: user performance z_{NP}

Normality Test (Shapiro-Wilk): Failed ($P < 0,050$); **Equal Variance Test:** Failed ($P < 0,050$)

Source of Variation	DF	SS	MS	F	P
Distance	2	14,820	7,410	0,586	0,566
Subject(Distance)	19	240,128	12,638		
Condition	4	21,749	5,437	5,965	<0,001
Distance x Condition	8	5,603	0,700	0,768	0,631
Residual	76	69,272	0,911		
Total	109	357,346	3,278		

Data source: z_{NP} and mobility (it2) in study-3

General Linear Model (Mobility={vertical, horizontal, tactical}, Condition={auto-lmf, ..., ohne-smf})

Dependent Variable: user performance z_{NP}

Normality Test (Shapiro-Wilk): Passed ($P = 0,003$); **Equal Variance Test:** Passed ($P = 0,578$)

Source of Variation	DF	SS	MS	F	P
Mobility	2	68,197	34,098	2,221	0,136
Subject(Mobility)	19	291,704	15,353		
Condition	4	38,860	9,715	7,498	<0,001
Mobility x Condition	8	23,307	2,913	2,249	0,033
Residual	76	98,468	1,296		
Total	109	517,899	4,751		

Main effects cannot be properly interpreted if significant interaction is determined. This is because the size of a factor's effect depends upon the level of the other factor. The effect of different levels of Mobility depends on what level of Condition is present. There is a statistically significant interaction between Mobility and Condition ($P = 0,033$). Power of performed test with $\alpha = 0,0500$: for Mobility : 0,226

Power of performed test with $\alpha = 0,0500$: for Condition : 0,992

Power of performed test with $\alpha = 0,0500$: for Mobility x Condition : 0,532

All Pairwise Multiple Comparison Procedures (Student-Newman-Keuls Method):

Comparisons for factor: **Condition within vertical**

Comparison	Diff of Means	p	q	P	P<0,05
ohne-smf vs. manu-lmf	2,886	5	7,171	<0,001	Yes
ohne-smf vs. manu-emf	2,619	4	6,508	<0,001	Yes
ohne-smf vs. auto-lmf	2,129	3	5,290	0,001	Yes
ohne-smf vs. auto-emf	1,738	2	4,318	0,003	Yes
auto-emf vs. manu-lmf	1,148	4	2,854	0,191	No
auto-emf vs. manu-emf	0,881	3	2,190	0,274	Do Not Test
auto-emf vs. auto-lmf	0,391	2	0,973	0,494	Do Not Test
auto-lmf vs. manu-lmf	0,757	3	1,881	0,383	Do Not Test
auto-lmf vs. manu-emf	0,490	2	1,218	0,392	Do Not Test
manu-emf vs. manu-lmf	0,267	2	0,664	0,640	Do Not Test

Comparisons for factor: **Condition within horizontal**

Comparison	Diff of Means	p	q	P	P<0,05
ohne-smf vs. manu-lmf	0,554	5	1,686	0,756	No
ohne-smf vs. manu-emf	0,525	4	1,597	0,673	Do Not Test
ohne-smf vs. auto-emf	0,271	3	0,823	0,830	Do Not Test
ohne-smf vs. auto-lmf	0,253	2	0,770	0,588	Do Not Test
auto-lmf vs. manu-lmf	0,301	4	0,916	0,916	Do Not Test
auto-lmf vs. manu-emf	0,272	3	0,828	0,828	Do Not Test
auto-lmf vs. auto-emf	0,0176	2	0,0536	0,970	Do Not Test
auto-emf vs. manu-lmf	0,283	3	0,862	0,815	Do Not Test
auto-emf vs. manu-emf	0,254	2	0,774	0,586	Do Not Test
manu-emf vs. manu-lmf	0,0290	2	0,0883	0,950	Do Not Test

Comparisons for factor: **Condition within tactical**

Comparison	Diff of Means	p	q	P	P<0,05
ohne-smf vs. manu-emf	3,100	5	3,852	0,060	No
ohne-smf vs. manu-lmf	2,702	4	3,357	0,091	Do Not Test
ohne-smf vs. auto-emf	1,017	3	1,264	0,646	Do Not Test
ohne-smf vs. auto-lmf	0,331	2	0,412	0,772	Do Not Test
auto-lmf vs. manu-emf	2,769	4	3,440	0,080	Do Not Test
auto-lmf vs. manu-lmf	2,371	3	2,946	0,100	Do Not Test
auto-lmf vs. auto-emf	0,686	2	0,852	0,549	Do Not Test
auto-emf vs. manu-emf	2,083	3	2,588	0,167	Do Not Test
auto-emf vs. manu-lmf	1,685	2	2,093	0,143	Do Not Test
manu-lmf vs. manu-emf	0,398	2	0,494	0,728	Do Not Test

Comparisons for factor: **Mobility within auto-lmf**

Comparison	Diff of Means	p	q	P	P<0,05
tactical vs. vertical	3,368	3	2,973	0,105	No
tactical vs. horizontal	1,499	2	1,369	0,340	Do Not Test
horizontal vs. vertical	1,870	2	2,858	0,052	Do Not Test

Comparisons for factor: **Mobility within auto-emf**

Comparison	Diff of Means	p	q	P	P<0,05
tactical vs. vertical	2,291	3	2,022	0,338	No
tactical vs. horizontal	0,830	2	0,759	0,595	Do Not Test
horizontal vs. vertical	1,461	2	2,233	0,124	Do Not Test

Comparisons for factor: **Mobility within manu-lmf**

Comparison	Diff of Means	p	q	P	P<0,05
horizontal vs. vertical	2,326	3	3,556	0,044	Yes
horizontal vs. tactical	0,571	2	0,522	0,715	No
tactical vs. vertical	1,754	2	1,549	0,282	No

Comparisons for factor: **Mobility within manu-emf**

Comparison	Diff of Means	p	q	P	P<0,05
horizontal vs. vertical	2,088	3	3,192	0,077	No
horizontal vs. tactical	0,998	2	0,912	0,524	Do Not Test
tactical vs. vertical	1,090	2	0,962	0,501	Do Not Test

Comparisons for factor: **Mobility within ohne-smf**

Comparison	Diff of Means	p	q	P	P<0,05
tactical vs. horizontal	1,577	3	1,441	0,570	No
tactical vs. vertical	1,571	2	1,386	0,334	Do Not Test
vertical vs. horizontal	0,00642	2	0,00981	0,995	Do Not Test

Data source: z_{NP} and distance (it2) in study-3

General Linear Model (Distance={nah, nah-mittel, mittel-entfernt}, Condition={auto-lmf, ..., ohne-smf})

Dependent Variable: user performance z_{NP}

Normality Test (Shapiro-Wilk): Failed ($P < 0,001$); **Equal Variance Test:** Passed ($P = 0,433$)

Source of Variation	DF	SS	MS	F	P
Distance	2	14,024	7,012	0,385	0,686
Subject(Distance)	19	345,878	18,204		
Condition	4	29,446	7,362	5,763	<0,001
Distance x Condition	8	24,698	3,087	2,417	0,022
Residual	76	97,077	1,277		
Total	109	517,899	4,751		

Main effects cannot be properly interpreted if significant interaction is determined. This is because the size of a factor's effect depends upon the level of the other factor. The effect of different levels of Distance depends on what level of Condition is present. There is a statistically significant interaction between Distance and Condition ($P = 0,022$).

Power of performed test with alpha = 0,0500: for Distance : 0,0500

Power of performed test with alpha = 0,0500: for Condition : 0,950

Power of performed test with alpha = 0,0500: for Distance x Condition : 0,598

All Pairwise Multiple Comparison Procedures (Student-Newman-Keuls Method):

Comparisons for factor: **Condition within nah**

Comparison	Diff of Means	p	q	P	P<0,05
auto-lmf vs. manu-lmf	2,588	5	5,121	0,005	Yes
auto-lmf vs. manu-emf	2,390	4	4,728	0,007	Yes
auto-lmf vs. auto-emf	0,354	3	0,700	0,874	No
auto-lmf vs. ohne-smf	0,285	2	0,565	0,691	Do Not Test
ohne-smf vs. manu-lmf	2,303	4	4,556	0,010	Yes
ohne-smf vs. manu-emf	2,104	3	4,163	0,012	Yes
ohne-smf vs. auto-emf	0,0682	2	0,135	0,924	Do Not Test
auto-emf vs. manu-lmf	2,235	3	4,421	0,007	Yes
auto-emf vs. manu-emf	2,036	2	4,028	0,006	Yes
manu-emf vs. manu-lmf	0,199	2	0,393	0,782	No

Comparisons for factor: **Condition within mittel-entf**

Comparison	Diff of Means	p	q	P	P<0,05
ohne-smf vs. auto-emf	1,390	5	2,749	0,304	No
ohne-smf vs. auto-lmf	0,910	4	1,801	0,583	Do Not Test
ohne-smf vs. manu-lmf	0,612	3	1,211	0,669	Do Not Test
ohne-smf vs. manu-emf	0,609	2	1,206	0,397	Do Not Test
manu-emf vs. auto-emf	0,780	4	1,544	0,696	Do Not Test
manu-emf vs. auto-lmf	0,301	3	0,595	0,907	Do Not Test
manu-emf vs. manu-lmf	0,00265	2	0,00524	0,997	Do Not Test
manu-lmf vs. auto-emf	0,778	3	1,539	0,525	Do Not Test
manu-lmf vs. auto-lmf	0,298	2	0,590	0,678	Do Not Test
auto-lmf vs. auto-emf	0,479	2	0,949	0,505	Do Not Test

Comparisons for factor: **Condition within nah-mittel**

Comparison	Diff of Means	p	q	P	P<0,05
ohne-smf vs. manu-lmf	1,714	5	5,253	0,004	Yes
ohne-smf vs. manu-emf	1,657	4	5,079	0,003	Yes
ohne-smf vs. auto-lmf	1,467	3	4,497	0,006	Yes
ohne-smf vs. auto-emf	0,991	2	3,038	0,035	Yes
auto-emf vs. manu-lmf	0,723	4	2,215	0,404	No
auto-emf vs. manu-emf	0,666	3	2,041	0,324	Do Not Test
auto-emf vs. auto-lmf	0,476	2	1,459	0,305	Do Not Test
auto-lmf vs. manu-lmf	0,247	3	0,756	0,855	Do Not Test
auto-lmf vs. manu-emf	0,190	2	0,582	0,682	Do Not Test
manu-emf vs. manu-lmf	0,0569	2	0,174	0,902	Do Not Test

Comparisons for factor: **Distance within Condition**

Condition	Comparison	Diff of Means	p	q	P	P<0,05
auto-lmf	nah vs. nah-mittel	2,297	3	2,826	0,130	No
	nah vs. mittel-entf	1,768	2	1,831	0,205	Do Not Test
	mittel-entf vs. nah-mittel	0,529	2	0,651	0,649	Do Not Test
auto-emf	nah vs. mittel-entf	1,894	3	1,961	0,360	No
	nah vs. nah-mittel	1,467	2	1,805	0,212	Do Not Test
	nah-mittel vs. mittel-entf	0,427	2	0,525	0,713	Do Not Test
manu-lmf	mittel-entf vs. nah	1,119	3	1,158	0,694	No
	mittel-entf vs. nah-mittel	1,074	2	1,321	0,358	Do Not Test
	nah-mittel vs. nah	0,0448	2	0,0552	0,969	Do Not Test
manu-emf	mittel-entf vs. nah-mittel	1,019	3	1,254	0,653	No
	mittel-entf vs. nah	0,922	2	0,955	0,505	Do Not Test
	nah vs. nah-mittel	0,0970	2	0,119	0,933	Do Not Test
ohne-smf	nah vs. mittel-entf	0,572	3	0,593	0,908	No
	nah vs. nah-mittel	0,544	2	0,670	0,639	Do Not Test
	nah-mittel vs. mittel-entf	0,0281	2	0,0346	0,981	Do Not Test

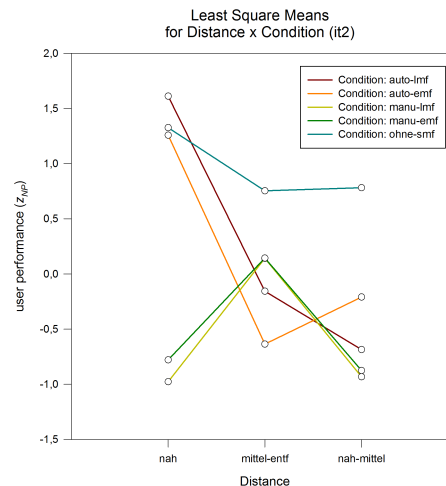
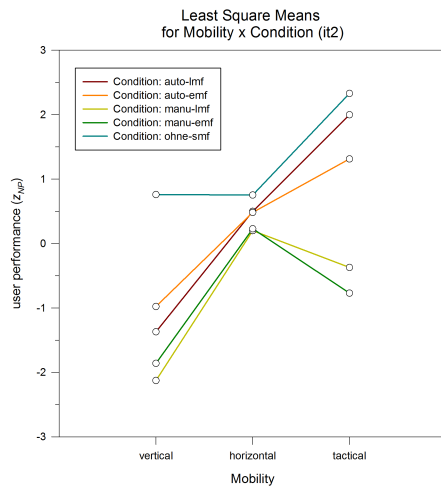


Abbildung A.3.3.: Interaktionsdiagramm der zweifaktoriellen Varianzanalyse bei it2 bezüglich der Wechselwirkung zwischen Bedingung und Mobilität (links) bzw. bevorzugter Distanz (rechts).

A. Anhang

abhängige Variable	IT	Signifikanzprüfung	Paarvergleich (SNK-Test)
Manipulationszeit (gesamt)	it1	$F(4,84)=2,57$ $p=0,044$	auto-emf vs. manu-lmf $p=0,047$
Manipulationszeit (\bar{x}_{big})	it1	$F(4,84)=2,51$ $p=0,048$	n.s.
Manipulationszeit (\bar{x}_{small})	it1	$\chi^2(4)=8,16$ $p=0,086$	-
Manipulationszeit (gesamt)	it2	$\chi^2(4)=13,77$ $p=0,008$	oIS vs. manu $p<0,05$
Manipulationszeit (\bar{x}_{big})	it2	$\chi^2(4)=9,64$ $p=0,047$	n.s.
Manipulationszeit (\bar{x}_{small})	it2	$\chi^2(4)=16,22$ $p=0,003$	oIS vs. mIS $p<0,05$
Sortierversuche (\bar{x}_{big})	it1	$\chi^2(4)=2,44$ $p=0,656$	-
Sortierversuche (\bar{x}_{small})	it1	$\chi^2(4)=27,44$ $p<0,001$	oIS vs. mIS $p<0,05$
Sortierversuche (\bar{x}_{big})	it2	$\chi^2(4)=4,06$ $p=0,398$	-
Sortierversuche (\bar{x}_{small})	it2	$\chi^2(4)=19,02$ $p<0,001$	oIS vs. mIS $p<0,05$
Fehlerrate (\bar{x}_{small})	it1	$F(4,84)=5,12$ $p<0,001$	oIS vs. manu $p<0,01$; oIS vs. auto-lmf $p=0,005$
Fehlerrate (\bar{x}_{small})	it2	$\chi^2(4)=14,60$ $p=0,006$	oIS vs. manu $p<0,01$; auto vs. manu $p<0,05$
Button-Events (gesamt)	it1	$\chi^2(4)=20,12$ $p<0,001$	manu vs. auto, oIS vs. auto, auto-emf vs. auto-lmf $p<0,05$
Button-Events (gesamt)	it2	$\chi^2(4)=20,93$ $p<0,001$	oIS vs. mIS, manu vs. auto, manu-emf vs. manu-lmf $p<0,05$
Effizient/ Effektivität	it1	$\chi^2(4)=12,04$ $p=0,017$	oIS vs. mIS $p<0,05$
Effizient/ Effektivität	it2	$\chi^2(4)=13,45$ $p=0,010$	oIS vs. mIS, auto vs. manu-lmf $p<0,05$
physiologischer Aufwand	it1	$F(4,84)=3,98$ $p=0,005$	oIS vs. manu $p=0,008$; oIS vs. auto $p<0,05$
physiologischer Aufwand	it2	$\chi^2(4)=7,38$ $p=0,117$	-
Nutzerperformanz	it1	$\chi^2(4)=19,82$ $p<0,001$	oIS vs. mIS, auto vs. manu $p<0,05$
Nutzerperformanz	it2	$\chi^2(4)=20,18$ $p<0,001$	oIS vs. mIS, auto vs. manu-lmf, auto-lmf vs. manu-emf $p<0,05$
Fehlerrate	it1	$F_{Condition}(1,84)=2,20$ $p=0,075$	-
		$F_{Target}(2,84)=125,93$ $p<0,001$	big vs. small $p<0,05$
		$F_{C \times T}(4,84)=2,64$ $p=0,039$	small: oIS vs. manu, oIS vs. auto-lmf $p<0,05$
Fehlerrate	it2	$F_{Condition}(1,84)=2,00$ $p=0,102$	-
		$F_{Target}(2,84)=142,302$ $p<0,001$	big vs. small $p<0,05$
		$F_{C \times T}(4,84)=2,27$ $p=0,068$	-
physiologischer Aufwand	it1, it2	$F_{IT}(1,84)=0,004$ $p=0,949$	-
		$F_{MF}(4,84)=4,997$ $p=0,001$	oIS vs. mIS $p \leq 0,005$
		$F_{IT \times MF}(4,84)=0,195$ $p=0,940$	-
Nutzerperformanz (auto)	it1, it2	$F_{IT}(1,84)=0,03$ $p=0,871$	-
		$F_{CS-a}(1,84)=2,14$ $p=0,159$	-
		$F_{IT \times CS-a}(1,84)=0,81$ $p=0,377$	-
Nutzerperformanz (manu)	it1, it2	$F_{IT}(1,84)=0,07$ $p=0,797$	-
		$F_{CS-m}(1,84)=0,46$ $p=0,505$	-
		$F_{IT \times CS-m}(1,84)=0,01$ $p=0,913$	-
Nutzerperformanz (oIS)	it1, it2	$F(1,21)=0,18$ $p=0,680$	-
Sortierversuche nach Mobilität	it1	$F_{Mobility}(2,76)=0,297$ $p=0,746$	-
		$F_{Condition}(4,76)=3,89$ $p=0,006$	nicht Untersuchungsgegenstand
		$F_{M \times C}(8,76)=0,94$ $p=0,487$	-
Sortierversuche nach Mobilität (auto)	it1	$F_{Mobility}(2,19)=0,13$ $p=0,884$	-
		$F_{CS-a}(1,19)=0,20$ $p=0,661$	-
		$F_{M \times CS-a}(2,19)=0,36$ $p=0,699$	-
Sortierversuche nach Mobilität (manu)	it1	$F_{Mobility}(2,19)=0,11$ $p=0,894$	-
		$F_{CS-m}(1,19)=0,005$ $p=0,943$	-
		$F_{M \times CS-m}(2,19)=0,20$ $p=0,820$	-
Sortierversuche nach Mobilität-2 (manu)	it1	$F_{Mobility}(3,18)=0,58$ $p=0,64$	-
		$F_{CS-m}(1,18)=0,002$ $p=0,968$	-
		$F_{M \times CS-m}(3,18)=0,34$ $p=0,799$	-
Sortierversuche nach Mobilität	it2	$F_{Mobility}(2,76)=0,56$ $p=0,640$	-
		$F_{Condition}(4,76)=3,20$ $p=0,018$	nicht Untersuchungsgegenstand
		$F_{M \times C}(8,76)=1,21$ $p=0,305$	-
Sortierversuche nach Mobilität (auto)	it2	$F_{Mobility}(2,19)=1,12$ $p=0,348$	-
		$F_{CS-a}(1,19)=0,43$ $p=0,520$	-
		$F_{M \times CS-a}(2,19)=0,64$ $p=0,537$	-
Sortierversuche nach Mobilität (manu)	it2	$F_{Mobility}(2,19)=1,662$ $p=0,216$	-
		$F_{CS-m}(1,19)=0$ $p=1,0$	-
		$F_{M \times CS-m}(2,19)=0,67$ $p=0,522$	-

Fortsetzung auf der nächsten Seite

abhängige Variable	IT	Signifikanzprüfung	Paarvergleich (SNK-Test)
Sortiersuche nach Mobilität-2 (manu)	it2	$F_{Mobility}(3,18)=1,05$ $p=0,395$	–
		$F_{CS-m}(1,18)=0,02$ $p=0,901$	–
		$F_{MxCs-m}(3,18)=0,465$ $p=0,710$	–
Nutzerperformanz nach Mobilität	it1	$F_{Mobility}(2,76)=0,49$ $p=0,620$	–
		$F_{Condition}(4,76)=7,32$ $p<0,001$	nicht Untersuchungsgegenstand
		$F_{MxC}(8,76)=0,29$ $p=0,967$	–
Nutzerperformanz nach Distanz	it1	$F_{Distance}(2,76)=0,59$ $p=0,566$	–
		$F_{Condition}(4,76)=5,97$ $p<0,001$	nicht Untersuchungsgegenstand
		$F_{DxC}(8,76)=0,77$ $p=0,631$	–
Nutzerperformanz nach Mobilität	it2	$F_{Mobility}(2,76)=2,22$ $p=0,136$	–
		$F_{Condition}(4,76)=7,50$ $p<0,001$	Haupteffekt nicht interpretierbar wegen Wechselwirkung
		$F_{MxC}(8,76)=2,25$ $p=0,033$	vertical: oIS vs. mIS $p\leq 0,003$ manu-lmf: horizontal vs. vertical $p=0,04$
Nutzerperformanz nach Distanz	it2	$F_{Distance}(2,76)=0,39$ $p=0,686$	–
		$F_{Condition}(4,76)=5,76$ $p<0,001$	Haupteffekt nicht interpretierbar wegen Wechselwirkung
		$F_{DxC}(8,76)=2,42$ $p=0,022$	nah: auto vs. manu $p\leq 0,007$ und oIS vs. manu $p<0,05$ nah-mittel: oIS vs. mIS $p<0,05$

Targettyp: große Targets (big, $\pm 2,5$ mm), kleine Targets (small, ± 1 mm)

Cursorwechsel: automatisch (auto, CS-a), manuell (manu, CS-m)

Mappingfunktion: mit Interaction Scaling (mIS: auto-lmf, auto-emf, manu-lmf, manu-emf), ohne Interaction Scaling (oIS: ohne-smf)

Tabelle A.3.6.: Zusammenfassung der durchgeführten Signifikanztests in Studie 3.

In den Tabellen A.3.7 wurden die durchschnittlichen z-Werte der Teilbereiche der Performanz und die resultierende gesamte Nutzerperformanz für die jeweiligen Bedingungen zusammengefasst, allgemein für alle Probanden und nach beobachtetem Verhalten:

Bedingung	Effizienz/Effektivität (z_{EE})		Physiologischer Aufwand (z_{PA})		Zufriedenheit (z_{TLX})		Nutzerperformanz (z_{NP})		
	\bar{x}	Md	\bar{x}	Md	\bar{x}	Md	\bar{x}	Md	
it1	auto-lmf	-0,10	-0,29	-0,08	0,08	0,05	0,05	-0,13	0,16
	auto-emf	0,16	0,09	-0,01	-0,02			0,20	-0,11
	manu-lmf	-0,23	-0,22	-0,12	-0,02	-0,26	0,14	-0,61	-0,83
	manu-emf	-0,19	-0,31	-0,11	-0,02			-0,56	-0,63
	ohne-smf	0,36	0,08	0,17	0,20			-0,74	0,82
it2	auto-lmf	0,02	-0,32	-0,08	-0,24	0,02	0,11	-0,04	-0,34
	auto-emf	0,08	-0,18	-0,07	-0,12			0,03	-0,01
	manu-lmf	-0,28	-0,44	-0,13	-0,18	-0,29	-0,32	-0,70	-1,04
	manu-emf	-0,23	-0,40	-0,10	-0,24			-0,62	-1,09
	ohne-smf	0,42	0,30	0,21	0,19			0,27	0,17

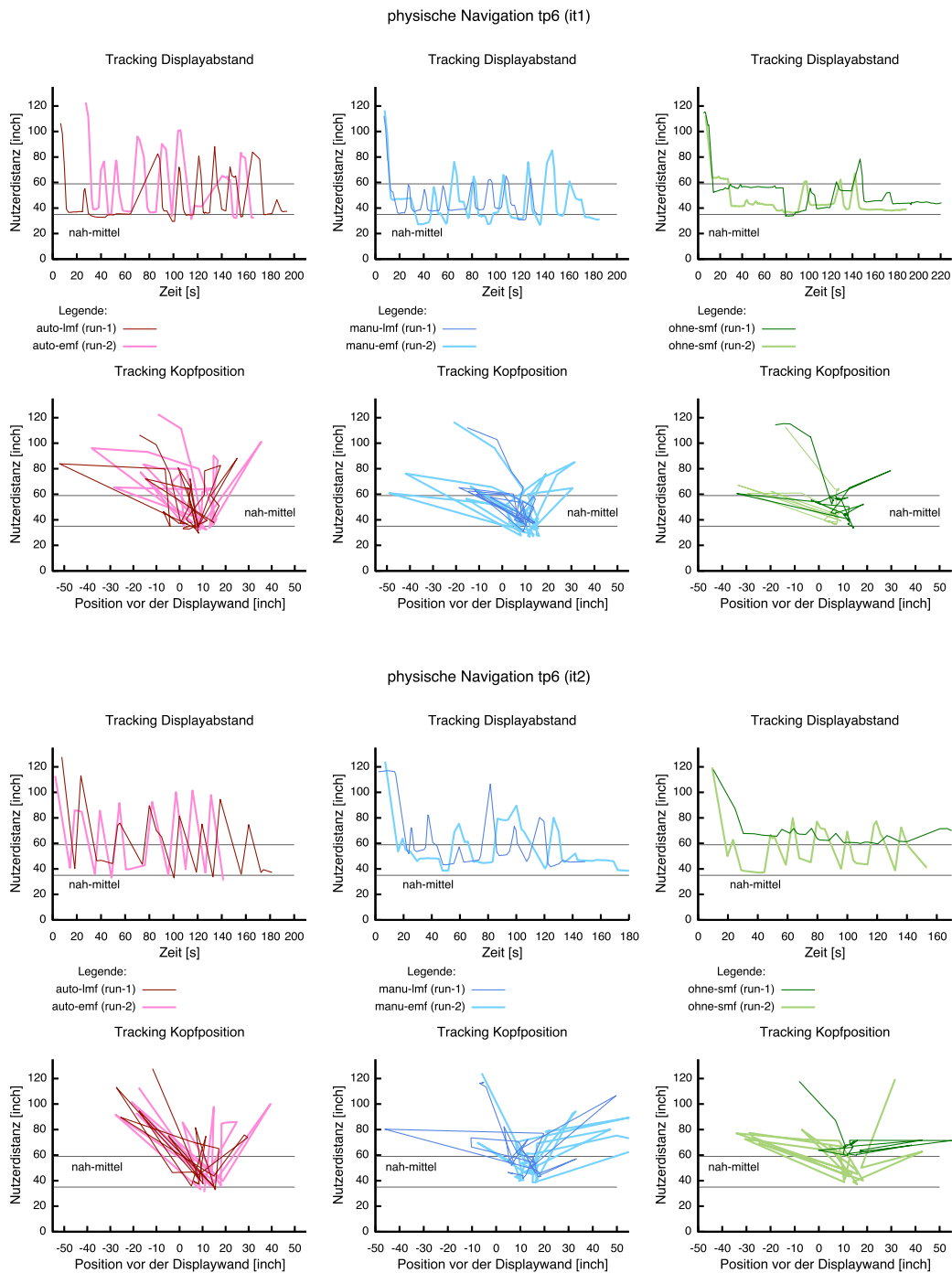
Bedingung	Mobilität/Laufbewegung						bevorzugte Distanz						
	vertikal		horizontal		Taktik		nah		nah-mittel		mittel-entfernt		
	\bar{x}	Md	\bar{x}	Md	\bar{x}	Md	\bar{x}	Md	\bar{x}	Md	\bar{x}	Md	
it1: z_{EE}	auto-lmf	-0,42	-0,15	-0,28	-0,67	0,90	0,84	0,10	0,72	-0,42	0,06	0,28	-0,05
	auto-emf	-0,23	-0,55	-0,05	-0,29	0,13	0,10	0,22	0,26	-0,32	-0,36	0,14	-0,11
	auto-emf	0,07	0,09	0,33	-0,10	0,05	0,25	0,35	0,29	0,11	0,08	0,17	-0,17
	manu-lmf	-0,16	-0,21	-0,21	-0,12	-0,47	-0,57	-0,22	-0,58	-0,21	-0,18	-0,28	-0,25
	manu-emf	-0,09	-0,24	-0,23	-0,51	-0,38	-0,29	-0,25	-0,45	-0,18	-0,14	-0,19	-0,56
	ohne-smf	0,54	0,45	0,36	0,08	-0,07	-0,05	0,55	0,34	0,04	0,01	0,84	0,17
it1: z_{NP}	auto-lmf	-0,42	-0,15	-0,28	-0,67	0,90	0,84	0,10	0,72	-0,42	0,06	0,28	-0,05
	auto-emf	-0,05	-0,34	0,22	-0,48	0,81	0,71	0,24	0,10	0,10	-0,40	0,37	-0,44
	manu-lmf	-0,74	-0,81	-0,63	-1,04	-0,23	0,02	-1,41	-1,18	-0,76	-0,77	0,00	-0,67
	manu-emf	-0,64	-0,70	-0,68	-1,18	-0,11	0,33	-1,40	-1,39	-0,70	-0,99	0,06	-0,26
	ohne-smf	0,39	0,19	0,65	0,83	1,78	1,93	-0,21	0,03	0,50	1,46	1,54	0,72
	auto-lmf	-0,40	-0,54	0,32	0,67	0,97	0,97	0,75	1,16	-0,22	-0,45	-0,14	-0,35
it2: z_{EE}	auto-emf	-0,04	-0,19	0,17	-0,24	0,33	0,33	0,46	0,56	0,17	-0,08	-0,55	-0,37
	manu-lmf	-0,57	-0,62	-0,02	-0,19	0,09	0,09	-0,28	-0,45	-0,33	-0,34	-0,18	-0,49
	manu-emf	-0,48	-0,49	0,06	0,27	-0,22	-0,22	-0,11	-0,31	-0,29	-0,50	-0,21	-0,29
	ohne-smf	0,40	0,29	0,44	0,50	0,46	0,46	0,46	0,55	0,43	0,30	0,36	-0,04
	auto-lmf	-1,11	-0,44	0,81	-0,35	2,00	2,00	1,61	1,39	-0,69	-1,05	-0,16	-0,35
it2: z_{NP}	auto-emf	-0,72	-0,03	0,65	-0,87	1,31	1,31	1,26	0,57	-0,21	-0,47	-0,64	-0,87
	manu-lmf	-1,70	-2,17	0,46	0,68	-0,37	-0,37	-0,98	-1,74	-0,93	-1,04	0,14	0,77
	manu-emf	-1,53	-1,79	0,52	1,09	-0,77	-0,77	-0,78	-1,75	-0,88	-1,09	0,14	0,14
	ohne-smf	0,52	0,08	1,04	1,62	2,33	2,33	1,33	0,92	0,78	-0,18	0,75	1,14

Tabelle A.3.7.: Die durchschnittlichen z-Werte der Nutzerperformanz nach Bedingung und Kategorie.

A.3.4.1. Repräsentative Laufprofile aus Studie 3

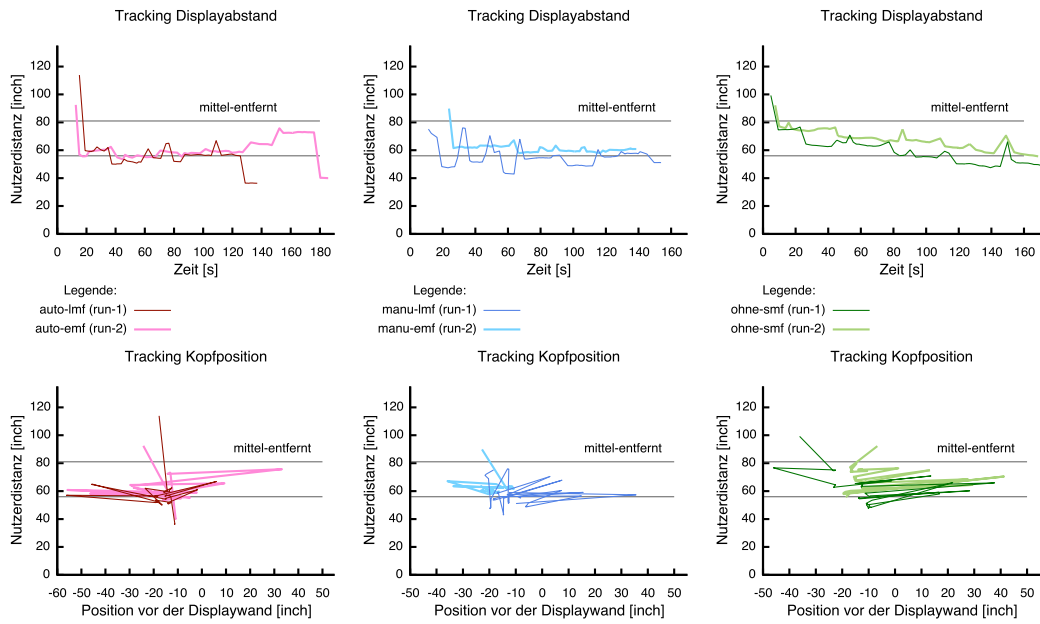
Nachfolgend sind repräsentative Laufprofile von Probanden für die unterschiedlichen Verhaltensweisen bezüglich Mobilität und bevorzugter Distanz bei der physischen Navigation dargestellt.

Der Proband sortierte die Objekte nacheinander ein und nutzte den gesamten Interaktionsraum (Mobilität vertikale Laufbewegung, bevorzugte Distanz nah-mittel):

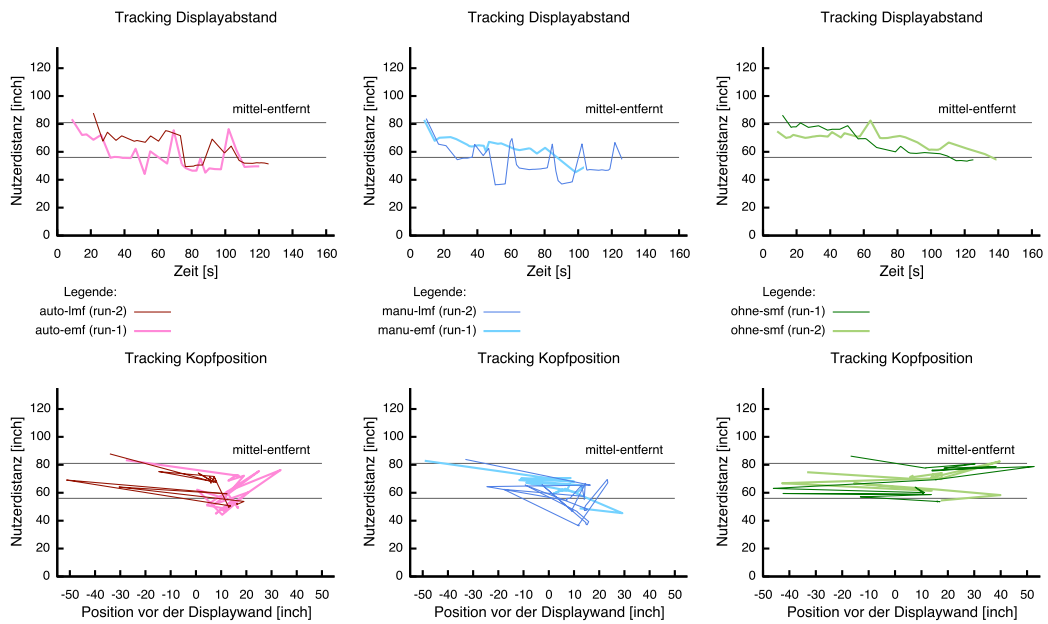


Der Proband sortierte die Objekte nacheinander ein und bevorzugte einen konstanten Interaktionsabstand (Mobilität horizontale Laufbewegung, bevorzugte Distanz mittel-entfernt):

physische Navigation tp09 (it1)

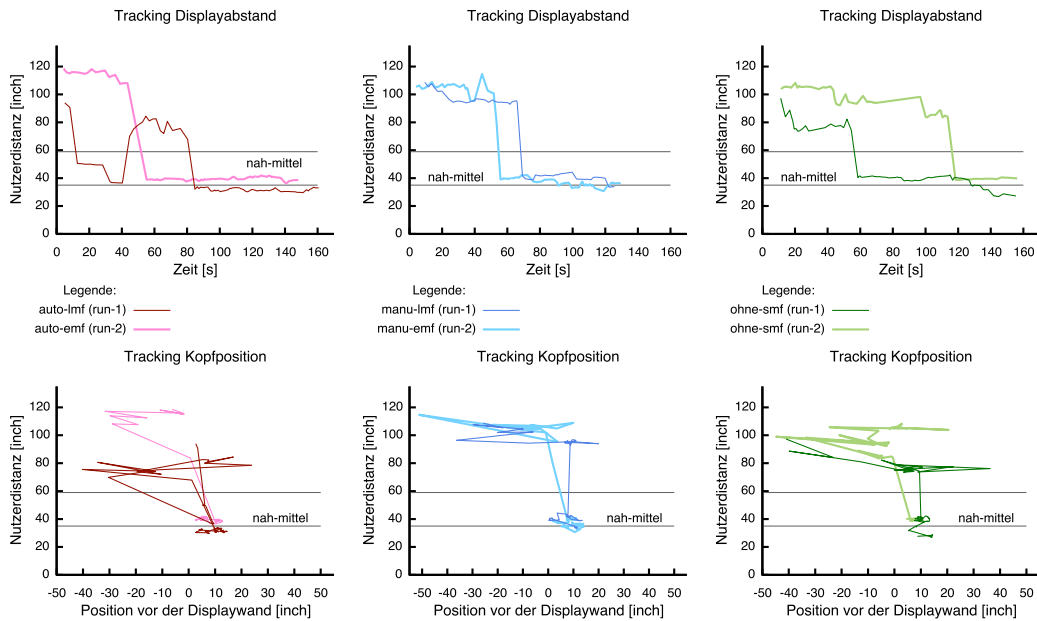


physische Navigation tp08 (it2)

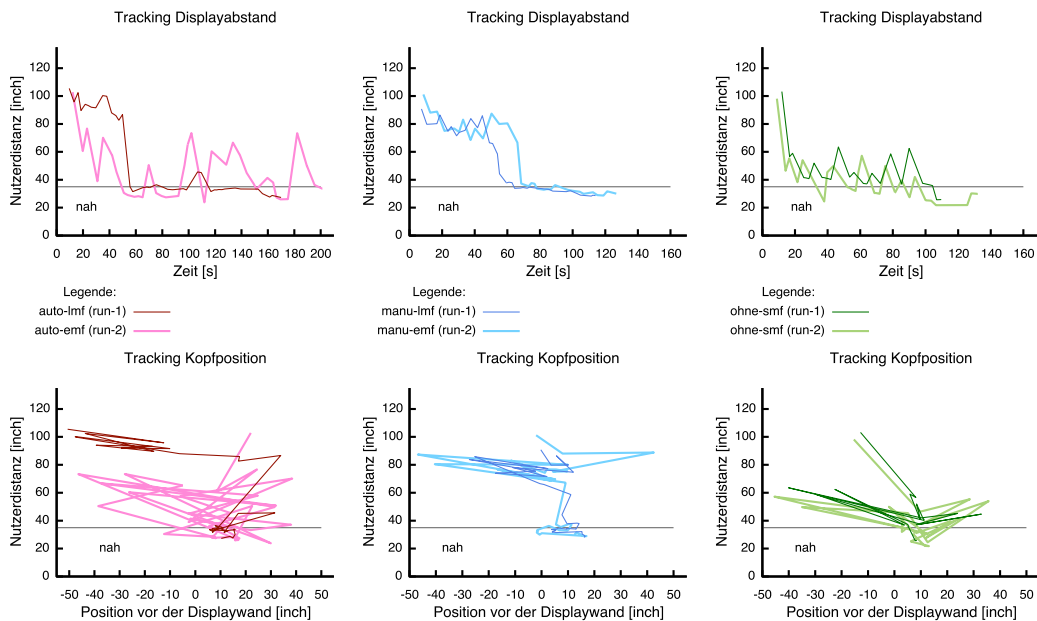


Der Proband manipulierte die Objekte grob aus der Entfernung mit anschließender Feinjustierung im Nahbereich (Mobilität Taktik, bevorzugte Distanz nah-mittel und nah (it1) und nah (it2)):

physische Navigation tp13 (it1)



physische Navigation tp13 (it2)



B. Anhang

Zur Ergänzung des Kapitels 2.2.1 Statistische Methoden werden an dieser Stelle die verwendeten Kenngrößen definiert, welche in den beschriebenen Nutzerstudien verwendet wurden (siehe Kapitel 6).

B.1. Kennwerte der deskriptiven Statistik

Für die Darstellung der **Häufigkeitsverteilung** durch Kennwerte wird eine kategoriale Variable festgelegt und die Anzahl der einzelnen Merkmalsausprägungen bestimmt. Ein natürliches Kategoriensystem bilden nominalskalierte Variablen. Bei intervallskalierten quantitativen Merkmalen muss ein Kategoriensystem entwickelt werden, indem die Anzahl der Klassen (oder Kategorien) und ihre Breite festgelegt wird.

Anzahl der Klassen: Aufgrund der Übersichtlichkeit sollten maximal 20 Klassen verwendet werden. Die Klassenanzahl (Km) kann nach einer der folgenden Regeln bestimmt werden, mit der Anzahl der Messwerte n und dem Interquartilabstand der Messwerte.

$$\begin{aligned} Km &\approx \sqrt{n} \\ Km &\approx 1 + 3,32 * \log(n) \\ Km &= \frac{2 * (P_{75} - P_{25})}{\sqrt[3]{n}} \end{aligned}$$

Breite der Klassen: Es wird der kleinste und größte Messwert bestimmt; man erhält somit das Intervall (Variationsbreite) aller Messwerte und unterteilt dieses in gleichgroße Teilintervalle.

Unter Umständen müssen die Anzahl und die Breite der Klassen anschließend händisch angepasst werden, um ein praktikables Kategoriensystem für die Messwerte zu erhalten, z.B. indem Ausreißer in Randbereichen zusammengefasst werden.

Die gebräuchlichsten **Maße der zentralen Tendenz** sind: der Modalwert, der Medianwert (Median) und das arithmetische Mittel.

Der Modalwert (Modus, Mo) ist der Wert einer Verteilung, der am häufigsten vorkommt bzw. der Wert in der grafischen Darstellung, bei dem die Verteilung ihr Maximum hat.

Der Median (Md) ist der Wert, der eine Verteilung halbiert. Die Differenz zwischen Median und den einzelnen Werten ist im Durchschnitt minimal.

Das arithmetische Mittel (Mittelwert, \bar{x}) gibt den Durchschnittswert einer Verteilung an. Beim arithmetischen Mittel fallen "Ausreißer" in den Daten stärker ins Gewicht als beim Modalwert oder Median. Das arithmetische Mittel wird als Summe aller Werte x_i dividiert durch die Anzahl der Werte n berechnet.

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$$

Aus der Position des arithmetischen Mittels, des Modalwertes und des Medians in einer Verteilung ist ersichtlich, ob eine Verteilung rechtssteil ($\bar{x} < Md < Mo$), linkssteil ($Mo < Md < \bar{x}$) oder symmetrisch ($\bar{x} = Mo = Md$) ist. Eine Verteilung kann über die Eigenschaften Schiefe (rechtssteil, linkssteil, symmetrisch; *Skewness*) und Exzess (breitgipflig, schmalgipflig; *Kurtosis*) beschrieben werden.

Die **Kennwerte der Dispersion** beschreiben, wie stark die einzelnen Werte in einer Verteilung vom Mittelwert abweichen. Es werden üblicherweise folgende Kennwerte verwendet: Variationsbreite, mittlere Abweichung, Varianz und Standardabweichung.

Die Variationsbreite (*range*) beschreibt den Bereich, in dem sich die Messwerte befinden und wird aus der Differenz des größten und kleinsten Wertes gebildet. Da sie aus den Extremwerten gebildet wird, fallen einzelne "Ausreißer" in der Verteilung stark ins Gewicht. Daher wird oft ein eingeschränkter Streubereich betrachtet, z.B. nur die mittleren 90% aller Werte. Dabei werden die unteren 5% (das 5. Perzentil) und die oberen 5% (das 95. Perzentil) der Verteilung abgeschnitten und damit auch die Ausreißer. Manchmal wird auch nur der Interquartilbereich genommen, d.h. der Streubereich der mittleren 50% der Verteilung (begrenzt durch 25. Perzentil und 75. Perzentil). Der mittlere Quartilabstand ist definiert als $(P_{75} - P_{25})/2$.

Die mittlere Abweichung (*average deviation, AD*) gibt den Durchschnitt der absoluten Abweichung jedes einzelnen Messwertes x_i vom arithmetischen Mittel \bar{x} aller Messwerte an.

$$AD = \frac{\sum_{i=1}^n (|x_i - \bar{x}|)}{n}$$

Die Varianz (s^2) ist als Durchschnitt der quadrierten Abweichungen der einzelnen Messwerte x_i vom arithmetischen Mittel \bar{x} definiert. Bei der Varianz werden alle Messwerte einzeln berücksichtigt, was eine gute Beschreibung der gesamten Variabilität aller Messwerte ermöglicht. Jedoch fallen aufgrund der

Quadrierung größere Abweichungen stärker ins Gewicht.

$$s^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}$$

Die Standardabweichung (Streuung, s) zeigt an wie stark die Messwerte um den Mittelwert streuen. Sie ist als positive Wurzel aus der Varianz definiert und hat dieselbe Maßeinheit wie die ursprünglichen Messwerte.

$$s = \sqrt{s^2} = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}}$$

Um die Vergleichbarkeit aus verschiedenen Merkmalsverteilungen herzustellen, müssen die einzelnen Messwerte x_i an dem Mittelwert \bar{x} und der Streuung s ihres Kollektives relativiert werden.

Die z-Transformation standardisiert die einzelnen Messwerte und normiert die Verteilung. Eine z-transformierte Verteilung hat immer einen Mittelwert von 0 und eine Streuung von 1. Die transformierten Werte werden als z-Werte bezeichnet.

$$z_i = \frac{x_i - \bar{x}}{s}$$

B.2. Kennwerte der Inferenzstatistik

Anhand der Kennwertverteilung der Stichprobe wird bestimmt, wie genau der Stichprobenwert den Populationswert wiedergibt. Die Streuung der Kennwertverteilung bestimmt wie gut ein einzelner Stichprobenwert (z.B. Stichprobenmittelwert \bar{x}) den unbekanntem Populationswert (z.B. Populationsmittelwert μ) schätzt. Diese Streuung der Kennwertverteilung wird als Standardfehler bezeichnet.

Der Standardfehler der Mittelwerte ($\sigma_{\bar{x}}$) berechnet sich aus der Varianz des Merkmals in der Population (σ^2) und der Stichprobengröße n . Die Populationsvarianz wird durch die Stichprobenvarianz (s^2) geschätzt, wobei die durchschnittliche Stichprobenvarianz die Populationsvarianz um den Faktor $(n-1)/n$ unterschätzt. Korrigiert man die Varianzschätzung erhält man den geschätzten Standardfehler der Mittelwerte ($\hat{\sigma}_{\bar{x}}$). Der Standardfehler wird zur Konstruktion von Konfidenzintervallen genutzt.

$$\sigma_{\bar{x}} = \sqrt{\frac{\sigma^2}{n}}$$

$$\hat{\sigma}_{\bar{x}} = \sqrt{\frac{s^2}{n-1}}$$

Das Konfidenzintervall (Vertrauensintervall) gibt den Bereich eines Merkmals an, in dem sich mit einer bestimmten Wahrscheinlichkeit (meist 95% oder 99%) alle möglichen Populationsparameter befinden, die z.B. den berechneten Stichprobenmittelwert erzeugt haben können. Die Grenzen eines x%-Konfidenzintervalles werden nach der Formel $\bar{x} \pm z_{x\%} * \hat{\sigma}_{\bar{x}}$ berechnet. Die z-Werte der Standardnormalverteilung können aus den Tabellen in Statistik-Lehrbüchern entnommen werden (z.B. [Bor99b]).

Hypothesen sind unterscheidbar nach Unterschiedshypothesen (behaupten Unterschiede zwischen Gruppen) und Zusammenhangshypothesen (treffen Aussagen über Korrelationen zwischen Variablen). Des Weiteren lassen sich Hypothesen als ungerichtet oder gerichtet und unspezifisch oder spezifisch charakterisieren. Die ungerichtete Hypothese behauptet (irgend)einen Unterschied bzw. Zusammenhang, während bei der gerichteten Hypothese auch die Richtung des Unterschiedes angegeben wird. Bei spezifischen Hypothesen wird die Größe des Unterschiedes bzw. Zusammenhanges angegeben, während bei der unspezifischen Hypothese keine Aussage zur Größe getroffen wird. Die Art der Hypothesenformulierung spielt bei der Signifikanzprüfung eine wichtige Rolle.

Die Prüfung der Hypothese erfolgt durch den Vergleich zweier einander ausschließender Hypothesen, d.h. der Alternativhypothese (H_1) und der Nullhypothese (H_0). Die Alternativhypothese beschreibt den vermuteten Unterschied zwischen den Gruppen beziehungsweise die vermutete Korrelation zwischen Variablen. Die Nullhypothese behauptet das Gegenteil, d.h. es gibt keinen Unterschied zwischen den Gruppen beziehungsweise keine Korrelation zwischen den Variablen. Die formulierten statistischen Hypothesen H_1 und H_0 geben Aussagen über den verwendeten statistischen Kennwert und dessen Ausprägung in der Population. Beispielsweise sind die folgenden drei statistischen Hypothesen für den Vergleich der Mittelwerte zweier Gruppen A und B denkbar:

	H_1	H_0
ungerichtet-unspezifisch:	$\mu_A \neq \mu_B$	$\mu_A = \mu_B$
gerichtet-unspezifisch:	$\mu_A > \mu_B$	$\mu_A \leq \mu_B$
gerichtet-spezifisch:	$\mu_A > \mu_B + x$	$\mu_A \leq \mu_B + x$

Die Prüfung der Hypothesen, d.h. die Entscheidung zugunsten der Alternativhypothese oder Nullhypothese ist mit einer bestimmten Wahrscheinlichkeit falsch, da die Hypothesen die Verhältnisse in der Population beschreiben, während die Untersuchungsergebnisse sich auf die Stichprobe beziehen. Somit kann eine an sich richtige Nullhypothese zugunsten der Alternativhypothese verworfen werden (α -Fehler) oder die Nullhypothese wird akzeptiert, obwohl die Alternativhypothese

richtig ist (β -Fehler). Die Wahrscheinlichkeiten der beiden Fehler verhalten sich gegenläufig und sind abhängig von der Stichprobengröße und der Effektgröße. Mit der Effektgröße wird festgelegt, wie stark der H_1 -Parameter der Stichprobe vom Populationskennwert abweichen muss, um als praktisch bedeutsamer Effekt zu gelten.

Bei dem Signifikanztest wird auf die Nullhypothese getestet. Daher muss bei akzeptierter Alternativhypothese die α -Fehlerwahrscheinlichkeit (Irrtumswahrscheinlichkeit) angegeben werden, die in Kauf genommen wird, wenn aufgrund der Ergebnisse die Nullhypothese verworfen wird. Für die Vergleichbarkeit und Qualität statistischer Untersuchungen wird eine Nullhypothese erst verworfen, wenn die Irrtumswahrscheinlichkeit $\leq 5\%$ (oder $\leq 1\%$) ist. Diese Schwellwerte werden als **Signifikanzniveau** (α -Niveau, α -Level) bezeichnet.

Bei der Überprüfung einer gerichteten Hypothese wird ein einseitiger Test durchgeführt, während bei der ungerichteten Hypothese ein zweiseitiger Test durchgeführt wird.

Statistische Verfahren zur Durchführung der Signifikanzprüfung

An dieser Stelle wird kurz das Vorgehen der Signifikanzprüfung mit dem t-Test und der Varianzanalyse erläutert. Der t-Test und die Varianzanalyse sind parametrische Verfahren, die eine Prüfung bestimmter Voraussetzungen erfordern, wie normalverteilte Grundgesamtheit, Varianzhomogenität und Sphärizität. Mit den folgenden Tests können die Voraussetzungen geprüft werden:

- normalverteilte Grundgesamtheit: deskriptive Maße (Schiefe und Exzess), Kolmogoroff-Smirnov-Anpassungstest, Shapiro-Wilk-Test
- Varianzhomogenität: F-Test, Levene-Test
- Sphärizität: Mauchly-Test

Der t-Test untersucht, ob sich zwei empirisch erhobene Mittelwerte systematisch voneinander unterscheiden. Der **t-Test für unabhängige Stichproben** betrachtet die Mittelwerte aus beiden Gruppen unabhängig voneinander.

Beispielsweise soll eine ungerichtete Hypothese mit dem t-Test überprüft werden. Die Nullhypothese besagt, dass die Populationswerte beider Gruppen identisch sind und der Unterschied zufällig zustande gekommen ist ($H_0 : \mu_A = \mu_B$). Die zugehörige Alternativhypothese postuliert, dass die Stichproben aus unterschiedlichen Populationen stammen und der Unterschied der Mittelwerte (\bar{x}_A, \bar{x}_B) systematisch ist ($H_1 : \mu_A \neq \mu_B$).

Als Prüfgröße t wird die Differenz der Mittelwerte \bar{x}_A und \bar{x}_B genommen und am geschätzten Standardfehler dieser Differenzen relativiert

$$t = \frac{\bar{x}_A - \bar{x}_B}{\hat{\sigma}_{(\bar{x}_A - \bar{x}_B)}}$$

mit dem geschätzten Standardfehler der Differenzen zweier Mittelwerte.

$$\hat{\sigma}_{(\bar{x}_A - \bar{x}_B)} = \sqrt{\frac{\sigma_A^2}{n_A} + \frac{\sigma_B^2}{n_B}}$$

Bei kleinen Stichprobenumfängen ($n_A + n_B < 50$) folgt die Prüfgröße einer t-Verteilung (*student's-t distribution*). Die Form der t-Verteilung hängt von den Freiheitsgraden (df) der Untersuchung ab, die aus den Größen der Teilstichproben berechnet werden: $df = (n_A - 1) + (n_B - 1) = n_A + n_B - 2$. Für große Stichproben ($n_A + n_B \geq 50$) geht die t-Verteilung in die Normalverteilung über.

Liegt ein Unterschied bezüglich der Mittelwerte vor, dann ist der empirisch ermittelte t-Wert positiv oder negativ. Die Signifikanzprüfung testet, ob der empirische t-Wert größer als der kritische t-Wert¹ entsprechend der berechneten Freiheitsgrade ist, damit er bei einem gegebenen α -Niveau und einseitiger bzw. zweiseitiger Fragestellung signifikant von Null abweicht ($|t_{\text{prüf}}| > t_{(df, \alpha)_{\text{krit}}}$).

Die Voraussetzungen für den t-Test für unabhängige Stichproben sind: Die abhängige Variable muss mindestens intervallskaliert sein, die Stichproben müssen normalverteilt sein, es muss Varianzhomogenität bestehen und es müssen unabhängige Stichproben vorliegen. Der t-Test ist relativ robust; es kommt jedoch zu Fehlentscheidungen wenn die Varianzen heterogen und die Stichprobengrößen sehr unterschiedlich sind. Dann sollte der Mann-Whitney-U-Test verwendet werden.

Beim **t-Test für abhängige Stichproben** werden die Differenzwerte der zueinander gehörigen Messwertpaare betrachtet. In diesem Fall ist die Stichprobengröße n die Zahl der Messwertpaare, was Auswirkungen auf die Schätzung des Standardfehlers und die Freiheitsgrade des t-Tests hat. Es wird für jedes Messwertpaar i die Differenz d_i gebildet und dann das arithmetische Mittel aller d_i -Werte berechnet, wobei n die Anzahl der Messwertpaare angibt.

$$\bar{x}_d = \frac{\sum_{i=1}^n d_i}{n}$$

Die Prüfgröße t berechnet sich aus \bar{x}_d und dem Standardfehler der Verteilung der Mittelwerte $\hat{\sigma}_{\bar{x}_d}$.

$$t = \frac{\bar{x}_d}{\hat{\sigma}_{\bar{x}_d}}$$

¹ Die kritischen t-Werte und F-Werte können aus statistischen Tabellen mit den angegebenen Freiheitsgraden und dem Signifikanzniveau entnommen werden (z.B. [Bor99b]).

Die Streuung der Differenzen in der Population (σ_d) wird aufgrund der Stichprobendifferenzen geschätzt.

$$\hat{\sigma}_{\bar{x}_d} = \frac{\hat{\sigma}_d}{\sqrt{n}}$$

$$\hat{\sigma}_d = \sqrt{\frac{\sum_{i=1}^n (d_i - \bar{x}_d)^2}{n-1}}$$

Das Ergebnis ist signifikant, wenn der beobachtete t-Wert größer ist als der kritische t-Wert mit gegebenem Signifikanzniveau und den Freiheitsgraden $df = n - 1$.

Die Voraussetzungen des t-Test für abhängige Stichproben sind: Mindestens intervallskalierte Daten der abhängigen Variable, Normalverteilung der Differenzen in den Stichproben und es müssen abhängige Stichproben vorliegen. Der t-Test für abhängige Stichproben ist ebenfalls relativ robust, jedoch sollte der Wilcoxon-Test verwendet werden, wenn die Voraussetzungen nicht erfüllt sind.

Die Varianzanalyse wird angewendet, um Unterschiede in drei oder mehr Bedingungen herauszufinden. Die Varianzanalyse basiert auf der F-Verteilung und wird auch als F-Test bezeichnet. Die **einfaktorielle Varianzanalyse** (*one-way ANOVA*) überprüft die Auswirkung von einer unabhängigen Variable mit mehr als zwei Bedingungen auf die abhängige Variable.

Beispielsweise sollen die Ergebnisse von drei Methoden verglichen werden. Die Nullhypothese H_0 behauptet, dass sich die Ergebnisse der drei Methoden nicht unterscheiden bzw. die Mittelwertparameter μ_i der Populationen identisch sind ($H_0 : \mu_1 = \mu_2 = \mu_3$). Die Alternativhypothese H_1 ist ungerichtet und besagt nur, dass sich mindestens zwei beliebige Methoden unterscheiden ($H_1 : \mu_i \neq \mu_{i'}$).

Das allgemeine Vorgehen bei der Varianzanalyse gliedert sich in vier Schritte: Bestimmung der Gesamtvarianz; Bestimmung der Varianz, die auf die unabhängige Variable zurückzuführen ist (Treatmentvarianz); Bestimmung der Varianz, die sich durch sonstige Fehlereinflüsse ergibt (Fehlervarianz); Vergleich der Varianzen (Signifikanzprüfung).

Für den Vergleich der Methoden werden die Mittelwerte für k Gruppen ($\bar{x}_1, \dots, \bar{x}_k$) und der Gesamtmittelwert (\bar{x}) berechnet (mit der Stichprobengröße $n = \sum n_i$).

$$\bar{x}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} \bar{x}_{ij}$$

$$\bar{x} = \frac{1}{n} \sum_{i=1}^k n_i * \bar{x}_i$$

Die Gesamtvarianz ergibt sich aus der Quadratsumme (QS_{gesamt}) dividiert durch die Freiheitsgrade der Varianz (df_{gesamt}). Die Quadratsumme (*sum of squares*, SS , QS) ist die Summe der quadrierten Abweichungen aller Messwerte einer

Verteilung vom Mittelwert. Die Freiheitsgrade (*degree of freedom, df*) geben die Anzahl der frei variierbaren Werte eines Kennwertes an. Die Berechnung der Gesamtvarianz ($\hat{\sigma}^2$, *mean square, MS*) ist identisch mit der Berechnung der Populationsvarianz.

$$\hat{\sigma}^2 = \frac{QS_{gesamt}}{df_{gesamt}}$$

Die Varianzen der Quadratsumme (QS_{gesamt}) und Freiheitsgrade (df_{gesamt}) werden in einen Treatmentanteil und Fehleranteil aufgespalten.

$$\begin{aligned} QS_{gesamt} &= QS_{treat} + QS_{fehler} \\ df_{gesamt} &= df_{treat} + df_{fehler} \end{aligned}$$

Treatmentvarianz bestimmt den Anteil der Unterschiedlichkeit aller Messwerte, der auf die Bedingungen zurückzuführen ist (*variances between groups*).

$$\begin{aligned} QS_{treat} &= \sum_{i=1}^k (x_i - \bar{x})^2 \\ df_{treat} &= k - 1 \\ \hat{\sigma}_{treat}^2 &= \frac{QS_{treat}}{df_{treat}} \end{aligned}$$

Fehlervarianz enthält die Messwertunterschiede, die nicht auf das Treatment zurückzuführen sind (*variances within groups*). Der Fehlervarianzanteil bestimmt das Ausmaß der Störvariablen.

$$\begin{aligned} QS_{fehler} &= \sum_{i=1}^k \sum_{j=1}^{n_i} (x_{ij} - \bar{x}_i)^2 \\ df_{fehler} &= n - k \\ \hat{\sigma}_{fehler}^2 &= \frac{QS_{fehler}}{df_{fehler}} \end{aligned}$$

Signifikanzprüfung testet, ob die Treatmentvarianz deutlich größer ist als die Fehlervarianz, dann wird die Nullhypothese zugunsten der Alternativhypothese verworfen.

$$F = \frac{\hat{\sigma}_{treat}^2}{\hat{\sigma}_{fehler}^2}$$

Das heißt, ist der empirische F-Wert größer als der kritische F-Wert¹ ($F(df_{treat}, df_{fehler}; \alpha)$), dann kann die Nullhypothese auf dem genannten α -Niveau verworfen werden.

Die einfaktorielle Varianzanalyse liefert nur Hinweise, ob sich Mittelwerte unterscheiden, aber nicht welche Mittelwerte sich unterscheiden. Der Einzelvergleich kann mit Post-Hoc-Tests, beispielsweise dem Tukey-Test oder Student-Newman-

Keuls-Test durchgeführt werden. Der Student-Newman-Keuls-Test (SNK) ist nicht so konservativ wie der Tukey-Test, d.h. es ist wahrscheinlicher einen gegebenen Unterschied, der statistisch signifikant ist, zu bestimmen [Sys08]. Die Voraussetzungen der einfaktoriellen Varianzanalyse sind normalverteilte Fehlerkomponenten, homogene Fehlervarianzen und die Unabhängigkeit der Fehlerkomponenten zwischen den Stichproben.

Bei der **einfaktoriellen Varianzanalyse mit Messwiederholung** wird die Quadratsumme QS_{gesamt} aus dem Anteil, der die Unterschiedlichkeit zwischen den Probanden beschreibt (QS_{zw}), und einem weiteren Anteil, der die Veränderungen innerhalb der Werte der einzelnen Probanden charakterisiert (QS_{in}), gebildet. Zusätzlich lässt sich QS_{in} in den Treatmenteffekt (QS_{treat}) und die Residualquadratsumme (QS_{res}), die den Interaktionseffekt (Proband x Treatment) und den Fehlereffekt enthält, zerlegen.

$$\begin{aligned} QS_{gesamt} &= QS_{in} + QS_{zw} \\ &= QS_{treat} + QS_{res} + QS_{zw} \end{aligned}$$

Die einfaktorielle Varianzanalyse mit Messwiederholungen setzt ebenfalls normalverteilte und homogene Fehlerkomponenten voraus, sowie homogene Korrelationen zwischen den Bedingungen.

Die **mehrfaktorielle Varianzanalyse** untersucht, wie sich mehrere unabhängige Variablen gleichzeitig auf eine abhängige Variable auswirken (z.B. zweifaktorielle Varianzanalyse). Es werden die Wechselwirkungen (oder Interaktion) von unabhängigen Variablen untersucht. Mit der **zweifaktoriellen Varianzanalyse** (*two-way ANOVA*) können drei Hypothesen gleichzeitig geprüft werden, die Haupteffekte der beiden Faktoren und ihre Wechselwirkung. Sie prüft folgende der Nullhypothesen:

1. Haupteffekt A: Die unter den k-Stufen des Faktor A untersuchten Personen unterscheiden sich nicht ($H_{0-A} : \mu_{A1} = \mu_{A2} = \dots = \mu_{Ak}$)
2. Haupteffekt B: Die unter den l-Stufen des Faktor B untersuchten Personen unterscheiden sich nicht ($H_{0-B} : \mu_{B1} = \mu_{B2} = \dots = \mu_{Bl}$)
3. Wechselwirkung AxB: Zwischen den Faktoren A und B besteht keine Interaktion, die Faktoren sind additiv ($H_{0-AxB} : \mu_{AiBj} = \mu_{Ai} + \mu_{Bj} - \mu_{gesamt}$)

Die Alternativhypothesen besagen, dass sich mindestens zwei Mittelwerte eines Faktors unterscheiden (z.B. $H_{1-A} : \mu_{Ai} \neq \mu_{Aj}$) bzw. dass eine Wechselwirkung vorliegt.

Bei der zweifaktoriellen Varianzanalyse entspricht das allgemeine Vorgehen der einfaktoriellen Analyse, jedoch erfolgt die Quadratsummenzerlegung in je einen

Anteil für die Haupteffekte, die Wechselwirkung und die Fehlervarianz.

$$QS_{gesamt} = QS_A + QS_B + QS_{A \times B} + QS_{fehler}$$

Bei der Prüfung der Wechselwirkung wird untersucht, wie die Faktoren zusammenwirken. Liegt eine signifikante Wechselwirkung vor, so muss dies bei der Interpretation der Haupteffekte berücksichtigt werden. Mit Hilfe von Interaktionsdiagrammen kann das Zusammenwirken der Faktoren sichtbar gemacht werden. Wenn die Linienzüge im Diagramm gleichsinnig verlaufen, dann kann der Haupteffekt über die Stufen des anderen Faktors generalisiert werden.

Die Voraussetzungen für die einfaktorielle Varianzanalyse gelten ebenso für die mehrfaktorielle Analyse. Bei Verletzung der Voraussetzung sollte die Kruskal-Wallis Varianzanalyse an Rängen bzw. die Friedman Varianzanalyse an Rängen mit wiederholten Messungen verwendet werden. Die Friedman Teststatistik χ^2 evaluiert die Nullhypothese, dass alle Rangsummen gleich sind. Ist der Wert von χ^2 groß, dann kann geschlossen werden, dass die Unterschiede in den Rangsummen größer sind als es zufallsbedingt zu erwarten ist. Sind die χ^2 -Werte nahe Null, dann gibt es keinen signifikanten Unterschied, d.h. die Ränge innerhalb jedes Probanden sind zufällig [Sys08].

Glossar

3D Interaktion: Die 3D Interaktion bezeichnet die räumliche Nutzereingabe durch physische Navigation vor einem großen Display.

Boxplot: Die verwendeten Boxplots zeigen für einen Datensatz als Whisker das 10. und 90. Perzentil, den Median, sowie das untere Quartil (das 25. Perzentil) und das obere Quartil (das 75. Perzentil) als Box. Zusätzlich werden die Ausreißer (als Kreuze) und der Mittelwert (als gestrichelte Linie) dargestellt.

Cursorberechnung: Bei der absoluten Cursorberechnung wird ein absolutes Mapping verwendet (1:1 Mapping). Bei der relativen Cursorberechnung wird ein relatives Mapping mit einem distanzabhängigen Mappingfaktor verwendet. Die hybride Cursorberechnung verwendet ein taskabhängiges Mapping für die 2D Manipulationstechniken, d.h. für die Selektion eine absolute Cursorberechnung und für die anderen Interaktionsaufgaben eine relative Cursorberechnung.

Cursorwechsel: Für das Umschalten zwischen den Präzisionsstufen werden eine automatische Anpassung und eine manuelle Anpassung untersucht. Das manuelle Umschalten zwischen absolutem und relativem Mapping erfolgt per Tastendruck durch den Nutzer (*manual-switching*). Das automatische Umschalten (*distance-aware*) erfolgt distanzabhängig (relative Cursorberechnung) und/oder taskabhängig (hybride Cursorberechnung).

GHD-Umgebung: Eine große hochauflösende Displayumgebung besteht aus Displays mit einer großen Anzeigefläche und hoher Pixeldichte, die unter Umständen die visuelle Sehschärfe des Nutzers übersteigen können.

Head-Tracking: Head-Tracking bezeichnet sowohl eine Eingabemodalität als auch eine Interaktionstechnik für die Navigation.

Interaction Scaling: Interaction Scaling definiert die distanzabhängige Präzision der Nutzerinteraktion und die distanzabhängige Informationsdarstellung, basierend auf der physischen Navigation des Nutzers in GHD-Umgebungen.

Interaktionsaufgabe: Die Interaktionsaufgabe (Interaktionstask) beschreibt welche Aufgabe der Nutzer durchführt, indem er mit dem System oder der Anwendung interagiert. Grundlegende Interaktionsaufgaben sind Manipulation (Selektion, Positionierung, Rotation, Skalierung), Navigation (Bewegung, Routenplanung), Systemkontrolle, Symboleingabe und Modellierung.

Interaktionstechnik: Bei der Interaktion in virtuellen Umgebungen werden meist mehrere oder komplexe Interaktionsaufgaben benötigt, die durch Interaktionstechniken umgesetzt werden.

Mappingfaktor: Der Mappingfaktor bildet die Bewegungen der Nutzereingabe (z.B. Eingabegerät, Kopfbewegung) verhältnismäßig auf Bewegungen oder Änderungen der virtuellen Ausgabe (z.B. Cursor, Schriftgröße) ab.

Physische Navigation: Die physische Navigation repräsentiert ein natürliches Verhalten von Nutzern, indem sie näher an die Displaywand heran gehen, um Details wahrnehmen zu können, und sich von der Displaywand fortbewegen, um einen Überblick über die gesamte visuelle Darstellung zu erhalten. Die physische Navigation (Bewegung von Augen, Kopf, Körper) ist eine direkte und natürliche Interaktion und bedeutet weniger kognitive Belastung für den Nutzer.

Projektion: Bei der perspektivischen Projektion wird, ausgehend von der Position/Orientierung des Eingabegerätes in der physischen Umgebung (6-DOF), ein imaginärer Strahl berechnet, der die Displayfläche schneidet und somit die virtuellen Positionsdaten an dieser Stelle projiziert (2-DOF). Bei der orthogonalen Projektion werden nur die Positionsdaten des Eingabegerätes in der physischen Umgebung (3-DOF) im rechten Winkel auf die Displayfläche projiziert und somit die zugehörigen virtuellen Positionsdaten (2-DOF) bestimmt.

Ray-Casting: Ray-Casting ist eine Interaktionstechnik, bei der ein imaginärer Strahl von dem Eingabegerät, dem Finger oder der Hand des Nutzers projiziert wird und wo der Strahl auf ein Objekt in der virtuellen Umgebung trifft, kann mit diesem interagiert werden.

SigmaPlot: SigmaPlot ist eine kommerzielle Softwareanwendung zur Statistik- und Datenanalyse, welche für die Prüfung auf Normalverteilung den Shapiro-Wilk-Test, für die Varianzhomogenität den Levene-Test und den Mauchly-Test für die Sphärizität verwendet.

Smart Meeting Room: Smart Meeting Room ist eine Multi-Display-Umgebung, in der heterogene Displays und Eingabegeräte zur Informationsdarstellung und Nutzerinteraktion verwendet werden. Ziel dieser intelligenten Umgebung ist es, die Benutzer bei ihrer Tätigkeit zu unterstützen, indem die Informationsdarstellung und die Interaktionsmöglichkeiten an die Betrachtungsposition und die Rolle des Nutzers im Raum angepasst werden.

Abkürzungs- und Symbolverzeichnis

Liste der verwendeten Abkürzungen:

dH	– dominante Hand
DOF	– degrees-of-freedom (Freiheitsgrade)
dpi	– dots per inch (Punkte pro Zoll; synonym für ppi)
GHD	– großes hochauflösendes Display
GHD-Umgebung	– große hochauflösende Displayumgebung
ft.	– feet (Fuß); 1 ft. = 12 in. = 30,48 cm
high-res	– high-resolution (hochauflösend)
in.	– inch (Zoll); 1 in. = 2,54 cm
IR	– infra-red (infrarot)
IS	– Interaction Scaling
LOD	– level-of-detail (Detaillierungsgrad)
low-res	– low-resolution (niedrigauflösend)
lp/mm	– Linienpaare pro Millimeter
MD-Umgebung	– Multi-Display-Umgebung
MF	– Mappingfaktor
ndH	– nicht-dominante Hand
n.s.	– nicht signifikant
Obj.	– Objekt
pp	– pixel pitch (Pixelabstand)
pt	– point (Maßeinheit der Schriftgröße)
px	– Pixel
px/cm	– Pixel pro Zentimeter
ppi	– pixel per inch (Pixel pro Zoll; px/in)
RGB	– Rot, Grün, Blau
SMR	– Smart Meeting Room
VR	– Virtual Reality (virtuelle Realität)
Wiimote	– Nintendo Wii Remote Controller

Liste der verwendeten Symbole:

$rPos$	– reale Koordinaten im physischen Interaktionsraum, 6-DOF
$pPos$	– physische Koordinaten in der virtuellen Umgebung, 3-DOF
$vPos$	– virtuelle Koordinaten in der virtuellen Umgebung, 2-DOF
mf	– Mappingfaktor
$minD, maxD$	– minimale/maximale Nutzerdistanz im Trackingbereich
sD	– Schwelldistanz für Mappingfunktion
udd	– (aktuelle) Nutzerdistanz zum Display
C_{MP}, C_{dH}	– Mittelpunkt-Cursor, Cursor der dominanten Hand
$pP_{i_{dH}}$	– physische Koordinate der dominanten Hand zum Zeitpunkt i ($i \in \mathbb{N}$)
V_i	– Abstandsvektor zw. dominanter und nicht-dominanter Hand zum Zeitpunkt i
φ_i	– Winkel zw. dominanter und nicht-dominanter Hand zum Zeitpunkt i
AD	– mittlere Abweichung
α	– Signifikanzniveau
$\chi^2(n)$	– Chi-Quadrat Variable mit n Freiheitsgraden
df	– Freiheitsgrade eines statistischen Tests (auch DF)
$F(n_1, n_2)$	– Wert der F-Verteilung mit n_1 Zähler- und n_2 Nennerfreiheitsgraden
H_0	– Nullhypothese im Signifikanztest
H_1	– Alternativhypothese im Signifikanztest
Md	– Medianwert
μ	– Mittelwert einer theoretischen Verteilung oder Population
MS	– Gesamtvarianz bei der Varianzberechnung
N	– Stichprobenumfang
p	– Irrtumswahrscheinlichkeit (auch P)
$q(r, n)$	– Wert der q-Verteilung mit r Stufen und n Freiheitsgraden
SS	– Quadratsumme bei der Varianzberechnung (auch QS)
s	– Streuung (Standardabweichung) für eine Stichprobe
s^2	– Varianz für eine Stichprobe
σ	– Streuung einer theoretischen Verteilung oder Population
σ^2	– Varianz einer theoretischen Verteilung oder Population
$t(n)$	– Wert der t-Verteilung mit n Freiheitsgraden (auch t_n)
\bar{x}	– arithmetisches Mittel einer Stichprobe

Verzeichnis der URLs

http://phys.org/news200924198.html	12
http://www.techfak.uni-bielefeld.de/~skopp/Lehre/MMI_WS1112/MMI_8.pdf	27
http://www.ascension-tech.com	43
http://www.nintendo.com/wii/	89
http://www.rocksclusters.org	116
http://www.naturalpoint.com/optitrack/products/tracking-tools/ . . .	116
http://www.cs.unc.edu/Research/vrpn/	116
http://www.bluez.org/	116
http://vis.ucsd.edu/~cglx/	117
http://idav.ucdavis.edu/~okreylos/ResDev/Vrui/index.html	117
http://sourceforge.net/projects/ftgl/	122

Letzter Zugriff der URLs erfolgte am 02.06. 2015

Abbildungsverzeichnis

Abb. 1.2.1:	Die vier Komponenten des Parameterraumes mit beispielhaften Einflussfaktoren.	16
Abb. 2.1.1:	Visualisierung der Berechnung des Sehwinkels eines Objektes.	21
Abb. 2.3.2:	Klassifizierung von Displays nach Displaygröße und Displayauflösung mit Beispielen.	34
Abb. 2.4.3:	Schematische Einordnung ausgewählter Arbeiten zu Interaktion in GHD-Umgebungen.	63
Abb. 3.0.1:	Schematische Darstellung der verschiedenen Interaktionsräume in großen hochauflösenden Displayumgebungen.	70
Abb. 3.1.2:	Beispiele für Displaywände mit unterschiedlicher Displaygröße, -anordnung und -auflösung, den resultierenden Interaktionsräumen und der Nutzeranzahl.	74
Abb. 3.1.3:	Illustration des Drift-Effektes.	83
Abb. 3.1.4:	Illustration der Ray-Casting Technik mit perspektivischer und orthogonaler Projektion.	85
Abb. 3.1.5:	Die Komponenten des Parameterraumes und ihre zugehörigen Einflussfaktoren für die distanzabhängige Interaktion. . .	87
Abb. 3.2.6:	Verwendete Eingabegeräte für die Manipulation und Navigation.	89
Abb. 3.2.7:	Einteilung des physischen Interaktionsraumes für die Mappingfunktionen.	91
Abb. 3.2.8:	Schematische Abbildung des Mappingfaktors abhängig von der Mappingfunktion und der Betrachtungsdistanz.	92
Abb. 3.3.9:	Nutzung des Interaktionsraumes für die Navigationstechniken zur Exploration von Graphen.	97
Abb. 3.3.10:	Aktive Sichtzellen mit unterschiedlichem Detailgrad.	98
Abb. 3.3.11:	Diskussionsrunde im Smart Meeting Room.	101
Abb. 3.3.12:	Kombinierte Mappingfunktionen für die distanzabhängige Interaktion bei absoluter Nutzereingabe und Kindern.	105
Abb. 3.3.13:	Illustration der symmetrischen und asymmetrischen Manipulationstasks.	108
Abb. 4.2.1:	Setup der Displaywand mit Trackingbereich im Altbau und Neubau.	115

Abb. 4.3.2:	Schematische Darstellung der Architektur im VC-Lab.	117
Abb. 5.1.1:	Navigationstechniken für die Exploration des ACM-Graphen.	120
Abb. 5.1.2:	Interaktionsraum für die Exploration des ACM-Graphen.	121
Abb. 5.1.3:	Anpassung der Schriftgröße abhängig von der Nutzerdistanz	122
Abb. 5.1.4:	Verwendung der Linsen-Technik für zwei Benutzer.	122
Abb. 5.1.5:	3D Geländedarstellung mit unterschiedlichen LOD-Levels.	124
Abb. 5.1.6:	Auswahl der Sichtzellen mit unterschiedlichem Detailgrad.	124
Abb. 5.2.7:	Übung zu feinmotorischer Manipulation.	125
Abb. 5.2.8:	Die verwendeten Targetcontainer in den Studien.	127
Abb. 5.2.9:	Spielerische Anwendung für Studie 4.	128
Abb. 5.2.10:	Umsetzung der asymmetrischen und symmetrischen 2D Manipulationstechniken.	129
Abb. 5.2.11:	Cursordarstellung und Tastenbelegung der Interaktionsmodi.	131
Abb. 5.2.12:	Schematische Darstellung des Zusammenspiels der Parameter für die distanzabhängigen Manipulationstechniken.	132
Abb. 6.1.1:	Die verwendeten Mappingfunktionen in Studie 1.	142
Abb. 6.1.2:	Grafische Darstellung der Ergebnisse aus Studie 1.	144
Abb. 6.1.3:	Die verwendeten Mappingfunktionen in Studie 2.	147
Abb. 6.1.4:	Grafische Darstellung der Ergebnisse aus Studie 2.	148
Abb. 6.1.5:	Die verwendeten Mappingfunktionen in Studie 4.	153
Abb. 6.1.6:	Ergebnisse in Studie 4.	155
Abb. 6.2.7:	Die verwendeten Mappingfunktionen in Studie 3.	160
Abb. 6.2.8:	Schematische Darstellung der Zielobjekte im Container mit den zugehörigen Eigenschaften.	162
Abb. 6.2.9:	Durchschnittliche Manipulationszeit und Sortiersuche der Targettypen in Studie 3.	165
Abb. 6.2.10:	Grafische Darstellung der relativen Fehlerrate abhängig von der Bedingung und dem Targettyp für it1 und it2.	166
Abb. 6.2.11:	Subjektive Bewertung der Interaktionstechniken und Varianten des Cursorwechsels im Posttest-Fragebogen.	167
Abb. 6.2.12:	Boxplot der Anzahl der Button-Events in Studie 3.	168
Abb. 6.2.13:	Bewertung der Arbeitsbelastung in Studie 3.	169
Abb. 6.2.14:	Boxplots der Nutzerperformanz in Studie 3.	170
Abb. 6.2.15:	Exemplarische Laufprofile für Kategorie Mobilität.	172
Abb. 6.2.16:	Grafische Darstellung der Nutzerperformanz entsprechend der Kategorien Mobilität und bevorzugter Distanz.	174
Abb. A.1.1:	Tastenbelegung der Wiimote.	195
Abb. A.3.2:	Boxplots der durchschnittlichen Manipulationszeit in Studie 3.	212
Abb. A.3.3:	Interaktionsdiagramm für die Wechselwirkung zwischen Bedingung und Verhalten bei it2 in Studie 3.	225

Tabellenverzeichnis

Tab. 2.2.1:	Übersicht der mathematischen Operationen für die Skalenniveaus.	28
Tab. 2.4.2:	Eignung der Sensortechnologien für Personen- und Geräte-Tracking für präzise Nutzerinteraktion in GHD-Umgebungen.	45
Tab. 2.4.3:	Zerlegung der Manipulations- und Navigationsaufgaben in ihre Task-Parameter.	48
Tab. 3.1.1:	Zusammenfassung der Einflussfaktoren einer Komponente auf Faktoren einer anderen Komponente im Parameterraum. .	72
Tab. 3.2.2:	Parameter der verwendeten Eingabegeräte.	90
Tab. 3.3.3:	Eigenschaften der verwendeten Displayumgebungen und deren Parameter für Interaction Scaling.	94
Tab. 3.3.4:	Task-Parameter für relative Nutzereingabe.	103
Tab. 3.4.5:	Kombinationen der Einflussfaktoren in den Beispielszenarien und die resultierenden Parameter für die Mappingmethode. .	112
Tab. 4.1.1:	Displaykonfiguration der verwendeten Displaywand.	114
Tab. 5.2.1:	Variation der Parameter für die Experimente mit den distanz-abhängigen Manipulationstechniken.	126
Tab. 5.2.2:	Variation der Task-Parameter innerhalb der Studien.	130
Tab. 6.1.1:	Häufigkeit der benötigten Selektionen für jeden Objekttyp und Mappingfunktion im ersten Durchlauf.	150
Tab. 6.2.2:	Häufigkeitsverteilung der Teilnehmer zu den Verhaltenskategorien und der Selbsteinschätzung in Studie 3.	171
Tab. 7.0.1:	Ergänzung der Einflussfaktoren der Benutzer-Komponente im Parameterraum.	189
Tab. A.1.1:	Beispielberechnung für unterschiedliche Körpergrößen. . . .	195
Tab. A.2.2:	Auswertung der Fragebögen aus Studie 1.	196
Tab. A.2.3:	Auswertung der Fragebögen aus Studie 2.	199
Tab. A.3.4:	Häufigkeit der benötigten Selektionen für jeden Objekttyp und Mappingfunktion im ersten und zweiten Durchlauf.	209
Tab. A.3.5:	Häufigkeitsverteilung der Sortiersuche und Fehlerrate in Studie 3.	212

Tab. A.3.6: Zusammenfassung der durchgeführten Signifikanztests in Studie 3.	227
Tab. A.3.7: Die durchschnittlichen z-Werte der Nutzerperformanz in Studie 3.	228

Literaturverzeichnis

- [AA07] ANDUJAR, Carlos ; ARGELAGUET, Ferran: Anisomorphic ray-casting manipulation for interacting with 2D GUIs. In: *Computers & Graphics* 31 (2007), Nr. 1, S. 15 – 25
- [ABW01] ALLEN, B. D. ; BISHOP, Gary ; WELCH, Greg: *Tracking: Beyond 15 Minutes of Thought*. SIGGRAPH Course Pack 11, 2001
- [AEN10] ANDREWS, Christopher ; ENDERT, Alex ; NORTH, Chris: Space to think: large high-resolution displays for sensemaking. In: *Proceedings of the 28th international conference on Human factors in computing systems*. New York, NY, USA : ACM, 2010 (CHI '10), S. 55–64
- [AEYN11] ANDREWS, Christopher ; ENDERT, Alex ; YOST, Beth ; NORTH, Chris: Information visualization on large, high-resolution displays: issues, challenges, and opportunities. In: *Information Visualization* 10 (2011), Oktober, Nr. 4, S. 341–355
- [AOSS05] ASHDOWN, Mark ; OKA, Kenji ; SATO, Yoichi ; SATO, Yoichi: Combining head tracking and mouse input for a GUI on multiple monitors. In: *CHI Extended Abstracts*, 2005, S. 1188–1191
- [ATK⁺05] AHLBORN, Benjamin A. ; THOMPSON, David ; KREYLOS, Oliver ; HAMANN, Bernd ; STAADT, Oliver G.: A Practical System for Laser Pointer Interaction on Large Displays. In: *Proceedings of the ACM symposium on Virtual reality software and technology*. New York, NY, USA : ACM, 2005 (VRST '05), S. 106–109
- [Bal04] BALAKRISHNAN, Ravin: “Beating“ Fitts’ law: virtual enhancements for pointing facilitation. In: *Int. J. Hum.-Comput. Stud.* 61 (2004), Dezember, Nr. 6, S. 857–874
- [BBB10] BI, Xiaojun ; BAE, Seok-Hyung ; BALAKRISHNAN, Ravin: Effects of interior bezels of tiled-monitor large displays on visual search, tunnel steering, and target selection. In: *Proceedings of the 28th international conference on Human factors in computing systems*. New York, NY, USA : ACM, 2010 (CHI '10), S. 65–74
- [BBM⁺06] BADILLO, Brian ; BOWMAN, Doug A. ; MCCONNELL, William ; NI, Tao ; SILVA, Mara G.: Literature Survey on Interaction Techniques for Large

Displays / Computer Science, Virginia Tech. 2006 (TR-06-21). – Technical Report

- [BCC⁺03] BAUDISCH, Patrick ; CUTRELL, Edward ; CZERWINSKI, Mary ; ROBINS, Daniel C. ; TANDLER, Peter ; BEDERSON, Benjamin B. ; ZIERLINGER, Alex: Drag-and-Pop and Drag-and-Pick: Techniques for Accessing Remote Screen Content on Touch- and Pen-Operated Systems. In: *INTERACT*, IOS Press, 2003, S. 57–64
- [BCHE05] BAUDISCH, Patrick ; CUTRELL, Edward ; HINCKLEY, Ken ; EVERSOLE, Adam: Snap-and-go: Helping Users Align Objects without the Modality of Traditional Snapping. In: *CHI*, Press, 2005, S. 301–310
- [BD06] BORTZ, Jürgen ; DÖRING, Nicola: *Forschungsmethoden und Evaluation für Human- und Sozialwissenschaftler*. 4., überarbeitete Auflage. Springer Berlin Heidelberg, 2006
- [BDN⁺06] BALL, Robert ; DELLANOCE, Michael ; NI, Tao ; QUEK, Francis ; NORTH, Chris: Applying Embodied Interaction and Usability Engineering to Visualization on Large Displays. In: *ACM British HCI - Workshop on Visualization & Interaction*, 2006, S. 57–65
- [Bel08] BELLER, Sieghard: *Empirisch forschen lernen - Konzepte, Methoden, Fallbeispiele, Tipps*. 2., überarbeitete Auflage. Verlag Hans Huber, 2008
- [BFBK00] BUXTON, William ; FITZMAURICE, George ; BALAKRISHNAN, Ravin ; KURTENBACH, Gordon: Large displays in automotive design. In: *Computer Graphics and Applications, IEEE 20* (2000), Nr. 4, S. 68–75
- [BGBL04] BLANCH, Renaud ; GUIARD, Yves ; BEAUDOUIN-LAFON, Michel: Semantic pointing: improving target acquisition with control-display ratio adaptation. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA : ACM, 2004 (CHI '04), S. 519–526
- [BGS01] BAUDISCH, Patrick ; GOOD, Nathaniel ; STEWART, Paul: Focus plus context screens: combining display technology with visualization techniques. In: *Proceedings of the 14th annual ACM symposium on User interface software and technology*. New York, NY, USA : ACM, 2001 (UIST '01), S. 31–40
- [BI12] BEZERIANOS, Anastasia ; ISENBERG, Petra: Perception of Visual Variables on Tiled Wall-Sized Displays for Information Visualization Applications. In: *IEEE Transactions on Visualization and Computer Graphics* 18 (2012), S. 2516–2525

- [BKLP04] BOWMAN, Doug A. ; KRUIJFF, Ernst ; LAVIOLA, Joseph J. ; POUPYREV, Ivan: *3D User Interfaces: Theory and Practice*. Redwood City, CA, USA : Addison Wesley Longman Publishing Co., Inc., 2004
- [BM86] BUXTON, William ; MYERS, Brad A.: A study in two-handed input. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* Bd. 17. New York, NY, USA : ACM, April 1986, S. 321–326
- [BN05a] BALL, Robert ; NORTH, Chris: Analysis of User Behavior on High-Resolution Tiled Displays. In: COSTABILE, Maria F. (Hrsg.) ; PATERNÒ, Fabio (Hrsg.): *Human-Computer Interaction - INTERACT 2005* Bd. 3585. Springer Berlin / Heidelberg, 2005, S. 350–363
- [BN05b] BALL, Robert ; NORTH, Chris: Effects of tiled high-resolution display on basic visualization and navigation tasks. In: *CHI '05 extended abstracts on Human factors in computing systems*. New York, NY, USA : ACM, 2005 (CHI '05), S. 1196–1199
- [BN07] BALL, Robert ; NORTH, Chris: Realizing embodied interaction for visual analytics through large displays. In: *Computers & Graphics* 31 (2007), Nr. 3, S. 380 – 400
- [BN08] BALL, Robert ; NORTH, Chris: The effects of peripheral vision and physical navigation on large scale visualization. In: *Proceedings of graphics interface 2008*. Toronto, Ont., Canada, Canada : Canadian Information Processing Society, 2008 (GI '08), S. 9–16
- [BNB07] BALL, Robert ; NORTH, Chris ; BOWMAN, Doug A.: Move to improve: promoting physical navigation to increase user performance with large displays. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA : ACM, 2007 (CHI '07), S. 191–200
- [Bor99a] BORITZ, James: *The effectiveness of three-dimensional interaction*, The University of British Columbia (Canada), Dissertation, 1999
- [Bor99b] BORTZ, Jürgen: *Statistik für Sozialwissenschaftler*. 5., vollständig überarbeitet und aktualisierte Auflage. Springer-Verlag, 1999
- [Bro96] BROOKE, John: SUS - A quick and dirty usability scale. In: JORDAN, P.W. (Hrsg.) ; THOMAS, B. (Hrsg.) ; WEERDMEESTER, B.A. (Hrsg.) ; MCCLELLAND (Hrsg.): *Usability Evaluation in Industry*. London, UK : Taylor & Francis, 1996, S. 189–194
- [BSS13] BRUDER, Gerd ; STEINICKE, Frank ; STURZLINGER, Wolfgang: To Touch or Not to Touch?: Comparing 2D Touch and 3D Mid-air Interaction on Stereoscopic Tabletop Surfaces. In: *Proceedings of the 1st Sympo-*

- sium on Spatial User Interaction*. New York, NY, USA : ACM, 2013 (SUI '13), S. 9–16
- [BW00] BISHOP, Gary ; WELCH, Greg: Working in the Office of "Real Soon Now". In: *IEEE Comput. Graph. Appl.* 20 (2000), Juli, Nr. 4, S. 76–78
- [BWB06] BENKO, Hrvoje ; WILSON, Andrew D. ; BAUDISCH, Patrick: Precise selection techniques for multi-touch screens. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA : ACM, 2006 (CHI '06), S. 1263–1272
- [CB03] CAO, Xiang ; BALAKRISHNAN, Ravin: VisionWand: interaction techniques for large displays using a passive wand tracked in 3D. In: *Proceedings of the 16th annual ACM symposium on User interface software and technology*. New York, NY, USA : ACM, 2003 (UIST '03), S. 173–182
- [CDN88] CHIN, John P. ; DIEHL, Virginia A. ; NORMAN, Kent L.: Development of an Instrument Measuring User Satisfaction of the Human-computer Interface. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA : ACM, 1988 (CHI '88), S. 213–218
- [CEB78] CARD, Stuart K. ; ENGLISH, William K. ; BURR, Betty J.: Evaluation of Mouse, Rate-Controlled Isometric Joystick, Step Keys, and Text Keys for Text Selection on a CRT. In: *Ergonomics* 21 (1978), Nr. 8, S. 601–613
- [CF03] COCKBURN, Andy ; FIRTH, Andrew: Improving the Acquisition of Small Targets. In: *British HCI Conference, 2003*, S. 181–196
- [CHBL05] COLLOMB, Maxime ; HASCOËT, Mountaz ; BAUDISCH, Patrick ; LEE, Brian: Improving drag-and-drop on wall-size displays. In: *Proceedings of Graphics Interface 2005*. School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada : Canadian Human-Computer Communications Society, 2005 (GI '05), S. 25–32
- [CP03] CHENG, Kelvin ; PULO, Kevin: Direct Interaction with Large-Scale Display Systems using Infrared Laser Tracking Devices. In: *Proceedings of the Asia-Pacific symposium on Information visualisation - Volume 24*. Darlinghurst, Australia, Australia : Australian Computer Society, Inc., 2003 (APVis '03), S. 67–74
- [CRM⁺06] CZERWINSKI, Mary ; ROBERTSON, George ; MEYERS, Brian ; SMITH, Greg ; ROBBINS, Daniel ; TAN, Desney: Large display research overview. In: *CHI '06 extended abstracts on Human factors in computing systems*. New York, NY, USA : ACM, 2006 (CHI EA '06), S. 69–74

- [CSR⁺03] CZERWINSKI, Mary ; SMITH, Greg ; REGAN, Tim ; MEYERS, Brian ; ROBERTSON, George G. ; STARKWEATHER, Gary: Toward Characterizing the Productivity Benefits of Very Large Displays. In: *Interact 2003*, IOS Press, 2003, S. 9–16
- [CVBC08] CASIEZ, Géry ; VOGEL, Daniel ; BALAKRISHNAN, Ravin ; COCKBURN, Andy: The Impact of Control-Display Gain on User Performance in Pointing Tasks. In: *Human-Computer Interaction 23* (2008), Nr. 3, S. 215–250
- [DB10] DAS, Kaushik ; BORST, Christoph W.: An evaluation of menu properties and pointing techniques in a projection-based VR environment. In: *3D User Interfaces (3DUI), 2010 IEEE Symposium on*, 2010, S. 47–50
- [DC02] DAVIS, James ; CHEN, Xing: Lumipoint: multi-user laser-based interaction on large tiled displays. In: *Displays 23* (2002), Nr. 5, S. 205–211
- [DFAB04] DIX, Alan ; FINLAY, Janet ; ABOWD, Gregory ; BEALE, Russell: *Human Computer Interaction*. 3rd. Prentice Hall, 2004
- [DIN99a] DIN DEUTSCHES INSTITUT FÜR NORMUNG E. V.: Drucktechnik - Schriftgrößen - Teil 2: Digitaler Satz und verwandte Techniken. Beuth Verlag GmbH, 1999 (DIN 16507-2:1999). – Norm
- [DIN99b] DIN DEUTSCHES INSTITUT FÜR NORMUNG E. V.: Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten - Teil 11: Anforderungen an die Gebrauchstauglichkeit - Leitsätze. Beuth Verlag GmbH, 1999 (ISO 9241-11:1998). – Norm
- [DIN12] DIN DEUTSCHES INSTITUT FÜR NORMUNG E. V.: Ergonomie der Mensch-System-Interaktion - Teil 303: Anforderungen an elektronische optische Anzeigen. Beuth Verlag GmbH, 2012 (ISO 9241-303:2011). – Norm
- [DK11] DOERR, Kai-Uwe ; KUESTER, Falko: CGLX: A Scalable, High-Performance Visualization Framework for Networked Display Environments. In: *IEEE Transactions on Visualization and Computer Graphics 17* (2011), Mai, Nr. 3, S. 320–332
- [Duc02] DUCHOWSKI, Andrew T.: A breadth-first survey of eye-tracking applications. In: *Behavior Research Methods, Instruments, & Computers 34* (2002), Nr. 4, S. 455–470
- [EBZ⁺12] ENDERT, Alex ; BRADEL, Lauren ; ZEITZ, Jessica ; ANDREWS, Christopher ; NORTH, Chris: Designing large high-resolution display workspaces. In: *Proceedings of the International Working Conference on*

- Advanced Visual Interfaces*. New York, NY, USA : ACM, 2012 (AVI '12), S. 58–65
- [FHH00] FRØKJÆR, Erik ; HERTZUM, Morten ; HORNBJÆK, Kasper: Measuring usability: are effectiveness, efficiency, and satisfaction really correlated? In: *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. New York, NY, USA : ACM, 2000 (CHI '00), S. 345–352
- [Fit54] FITTS, Paul M.: The information capacity of the human motor system in controlling the amplitude of movement. In: *Journal of Experimental Psychology* 47 (1954), S. 381–391
- [FK05] FREES, Scott ; KESSLER, G. D.: Precise and rapid interaction through scaled manipulation in immersive virtual environments. In: *Virtual Reality, 2005. Proceedings. VR 2005. IEEE, 2005*, S. 99–106
- [FTS⁺10] FUKAZAWA, R. ; TAKASHIMA, K. ; SHOEMAKER, G. ; KITAMURA, Y. ; ITOH, Y. ; KISHINO, F.: Comparison of Multimodal Interactions in Perspective-corrected Multi-display Environment. In: *3D User Interfaces (3DUI), 2010 IEEE Symposium on*, 2010, S. 103–110
- [FVB06] FORLINES, Clifton ; VOGEL, Daniel ; BALAKRISHNAN, Ravin: Hybrid-Pointing: fluid switching between absolute and relative pointing with a direct input device. In: *Proceedings of the 19th annual ACM symposium on User interface software and technology*. New York, NY, USA : ACM, 2006 (UIST '06), S. 211–220
- [FWC84] FOLEY, James D. ; WALLACE, Victor L. ; CHAN, Peggy: The human factors of computer graphics interaction techniques. In: *IEEE Comput. Graph. Appl.* 4 (1984), November, Nr. 11, S. 13–48
- [GB05] GROSSMAN, Tovi ; BALAKRISHNAN, Ravin: The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA : ACM, 2005 (CHI '05), S. 281–290
- [GBK⁺01] GROSSMAN, Tovi ; BALAKRISHNAN, Ravin ; KURTENBACH, Gordon ; FITZMAURICE, George ; KHAN, Azam ; BUXTON, Bill: Interaction techniques for 3D modeling on large displays. In: *Proceedings of the 2001 symposium on Interactive 3D graphics*. New York, NY, USA : ACM, 2001 (I3D '01), S. 17–23
- [GLB⁺04] GUIARD, Yves ; BEAUDOUIN-LAFON, Michel ; BASTIN, Julien ; PASVEER, Dennis ; ZHAI, Shumin: View size and pointing difficulty in multi-scale navigation. In: *Proceedings of the working conference on Advanced visual interfaces*. New York, NY, USA : ACM, 2004 (AVI '04), S. 117–124

- [Ger11] GERTZ, Thomas: *Interaktive Exploration von Graphen in einer hochauflösenden Displayumgebung*, Universität Rostock, Diplomarbeit, Mai 2011
- [GFD⁺12] GUENTER, Brian ; FINCH, Mark ; DRUCKER, Steven ; TAN, Desney ; SNYDER, John: Foveated 3D graphics. In: *ACM Trans. Graph.* 31 (2012), November, Nr. 6, S. 164:1–164:10
- [Gir13] GIRKE, Steffen: *Perzeptionsgesteuerte Level of Detail Auswahl für die Geländedarstellung in großen Displayumgebungen auf der Basis von Trackinginformationen*, Universität Rostock, Diplomarbeit, April 2013
- [Gra96] GRAHAM, Evan D.: Virtual Pointing on a Computer Display: Non-Linear Control-Display Mappings. In: DAVIS, Wayne A. (Hrsg.) ; BARTELS, Richard M. (Hrsg.): *Graphics Interface*. Toronto, Ontario, Canada : Canadian Human-Computer Communications Society, Mai 1996, S. 39–46
- [GSW01] GUIMBRETIÈRE, François ; STONE, Maureen C. ; WINOGRAD, Terry: Fluid interaction with high-resolution wall-size displays. In: *UIST 2001*, 2001, S. 21–30
- [Gui87] GUIARD, Yves: Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as a Model. In: *Journal of Motor Behavior* 19 (1987), S. 486–517
- [Hal90] HALL, Edward T.: *The hidden dimension*. Reprint. New York [u.a.] : Doubleday, 1990
- [HAM11] HRIMECH, Hamid ; ALEM, Leila ; MERIENNE, Frederic: How 3D interaction metaphors affect user experience in collaborative virtual environment. In: *Adv. in Hum.-Comp. Int.* 2011 (2011), Januar, S. 8:1–8:11
- [Har06] HART, Sandra G.: Nasa-Task Load Index (NASA-TLX); 20 Years Later. In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 50 (2006), Nr. 9, S. 904–908
- [HK05] HEIDER, Thomas ; KIRSTE, Thomas: Multimodal Appliance Cooperation Based on Explicit Goals: Concepts & Potentials. In: *Proceedings of the 2005 Joint Conference on Smart Objects and Ambient Intelligence: Innovative Context-aware Services: Usages and Technologies*. New York, NY, USA : ACM, 2005 (sOc-EUSAI '05), S. 271–276
- [Hor06] HORNBAEK, Kasper: Current practice in measuring usability: Challenges to usability studies and research. In: *International Journal of Human-Computer Studies* 64 (2006), Nr. 2, S. 79 – 102

- [HPPK98] HINCKLEY, Ken ; PAUSCH, Randy ; PROFFITT, Dennis ; KASSELL, Neal F.: Two-handed virtual manipulation. In: *ACM Trans. Comput.-Hum. Interact.* 5 (1998), September, Nr. 3, S. 260–302
- [HRS00] HÄGER-ROSS, Charlotte ; SCHIEBER, Marc H.: Quantifying the Independence of Human Finger Movements: Comparisons of Digits, Hands, and Movement Frequencies. In: *The Journal of Neuroscience* 20 (2000), Nr. 22, S. 8542–8550
- [HW12] HINCKLEY, Ken ; WIGDOR, Daniel: Input Technologies and Techniques. In: JACKO, Julie A. (Hrsg.): *Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications*. 3rd. CRC Press Taylor & Francis Group, 2012, Kapitel 9
- [IDW⁺13] ISENBERG, Petra ; DRAGICEVIC, Pierre ; WILLETT, Wesley ; BEZERIANOS, Anastasia ; FEKETE, Jean-Daniel: Hybrid-Image Visualization for Large Viewing Environments. In: *IEEE Transactions on Visualization and Computer Graphics* 19 (2013), Dezember, Nr. 12, S. 2346–2355
- [IKHW08] INGRAM, James N. ; KÖRDING, Konrad P. ; HOWARD, Ian S. ; WOLPERT, Daniel M.: The statistics of natural hand movements. In: *Experimental Brain Research* Bd. 188. Springer-Verlag, Juni 2008, S. 223–236
- [Jac97] JACOB, Robert J. K.: Input Devices and Techniques. In: TUCKER, Allen B. (Hrsg.): *The Computer Science and Engineering Handbook*. CRC Press, 1997, S. 1494–1511
- [JHKB10] JEON, Seokhee ; HWANG, Jane ; KIM, Gerard J. ; BILLINGHURST, Mark: Interaction with large ubiquitous displays using camera-equipped mobile phones. In: *Personal Ubiquitous Comput.* 14 (2010), Februar, S. 83–94
- [JPJ09] JOTA, Ricardo ; PEREIRA, J. M. ; JORGE, Joaquim A.: A Comparative Study of Interaction Metaphors for Large-Scale Displays. In: *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*. New York, NY, USA : ACM, 2009 (CHI '09), S. 4135–4140
- [JSHKH13] JAKOBSEN, Mikkel R. ; SAHLEMARIAM HAILE, Yonas ; KNUDSEN, Søren ; HORNBÆK, Kasper: Information Visualization and Proxemics: Design Opportunities and Empirical Findings. In: *IEEE Transactions on Visualization and Computer Graphics* 19 (2013), Dezember, Nr. 12, S. 2386–2395
- [KAHF05] KIM, Hyosun ; ALBUQUERQUE, Georgia ; HAVEMANN, Sven ; FELLNER, Dieter W.: Tangible 3D: Hand Gesture Interaction for Immersive

- 3D Modeling. In: *Proc. Eurographics Conference on Virtual Environments (EGVE) 2005*. Aire-la-Ville, Switzerland, Switzerland : Eurographics Association, Oktober 2005, S. 191–199
- [KBSM10] KOPPER, Regis ; BOWMAN, Doug A. ; SILVA, Mara G. ; MCMAHAN, Ryan P.: A human motor behavior model for distal pointing tasks. In: *International Journal of Human-Computer Studies* 68 (2010), Nr. 10, S. 603 – 615
- [KBSR07] KÖNIG, Werner A. ; BIEG, Hans-Joachim ; SCHMIDT, Toni ; REITERER, Harald: Position-Independent Interaction for Large High-Resolution Displays. In: *Interfaces and Human Computer Interaction (IHCI'07)*, IADIS Press, Juli 2007, S. 117–125
- [KCHI07] KEIJSER, Jeron ; CARPENDALE, Sheelagh ; HANCOCK, Mark ; ISENBERG, Tobias: Exploring 3D Interaction in Alternate Control-Display Space Mappings. In: *3D User Interfaces, 2007. 3DUI '07. IEEE Symposium on, 2007*
- [KGDR09] KÖNIG, Werner A. ; GERKEN, Jens ; DIERDORF, Stefan ; REITERER, Harald: Adaptive Pointing — Design and Evaluation of a Precision Enhancing Technique for Absolute Pointing Devices. In: *Proceedings of the 12th IFIP TC 13 International Conference on Human-Computer Interaction: Part I*. Berlin, Heidelberg : Springer-Verlag, 2009 (INTERACT '09), S. 658–671
- [Kha11] KHAN, Taimur K.: A Survey of Interaction Techniques and Devices for Large High Resolution Displays. In: MIDDEL, Ariane (Hrsg.) ; SCHELLER, Inga (Hrsg.) ; HAGEN, Hans (Hrsg.): *Visualization of Large and Unstructured Data Sets - Applications in Geospatial Planning, Modeling and Engineering (IRTG 1131 Workshop)* Bd. 19. Dagstuhl, Germany : Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2011 (OpenAccess Series in Informatics (OASICs)), S. 27–35
- [Kön10] KÖNIG, Werner A.: *Design and evaluation of novel input devices and interaction techniques for large, high-resolution displays*, Universität Konstanz, Dissertation, September 2010
- [Kre08] KREYLOS, Oliver: Environment-Independent VR Development. In: *Proceedings of the 4th International Symposium on Advances in Visual Computing*. Berlin, Heidelberg : Springer-Verlag, 2008 (ISVC '08), S. 901–912
- [KSMB08] KOPPER, Regis ; SILVA, Mara G. ; MCMAHAN, Ryan P. ; BOWMAN, Doug A.: Increasing the Precision of Distant Pointing for Large High-

- Resolution Displays / Computer Science, Virginia Tech. 2008 (TR-08-17). – Technical Report
- [Lak10] LAKIE, Martin: The influence of muscle tremor on shooting performance. In: *Experimental Physiology* 95 (2010), Nr. 3, S. 441–450
- [Lew93] LEWIS, James R.: IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instructions for Use / Human Factors Group. Boca Raton, FL, 1993 (54.786). – Technical Report
- [Lew95] LEWIS, James R.: IBM computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use. In: *International Journal of Human-Computer Interaction* 7 (1995), Nr. 1, S. 57–78
- [LFKZ01] LAVIOLA, Joseph J. Jr. ; FELIZ, Daniel A. ; KEEFE, Daniel F. ; ZELEZNIK, Robert C.: Hands-free multi-scale navigation in virtual environments. In: *Proceedings of the 2001 symposium on Interactive 3D graphics*. New York, NY, USA : ACM, 2001 (I3D '01), S. 9–15
- [LK07] LAVIOLA, Joseph J. ; KATZOURIN, Michael: An Exploration of Non-Isomorphic 3D Rotation in Surround Screen Virtual Environments. In: *3D User Interfaces, 2007. 3DUI '07. IEEE Symposium on*, 2007, S. 49–54
- [LS12] LEHMANN, Anke ; STAADT, Oliver: Distance-adapted 2D Manipulation Techniques for Large High-Resolution Display Environments. In: *Proceedings of the International Conference on Computer Graphics Theory and Applications*. Rom, Italy : SciTePress, Februar 2012, S. 387–394
- [LS13] LEHMANN, Anke ; STAADT, Oliver: Distance-Aware Bimanual Interaction for Large High-Resolution Displays. In: CSURKA, Gabriela (Hrsg.) ; KRAUS, Martin (Hrsg.) ; LARAMEE, Robert S. (Hrsg.) ; RICHARD, Paul (Hrsg.) ; BRAZ, José (Hrsg.): *Computer Vision, Imaging and Computer Graphics. Theory and Application* Bd. 359. Springer Berlin Heidelberg, 2013, S. 97–111
- [LSC08] LUBOSCHIK, Martin ; SCHUMANN, Heidrun ; COORDS, Hilko: Particle-Based Labeling: Fast Point-Feature Labeling Without Obscuring Other Visual Features. In: *IEEE Transactions on Visualization and Computer Graphics* 14 (2008), Nr. 6
- [LSS10] LÖFFLER, Falko ; SCHWANKE, Sebastian ; SCHUMANN, Heidrun: A Hybrid Approach for High Quality Real-time Terrain Rendering and Optimized A-priori Error Estimation. In: *Proceedings of the International Conference on Computer Graphics Theory and Applications*. Angers, France, Mai 2010, S. 233–238

- [LSST11] LEHMANN, Anke ; SCHUMANN, Heidrun ; STAADT, Oliver ; TOMINSKI, Christian: Physical Navigation to Support Graph Exploration on a Large High-Resolution Display. In: *Proceedings of the 7th international conference on Advances in visual computing - Volume Part I*. Berlin, Heidelberg : Springer-Verlag, 2011 (ISVC'11), S. 496–507
- [LZB98] LEGANCHUK, Andrea ; ZHAI, Shumin ; BUXTON, William: Manual and cognitive benefits of two-handed input: an experimental study. In: *ACM Trans. Comput.-Hum. Interact.* 5 (1998), Dezember, Nr. 4, S. 326–359
- [Mac92] MACKENZIE, I. S.: Fitts' Law as a Research and Design Tool in Human-Computer Interaction. In: *Human-Computer Interaction 7* (1992), Nr. 1, S. 91–139
- [MAK⁺88] MEYER, David E. ; ABRAMS, Richard A. ; KORNBLUM, Sylvan ; WRIGHT, Charles E. ; SMITH, J. E. K.: Optimality in human motor performance: ideal control of rapid aimed movements. In: *Psychological Review* 95 (1988), S. 340–370
- [MBN⁺02] MYERS, Brad A. ; BHATNAGAR, Rishi ; NICHOLS, Jeffrey ; PECK, Choon H. ; KONG, Dave ; MILLER, Robert ; LONG, A. C.: Interacting at a distance: measuring the performance of laser pointers and other devices. In: *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves*. New York, NY, USA : ACM, 2002 (CHI '02), S. 33–40
- [Min95] MINE, Mark: Virtual Environment Interaction Techniques / Department of Computer Science, University of North Carolina at Chapel Hill. 1995. – Technical Report
- [MR94] MACKENZIE, I. S. ; RIDDERSMA, Stan: Effects of output display and control—display gain on human performance in interactive systems. In: *Behaviour & Information Technology* 13 (1994), Nr. 5, S. 328–337
- [MRB05] MALIK, Shahzad ; RANJAN, Abhishek ; BALAKRISHNAN, Ravin: Interacting with large displays from a distance with vision-tracked multi-finger gestural input. In: *Proceedings of the 18th annual ACM symposium on User interface software and technology*. New York, NY, USA : ACM, 2005 (UIST '05), S. 43–52
- [NBC06] NI, Tao ; BOWMAN, Doug A. ; CHEN, Jian: Increased display size and resolution improve task performance in Information-Rich Virtual Environments. In: *Proceedings of Graphics Interface 2006*. Toronto, Ont., Canada, Canada : Canadian Information Processing Society, 2006 (GI '06), S. 139–146

- [NL94] NIELSEN, Jakob ; LEVY, Jonathan: Measuring usability: preference vs. performance. In: *Communications of the ACM* 37 (1994), April, Nr. 4, S. 66–75
- [NPBL11] NANCEL, Mathieu ; PIETRIGA, Emmanuel ; BEAUDOUIN-LAFON, Michel: Precision Pointing for Ultra-High-Resolution Wall Displays / INRIA, Parc Orsay Université. 2011 (RR-7624). – Technical Report
- [NSMG03] NIELSEN, Michael ; STÖRRING, Moritz ; MOESLUND, Thomas B. ; GRANUM, Erik: A Procedure for Developing Intuitive and Ergonomic Gesture Interfaces for HCI. In: CAMURRI, A. (Hrsg.) ; VOLPE, G. (Hrsg.): *Gesture-Based Communication in Human-Computer Interaction, 5th International Gesture Workshop*. Genova, Italy : Springer Berlin Heidelberg, April 2003 (Lecture notes in artificial intelligence. Subseries of Lecture notes in computer science), S. 409–420
- [NSS⁺06] NI, Tao ; SCHMIDT, Greg S. ; STAADT, Oliver G. ; LIVINGSTON, Mark A. ; BALL, Robert ; MAY, Richard: A Survey of Large High-Resolution Display Technologies, Techniques, and Applications. In: *Virtual Reality Conference, 2006*, 2006, S. 223 – 236
- [NWP⁺11] NANCEL, Mathieu ; WAGNER, Julie ; PIETRIGA, Emmanuel ; CHAPUIS, Olivier ; MACKAY, Wendy: Mid-air pan-and-zoom on wall-sized displays. In: *Proceedings of the 2011 annual conference on Human factors in computing systems*. New York, NY, USA : ACM, 2011 (CHI '11), S. 177–186
- [PBWI96] POUPYREV, Ivan ; BILLINGHURST, Mark ; WEGHORST, Suzanne ; ICHIKAWA, Tadao: The Go-Go Interaction Technique: Non-Linear Mapping for Direct Manipulation in VR. In: *ACM Symposium on User Interface Software and Technology'96*, 1996, S. 79–80
- [PDW⁺11] PONTO, Kevin ; DOERR, Kai ; WYPYCH, Tom ; KOOKER, John ; KUESTER, Falko: CGLXTouch: A multi-user multi-touch approach for ultra-high-resolution collaborative workspaces. In: *Future Gener. Comput. Syst.* 27 (2011), Juni, Nr. 6, S. 649–656
- [PNB09] PECK, Sarah M. ; NORTH, Chris ; BOWMAN, Doug: A Multiscale Interaction Technique for Large, High-Resolution Displays. In: *3D User Interfaces 0* (2009), S. 31–38
- [PPK13] PAPADOPOULOS, Charilaos ; PETKOV, Kaloian ; KAUFMAN, Ari: Building the Reality Deck. In: *POWERWALL: International Workshop on Interactive, Ultra-High-Resolution Displays, part of the SIGCHI Conference on Human Factors in Computing Systems*, 2013

- [Pre99] PREIM, Bernhard: *Entwicklung interaktiver Systeme - Grundlagen, Fallbeispiele und innovative Anwendungsfelder*. Berlin ; Heidelberg [u.a.] : Springer, 1999 (Springer-Lehrbuch)
- [PWF00] POUPYREV, Ivan ; WEGHORST, Suzanne ; FELS, Sidney: Non-isomorphic 3D rotational techniques. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA : ACM, 2000 (CHI '00), S. 540–547
- [Rad14] RADLOFF, Axel: *Smart Views in Smart Meeting Rooms*, Universität Rostock, Dissertation, 2014
- [RB05] RAMOS, Gonzalo ; BALAKRISHNAN, Ravin: Zliding: fluid zooming and sliding for high precision parameter manipulation. In: *Proceedings of the 18th annual ACM symposium on User interface software and technology*. New York, NY, USA : ACM, 2005 (UIST '05), S. 143–152
- [RCB⁺05] ROBERTSON, G. ; CZERWINSKI, M. ; BAUDISCH, P. ; MEYERS, B. ; ROBINS, D. ; SMITH, G. ; TAN, D.: The large-display user experience. In: *Computer Graphics and Applications, IEEE 25* (2005), Nr. 4, S. 44 – 51
- [RFHN06] RASCH, Björn ; FRIESE, Malte ; HOFMANN, Wilhelm ; NAUMANN, Ewald: *Quantitative Methoden Band 1: Einführung in die Statistik*. 2., erweiterte Auflage. Springer Medizin Verlag, 2006 (Springer-Lehrbuch)
- [RLS11] RADLOFF, Axel ; LUBOSCHIK, Martin ; SCHUMANN, Heidrun: Smart Views in Smart Environments. In: DICKMANN, Lutz (Hrsg.) ; VOLKMANN, Gerald (Hrsg.) ; MALAKA, Rainer (Hrsg.) ; BOLL, Susanne (Hrsg.) ; KRÜGER, Antonio (Hrsg.) ; OLIVIER, Patrick (Hrsg.): *Smart Graphics - 11th International Symposium, SG 2011, Bremen, Germany, July 18-20, 2011. Proceedings* Bd. 6815, Springer, 2011, S. 1–12
- [RLSS12] RADLOFF, Axel ; LEHMANN, Anke ; STAADT, Oliver ; SCHUMANN, Heidrun: Smart Interaction Management: An Interaction Approach for Smart Meeting Rooms. In: *Proceedings of the 8th International Conference on Intelligent Environments (IE'12)*. Guanajuato, Mexico : IEEE Computer Society, Juni 2012, S. 228–235
- [SADK⁺09] SHUPP, Lauren ; ANDREWS, Christopher ; DICKEY-KURDZIOLEK, Margaret ; YOST, Beth ; NORTH, Chris: Shaping the Display of the Future: The Effects of Display Size and Curvature on User Performance and Insights. In: *Human-Computer Interaction 24* (2009), S. 230–272
- [Saf08] SAFFER, Dan: *Designing Gestural Interfaces: Touchscreens and Interactive Devices*. O'Reilly Media, Inc., 2008

- [SAG09] SWAN, J. E. ; ADELSTEIN, Bernhard D. ; GABBARD, Joseph L.: *Conducting Human-Subject Experiments with Virtual and Augmented Reality*. Tutorial presented at IEEE Virtual Reality 2009, März 2009
- [SAKH06] STAADT, Oliver G. ; AHLBORN, Benjamin A. ; KREYLOS, Oliver ; HAMMANN, Bernd: A foveal inset for large display environments. In: *Proceedings of the 2006 ACM international conference on Virtual reality continuum and its applications*. New York, NY, USA : ACM, 2006 (VRCIA '06), S. 281–288
- [SBA07] STØDLE, Daniel ; BJØRNDALEN, John M. ; ANSHUS, Otto J.: A system for hybrid vision- and sound-based interaction with distal and proximal targets on wall-sized, high-resolution tiled displays. In: *Proceedings of the 2007 IEEE international conference on Human-computer interaction*. Berlin, Heidelberg : Springer-Verlag, 2007 (HCI'07), S. 59–68
- [SBY⁺06] SHUPP, Lauren ; BALL, Robert ; YOST, Beth ; BOOKER, John ; NORTH, Chris: Evaluation of viewport size and curvature of large, high-resolution displays. In: *Proceedings of Graphics Interface 2006*. Toronto, Ont., Canada, Canada : Canadian Information Processing Society, 2006 (GI '06), S. 123–130
- [Seo05] SEOW, Steven C.: Information theoretic models of HCI: a comparison of the Hick-Hyman law and Fitts' law. In: *Hum.-Comput. Interact.* 20 (2005), September, Nr. 3, S. 315–352
- [Shn96] SHNEIDERMAN, Ben: The eyes have it: a task by data type taxonomy for information visualizations. In: *Visual Languages, 1996. Proceedings., IEEE Symposium on*, 1996, S. 336–343
- [SLL⁺14] SCHEEL, Christian ; LÖFFLER, Falko ; LEHMANN, Anke ; SCHUMANN, Heidrun ; STAADT, Oliver: Dynamic Level of Detail for Tiled Large High-Resolution Displays. In: *Virtuelle und Erweiterte Realität*. Bremen, Germany : Shaker Verlag, September 2014, S. 109–119
- [SRH05a] STEINICKE, Frank ; ROPINSKI, Timo ; HINRICHS, Klaus: Multimodal Interaction Metaphors for Manipulation of Distant Objects in Immersive Virtual Environments. In: *WSCG (Short Papers)'05*, 2005, S. 45–48
- [SRH05b] STEINICKE, Frank ; ROPINSKI, Timo ; HINRICHS, Klaus H.: VR and Laser-Based Interaction in Virtual Environments Using a Dual-Purpose Interaction Metaphor. In: *VR 2005 Workshop Proceedings on New Directions in 3D User Interfaces*. Bonn : IEEE Press, 2005, S. 61–64
- [Ste08] STEED, Anthony: A simple method for estimating the latency of interactive, real-time graphics simulations. In: *Proceedings of the 2008 ACM*

- symposium on Virtual reality software and technology*. New York, NY, USA : ACM, 2008 (VRST '08), S. 123–129
- [STKB10] SHOEMAKER, Garth ; TSUKITANI, Takayuki ; KITAMURA, Yoshifumi ; BOOTH, Kellogg S.: Body-Centric Interaction Techniques for Very Large Wall Displays. In: *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*. New York, NY, USA : ACM, 2010 (NordiCHI '10), S. 463–472
- [Sys08] SYSTAT SOFTWARE: *SigmaPlot Benutzerhandbuch Teil 2 - Statistik*. 2008
- [TC05] THOMAS, James J. ; COOK, Kristin A.: *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE Computer Society Press, 2005
- [TGSP06] TAN, Desney S. ; GERGLE, Darren ; SCUPELLI, Peter ; PAUSCH, Randy: Physically large displays improve performance on spatial tasks. In: *ACM Trans. Comput.-Hum. Interact.* 13 (2006), März, S. 71–99
- [Thi10] THIEDE, Conrad: *Visuelle Informationsdarstellung in Smart Environments*, Universität Rostock, Dissertation, 2010
- [THS⁺01] TAYLOR, Russell M. II ; HUDSON, Thomas C. ; SEEGER, Adam ; WEBER, Hans ; JULIANO, Jeffrey ; HELSER, Aron T.: VRPN: A Device-independent, Network-transparent VR Peripheral System. In: *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*. New York, NY, USA : ACM, 2001 (VRST '01), S. 55–61
- [TMM⁺09] THELEN, Sebastian ; MEYER, Joerg ; MIDDEL, Ariane ; OLECH, Peter-Scott ; EBERT, Achim ; HAGEN, Hans: Tag-based Interaction with Large High-Resolution Displays. In: *4th IASTED International Conference HUMAN-COMPUTER INTERACTION (HCI 2009)*. St. Thomas, US Virgin Islands, November 2009, S. 356 – 363
- [TSPP01] TAN, Desney S. ; STEFANUCCI, Jeanine K. ; PROFFITT, Dennis R. ; PAUSCH, Randy: The Infocockpit: providing location and place to aid human memory. In: *Proceedings of the 2001 workshop on Perceptive user interfaces*. New York, NY, USA : ACM, 2001 (PUI '01), S. 1–4
- [TTLNS05] TYNDIUK, Florence ; THOMAS, Gwenola ; LESPINET-NAJIB, Véronique ; SCHLICK, Christopher: Cognitive comparison of 3D interaction in front of large vs. small displays. In: *Proceedings of the ACM symposium on Virtual reality software and technology*. New York, NY, USA : ACM, 2005 (VRST '05), S. 117–123
- [Ulr02] ULRICH, Thatcher: Rendering massive terrains using chunked level of detail control / Oddworld Inhabitants. 2002. – Technical Report

- [VB04] VOGEL, Daniel ; BALAKRISHNAN, Ravin: Interactive Public Ambient Displays: Transitioning from Implicit to Explicit, Public to Personal, Interaction with Multiple Users. In: *Proceedings of the 17th annual ACM symposium on User interface software and technology*. New York, NY, USA : ACM, 2004 (UIST '04), S. 137–146
- [VB05] VOGEL, Daniel ; BALAKRISHNAN, Ravin: Distant Freehand Pointing and Clicking on Very Large, High Resolution Displays. In: *Proceedings of the 18th annual ACM symposium on User interface software and technology*. New York, NY, USA : ACM, 2005 (UIST '05), S. 33–42
- [War04] WARE, Colin: *Information Visualization: Perception for Design*. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 2004
- [WF02] WELCH, Greg ; FOXLIN, Eric: Motion Tracking: No Silver Bullet, but a Respectable Arsenal. In: *IEEE Comput. Graph. Appl.* 22 (2002), November, Nr. 6, S. 24–38
- [WM99] WANG, Yanqing ; MACKENZIE, Christine L.: Object manipulation in virtual environments: relative size matters. In: *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit*. New York, NY, USA : ACM, 1999 (CHI '99), S. 48–55
- [WOLS10] WILLERT, Malte ; OHL, Stephan ; LEHMANN, Anke ; STAADT, Oliver: The Extended Window Metaphor for Large High-Resolution Displays. In: *JVRC10: Joint Virtual Reality Conference of EGVE - EuroVR - VEC*. Stuttgart, Germany : Eurographics Association, 2010, S. 69–76
- [WST13] WAGNER, Benjamin ; STRIEBING, Björn ; TIMMERMANN, Dirk: A System for Live Localization In Smart Environments. In: *IEEE International Conference on Networking, Sensing and Control*. Paris, France, April 2013
- [YKSJ07] YI, Ji S. ; KANG, Youn a. ; STASKO, John ; JACKO, Julie: Toward a Deeper Understanding of the Role of Interaction in Information Visualization. In: *IEEE Transactions on Visualization and Computer Graphics* 13 (2007), November, Nr. 6, S. 1224–1231
- [Zha98] ZHAI, Shumin: User performance in relation to 3D input device design. In: *SIGGRAPH Comput. Graph.* 32 (1998), November, Nr. 4, S. 50–54

Thesen

1. Große hochauflösende Displays (GHDs) bieten eine große Anzeigefläche mit hoher Darstellungsgenauigkeit. Sie ermöglichen es Nutzern mit Informationen auf eine neue Art und Weise zu interagieren, z.B. mithilfe physischer Navigation.
2. In großen hochauflösenden Displayumgebungen soll der Nutzer in seiner gegenwärtigen Tätigkeit unterstützt werden, indem Visualisierung und Interaktion an ihn angepasst werden. In GHD-Umgebungen werden verschiedene Aufgaben nah oder entfernt vom Display ausgeführt, die oftmals eine unterschiedliche Präzision und Sensitivität der Nutzereingabe erfordern.
3. Das vorgestellte Interaction Scaling Konzept ermöglicht es den aktuellen Betrachtungsabstand des Nutzers zum Display für den Wechsel zwischen den Aufgabenbereichen zu nutzen. Dazu wird abhängig von der Interaktionsaufgabe und den verfügbaren Freiheitsgraden der Eingabemodalität ein distanzabhängiges Mapping von physischen Bewegungen (Körper, Eingabegerät) im realen Interaktionsraum auf virtuelle Bewegungen (Cursor, Detailgrad) im virtuellen Raum angewendet.
4. Die unterschiedlichen Einflussfaktoren für die Interaktion in GHD-Umgebungen lassen sich in einen Parameterraum mit den Komponenten physische Umgebung, virtuelle Umgebung, Benutzer und Mapping einordnen. Anhand von Beispielszenarien wurden die Gewichtung der Einflussfaktoren diskutiert und welche Anpassung für die Mappingmethode daraus resultiert.
5. Interaction Scaling wird für die distanzabhängige Informationsdarstellung und Interaktionspräzision verwendet, indem es das natürliche Verhalten der physischen Navigation in GHD-Umgebungen nutzt. Dazu wurden bekannte Navigations- und Manipulationstechniken erweitert, sowie distanzabhängige und taskbasierte Mappingmethoden integriert.
6. Da die individuellen und subjektiven Eigenschaften der Nutzer einen starken Einfluss auf die Mappingmethode haben, ist es denkbar, dass bei der distanzabhängigen Interaktion in GHD-Umgebungen die geeignete Mappingmethode und der angepasste Mappingfaktor zwischen den Nutzern variiert. Es wurden eine Reihe von Nutzerstudien durchgeführt, um diese Eigenschaften zu untersuchen.

-
7. Mit Hilfe von iterativen Studien konnte aufgezeigt werden, dass Interaction Scaling die Nutzerperformanz für 2D Manipulationstechniken zunehmend verbessert, wenn die benötigte Präzision steigt.
 8. Die relative Cursorberechnung ermöglicht eine präzise und grobe Objektmanipulation, führt jedoch die Seiteneffekte Clutching und Drift herbei. Die Möglichkeit zur Reduzierung des Drift-Effektes per Cursor Reset hat sich als weniger benutzerfreundlich herausgestellt. Es wurde auf die Präferenz von Nutzern eingegangen, indem ein interaktives Abschalten der distanzabhängigen Präzision umgesetzt wurde (manueller Cursorwechsel). Darüber hinaus hat sich die hybride Cursorberechnung als geeignet erwiesen, bei der eine taskbasierte Präzisionsanpassung durchgeführt wird, indem eine relative Cursorberechnung während der Manipulationstasks verwendet wird und eine absolute Cursorberechnung sonst.
 9. Es ist von der Interaktionstechnik abhängig, ob das automatische oder das manuelle Umschalten bevorzugt wird und nicht personenabhängig. Während die Präferenz für die asymmetrische oder symmetrische 2D Manipulationstechnik personenabhängig ist.
 10. In GHD-Umgebungen sollte eine semi-automatische distanzabhängige Anpassung der Präzision der Interaktion verwendet werden, so dass der Nutzer die Kontrolle über die Interaktionsgenauigkeit hat und auf Wunsch diese vom System übernommen wird. Des Weiteren müssen die Präferenzen der Benutzer bei der distanzabhängigen Interaktion berücksichtigt werden, z.B. die individuelle Distanz aus der sie in GHD-Umgebungen bevorzugt arbeiten.