

TECHNISCHE UNIVERSITÄT ILMENAU  
Fakultät für Elektrotechnik und Informationstechnik

## Diplomarbeit

# Benutzergerechte Menügestaltung für Virtuelle Umgebungen im Expertenkontext



Vorgelegt von:	Armin Dressler
Geboren am:	20.09.1975 in Düsseldorf
Studiengang:	Medientechnologie
Studienrichtung:	Medienproduktion
Verantwortliche Professorin:	Prof. Dr. phil. H. Krömker
Betreuende wiss. Mitarbeiter:	Dipl.-Ing. K. Hoffmann de Linares Dipl.-Ing. S. Metag
Externer Betreuer:	Dipl.-Phys. T. Holtkämper
Beginn der Arbeit:	01.04.2007
Abgabe der Arbeit:	22.10.2007

Ilmenau, den 20.10.2007



# Diplomthemenblatt

Diplomthemenblatt Platzhalter 1



# Danksagung

Prof. Dr. Heidi Krömker  
Katja Hoffmann de Linares  
Sebastian Metag  
Stephan Husung  
Dr. Manfred Bogen  
Thorsten Holtkämper  
Sascha Scholz  
Dr. Gerold Wesche  
den Mitarbeitern von IAIS.VE  
Julia Große Bley  
Pascale Bonus  
Max Ponert  
Oliver Havlat  
meiner Familie  
meiner WG  
meinen Freunden

für das Ermöglichen der Arbeit,  
für Hilfe, Unterstützung, Zeit, Verständnis und Motivation:  
Vielen Dank!



---

## **Kurzdarstellung**

In dieser Diplomarbeit wird der Entwurf, die Entwicklung und die Implementierung eines 3D-Menüsystems für immersive Virtual-Reality-Applikationen beschrieben. Dafür werden existierende 2D- und 3D-Menülösungen untersucht und ihre Eigenschaften klassifiziert. Die Konzeption des neuen Menüs basiert auf den Erfahrungen und den Anforderungen einer VR-Anwendung zur Visualisierung seismischer Daten. Es werden Lösungsvorschläge für grundlegende Herausforderungen der 3D-Menü-Interaktion präsentiert: Ein optimiertes Menü-Layout und eine an die Strahlinteraktion angepasste Bedienung verbessern die Effizienz und die Fehlerrate. Regelsysteme für die Positionierung und Skalierung optimieren situationsabhängig die Erreichbarkeit des Menüs, minimieren Verdeckungsprobleme und gewährleisten die Lesbarkeit der Menübeschriftung. In Benutzerstudien konnte die Wirksamkeit der Verbesserungen im Vergleich zu anderen Menüformen bestätigt werden.

## **Abstract**

This diploma thesis describes the concept, development and implementation of a 3D menu system for immersive virtual reality applications. In a first step the characteristics of existing 2D and 3D menu solutions are examined and classified. Based on the experience made with a VR application for visualizing seismic data this work develops a conceptual design for a novel menu system. We propose solutions for some fundamental challenges of 3D menu interaction: An optimized menu layout improves performance and error rate for pick ray based interaction. A rule-based and situation-dependent system for menu positioning and scaling enhances reachability, reduces occlusion, and allows for readability of 3D text and menu labelling. These improvements, in comparison to other menu forms, have been confirmed in user studies.





# Inhaltsverzeichnis

<b>Diplomthemenblatt</b>	<b>iii</b>
<b>Danksagung</b>	<b>v</b>
<b>Kurzdarstellung/Abstract</b>	<b>vii</b>
<b>Inhaltsverzeichnis</b>	<b>ix</b>
<b>Abbildungsverzeichnis</b>	<b>xiii</b>
<b>Tabellenverzeichnis</b>	<b>xv</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Ziel . . . . .	2
1.3 Aufbau . . . . .	2
<b>2 Virtuelle Umgebungen</b>	<b>5</b>
2.1 Displays . . . . .	5
2.2 Tracking . . . . .	8
2.3 Eingabegeräte . . . . .	9
2.4 Interaktionstechniken . . . . .	11
2.5 VR-Software-Frameworks . . . . .	12
<b>3 Steuerung mit Menüs und Widgets</b>	<b>13</b>
3.1 Systemsteuerung . . . . .	13
3.2 Widgets . . . . .	14
3.3 Menüs . . . . .	14
3.4 Systematische Klassifizierung . . . . .	15
3.4.1 Widget-Eigenschaften . . . . .	16
3.4.2 Container-Widgets . . . . .	17
3.4.3 Menü-Eigenschaften . . . . .	19
3.5 Zusammenfassung . . . . .	22
<b>4 Dimensionalität von Menüs</b>	<b>23</b>
4.1 Fitts' Law . . . . .	23
4.2 Interaktion mit Steuerelementen . . . . .	24
4.3 Menüs mit einem Freiheitsgrad . . . . .	24
4.3.1 Listen-Menü . . . . .	25

4.3.2	Ring-Menü . . . . .	27
4.4	Menüs mit zwei Freiheitsgraden . . . . .	29
4.4.1	Portierung von 2D-Menüs in VR . . . . .	29
4.4.2	Physikalische Hilfsmittel für die Menüdarstellung . . . . .	29
4.4.3	Kreis-Menü . . . . .	31
4.5	Menüs mit drei Freiheitsgraden . . . . .	32
<b>5</b>	<b>VR-Menüs im Expertenkontext</b>	<b>35</b>
5.1	Anwendungsszenario . . . . .	35
5.2	Der VRGeo-Demonstrator . . . . .	36
5.3	Kernfunktionalität . . . . .	38
5.4	Menü-Implementierungen auf PDA und Tablet-PC . . . . .	38
5.4.1	PDA . . . . .	38
5.4.2	Erfahrungen mit Menüs auf einem PDA . . . . .	39
5.4.3	Tablet-PC . . . . .	40
5.4.4	Erfahrungen mit Menüs auf einem Tablet-PC . . . . .	40
5.4.5	Immersionsverlust . . . . .	41
5.5	Anforderungen an das Menüsystem . . . . .	43
<b>6</b>	<b>Entwurf eines anforderungsgerechten 3D-Menüsystems</b>	<b>47</b>
6.1	Containerform . . . . .	47
6.1.1	Das „Context-Control“-Konzept . . . . .	48
6.1.2	Interaktion mit dem Container . . . . .	49
6.2	Menüelemente . . . . .	49
6.2.1	Blockzentriertes Layout . . . . .	49
6.2.2	Darstellung der Widgets . . . . .	50
6.2.3	Interaktion mit den Widgets . . . . .	50
6.3	Verdeckungsprobleme . . . . .	52
6.3.1	Transparenz . . . . .	52
6.3.2	Regelbasierte, aktive Positionierung . . . . .	53
6.3.3	Hierarchienavigation . . . . .	55
6.4	Visuelles Feedback . . . . .	55
6.5	Farbgebung . . . . .	56
6.6	Zusammenfassung der Menüeigenschaften . . . . .	56
<b>7</b>	<b>Implementierung des 3D-Menüsystems</b>	<b>61</b>
7.1	Das VR-Framework Avango . . . . .	61
7.1.1	Nodes und Fields . . . . .	61
7.1.2	Scripting . . . . .	62
7.1.3	Tools und Dragger . . . . .	62
7.2	Aufbau des Menüs . . . . .	64
7.3	Klassenstruktur der Menü-Software . . . . .	65
7.4	Entfernungsunabhängige Slider-Interaktion . . . . .	67
7.5	Regelbasierte Positionierung . . . . .	68
7.5.1	Strahlentest . . . . .	69

7.5.2	Skalierung . . . . .	70
7.5.3	Entfernungsabhängige Größenanpassung . . . . .	70
7.5.4	Lesbarkeit von Schrift . . . . .	71
7.5.5	Positionsbegrenzung an den Screengrenzen . . . . .	73
7.5.6	Ablauf der Menüpositionierung . . . . .	74
<b>8</b>	<b>Evaluation: Performancevergleich der Slider-Interaktion</b>	<b>75</b>
8.1	Methode . . . . .	77
8.1.1	Materialien . . . . .	78
8.1.2	Stichprobe . . . . .	79
8.1.3	Durchführung . . . . .	79
8.2	Ergebnisse . . . . .	81
8.2.1	Performance . . . . .	81
8.2.2	Fehlerrate . . . . .	82
8.2.3	Bewertung und Akzeptanz . . . . .	84
8.2.4	Beobachtungen, Kommentare und Kritik . . . . .	85
8.3	Zusammenfassung und Diskussion . . . . .	86
8.4	3D-Menüs im VRGeo-Demonstrator . . . . .	87
<b>9</b>	<b>Zusammenfassung und Ausblick</b>	<b>91</b>
	<b>Literaturverzeichnis</b>	<b>93</b>
<b>A</b>	<b>Anhang</b>	<b>101</b>
	Benutzerstudie: Fragebogen . . . . .	103
	Benutzerstudie: Auswertung des Fragebogens . . . . .	107
	Benutzerstudie: Beobachtungen, Kommentare und Vorschläge . . . . .	111
	<b>Erklärung</b>	<b>113</b>
	<b>Geheimhaltungsvereinbarung</b>	<b>115</b>
	<b>Thesen</b>	<b>117</b>
	<b>Vereinbarung über die Verwertungsrechte</b>	<b>119</b>
	<b>CD-Rom mit Originalmaterial</b>	<b>123</b>



# Abbildungsverzeichnis

2.1	Displays für VR-Systeme . . . . .	7
2.2	VR-Interaktionsgeräte . . . . .	10
4.1	1-DOF Listen-Menüs . . . . .	26
4.2	1-DOF Ring-Menüs . . . . .	28
4.3	2-DOF Menüs . . . . .	30
4.4	2-DOF Kreis-Menüs . . . . .	32
4.5	3-DOF Menü . . . . .	33
5.1	Der VRGeo-Demonstrator . . . . .	37
5.2	Menüs auf PDA und Tablet-PC . . . . .	39
6.1	Context-Controls im VRGeo-Demonstrator . . . . .	48
6.2	Texteingabe-Widgets im 3D-Menü . . . . .	51
6.3	3D-Menü mit verschiedenen Widgets und Transparenzen . . . . .	52
6.4	Hierarchienavigation durch Menü-Rotation . . . . .	55
7.1	Schematischer Aufbau des 3D-Menüs . . . . .	64
7.2	Z-Fighting und Sortierung von transparenten Objekten . . . . .	65
7.3	Klassendiagramm der 3D-Menü-Software . . . . .	66
7.4	Slider-Interaktion . . . . .	67
7.5	Menüpositionierung in maximaler Entfernung vor dem angeklickten Objekt . . . . .	69
7.6	Strahlbasierte Menüpositionierung vor störenden Objekten . . . . .	70
7.7	Entfernungsabhängige Skalierung . . . . .	71
7.8	Lesbarkeit von Text . . . . .	72
8.1	Screenshots von der Benutzerstudie . . . . .	80
8.2	Interaktionszeiten in der Benutzerstudie . . . . .	82



# Tabellenverzeichnis

5.1	Objekte, Funktionen und Interaktionsarten im VRGeo-Demonstrator . . . . .	36
5.2	Mögliche Gründe für Immersionsverlust . . . . .	42
8.1	Erfolgsquote der Slider-Interaktion in der Benutzerstudie . . . . .	83





# 1 Einleitung

Die Steuerung komplexer Software über Menüsysteme ist bei Desktop-Anwendungen weit verbreitet. Seit der Einführung von grafischen Benutzeroberflächen mit Mausbedienung haben sich Standards etabliert, die jedem Anwender geläufig sind. In Virtuellen Umgebungen treten mit der Verfügbarkeit der dritten Dimension neue Herausforderungen auf, bei denen die üblichen 2D-Konzepte nicht ausreichen.

Durch die räumliche Anordnung der Inhalte und Bedienelemente ergeben sich einerseits Verdeckungs- und Erreichbarkeitsprobleme, wenn störende Objekte dem Benutzer die Sicht versperren. Andererseits wird die Interaktion durch die zusätzlich verfügbaren Translations- und Rotationsfreiheitsgrade des Eingabegerätes deutlich schwieriger. Anstatt eine Maus auf der Tischoberfläche schnell und präzise zu positionieren, arbeitet der Benutzer meist freihändig in einem virtuellen Raum und interagiert teils auf große Distanz mit virtuellen Objekten. Weiterhin hängt die Erkennbarkeit dieser Objekte – insbesondere die Lesbarkeit von Text – und die Interaktion nicht nur von der Größe der Objekte ab, sondern zusätzlich von der Entfernung, dem Betrachtungswinkel, der Darstellungsqualität sowie der Bildschirmauflösung des verwendeten VR-Displays.

## 1.1 Motivation

Viele Anwendungen der Virtuellen Realität (*VR*) werden für einen sehr speziellen Anwendungsfall entwickelt, bieten nur einen eingeschränkten Funktionsumfang und werden von speziell geschultem Personal bedient. Die Steuerung dieser Anwendungen wird daher nicht selten auf spezielle Bedürfnisse zugeschnitten. Die meisten existierenden Konzepte eignen sich nicht als generelle Lösung für die Bedienung von VR-Applikationen. Dies liegt einerseits daran, dass für die Interaktion in Virtuellen Umgebungen eine Vielzahl verschiedener Eingabegeräte und Displays zur Verfügung steht, auf die die jeweiligen Menülösungen speziell optimiert sind. Andererseits werden besondere Herausforderungen wie z.B. Verdeckung, Positionierung, Textlesbarkeit oder die Ungenauigkeit der Bedienung oft nur angesprochen, aber nicht gelöst.

Die vorliegende Diplomarbeit wird im Expertenkontext des VRGeo-Projektes (*Virtual Reality for the Geosciences*) am Fraunhofer Institut für Intelligente Analyse- und Informationssysteme-

me (IAIS) angefertigt. Das Projekt hat zum Ziel, die Nutzbarkeit der Virtuellen Realität im Kontext der Geowissenschaften zu erforschen und zu verbessern. Ein Bestandteil des Projektes ist die Entwicklung eines Softwareprototypen für die Darstellung und Exploration seismischer Daten in Virtuellen Umgebungen [VRGeo].

Um die genannten Probleme von VR-Menüs zu umgehen, wurde die Steuerung dieses Demonstrators auf externe Geräte ausgelagert. Der Benutzer hat über einen PDA oder Tablet-PC Zugriff auf die Menüs und damit auf alle Funktionen der Software. Nach den Erfahrungen mit dieser Menüform sollen die Menüs zur Programmsteuerung nun in die VR-Szene des Demonstrators integriert werden. Dafür ist es notwendig Lösungsvorschläge für die bekannten Herausforderungen zu entwickeln und zu implementieren, sowie die Nutzbarkeit der neuen 3D-Menüs im VRGeo-Demonstrator zu überprüfen.

### 1.2 Ziel

In der vorliegenden Diplomarbeit wird ein 3D-Menü für die Steuerung von immersiven VR-Anwendungen entwickelt und dabei, ausgehend von den Erfahrungen und Anforderungen im VRGeo-Projekt, auf die Besonderheiten der Interaktion in Virtuellen Umgebungen eingegangen.

Im Fokus stehen die Entwicklung eines Menükonzeptes, mit dem sowohl Einsteiger als auch erfahrene Anwender eine komplexe VR-Applikation bedienen können sowie die Ausarbeitung von Lösungsvorschlägen für VR-spezifische Probleme, z.B. eine sinnvolle Positionierung der Menüs, Vermeidung von Verdeckung und Sicherstellen der Textlesbarkeit.

Zur Evaluation der entwickelten Konzepte wird eine Benutzerstudie durchgeführt, in der das neu entwickelte 3D-Menü mit den im VRGeo-Projekt bisher verwendeten PDA- und Tablet-PC-Menüs verglichen wird. Die Integration der 3D-Menüs in den VRGeo-Demonstrator ermöglicht die Erprobung der Menüs in einem spezifischen Anwendungsfall.

### 1.3 Aufbau

In Kapitel 2 wird zunächst eine Einführung in die Systeme der Virtuellen Realität gegeben und die benötigten Grundlagen für die Interaktion in Virtuellen Umgebungen erläutert.

Anschließend werden in Kapitel 3 die Grundlagen zur indirekten Steuerung von Applikationen vorgestellt. Dabei werden die Begriffe Menü, Widget und Container definiert und ihre unterschiedlichen Ausprägungsformen und Eigenschaften systematisch klassifiziert.

Kapitel 4 beschreibt die verschiedenen Arten der Menüinteraktion und geht auf wichtige Aspekte wie Effizienz und Dimensionalität ein. Exemplarisch werden eine Reihe unterschiedlicher Menüformen nach der Anzahl ihrer Interaktionsdimensionen gegliedert und vorgestellt.

Kapitel 5 erläutert den Expertenkontext, der den praktischen Rahmen der vorliegenden Diplomarbeit definiert. Neben der Vorstellung des VRGeo-Projektes wird auch der VRGeo-Demonstrator beschrieben, eine VR-Anwendung zur interaktiven Exploration seismischer Datensätze. Dabei wird aus der Erfahrung der bisherigen Applikationssteuerung mittels PDA und Tablet-PC das Anforderungsprofil für das neue 3D-Menü entwickelt.

In Kapitel 6 wird auf Grundlage der genannten Anforderungen das neue „Context-Control“-Konzept entworfen. Problematische Aspekte der Menü-Benutzung in VR-Anwendungen werden angesprochen und Lösungsvorschläge unterbreitet. Im Entwurf wird außerdem auf die Interaktion mit dem Menü eingegangen sowie verschiedene Gestaltungsaspekte vorgestellt.

Kapitel 7 beschreibt die Implementierung des entworfenen Menüs. Neben einer kurzen Vorstellung des verwendeten VR-Frameworks „Avango“ wird der Aufbau der Menü-Software erläutert sowie eine spezielle Interaktionstechnik für die Bedienung eines VR-Schiebereglers vorgestellt und Algorithmen zur regelbasierten Positionierung und Skalierung des Menüs beschrieben.

In einer Benutzerstudie wird die Menüinteraktion des neu entwickelten 3D-Menüs mit den PDA- und Tablet-PC-Menüs sowie mit einer Variation des neuen Menüs verglichen. In Kapitel 8 wird die Evaluation beschrieben, die Ergebnisse vorgestellt und diese diskutiert. Im Rahmen des VRGeo-Projektes hatten Experten aus verschiedenen Bereichen der Öl- und Gasexploration die Möglichkeit die 3D-Menüs in der Demonstrator-Software zu testen. Die Rückmeldungen werden ebenfalls in diesem Kapitel präsentiert.

Zum Abschluss der Arbeit werden in Kapitel 9 die wesentlichen Ergebnisse zusammengefasst.



## 2 Virtuelle Umgebungen

Die in dieser Arbeit vorgestellten 3D-Menüs basieren auf etablierten Techniken, die in Systemen der Virtuellen Realität (VR-Systeme) zum Einsatz kommen. Das vorliegende Kapitel gibt daher zunächst einen Überblick über den grundlegenden Aufbau von VR-Systemen, bevor auf die einzelnen Komponenten dieser Systeme und auf verschiedene Methoden der Interaktion mit ihnen eingegangen wird.

Voraussetzung für die Wahrnehmung einer Virtuellen Realität ist Immersion, das sensorische und mentale Gefühl des Eintauchens in eine virtuelle Welt [Sherman u. Craig 2003]. Eine künstliche, dreidimensionale Welt muss dem Betrachter zunächst visuell zugänglich gemacht, d.h. aus der Perspektive des Nutzers dargestellt werden. Um den Eindruck der Immersion aufrecht zu erhalten, reagiert das System auf Eingaben des Nutzers mit sensorischen, zumeist visuellen, Antworten. Beispielsweise reagiert die Betrachtungsperspektive der virtuellen Welt auf die Kopf- und Körperbewegungen des Betrachters. Zusätzlich soll dem Benutzer der Eindruck vermittelt werden, er könne direkt oder indirekt mit der Welt oder einzelnen Elementen darin interagieren. Der Aufbau von VR-Systemen folgt üblicherweise den durch diese Grundelemente definierten Ansprüchen. Die visuelle Darstellung wird über Display- oder Projektionssysteme realisiert, die mittels Stereoskopie einen räumlichen Tiefeneindruck vermitteln können. Zur Bestimmung der Position und Orientierung des Nutzers und eines in der Hand gehaltenen Eingabegerätes werden *Tracking-Systeme* verwendet, deren Daten schließlich in einem VR-Software-Framework zur Beschreibung und Steuerung der virtuellen Welt sowie zur Berechnung der 3D-Grafik herangezogen werden. Optional können VR-Systeme auch über akustisches, haptisches oder olfaktorisches Feedback verfügen [Bowman et al. 2005].

### 2.1 Displays

Um beim Betrachter den Eindruck von räumlicher Tiefe entstehen zu lassen, müssen Display-Systeme für Virtuelle Realität in der Lage sein, unterschiedliche Bilder für das linke und das rechte Auge des Betrachters darzustellen. Die Separation der stereoskopischen Bilder geschieht z.B. bei einem *Head-Mounted Display (HMD)* durch Verwendung zweier getrennter *LCD-Panels*, die in einen Helm oder eine Brille integriert jeweils ein Bild für das entsprechende

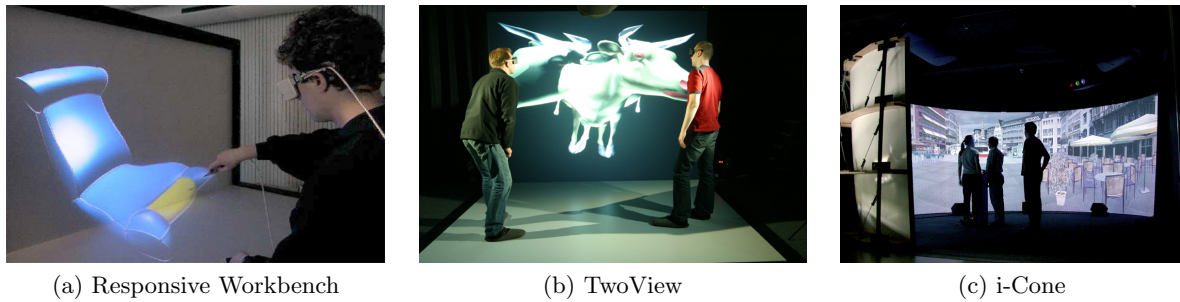
Auge darstellen (Abbildung 2.2a auf Seite 10). Der Betrachter wird so von seiner Außenwelt vollständig abgeschirmt und kann seinen eigenen Körper, z.B. seine Hand, nicht im VR-Szenario sehen. Alternativ werden vor dem Auge halbdurchlässige Spiegel platziert, so dass der Benutzer virtuelle Projektionen zusätzlich zu seiner realen Umgebung wahrnimmt. Der Anwender kann so den Informationsgehalt seiner Umgebung erweitern. Dieses Verfahren bezeichnet man als *Augmented Reality* oder *AR* [Sherman u. Craig 2003].

Soll sich der Betrachter frei vor oder in einem Display-System bewegen können, kommen projektionsbasierte Displaysysteme, beispielsweise eine *PowerWall* [Fakespace], *Responsive Workbench* (Abbildung 2.1a) [Krüger u. Fröhlich 1994; Krüger et al. 1995] oder eine *Cave* [Cruz-Neira et al. 1993] zum Einsatz. Diese verfügen nicht über zwei separate Bildschirme für die Darstellung der Bilder für das linke und rechte Auge, sondern trennen die Ansichten über spezielle Filtertechniken. Dabei unterscheidet man zwischen *Passiv-Stereo-* und *Aktiv-Stereo-*Verfahren. Bei den passiven Verfahren werden sowohl projektorseitig wie auch in der *3D-Brille* des Betrachters spezielle Filter eingesetzt. Die Polarisationsfiltertechnik nutzt für die Kanaltrennung die Eigenschaften von linear oder zirkular polarisiertem Licht, wofür spezielle polarisationserhaltende Leinwände bzw. Projektionsflächen benötigt werden. Die Interferenzfiltertechnik [Infitec] nutzt hingegen kammartig verschachtelte Bandpassfilter.

Zu den aktiven Verfahren zählt die sog. *Shutter-*Technik. Hierbei stellt das Display-System die Bildkanäle in schneller Folge abwechselnd dar. Die *Shutter-Brille* des Betrachters verdunkelt synchron zum Display abwechselnd das rechte und linke Auge, so dass nur das für das jeweilige Auge bestimmte Bild sichtbar wird. Voraussetzung für die Shuttertechnik ist eine möglichst hohe Bildwiederholrate, um ein Flackern des Bildes zu vermeiden. Bei der aktiven Interferenzfiltertechnik werden nicht, wie bei der passiven Variante, zwei Projektoren mit getrennten Filtern verwendet, sondern die Teilperspektiven werden abwechselnd von einem Projektor dargestellt. Synchron zu den Bildern wird im Projektor der Filter umgeschaltet [Barco Active Infitec+].

Um die Perspektive des 3D-Bildes korrekt berechnen zu können, muss mittels *Head-Tracking* die Position der Augen und die Blickrichtung des Betrachters erfasst werden. Dafür genügt es, die exakten Koordinaten der 3D-Brille über ein *Tracking-*Verfahren (siehe 2.2) zu bestimmen. Die meisten Display-Systeme können lediglich ein Stereo-Bild, also zwei Bildkanäle, darstellen, so dass nur ein Betrachter die perspektivisch korrekte Darstellung sehen kann. Alle übrigen Benutzer sehen ein mitunter stark verzerrtes Bild.

Nur wenige Mehrbenutzersysteme sind in der Lage, mehrere stereoskopische Ansichten zur gleichen Zeit darzustellen. Das *TwoView-*Display (Abbildung 2.1b) [Riege et al. 2006] kann über eine Kombination von Shutter- und Polarisationsfiltertechnik zwei Betrachter mit einer jeweils korrekten perspektivischen Stereo-Ansicht versorgen. Dafür werden zwei



**Abbildung 2.1:** *Displays für VR-Systeme* — (a) Zweiseitige Responsive Workbench (entnommen aus [Wesche 2004a]) (b) TwoView-Display mit zwei Perspektiven einer VR-Szene (c) 240° i-Cone Panorama-Display

Projektoren eingesetzt, die jeweils ein Aktiv-Stereo-Bild per Shutter-Technik erzeugen. Die Ansichten der zwei Betrachter werden mit zusätzlichen zirkularen Polarisationsfiltern vor den Projektoren und an den Shutter-Brillen getrennt. Um den Helligkeitsverlust zu minimieren, werden die Brillen nicht mit einem kompletten Polarisationsfilter ausgestattet, sondern nutzen die linear-polarisierenden Eigenschaften der Shutter-LCDs zusammen mit einem  $\lambda/4$ -Filter als Zirkularpolarisator. Bei der *Two-User Responsive Workbench* [Agrawala et al. 1997] zeigt ein Projektor insgesamt vier Ansichten für zwei Betrachter in schneller Folge. Modifizierte Shutter-Brillen geben die Sicht nacheinander für jeweils ein Auge frei. Das *Multi-Viewer Stereo Display* [Fröhlich et al. 2005] dagegen kann Bildpaare für bis zu vier Benutzer im Zeitmultiplex-Verfahren anzeigen. Die Trennung der Teilperspektiven für linkes und rechtes Auge erfolgt über Polarisationsfilter. Pro Benutzer sind zwei Projektoren mit entsprechenden Filtern notwendig. Eine rotierende Shutter-Scheibe lässt das Licht reihum von jeweils einem Projektorpaar passieren; die Projektoren der übrigen Betrachter sind in dieser Zeit verdunkelt. Modifizierte Shutter-Brillen werden mit der Scheibe synchronisiert und erlauben dem jeweils zugehörigen Nutzer die Sicht auf die Projektionsfläche. Auch das oben beschriebene Wellenlängenmultiplex-Verfahren von Infitec kann über schmalbandigere Filter für zwei und mehr Benutzer konfiguriert werden.

Gruppen-Displays wie die panoramaförmige *i-Cone* (Abbildung 2.1c) [Simon u. Göbel 2002] verzichten hingegen auf die Darstellung unterschiedlicher Perspektiven für die einzelnen Nutzer. Die Perspektive wird für einen feststehenden, fiktiven Nutzer berechnet, der genau im Zentrum des zylindrischen Displays steht. Dadurch ist das Bild für alle Nutzer, die nicht exakt im Zentrum stehen, leicht verzerrt. Für weit entfernte Objekte sind diese Verzerrungen vernachlässigbar und kaum wahrzunehmen, jedoch werden die Verzerrungen stärker, je näher ein virtuelles Objekt dargestellt wird. Interaktive 3D-Elemente, wie z.B. der Selektionsstrahl

aus einem Interaktionsgerät, werden von den einzelnen Teilnehmern dadurch am falschen Ort wahrgenommen. Mittels *Multi-Viewpoint Images* [Simon u. Scholz 2005] können solche Elemente jedoch perspektivisch korrigiert in die Szene gezeichnet werden, so dass jeder Nutzer seine eigenen Interaktionselemente korrekt sieht. Um die stereoskopische Tiefenwahrnehmung über die volle 240°-Ansicht zu erhalten, muss das Bild zusätzlich in Teilansichten unterteilt berechnet werden [Simon et al. 2004]. Durch diese Techniken ist es möglich, dass trotz fester Perspektive mehrere Personen gleichzeitig mit der 3D-Szene interagieren können.

## 2.2 Tracking

Mit Hilfe von Tracking-Systemen werden Position und Lage realer Objekte im Raum ermittelt. Man kann die verschiedenen Systeme anhand der zur Verfügung stehenden Freiheitsgrade und der eingesetzten Technologie klassifizieren. Systeme mit nur drei Freiheitsgraden (*3-DOF*) können ausschließlich die Position *oder* die Rotation eines Objektes messen. Für VR-Anwendungen haben sich aber vor allem optische und elektromagnetische Tracking-Systeme durchgesetzt, die mit sechs Freiheitsgraden gleichzeitig sowohl die Position als auch die Rotation eines Objektes bestimmen können (*6-DOF*).

Bei elektromagnetischem Tracking (z.B. [Polhemus Fastrack], [Ascension Flock Of Birds]) erzeugt ein Sender (*Transmitter*) ein schwaches magnetisches Feld. In diesem können mehrere, deutlich kleinere, Empfänger (*Receiver*) positioniert und bewegt werden. Über die Messung elektromagnetisch induzierter Ströme kann der daran angeschlossene Tracking-Computer die Position und Lage des Empfängers relativ zum Sender berechnen.

Optische Tracking-Systeme (z.B. [A.R.T.]) benutzen üblicherweise mehrere Kameras, die mit Infrarotlicht angeleuchtete retroreflektive *Marker* verfolgen. Über Triangulation der erkannten Punkte in den Kamerabildern kann so die genaue Position des Markers berechnet werden. Werden nun mehrere Marker in einer festen Konstellation zu einem *Target* zusammengefügt, kann neben der genauen Position auch die Ausrichtung erkannt und das Target zugeordnet werden. Das Kamerasystem und die einzelnen Targets müssen vor der Benutzung einmalig kalibriert werden. Die verschiedenen Targets können dann für Head-Tracking an 3D-Brillen oder an Eingabegeräten für die 3D-Interaktion befestigt werden. Befestigt man eine entsprechend große Anzahl solcher Targets an geeigneten Körperstellen, kann damit auch *Body-Tracking* realisiert werden.

Optische Targets sind für die Erfassung von Hand- und Fingerbewegungen jedoch nur bedingt geeignet. Diese können aber über sog. Datenhandschuhe (*Data Gloves*) erfasst werden. Die Handschuhe sind, je nach Ausführung, mit einer unterschiedlichen Anzahl an Sensoren ausgestattet, um die Haltung der Hand und die Krümmung der einzelnen Fingergelenke zu



erfassen. Manche Modelle können über zusätzliche Tastflächen die Berührung der Fingerspitzen erkennen. Die gewonnenen Daten können zusammen mit dem Body-Tracking für *Motion-Capture*-Anwendungen oder zur Gestenerkennung im Zusammenhang mit Interaktionstechniken verwendet werden (siehe 2.3).

Eine Vielzahl weiterer Tracking-Verfahren kann für die Bestimmung von Position oder Rotation herangezogen werden. Zu diesen zählen mechanische Konstruktionen, Beschleunigungssensoren, die Messung von Erdmagnetfeld und Gravitation, optische Verfahren mit aktiven LED-Targets, Ultraschallsysteme und bildbasierte Verfahren, welche die Objektpositionen, z.B. die der Augen oder der Fingerspitze eines Benutzers, auch ohne Marker und Targets ermitteln können. Viele dieser Verfahren bieten nur drei Freiheitsgrade, sind für spezielle Anwendungen ausgelegt oder haben eine stark eingeschränkte Reichweite, beispielsweise für die Benutzung an einem Schreibtischarbeitsplatz. Eine Kombination verschiedener Technologien kann einen Mangel an Freiheitsgraden jedoch ausgleichen. So kann z.B. eine 3-DOF Positionserkennung mittels LED-Tracking durch eine Kombination der Messung von Erdmagnetfeld, Gravitation und Beschleunigungssensoren (wie z.B. beim Intersense InertiaCube [InterSense]) um eine 3-DOF Rotationserkennung zu einem vollwertigen 6-DOF-Tracking ergänzt werden.

## 2.3 Eingabegeräte

Eingabegeräte dienen dem Benutzer zur Interaktion mit dem VR-System. Sie bilden die Schnittstelle zwischen realen Objekten und der virtuellen Welt. Während viele Interaktionsgeräte für spezielle Anwendungen konzipiert sind, z.B. ein Lenkrad für einen Fahrsimulator, gibt es eine Reihe von Geräten, die eine generische Bedienung vieler Anwendungen ermöglichen [Sherman u. Craig 2003]. Dabei kann zwischen der Zahl der zur Verfügung stehenden Freiheitsgrade und der physikalischen Maßeinheit des Sensors unterschieden werden.

*Elastische* Interaktionsgeräte messen die Auslenkung eines an Federn aufgehängten Handgriffs im Verhältnis zu seiner Ruhestellung. Dagegen messen sog. *isometrische* Eingabegeräte die Kraft, die auf einen unbeweglichen Sensor einwirkt. Zur ersten Kategorie gehören z.B. Joystick oder Spacemouse [3Dconnexion], zur zweiten u.a. der Spaceball [3Dconnexion]. Die dritte Alternative sind sog. *isotonische* Geräte, die ohne Widerstand frei bewegt werden können, beispielsweise eine Computer-Maus oder der Fingerball [Zhai 1998]. Ähnlich wie bei den Tracking-Systemen gibt es Geräte für Positions- oder Rotationseingabe mit bis zu drei Freiheitsgraden und solche mit vollständiger 6-DOF-Erkennung. Zusätzlich zu den Freiheitsgraden enthalten viele Geräte weitere Sensoren wie z.B. Taster, Schalter, Schiebe- oder Drehregler. Die in dieser Arbeit beschriebenen Interaktionstechniken setzen frei bewegliche Eingabegeräte mit 6-DOF-Tracking voraus.



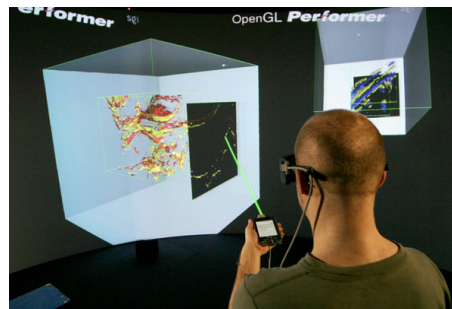
(a) HMD + Datenhandschuhe



(b) Wireless Stylus



(c) PDA



(d) Strahlinteraktion mit PDA

**Abbildung 2.2:** *VR-Interaktionsgeräte* — (a) Benutzer mit HMD und Datenhandschuh (entnommen aus [Bowman u. Wingrave 2001]) (b) Kabelloser Stylus mit Tastern, A.R.T.-Target und Stiftspitze zur Bedienung von Touch-Screens (c) PDA mit Darstellung von Menüs und A.R.T.-Target (d) Strahlinteraktion in einer VR-Szene mit PDA

Anwendungen, die mit einem Datenhandschuh (Abbildung 2.2a) bedient werden, werten neben der Handposition und Orientierung auch die Handhaltung aus. Für die Interaktion mit der VR-Welt werden verschiedene Gesten festgelegt, über die der Benutzer Kommandos an das System absetzen kann. *Stylus*-artige Interaktionsgeräte (Abbildung 2.2b) werden in der Hand gehalten und haben meist eine längliche Form, sowie ein oder mehrere Tasten auf der Oberfläche. Auch Geräte mit einer eigenen, komplexen Benutzerschnittstelle können als Interaktionsgerät genutzt werden: Ein *PDA* zeigt dem Benutzer kontextabhängige Informationen, Menüs zur Objektmanipulation oder Funktionen zur Systemsteuerung an. Die 2D-Benutzeroberfläche des PDA (Abbildung 5.2a) wird mit einem Stift auf dem berührungsempfindlichen *Touch-Screen* bedient. Das zusätzliche 6-DOF-Tracking des PDAs ermöglicht zusammen mit den eingebauten Knöpfen darüber hinaus die direkte Interaktion mit der 3D-Szene (Abbildung 2.2d) [Simon et al. 2005; Conrad 2005]. Einen ähnlichen Ansatz verfolgt die Benutzung eines *Tablet-PC* in Virtuellen Umgebungen. Dieser bietet im Vergleich zum PDA eine deutlich höhere Bildschirmauflösung für die Darstellung einer 2D-Benutzerschnittstelle. Für die 3D-Interaktion kommt ein modifizierter Stylus zum Einsatz, an dessen hinterem Ende eine

Stiftspitze integriert ist. So kann der Benutzer mit dem Stylus sowohl die 3D-Interaktion durchführen als auch den Touch-Screen des Tablet-PCs bedienen [Holtkämper et al. 2007].

## 2.4 Interaktionstechniken

Die Interaktion mit dem VR-System erlaubt dem Benutzer die Beeinflussung bestimmter Eigenschaften der virtuellen Welt oder der darin enthaltenen Objekte. Die Rohwerte des Interaktionsgerätes werden durch eine Transferfunktion auf geeignete Weise in die virtuelle Welt übertragen. In der Literatur finden sich eine Vielzahl verschiedener Interaktionstechniken, diese lassen sich jedoch in wenige grundlegende Aufgabenbereiche unterteilen [Herndon et al. 1994; Mine 1995]:

- *Navigation*: Der Benutzer bewegt sich bzw. die virtuelle Kamera in der Welt
- *Selektion*: Das Auswählen bestimmter Objekte
- *Manipulation*: Das Bewegen eines Objektes im Raum
- *Steuerung*: Die Benutzung von Menüs und anderen Elementen der VR-Benutzeroberfläche

Steuerungsaufgaben (siehe Kapitel 3 ab Seite 13) und auch weitere in der Literatur genannte Interaktionen sind im wesentlichen Spezialfälle von Selektion und Manipulation. Diese zwei Interaktionsaufgaben bilden die Grundlage der in dieser Arbeit vorgestellten 3D-Menü-Interaktionen. Die relevanten Techniken werden daher im Folgenden erläutert. Die Beschreibungen der Interaktionstechniken gehen von der Benutzung eines 6-DOF-Stylus mit einem Taster aus. Auf die Besonderheiten bei der Interaktion mit einem Datenhandschuh wird anschließend eingegangen.

[Lemoine et al. 2003] unterscheiden zwischen den Metaphern der *Virtual Hand* und des *Virtual Pointer*. Bei der *Simple Virtual Hand* wird die virtuelle Hand in Form eines *Cursors* an der Position des Interaktionsgerätes in der virtuellen Welt dargestellt. Schneidet der Cursor ein Objekt, kann der Benutzer dieses über einen Taster am Eingabegerät selektieren und bewegen. Die Lageänderungen des Interaktionsgerätes werden dabei direkt auf das virtuelle Objekt übertragen. Diese Interaktion fühlt sich durch die Analogie zur realen Welt sehr natürlich an, jedoch ist die Reichweite auf das direkte Umfeld des Benutzers beschränkt. Verschiedene Techniken erweitern den Interaktionsradius. Beispielsweise kann der Benutzer bei der *Go-Go*-Technik [Poupyrev et al. 1996] seinen virtuellen Arm interaktiv verlängern, um weit entfernte Objekte zu greifen.

Die Virtual-Pointer-Techniken basieren dagegen auf einer Strahl-Interaktion. Der 3D-Cursor hat die Form eines Laser-Strahls, welcher an der Position des Interaktionsgerätes entspringt. Die Selektion wird durch das Zeigen mit dem Strahl auf ein Objekt erreicht. Beim *Simple Dragging*

kann der Benutzer durch Betätigen eines Tasters das Objekt greifen. Der Strahlabschnitt zwischen Interaktionsgerät und Objekt dient dabei als virtueller Stock, an dessen Ende das Objekt befestigt ist. Auf diese Weise kann das Objekt bewegt und an der Zielposition abgesetzt werden. Der Rotationsmittelpunkt liegt, wie auch bei der Virtual Hand, in der Hand des Benutzers. Die *Homer*-Technik [Bowman u. Hodges 1997] dagegen verschiebt das Zentrum der Rotation in den Mittelpunkt des Objektes. In beiden Fällen ist es jedoch schwierig, das Objekt weit von sich weg zu schieben oder aus weiter Entfernung zu sich heran zu ziehen, da die Positionsänderungen direkt übertragen werden. Dieses Problem wird durch das *Scaled-Grab*-Verfahren gelöst [Simon et al. 2005]. Der Rotationsmittelpunkt befindet sich im Schnittpunkt von Strahl und Objekt (*Pivot-Punkt*). Die Reichweite der Hand wird um die Distanz zum Objekt hochskaliert. Auf diese Weise befinden sich alle Objekte, unabhängig von ihrer Entfernung, in der Reichweite einer Armlänge. Außerdem können die Objekte sowohl um den Benutzer rotiert werden als auch lokal um den Pivot-Punkt. Der Benutzer darf während der Interaktion jedoch seinen Standpunkt nicht verändern.

Bei der Benutzung eines Datenhandschuhs gelten die oben genannten Interaktionstechniken ebenfalls. Um Aktionen auszulösen formt der Benutzer festgelegte Gesten mit seiner Hand. Beispielsweise dient ein ausgestreckter Zeigefinger der Markierung von Objekten, während eine geschlossene Faust das Greifen und Festhalten eines Objektes symbolisiert. Durch Öffnen der Hand wird das Objekt an der aktuellen Position losgelassen. Bei Handschuhen mit Fingerkontakten können zudem bestimmte Fingerberührungen für die Interaktion genutzt werden.

## 2.5 VR-Software-Frameworks

Ein VR-Software-Framework bildet die Grundlage für die Entwicklung von interaktiven VR-Anwendungen. Es bietet dem Entwickler Zugriff auf alle relevanten Technologien und enthält Software-Schnittstellen für das Tracking-System, die Eingabegeräte, sowie für die logische Abbildung der virtuellen Welt in einem Szenengraphen und für die Darstellung der 3D-Grafik (*Rendering*). Des Weiteren enthält das Framework eine Reihe von hilfreichen Software-Bibliotheken und unterstützt idealerweise Scriptsprachen, die das zügige Entwickeln von VR-Software erlauben (*Rapid Application Development*).

Die 3D-Menüs in dieser Arbeit wurden mit dem *Avango*-VR-Framework entwickelt. Eine Beschreibung dieses Systems findet sich in Abschnitt 7.1 auf Seite 61.

## 3 Steuerung mit Menüs und Widgets

Im Kontrast zur direkten Interaktion mit der virtuellen Welt mittels Navigation, Selektion und Manipulation betrachtet dieses Kapitel die indirekte Interaktion über Menüs und andere Bedienelemente. Der Benutzer kann dadurch Arbeitsabläufe durchführen, die über direkte Interaktionen nur schwer zu realisieren sind [Mine 1995]. Die Benutzung der Gesamtheit der Bedienelemente und Menüs wird im weiteren Verlauf als *Steuerung* bezeichnet.

Die Steuerung kann als eine spezifische Kombination aus Manipulation und Selektion betrachtet werden: Ähnlich wie bei der direkten Interaktion mit Objekten werden auch Menüeinträge über Interaktionstechniken ausgewählt und anschließend aktiviert. Die Interaktion mit Menüs geht jedoch über diese Techniken hinaus und bietet viele Eigenarten, welche die Steuerung zu einer eigenen Interaktionsart machen [Kim et al. 2000].

Für das Verständnis der Menüinteraktion werden in diesem Kapitel zuerst die Grundbegriffe der Systemsteuerung, Bedienelemente und Menüs erläutert. Anschließend werden Kategoriensysteme zur Klassifizierung von Bedienelementen und Menüs vorgestellt. Diese Klassifizierung hilft bei der systematischen Beschreibung von Menüs und ihren Eigenschaften. Dadurch werden unterschiedliche Menüsysteme miteinander vergleichbar. Für die Konzeption neuer VR-Menüs stellt das System eine Vielzahl an Kriterien bereit und bietet eine Auswahl an möglichen Optionen.

### 3.1 Systemsteuerung

Nach Bowman et al. bezeichnet *Systemsteuerung* (*System-Control*) den Vorgang, bei dem der Benutzer Befehle an das System absetzt, um

1. eine bestimmte Funktion auszuführen,
2. den Interaktionsmodus zu ändern oder
3. den Systemzustand zu ändern.

Im ersten Fall gibt der Benutzer den Auftrag, einmalig eine vorgefertigte System-Funktion auszuführen, beispielsweise das Speichern der aktuellen Szene oder die Durchführung einer Berechnung, deren Ergebnis anschließend sichtbar wird. Die Punkte zwei und drei ändern die

aktuelle Szene nicht direkt, sondern bereiten das System auf zukünftige Eingaben des Benutzers vor. Das Auswählen eines Werkzeuges bestimmt z.B. die weitere Form der Interaktion, während die Selektion eines Objektes zu einer Kontext-Änderung führt. Dem System wird dadurch mitgeteilt, dass zukünftige Funktionen sich z.B. auf das zuletzt ausgewählte Objekt beziehen sollen [Bowman et al. 2005].

## 3.2 Widgets

Maschinen und Geräte werden über mechanische Elemente bedient, etwa durch Taster, Schalter, Schiebe- oder Drehregler. Der Benutzer bekommt Rückmeldungen des Systems über Anzeigesysteme wie Lampen oder Displays und über die Haptik. In Analogie zur realen Welt existieren eine Reihe von virtuellen Nachbildungen, um Funktionen zu aktivieren oder Eigenschaften von Objekten zu überprüfen und zu verändern [Bowman et al. 2005].

Im Englischen werden diese Elemente mit dem Sammelbegriff *Widget*<sup>1</sup> bezeichnet. Die alternative Bezeichnung *Control* und die deutschen Begriffe *Bedienelement* oder *Steuerelement* heben die Bedeutung besser hervor, wobei zwischen zwei Arten unterschieden wird: Widgets für die geometrische Manipulation und Widgets für die Anwendungs- und Systemsteuerung [Leiner et al. 1997]. Zur erstgenannten Sorte gehören z.B. *Anfasser*. Anfasser sind Bedienelemente, welche an einem selektierten Objekt dargestellt und mittels einer direkten Interaktionstechnik gegriffen und verändert werden können. Als Beispiel seien grafische Programme genannt, die nach der Auswahl eines Objektes kleine Rechtecke an den Begrenzungen darstellen (Anfasser oder *control point*). Diese Anfasser erlauben die direkte, geometrische Manipulation des ausgewählten Objektes.

Die vorliegende Diplomarbeit fokussiert jedoch ausschließlich auf Bedienelemente zur System- und Anwendungssteuerung, also Menüs, Schaltflächen oder Schieberegler, die in den folgenden Abschnitten vorgestellt werden.

## 3.3 Menüs

Unter einem *Menü* versteht man eine Liste von Einträgen zur Systemsteuerung, die vier Ausprägungsformen haben können:

1. Direktes Ausführen einer Funktion
2. Ändern der Interaktionsart oder des Systemzustands

---

<sup>1</sup>Das englische Wort *Widget* wird umgangssprachlich mit „Dingsbums“ übersetzt. Es ist ein Kunstwort aus *Window* und *Gadget*, was übersetzt „Technische Vorrichtung“ bedeutet. (Vgl. [LEO])

3. Aktivierung eines hierarchisch untergeordneten Menüs mit weiteren Einträgen
4. Öffnen einer Dialog-Instanz mit weiteren Eingabemöglichkeiten

Auf die Punkte eins und zwei wurde im Abschnitt 3.1 bereits eingegangen. Die in Punkt drei genannte hierarchische Ordnung wird genutzt, wenn die Anzahl der Menüeinträge einen sinnvollen Umfang des Menüs übersteigt, also zu Einbußen bei der Darstellung, Bedienbarkeit oder Übersichtlichkeit der Liste führt. In diesem Fall können thematisch oder funktional ähnliche Einträge in einem Untermenü zusammengefasst werden, dessen Titel einen verständlichen Oberbegriff zur enthaltenen Sammlung bildet. Darüber hinaus gibt es zur Strukturierung von Menüs spezielle Elemente, wie z.B. eine Trennlinie oder eine Informationszeile (*Etikett*, engl.: *label*). Die in Punkt vier bezeichnete Dialog-Instanz (z.B. ein Dialogfenster, siehe Abschnitt 3.4.2) dient dem Benutzer zur Information oder bietet die Möglichkeit weiterer Eingaben oder Einstellungen. In vielen Fällen ist darüber auch ein Abbruch der zuletzt gewählten Funktion möglich, bei dem das System in den vorherigen Zustand zurück versetzt wird.

Die einzelnen Menüeinträge enthalten meist eine beschreibende Textzeile oder ein *Icon* als symbolhafte Abbildung der Funktion, manchmal auch beides. Zusätzlich werden bei Bedarf Hinweise auf mögliche *Shortcuts*<sup>2</sup> oder ein Symbol zur Kennzeichnung eines Untermenüs angezeigt. Außerdem gibt es Einträge, die über ein Symbol (z.B. ein Häkchen) den Aktivierungszustand einer Funktion anzeigen. Steht eine Funktion im aktuellen Kontext des Programms nicht zur Verfügung, wird der Eintrag meist mit vermindertem Kontrast (*ausgegraut*) angezeigt und kann nicht angewählt werden.

### 3.4 Systematische Klassifizierung

In den vergangenen Jahren wurde eine Vielzahl unterschiedlicher Widgets und Menüs für VR-Systeme entwickelt. Dies ist unter anderem darin begründet, dass sich, im Gegensatz zu 2D-Desktop-Betriebssystemen, noch keine wiederverwendbaren Standardkomponenten durchgesetzt haben. Die Auswahl der teils grundverschiedenen VR-Frameworks, Displaysysteme, Eingabegeräte und Interaktionstechniken erschweren weiterhin die Standardisierung. Es existieren daher diverse Speziallösungen, die jeweils für bestimmte Anwendungsfälle entwickelt wurden. Um diese Lösungen dennoch miteinander vergleichen und beurteilen zu können, ist eine systematische Unterteilung in differenzierbare Kriterien notwendig. Dies ermöglicht die gezielte Betrachtung der verschiedenen Eigenschaften und macht die Systeme vergleichbar.

---

<sup>2</sup>dt.: „Abkürzung“; eine alternative Möglichkeit einen Menübefehl auszuführen; meist über einen zweiten Eingabe-Kanal, z.B. über die Tastatur oder ein Sprachkommando (multimodale Systemsteuerung)

#### 3.4.1 Widget-Eigenschaften

Widgets sind eine Verschmelzung von Erscheinungsform und Verhalten [Conner et al. 1992]. Wie bereits erwähnt, basieren virtuelle Bedienelemente auf der Idee ihrer mechanischen Vorbilder. So verwundert es kaum, dass in der hier vorgestellten Liste eine ganze Reihe solcher Nachbildungen auftauchen. Daneben gibt es aber auch einige Elemente, die keine mechanische Entsprechung haben. Sie profitieren von den dynamischen Eigenschaften einer grafischen Benutzerschnittstelle. Praktisch jedes grafische Element einer Benutzeroberfläche ist im weitesten Sinne ein Widget. Daher enthält die folgende Auflistung nur die für die Anwendungs- und Systemsteuerung relevanten Arten. Im Projekt CONTIGRA der TU-Dresden existiert eine umfassende Klassifikation von 3D-Widgets, deren Struktur in die Liste eingeflossen ist [Projekt CONTIGRA; CONTIGRA Widgets; Dachsel et al. 2005].

- *Information*: Diese Elemente dienen dem User ausschließlich zur Information; eine Interaktion mit ihnen ist nicht möglich. Beschriftung, Status-Information, *Tool Tips* oder *Icons* sind textuelle oder bildliche Informationen. Trennlinien dienen der übersichtlicheren Strukturierung. Fortschrittsbalken oder sog. *Spinner* informieren den Benutzer über Verlauf und Dauer einer länger andauernden Aktion.
- *Aktivierung*: Schaltflächen oder Taster (*Button*) setzen einen Befehl an das System bzw. an die Anwendung ab.
- *Zwei Zustände*: Damit sind binäre Schalter gemeint, die immer einen von zwei Zuständen einnehmen. Der aktuelle Zustand wird grafisch dargestellt. Es gibt eine Reihe von Darstellungsformen dieser Schalter: Druckschalter, Kippschalter, Schiebeschalter, Auswahlkästchen (*Checkbox*).
- *Multiple Zustände*: Dabei handelt es sich um Schalter mit mehr als zwei Zuständen: Auswahllisten oder *Radiobuttons* bestehen aus einer Gruppe binärer Schalter, von denen jeweils nur einer aktiv ist. Drehschalter oder Schiebeschalter sind dagegen in grafischen Oberflächen eher selten zu finden.
- *Skalare Wertgeber*: Diese Elemente repräsentieren Zahlenwerte auf einer diskreten oder kontinuierlichen Zahlenachse. Darstellungen als Schieberegler, Drehknopf oder Textanzeige mit  $\pm$ -Tastern sind üblich. Seltener sind kontinuierliche Wippschalter. Ein Rollbalken bzw. eine Bildlaufleiste (*Scroll-Bar*) repräsentiert einen Abschnitt auf der Achse.
- *Mehrdimensionale Wertgeber*: 2D-Regler oder 3D-Regler sind Flächen oder Raumabschnitte, in denen eine Markierung positioniert werden kann. Diese beschreibt den aktuellen Wert auf mehreren Achsen. 2D-Regler kommen z.B. bei Spezial-Bedienelementen wie Farbwählern zum Einsatz.



- *Spezielle Bedienelemente*: Die Elemente dieser Gruppe beinhalten spezielle Lösungen für komplexe Interaktionen. Sie sind oftmals zusammengesetzt aus den vorangegangenen generischen Elementen, können aber auch eigene Darstellungs- und Interaktionsformen enthalten. Beispiele sind Text-Eingabefelder, die über spezielle Texteingabemöglichkeiten editiert werden. Meist steht dafür eine Tastatur zur Verfügung, ansonsten müssen spezielle Interaktionsformen für die Texteingabe hinzukommen. Solche Texteingabe-Widgets zählen ebenfalls zu den Spezialelementen. Andere Beispiele sind Listenauswahl (kann mehrere aktive binäre Schalter enthalten), die sog. ComboBox (eine Kombination aus Textfeld und Drop-Down-Menü mit einer Auswahlliste), Registerkarten mit Reitern für die Unterteilung komplexer Dialoge, Farbwähler, Dateiauswahl-Dialoge oder eine Leinwand (*canvas*) zur freien grafischen Eingabe, wie z.B. für Zeichnungen.
- *Container*: Die Bedienelemente werden in sog. Containern eingebunden und dargestellt. Diese Container können z.B. Fenster, Menüleisten, Menüs, Symbolleisten oder Paletten sein. Der folgende Abschnitt geht auf diese speziellen Widgets ein.

### 3.4.2 Container-Widgets

Container-Widgets dienen der Zusammenfassung und Anordnung mehrerer Bedienelemente [Dachselt u. Hinz 2005]. Die darin enthaltenen Elemente sind meist thematisch gruppiert und beziehen sich auf den gleichen Kontext in der Applikation. Container helfen aber nicht nur bei der Strukturierung der Benutzeroberfläche, sondern dienen auch dazu, Widget-Gruppen gemeinsam ein- und auszublenden und zu platzieren. Die hier genannten Beispiele sind größtenteils von 2D-Benutzeroberflächen bekannt, können jedoch auch in gleicher oder abgewandelter Form in immersiven VR-Anwendungen zum Einsatz kommen.

- *Fenster*: Fenster dienen der logischen und visuellen Unterteilung oder Abgrenzung einer Anwendung. Sie sind die vielseitigsten Container und können praktisch jedes andere Widget oder Containerelement enthalten. Ein Fenster enthält üblicherweise den Arbeitsbereich des Benutzers in seinem spezifischen Kontext und repräsentiert z.B. ein Dokument oder einen Datensatz. Dagegen dienen sog. Dialogfenster der Information, warnen den Benutzer oder fragen Eigenschaften, Eingaben und Entscheidungen ab.
- *Menüzeile*: Eine Menüzeile bietet Platz für eine Reihe thematisch geordneter Menüs. Diese ist meist dauerhaft sichtbar, wird an einer festen Position innerhalb der Welt oder des Fensters angezeigt und dient als zentrale Zugangsstelle für die Steuerung von System und Anwendung. In der Menüzeile werden die Titelnamen der Menüs oder aussagekräftige Icons dargestellt.

- *Drop-Down-Menü*: Wird ein Eintrag aus der Menüzeile angewählt, klappt ein meist rechteckiges Menü unterhalb des Eintrags auf. Nach Anwahl eines Eintrags wird das Menü entfernt; die Menüzeile bleibt weiterhin sichtbar.
- *Pop-Up Menü*: Ein Pop-Up Menü erscheint in unmittelbarer Nähe des angewählten Objektes und ist nur für die Dauer seiner Benutzung sichtbar: Der Benutzer aktiviert das Menü und wählt daraufhin einen Eintrag aus. Unmittelbar nach Anwahl des Eintrags wird das Pop-Up-Menü automatisch geschlossen.
- *Toolbar*: Die Toolbar (*Werkzeuggestreife* oder *Symbolleiste*) besteht meist aus einer Reihe von Icons, manchmal gemischt mit anderen Bedienelementen. Sie ist an zentraler Stelle jederzeit sichtbar und bietet dem Nutzer die Möglichkeit, Eigenschaften und Einstellungen schnell zu überblicken und zu verändern. Die Auswahl von symbolisierten Werkzeugen erlaubt das schnelle Umschalten der Interaktionsmodi.
- *Toolbox/Palette*: Ähnlich der Toolbar handelt es sich bei der Toolbox (*Werkzeugkoffer*) um eine Ansammlung von Icons oder anderen Elementen zur Aktivierung von Interaktionsmodi und zum Ändern von Einstellungen. Diese werden jedoch nicht wie die Werkzeuggestreife an zentraler Stelle angezeigt, sondern können, in Analogie zum Werkzeugkoffer, frei platziert werden. Die Toolbox wird oft als „schwebende Palette“ angezeigt: ein Fenster, das immer im Vordergrund ist und daher jederzeit zugreifbar bleibt. In manchen Programmen können mehrere Paletten „ineinandergesteckt“ werden. Die einzelnen Toolboxen erscheinen dann platzsparend als Registerkarten in einer Palette und können über die Reiter ausgewählt werden. Ein späteres Herauslösen ist möglich, so dass die entsprechende Toolbox wieder einzeln zur Verfügung steht.
- *Tear-Off-Menü*: Das Abreißmenü ist eine Kombination aus einem Drop-Down Menü und einer Toolbox. Das heruntergeklappte Menü kann dabei von der Menüzeile „abgerissen“ werden und bleibt danach dauerhaft als Toolbox sichtbar, bis der Benutzer diese manuell schließt. Dies hat den Vorteil, dass ein großer Teil der Funktionalität in platzsparenden Menüs untergebracht werden kann. Der Benutzer legt sich nur die Teile des Menüs dauerhaft in den Arbeitsbereich, die er für seine aktuellen Arbeitsschritte regelmäßig benötigt.
- *Alternative Containerformen*: Als Alternative zu den meist rechteckigen Containern können spezielle Formate mit anderen Grundformen gewählt werden, z.B. Scheiben oder Ringe, Zylinder, Würfel, Kugeln oder auch visuelle Metaphern wie Räume, bestimmte Objekte, oder Landschaften. Die Entscheidung für ein solches Format kann verschiedene Gründe haben: Die Form kann realen Objekten nachempfunden sein, das Verständnis in speziellen Lernumgebungen fördern, der Struktur der darzustellenden Daten entsprechen, oder die

Grundlage für neue Interaktionstechniken bilden. Zusätzlich kann die Präsentationsform das Erinnern von Position und Funktion unterstützen oder die Effizienz von Abläufen verbessern. Auch Marketing-Überlegungen führen zu optischen Alleinstellungsmerkmalen, z.B. bei Computerspielen oder zur Abhebung von der Konkurrenz.

### 3.4.3 Menü-Eigenschaften

Menüs können durch eine Reihe von Eigenschaften charakterisiert werden. Diese Eigenschaften beschreiben die technischen Aspekte eines Menüs und machen verschiedene Menüsysteme untereinander vergleichbar. Jacoby und Ellis haben ein Paradigmensystem eingeführt, das über eine Reihe von Fragen mit Antwortoptionen ein Menüsystem beschreibt. Diese Paradigmen waren sehr stark durch ihr vorgestelltes 3D-Menüsystem geprägt, welches mit HMD und Datenhandschuh bedient wurde [Jacoby u. Ellis 1992]. Dachzelt und Hübner haben diese Systematik aufgegriffen und für die grundlegende Beschreibung von VR-Menüs erweitert [Dachzelt u. Hübner 2007]. In der folgenden Übersicht werden die Menü-Eigenschaften, in Kategorien unterteilt, kurz vorgestellt und jeweils mögliche Ausprägungen und Optionen aufgezeigt.

#### Nutzungsvorhaben

Menüsysteme werden im Hinblick auf ihren jeweiligen Anwendungsfall entworfen. An erster Stelle stehen daher Überlegungen zu Zweck, Inhalt und Art des Menüs. Die Anzahl der Einträge und der Hierarchiestufen sind wichtige Kriterien, da der Umfang und die benötigte Hierarchiestruktur das Menüdesign direkt beeinflussen. Ein Menü, das für zehn Einträge eine gute Performance liefert, ist nicht zwangsläufig auch für hundert Einträge effektiv [Bowman et al. 2005].

- *Anzahl der möglichen Einträge*: feste oder variable Anzahl, mit oder ohne Obergrenze
- *Hierarchie*: eine, zwei oder mehr Stufen

#### Erscheinungsweise und Struktur

Die geometrische Form des Menüs hängt mit der inhaltlichen Struktur und der Art der Einträge zusammen. Sie beeinflusst die Sichtbarkeit, Erkennbarkeit und Bedienbarkeit der Einträge sowie die Erinnerung des Benutzers an die Position eines Eintrags.

- *Geometrische Repräsentation*: Rechteck, Scheibe, Ring, Kugel, Zylinder, Würfel, andere Formen (siehe auch 3.4.2)

- *Struktur der Anordnung*: als Liste (zyklisch oder azyklisch), als Matrix, gemäß der geometrischen Form, freie Platzierung
- *Darstellungsform*: Text und/oder Icons (Bilder oder Objekte)
- *Größe und Abstand*

#### **Platzierung**

Die Platzierung des Menüs in der Szene beeinflusst die Zugänglichkeit durch den Benutzer sowie die gegenseitige Verdeckung von Menüs und Objekten. Die Referenz gibt das für die Positionierung maßgebliche Koordinatensystem vor. Welt-referenzierte Menüs erscheinen an einer festen Position in der Szene, Menüs mit Display-Referenz erscheinen an einer Fixposition auf dem Display (z.B. am Bildschirmrand). Bei Objekt, Kopf- und Körper-Referenz hängt die Position des Menüs von der Referenzposition ab. Körper-referenzierte Menüs können je nach Form und Interaktion auch blind bedient werden, d.h. der Benutzer ist ohne eine grafische Repräsentation des Menüs in der Lage, durch eine Geste den Eintrag auszuwählen.

- *Referenz*: Welt, Objekt, Kopf, Körper (z.B. Hand), Display.
- *Orientierung*: feste Ausrichtung, Abhängig von Referenz, immer Richtung Benutzer
- *Neupositionierung*: Verschieben durch den Benutzer möglich/unmöglich
- *Dynamik*: Die Positionierung wird bei Aufruf festgelegt und bleibt erhalten oder die Position folgt der Referenz

#### **Aufruf und Verfügbarkeit**

Diese Eigenschaften beschreiben, wie der Benutzer das Menü aufruft und wieder schließt. Die Kontextabhängigkeit kann z.B. bedeuten, dass ein Menü nur sichtbar ist, wenn ein bestimmter Interaktionsmodus aktiv ist; die Sichtbarkeit eines Untermenüs ist dagegen von der Selektion eines hierarchisch höhergestellten Menüeintrags abhängig. Animationen helfen beim Verständnis von Zusammenhängen: Hierarchie-Strukturen werden deutlicher hervorgehoben und der Benutzer bekommt optische Hinweise auf Ursprung, Zugehörigkeit und Verbleib eines Menüs.

- *Sichtbarkeit*: dauerhaft, zeitweise, Entscheidung des Benutzers
- *Aufruf*: Anwählen von Icon/Miniatur, Kontext-abhängig (abhängig von anderen Menüeinträgen, Objekten oder der Situation), jederzeit (z.B. über eine Menü-Taste), gar nicht (immer sichtbar)

- *Animation*: bei Erscheinen, Verschwinden, Hierarchiewechsel, Auswahl, Aktivierung etc.: z.B. einblenden, ausblenden, zoomen, rotieren, verschieben, pulsieren, vergrößern etc.
- *Minimieren*: Verkleinern oder „Zusammenfallen“ des Menüs ohne es vollständig zu entfernen
- *Schließen*: automatisch nach Selektion, Entscheidung des Benutzers, dauerhaft sichtbar

#### **Interaktion**

Die Interaktionseigenschaften legen den physikalischen Rahmen der Menübenutzung fest. Die Auswahl des Interaktionsgerätes im Zusammenhang mit Tracking und Displaytyp hat großen Einfluss auf die Interaktionsmöglichkeiten. Die Dimensionalität bestimmt, in wie vielen Freiheitsgraden der Benutzer mit dem Menü interagiert und wie die Freiheitsgrade des Eingabegerätes auf die Menü-Dimensionen übertragen werden. Davon hängt ab, wie der Benutzer einen Eintrag auswählt und diesen aktiviert. Feedback in unterschiedlichen Formen bestätigt den Erfolg der Interaktion. Der Selektionspfad informiert den Benutzer über die Verschachtelung der aktuellen Hierarchiestufe.

- *Eingabegerät*: Maus, Tastatur, Joystick, Datenhandschuh, Gerät mit 3-DOF oder 6-DOF Tracking, Blicksteuerung, Bilderkennung sowie andere, spezielle oder mehrere Eingabegeräte
- *Einsatzbereich*: Desktop (2D), Desktop-VR, immersives VR, AR; Bildschirm, Projektion, HMD
- *Dimensionalität*: Menüdimensionen (1-DOF, 2-DOF, 3-DOF), verschiedene Übertragungsfunktionen vom Eingabegerät auf die Menüinteraktion (siehe Kapitel 4 ab Seite 23)
- *Selektion*: Schnitt von Cursor mit Objekt, Objekt in einen Selektionsbereich bewegen
- *Auswahl*: Berührung, Taste drücken, Zeitspanne abwarten
- *Feedback*: Objekt oder Hintergrund einfärben; Helligkeit, Kontrast, Form oder Größe ändern, Selektionsgeometrie hinzufügen; Animation
- *Selektionspfad*: Darstellung des Verlaufs oder der Hierarchie, mit oder ohne Möglichkeit des Sprungs zu einem vorherigen Punkt

#### **Benutzerfreundlichkeit**

Die Benutzerfreundlichkeit von Menüsystemen kann in Studien überprüft und mit anderen Menüs verglichen werden.

- *Evaluationskriterien*: Selektionsgeschwindigkeit, Fehlerrate, Effizienz, Bedienkomfort, Einfachheit (für Anfänger bzw. Experten), Lernkurve
- *Vergleich*: ganze Menüs/Menüsysteme, Darstellungsformen, Interaktionsformen

#### **Kombinationsmöglichkeiten**

Verschiedene Menüarten lassen sich zu komplexen Menüsystemen zusammenschließen oder innerhalb einer Applikation parallel nutzen. Manche Darstellungsformen schließen die Kombination mit anderen Menüs aus. Gründe können geometrische oder strukturelle Unterschiede sein, oder dass die Interaktionsformen nicht vereinbar sind. Wenn Darstellung und Interaktion der einzelnen Menüsysteme aufeinander abgestimmt sind, können sie sich gegenseitig ergänzen.

In 2D-Applikationen stehen nicht selten Drop-Down-Menüs, Pop-Up-Menüs, mehrere Werkzengleisten und Paletten gleichzeitig zur Verfügung, um den schnellen Zugriff auf möglichst viele Funktionen zu ermöglichen. Die Darstellung der verfügbaren Funktionen dient gleichzeitig als Gedankenstütze und hilft dem Benutzer beim Erlernen des Programms.

### **3.5 Zusammenfassung**

In den vorangegangenen Abschnitten wurde eine Klassifikation von Widget- und Menü-Eigenschaften vorgestellt. Dies ermöglicht die Beurteilung von bekannten Menü-Systemen im Vergleich zu den Anforderungen an eine Applikation und macht die Systeme untereinander vergleichbar. Diese Übersicht dient nicht nur als Entscheidungshilfe, welche Menüart für eine spezifische Aufgabe gut geeignet ist, sie ermöglicht auch die Ableitung von Paradigmen für die Neuentwicklung eines Menüsystems. Sie zeigt für die verschiedenen Fragestellungen die verfügbaren Optionen auf und hilft diese im Kontext der Anforderungen zu bewerten.

## 4 Dimensionalität von Menüs

Die Dimensionalität ist eine geeignete Eigenschaft, um die technischen Unterschiede der verschiedenen Menü-Implementierungen aufzuzeigen. Sie definiert einerseits die Anzahl der Freiheitsgrade, mit denen der Benutzer einen Eintrag aus dem Menü auswählen kann [Mine 1995]. Andererseits definiert sie die Anzahl der Freiheitsgrade, die der Benutzer mit seinem Eingabegerät steuern kann und wie diese Eingabe-Dimensionen auf die Interaktion mit dem Menü übertragen werden.

In diesem Kapitel werden zunächst die Grundlagen für die effiziente Auswahl von Elementen erläutert. Danach werden die Herausforderungen der Menü-Implementierung anhand von Beispielen vorgestellt. Die Beispiele werden in drei Gruppen gemäß ihrer Dimensionalität unterteilt (ein, zwei und drei Freiheitsgrade) und unter Berücksichtigung der in Kapitel 3 vorgestellten Eigenschaften beschrieben. Da aus Platzgründen nicht auf alle in der Literatur beschriebenen Menüsysteme eingegangen werden kann, stehen die Beispiele stellvertretend für ihre jeweilige Interaktionsart.

Die Darstellung eines Menüs als 2D-, 3D- oder stereoskopische 3D-Grafik ist von seiner Dimensionalität unabhängig. Jedoch stehen in den verschiedenen 2D- und 3D-Systemen meist unterschiedliche Interaktionsgeräte und -techniken zur Verfügung, die durch ihre Freiheitsgrade einen direkten Einfluss auf die Interaktion mit den Menüs haben.

### 4.1 Fitts' Law

Es gibt viele Faktoren, welche die Effizienz der Interaktion bei Auswahlaufgaben beeinflussen. Dazu gehören Nutzer-spezifische Eigenschaften wie seine Fähigkeiten, seine Erfahrung und Übung sowie das Eingabegerät, die Interaktionstechnik, Rückmeldungen des Systems und die Darstellungsform der virtuellen Elemente: Sichtbarkeit, Sortierung, Beschriftung, Größe und Entfernung. Der Einfluss der beiden letzten Faktoren – Größe und Entfernung – wird im sog. Fitts' Law [Fitts 1954] beschrieben. Dieses Gesetz ist ein eindimensionales Modell der menschlichen Bewegung. Es besagt, dass die Zeit, um von einem Startpunkt aus möglichst

schnell ein Ziel zu erreichen, von der Zielentfernung und der Zielbreite abhängt.

$$T = a + b \log_2 \left( \frac{D}{W} + 1 \right) \quad (4.1)$$

$T$  ist die Zeit bis zum Erreichen des Ziels. Die Werte für  $a$  und  $b$  können durch lineare Regression empirisch ermittelt werden.  $D$  ist die Distanz vom Startpunkt bis zum Zielzentrum.  $W$  ist die Breite des Ziels, gemessen entlang der Achse vom Start- zum Zielpunkt. Das Gesetz sagt voraus, in wie weit das Zeigen auf kleinere oder weiter entfernte Ziele mehr Zeit in Anspruch nimmt. Die Formel zeigt damit einen Kompromiss aus Genauigkeit und Geschwindigkeit auf. Untersuchungen haben ergeben, dass dieses eindimensionale Gesetz in abgewandelter Form auch auf zweidimensionale und dreidimensionale Zielaufgaben angewendet werden kann und dadurch insbesondere für die Interaktion und Systemsteuerung in 2D- und 3D-Benutzerschnittstellen relevant ist. [MacKenzie u. Buxton 1992; Accot u. Zhai 2003; Grossman u. Balakrishnan 2004].

## 4.2 Interaktion mit Steuerelementen

Die Anzahl der Menü-Dimensionen bestimmt den strukturellen Aufbau, das Layout und die Interaktion mit dem Menü. Die Interaktion zwischen menschlichem Körper (z.B. der Hand des Benutzers) und dem Bedienelement nennt man *Control-Body-Linkage* [Bowman et al. 2005]. Je natürlicher die Eingabedimensionen auf das virtuelle Element übertragen werden, desto direkter, intuitiver und effektiver wird die Steuerung. Die Anordnung der Elemente und die Interaktionsdimensionen sollen nach Möglichkeit weitestgehend mit der Aufgabe übereinstimmen [Darken u. Durost 2005]. Die Änderungen der nicht für die Interaktion relevanten Freiheitsgrade sollen keinen Einfluss auf die Interaktion haben [Mine 1995]. Zusätzliche zu kontrollierende Freiheitsgrade fügen der Interaktion immer auch einen weiteren Schwierigkeitsgrad hinzu. Die Beschreibung der eindimensionalen Menüselektion im nächsten Abschnitt veranschaulicht diesen Sachverhalt.

## 4.3 Menüs mit einem Freiheitsgrad

Die Auswahl aus einer Liste ist im Grunde eine eindimensionale Aufgabe [Bowman et al. 2005]. Die Anordnung der Elemente und die Interaktion mit dieser Liste findet in einer Dimension statt, d.h. es kann nur ein Parameter beeinflusst werden. Dafür gibt es unterschiedliche Möglichkeiten: Die Auswahl eines Menüpunktes erfolgt z.B. über Tasten, über das Verschieben des Interaktionsgerätes entlang einer Achse oder durch die Rotation um eine Achse. Die jeweils



nicht genutzten Dimensionen (Translation und Rotation um die verbleibenden Achsen) werden ignoriert [Mine 1995]. Am Beispiel von Drop-Down- und Ring-Menüs werden die Konzepte eindimensionaler Menüelektronik vorgestellt. Anhand des Drop-Down-Menüs wird exemplarisch auf die Schwierigkeiten bei der Übertragung eines Menükonzeptes von 2D-Desktop-Systemen in immersive VR-Anwendungen eingegangen.

### 4.3.1 Listen-Menü

Ein Menüsystem für Drop-Down-Menüs besteht aus einer eindimensionalen Menüzeile sowie aus den aufklappenden Menüs. Betrachtet man das Menüsystem als eine Kombination einzelner Menüs, ergibt sich jeweils nur ein Freiheitsgrad für die Auswahl aus einer Liste. Wenn wir von einer Mausbedienung an einem 2D-Display ausgehen, muss die Maus für die jeweilige Auswahl nur entlang einer Achse verschoben werden. Die Menüeinträge des Drop-Down-Menüs sind ausreichend breit, so dass eine Abweichung auf der horizontalen Achse praktisch keinen Einfluss auf die Auswahl hat. Eine Menüzeile dagegen ist vergleichsweise schmal. Der Einfluss der vertikalen Achse führt leicht zu Fehlern, die durch eine akkuratere und damit langsamere Bedienung durch den Benutzer ausgeglichen wird. Die Platzierung der Menüzeile am oberen Bildschirmrand verhindert diese Problematik: Der Benutzer kann bei der horizontalen Interaktion die Begrenzung des Wertebereichs an der Bildschirmgrenze ausnutzen, indem er den Mauszeiger entlang dieser virtuellen Kante verschiebt. Die physikalische oder virtuelle Begrenzung der nicht benötigten Freiheitsgrade ist ein probates Mittel, um die Interaktion zu vereinfachen. Der Einfluss nicht benötigter Freiheitsgrade des Eingabegeräts wird dadurch ausgeschlossen.

### Umsetzung in VR

Das gleiche Menüsystem, eins zu eins übertragen in eine immersive VR-Anwendung, verlangt vom Benutzer eine sehr viel komplexere Interaktion: Anstatt die Maus auf dem Tisch in einer Dimension zu verschieben und den Mauszeiger auf einem Menüeintrag zu platzieren, muss der Benutzer seine Hand frei im Raum positionieren, bis der 3D-Cursor den Menüeintrag berührt. Dabei fehlt dem Benutzer jegliche Form von physikalischer oder virtueller Unterstützung. Die fehlende physikalische Unterstützung führt zu einer schnelleren Ermüdung der beteiligten Muskelgruppen. Zudem belegen Studien, dass die Interaktion langsamer wird, wenn große Muskelgruppen (z.B. Schulter, Oberarm) an der Bewegung beteiligt sind. Aus einer simplen 1-DOF-Interaktion ist eine fehleranfällige und langsame 3-DOF-Interaktion geworden [Hand 1997; Zhai et al. 1996; Grossman u. Balakrishnan 2004].

### **Strahl-Interaktion**

Die Nutzung einer strahlbasierten Interaktion mindert diese Schwierigkeit. Eine genaue räumliche Übereinstimmung zwischen Hand und Menüeintrag ist nicht mehr notwendig; der Benutzer kann auch über eine Distanz hinweg auf den Menüeintrag zeigen und bekommt durch den Strahl eine deutlich sichtbare Rückmeldung über seine Bewegungen [Hand 1997]. Zeigt der Strahl auf das Menü, kann der Benutzer durch Translation und Rotation der Hand den Eintrag auswählen. Der Schnittpunkt von Strahl und Menüebene bewegt sich in beiden Fällen auf der gleichen Achse. Der Benutzer muss zwei Freiheitsgrade für die eindimensionale Auswahl berücksichtigen.

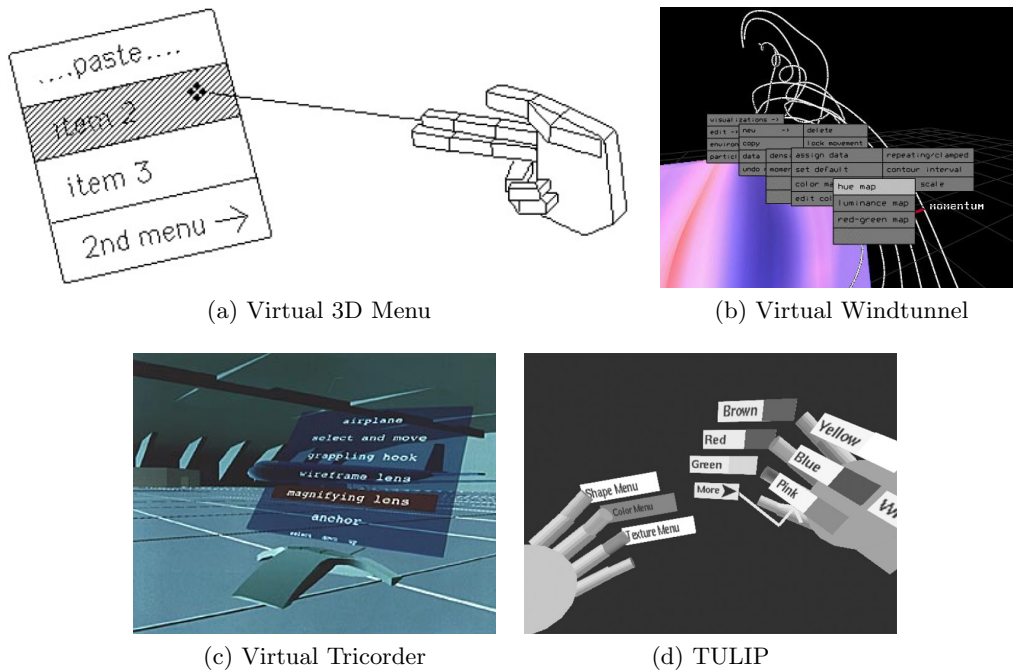
### **Implementierungen**

Das erste implementierte VR-Menü ist eine textbasierte Auswahlliste, die per Strahlinteraktion und verschiedenen Selektions- und Aktivierungsgesten mit einem Datenhandschuh bedient wird (Abbildung 4.1a). Das Menü schwebt nach einer festgelegten Kommandogeste einen Meter vor dem Kopf des Benutzers. Aus der Spitze des Zeigefingers entspringt ein Strahl, der mit dem gewünschten Menüeintrag geschnitten wird. Anschließend formt der Benutzer aus Daumen und Zeigefinger eine Null, um den Eintrag auszuwählen [Jacoby u. Ellis 1992].

Die „Virtual Windtunnel“-Applikation (Abbildung 4.1b) der NASA wird ebenfalls per Datenhandschuh und Gesten bedient, wobei die komplette Steuerung der Applikation mit drei Handgesten auskommt: Faust, Zeigen (ausgestreckter Zeigefinger) oder keine Geste. Per Fingerzeig erscheint ein Textmenü im Sichtfeld vor der Hand. Per Handrotation zeigt der Benutzer auf einen Eintrag. Wird die Handgeste aufgelöst, wird der angewählte Eintrag ausgeführt. Das Menü unterstützt mehrere Hierarchien durch Darstellung vieler Menüflächen, die sich zum Teil gegenseitig verdecken [Bryson 1993].

Mit dem „Virtual Tricorder“, einem Mehrzweck-Eingabegerät, wird ein Hand-referenziertes Auswahlmenü vorgestellt (Abbildung 4.1c). Das Menü wird in fester Position über dem Eingabegerät dargestellt und folgt dessen Position. Der Eintrag wird mittels Auf/Ab-Tasten ausgewählt und über einen Selektionsknopf aktiviert [Wloka u. Greenfield 1995; Bowman et al. 2005].

Das „TULIP“-Menü wird mit einem Datenhandschuh mit Fingerkontakten bedient und verzichtet auf das räumliche Zeigen auf einen Menüeintrag. Die zur Verfügung stehenden Funktionen werden als virtuelle Texte an der Position der Finger dargestellt (Abbildung 4.1c). Durch Berühren einer Fingerspitze mit dem Daumen kann ein Eintrag selektiert werden. Reichen vier Finger nicht für alle Einträge, werden nur drei Funktionen angeboten. Der kleine Finger dient dann zum Blättern im Menü, wobei die weiteren Einträge zur besseren Übersicht



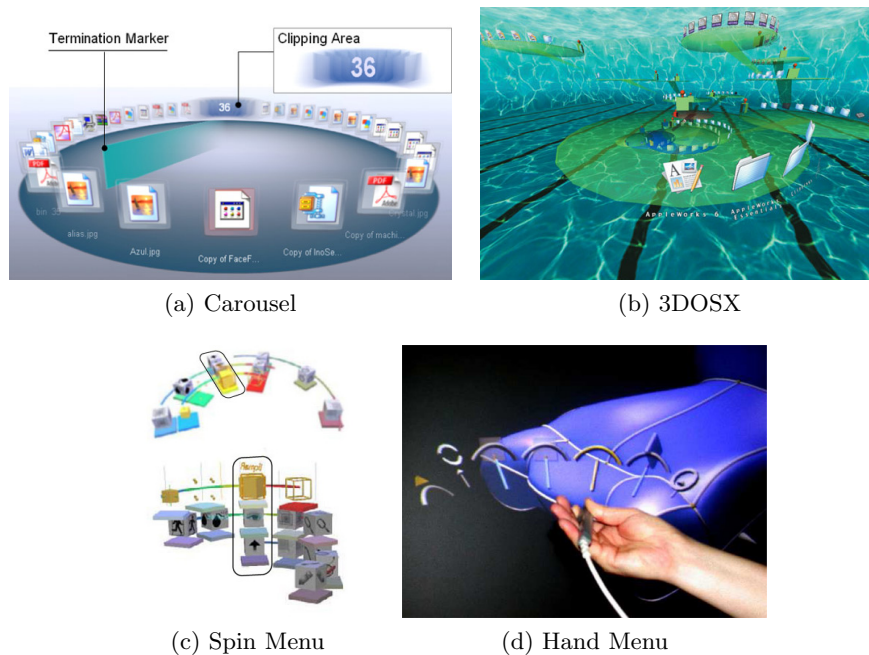
**Abbildung 4.1:** 1-DOF Listen-Menüs — (a) HMD-Visualisierung des 3D-Menüs mit Handgesten (entnommen aus [Jacoby u. Ellis 1992]) (b) NASA Virtual Windtunnel Menüs (entnommen aus [NASA VWT]) (c) Virtual Tricorder (entnommen aus [CONTIGRA Widgets]) (d) TULIP-Menü (entnommen aus [Bowman u. Wingrave 2001])

auf der virtuellen Handfläche dargestellt werden. Mit den Fingern der nicht-dominanten Hand können verschiedene Hauptmenüs ausgewählt werden [Bowman u. Wingrave 2001].

### 4.3.2 Ring-Menü

Bei Ring-Menüs sind die Elemente ringförmig um ein Rotationszentrum auf einer virtuellen Scheibe angeordnet und bilden somit eine zyklische Liste. Die Menüelemente bestehen aus bildhaften oder geometrischen Symbolen zur Repräsentation ihrer Funktionalität (oft ohne zusätzliche Beschriftung). Ring-Menüs sind besonders bei einer kleinen Anzahl an Elementen sehr effektiv [Gerber u. Bechmann 2004].

Im „JDCAD Ring-Menü“ werden Symbole zur Erzeugung geometrischer Objekte in einem Ring dargestellt. Das vorderste, dem Benutzer zugewandte Symbol, wird von einem Auswahlrahmen umschlossen und kann per Tastendruck aktiviert werden. Die Handrotation wird auf die Rotation des Rings übertragen. Da für diese Interaktion nur genau eine Rotationsachse relevant ist, haben alle sonstigen Rotationen und Translationen des Eingabegerätes keinen Einfluss auf die Auswahl [Liang u. Green 1994].



**Abbildung 4.2:** *1-DOF Ring-Menüs* — (a) Carousel mit Icon-Stapel an der Ring-Rückseite (entnommen aus [Wang et al. 2005]) (b) Der Datei-Browser 3DOSX mit hierarchischer Verzeichnisstruktur (entnommen aus [Chin 2002]) (c) Spin Menu mit konzentrischen (oben) und gestapelten (unten) Hierarchien (entnommen aus [Gerber u. Bechmann 2005]) (d) Hand Menu mit Werkzeugauswahl (entnommen aus [Wesche 2004a])

Das „Spin Menu“ ist als Kontext-Menü ausgelegt. Es erscheint auf Knopfdruck vor dem Eingabegerät und wird ebenfalls per Handrotation gedreht. Es unterstützt Hierarchien in drei Darstellungsformen: Konzentrische Ringe, gestapelte Ringe und orthogonal gekreuzte Ringe, wobei sich die letzte Version in Benutzerstudien als sehr fehleranfällig erwiesen hat (Abbildung 4.2c). Der Selektionspfad kann durch nebeneinanderliegende bzw. übereinandergestapelte Symbole nachvollzogen werden [Gerber u. Bechmann 2005].

Das „Carousel“-Menü ist eine Implementierung von Ring-Menüs mit dem Hauptaugenmerk auf die übersichtliche Darstellung sehr vieler Einträge. Die vorderen Icons werden gespreizt dargestellt, die hinteren deutlich enger (Abbildung 4.2a). Die Einträge am hintersten Ende der Scheibe verschwinden in einem symbolisierten Icon-Stapel [Wang et al. 2005]. Anwendung findet diese Idee in einem 3D-Datei-Browser, der die Dokument- und Verzeichnis-Hierarchien mit im Raum schwebenden und animierten Ring-Menüs darstellt [Chin 2002].

Eine weitere Möglichkeit ist die Anordnung der Elemente in einem azyklischen Kreisbogen (Abbildung 4.2d). Das Menü ist Hand-referenziert, „hängt“ also an der Hand des Benutzers, solange es sichtbar ist. Über die Handrotation in horizontaler Ebene zeigt der Benutzer mit dem Eingabegerät direkt auf einen Eintrag und kann diesen per Tastendruck aktivieren. Auch

in diesem Fall wirkt sich nur die Rotation um eine Achse auf die Selektion aus, alle anderen Freiheitsgrade haben keinerlei Einfluss. Da das Menü Körper-referenziert ist und die jeweiligen Einträge über festgelegte Handwinkel ausgewählt werden, ist mit Übung und Erfahrung eine blinde Steuerung des Menüs möglich [Wesche 2004a, b].

## 4.4 Menüs mit zwei Freiheitsgraden

Menüs mit zwei Freiheitsgraden sind prinzipiell ähnlich aufgebaut wie eindimensionale Menüs. Die Anordnung der Elemente und die Interaktion finden jedoch in zwei Dimensionen statt. Für die eindeutige Auswahl eines Eintrags ist die Beeinflussung von zwei Parametern notwendig, z.B. zwei Translationskoordinaten auf einer Ebene oder zwei Rotationsfreiheitsgrade. Ähnlich der eindimensionalen Menüinteraktion kann das Zeigen auf einen Eintrag direkt oder per Strahlinteraktion erfolgen.

### 4.4.1 Portierung von 2D-Menüs in VR

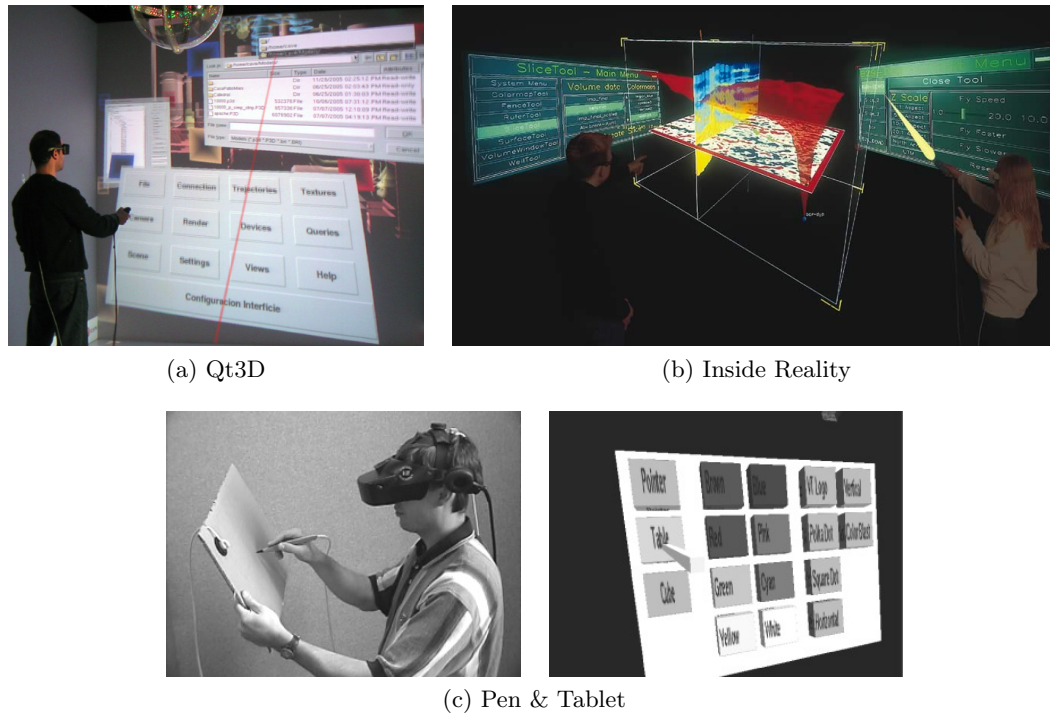
Die Bedienelemente werden über- und nebeneinander in einer Ebene angeordnet. Ein Großteil der 2D-Desktop-Programme nutzt die zweidimensionale Anordnung der Bedienelemente in Fenstern, Paletten und auf der Arbeitsoberfläche. Da für Desktopsysteme, im Gegensatz zu VR-Systemen, eine Reihe von Frameworks für die Gestaltung und Erstellung von 2D-Benutzerschnittstellen zur Verfügung stehen, sind diese weit verbreitet und jedem Computer-Benutzer geläufig. Es bietet sich daher an, diese Implementierung in die VR-Welt zu übertragen.

Mit „Qt3D“ existiert eine VR-Portierung des Qt-Frameworks<sup>3</sup>. Die 2D-Fenster werden als texturierte Ebenen in der VR-Szene dargestellt (Abbildung 4.3a). Der Benutzer bedient die Software auf Distanz per Strahlinteraktion. Der Schnittpunkt von Strahl und Ebene wird in zweidimensionale Mauskoordinaten zurückgerechnet und an das Qt-System übergeben. Der große Vorteil dieses Ansatzes besteht darin, dass ein solches Menüsystem für Desktop-VR-Systeme und immersive VR-Anwendungen gleichermaßen benutzt werden kann, ohne die sonst übliche Neuentwicklung eines VR-Menüs. Zusätzlich können auch normale 2D-Anwendungen, z.B. ein Webbrowser, in einem solchen 3D-Fenster bedient werden [Andujar et al. 2006].

Das Programm „Inside Reality“ von Schlumberger, eine Applikation für die Exploration von Öl- und Gasfeldern, nutzt ebenfalls ein Menüsystem mit 2D-Ebenen (Abbildung 4.3b). Um die Objekte in der Szene nicht zu verdecken, werden die Menüs hinter den Objekten entsprechend groß dargestellt, so dass die Bedienung mittels Strahlinteraktion möglich ist [Inside Reality].

---

<sup>3</sup>Bei Qt handelt es sich um eine plattformübergreifende Klassenbibliothek zur Entwicklung von Programmen mit grafischer Benutzeroberfläche.



**Abbildung 4.3:** 2-DOF Menüs — (a) Qt3D, Qt-Menüs als Textur auf Ebenen im Raum (entnommen aus [Andujar et al. 2006]) (b) Schlumberger Inside Reality, eine Software zur Öl- & Gas-Exploration in VR (entnommen aus [Inside Reality]) (c) Pen & Tablet Menü mit Virtueller Menüdarstellung auf einem realen Objekt, links: Benutzer mit HMD und Tablet, rechts: Virtuelle Darstellung des Menüs (entnommen aus [Bowman u. Wingrave 2001])

### 4.4.2 Physikalische Hilfsmittel für die Menüdarstellung

Anstelle VR-Menüs frei im Raum zu positionieren, können diese auch auf „echten“ Objekten dargestellt und bedient werden. Da zweidimensionale Menüs auf einer Ebene angeordnet werden können, macht es Sinn, diese auf einer physikalischen Ebene darzustellen. Die Oberfläche gibt außerdem taktile Rückmeldung über die Interaktion und macht die Bedienung natürlicher und intuitiver.

Die „3D Palette“ nutzt eine transparente, rechteckige Tafel mit 6-DOF Tracking, um die Menüpunkte virtuell auf der Oberfläche der Scheibe darzustellen. Die Auswahl eines Menüpunktes, dargestellt durch Icons oder kleine 3D-Objekte, geschieht durch Antippen der Symbole mit einem 6-DOF-Stylus. Die Menütafel wird in der nicht-dominanten Hand gehalten und kann jederzeit ins Blickfeld bewegt werden. Auch große virtuelle Objekte können auf der Oberfläche platziert und ganz natürlich durch Bewegen und Drehen der Hand von allen Seiten betrachtet werden [Billinghurst et al. 1997]. Abbildung 4.3c zeigt das „Pen & Tablet“-Menü, dessen Interaktion identisch ist.

Wie in Kapitel 2, Abschnitt 2.3 bereits erläutert, können virtuelle Menüs auch auf einem Tablet-PC oder PDA mit Touch-Screen dargestellt werden. Die Display-Auflösung erlaubt die Darstellung komplexer Menü-Steuerungen in gewohnter 2D-Qualität. Allerdings ist es, im Gegensatz zur 3D-Palette, nicht möglich virtuelle Objekte auf der Oberfläche räumlich darzustellen.

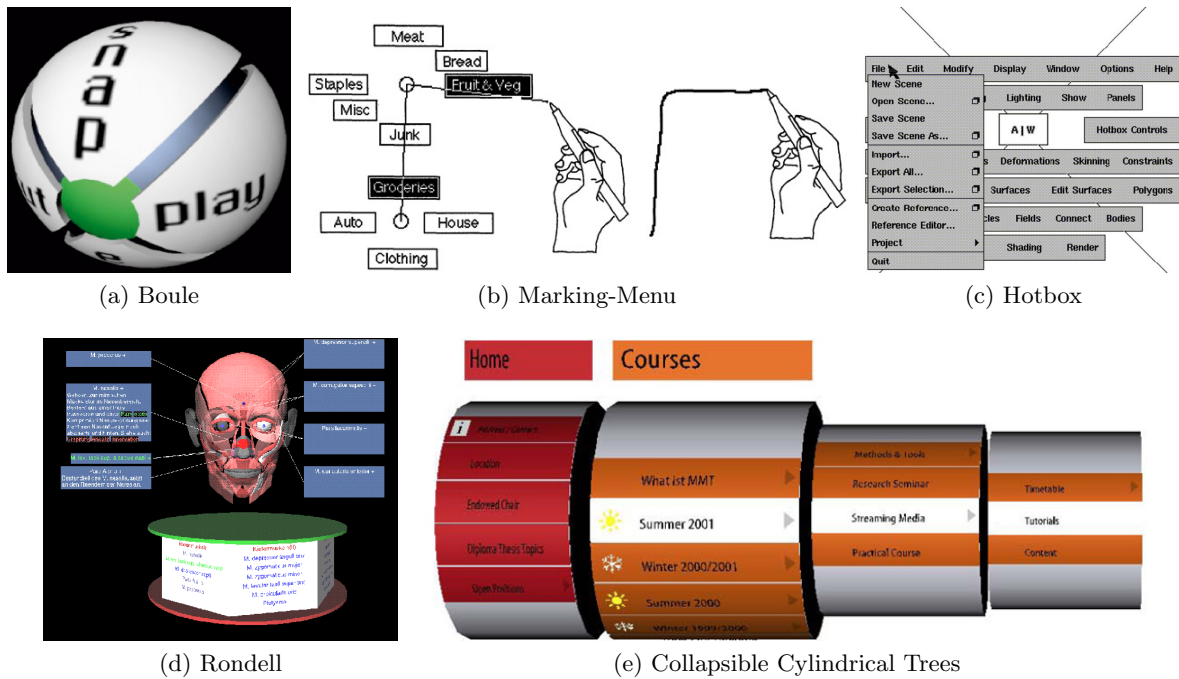
### 4.4.3 Kreis-Menü

Kreis-Menüs sind Kontext-Menüs, deren Einträge kreisförmig um die Cursorposition aufgebaut werden. Im Vergleich zu Listen-Menüs werden die Einträge über die Bewegungsrichtung und nicht über die Bewegungsdistanz selektiert. Da nur wenige, relativ große Einträge in unmittelbarer Nähe der Cursorposition dargestellt werden, gilt diese Menüform als sehr schnell und effizient (vgl. Fitts Law in Abschnitt 4.1). Hierarchien werden durch weitere Menüs dargestellt, die nach der Anwahl eines Eintrags um den Cursor zentriert erscheinen. Dabei kann die vorherige Hierarchiestufe ausgeblendet oder teilweise verdeckt werden. Die Auswahl eines Eintrags kann durch einen Strich vom Zentrum zum Eintrag markiert werden, weshalb diese Menüart auch Markierungs-Menü genannt wird. Bei Selektionen über mehrere Hierarchiestufen ergibt sich eine Zickzack-Linie. Durch die handreferenzierte Position und die absolute Richtungsangabe können Markierungs-Menüs blind bedient werden. Der Benutzer zeichnet dann ohne die Darstellung des Menüs die Markierungslinie als Geste und aktiviert damit die gewünschte Funktion (Abbildung 4.4b) [Kurtenbach u. Buxton 1993].

Das „Boule“-Menü ist eine spezielle VR-Variante eines Kreis-Menüs. Das Menü wird als Kugel angezeigt und enthält vier Menüpunkte. Anstelle der Translationsbewegung zu einem Eintrag, rotiert der Benutzer die Kugel zum gewünschten Eintrag durch Handrotation (Abbildung 4.4a). Auch diese Menüform ist durch die Hand-Referenzierung und die absoluten Rotationswinkel blind zu bedienen [Häfner et al. 1999, 2000].

Die „Hotbox“ dient als Ersatz der traditionellen Menüzeile und bietet schnellen Zugriff auf die ausufernde Funktionalität von komplexen Programmen. Sie kombiniert Kreis-Menüs mit Menüzeilen und Drop-Down-Menüs (Abbildung 4.4c). Ähnlich dem Kreis-Menü werden die Einträge um die Cursor-Position angeordnet, können dort jedoch mehrere Zeilen und Spalten einnehmen. Die einzelnen Einträge können bei Aktivierung ihrerseits Kreis-Menüs oder Drop-Down-Menüs hervorrufen. Die Hotbox bleibt halbtransparent sichtbar, solange die Space-Taste auf der Tastatur gedrückt bleibt. Der Benutzer kann mehrere Kommandos per Mausbedienung absetzen, bevor er die Hotbox schließt [Kurtenbach et al. 1999].

Die Bedienung des „Rondell“-Menüs erinnert an einen Postkartenständer: Auf mehreren Seiten eines drehbaren Rondells sind Listen-Menüs dargestellt (Abbildung 4.4d). Die Drehung nach links oder rechts wird über Mausklicks durchgeführt. Die Listeneinträge können angeklickt

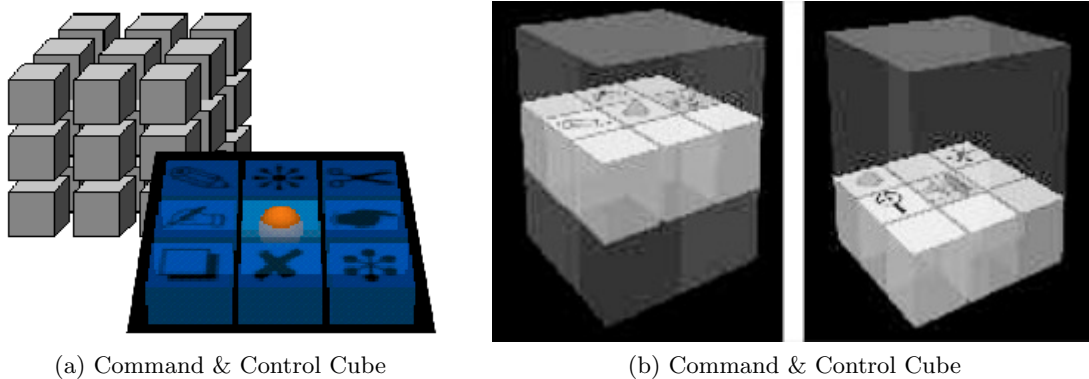


**Abbildung 4.4:** 2-DOF Kreis-Menüs — (a) Boule-Menü (entnommen aus [Häfner et al. 1999]) (b) Markierungs-Menü mit resultierender Geste (entnommen aus [Kurtenbach u. Buxton 1993]) (c) Hotbox, Menüsystem der 3D-Applikation Autodesk Maya, mit Drop-Down-Menü (entnommen aus [Kurtenbach et al. 1999]) (d) Rondell (entnommen aus [Preim et al. 1997]) (e) Collapsible Cylindrical Trees, mit zwei Hauptkategorien und zwei Hierarchiestufen (entnommen aus [Dachselt u. Ebert 2001])

werden. Die einzelnen Menüs können explorativ betrachtet werden und wecken durch die optisch ansprechende Darstellung das Interesse des Benutzers [Preim et al. 1997].

Bei den „Collapsible Cylindrical Trees“ handelt es sich um die Visualisierung einer Baumstruktur. Die Einträge einer Ebene werden Zeilenweise auf einem drehbaren, horizontal liegenden Zylinder dargestellt. Die Rotation erfolgt über Mausbewegungen in der vertikalen Achse. Die Hierarchiestufen können über kurze Links-Rechts-Bewegungen der Maus erreicht werden, wobei die Einträge der folgenden Hierarchiestufe teleskopartig nach rechts ausgefahren werden (Abbildung 4.4e). Die Mauswege zur Steuerung der Zylinder sind im Vergleich zu anderen 2D-Menüs stark verkürzt, da der Mauszeiger nicht über die Darstellungsdistanz bewegt werden muss, sondern die horizontalen Mausbewegungen als Schalter für das Auf und Ab in der Hierarchie fungieren [Dachselt u. Ebert 2001].





**Abbildung 4.5:** *3-DOF Menüs* — (a) Würfelstruktur des  $C^3$  mit Darstellung einer Menülage (entnommen aus [CONTIGRA Widgets]) (b)  $C^3$  in einer VR-Applikation, mittlere und untere Lage (entnommen aus [Grosjean u. Coquillart 2001])

## 4.5 Menüs mit drei Freiheitsgraden

Menüs mit drei Freiheitsgraden haben keine Entsprechung auf 2D-Desktop-Systemen, da sie auf die Eingabe von drei Parametern angewiesen sind. Eine Maus kann jedoch nur auf zwei Achsen verschoben werden, so dass 3-DOF Menüs prinzipiell nicht mit einer Maus bedient werden können. Da die Interaktion in drei Freiheitsgraden oft als schwierig erachtet wird und eine 2D-Kompatibilität nicht gegeben ist, finden sich in der Literatur nur wenige Implementierungen für innovative 3-DOF Menüs. Ein- und zweidimensionale Menüs, die auf einer Ebene in 3D dargestellt und mit direkter Interaktion bedient werden, sind zwar gewissermaßen 3-DOF Menüs, da für ihre Bedienung die Position der Hand in drei Raumkoordinaten der Position des Bedienelementes übereinstimmen muss, jedoch schöpft diese Interaktionsform keinen zusätzlichen Nutzen aus der dritten Dimension. Eine seltene Ausnahme ist der „Command & Control Cube“ (CCC oder  $C^3$ ) [Grosjean u. Coquillart 2001; Grosjean et al. 2002].

Der  $C^3$  ist ein würfelförmiges Menü, unterteilt in  $3 \times 3 \times 3$  kleinere Würfel, zwischen denen jeweils eine kleine Lücke frei ist. Er enthält insgesamt 27 Einträge mit Icons, wobei der mittlere Würfel die neutrale Position darstellt, über die das Menü verlassen werden kann, ohne eine Funktion auszulösen (Abbildung 4.5a). Das Menü wird mit der nicht-dominanten Hand bedient, in der sich ein Taster mit 6-DOF-Tracking befindetet. Nach Druck auf den Taster erscheint der Würfel zentriert um die Tasterposition und zeigt die mittlere Lage mit acht würfelförmigen Symbolen rund um das Zentrum an. Die Bedienung funktioniert ähnlich der Markierung innerhalb eines Ring-Menüs, erweitert auf drei Dimensionen. Bewegt der Benutzer die Hand in der dargestellten Ebene auf ein Symbol und lässt den Taster los, wird dieser Eintrag ausgewählt. Bewegt er die Hand nach oben oder unten, wird entsprechend die obere oder untere Lage der

Würfelstruktur mit je neun Einträgen angezeigt (Abbildung 4.5b). Auch dieses Menü erlaubt die blinde Bedienung durch Handreferenz und absolute Positionseingabe. Da ohne zusätzliche Hierarchien 26 verschiedene Kommandos durch einfache Gesten abgesetzt werden können, bezeichnen die Autoren dieses Menü als die Portierung der Tastatur-Shortcuts in VR.

## 5 VR-Menüs im Expertenkontext

In den folgenden Abschnitten werden Interaktionsmethoden für ein Visualisierungssystem für seismische Daten und der daraus resultierende Expertenkontext vorgestellt. Die bis dato verwendeten Menü-Implementierungen auf PDA und Tablet-PC werden beschrieben und diskutiert. Aus den eigenen Erfahrungen sowie Studien aus der Literatur wird ein Anforderungsprofil für das neue VR-Menüsystem erarbeitet.

### 5.1 Anwendungsszenario

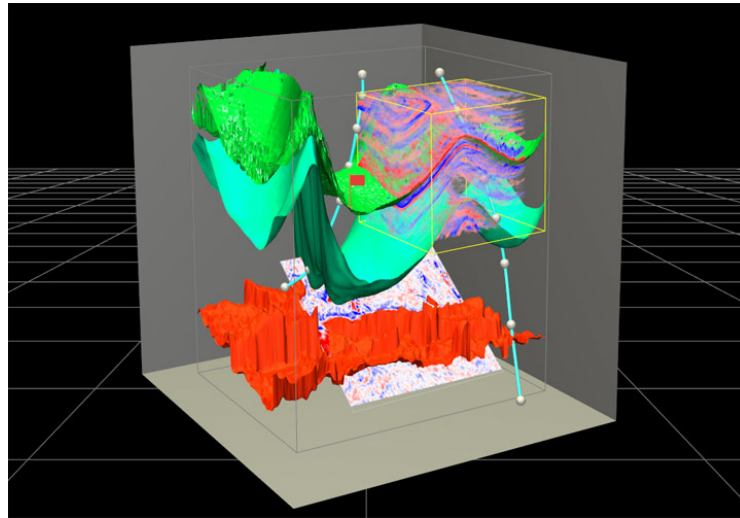
Die vorliegende Diplomarbeit wurde im Rahmen des VRGeo-Projektes (*Virtual Reality for the Geosciences*) am Fraunhofer Institut für Intelligente Analyse- und Informationssysteme (IAIS) angefertigt. Das VRGeo-Konsortium ist ein Zusammenschluss internationaler Unternehmen, die in der Öl- und Gasbranche tätig sind bzw. VR-Produkte und -Technologien für diesen Markt anbieten oder erforschen [VRGeo]. Das Projekt hat das Ziel, neuartige VR-Anwendungen und VR-Systeme für die Branche nutzbar zu machen. Drei Disziplinen sind im Rahmen der Forschung innerhalb des Projektes besonders hervorzuheben: Rendering-Verfahren, VR-Systeme und Interaktion. Bei den Rendering-Verfahren liegt der Fokus hauptsächlich auf der echtzeitfähigen Darstellung von großen Volumendatensätzen und Geometrien. VR-Systeme bezeichnen die Gesamtheit aus VR-Displays sowie deren Ansteuerung durch PCs und PC-Cluster sowie die Software in Form von VR-Frameworks. Bei der Interaktion schließlich geht es um Eingabegeräte, Interaktionstechniken und Benutzerschnittstellen. Die vorliegende Diplomarbeit ist in letztgenannter Kategorie einzuordnen. Zweimal im Jahr findet am Fraunhofer IAIS das sog. VRGeo-Meeting statt. Auf dieser Konferenz werden den Teilnehmern die neuesten Forschungsergebnisse des IAIS und seiner Partner sowie Produkt- und Technologienerneuerungen vorgestellt. In einer umfangreichen Demo-Session können die Teilnehmer (u.a. Experten im Bereich VR, Geologie und Geophysik) eigene Erfahrungen mit der vorgestellten Hard- und Software sammeln.

## 5.2 Der VRGeo-Demonstrator

Der VRGeo-Demonstrator ist eine Demonstrations- und Entwicklungs-Plattform, in der die neuesten Technologien aus den o.g. Forschungsschwerpunkten implementiert sind. Dabei handelt es sich um eine prototypische immersive VR-Applikation zur Betrachtung seismischer Volumendaten.

**Tabelle 5.1:** Objekte, Funktionen und Interaktionsarten im VRGeo-Demonstrator

Objekt	Funktion	Typ	Interaktionsart
Arbeitsbereich	Translation & Rotation	6-DOF	Direkte Interaktion
	Laden/Speichern	Befehl, Listenauswahl	Menüsteuerung
	Löschen	Befehl	
	Umbenennen	Befehl, Texteingabe	
	Größe ändern	Wert	
	Neue Objekte erzeugen	Befehl	
Volumenlinse	Translation & Rotation	6-DOF	Direkte Interaktion
	Löschen	Befehl	Menüsteuerung
	Umbenennen	Befehl, Texteingabe	
	Größe ändern	Wert	
	Darstellungsqualität	Wert	
	Darstellungsoptionen	Checkbox	
Schnittebene	Translation & Rotation	6-DOF	Direkte Interaktion
	Löschen	Befehl	Menüsteuerung
	Umbenennen	Befehl, Texteingabe	
	Größe ändern	Wert	
Bohrpfade (Ankerpunkte)	Translation	3-DOF Position	Direkte Interaktion
Markierung	Translation	3-DOF Position	Direkte Interaktion
	Löschen	Befehl	Menüsteuerung
	Umbenennen	Befehl, Texteingabe	
Global	Snappingoptionen	Radiobutton, Check- box, Wert	Menüsteuerung
	Darstellungsoptionen	Checkbox	
	Auswahl sichtbarer Objekte	Checkbox	
	Programm beenden	Befehl	



**Abbildung 5.1:** *Der VRGeo-Demonstrator* — Das Bild zeigt einen Arbeitsbereich (graue Box) im VRGeo-Demonstrator. Darin enthalten sind Horizons (grüne und rote Bodenschichten), eine Schnittebene (weiße Fläche), eine Volumenlinse (rot/blau Schichtdarstellung), Bohrpfade und eine Markierung in Form einer kleinen Flagge.

Der Demonstrator wird in der 2007er-Version unter anderem auf dem TwoView-Display gezeigt. Zwei Benutzer, ausgestattet mit 3D-Brille und einem kabellosen 6-DOF Eingabegerät mit mehreren Tastern, können in der virtuellen Ansicht zusammen arbeiten. Die VR-Szene ist in Arbeitsbereiche unterteilt, die jeweils einen Volumendatensatz repräsentieren und weitere 3D-Objekte enthalten können. Zu diesen Objekten gehören Volumenlinsen, Schnittebenen, Bohrpfade und Markierungen (siehe Abbildung 5.1). Volumenlinsen zeigen einen würfelförmigen Ausschnitt aus den Volumendaten, Schnittebenen zeigen die Daten entlang eines zweidimensionalen Schnitts. Die Bohrpfade bestehen aus beweglichen Ankerpunkten, zwischen denen ein Pfad interpoliert wird. Markierungen sind frei positionierbare Objekte zum Ermitteln von Messwerten. Die räumliche Interaktion findet mit Hilfe des 6-DOF Eingabegerätes und der Scaled-Grab-Strahlinteraktion statt (siehe Abschnitt 2.4). Neben der direkten Interaktion mit den Objekten macht eine Vielzahl von weiteren Funktionen eine Systemsteuerung über Menüs unverzichtbar. Tabelle 5.1 beschreibt die verfügbaren Objekte und listet die jeweils wichtigste Funktionalität auf. Die Funktionen sind nach den Möglichkeiten ihrer Manipulation über direkte Interaktion bzw. Steuerung über ein Menüsystem gruppiert. Neben den Funktionen werden die Elementtypen aufgelistet, die ein Menüsystem zur Steuerung der jeweiligen Funktion anbieten muss. Aus den genannten Funktionen und Typen kann eine Kernfunktionalität zusammengestellt werden, welche die Bedienung der Applikation ermöglicht.

## 5.3 Kernfunktionalität

Die Kernfunktionalität des VRGeo-Demonstrators ist in zwei Bereiche unterteilt. Zum Einen gibt es die 6-DOF Interaktion, die dem Benutzer die Kontrolle über die Anordnung der Elemente in der Szene gibt. Zum Anderen gibt es eine Reihe von Funktionen, die der Benutzer indirekt über ein Menüsystem steuern kann (siehe Tabelle 5.1). Nach der in Kapitel 3 vorgestellten Klassifikation können die Menüfunktionen durch Widgets aus folgenden Kategorien bereitgestellt werden: Aktivierung (Befehl), skalare Wertgeber (Wert), zwei Zustände (Checkbox), multiple Zustände (Radiobutton), sowie ein Container, innerhalb dessen die Elemente angeordnet werden können.

## 5.4 Menü-Implementierungen auf PDA und Tablet-PC

Bis dato wurden im VRGeo-Projekt zwei verschiedene Menüsysteme implementiert, welche die Menüs für den Demonstrator auf einem PDA und einem Tablet-PC dargestellt haben. Zur Implementierung wurde das Microsoft .NET Compact Framework verwendet. Der PDA läuft unter Windows CE, der Tablet-PC unter Windows XP. Die Kommunikation zwischen den per WLAN an das Netzwerk angebotenen Geräten und dem VR-Framework *Avango* (siehe Abschnitt 7.1) läuft über das *Remote User Interface*. In den folgenden Abschnitten werden die beiden Implementierungen jeweils kurz vorgestellt und die Erfahrungen aus der Entwicklung und aus den Rückmeldungen von informellen Benutzerstudien im Rahmen der VRGeo-Meetings aufgezeigt.

### 5.4.1 PDA

Der PDA dient als 6-DOF Eingabegerät (vgl. Abschnitt 2.3 auf Seite 10). Der Benutzer kann per Strahlinteraktion Objekte auswählen und frei im Raum bewegen. Das PDA-Display zeigt ein mit einem Stift bedienbares kontextabhängiges Menü. Als Widget-Container dient ein Full-Screen-Fenster mit Karteireitern am unteren Bildschirmrand, über deren Anwahl der Benutzer verschiedene Menüseiten zu den einzelnen Objektarten aufrufen kann (siehe Abbildung 5.2a). Es werden nur die im aktuellen Kontext verfügbaren Funktionen angezeigt: Die Reiter werden ausgeblendet, wenn ein entsprechendes Objekt nicht verfügbar ist. Auf den jeweiligen Menüseiten werden nicht verfügbare Funktionen ausgegraut dargestellt und können nicht bedient werden. Damit der Benutzer nicht permanent zwischen den Menüseiten wechseln muss, wird bei Selektion eines Objektes in der 3D-Szene automatisch die entsprechende Menüseite auf dem PDA aufgeschlagen. So hat der Benutzer beim Blickwechsel von der 3D-Interaktion auf das PDA-Display direkt die zum zuletzt ausgewählten Objekt passenden



(a) PDA



(b) Tablet-PC

**Abbildung 5.2:** Menüs auf PDA und Tablet-PC — (a) PDA mit Darstellung der Menüseite für Volumenlinsen. (b) Tablet-PC mit Darstellung des kompletten Demonstrator-Menüs.

Kontextinformationen und -funktionen auf dem Bildschirm. Er kann jedoch auf dem PDA über die Karteireiter und über Drop-Down-Menüs die Menüseiten der anderen Objekte aufrufen.

### 5.4.2 Erfahrungen mit Menüs auf einem PDA

Das PDA-Menü war die erste Implementierung eines Menüsystems im VRGeo-Demonstrator. Es erlaubt die vollständig interaktive Nutzung aller Funktionen, ohne Befehle in einem Terminal absetzen zu müssen.

Der PDA wird in der nicht-dominanten Hand gehalten. Das Menü auf dem PDA-Display wird mit Hilfe eines Stiftes mit der dominanten Hand bedient. Die 3D-Interaktion mit der nicht-dominanten Hand ist ungewohnt und oft langsamer und unpräziser als mit der dominanten Hand, so dass diese Interaktion von den Testnutzern als nicht optimal eingestuft wurde. Zudem ist der PDA als 3D-Interaktionsgerät etwas zu groß und unhandlich. Einige Benutzer hatten Probleme mit der Lesbarkeit des PDA-Displays durch die 3D-Brille: Schrift und Steuerelemente sind aus Platzgründen sehr klein dargestellt, das kontrastreiche und hintergrundbeleuchtete Display blendet in dunkler Umgebung und die Kombination aus Shutter und Polarisation der 3D-Brille zeigt den Display-Inhalt leicht flackernd und mit leichten Artefakten. Dies alles führt schnell zu einer spürbaren Augenermüdung. Zudem verrutschen die 3D-Brillen

beim Herabblicken auf den PDA. Der Erweiterbarkeit der Menüs sind auf dem PDA deutliche Grenzen gesetzt. So können ohne Übersichtsverlust kaum mehr als sechs oder sieben Karteireiter nebeneinander dargestellt werden und die Anzahl an Steuerelementen pro Seite ist ebenfalls begrenzt. Eine funktionale Gliederung der Elemente ist aus Platzgründen kaum möglich. Abhilfe schaffen nur Scroll-Balken oder zusätzliche Hierarchie-Stufen.

Problematisch ist bei den verwendeten PDA im Zusammenhang mit der .NET-Implementierung auch die Leistungsfähigkeit der Geräte. Die Übertragung von Daten per WLAN sowie häufige und komplexe Updates der Benutzeroberfläche erweisen sich bei der Benutzung als Engpass. Der Benutzer muss Teile seiner Interaktion an die Reaktionsgeschwindigkeit der grafischen Benutzeroberfläche anpassen.

Aus diesen Erfahrungen heraus wurde eine zweite Version der Demonstrator-Menüs auf Basis eines Tablet-PC entwickelt.

### 5.4.3 Tablet-PC

Für die Interaktion benötigt der Benutzer einen modifizierten 6-DOF-Stylus mit Tastern und eingebauter Stiftspitze, wie auf Seite 10 beschrieben. Ähnlich wie das PDA-Menü zeigt auch das Tablet-Menü immer den aktuellen Kontext des Benutzers. Da alle Funktionen in einem Full-Screen-Fenster auf dem größeren Display Platz finden, kann auf eine Unterteilung in objektbezogene Menü-Seiten verzichtet werden (siehe Abbildung 5.2b). Nicht verfügbare Funktionen werden ausgegraut dargestellt.

Die 3D-Interaktion wird mit dem Stylus in der dominanten Hand durchgeführt. Die Bedienung des Tablet-PC geschieht über die in den Stylus integrierte Stiftspitze. Der Tablet-PC kann wahlweise auf dem Arm getragen oder auf einem Pult vor dem Benutzer abgestellt werden. Durch die deutlich höhere Bildauflösung und das größere Format können Schrift und Steuerelemente größer dargestellt und funktional gruppiert werden.

### 5.4.4 Erfahrungen mit Menüs auf einem Tablet-PC

Die VR-Interaktion mit der dominanten Hand ist für den Benutzer intuitiver und angenehmer. Der Stylus als Interaktionsgerät ist von Form, Größe und Gewicht her für diese Aufgabe besser geeignet als der PDA. Die Bedienung des berührungsempfindlichen Tablet-PC-Displays mit der Stiftspitze am unteren Ende des Stylus ist jedoch gewöhnungsbedürftig, da der Stylus in der Hand leicht gedreht werden muss. Weil alle Funktionen des Demonstrators gleichzeitig auf dem Display Platz finden, entfällt das Auswählen des Objektcontextes wie auf dem PDA. So sind die Funktionen hierarchisch über- und untergeordneter Objekte jederzeit sicht- und bedienbar. Durch die statische und objektbezogene Unterteilung des Displays hat der Benutzer



eine erkennbare räumliche Referenz für die Funktionen. Für Verwirrung sorgt jedoch, dass im jeweiligen Kontext nicht nutzbare Funktionen zwar sichtbar sind, aber nicht ausgewählt werden können. Die ausgegraute Darstellung hebt sich nicht deutlich genug vom Rest der Benutzeroberfläche ab.

Die deutlich höhere Prozessor- und WLAN-Performance des Tablet-PC gegenüber dem PDA macht sich positiv bemerkbar. Die höhere Reaktionsgeschwindigkeit macht ein flüssigeres Arbeiten möglich und verringert das Gefühl, die eigene Arbeitsgeschwindigkeit an das Interaktionsgerät anpassen zu müssen.

Die Lesbarkeit der Schrift ist gegenüber der PDA-Darstellung deutlich verbessert, jedoch bleiben die Nachteile hinsichtlich Helligkeit, Kontrast, Flackern, Artefakten und rutschender Brille erhalten. Zusätzlich belastet der Tablet-PC mit seinem Gewicht und dem großen Formfaktor, wenn er während der Arbeit dauerhaft auf dem Arm gehalten wird. Die Nutzung eines Pultes reduziert diese Belastung. Der Benutzer ist dann vor dem VR-Display weniger mobil, was sich während der Demo-Sessions jedoch als unproblematisch herausstellte.

### 5.4.5 Immersionsverlust

Ein weiterer Nachteil sowohl bei PDA- als auch bei Tablet-PC-Menüs ist der notwendige Blickwechsel zwischen VR- und Menü-Display. Jeder Blickwechsel macht eine Neufokussierung und Neuorientierung notwendig. Durch diese Unterbrechungen entstehen drei verschiedene Modi, in denen der Benutzer mit der virtuellen Welt interagiert: Der erste Modus ist die reine VR-Interaktion, bei der der Benutzer vollständig auf die VR-Projektion fokussiert handelt. Er betrachtet virtuelle Objekte und kann diese direkt bzw. per Strahlinteraktion manipulieren. Im zweiten Modus ist der Benutzer auf das Menü-Display fokussiert. Er interagiert mit der 2D-Benutzeroberfläche und sieht das Ergebnis seiner Handlung auf dem Menü-Display. Der dritte Modus ist eine Mischform aus den beiden anderen. Der Benutzer ist in kurzen Abständen abwechselnd auf beide Displays fokussiert, interagiert mit einem Element des Menüs und betrachtet die Veränderungen wechselweise in der VR-Szene und im Menü. Zwischendurch interagiert er kurz in 3D und verändert dann wieder kontextbezogene Parameter im Menü. Der ständige Wechsel erfordert einen hohen Grad an Aufmerksamkeit und eine ständige Neuorientierung des Benutzers und führt letztendlich zu einem zeitweiligen Immersionsverlust. Wie in Kapitel 2 beschrieben, ist Immersion das Gefühl des Eintauchens in eine künstliche Welt, sei es ein spannendes Buch, ein Film, ein Computerspiel oder eine VR-Anwendung. Je mehr Sinne daran beteiligt sind, desto höher der Immersionsgrad. Durch die aktive Teilnahme an der Welt kann der Grad der Immersion weiter erhöht werden, so z.B. durch die Kontrolle der Perspektive durch Head-Tracking und die Interaktion mit virtuellen Objekten. Störende Einflüsse von außerhalb dieser Welt können den Grad der Immersion jedoch genauso negativ

**Tabelle 5.2:** Mögliche Gründe für Immersionsverlust

<b>Physikalische Eigenschaften</b>	
Haptik, Gewicht	Reale Gegenstände sind im Gegensatz zu virtuellen Objekten ertastbar und haben ein Gewicht. Sie gehorchen physikalischen Gesetzen, etwa der Schwerkraft, und können nicht frei im dreidimensionalen Raum platziert werden.
<b>Visuelle Darstellung</b>	
Beleuchtung	Reale und virtuelle Objekte werden von unterschiedlichen Lichtquellen angeleuchtet und reflektieren das Licht in unterschiedlicher Form. Auch eine gegenseitige Spiegelung ist normalerweise nicht möglich.
Helligkeit, Kontrast	Die visuelle Wahrnehmung realer Objekte kann einen wesentlich höheren Kontrastumfang annehmen, als dies bei der Darstellung virtueller Objekte aktuell möglich ist. Dies gilt nicht nur für reflektive Materialien, sondern insbesondere auch für reale Lichtquellen (z.B. auch ein PDA/Tablet-PC-Display).
Detailauflösung	Virtuelle Darstellungen werden, abhängig von der Display-Technik, oft mit einer deutlich geringeren Bildpunktdichte aufgelöst, als sie unser Auge wahrnehmen kann. So erscheinen virtuelle neben realen Objekten als unscharf oder grob gerastert.
Verdeckung	Virtuelle Objekte können keine realen Objekte verdecken. Dagegen verdecken reale Objekte die virtuellen auch dann, wenn das reale Objekt räumlich hinter dem virtuellen positioniert ist.
Display-Grenzen	Die virtuelle Welt wird in einem durch die Display-Dimensionen begrenzten Raum dargestellt. Außerhalb dieser festen Grenzen werden die Objekte weggeschnitten. Reale Objekte sind jedoch auch dort sichtbar und erzeugen so an der Kante einen Bruch zwischen realer und virtueller Welt.
Fokus	Um ein virtuelles Objekt zu betrachten, muss das Auge auf das VR-Display fokussieren anstelle das Objekt an seiner räumlichen Position zu fokussieren. Werden reale und virtuelle Objekte in gleicher Entfernung dargestellt, kann das Auge nicht beide gleichzeitig scharf sehen.
<b>Interaktion</b>	
Reaktion	Virtuelle Objekte können begrenzt auf (getrackte) reale Objekte reagieren, jedoch können reale Objekte nicht durch virtuelle Objekte beeinflusst werden.
Reaktionsgeschwindigkeit	Die Framerate schwankt oder ist zu langsam für die realistische Darstellung von Bewegungen. Zeitverzögerungen stören den Eindruck der „Echtheit“ oder „Direktheit“ der Interaktion.
Interaktionsart	Die Interaktion mit realen und virtuellen Objekten ist nicht identisch, z.B. werden virtuelle Objekte per Strahlinteraktion bewegt, während reale Objekte mit der Hand gegriffen und bewegt werden.

beeinflussen wie eine fehlerhafte Darstellung der Welt selbst. So nennen [Robertson et al. 1997] beispielsweise die Display-Auflösung, Bildwiederholrate und die Zeitverzögerungen bei der Umsetzung von Bewegungen als Ursachen für Immersionsverlust.

Es gibt eine Reihe weiterer Aspekte, die den Immersionsverlust begünstigen. Wenn der Benutzer gleichzeitig den sensorischen Eindrücken von virtuellen und realen Objekten ausgesetzt ist und diese Eindrücke nicht miteinander harmonieren, wird dies als Störung der Immersion wahrgenommen. Auslöser können unterschiedliche physikalische Eigenschaften, die visuelle Darstellung oder die Interaktionsart sein. In Tabelle 5.2 werden einige der möglichen Aspekte vorgestellt. Es ist davon auszugehen, dass ein hoher Grad an Immersion nur gegeben ist, wenn die Illusion der künstlichen Welt störungsfrei wahrgenommen werden kann. Bietet sich dem Benutzer der direkte Vergleich der virtuellen mit der realen Welt, können Inkonsistenzen in einem oder mehreren der genannten Aspekte zu einem Bruch der Illusion und somit zu verminderter Immersion führen.

## 5.5 Anforderungen an das Menüsystem

In den vorangegangenen Abschnitten wurden zahlreiche problematische Sachverhalte erläutert, die bei der Benutzung von PDA- oder Tablet-PC-Menüs entstehen. Die Anforderungen an ein neues Menüsystem sollen in erster Linie diese Kritikpunkte aufgreifen. Daher lautet eine grundlegende Vorgabe für die neuen Menüs, dass diese vollständig in die immersive VR-Applikation integriert werden sollen. Dies bedeutet, dass die Menü-Interaktion mit den im Demonstrator verwendeten 6-DOF-Eingabegeräten auskommen muss. Weitere spezialisierte Geräte, wie etwa ein PDA, sollen nicht zum Einsatz kommen. Dies hat einige direkte Verbesserungen gegenüber der Tablet- und PDA-Interaktion zur Folge: Sowohl die VR-Interaktion als auch die Menü-Interaktion werden über einen Stylus in der dominanten Hand realisiert. Die 3D-Interaktion ist für die Benutzer daher angenehmer und außerdem schneller und präziser als mit dem PDA. Auch das Umgreifen des Stylus zur Bedienung des Tablet-PCs entfällt. Da die komplette Interaktion innerhalb der VR-Szene stattfindet, ist die Gefahr des Immersionsverlustes durch die Bedienung von Tablet-PC und PDA nicht mehr gegeben.

Mit der Implementierung von immersiven VR-Menüs ergeben sich jedoch neue Herausforderungen, die in den Anforderungen adressiert werden müssen. [Lemoine et al. 2003] fassen in der Auswertung einer Benutzerstudie zur Interaktion mit 3D-Menüs einige interessante problematische Aspekte zusammen, die es bei einer Neuentwicklung zu beachten gilt:

- *3-DOF-Interaktion*: Das direkte Berühren eines virtuellen Elementes mit der Hand erweist sich für die Benutzer als schwierig (vgl. Seite 25). Eine schlechte Tiefenwahrnehmung aufgrund der Menügestaltung erschwert diese Aufgabe zusätzlich.

- *Größe*: Für manche Benutzer sind die in der Studie getesteten Menüs zu weit entfernt und zu groß. Gleichzeitig ist der Abstand zwischen den virtuellen Schaltflächen zu klein. Als problematisch erweist sich dabei das ungenaue Tracking.
- *Position*: Menüs auf Kopfhöhe sind schwierig zu bedienen und sorgen für Muskelermüdungen, da der Arm während der Interaktion hoch gehalten werden muss. Die Benutzer haben daher den Wunsch, das Menü zu verschieben.
- *Verdeckung*: Teile des Menüs werden manchmal von anderen Objekten verdeckt und sind dadurch kaum zu bedienen.
- *Textlesbarkeit*: Der Menü-Text ist aus ungünstigen Winkeln und bei größerer Entfernung schlecht oder gar nicht lesbar.

Lemoine et al. kommen nach ihrer Studie zu dem Schluss, dass Darstellung und Interaktion möglichst einfach verständlich, simpel und intuitiv sein sollen und sich sowohl für Experten als auch für Anfänger eignen müssen. [Bowman et al. 2001] empfehlen in ihrem Buch, das Design von 3D-Benutzerschnittstellen auf Erkenntnissen der 2D-Interface-Forschung aufzubauen und insbesondere auf Einfachheit, Konsistenz, Rückmeldung sowie Fehlervermeidung zu achten. Dabei müsse auf spezifische Eigenschaften der 3D-Interaktion Rücksicht genommen werden.

Nach den eigenen Erfahrungen mit den PDA- und Tablet-PC-Menüs im VRGeo-Demonstrator sowie Erkenntnissen aus Studien und Hinweisen aus der Literatur ergeben sich eine Reihe von Anforderungen an das neu zu gestaltende Menüsystem. Neben der Bereitstellung der Kernfunktionalität soll im Entwurf besonders auf Usability-Aspekte geachtet werden.

- *Immersives VR-Menü*: Das Menü soll innerhalb der 3D-Szene des VRGeo-Demonstrators dargestellt werden und keine zusätzlichen speziellen Menü-Interaktionsgeräte benötigen.
- *Bedienung mit 6-DOF Stylus*: Das für den Demonstrator verwendete Interaktionsgerät wird auch für die Bedienung der VR-Menüs benutzt. Die Interaktion mit dem Menü darf nicht im Widerspruch mit der bisherigen Interaktion mit dem Demonstrator stehen.
- *Widgets/Funktionalität*: Das Menü soll nicht nur in der Lage sein, Befehle abzusetzen, sondern auch andere Widgets für die Manipulation von Parametern bereitstellen. Zu den geforderten Funktionen zählen Befehle, Listenauswahl, Checkbox, Radiobutton, Wertgeber, Info-Text, Texteingabe und hierarchische Untermenüs.
- *Ergonomische Interaktion*: Die Interaktion mit und die Manipulation von Menü-Einträgen soll intuitiv und ergonomisch funktionieren.
- *Positionierung*: Das Menü soll in der VR-Szene auf eine Art und Weise positioniert werden, dass es nicht von anderen Objekten verdeckt wird und jederzeit vollständig

bedienbar bleibt. Gleichzeitig soll es so wenig wie möglich von der Szene verdecken. Der Benutzer muss in der Lage sein, das Menü zu verschieben.

- *Textlesbarkeit*: Die Lesbarkeit von Beschriftungen im Menü soll jederzeit gewährleistet sein.
- *Gestaltung*: Das Design soll einfach, verständlich und informativ sein.
- *Rückmeldung*: Durch geeignete Verfahren soll dem Benutzer Sicherheit und Bestätigung bei der Interaktion gegeben werden.
- *Zielgruppe*: Für Anfänger soll das Menü einen leicht verständlichen Einstieg in die jeweilige Anwendung ermöglichen, gleichzeitig für Experten genug Möglichkeiten zum effizienten und produktiven Arbeiten bieten.

Darüberhinaus gibt es technische Voraussetzungen, um die 3D-Menüs innerhalb einer VR-Anwendung einsetzen zu können: Die Bedienung der 3D-Menüs fordert eine durchweg interaktive Bildwiederholrate. Je nach dargestellten Datensätzen und der Rendering-Qualität können die Bildraten im VRGeo-Demonstrator zeitweise bis auf wenige Bilder pro Sekunde absacken und damit die Interaktion mit den 3D-Menüs erschweren. Externe Menüs wie PDA und Tablet-PC laufen hingegen asynchron zur VR-Anwendung und können in solchen Fällen problemlos bedient werden. Für solch kritische VR-Anwendungen existieren jedoch bereits Lösungsansätze, welche die Darstellung der 3D-Interaktionselemente (z.B. Strahl und Menü) vom zeitkritischen Rendering der restlichen 3D-Szene trennen. Ein solches Verfahren ist das sog. *Multi-Frame-Rate-Rendering* [Springer et al. 2007b, a]. Bei der Konzeption des Menüs wird davon ausgegangen, dass die Bedingung einer interaktiven Bildrate erfüllt ist, da dies eine Grundvoraussetzung für das Funktionieren von Interaktion in VR ist.



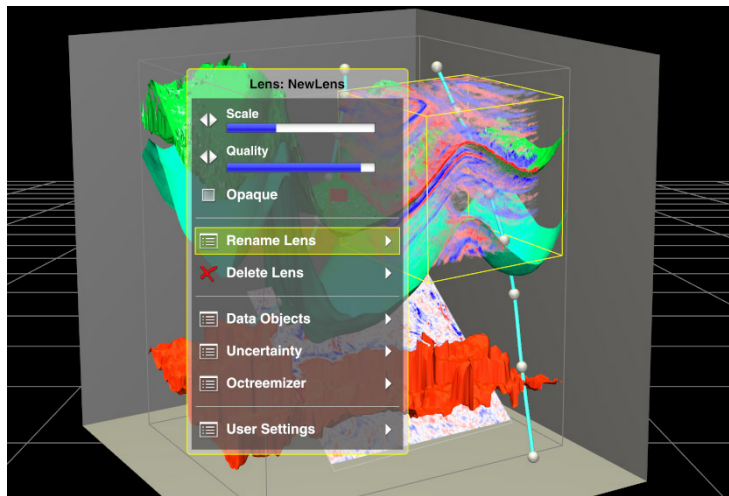
# 6 Entwurf eines anforderungsgerechten 3D-Menüsystems

Im vorangegangenen Kapitel wurden auf Grundlage der Erfahrungen im VRGeo-Projekt sowie aus einer Benutzerstudie von Lemoine et al. die Anforderungen für das neue VR-Menü definiert. Im Folgenden wird der Entwurf eines anforderungsgerechten Menüsystems vorgestellt. Die Konzeption des Menüs orientiert sich an der in Kapitel 3 vorgestellten Klassifizierung. Die konkrete Ausarbeitung und Implementierung des Entwurfs folgt in Kapitel 7.

## 6.1 Containerform

Eine grundlegende Entscheidung bei der Konzeption betrifft die Frage nach der Containerform des Menüs. In Kapitel 4, insbesondere in den Abschnitten 4.3 bis 4.5, wurden eine Reihe von Menüsystemen vorgestellt. Dabei ist festzustellen, dass keine der in VR verfügbaren Menülösungen vollständig die Anforderungen erfüllt. In den meisten Fällen fehlt die Möglichkeit, neben Befehlen auch skalare Werte einzugeben. Darüber hinaus sind viele Menüs symbolbasiert und bieten keine Möglichkeiten für textuelle Information oder Beschriftung, unterstützen keine Hierarchien, sind in der Anzahl der Elemente beschränkt oder benötigen spezielle Interaktionsformen oder -geräte. Einzig die portierten 2D-Menüs bieten in Sachen Funktionalität und Vielfalt eine umfassende Lösung. Lediglich hinsichtlich ergonomischer Bedienung, Verdeckung und Lesbarkeit in VR gibt es Einschränkungen. Diese müssen bei der weiteren Konzeption des 3D-Menüs berücksichtigt werden.

In einer Vergleichsstudie von [Gerber u. Bechmann 2004] bevorzugten die Benutzer bei einer Selektionsaufgabe mittels Ringmenü,  $C^3$  und 3D-Fenster (ein 2D-Menü als Ebene im Raum) die Strahlinteraktion mit dem 2D-Menü (für eine Erläuterung dieser Menüarten siehe Kapitel 4). Die Performance dieses 2D-Menüs war besser als mit dem  $C^3$  und gleichauf mit dem Ringmenü. Bemerkenswert ist bei diesem Test, dass das 2D-Menü selbst bei doppelter Anzahl der Freiheitsgrade (2-DOF im Vergleich zum 1-DOF-Ringmenü) und größerer Anzahl der dargestellten Menüelemente (25 vs. 10) gleich schnell war. Für ein angepasstes 2D-Menüsystem in VR spricht außerdem, dass durch die weite Verbreitung auf Desktopsystemen die Benutzer



**Abbildung 6.1:** *Context-Controls im VRGeo-Demonstrator* — Auf dem Bild ist ein Context-Control für eine Volumenlinse im VRGeo-Demonstrator zu sehen. Einige Parameter der Linse lassen sich direkt über die zwei Slider und eine Checkbox verändern. Darunter sind eine Reihe von Untermenüs angeordnet. Die Sicherheitsabfrage der Löschfunktion – üblicherweise über ein Dialogfenster realisiert – wird hier ebenfalls als Untermenüpunkt eingebunden. Die komplette Programmsteuerung wird einheitlich über diese Container-Form realisiert.

mit dem prinzipiellen Aufbau und der Benutzung vertraut sind und diese daher gerade für Anfänger leichter verständlich sind.

### 6.1.1 Das „Context-Control“-Konzept

Kontextmenüs eignen sich insbesondere, um dem Benutzer anstelle überladener globaler Menüs kleinere Menüeinheiten mit weniger Elementen anbieten zu können. Die Inhalte dieser Menüs beziehen sich nur auf den aktuellen Nutzungskontext, also z.B. auf die Interaktionsart und auf Eigenschaften des ausgewählten Objekts. Kontextmenüs werden als Pop-Up-Menü oder Palette dargestellt. In einem Pop-Up-Menü sind üblicherweise nur Befehle enthalten. In einer Palette können neben Befehlen auch eine Reihe von Steuerungselementen enthalten sein. Um ein einheitliches Menükonzept für alle Nutzungskontexte zu ermöglichen, wird hier das Konzept des „Context-Control“-Menüs neu eingeführt.

Das Context-Control-Menü ist eine Mischform aus Control-Palette und Kontextmenü: Es bietet einerseits die Funktionalität eines Kontextmenüs, welches im Kontext einer Situation oder eines Objektes auf Knopfdruck die gerade gültigen Befehle in einem Pop-Up-Menü auflistet. Andererseits können gleichberechtigt mit den Befehlen auch Parameter dargestellt und verändert werden (siehe Abbildung 6.1). Die Containerform als schwebende Palette mit Fensterrand erlaubt dabei auch die Nutzung als normale Palette (z.B. als Werkzeugkasten oder globales Menü) sowie das Verschieben des Menüs durch den Benutzer.



### 6.1.2 Interaktion mit dem Container

Das grundlegende Interaktionsmodell der Applikation ist durch den Expertenkontext vorgegeben: Mittels eines Stylus-artigen Eingabegerätes interagiert der Benutzer per Strahlinteraktion mit den Objekten. Der Stylus muss über mindestens zwei Tasten verfügen. Die erste Taste dient der Selektion und Manipulation von 3D-Objekten, die zweite Taste ruft auf einem Objekt das Kontextmenü auf. Da das Menü selbst auch ein 3D-Objekt ist, kann es über die erste Taste am Rand angefasst, verschoben und gedreht werden, wobei die Roll-Achse fixiert ist, so dass das Menü immer aufrecht dargestellt wird. Die Interaktion mit den Menüeinträgen geschieht wahlweise mit einer der beiden Tasten. Ein Klick auf ein anderes 3D-Objekt oder in den leeren Raum schließt das Kontextmenü.

Wird das Menü als schwebende Palette in die Applikation integriert, bleibt das Menü dauerhaft sichtbar und kann mit der ersten Maustaste bedient werden. Eine zweite Taste ist dann nicht erforderlich.

## 6.2 Menüelemente

Im Zusammenhang mit den Menüelementen müssen folgende Fragen geklärt werden: Welche Widgets werden angeboten? Wie werden diese angeordnet? Wie wird mit den Elementen interagiert?

### 6.2.1 Blockzentriertes Layout

Die Anordnung und Darstellung der Widgets ist maßgeblich für die Übersichtlichkeit, Verständlichkeit und Geschwindigkeit der Interaktion. Bowman et al. empfehlen die Anwendung von Design-Aspekten aus 2D-Benutzerschnittstellen auch für VR-Menüs, unter anderem Einfachheit und Konsistenz [Bowman et al. 2001]. In Kapitel 4 wurde erklärt, dass Menüinteraktionen auf möglichst wenigen Freiheitsgraden beruhen sollen und dass große Objekte schneller erreicht werden können als kleine. Dies führt zu der Entscheidung, die Darstellung aller Widgets weitestgehend zu vereinheitlichen. Die Inhalte des Menüs werden als eindimensionale Liste untereinander dargestellt. Dies bietet die Möglichkeit, die Interaktion auf ein Minimum an Freiheitsgraden zu reduzieren, alle Elemente gleichartig darzustellen sowie eine übersichtliche Beschriftung der Elemente linksbündig untereinander zu gewährleisten. Alle Einträge bilden gleichgroße Blöcke, die nach Fitts' Law entlang einer Achse schnell zu erreichen sind (vgl. Abschnitt 4.1). In der Art der Anordnung unterscheidet sich das Context-Control-Menü praktisch nicht von herkömmlichen 2D-Drop-Down-Menüs, mit dem Unterschied, dass neben Befehlen in dieser Form auch Parameter beeinflusst werden können (vgl. Abbildung 6.1 und 6.3).

## 6.2.2 Darstellung der Widgets

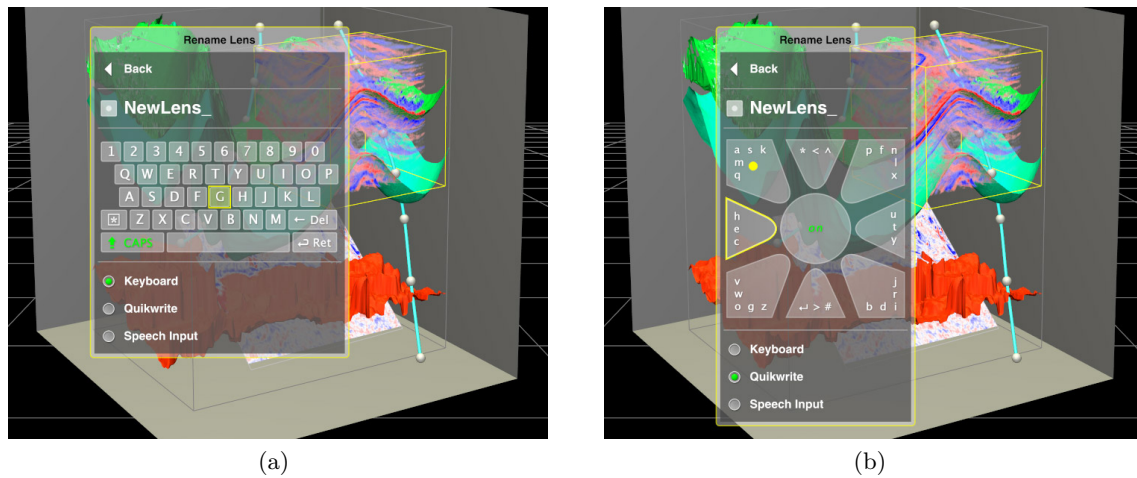
In Abschnitt 5.5 wurden die benötigten Widgets aufgelistet. Jedes Widget erhält eine aussagekräftige und möglichst selbsterklärende Beschriftung. Um die Widgets bei ansonsten identischem Layout unterscheiden zu können, wird jeder Eintrag mit einem Icon ausgestattet. Diese Icons sollten nach Möglichkeit symbolhaft, selbsterklärend und eindeutig in ihrer Aussage sein, zudem auch in unterschiedlichen Größen und Perspektiven gut zu erkennen sein. Sie bilden das Hauptunterscheidungsmerkmal der verschiedenen Widget-Arten. Während normale Befehle ein mit dem Befehl assoziiertes Symbol erhalten, bekommen die anderen Widget-Arten wie etwa Checkboxes und Radiobuttons Standard-Icons, die der Elementform aus 2D-GUIs nachempfunden sind. Einträge, die ein Untermenü kennzeichnen und daher eine hierarchisch untergeordnete Menüstufe öffnen, erhalten ein Menü-Icon und zusätzlich am rechten Rand ein nach rechts zeigendes Dreieck (vgl. Abbildung 6.3a auf Seite 52). Auch diese Form der Kennzeichnung ist aus 2D-Menüsystemen bekannt. Während dort jedoch Menüpunkte, die ein Dialogfenster öffnen, meist durch drei nachgestellte Punkte gekennzeichnet werden (z.B. „Speichern als. . .“), wird ein solcher Eintrag im neuen Menü ebenfalls als Untermenü dargestellt, da auch die Funktionalität von Dialogfenstern innerhalb des Menüsystems abgebildet wird.

Lediglich spezielle Bedienelemente, die eine komplexere Interaktion oder grafische Repräsentation benötigen, erhalten ein vom Standard (Icon + Text) abweichendes Layout. Dazu gehört neben Widgets zur Texteingabe z.B. der skalare Wertgeber in Form eines Schiebereglers. Dieser enthält unterhalb des Textes zusätzlich einen Balken, welcher den Wertebereich darstellt, sowie die Markierung des aktuell eingestellten Wertes.

Für die Texteingabe sind zwei Widgets vorgesehen: Die virtuelle Tastatur enthält ein reduziertes Tastaturlayout und einige Sondertasten als Buttons zum Anklicken. Diese sind im Vergleich zu den sonstigen Menüeinträgen relativ klein, aber groß genug für eine zügige und sichere Bedienung per Strahlinteraktion. Das gewohnte Tastenlayout kommt dem Benutzer im Erlernen dieser Texteingabe entgegen (Abbildung 6.2a). Die zweite Variante ist eine Adaption der „Quikwriting“-Texteingabe von Ken Perlin (Abbildung 6.2b). Diese kommt mit wenigen, großen, um ein Zentrum herum angeordneten Elementen aus und wird über Gesten bedient [Perlin 1998; Quikwriting].

## 6.2.3 Interaktion mit den Widgets

Wie oben beschrieben, können die Widgets mit einer der beiden Stylus-Tasten bedient werden. Das blockzentrierte Layout vergrößert die Interaktionsfläche jedes Eintrags auf seine volle Höhe und die volle Menübreite. So ist sichergestellt, dass alle Elemente durch eine 1-DOF-Bewegung erreicht werden können und unabhängig von ihrer jeweiligen Textlänge gleich groß

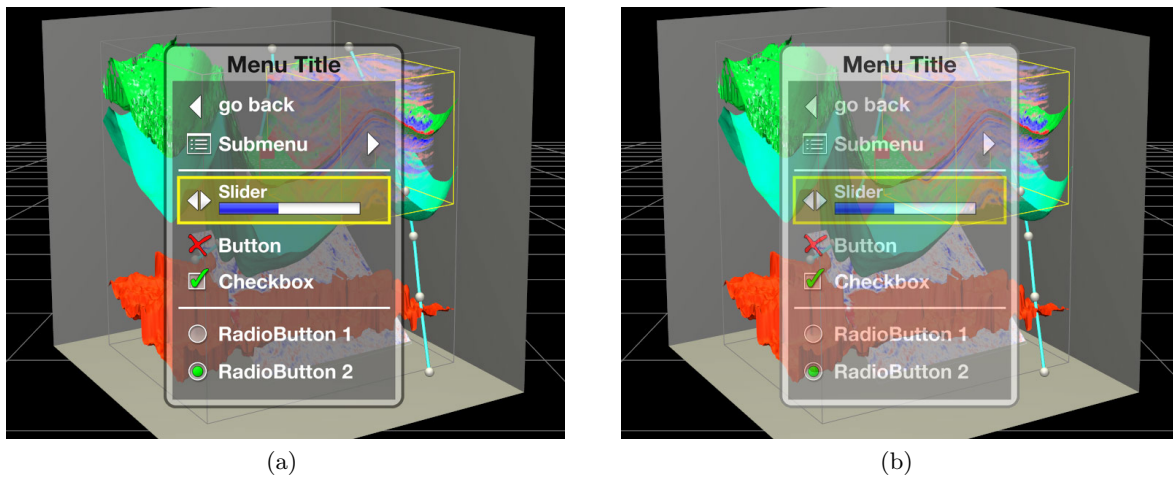


**Abbildung 6.2:** *Texteingabe-Widgets im 3D-Menü* – Das 3D-Menü bietet zwei spezielle Bedienelemente für die Texteingabe. Die im Menü aufgelistete dritte Variante der Spracheingabe befindet sich im VRGeo-Projekt in der Entwicklung (vgl. Abschnitte 8.4 und 9) (a) Die virtuelle Tastatur ermöglicht die Texteingabe über ein gewohntes Tasten-Layout. Die Tasten können per Strahlinteraktion angewählt werden. (b) Die „Quikwriting“-Texteingabe nach Ken Perlin wird über Zeige-Gesten gesteuert und bedarf einiger Übung [Perlin 1998; Quikwriting].

sind. Widgets, die der Parameteränderung dienen, verändern durch Anklicken ihr Aussehen. So werden bei Checkboxes und Radiobuttons z.B. die Icons durch Symbole ersetzt, die den Markierungszustand des Eintrags widerspiegeln. Im Schieberegler verändert sich entsprechend die Markierung für den aktuellen Wert (vgl. Abbildungen 6.1 und 6.3).

Auch der Schieberegler wird als Block gegriffen. Das sonst übliche Greifen eines kleinen Anfassers auf der Skala des Reglers entfällt und beschleunigt so nach Fitts' Law das Fassen des Reglers.

Die strahlbasierte Interaktion mit einem Schieberegler unterliegt in VR verschiedenen Störeinflüssen: Die fehlende physikalische Unterstützung der Hand und die von einem auf zwei Freiheitsgrade erhöhte Komplexität erschweren die Interaktion (siehe Abschnitt 4.3.1). Hinzu kommt, dass unterschiedliche Entfernungen zwischen Benutzer und Menü die Interaktionsgenauigkeit mit dem Regler beeinflussen. Um diese Einflüsse zu eliminieren und damit die Interaktion von der Darstellung des Reglers zu trennen, wird die Interaktion mit dem Schieberegler auf eine 1-DOF-Rotation des Handgelenks in horizontaler Ebene begrenzt. D.h. der Benutzer kann den Regler-Block greifen und dann durch Handgelenksrotation nach rechts oder links den Wert verändern. Die genaue Vorgehensweise ist bei der Implementierung in Abschnitt 7.4 beschrieben.



**Abbildung 6.3:** 3D-Menü mit verschiedenen Widgets und Transparenzen – Im Menüpanel sind die Widget-Arten Button, Submenü, Schieberegler, Checkbox sowie eine Gruppe mit Radiobuttons dargestellt. Im Beispiel ist der Slider aktiviert und mit einem Highlight hinterlegt. Im Gegensatz zu herkömmlichen Schiebereglern muss kein kleiner Anfasser gegriffen werden, sondern jedes Widget besteht aus einem großen Block in einer Liste. (a) Schrift und Icons werden opak auf einem semitransparentem Panel angezeigt und erzeugen einen hohen Kontrast zum Hintergrund. (b) Wird das gesamte Menü gleichermaßen transparent dargestellt, wird der Kontrast und damit die Lesbarkeit der Schrift reduziert.

### 6.3 Verdeckungsprobleme

Beim Platzieren virtueller Objekte in einer 3D-Szene entstehen aus Sicht des Benutzers verschiedenartige Verdeckungsprobleme: Das hinzugefügte Objekt kann dahinterliegende Objekte verdecken, sich mit diesen schneiden oder selbst von anderen Objekten verdeckt werden. Für ein Menü gibt es daher in vielen Fällen keine optimale Position in der Szene. Mit verschiedenen Lösungsansätzen sollen diese Probleme jedoch weitestgehend minimiert werden.

#### 6.3.1 Transparenz

Da ein Kontextmenü inhaltlich an ein Objekt gebunden ist, wird das Menü üblicherweise in unmittelbarer Nähe des Objektes dargestellt. Der Benutzer kann so auf den Ort seiner Interaktion fokussiert bleiben. Wird nun das Menü vor dem Objekt platziert, ist die vollständige oder teilweise Verdeckung des Objektes durch das Menü sehr wahrscheinlich. In üblichen Kontextmenüs aktiviert der Benutzer einen Befehl, woraufhin das Menü verschwindet. Das Context-Control-Menü bietet jedoch vielfältigere Steuerungsmöglichkeiten und wird daher oft nicht nach einmaliger Aktivierung eines Eintrags ausgeblendet. Zusätzlich können veränderbare Parameter die Eigenschaften des Objektes beeinflussen, so dass der Benutzer eine visuelle

Kontrolle der Veränderungen wünscht. Durch einen semitransparenten Menühintergrund bleibt die Sicht auf das Objekt für viele praktische Anwendungen ausreichend erhalten. Durch eine opake Darstellung der Schrift auf dem halbtransparenten Hintergrund bleibt die Lesbarkeit der Schrift dank eines hohen Kontrastes erhalten (siehe Abbildung 6.3a). Wird ein 2D-Menü lediglich als Textur dargestellt, ist diese Form der Transparenz ohne zusätzlichen Aufwand nicht möglich (vgl. Qt3D im Abschnitt 4.4.1). Eine vollständig halbtransparente Menütextur hätte einen verminderten Kontrast und wäre schlechter lesbar (Abbildung 6.3b).

### 6.3.2 Regelbasierte, aktive Positionierung

In der Literatur werden verschiedene Möglichkeiten der Positionierung von Menüs in der 3D-Szene vorgestellt (siehe Abschnitt 3.4.3) [Jacoby u. Ellis 1992; Bowman et al. 2005; Dachselt u. Hübner 2007]. Die Position des Menüs bezieht sich dabei jeweils auf ein Referenzkoordinatensystem:

- *Weltreferenzierte Menüs* erscheinen immer an der gleichen Position in der virtuellen Welt. Optional kann der Benutzer diese Position frei festlegen.
- *Objektreferenzierte Menüs* werden immer an der gleichen Stelle vor, über, neben oder unter einem Objekt dargestellt.
- *Kopfreferenzierte Menüs* schweben immer im gleichen Blickwinkel vor dem Kopf, bewegen sich also mit, wenn der Benutzer seinen Kopf bewegt.
- *Körperreferenzierte Menüs* stehen in einer festen Position relativ zu einem Körperteil, z.B. der Hand des Benutzers.
- *Displayreferenzierte Menüs* werden immer an der gleichen Stelle auf dem Display, meist in der Screenebene am Rand, dargestellt.

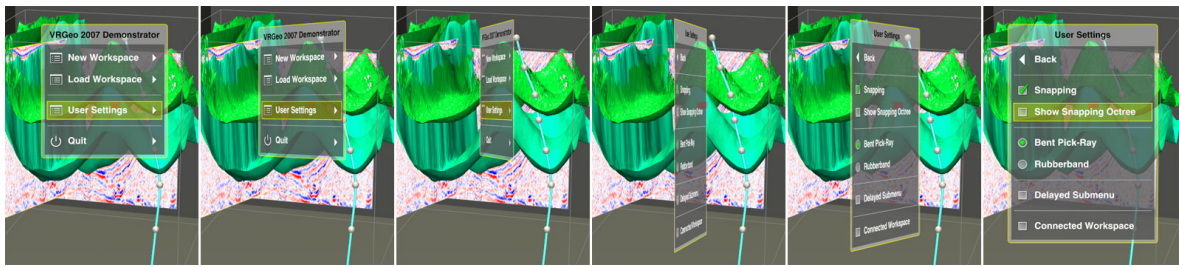
All diese Verfahren richten sich nach festen relativen Positionen und berücksichtigen neben der Referenz keine weiteren Objekte in der VR-Welt. Diese Verfahren werden im weiteren Verlauf als *passive Positionierung* bezeichnet. Diese passiven Verfahren führen zu verschiedenen Problemen: Das Menü wird von anderen Objekten verdeckt, schneidet diese, wird außerhalb des sichtbaren Bereichs dargestellt, ist für den Benutzer unerreichbar/nicht bedienbar (durch Verdeckung oder durch zu große Entfernung) oder nicht erkennbar (falscher Winkel, Schrift zu klein).

Aus diesem Grund wird ein *aktives Positionierungsverfahren* angestrebt. Dieses soll für die Sichtbarkeit, Bedienbarkeit und Erreichbarkeit des Menüs sorgen, wenn der Benutzer ein Kontextmenü anfordert. Das Regelsystem definiert, ausgehend von einer Referenz, verschiedene

dynamische Aspekte für die Positionierung. Das Verfahren wird detailliert in Abschnitt 7.5 erläutert.

- *Referenz*: Die Positionsbasis bildet der Punkt auf dem Objekt, an dem der Benutzer das Objekt angeklickt hat. Die gedachte Linie von dem Objektpunkt zum Augpunkt des Betrachters dient als „Schnur“, an der das Menü mit der oberen Kante aufgehängt wird. So hat das Menü aus Sicht des Benutzers eine optische Referenz zum angeklickten Punkt.
- *Entfernung*: Entlang dieser gedachten Linie wird das Menü vor dem Objektpunkt platziert, jedoch vom Benutzer nicht weiter entfernt als ein festgelegter maximaler Abstand. Dieses Maximum sorgt dafür, dass das Menü in einer Entfernung dargestellt wird, in welcher der Benutzer dieses bequem bedienen kann. Als Maximumbegrenzung kann z.B. auch die Bildebene des Displays dienen.
- *Winkel*: Das Menü wird im Raum so gedreht, dass der Benutzer von vorne auf die Fläche des Menüs schaut. Die Strahlinteraktion geht von der Hand aus und hat ihren Ursprung daher deutlich unterhalb der Augenhöhe. Um den Interaktionswinkel auf das Menü nicht zu flach zu gestalten, muss darauf geachtet werden, dass das Menü derart geneigt ist, dass sowohl der Blickwinkel als auch der Interaktionswinkel geeignet sind.
- *Größe*: Um in größerer Entfernung nicht zu kleine und in unmittelbarer Nähe nicht zu große Menüs darzustellen, wird die Größe des Menüs an die Entfernung angepasst. Je näher das Menü am Benutzer dargestellt wird, desto kleiner wird es. Der Blick- und Interaktionswinkel zum Menü soll dabei abhängig von der Entfernung unverändert bleiben. Ein weiter entferntes Menü kann daher genauso bequem bedient werden wie ein nah dargestelltes.
- *Textlesbarkeit*: Die Lesbarkeit des Textes im Menü soll jederzeit sichergestellt sein. Unterschreitet der Text eine leserliche Größe, wird das Menü um einen entsprechenden Faktor hochskaliert.
- *Verdeckungsprobleme*: Wird das Menü aus Sicht des Betrachters von Objekten verdeckt oder geschnitten, soll das Menü ausweichen und *vor* den störenden Objekten neu positioniert werden.
- *Sichtbarkeit*: Schlussendlich muss sichergestellt sein, dass das Menü vollständig im sichtbaren Bereich (*View-Frustum*) und nicht außerhalb der Displaybegrenzungen liegt. Ist dies nicht der Fall, wird das Menü soweit in das Frustum verschoben, dass es vollständig sichtbar wird.

Durch die beiden letzten Schritte wird die ursprüngliche Position des Menüs verändert, so dass eine Wiederholung der vorherigen Schritte (Winkel und Skalierung) notwendig ist. Die



**Abbildung 6.4:** *Hierarchienavigation durch Menü-Rotation* — Im Hauptmenü des VRGeo-Demonstrators wird der Menüpunkt „User Settings“ aktiviert. Das Menü-Panel rotiert in einer Animation auf die Rückseite und zeigt dort das entsprechende Untermenü. Mit dem „Back“-Eintrag gelangt der Benutzer zurück zum Ursprungsmenü. Die Animation läuft dann entsprechend rückwärts.

Positionsfindung wird daher in mehreren Schritten vorgenommen. Wird keine Position gefunden, die allen Kriterien entspricht, so sorgt der letzte Schritt dafür, dass durch das Verschieben in das View-Frustum wenigstens die Möglichkeit gegeben ist, Teile des Menüs zu sehen und ggf. das Menü von Hand zu verschieben.

### 6.3.3 Hierarchienavigation

Die Navigation in mehrstufigen Hierarchien verbraucht bei vielen Menüsystemen zusätzlichen Platz auf dem Bildschirm. So werden Untermenüs in 2D-GUIs meist neben dem aktuellen Menü gezeichnet. Wie im vorangegangenen Abschnitt erläutert, ist gerade die Platzierung neuer Objekte ein grundlegendes Problem in VR, so dass die Platzierung weiterer Untermenüs durch Verdeckungs- und Sichtbarkeitsprobleme nur eingeschränkt möglich ist. Daher wird an dieser Stelle ein Ansatz verfolgt, der keinerlei zusätzlichen Raum für die Navigation in Untermenüs beansprucht: Es ist immer nur das Menüfenster mit der aktuellen Hierarchiestufe zu sehen. Die Menühierarchien werden virtuell auf der Rückseite des Fensters dargestellt. Aktiviert der Benutzer ein Untermenü, wird das Menü auf die Rückseite rotiert (siehe Abbildung 6.4). Über eine „Zurück“-Schaltfläche kann das Fenster rückwärts rotiert werden und die vorherige Hierarchiestufe wird sichtbar. Die Untermenüs werden nur scheinbar auf der Rückseite dargestellt. Sie werden bei Auswahl eines Untermenüs erst in dem Moment dort platziert. So ist es möglich, beliebig tiefe Verschachtelungen darzustellen und durch die Hierarchien zu blättern. Der Raumbedarf des Menüs ändert sich dabei nur geringfügig, je nachdem wie breit und hoch das jeweilige Menü ist. Es wird jedoch niemals mehr als ein Menüfenster angezeigt.

## 6.4 Visuelles Feedback

Zur sicheren und fehlerfreien Aktivierung eines Menüeintrags gehört eine eindeutige Rückmeldung an den Benutzer, dass das Menü aktiv und der gewünschte Eintrag ausgewählt ist. Dank Strahlinteraktion bekommt der Benutzer ausreichende Rückmeldung über die Ausrichtung des Interaktionsgerätes und über den Schnitt des Strahls mit der Menüebene. Das „Berühren“ eines Menüeintrags mit dem Strahl wird über ein farbig hinterlegtes Highlight des gesamten Widget-Blocks bestätigt (siehe Abbildungen in diesem Kapitel).

Zusätzlich zur Markierung des Menüeintrags kann auch eine Rückmeldung innerhalb der 3D-Szene gegeben werden. Damit ist es möglich, dem Benutzer eine „Vorschau“ auf das Ergebnis seiner beabsichtigten Menüinteraktion zu zeigen. Beispielsweise kann neben einem Menü zur Erzeugung von verschiedenen 3D-Objekten beim Highlighting eines Eintrags bereits eine verkleinerte Version des Objekts dargestellt werden.

## 6.5 Farbgebung

Die Farbwahrnehmung in der virtuellen Welt soll durch das Menü möglichst wenig beeinflusst werden. Die Hauptelemente des Menüs – Hintergrund, Text und Icons – werden daher in neutralen Grautönen gestaltet. Der halbtransparente Hintergrund erlaubt die Sicht auf dahinterliegende Objekte und sorgt für einen komfortablen Kontrast zu Icons und Text. Die Form der Icons und die Beschriftung sollen die Funktionalität und die Zustände eindeutig beschreiben. In den Icons unterstützen Farben lediglich die schnellere Erkennbarkeit von Zuständen und Funktionen. So erhalten z.B. die Checkbox- und Radiobutton-Icons ein grün eingefärbtes grafisches Element zur Anzeige ihrer Aktivierung. Destruktive Funktionen, wie z.B. das Löschen eines Objektes, erhalten zur Warnung ein rot eingefärbtes Symbol.

Das Highlight eines Menüeintrags zeigt dem Benutzer den aktuellen Selektionszustand an. Die Signalfarbe gelb bildet einen starken Kontrast zum farblosen Menühintergrund, lenkt dabei den Fokus auf das jeweilige Element und hebt dieses aus der Liste der Einträge deutlich hervor.

Der virtuelle Schieberegler erhält eine opake, weiße Anzeige für den Wertebereich mit einer blauen Markierung für den eingestellten Wert. Diese Farben heben sich sowohl voneinander, als auch vom Hintergrund und Highlight deutlich ab.

Die gewählten Farbtöne grün, gelb, rot, blau, grau und weiß bilden durch ihren großen Abstand im RGB-Farbraum deutliche Kontraste und sind deshalb gut unterscheidbar.



## 6.6 Zusammenfassung der Menüeigenschaften

Die verschiedenen Gestaltungs- und Interaktionsaspekte werden abschließend in einer Übersicht zusammengefasst. Die Gliederung richtet sich nach der Klassifizierung aus Kapitel 3.

### Widgets

- *Container*: Das Menü-Fenster ist eine Kombination aus Palette und Pop-Up-Menü inkl. Titelzeile und Rand.
- *Anzahl der möglichen Einträge*: Die Länge des Menüs ist beliebig, solange der Platz im 3D-Raum/auf dem Display ausreicht. Eine Scroll-Funktion ist vorgesehen, jedoch hinsichtlich Performance und Übersichtlichkeit nicht empfehlenswert.
- *Hierarchie*: Es sind beliebig tiefe Hierarchieversachtelungen möglich. Es ist immer nur eine Hierarchiestufe sichtbar, so dass kein zusätzlicher Platz benötigt wird.
- *Steuerungselemente*: Das Menü kann Widgets aus den folgenden Bereichen enthalten: Information, Aktivierung (auch für Menühierarchien), binäre Schalter, multiple Schalter, skalare Wertgeber, Trennlinie, Texteingabe.

### Erscheinungsweise und Struktur

- *Geometrische Repräsentation*: Das Menü wird als zweidimensionales, halbtransparentes, graues Rechteck dargestellt.
- *Struktur der Anordnung*: Die Einträge sind als azyklische Liste untereinander angeordnet.
- *Darstellungsform*: Jeder Eintrag wird durch ein Icon und Text repräsentiert. Spezielle Einträge können weitere grafische Elemente enthalten. Die Farbgebung ist größtenteils neutral in Grautönen gehalten und wird durch farbige Elemente unterstützt, wenn diese zum Verständnis oder zur Übersichtlichkeit beitragen.
- *Größe und Abstand der Einträge*: Jeder Eintrag entspricht einem großen, rechteckigen Block auf voller Menübreite; die Einträge grenzen ohne Zwischenraum aneinander.

### Platzierung

- *Referenz*: Die aktive Positionierung nutzt den Selektionspunkt am Objekt und den Augpunkt des Benutzers als Referenzen. Ein Regelsystem steuert die Positionsfindung und Skalierung des Menüs.
- *Orientierung*: Das Menü wird in Richtung Benutzer ausgerichtet.

- *Neupositionierung*: Das Menü kann mittels Strahlinteraktion am Fensterrand angefasst und verschoben werden.
- *Dynamik*: Das Menü schwebt an fester Position im Raum und folgt keinem Referenzkoordinatensystem.

### Aufruf und Verfügbarkeit

- *Sichtbarkeit*: Das Kontextmenü ist nur zeitweise, für die Dauer der Benutzung, sichtbar.
- *Aufruf*: Durch Klick mit dem Kontext-Button (zweite Stylus-Taste) auf ein Objekt wird das Menü aufgerufen.
- *Animation*: Der Übergang zwischen Menühierarchien wird durch Rotation des Menüfensters angezeigt (vergleichbar mit dem Vor- und Zurückblättern in einem Buch).
- *Schließen*: Durch Klick in die 3D-Szene wird das Kontextmenü geschlossen.

### Interaktion

- *Eingabegerät*: Als Interaktionsgerät dient ein 6-DOF Stylus mit mindestens zwei Tastern.
- *Einsatzbereich*: Die Menügestaltung, Positionierung und Interaktion ist auf die Verwendung am TwoView-Display optimiert.
- *Dimensionalität*: Das Menüfenster ist zweidimensional, die Menüeinträge bilden jedoch eine eindimensionale Liste.
- *Selektion*: Der Schnittpunkt zwischen Selektionsstrahl und Menüebene bildet den Cursor für die Auswahl eines Widgets.
- *Auswahl*: Die Auswahl erfolgt durch Klick auf einen Stylus-Taster.
- *Feedback*: Der Hintergrund des selektierten Widgets wird gelb eingefärbt (Highlight). Das Vor- und Zurückblättern in der Hierarchie wird durch Rotation der Menüebene angezeigt. Parameteränderungen im Menü werden sofort in der Szene sichtbar.
- *Selektionspfad*: Über „Zurück“-Schaltflächen kann die vorherige Hierarchiestufe erreicht werden.

### Benutzerfreundlichkeit

- *Studien*: Die Konzeption des Menüs basiert auf dem Feedback von Benutzern im Rahmen der VRGeo-Demo-Sessions, auf Erfahrungen mit PDA- und Tablet-Menüs im VRGeo-Projekt sowie zwei Studien von Lemoine, Vexo, Thalman (siehe Abschnitt 5.5) und Gerber, Bechmann (Abschnitt 6.1).

- *Evaluationskriterien:* Eine Benutzerstudie zu den neu konzipierten Menüs wird in Kapitel 8 vorgestellt. Die Studie misst die Selektionsgeschwindigkeit und Manipulationsgeschwindigkeit bei Interaktion mit Schieberegler sowie die Fehlerrate. Die Benutzer beurteilen den Schwierigkeitsgrad der Menüsysteme und vergleichen diese miteinander. Dadurch kann die Akzeptanz der Menüformen sowie die Zufriedenstellung der Benutzer ermittelt werden.
- *Vergleich:* Das neue Menü wird mit einer an „herkömmliche“ Menüs angelehnten Abwandlung der Schiebereglerinteraktion sowie mit VR-Menüs auf PDA und Tablet-PC verglichen.



# 7 Implementierung des 3D-Menüsystems

In den beiden vorangegangenen Kapiteln wurden die Anforderungen an das Menüsystem vorgestellt sowie die Konzeption eines anforderungsgerechten Menüs erarbeitet. In diesem Kapitel werden nach einer Erläuterung des eingesetzten VR-Frameworks „Avango“ [Avango] die Konzepte und Implementierungen der wichtigsten Funktionen der 3D-Menü-Software beschrieben.

## 7.1 Das VR-Framework Avango

Wie in Abschnitt 2.5 bereits erwähnt, stellt ein VR-Software-Framework die Technologien zur effizienten Entwicklung von VR-Applikationen zur Verfügung. Neben der Anbindung an einen Szenengraphen und einen Renderer enthält das Framework Softwareschnittstellen zu verschiedenen Ein- und Ausgabegeräten und Möglichkeiten des Rapid-Application-Developments.

Avango ist ein objektorientiertes Virtual-Reality-Framework zur Erstellung interaktiver VR-Applikationen. Hauptkomponenten sind die auf OpenGL basierende Szenengraph-Programmierschnittstelle *Performer*, ein an *Inventor* angelehnter Field-Mechanismus, sowie ein auf der funktionalen Programmiersprache *Scheme* basierendes Scripting-Interface. Weitere Komponenten sind ein Netzwerk-Layer zur Realisierung verteilter Applikationen, eine Abstraktion für Ausgabegeräte und ein Device-Daemon zur Anbindung von Eingabegeräten und Tracking-Systemen [Avango; Tramberend 1999, 2003; Performer; Inventor].

### 7.1.1 Nodes und Fields

Der Szenengraph ist eine hierarchisch aufgebaute Repräsentation von Objekten in der 3D-Szene. Die einzelnen Szenengraph-Knoten sind Instanzen von Avango- oder Performer-Typen und werden als *Nodes* bezeichnet. Avango nutzt das *Field*-Konzept aus *Inventor*. Nodes und andere Objekte dienen als *Field-Container* und können so ihre Attribute als Fields von außen zugänglich machen. Fields und Field-Container unterstützen ein Streaming-Interface und können Avango-Objekte im Netzwerk verteilen. Dies macht es möglich, Avango-Applikationen auf entfernten Maschinen zur Remote-Kollaboration oder in Cluster-Systemen zu betreiben.

Avango-Fields repräsentieren einzelne Werte (*Single-Fields*) oder eine Liste von Werten (*Multi-Fields*), wobei die Werte typgebunden sind (z.B. Bool, Integer, Float, Vector, Matrix oder auch Node). Der Zugriff auf die Field-Werte geschieht über `getValue()`- bzw. `setValue()`-Funktionen, bei Multi-Fields zusätzlich über `add1Value()`- und `remove1Value()`-Funktionen.

Über *Field-Connections* können Verbindungen zur Benachrichtigung von Field-Änderungen hergestellt werden. Ändert sich der Wert des *Source-Fields*, wird der Wert an alle verbundenen Fields weitergegeben. Durch Field-Connections können logische Verbindungen abgebildet werden, die allein über den Szenengraphen nicht möglich wären.

In jedem Frame wird der durch die Field-Connections stattfindende Datenfluss evaluiert. Beginnend mit den Sensor-Knoten (z.B. für Eingabegeräte, Zeitgeber etc.), die ihre Werte über Fields der Applikation zur Verfügung stellen, werden die Daten an sog. Effektor-Knoten (z.B. Avango-Objekte, Nodes) weitergereicht. Diese repräsentieren die eigentliche Applikation, reagieren auf die geänderten Field-Werte und können ihrerseits anderen Objekten über Field-Änderungen neue Werte zur Verfügung stellen. Am Ende der Benachrichtigungskette werden die Aktuatoren (z.B. die virtuelle Kamera) mit den aktuellen Werten versorgt. Diese stellen selbst keine Werte zur Verfügung, sondern reagieren lediglich auf Field-Änderungen.

### 7.1.2 Scripting

Bei der Applikationsentwicklung in Avango kann auf eine Vielzahl an vorgefertigten Komponenten zurückgegriffen werden. Weitere Komponenten und Nodes werden in C++ implementiert. Die weitere Entwicklung der interaktiven Anwendung kann auf zwei Arten geschehen: Zum einen kann die Implementierung sämtlicher Funktionalität weiterhin in C++ stattfinden. Zum anderen bietet Avango mit der an *Lisp* angelehnten funktionalen Programmiersprache *Scheme* die Möglichkeit der schnellen, prototypischen Entwicklung ganzer VR-Anwendungen. Dabei werden die Avango-Nodes in Scheme instanziiert, in den Szenengraphen integriert und über Field-Connections verbunden. Das Scheme-Script reagiert dabei typischerweise während der Benachrichtigungs- und Evaluationsphase über Callback-Mechanismen auf Field-Änderungen in Sensor- und Effektor-Knoten. Um auch komplexere Software in Scheme entwickeln zu können, wird in dieser Arbeit eine Scheme-Erweiterung namens *OOPS* benutzt, mit der objektorientierte Programmierung in Scheme möglich wird.

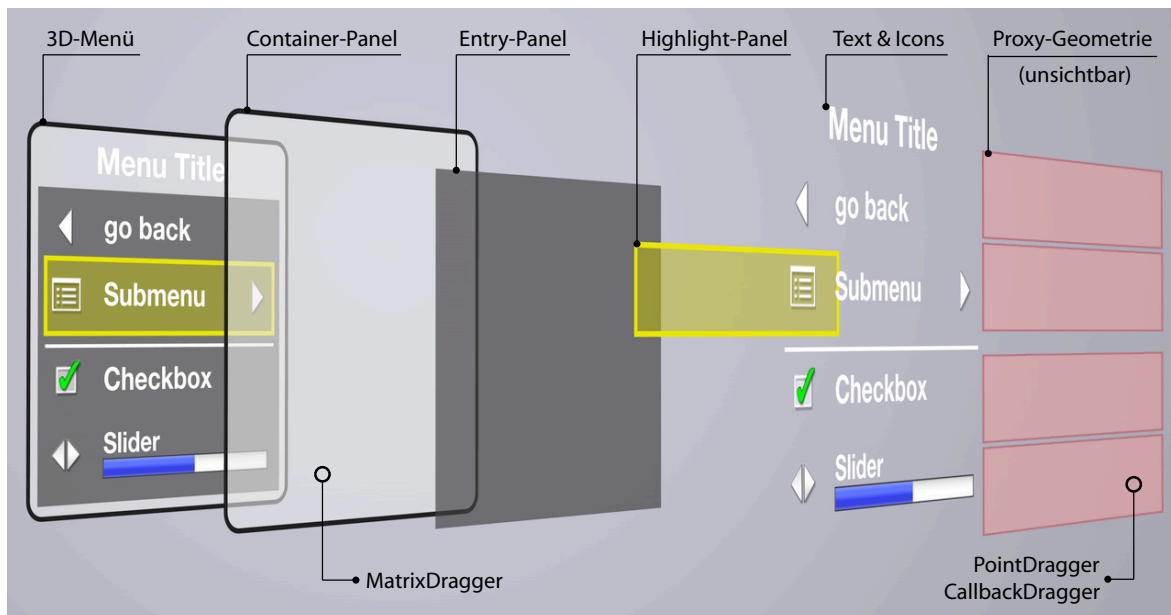
### 7.1.3 Tools und Dragger

Für die Interaktion mit 3D-Objekten über ein Eingabegerät wird in Avango das sog. Tool-Dragger-Konzept verwendet. Die für das Verständnis der Interaktion mit dem 3D-Menü

wichtigsten Eigenschaften von Tools und Draggern werden im Folgenden erläutert. Das Tool-Dragger-Konzept wird ausführlich in [Tramberend 2003] beschrieben.

- *Tool*: Tools sind die softwareseitigen Werkzeuge, mit denen Interaktionen ausgeführt werden. Mehrere Tools können einem `ToolHolder` zugeordnet und von einem `ToolManager` verwaltet werden. Der Tool-Holder bekommt über Field-Connections die aktuelle Position sowie den Zustand der Taster des Eingabegerätes mitgeteilt. Die Tools bilden die 6-DOF-Eingabewerte auf ein Interaktionsmodell ab. Das `DragTool` dient der direkten Translation oder Rotation von Objekten. Das `ScaledGrabTool` setzt die Bewegungen gemäß der Scaled-Grab-Interaktion um und mit dem `PointTool` können Objekte durch Zeigen ausgewählt werden. Jedes Tool kann mit einem eigenen Namen versehen werden.
- *Dragger*: Diese sind einem Objekt im Szenengraphen zugeordnet und bestimmen mit welchen Objekten interagiert werden kann. Für jeden Tool-Typen gibt es verschiedene Dragger. So dient ein `MatrixDragger` der 6-DOF-Interaktion, ein `ScaledGrabDragger` der Scaled-Grab-Interaktion, ein `CallbackDragger` führt bei einem Klick einen Callback aus und der `PointDragger` führt entsprechende Callbacks für das Eintreten und Verlassen des Strahls aus. Über eine `ToolDraggerMatchMap` wird festgelegt, welche Tools und Dragger miteinander verbunden werden können. Der `ToolDraggerConnector` schließlich stellt über Field-Connections die Verbindung zwischen Tools und Draggern her. Soll ein Dragger nur auf bestimmte Tools reagieren, kann dem Dragger eine Liste von Tool-Namen zugewiesen werden.
- *Ray-Pick-Service*: Dieser Service wird von den Tools für die Selektion eines Objektes benutzt. Ausgehend von der Position und Ausrichtung des Tools wird ein Schnitt eines Strahls (*Intersection-Ray*) mit den Objekten der 3D-Szene berechnet. Wird ein Objekt geschnitten, wird im Szenengraph der zuständige Dragger ermittelt.

Für die Interaktion mit dem 3D-Menü repräsentiert das Eingabegerät des Benutzers mehrere Tools: zwei `DragTools`, die jeweils mit dem ersten oder zweiten Taster des Eingabegerätes aktiviert werden, sowie ein `ScaledGrabTool` und ein `PointTool`. Das `ScaledGrabTool` ist ebenfalls mit der ersten Taste verbunden und ermöglicht die Scaled-Grab-Interaktion mit den Objekten in der Szene sowie mit dem Menü-Panel, denen jeweils `ScaledGrabDragger` zugeordnet sind. Das `DragTool` mit der zweiten Taste bekommt den Namen „ContextTool“. Es dient der Aktivierung des Kontextmenüs, wenn ein Objekt angeklickt wird, das mit einem `CallbackDragger` ausgestattet ist, welches auf ein Tool mit dem Namen „ContextTool“ reagiert. Das `DragTool` mit der ersten Taste dient dem Aktivieren von Menüeinträgen, die jeweils einen `CallbackDragger` besitzen. Jeder Menüeintrag enthält außerdem jeweils einen `PointDragger`, so dass mit Hilfe des `PointTools` ein Highlighting der Menüeinträge mit dem



**Abbildung 7.1:** Schematischer Aufbau des 3D-Menüs — Das 3D-Menü besteht aus mehreren koplanar angeordneten Objekt-Schichten. Die Grafik zeigt die Objekte des Menüs von hinten (links) nach vorne (rechts). Das hinterste Panel sowie die unsichtbaren vordersten Geometrien sind mit Draggern ausgestattet und ermöglichen die Interaktion mit dem Menü und den Menüeinträgen.

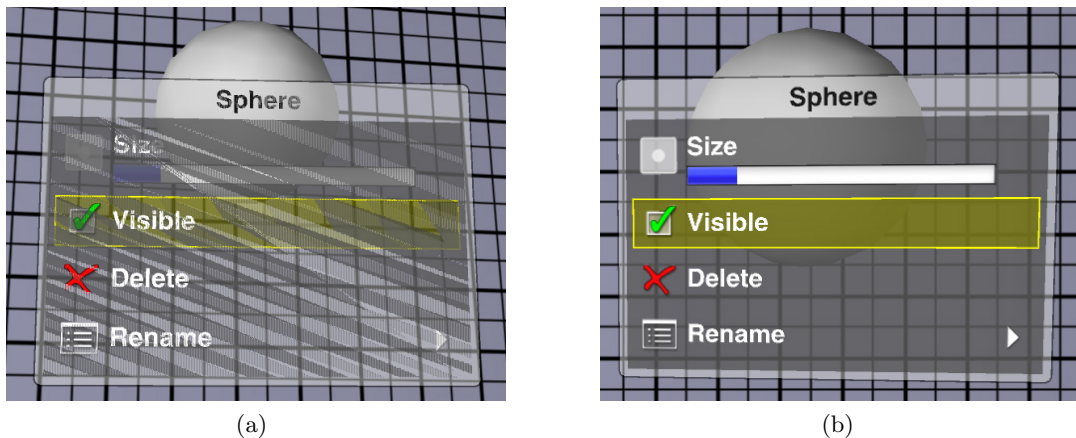
Selektionsstrahl möglich ist. Abbildung 7.1 zeigt die Zuordnung der Dragger zu den im Menü enthaltenen Objekten.

## 7.2 Aufbau des Menüs

Die graphischen Elemente des Menüs werden koplanar in einer Ebene dargestellt. Abbildung 7.1 zeigt die einzelnen Elemente und den Aufbau des Menü-Fensters. Die Panels sind in C++ geschriebene Avango-Nodes, die eine rechteckige Geometrie, wahlweise mit abgerundeten Ecken und optionalem Rand als Polygone erzeugen. Die Icons bestehen aus einer quadratischen Geometrie mit RGBA-Textur. Die Schrift wird über den von Avango bereitgestellten **Text-Node** als Polygon-Geometrie erzeugt. Farben und Transparenzen werden über Materialien definiert. Die Farben werden ausschließlich dem **Emission-Kanal** der Materialien zugewiesen, damit die Darstellung der Menüobjekte nicht von der Beleuchtung der Szene abhängt. Die **Ambient-**, **Diffuse-** und **Specular-Kanäle** sind dementsprechend abgeschaltet.

Koplanare Polygone werden durch Rundungsfehler im *Z-Puffer* oft fehlerhaft mit sog. *Z-Fighting* dargestellt. Dieser Fehler sorgt dafür, dass die einzelnen Pixel pseudo-zufällig einem der in der Ebene liegenden Objekte zugeordnet werden und dadurch störende Muster entstehen, die darüber hinaus bei Animationen stark flackern [Möller u. Haines 1999]. Ein



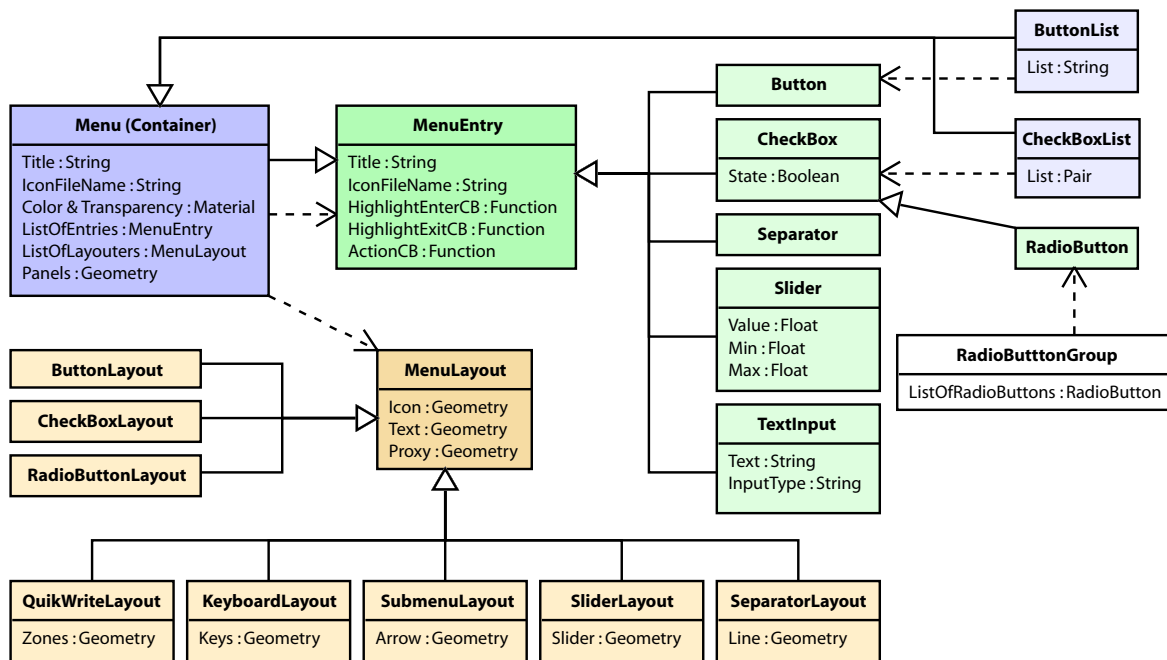


**Abbildung 7.2:** *Z-Fighting und Sortierung von transparenten Objekten* – (a) Ohne `pfLayer` und `DrawBin` werden die koplanaren Objekte in undefinierter Reihenfolge gezeichnet. Dies führt zu falsch sortierten transparenten Objekten und Z-Fighting. (b) Menüdarstellung mit korrekt sortierten Objekten.

`Layer-Node` eliminiert diesen Fehler, indem den Objekten eine visuelle Priorität zugeordnet wird. Diese Priorität hat jedoch keinen Einfluss auf den Transparenzkanal. Die Sortierung der transparenten Objekte wird anhand der Mittelpunkte der Geometrien ermittelt. Je nach Perspektive auf das Menü werden die Objekte falsch sortiert, so dass eine manuelle Sortierung notwendig ist. Dazu werden das hintere Panel, das innere Panel, das Highlight und das Icon mit jeweils aufsteigender Nummer einem `DrawBin` zugeordnet und dadurch in der gewünschten Reihenfolge gezeichnet [Eckel u. Jones 2000]. Abbildung 8.1 zeigt die Darstellung ohne `pfLayer` und `DrawBin` im Vergleich zur korrekt sortierten Darstellung.

Die vorderste Schicht des Menüs bilden rechteckige Proxy-Geometrien, die jeweils die aktive Fläche eines Widgets bedecken. Diese Geometrien sind unsichtbar, werden aber bei der Strahlinteraktion berücksichtigt und aktivieren bei Strahlberührung das Highlight für den jeweiligen Eintrag bzw. aktivieren bei Klick den Callback des Widgets und führen dadurch die Funktionalität des Menüeintrags aus. Zu diesem Zweck sind der Proxy-Geometrie die für das jeweilige Widget zuständigen `CallbackDragger` und `PointDragger` zugeordnet.

Dem Menü-Panel ist ein `MatrixDragger` zugeordnet und kann überall dort per Strahlinteraktion gegriffen werden, wo es nicht von einer Proxy-Geometrie bedeckt ist. So lässt sich das Menü an der Titelzeile oder am Rand greifen und verschieben.



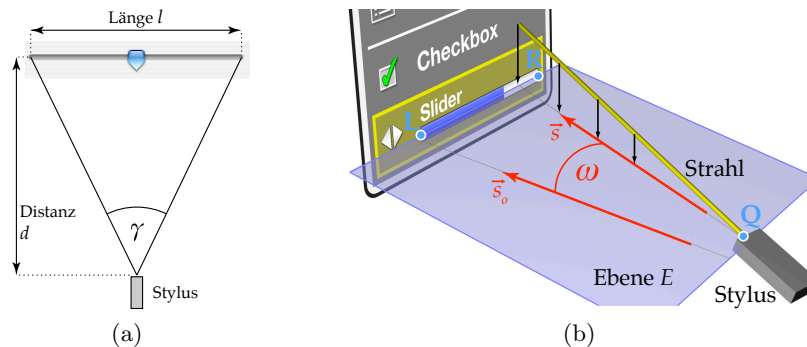
**Abbildung 7.3:** Klassendiagramm der 3D-Menü-Software — Jedes Menüwidget ist von der MenuEntry-Klasse abgeleitet (grün) und wird von einem Layouter dargestellt (orange). Menü-Container können Menüs, Submenüs, Button- und Checkbox-Listen darstellen (blau). Zusammengehörige Radiobuttons werden einer RadioButtonGroup zugewiesen (weiß).

### 7.3 Klassenstruktur der Menü-Software

Die Funktionalität des Menüs wird über eine Reihe von OOPS-Klassen bereitgestellt. Die wichtigsten Klassen werden hier kurz vorgestellt und in Abbildung 7.3 schematisch dargestellt.

Die Basis bildet die MenuEntry-Klasse, welche die Grundfunktionalität eines Menüeintrags bereitstellt. Dazu gehören neben einem Titel auch die Angabe einer Grafikdatei für das Icon sowie die Möglichkeit Callback-Funktionen für bestimmte Ereignisse zu definieren. Zu diesen Ereignissen zählen das Highlighting und das Anklicken des Eintrags. Alle weiteren Widgets sind von diesem Typ abgeleitet und stellen für die jeweilige logische Funktionalität zusätzliche Attribute und Methoden bereit.

Die Menu-Klasse definiert den Menü-Container und ist ebenfalls von MenuEntry abgeleitet. Als Container definiert diese Klasse zusätzlich die graphische Repräsentation des Menü-Panels und verwaltet eine Liste von Einträgen, die im Container dargestellt werden sollen. Des Weiteren enthält die Klasse Attribute und Methoden, mit denen das Aussehen des Menüs gestaltet werden kann, z.B. Schriftgröße, Materialien (Farben und Transparenzen), Abstände und Ränder. Jeder Menü-Container kann andere Container als Menüeinträge zum Aufbau von Hierarchien enthalten.



**Abbildung 7.4:** *Slider-Interaktion* – (a) Der Slider-Interaktionswinkel ist bei Strahlinteraktion abhängig von der Distanz und der Länge des Reglers. (b) Der Vektor  $\vec{s}_0$  ist die Projektion des Interaktionsstrahls aus dem vorangegangenen Frame,  $\vec{s}$  die aus dem aktuellen Frame. Die Winkeländerung  $\omega$  bestimmt die Wertänderung des Sliders.

Soll ein Menü aus einer Liste von Schaltflächen oder Checkboxes bestehen, können diese über spezielle Listen-Container realisiert werden. `ButtonList` und `CheckboxList` sind jeweils vom `Menu-Container` abgeleitet. Anstelle von individuellen `MenuItem`-Objekten werden diese Menüs durch die Übergabe von Listen aufgebaut, die lediglich die Namen der Buttons bzw. zusätzlich den Aktivierungszustand der Checkboxes enthalten. Den Callbacks wird jeweils der Listen-Index sowie Name und Aktivierungszustand übergeben.

Radiobuttons müssen für die korrekte Funktionsweise gruppiert werden. Zu diesem Zweck gibt es die `RadioButtonGroup`-Klasse. Dieser können mehrere Objekte des Typs `RadioButton` zugewiesen werden. Alle in der Gruppe enthaltenen Einträge werden daraufhin untereinander benachrichtigt, wenn einer der Radiobuttons seinen Zustand ändert.

Die Darstellung der Menü-Einträge innerhalb des Containers übernehmen zusätzliche Klassen, die von der Basisklasse `MenuLayout` abgeleitet sind. Für jedes Widget gibt es einen speziellen Layouter, der die visuelle Repräsentation des Eintrags generiert, also z.B. Objekte für Text, Icon und Proxy-Geometrie erstellt, positioniert, skaliert und in den Szenengraphen einhängt. Jeder Layouter meldet seinen Platzbedarf an den Container, so dass dieser jedem Layouter seine Position innerhalb des Menüpanels sowie die Breite des Panels mitteilen kann.

## 7.4 Entfernungsunabhängige Slider-Interaktion

Bei der Strahlinteraktion mit einem für WIMP-Benutzeroberflächen typischen Schieberegler ist der Interaktionswinkel abhängig von der Länge des Reglers sowie von der Entfernung des Reglers zum Interaktionsgerät (Abbildung 7.4a). Je weiter der Regler entfernt ist (Distanz  $d$ ) und je kürzer dieser dargestellt wird (Länge  $l$ ), desto kleiner wird der Winkel  $\gamma$ , innerhalb

dessen der Benutzer den Reglerwert einstellen kann:

$$\gamma = 2 \cdot \arctan \frac{l/2}{d} \quad (7.1)$$

Je kleiner der Wert für  $\gamma$  ist, desto exakter muss der Benutzer interagieren, um einen bestimmten Wert auf dem Schieberegler einzustellen.

Für eine von der Darstellung des Reglers entkoppelte Interaktion, darf diese weder von  $l$  noch von  $d$  abhängen. Der Winkelbereich  $\gamma$  wird auf einen Wert festgelegt, der eine möglichst angenehme Interaktion mittels Handgelenkrotation erlaubt. In den 3D-Menü-Testanwendungen (u.a. dem VRGeo-Demonstrator und der Benutzerstudie) ist dieser Winkel mit  $\gamma = 30^\circ$  definiert<sup>4</sup>. Dieser Wert wurde durch einen informellen Test mit drei Personen ermittelt. Deutlich größere Werte wurden als unangenehm empfunden (unnatürliche Dehnung des Handgelenks), deutlich kleinere Werte wurden als zu empfindlich für die Einstellung des Reglerwertes empfunden.

Die Reglerwertveränderung basiert auf der Winkeländerung des Interaktionsgerätes pro Frame. Der Interaktionsstrahl  $\vec{S}$  wird in jedem Frame als Vektor  $\vec{s}$  auf die Ebene  $E$  projiziert, die aus drei Punkten aufgespannt wird: der Position des Interaktionsgerätes  $Q$  sowie dem linken und rechten Ende des Schiebereglers  $L$  und  $R$  (siehe Abbildung 7.4b). Der Vektor wird als  $\vec{S}_0$  gespeichert und im nächsten Frame als Projektion  $\vec{s}_0$  mit  $\vec{s}$  verglichen. Die Winkeldifferenz  $\omega$  wird anschließend unter Berücksichtigung von  $\gamma$  auf den Wertebereich des Reglers ( $V_{Max} - V_{Min}$ ) umgerechnet und ergibt die Wertdifferenz, die zum Reglerwert  $V$  des letzten Frames addiert wird:

$$V_{New} = V + \frac{\omega}{\gamma} (V_{Max} - V_{Min}) \quad \text{mit} \quad \omega = \arccos \frac{\vec{s} \cdot \vec{s}_0}{|\vec{s}| \cdot |\vec{s}_0|} \quad (7.2)$$

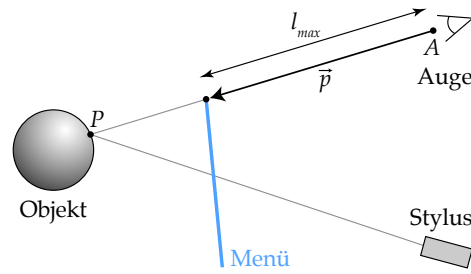
$V_{New}$  wird anschließend unter Beachtung des gültigen Wertebereichs als aktueller Reglerwert  $V$  dem Regler zugewiesen. Eine bestimmte Winkeländerung resultiert immer in einer gleichen Veränderung des Wertebereichanteils und ist damit unabhängig von Größe und Entfernung des Reglers. Der Interaktionsstrahl wird für die Dauer der Slider-Interaktion abgeschaltet, da es sich bei dieser Interaktionsform nicht um eine Strahlinteraktion handelt.

### 7.5 Regelbasierte Positionierung

Im Abschnitt 6.3.2 wurden die Grundlagen für die regelbasierte Positionierung erläutert. Der erste Referenzpunkt für die Menüplatzierung ist der Schnittpunkt  $P$  von Strahl und Objekt,

---

<sup>4</sup>Dieser Winkel entspricht etwa der Interaktion mit einem 1m langen Regler in 90cm Entfernung.



**Abbildung 7.5:** Menüpositionierung – Positionsberechnung vor dem angeklickten Objekt unter Berücksichtigung der maximalen Menüentfernung  $l_{max}$ .

an dem der Benutzer dieses angeklickt hat. Der Vektor vom Augpunkt  $A$  zum Punkt  $P$  wird auf eine maximale Länge  $l_{max}$  beschränkt, damit das Menü bei weit entfernten Objekten nicht ebenso weit entfernt dargestellt wird, sondern eine maximale Entfernung nicht überschreitet. Der resultierende Vektor bildet den Positionierungsvektor  $\vec{p}$ :

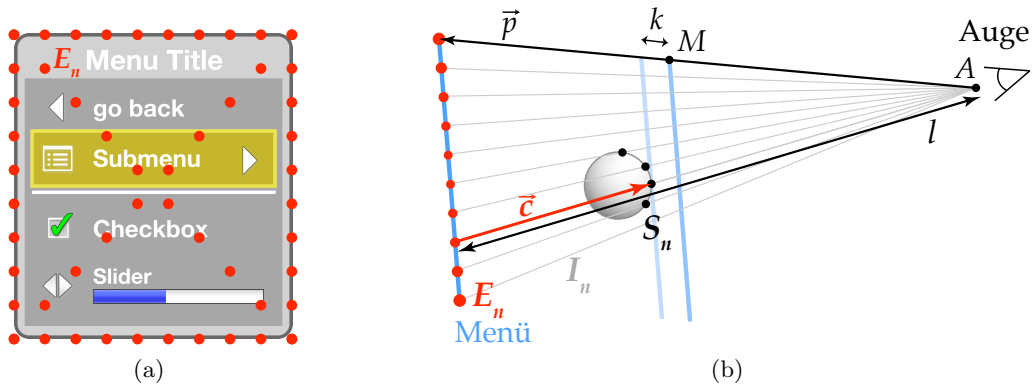
$$\vec{p} = \frac{\vec{AP}}{|\vec{AP}|} \cdot \min(l_{max}, |\vec{AP}|) \quad (7.3)$$

Der Punkt  $A + \vec{p}$  ist die „optimale Menüposition“, wenn Verdeckungsprobleme nicht berücksichtigt werden (siehe Abbildung 7.5). Wenn das Menü mit der Oberkante an dieser Position dargestellt wird, schwebt das Menü aus Sicht des Benutzers direkt unterhalb des angeklickten Punktes vor dem Objekt.

### 7.5.1 Strahlentest

Ob andere Objekte das Menü an dieser Position verdecken, wird über einen Strahlentest ermittelt. Dazu werden vom Augpunkt ausgehend Schnittstrahlen erzeugt, deren Endpunkte eine möglichst engmaschige Abtastung der Menü-Oberfläche ergeben. Je nach Größe der in der Szene vorhandenen Objekte muss die Distanz zwischen den abgetasteten Punkten variiert werden, wobei Kleinstobjekte nicht zwingend erkannt werden müssen, da sie die Sicht auf das Menü nicht behindern. Im VRGeo-Demonstrator hat sich eine relativ engmaschige Abtastung des Menürandes sowie der Diagonalen als praktikabel bewährt (Abbildung 7.6a).

Jeder Intersection-Ray  $I_n$  hat seinen Ursprung im Augpunkt  $A$ , einen Endpunkt  $E_n$  auf der Menüoberfläche und eine Länge  $l_n = |\vec{AE}_n|$ . Findet ein Schnitt mit einem Objekt statt, kann der Schnittpunkt  $S_n$  berechnet werden. Für jeden Schnitt wird ein Vektor  $\vec{c}_n = \vec{E_n S_n}$  berechnet. Der längste aller so berechneten Vektoren definiert den Korrekturvektor  $\vec{c}$ . Die Länge des für diese Berechnung herangezogenen Strahls ist  $l$  (siehe Abbildung 7.6b). Findet kein Schnitt statt, werden  $\vec{c}$  als Nullvektor und  $l = 1$  definiert.



**Abbildung 7.6:** Strahlentest – (a) Endpunkte  $E_n$  der Intersection-Rays  $I_n$  auf der Menüoberfläche. (b) Verschiebung des Menüs vor das störende Objekt mittels Intersection-Rays.

Der bisherige Positionierungsvektor  $\vec{p}$  wird nun nach dem Strahlensatz verkürzt, d.h. das Menü wird so weit zum Benutzer hin verschoben, dass es nicht mehr verdeckt wird. Anschließend wird die Position nochmals um eine kleine Strecke  $k$  (wenige Zentimeter) zum Betrachter hin verschoben, damit das Menü knapp vor dem Objekt dargestellt wird und nicht unmittelbar an diesem „klebt“. Die neue „optimale Menüposition“  $M$  lautet unter Berücksichtigung der Verdeckung daher:

$$M = A + \frac{l - |\vec{c}|}{l} \cdot \vec{p} - k \cdot \frac{\vec{p}}{|\vec{p}|} \quad (7.4)$$

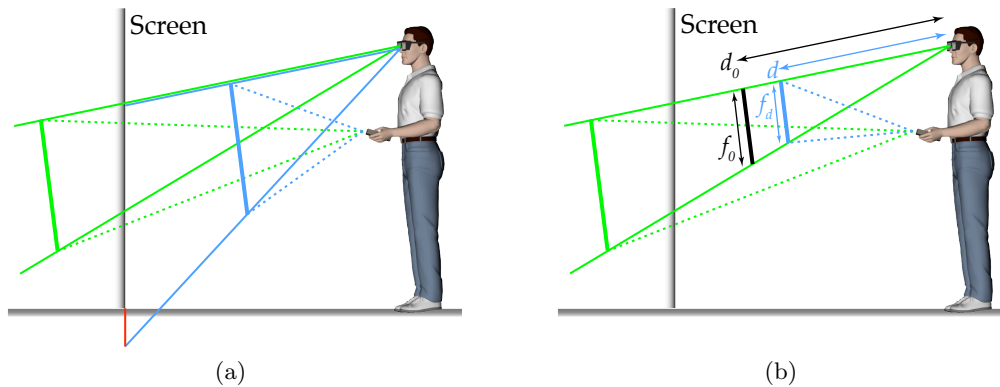
## 7.5.2 Skalierung

Wie im Abschnitt 6.3.2 erwähnt, verfolgt die Skalierung des Menüs zwei Ziele: Zum einen sorgt die Anpassung der Menügröße an die Entfernung des Menüs zum Benutzer dafür, den Blick- und Interaktionswinkel weitgehend konstant zu halten. Andererseits wird sichergestellt, dass die Schrift im Menü jederzeit lesbar ist.

## 7.5.3 Entfernungsabhängige Größenanpassung

Die variable Darstellungsdistanz des Menüs durch die kontextabhängige Positionierung führt zu drei Einschränkungen für die Darstellung und für die Interaktion mit dem Menü:

- *Verdeckung:* Wird ein Menü näher am Betrachter dargestellt, wird der Betrachtungswinkel größer und das Menü verdeckt größere Teile der Szene.
- *Sichtbarkeit:* Durch den vergrößerten Betrachtungswinkel werden Teile des Menüs am Bildrand abgeschnitten und können nicht angezeigt werden.



**Abbildung 7.7:** Entfernungabhängige Skalierung – (a) Ohne Skalierung vergrößern sich Betrachtungs- und Interaktionswinkel; Teile des Menüs werden am Bildrand abgeschnitten (rote Linie). (b) Die entfernungsabhängige Skalierung hält den Platzbedarf der Projektion sowie Betrachtungs- und Interaktionswinkel weitgehend konstant.

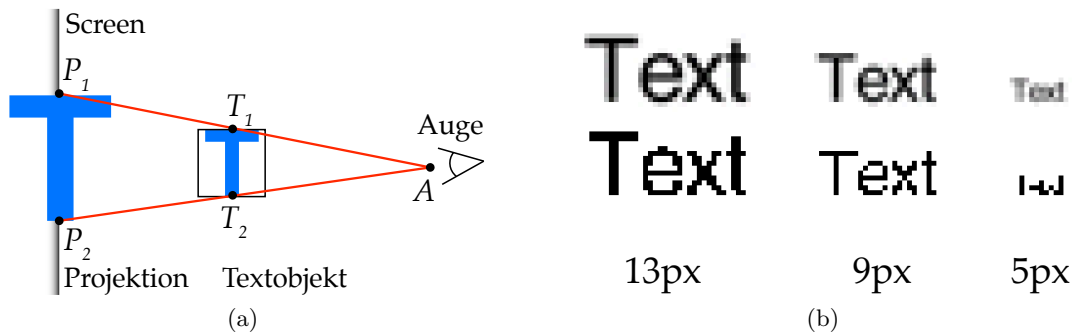
- *Interaktion:* Der ebenfalls vergrößerte Interaktionswinkel erschwert die Bedienung, da zum Erreichen der Menüelemente größere Bewegungen notwendig sind.

Abbildung 7.7a zeigt zwei gleich große Menüs, die weit entfernt (grüne Linien) und nah am Benutzer (blaue Linien) dargestellt sind. Die Projektion des nah dargestellten Menüs auf die Bildebene ragt über den Rand des Displays hinaus (rote Linie) und ist nicht vollständig sichtbar. Dargestellt ist auch die Zunahme des Interaktionswinkels (gestrichelte Linien) bei abnehmender Entfernung.

Wird nun das Menü in Abhängigkeit der Entfernung skaliert, so dass der Betrachtungswinkel identisch bleibt, verringern sich die genannten Probleme. Das Menü bleibt vollständig sichtbar, da die Projektion auf die Bildebene unabhängig von der Entfernung gleich groß ist. Der Interaktionswinkel bleibt mit zunehmender Nähe weitgehend konstant (siehe Abbildung 7.7b). Erst wenn das Menü sehr nah an den Kopf des Betrachters heranrückt, verkleinert sich der Interaktionswinkel deutlich, wenn das Interaktionsgerät in der ursprünglichen Position verharrt. Daher ist es sinnvoll, für die Positionierung des Menüs eine Mindestentfernung oder eine Mindestgröße  $f_{min}$  festzulegen.

Die Skalierung wird auf eine bestimmte Entfernung  $d_0$ , gemessen vom Augpunkt zur Menüposition, und einen Skalierungsfaktor  $f_0$  genormt. Die Werte werden so gewählt, dass das Menü in der gewählten Distanz und Größe angenehm zu bedienen ist. Wird nun das Menü durch die dynamische Positionierung in einer anderen Entfernung  $d$  dargestellt, kann über den Strahlensatz der angepasste Faktor  $f_d$  berechnet werden:

$$f_d = \max \left( f_{min}, d \cdot \frac{f_0}{d_0} \right) \quad (7.5)$$



**Abbildung 7.8:** *Lesbarkeit von Text* – (a) Berechnung der Darstellungsgröße von Text über Projektion auf den Screen. (b) 3D-Rendering von Text mit einer Buchstabenhöhe von 13, 9 und 5 Pixeln, mit Antialiasing (oben) und ohne Kantenglättung (unten).

### 7.5.4 Lesbarkeit von Schrift

Die Skalierung des Menüs hat die Lesbarkeit der im Menü enthaltenen Schrift bisher nicht beachtet. Die Lesbarkeit der Menübeschriftung ist jedoch von elementarer Bedeutung für eine zuverlässige Menüinteraktion. Ob die Schriftzeichen lesbar sind, hängt von verschiedenen Faktoren ab: Schriftart, Anzahl der verfügbaren Pixel pro Zeichen und Kantenglättung (Antialiasing oder Texturfilterung). Die Anzahl der Pixel bezieht sich auf die Darstellung der Schrift in der Bildebene, also der Projektion der Textobjekte auf den Screen. Die Größe der Projektion hängt ab von der Größe des Textobjektes, der Distanz vom Betrachter zum Textobjekt und der Distanz vom Betrachter zum Screen. Abbildung 7.8a veranschaulicht die Zusammenhänge. Da die Textobjekte durch die Rotation des Menüs nicht zwangsläufig parallel zum Screen ausgerichtet sind, werden die zwei relevanten Punkte  $T_1$  und  $T_2$  getrennt auf den Screen projiziert. Die Projektionen  $P_n$  werden berechnet als Schnitt des Strahls  $A + \overrightarrow{AT_n}$  mit der Screenenebene. Die Größe der Textprojektion in der Bildebene entspricht  $|\overrightarrow{P_1P_2}|$ . Da die Menüs in der vorliegenden Implementierung stets aufrecht ausgerichtet sind, also keine Rotation um die Roll-Achse aufweisen, ist für die Berechnung der zur Verfügung stehenden Pixel die vertikale Screenauflösung relevant. Die absolute Y-Auflösung des Displays dividiert durch die Screenhöhe in Meter ergibt die relative vertikale Screenauflösung in Pixel pro Meter, so dass die Anzahl der für die Schriftdarstellung verfügbaren Pixel nach Formel 7.6 berechnet werden kann.

Für die horizontale Abmessung der Projektion kann äquivalent verfahren werden. Wichtig ist, dass nicht die tatsächliche, individuelle Größe eines Buchstabens für die Berechnung herangezogen wird, sondern ein Einheitsmaß für die Schrift. Dafür kann ein Quadrat verwendet werden, dessen Kantenlängen der Höhe eines Großbuchstabens entspricht, wie in Abbildung 7.8a dargestellt.



$$\text{Pixelanzahl} = |\overrightarrow{P_1P_2}| \cdot \frac{\text{absolute Y-Auflösung}}{\text{Screenhöhe}} \quad (7.6)$$

Eine einfache Beispielrechnung verdeutlicht die Formel: Der Benutzer steht 2m vor dem TwoView-Display. Vor ihm schwebt in 1,2m Entfernung in Augenhöhe, parallel zum Screen, ein 2cm großer Text. Die Projektion auf dem Screen ist ca. 3,3cm groß. Die Screenauflösung beträgt 1024 Pixel auf 2,5m Höhe. Der Text kann also mit 13 Pixel hohen Buchstaben dargestellt werden. Steht der Benutzer bei ansonsten identischer Szene nur 80cm vom Screen entfernt, sind die Buchstaben nur noch 1,3cm bzw. 5 Pixel hoch. Wie Abbildung 7.8b zeigt, ist die Schrift selbst mit Antialiasing praktisch nicht mehr lesbar. Geometrische Verzerrungen durch rotierten Text schränken die Lesbarkeit noch weiter ein. Die Projektion der Schrift in diesem Beispiel sollte für eine durchweg gute Lesbarkeit also mindestens etwa 10 Pixel groß sein. Dies entspricht einer Darstellungsgröße von ca. 2,4cm in der Bildebene.

Nachdem die Menüpositionierung und entfernungsabhängige Skalierung erfolgt ist, wird für die vier Ecken der beschrifteten Fläche des Menüs sowohl eine horizontale wie auch eine vertikale Projektionsgrößenbestimmung durchgeführt. Der Kleinste dieser insgesamt acht Werte  $s_{H1..4}$  und  $s_{V1..4}$  wird für die Berechnung des Textskalierungsfaktors  $f_t$  herangezogen, wobei  $s_{min}$  der Mindestdarstellungsgröße entspricht. Ist die Schrift größer als die Mindestgröße, soll das Menü in der Größe nicht verändert werden. Daher wird der Skalierungsfaktor auf ein Minimum von 1 begrenzt:

$$f_t = \max\left(1, \frac{s_{min}}{\min(s_{H1}, s_{H2}, s_{H3}, s_{H4}, s_{V1}, s_{V2}, s_{V3}, s_{V4})}\right) \quad (7.7)$$

Der endgültige Skalierungsfaktor  $f$  des Menüs errechnet sich aus dem entfernungsabhängigen Skalierungsfaktor  $f_d$  und dem Textgrößenkorrekturfaktor  $f_t$ :

$$f = f_t \cdot f_d \quad (7.8)$$

### 7.5.5 Positionsbegrenzung an den Screengrenzen

Nah am Benutzer platzierte Objekte geraten leicht in Bereiche, in denen sie an den Displayrändern abgeschnitten sind oder ganz außerhalb des sichtbaren Bereichs, also außerhalb des View-Frustums, liegen. In Abbildung 7.7a auf Seite 71 wird das Menü aufgrund des Betrachtungswinkels am unteren Bildrand abgeschnitten. Wenn Teile des Menüs außerhalb des View-Frustums liegen, ist die Lesbarkeit und Bedienbarkeit stark eingeschränkt. Der Benutzer muss seine Kopfposition verändern oder das Menü in den sichtbaren Bereich schieben, um damit interagieren zu können. Bei der Positionierung des Menüs wird das Menüpanel in einem solchen Fall daher automatisch in den sichtbaren Bereich verschoben.

Ähnlich der Textprojektion im letzten Abschnitt, werden die vier Menüeckpunkte  $P_{1..4}$  in die Bildebene projiziert und auf ihre Lage innerhalb des sichtbaren Bereichs überprüft. Beim TwoView-Display genügt es, die Koordinaten der Projektionspunkte  $P'_{1..4}$  auf einen gültigen Wertebereich innerhalb eines Rechtecks zu überprüfen. Liegen ein oder mehrere Punkte außerhalb des Displays, wird deren Entfernung zur Displaykante berechnet. Die jeweils horizontal und vertikal am weitesten außerhalb liegenden Punkte heißen  $P_h$  und  $P_v$ , ihre Projektionen entsprechend  $P'_h$  und  $P'_v$ , die Abstände zur Bildkante sind mit  $h$  und  $v$  benannt. Über den Strahlensatz können aus der horizontalen und vertikalen Abweichung die Komponenten für den Korrekturvektor  $\vec{c}$  errechnet werden, um den die Menüposition  $M$  verschoben werden muss. Für die Formel wird davon ausgegangen, dass der Screen in der  $XY$ -Ebene des Weltkoordinatensystems liegt, wobei  $h$  und  $v$  parallel zur  $X$ - bzw.  $Y$ -Achse liegen.  $A$  ist der Augpunkt des Benutzers.

$$M_{neu} = M + \vec{c} \quad \text{mit} \quad \vec{c} = \begin{pmatrix} h \cdot \frac{|\overrightarrow{AP_h}|}{|\overrightarrow{AP'_h}|} \\ v \cdot \frac{|\overrightarrow{AP_v}|}{|\overrightarrow{AP'_v}|} \\ 0 \end{pmatrix} \quad (7.9)$$

Anschließend liegen die betroffenen Eckpunkte des Menüs im sichtbaren Bereich.

### 7.5.6 Ablauf der Menüpositionierung

Nachdem der Benutzer auf einem Objekt das Kontextmenü angefordert hat, wird zunächst die Standard-Menüposition vor dem Objekt berechnet, das Menü entfernungsabhängig skaliert und die Lesbarkeit des Textes sichergestellt. Anschließend wird der Strahlentest durchgeführt und das Menü, falls nötig, vor die störenden Objekte verschoben und wieder entfernungsabhängig skaliert. Erst nach diesen Schritten stehen Position und Größe des Menüs fest, so dass der Schnitt mit den Screengrenzen berechnet werden kann. Falls das Menü an den Bildgrenzen abgeschnitten ist, wird es in den sichtbaren Bereich verschoben. Da die horizontale und vertikale Verschiebung das Menü evtl. in oder hinter andere störende Objekte verschiebt, werden die Schritte ab dem Strahlentest noch einmal wiederholt. Die beschriebenen Verfahren können nicht immer eine „optimale“ Position für das Menü finden. Wenn störende Objekte sehr nah am Benutzer die Sicht auf das Menü verdecken oder die Bildschirmgröße für die vollständige Darstellung des Menüs nicht ausreicht, wird nach zweimaligem Durchlauf der Positionsberechnung der Vorgang beendet, da angenommen wird, dass mit diesen Verfahren keine bessere Positionierung erzielt werden kann. Alle genannten Schritte werden innerhalb eines Frames ausgeführt, so dass der Benutzer erst das fertig positionierte Menü angezeigt bekommt.

## **8 Evaluation: Performancevergleich der Slider-Interaktion in einem VR-Szenario zwischen 3D-Menüs, PDA und Tablet-PC**

Der Demonstrator im VRGeo-Projekt bietet einen großen Funktionsumfang, der die Systemsteuerung über komplexe, dynamische und hierarchisch gegliederte Menüs notwendig macht. In der Vergangenheit wurden die Menüs auf externen Geräten, zuerst auf PDAs, später auf einem Tablet-PC, angezeigt und bedient. Die Erfahrung im Umgang mit diesen Menüs machte die Neuentwicklung von integrierten 3D-Menüs erforderlich. Im Rahmen der vorliegenden Diplomarbeit wurden die Erfahrungen und Anforderungen, die Konzeption und schließlich die Implementierung der neuen 3D-Menüs beschrieben. Die Ziele waren die Verbesserung der Nutzungserfahrung im Vergleich zu den PDA- und Tablet-Menüs, die Integration der Menüs in die VR-Umgebung sowie die Optimierung der Menüinteraktion.

Einige der Gestaltungsaspekte zielen auf die Verbesserung der Benutzungsgeschwindigkeit ab. Die eindimensionale Anordnung der Menüeinträge sowie die Gestaltung jedes Menüeintrags als großer Block sollen das Anwählen eines Eintrags vereinfachen. Der entfernungs- und größenunabhängige Slider entkoppelt die Darstellung des Reglers von der Interaktion und soll dadurch bei ungünstiger Positionierung des Reglers das Erreichen des Zielwertes beschleunigen. Durch die Integration der Menüs in VR entfällt der Blick- und Fokuswechsel zwischen VR-Display und PDA/Tablet-Display. Auch dies soll zur Steigerung der Effizienz beitragen. Die genannten Vereinfachungen der Interaktion sollten auch die Effektivität verbessern: Wenn neben der steigenden Performance zusätzlich die Fehlerrate sinkt, wirkt sich dies in einer höheren Zufriedenstellung der Benutzer und damit höheren Akzeptanz der neuen Menüs aus.

Die genannten Aspekte sollen in einer Benutzerstudie anhand eines einfachen VR-Testszenarios überprüft werden. Dieses Szenario stellt einen typischen Interaktionsablauf mit einem Schieberegler im Kontextmenü dar: Der Benutzer selektiert ein Objekt in VR und ruft dessen Kontextmenü auf. Das Kontextmenü zeigt einen Schieberegler an, mit dem der Benutzer einen Parameter verändern kann. Der Benutzer greift den Regler und kontrolliert die Veränderung des Wertes visuell in der 3D-Szene. Nach Erreichen des Zielwertes läßt der

Benutzer den Regler los. Dieser Ablauf wird mit insgesamt sechs verschiedenen Menüvarianten getestet:

- *PDA*: Der PDA dient als 3D-Interaktionsgerät und Menü. Die Bedienung des Menüs erfolgt mit dem mitgelieferten Plastik-Stift.
- *Tablet-PC*: Der Tablet-PC dient als Menü. Die 3D-Interaktion und die Menübedienung erfolgt über einen 6-DOF-Stylus mit eingebauter Stiftspitze.
- *Block-3D-Menü*: Das neu entwickelte 3D-Menü in VR mit Block-Elementen und entfernungsunabhängiger Slider-Interaktion mittels Handrotation. Das Menü wird in zwei Varianten präsentiert: einmal nah am Benutzer und einmal deutlich weiter entfernt.
- *WIMP-3D-Menü*: Eine Variante des 3D-Menüs mit einem Slider, der einen Anfasser besitzt und über Strahlinteraktion bedient wird. Auch diese Form wird nah und weiter entfernt vom Benutzer getestet.

Der Slider ist als Hauptgegenstand dieser Benutzerstudie besonders geeignet: Die Slider-Interaktion beinhaltet alle oben genannten Gestaltungsaspekte und ermöglicht einen Vergleich dieser Kriterien zwischen den Menüformen.

Die Auswahl der Menüvarianten ermöglicht einen Vergleich der im VRGeo-Projekt eingesetzten Menüformen sowie den Vergleich von zwei verschiedenen Platzierungsvarianten von VR-Menüs: in unmittelbarer Nähe des Benutzers, so wie es die regelbasierte Positionierung vorsieht, oder weiter entfernt in der 3D-Szene, wie es z.B. in *Inside Reality* realisiert wurde (siehe Abbildung 4.3b auf Seite 30).

Die oben genannten Vermutungen über die Verbesserung einzelner Interaktionsaspekte werden im Rahmen der Benutzerstudie als Testhypothesen formuliert. Diese beziehen sich auf die drei Usability-Maße Effizienz, gemessen als Ausführungsgeschwindigkeit der Testaufgabe, Effektivität, repräsentiert durch die Fehlerrate bei der Interaktion, sowie die Zufriedenstellung. Letztere wird als subjektive Beurteilung der Menüvarianten durch die Testteilnehmer ermittelt.

1. Das Anwählen eines Schiebereglers in VR als Block-Element geschieht schneller als das Anwählen eines Schiebereglers mit Anfasser, wie er typischerweise in WIMP-Benutzeroberflächen implementiert ist.
2. Das Greifen und Einstellen eines Zielwertes mit einem WIMP-Schieberegler in VR dauert länger, je weiter dieser vom Benutzer entfernt ist.
3. Die Greif- und Einstellzeit des Block-Schiebereglers ist unabhängig von der Entfernung und nicht langsamer als mit einem WIMP-Regler.
4. Die Benutzung der 3D-Slider in VR ist schneller als die Benutzung der Regler auf PDA- und Tablet-PC.

5. Die Fehlerhäufigkeit (Highlighting des Reglers, Verfehlen des Reglers beim Greifversuch sowie Anzahl der Versuche, den korrekten Zielwert einzustellen) ist beim Block-Slider nicht schlechter als bei der Besten der anderen Varianten.
6. Die neuen 3D-Menüs mit Block-Elementen erzeugen bei den Benutzern eine bessere subjektive Bewertung und Akzeptanz als die anderen getesteten Menüformen.

In einer Nutzerstudie zeigte [Conrad 2005], dass die Testnutzer die Farben eines 3D-Objektes in einer VR-Szene mit PDA-Slidern schneller einstellen konnten als mit WIMP-ähnlichen 3D-Slidern. Die Ursachen für das schlechtere Abschneiden der 3D-Slider werden nicht beschrieben. Mit der hier vorgestellten Studie soll gezeigt werden, dass 3D-Steuerungselemente, wenn sie an die speziellen Anforderungen der 3D-Interaktion angepasst sind, dem PDA-Interface überlegen sind.

Zusätzlich zu diesem Vergleichstest wurden in einer VRGeo-Demonstration während eines VRGeo-Meetings die Benutzer in informellen Gesprächen zu ihren Eindrücken und Erfahrungen befragt. Während der Demonstration konnten die Benutzer zwischen PDA, Tablet-PC und 3D-Menüs zur Steuerung des Demonstrators wählen und alle Varianten testen.

## 8.1 Methode

Für die Vergleichstests der Hypothesen eins bis fünf dient die Menüform als unabhängige Variable. Die Menüform umfasst insgesamt sieben Varianten: *Block-nah* und *Block-fern* für die nah und entfernt dargestellten 3D-Menüs mit Block-Elementen und entfernungsunabhängiger Slider-Interaktion, *WIMP-nah* und *WIMP-fern* für die Variante mit Anfasser und Strahlinteraktion, sowie *PDA*, *Tablet-1* und *Tablet-2*. Die 3D-Menüs sind in der nahen und entfernten Darstellung gleich groß, werden also nicht entfernungsabhängig skaliert. Die zweite Tablet-PC Variante wurde notwendig, nachdem bei einer Zwischenauswertung ein Problem mit der Softwarekonfiguration auf dem Tablet-PC festgestellt wurde. Da die Probleme zu erheblichen Performance-Einbußen und Benutzerfrustration führten, wurde beschlossen, die restlichen Probanden mit einer verbesserten Tablet-Konfiguration arbeiten zu lassen. Die Daten für die ursprüngliche Konfiguration werden hier dennoch aufgeführt, um die Auswirkungen von kleinen Softwarefehlern auf die Performance und Nutzerakzeptanz zu verdeutlichen. Die Ursachen für die Probleme werden im Abschnitt 8.2.4 bei den Ergebnissen erläutert.

Die Zeitmessungen umfassen zwei Werte: Die Zeit vom Aufruf des Kontextmenüs bis zum ersten Greifen des Schiebereglers (*Greifzeit*) und die Zeit vom ersten Greifen bis zum Loslassen des Reglers bei korrekt eingestelltem Wert (*Einstellzeit*). Die *Gesamtinteraktionszeit* wird nachträglich als Summe der zwei genannten errechnet. Die Fehlerhäufigkeiten sind Zählvariablen: *Highlighting* gibt an, wie oft der Regler mit dem Strahl berührt wurde, *Fehlclicks*

bezieht die Klicks neben den Slider, *Versuche* steht für die Anzahl der Einstellvorgänge, bis der Zielwert erreicht wird. Aus technischen Gründen können nicht alle Variablen für jede Menüform erhoben werden: Für PDA und Tablet-PC liegt aufgrund der Interaktionsform keine Zählung des Highlightings vor und die Anzahl der Fehlklicks beim Greifen des Reglers auf dem Tablet-PC kann nicht erfasst werden. Alle Zählungen und Zeiten werden automatisiert in Logfiles erfasst.

Um Reihenfolgeeffekte auszuschließen, wird die Abfolge der Testdurchläufe randomisiert. Die Benutzer können die Interaktion mit jeder Menüform direkt vor dem jeweiligen Test trainieren. Die Trainingszeit ist nicht vorgeschrieben oder begrenzt, um der unterschiedlichen Erfahrung der Benutzer Rechnung zu tragen.

### 8.1.1 Materialien

Der Test wird in einem abgedunkelten Raum mit dem TwoView-Display durchgeführt. Lediglich der rückwärtige Teil des Raumes ist beleuchtet, um eine Orientierung im Raum zu ermöglichen. Außer der Testperson und dem Testleiter sind keine weiteren Personen im Raum anwesend. Als 3D-Interaktionsgerät kommt ein Stylus mit zwei nebeneinander liegenden Buttons zum Einsatz. Das 6-DOF-Tracking des Stylus wird durch vier *ARTtrack1*-Kameras und einem angeklebten Target realisiert. Der PDA ist ein HP iPAQ 5450, der ebenfalls über ein Target getrackt wird. Als Button dient der Taster links oben am Gehäuserand. Das PDA-Betriebssystem ist Windows Mobile 2003 Phone Edition. Beim Tablet-PC handelt es sich um ein PaceBlade SlimBook P120 mit Windows XP. Beide Geräte kommunizieren über WLAN mit der VR-Applikation. Die PDA- und Tablet-PC-Software wird mit Microsoft .NET Compact Framework entwickelt. Für die VR-Software kommt Avango zum Einsatz. Die Benutzer tragen eine 3D-Shutterbrille der Marke *CrystalEyes* mit zusätzlichen, aufgeklebten Polarisationsfiltern und Tracking-Targets.

Der Fragebogen umfasst insgesamt 33 Items. 18 Items erfassen das subjektive Empfinden des Schwierigkeitsgrades von Teilaufgaben, fünf Items erfassen die subjektive Einstellung zur Präferenz von Aspekten zu 3D-Schiebereglern, drei Items ermitteln die Präferenz zwischen den 3D-Reglern und denen auf PDA/Tablet-PC. Anschließend vergeben die Benutzer Schulnoten für die angebotenen Menü-Interaktionsformen (vier Items) und bewerten ihre Erfahrungen mit Virtual Reality und PDA/Tablet-PC. In einer abschließenden Frage können die Benutzer Vorschläge und Kritik frei äußern.

Der drei Seiten umfassende Fragebogen kann im Anhang ab Seite 103 eingesehen werden. Auf den darauf folgenden Seiten werden die Mittelwerte aller Antworten angegeben und die freien Antworten der Benutzer sowie die Beobachtungen des Testleiters aufgelistet.

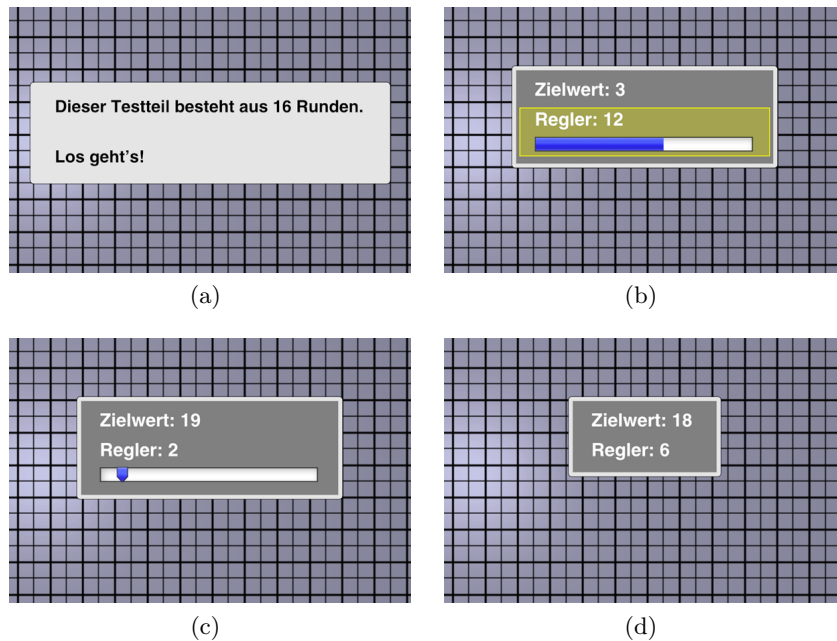
### 8.1.2 Stichprobe

Die insgesamt 19 Probanden der Studie sind Mitarbeiter, Studentische Hilfskräfte und Praktikanten der Abteilung Virtual Environments am Fraunhofer Institut für Intelligente Analyse- und Informationssysteme, im Alter zwischen 20 und 45 Jahren, davon 17 Männer und zwei Frauen. Unter den Teilnehmern befinden sich sowohl VR-Anfänger als auch erfahrene VR-Entwickler. Alle Teilnehmer haben den Test vollständig durchgeführt. Lediglich vereinzelte Messwerte sind wegen technischer Störungen, wie z.B. einem Netzwerk-Ausfall, nicht in die Auswertung eingeflossen. Zwölf Probanden haben den Test mit der ersten Tablet-PC-Version durchgeführt, sieben haben die verbesserte Version getestet.

### 8.1.3 Durchführung

Nachdem die Benutzer den Raum betreten, werden diese mit der Funktionsweise der beteiligten Geräte vertraut gemacht und der Ablauf des Tests erläutert. Die Testperson steht mittig vor dem TwoView-Display, ca. 1,3m von der Projektionsfläche entfernt. Die VR-Szene besteht aus einem großen virtuellen Raum mit schwarzen Linien auf graublauen Wänden. Die Szene ermöglicht dem Benutzer die räumliche Orientierung und eine Gewöhnung an die stereoskopische VR-Darstellung. Das Testprogramm gibt die zufällige Reihenfolge der Tests vor, so dass der Benutzer jeweils nach dem Start eines Testdurchlaufs vom Versuchsleiter die benötigten Geräte angereicht bekommt: den PDA, den Tablet-PC samt 6-DOF-Stylus oder ausschließlich den Stylus für die Interaktion mit den 3D-Menüs. Für den Testdurchlauf mit dem Tablet-PC kann der Nutzer frei entscheiden, ob er das Gerät auf dem Arm tragen oder vor sich auf ein Pult stellen möchte.

Zu Beginn eines Testdurchlaufs befindet sich eine virtuelle Informationstafel vor dem Benutzer (Abbildung 8.1a). Für die Tests Block-nah, WIMP-nah, PDA und Tablet-PC ist die Tafel 1m vom Benutzer entfernt, für die Tests Block-fern und WIMP-fern beträgt die Entfernung etwa 2,5m. Zeigt der Benutzer mit dem Strahl auf die Tafel und klickt diese mit der Kontextmenütaste an, verschwindet die Tafel und es erscheint auf gleicher Höhe ein Menüpanel (in 0,8m bzw. 2,3m Entfernung vom Benutzer). Das Menü enthält zwei Textzeilen mit dem Zielwert und dem aktuellen Wert des Schiebereglers (Abbildung 8.1d). Bei den 3D-Menü-Tests (Block und WIMP) ist außerdem ein Schieberegler im Menü enthalten (Abbildungen 8.1b und 8.1c). Bei den PDA- und Tablet-PC-Tests erscheint der Regler auf dem Display des jeweiligen Geräts. Der Benutzer muss den Regler greifen, auf den richtigen Wert einstellen und loslassen. Steht der Regler nicht auf dem korrekten Wert, muss der Benutzer den Reglerwert korrigieren. Ist der Zielwert erreicht, verschwindet das Menü und die Infotafel erscheint erneut. Darauf steht nach jedem Durchgang die benötigte Zeit und die Anzahl der noch zu absolvierenden



**Abbildung 8.1:** Screenshots von der Benutzerstudie – (a) Die Informationstafel erscheint zwischen den Testdurchläufen und informiert den Proband über die verbleibende Rundenzahl und seine Durchschnittszeiten. (b) Beim Block-Slider kann der Benutzer den gesamten Reglerblock greifen und auf den Zielwert einstellen. (c) Der WIMP-Slider muss am blauen Anfasser gegriffen und auf den Zielwert eingestellt werden. (d) Während der PDA- bzw. Tablet-PC-Tests werden im Menü nur die Start- und Zielwerte angezeigt. Der Slider befindet sich auf dem Touchscreen des jeweiligen Gerätes.

Durchgänge mit dieser Menüform. Zu Beginn einer neuen Menüform kann der Benutzer beliebig lange trainieren. Der Ablauf ist identisch, jedoch werden die Runden nicht gezählt und keine Daten im Logfile gespeichert. Der Testleiter beendet das Training nach Aufforderung des Probanden. Für jede der sechs Menüformen absolviert der Benutzer 16 Testdurchläufe mit verschiedenen Regler-Start- und Zielwerten in zufälliger Reihenfolge. Nach Beendigung der 16 Durchläufe erscheint eine Infotafel mit der Durchschnittszeit für diese Menüform. Nach erneutem Klick auf diese Tafel beginnt der nächste Test mit einer anderen Menüform. Die Gesamtzahl der Einstellvorgänge pro Proband beträgt  $6 \cdot 16 = 96$ , zuzüglich der nicht gezählten Trainingsdurchläufe.

Der Wertebereich des Sliders reicht von 0 bis 20. Die 16 Start- und Zielwertpaare sind für jede Menüform identisch und umfassen Wertdifferenzen zwischen drei und 17 Einheiten, sowohl in positiver als auch in negativer Richtung.

Im Anschluss an den Praxistest beantworten die Probanden einen dreiseitigen Fragebogen zu ihren Eindrücken über die Schwierigkeit der einzelnen Aufgaben, ihren Bewertungen, Präferenzen und ihrem Erfahrungshorizont. Am Ende des Fragebogens können die Probanden



sowohl schriftlich als auch im Gespräch Verbesserungsvorschläge machen, Kritik äußern oder Auffälligkeiten schildern.

Die Dauer des gesamten Tests inkl. Beantwortung des Fragebogens beträgt ca. 25 bis 30 Minuten pro Proband. Der Testleiter fertigt während des praktischen Tests Notizen zu Auffälligkeiten oder Bemerkungen der Versuchspersonen an.

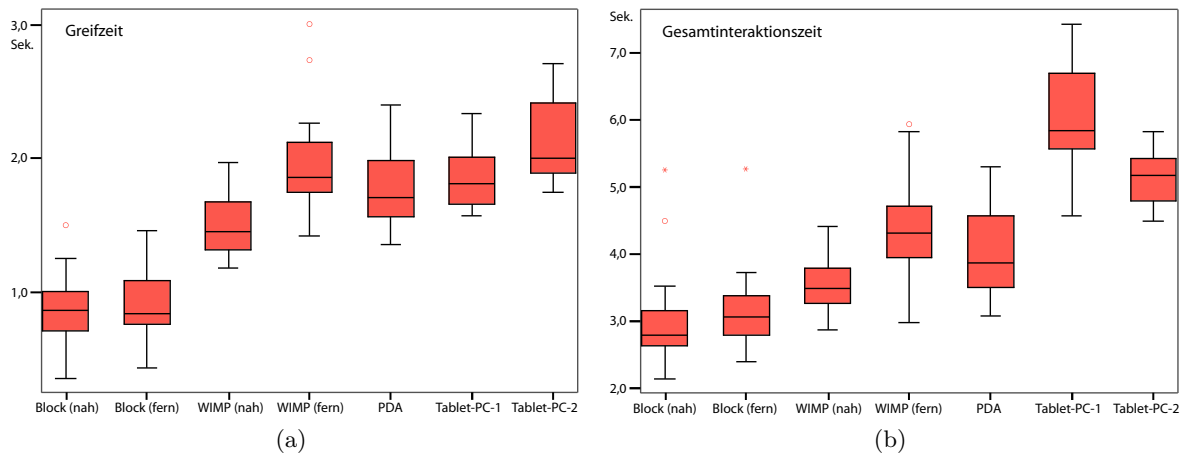
## 8.2 Ergebnisse

Alle Probanden haben die Tests vollständig durchgeführt und den Fragebogen beantwortet. Die Auswertungen erfolgten in SPSS 13. Die Werte der jeweils 16 Durchgänge für jede Menüform wurden pro Proband gemittelt und anschließend mit einer einfaktoriellen Varianzanalyse (ANOVA) und dem Scheffé Post-Hoc-Test ausgewertet. Durch den geringen Stichprobenumfang können selbst praktisch bedeutsame Mittelwertunterschiede nicht immer durch statistische Signifikanz auf dem gewählten  $\alpha$ -Niveau von 5% abgesichert werden [Bortz 2005; Beller 2004]. Aus diesem Grund werden hier auch Tendenzen aufgezeigt, wenn die Mittelwerte einen relevanten Unterschied nahelegen.

### 8.2.1 Performance

Es wurde erwartet, dass die Benutzer den Block-Slider schneller greifen können als die anderen Varianten. Zusätzlich sollte die Greifzeit für die Block-Slider, im Gegensatz zu den 3D-WIMP-Slidern, nicht von der Entfernung abhängig sein. Verglichen werden die Greifzeit der sieben unterschiedlichen Menüvarianten. Die ANOVA bestätigt signifikante Unterschiede zwischen den Varianten ( $F = 47,58$ ;  $p < 0,001$ ). Dabei zeigen sich im Post-Hoc-Test für die unterschiedlichen Entfernungen bei den Block-Slidern keine Unterschiede, wohl aber bei den WIMP-Slidern. Während die Block-Slider mit Mittelwerten von 0,87s und 0,88s für nah und fern praktisch gleichschnell gegriffen werden, dauert das Greifen des fernen WIMP-Sliders mit 1,95s signifikant länger als bei der nahen Positionierung mit 1,50s ( $p = 0,002$ ). Das Greifen der Block-Slider ist darüber hinaus signifikant schneller als bei allen anderen Menüformen (jeweils mit  $p < 0,001$ ). Die Benutzer brauchen im Schnitt 1,78s für das Greifen des PDA-Sliders sowie 1,87s für Tablet-1 und 2,15s für Tablet-2. Abbildung 8.2a zeigt die Greifzeiten im Boxplot.

Als nächstes wird die Einstellzeit betrachtet. Auch hier zeigt die ANOVA signifikante Gruppenunterschiede ( $F = 32,14$ ;  $p < 0,001$ ). Die Einstellzeiten der Block-Slider unterscheiden sich mit 2,15s (nah) bzw. 2,29s (fern) nicht signifikant. Die gemessenen mittleren Einstellzeiten für die WIMP-Slider unterscheiden sich mit 2,01s (nah) und 2,50s (fern) zwar deutlich, jedoch zeigt sich hier lediglich eine Tendenz ( $p = 0,149$ ). WIMP-nah ist im Mittel zwar etwas schneller



**Abbildung 8.2:** *Interaktionszeiten in der Benutzerstudie* – Die Graphen zeigen die nach den Menüarten aufgeschlüsselten Greif- und Gesamtzeiten. Der Vergleich der Graphen zeigt, dass die Performance-Überlegenheit der Block-Slider zu großen Teilen durch das schnellere Greifen zustande kommt. (a) Greifzeit, gemessen vom Aufruf des Menüs bis zum ersten Greifen des Sliders. (b) Gesamtzeit, gemessen vom Aufruf des Menüs bis zum Erreichen des korrekten Zielwertes.

als die Block-Varianten, jedoch nicht signifikant. Damit können die ersten drei Hypothesen bestätigt werden.

Des Weiteren wurde erwartet, dass die Benutzer mit den 3D-Slidern schneller den Zielwert erreichen als mit den PDA- und Tablet-Slidern. Zur Überprüfung wird die gleiche Analyse nochmals mit der Gesamtzeit des Einstellvorgangs durchgeführt (vgl. Abbildung 8.2b). Die ANOVA zeigt signifikante Gruppenunterschiede ( $F = 37,93$ ;  $p < 0,001$ ). Die beiden Block-Varianten sind mit 3,02s (nah) und 3,16s (fern) signifikant schneller als PDA mit 4,04s, Tablet-1 mit 6,07s und Tablet-2 mit 5,13s ( $p < 0,015$ ). Für die beiden WIMP-Slider zeigt die Statistik jedoch keine durchgehend signifikante Überlegenheit gegenüber PDA und Tablet-PC: WIMP-nah ist mit 3,50s im Mittel deutlich schneller als die Tablet-PCs ( $p < 0,001$ ) und mit einer Differenz von 0,54s immerhin tendenziell schneller als der PDA ( $p = 0,398$ ). Für WIMP-fern zeigt die Statistik zwar *keine signifikanten* Unterschiede gegenüber PDA und Tablet-2, WIMP-fern ist jedoch im Mittel 0,4s langsamer als der PDA ( $p = 0,739$ ), aber 0,68s schneller als Tablet-2 ( $p = 0,499$ ) und 1,62s schneller als Tablet-1 ( $p < 0,001$ ). Hypothese vier kann daher nur mit Einschränkungen bestätigt werden, die im Diskussionsteil erörtert werden.

### 8.2.2 Fehlerrate

Für den Vergleich der Fehlerhäufigkeit werden drei Variablen von den Menüvarianten verglichen: Highlighting, Fehlversuche beim Greifen und die Anzahl der Versuche bis zum Erreichen des Zielwerts. Es wurde erwartet, dass die Block-Slider in allen drei Disziplinen mindestens

gleich gut abschneiden wie die anderen Varianten. Die ANOVAs zeigen erneut signifikante Unterschiede (Highlight:  $F = 44,59$ ;  $p < 0,001$ , Fehler beim Greifen:  $F = 15,76$ ;  $p < 0,001$ , Anzahl der Versuche:  $F = 31,28$ ;  $p < 0,001$ ). Der Block-Slider wird seltener mit dem Strahl berührt als der WIMP-Slider ( $p < 0,051$ ). Die Benutzer verfehlen beim Anklicken tendenziell seltener den Block-Slider als den WIMP-Slider ( $p < 0,172$ ), jedoch greifen sie den PDA-Slider ebenso zuverlässig. Die Anzahl der Versuche, bis der Wert korrekt eingestellt ist, unterscheidet sich nur gegenüber dem Tablet-1 signifikant, der in dieser Disziplin deutliche Mängel aufweist und deshalb im Laufe des Tests durch die überarbeitete, zweite Tablet-Version ersetzt wurde.

Der Blick auf die Fehlerhäufigkeiten gibt Auskunft über die Effektivität der Interaktionen. Die Block-Slider zeigen durchweg niedrigere Fehlerraten als die anderen Arten. Für das Greifen der Block-Slider ist in mindestens 75% aller Testläufe nur ein Highlighting notwendig (siehe Tabelle 8.1). In maximal 3,6% der Fälle berühren die Benutzer den Regler mehr als zwei mal. Das Greifen der WIMP-Slider gelingt dagegen nur bei 52% (nah) bzw. 27% (fern) nach der ersten Strahlberührung. In 10% (nah) bzw. 29% (fern) der Fälle sind mehr als zwei Berührungen notwendig. Die Erfolgsquote beim Klick auf den Slider zeigt ebenfalls Nachteile für die WIMP-Slider. Die Benutzer verfehlen beim ersten Klick nur in maximal 3% der Fälle die Block-Slider, während WIMP-nah in 13% der Fälle verfehlt wird und WIMP-fern in 27% der Fälle. Der erste Versuch, den korrekten Zielwert einzustellen gelingt bei allen Reglerarten mit 83%–90% in etwa gleich gut. Lediglich Tablet-1 zeigt mit nur 56% ein deutliches Problem.

**Tabelle 8.1:** Erfolgsquote der Slider-Interaktion in der Benutzerstudie – Eine hohe Erfolgsquote bedeutet eine gelungene Interaktion mit geringer Fehlerrate. Die rot markierten Werte zeigen die Schwachstellen der einzelnen Menüarten. Die schlechten Werte beim Highlighting der WIMP-Slider weisen darauf hin, dass die Benutzer Probleme haben, zuverlässig mit dem Strahl auf den Anfasser zu zeigen. Entsprechend sinkt auch die Erfolgsrate beim Greifen des Anfassers. Der schwache Wert bei den Einstellversuchen führte beim Tablet-1 zur Entdeckung eines Touchscreen-Treiberproblems, der bei Tablet-2 korrigiert wurde.

Menüart	Highlighting		Greifversuche	Einstellversuche
	1. Versuch	3 und mehr Versuche	1. Versuch	1. Versuch
Block (nah)	79%	1,3%	98%	83%
Block (fern)	75%	3,6%	97%	85%
WIMP (nah)	<b>52%</b>	<b>10%</b>	<b>87%</b>	90%
WIMP (fern)	<b>27%</b>	<b>29%</b>	<b>73%</b>	83%
PDA	–	–	95%	89%
Tablet-1	–	–	–	<b>56%</b>
Tablet-2	–	–	–	86%

Somit kann Hypothese fünf bestätigt werden: Der Block-Slider zeigt in der Fehleranalyse keine Schwächen gegenüber den anderen Verfahren und ist diesen in vielen Punkten überlegen.

### 8.2.3 Bewertung und Akzeptanz

Während die bisherigen Auswertungen auf den Daten der Logfiles beruhen, kann die sechste Hypothese durch Auswertung des Fragebogens überprüft werden. Die Auswertung mittels ANOVA ergibt zwar signifikante Ergebnisse, im Post-Hoc-Test scheitern jedoch viele Vergleiche aufgrund der geringen Stichprobengröße am notwendigen  $\alpha$ -Niveau von 5%. Dennoch lassen sich in solchen Fällen anhand der Mittelwerte deutliche Tendenzen ablesen, die im Folgenden mit in die Auswertung einfließen. Die Mittelwerte aller Items sind im Anhang ab Seite 107 zu finden.

Es wurde angenommen, dass die Block-Regler im subjektiven Empfinden der Benutzer als einfacher zu bedienen wahrgenommen werden und die Akzeptanz im Vergleich zu den anderen Reglerarten bessere Werte erreicht. Die Benutzer waren aufgefordert, die drei Phasen der Reglerinteraktion (Greifen, Wertänderung und Zielwert exakt einstellen) in ihrem Schwierigkeitsgrad zu bewerten. Die fünfstufige Skala reichte von eins, für „sehr schwierig“, bis fünf für „sehr einfach“.

Das subjektive Schwierigkeitsempfinden entspricht in weiten Teilen der gemessenen Interaktionsgeschwindigkeit. Die drei schnellsten Varianten Block-nah, Block-fern und WIMP-nah werden von den Probanden durchweg für einfacher befunden als die langsameren. Tablet-1 wird, seiner Performance entsprechend, am schlechtesten bewertet.

Das Greifen des Block-Reglers empfinden die Benutzer einfacher als bei allen anderen Arten ( $p < 0,085$ ). Besonders schlecht werden WIMP-fern und Tablet-1 bewertet. Das Verändern des Reglerwertes wird bei beiden Blockvarianten zusammen mit WIMP-nah für einfach befunden. PDA, WIMP-fern und Tablet-2 liegen fast gleichauf im mittleren Schwierigkeitsgrad, während Tablet-2 etwas schlechter bewertet wird. Das exakte Einstellen des Zielwertes wird bei Block-nah am besten bewertet. Block-fern und WIMP-nah werden ebenfalls vergleichsweise gut bewertet. Während bei PDA und Tablet-2 das Einstellen tendenziell noch einfach geht, werden WIMP-fern und Tablet-1 eher als schwierig eingeschätzt.

Die beiden Blockvarianten liegen mit Werten zwischen 3,74 und 4,74 in allen drei Kategorien im Bereich einfach bis sehr einfach. Bei WIMP-fern wird vor allem das Greifen mit 1,89 als schwierig bewertet. WIMP-nah schneidet mit Werten zwischen 3,42 und 3,89 nur wenig schlechter ab als die Block-Slider. Während PDA und Tablet-2 mit Werten zwischen 2,86 bis 3,29 im mittleren Schwierigkeitsbereich liegen, wird die erste Tablet-Variante mit Werten zwischen 1,75 und 2,42 für durchweg schwierig zu bedienen eingestuft.

Ein ähnliches Bild ergibt sich bei den Fragen, bei denen die Benutzer angeben sollten, welche der genannten Optionen sie bevorzugen würden. Auf die Frage, ob sie lieber mit PDA/Tablet-PC den Wert einstellen oder mit einem 3D-Regler in VR, entscheiden sich die Probanden mit Werten zwischen 3,86 und 4,67 mehrheitlich für die VR-Lösung. Erneut schneidet Tablet-1 besonders schlecht ab, während Tablet-2 nur wenig besser als der PDA bewertet wird und auch im direkten Vergleich mit dem PDA knapp besser abschneidet. Das Greifen des ganzen Reglerblocks wird dem Greifen eines Anfassers bevorzugt, bei den entfernten Menüs sehr deutlich, bei den nahen nur noch tendenziell. Ähnlich verhält es sich mit dem Einstellen des Wertes: Die Handrotation der Block-Slider wird vor allem bei den entfernten Menüs bevorzugt. Die Werteinstellung mit dem WIMP-Regler war im Nahbereich praktisch gleichwertig. Diese Einschätzung deckt sich mit den gemessenen Zeiten für das Greifen und Einstellen. Auch die Bevorzugung der nahen Regler gegenüber den fernen passt zu den Messungen.

Am Ende des Fragebogens sollten die Benutzer Schulnoten von eins, für „sehr gut“, bis fünf, für „mangelhaft“, für die Interaktionsarten vergeben. Die Noten entsprechen der bisherigen Auswertung: Die Block-Regler werden mit einer Durchschnittsnote von 1,47 für sehr gut befunden, gefolgt von den WIMP-Reglern mit einem Durchschnitt von 2,45. PDA und Tablet-2 liegen mit 3,16 bzw. 3,29 dicht beieinander, während die Probleme mit Tablet-1 zu einer Note von 4,5 führen. Die genannten Ergebnisse bestätigen die höhere Akzeptanz und bessere Bewertung der Block-Slider und somit die sechste Hypothese.

#### **8.2.4 Beobachtungen, Kommentare und Kritik**

Sowohl bei der Beobachtung der Probanden während der Testdurchführung als auch durch die Möglichkeit der Kritik im Fragebogen und im Gespräch konnten weitere interessante Aspekte festgehalten werden. Als störend erweist sich vor allem bei der PDA- und Tablet-Benutzung die *CrystalEyes*-Brille: Durch ihr Gewicht und dem meist lockeren Sitz rutscht die Brille beim Herabblicken auf das PDA/Tablet-Display und schränkt zudem durch ihre Form das Gesichtsfeld ein. Der Blickwechsel zwischen den Displays ist der Hauptkritikpunkt an der Interaktion mit den Slidern auf PDA/Tablet.

Die Kritik an der ersten Tablet-Version bezieht sich auf einen zu kleinen Anfasser, der mit dem klobigen Stylus nur schwer zu greifen sei sowie auf den schlecht funktionierenden Touchscreen: Die User verlieren während der Einstellphase immer wieder den Slider und müssen erneut auf das Display blicken um den Anfasser neu zu greifen. Wegen der Regelmäßigkeit dieser Probleme untersuchten wir das Phänomen und stellten fest, dass der Touchscreen-Treiber bei kurzzeitigem Innehalten auf einer Position einen „rechts-Klick“ simuliert und dabei den Anfasser loslässt. Die Zeitspanne dafür ist so kurz, dass allein der Blickwechsel auf das VR-Display zur Kontrolle des eingestellten Wertes dafür ausreicht. Im Ergebnis kann der

Benutzer den Wert nicht korrigieren sondern muss den Regler neu greifen. Erst nach einem Treiber-Update konnte die störende Funktion abgeschaltet werden. Die Kritik am zu kleinen Anfasser wurde durch eine Neuimplementierung des Schiebereglers gelöst und der Anfasser gegenüber der Windows-Standardgröße deutlich vergrößert. Diese zwei Änderungen führten zur zweiten Tablet-PC-Version.

Weitere Kritikpunkte, Kommentare und Beobachtungen sind im Anhang ab Seite 111 aufgelistet.

### 8.3 Zusammenfassung und Diskussion

Die Benutzerstudie bestätigt die angestrebten Usability-Verbesserungen des neuen 3D-Menüs: Bei Performance, Fehlerrate, Akzeptanz und Bewertung können die 3D-Menüs zumeist bessere Werte erzielen als die anderen Menüvarianten.

Wie nach Fitts' Law zu erwarten, kann der Slider dank vergrößerter Fläche deutlich schneller gegriffen werden. Dadurch entstehen bei der Selektion des Menüeintrags seltener Fehler in Form von mehrfachem Berühren des Reglers mit dem Strahl oder ungewollte Klicks neben den Regler. Dies gilt nicht nur für den Slider sondern für alle Elemente des neuen 3D-Menüs. Dank Block-basiertem Ansatz sind alle Menüelemente gleich groß, so dass diese Verbesserungen direkt auf das ganze Menü übertragen werden können.

Die Slider-Interaktion zeigt sich im Test wie geplant unabhängig von der Entfernung zum Benutzer. Zwar kann der Zielwert nicht schneller erreicht werden als bei den anderen Varianten, jedoch dauert es auch nicht länger als mit der gewohnten Strahlinteraktion und die Zuverlässigkeit, mit der der Zielwert eingestellt werden kann, ist ebenso hoch.

Die Positionierungsfunktion der neuen Menüs wird in diesem Test zwar nicht explizit berücksichtigt, jedoch scheint vor dem Hintergrund der Testergebnisse auch dieser Ansatz richtig zu sein: Das Regelsystem sorgt für eine Position nah am Benutzer, die von den Probanden im Test bevorzugt wird. Eine entfernte Positionierung verunsichert die Benutzer: Selbst bei absolut identischer Interaktion halten die Benutzer den entfernten Block-Slider für etwas schwieriger zu bedienen als den nahen.

Die in den Hypothesen formulierte grundsätzliche Überlegenheit von 3D-Menüs erweist sich als falsch. Die Bereitstellung von Menüs in VR führt nicht zwangsläufig zu einer verbesserten Interaktion gegenüber externen Menüs, wie auch die Studie von Conrad zeigt. Zwar entfällt der Blickwechsel zwischen den Displays, der Zeitgewinn wird jedoch durch falsche Positionierung oder zu kleine Bedienelemente zunichte gemacht.

Solch unvorteilhafte Eigenschaften oder unzuverlässige Funktionen führen zu einem drastischen Einbruch der Performance und zur Erhöhung der Fehlerraten, die sich in einer erhöhten

Nutzerfrustration äußern. Dies ist während des Tests besonders an den Reaktionen während der Bedienung des fehlerhaften Tablet-1 und den entfernten WIMP-Slidern zu beobachten. Die Fehler und Ungenauigkeiten führen nicht nur zu einer langsameren Interaktion, die Nutzer müssen sich bei der Interaktion auch sichtbar mehr anstrengen und konzentrieren. Im Vergleich zu den leichter zu bedienenden Varianten führt das zu schlechteren Bewertungen.

Durch die Korrekturen am Tablet-PC verbessern sich die Messwerte und die Bewertungen, jedoch nicht in dem Maße, wie wir es mit Blick auf die VRGeo-Meetings erwartet hätten. Dort wurde die Tablet-Version von den Benutzern besser angenommen als die PDA-Version. Dies kann mehrere Ursachen haben: Die Komplexität der PDA-Menüs in dieser Nutzerstudie ist sehr gering, wohingegen die Menüs im VRGeo-Demonstrator sehr umfangreich sind. Die Performance der .Net-GUI und die WLAN-Anbindung lassen den PDA daher teilweise träge auf Benutzereingaben reagieren. Zudem ist in der hier vorgestellten Studie keine 6-DOF-Interaktion enthalten. Der PDA ist für eine komplexe 6-DOF-Interaktion nur bedingt geeignet, da er üblicherweise in der nicht-dominanten Hand gehalten wird. Der Tablet-PC hingegen kann mit seiner schnelleren GUI, der Stylus-Interaktion und dem größeren Display im VRGeo-Demonstrator Vorteile verbuchen, nicht jedoch in unserer Studie.

Die Testaufgabe besteht aus einer einzelnen, speziellen Interaktion und stellt nur einen kleinen Ausschnitt aus einem realen Anwendungsszenario dar. Die Studie zeigt jedoch, dass durch Anpassung bestimmter Gestaltungsparameter und eine an die Besonderheiten von VR angepasste Interaktion die Fehlerrate deutlich gesenkt, die Interaktion beschleunigt und die Zufriedenstellung der Nutzer verbessert werden kann. Dabei können die neu entwickelten 3D-Menüs in allen Testkriterien überzeugen und erhalten durchweg bessere Bewertungen als die anderen Menüformen. Mit der Integration der 3D-Menüs in den VRGeo-Demonstrator können die Erfahrungen aus der Nutzerstudie im Anwendungskontext überprüft werden. Im folgenden Abschnitt wird die Evaluation im Rahmen eines VRGeo-Meetings beschrieben.

## 8.4 3D-Menüs im VRGeo-Demonstrator

Der VRGeo-Demonstrator bietet in der aktuellen Version drei Menüformen für die Systemsteuerung: Wie in Abschnitt 5.4 ab Seite 38 beschrieben, verfügt die Software über ein PDA- und ein Tablet-PC-Interface. Als dritte Variante wurden die 3D-Menüs implementiert. Während PDA und Tablet-PC einen identischen Funktionsumfang bieten, enthält das 3D-Menü eine Reihe weiterer Untermenüs, verzichtet jedoch auf das *Color Tool* zum Einstellen von Farbtabelle und Transparenzen für die Darstellung der seismischen Volumendaten.

Bis zu sechs Benutzer können im Demonstrator gleichzeitig interagieren, wobei das TwoView-Display dynamisches Headtracking für zwei Benutzer gleichzeitig unterstützt und auch aus

Platzgründen üblicherweise nicht mehr als zwei Nutzer aktiv mit der Applikation arbeiten. Es stehen insgesamt sechs 3D-Brillen, zwei PDA, drei Styli (jeweils mit 6-DOF-Tracking) und ein Tablet-PC zur Verfügung. Die Strahlen der Benutzer haben unterschiedliche Farben. Öffnet ein Benutzer ein Kontextmenü, wird dieses mit einem farbigen Rahmen entsprechend der Nutzerfarbe dargestellt und kann nur von diesem Benutzer bedient werden. Jeder andere Benutzer kann unabhängig davon ein eigenes Kontextmenü öffnen.

Die ca. 15 Teilnehmer des VRGeo-Meetings konnten den Demonstrator in einer zweistündigen Demosession testen und die verschiedenen Menüformen ausprobieren. Viele Teilnehmer kannten die PDA- und Tablet-Menüs aus früheren Demos. Die Resonanz auf die 3D-Menüs war durchweg positiv: Die Teilnehmer bevorzugten schon nach kurzem Test weitgehend die 3D-Kontextmenüs in VR und benutzten ergänzend den Tablet-PC zum Einstellen der Farbpalette. Der PDA wurde meist nach kurzem Test aus der Hand gelegt. Der Anspruch, die 3D-Menüs sowohl für Anfänger als auch für erfahrene Nutzer zugänglich zu machen, konnte erfüllt werden. Vereinzelt nutzten Teilnehmer während der Demo zum ersten Mal eine Virtual-Reality-Anwendung. Jedem Teilnehmer war nach kurzer Einführung in die Funktionsweise der Interaktionsgeräte die Bedienung der Menüs verständlich.

Die Benutzer gaben während der Demo und in einer Besprechung am Ende des Meetings größtenteils positive Rückmeldungen: Die Benutzung von integrierten Menüs sei einfacher und übersichtlicher, da keine zusätzlichen Geräte verwendet werden müssten und die Interaktion mit der Applikation dadurch vereinfacht würde. Die Probleme des Blickwechsels bei Verwendung der PDAs und Tablet-PCs wurden in vorherigen VRGeo-Meetings schon angesprochen und konnten durch die neuen Menüs gelöst werden. Die Positionierung und Skalierung der Menüs funktionierte fehlerfrei: Die Benutzer bestätigten, dass die Menüs jederzeit sinnvoll und gut bedienbar positioniert worden seien und die Beschriftung in jeder Situation lesbar gewesen sei. Auch die transparente Darstellung wurde explizit als sinnvoll und hilfreich hervorgehoben.

Die Anforderungen an die Texteingabe in VR konnten dagegen nicht zur vollen Zufriedenheit der Teilnehmer erfüllt werden. Die Quikwrite-Implementierung stieß erwartungsgemäß auf wenig Begeisterung. Die Einarbeitung in diese ungewöhnliche Texteingabemethode war in der kurzen Zeit unmöglich. Die virtuelle Tastatur konnte von allen Teilnehmern zur Eingabe von Objektbeschriftungen problemlos genutzt werden, jedoch war diese Form der Eingabe für längere Texte nicht geeignet.

Einige erfahrene Anwender wünschten sich neben der Bedienung von 3D-Menüs ein Äquivalent zu den von 2D-GUIs bekannten *Hotkeys* bzw. *Shortcuts*. Der schnelle Wechsel von Einstellungen und Funktionen sei im professionellen Einsatz durch eine Menübedienung nicht effizient genug. Als möglicher Lösungsansatz in VR wurden Sprachkommandos in Verbindung mit Gesten genannt. Als Ergänzung zur Strahlinteraktion soll der Benutzer über Sprache und



Gesten einzelne Funktionen der Software sowie die Menüs aufrufen und Einträge aktivieren können. Eine Sprachsteuerung für den VRGeo-Demonstrator ist im VRGeo-Projekt bereits in Entwicklung.

Im Rahmen der Konzeption des 3D-Menüs wurde im Abschnitt 5.5 bereits auf die Framerate-Abhängigkeit der Menüinteraktion hingewiesen. In den meisten Situationen ist diese Grundvoraussetzung im VRGeo-Demonstrator erfüllt. In den Momenten einer verminderten Framerate ist die Interaktion mit den virtuellen Objekten und damit auch die Menüinteraktion nur noch stark eingeschränkt möglich. Das bereits erwähnte *Multi-Frame-Rate-Rendering* wurde im Rahmen des VRGeo-Meetings in einer gesonderten Demonstration unter anderem mit den 3D-Menüs vorgestellt. Die gezeigten Resultate machen Hoffnung, dass die Render-Geschwindigkeit in Zukunft nur noch wenig Einfluss auf die Interaktionsqualität haben wird und die 3D-Menüs daher uneingeschränkt eingesetzt werden können.

Die Ergebnisse des VRGeo-Meetings belegen, dass die Anwender die 3D-Menüs gegenüber den PDA- und Tablet-PC-Menüs bevorzugen. Das Feedback war insgesamt sehr positiv und die Akzeptanz der 3D-Menüs war sehr hoch. Die Positionierung, Skalierung und Gestaltung des Menüs minimierte die Verdeckungsprobleme und machten das Menü in jeder Situation bedienbar. Das Menü eignet sich für die Steuerung des komplexen VRGeo-Demonstrators und ist sowohl von erfahrenen VR-Nutzern als auch von Anfängern leicht zu bedienen.



## 9 Zusammenfassung und Ausblick

In dieser Arbeit wurde ein 3D-Menü zur Steuerung komplexer immersiver VR-Anwendungen vorgestellt. Dabei wurden insbesondere die Anforderungen an die Interaktion mit dem VRGeo-Demonstrator berücksichtigt. Das vorgestellte Menüsystem orientiert sich im Leistungsumfang, bei der Darstellung und der Bedienung an den Konzepten von bekannten 2D-Menüs und ist dadurch sowohl für erfahrene wie auch unerfahrene Benutzer gleichermaßen verständlich und intuitiv bedienbar.

Die Gestaltungs- und Interaktionsaspekte des Menüs sind auf die speziellen Herausforderungen der Interaktion in Virtuellen Umgebungen optimiert. Dabei werden für viele Problemfelder konkrete Lösungsvorschläge vorgestellt: Die Zuverlässigkeit der Menüinteraktion wird durch das Block-Layout der Menüeinträge in einer Liste sichergestellt. Dadurch kann die Fehlerhäufigkeit bei der Auswahl eines Menüpunktes minimiert und die Interaktionsgeschwindigkeit deutlich verbessert werden. Eine situationsabhängige Skalierung des Menüs unterstützt die Bedienung durch gleichbleibende Interaktionswinkel und gewährleistet die Lesbarkeit der Menübeschriftung. Verdeckungs- und Erreichbarkeitsprobleme werden durch eine regelbasierte Positionierung, durch die Verwendung von Transparenzen und durch eine platzsparende Hierarchienavigation verbessert. Durch das Regelsystem wird das Menü an einer Position dargestellt, in der es für den Benutzer zuverlässig benutzbar ist, seine Sicht auf das Menü nicht durch störende Objekte beeinträchtigt und das Menü nicht am Bildrand abgeschnitten wird. Die Navigation in Hierarchien benötigt für die Darstellung der Menüstruktur nur ein Fenster und verhindert damit das Überfüllen der 3D-Szene mit weiteren Menüteilen. Das „Context-Control“-Konzept leistet durch die Zusammenfassung von Applikationsmenüs, Kontextmenüs, Dialogfenstern und Control-Paletten einen Beitrag zur einheitlichen, übersichtlichen und platzsparenden Bedienung innerhalb der VR-Applikation. Es integriert die Funktionen zur Systemsteuerung und Parameteränderung innerhalb einer Menüform. Für die Interaktion mit einem Schieberegler in VR wurde eine darstellungsunabhängige Interaktionsform entwickelt.

In einer Benutzerstudie wurden die bisher im VRGeo-Demonstrator verwendeten PDA- und Tablet-PC-Menüs mit zwei Varianten der 3D-Menüs verglichen. Dabei zeigte sich in Bezug auf Effizienz und Effektivität eine klare Überlegenheit des neu entwickelten Menüs bei gleichzeitig höherer Akzeptanz und Bewertung durch die Benutzer. In einer informellen

Studie mit den Teilnehmern eines VRGeo-Meetings konnte der Zuspruch zu den neuen Menüs ebenfalls bestätigt werden.

Trotz der vorgestellten Problemlösungen bleiben für die Zukunft weitere Herausforderungen an die Steuerung von VR-Applikationen bestehen. Für zwei in der Arbeit angesprochene Probleme sind bereits Lösungen in der Entwicklung. Dies betrifft zum einen die Abhängigkeit der Interaktionsqualität von der Framerate der VR-Applikation. Sinkt die Darstellungsgeschwindigkeit unter einen kritischen Wert, wird die zuverlässige Interaktion mit 3D-Elementen praktisch unmöglich. Das Verfahren des *Multi-Frame-Rate-Rendering* bietet einen vielversprechenden Lösungsansatz [Springer et al. 2007b, a]. Zum anderen wünschen sich gerade die erfahreneren Benutzer eine Art „Power-Modus“ für die Steuerung der Applikation. Nach Vorbild der „Hotkeys“ und „Shortcuts“ von Desktop-Systemen sollen dabei Gesten und Sprachkommandos zum Einsatz kommen. Die Sprachbedienung wird aktuell im VRGeo-Projekt erforscht und implementiert. Die ersten Erfahrungen, auch in Kombination mit den 3D-Menüs, sind vielversprechend.

# Literaturverzeichnis

## **3Dconnexion**

*3Dconnexion*. <http://www.3dconnexion.com>. – [Online, Version: 17. Juli 2007].

## **Accot u. Zhai 2003**

ACCOT, Johnny ; ZHAI, Shumin: Refining Fitts' law models for bivariate pointing. In: *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA : ACM Press, 2003, S. 193–200.

## **Agrawala et al. 1997**

AGRAWALA, Maneesh ; BEERS, Andrew C. ; MCDOWALL, Ian ; FRÖHLICH, Bernd ; BOLAS, Mark ; HANRAHAN, Pat: The two-user Responsive Workbench: support for collaboration through individual views of a shared space. In: *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. New York, NY, USA : ACM Press/Addison-Wesley Publishing Co., 1997, S. 327–332.

## **Andujar et al. 2006**

ANDUJAR, C ; FAIREN, M. ; ARGELAGUET, F.: A Cost-Effective Approach for Developing Application-Control GUIs For Virtual Environments. In: *3DUI '06: Proceedings of the 3D User Interfaces*. Washington, DC, USA : IEEE Computer Society, 2006, S. 45–52.

## **A.R.T.**

*Optische Tracking-Systeme von Advanced Realtime Tracking GmbH*. <http://ar-tracking.de>. – [Online, Version: 16. Juli 2007].

## **Ascension Flock Of Birds**

*Elektromagnetisches Tracking-System von Ascension Technology Corporation*. <http://www.ascension-tech.com/products/flockofbirds.php>. – [Online, Version: 16. Juli 2007].

## **Avango**

*Avango VR-Framework*. <http://sourceforge.net/projects/avango>. – [Online, Version: 14. Juli 2007].

## **Barco Active Infitec+**

*Single Active Infitec Stereo Projector*. <http://www.barco.com/VirtualReality/en/products/product.asp?element=2795>. – [Online, Version: 15. Oktober 2007].

## **Beller 2004**

BELLER, Sieghard: *Empirisch forschen lernen: Konzepte, Methoden, Fallbeispiele, Tipps*. Bern, Schweiz : Verlag Hans Huber, 2004.

## **Billinghurst et al. 1997**

BILLINGHURST, Mark ; BALDIS, Sisinio ; MATHESON, Lydia ; PHILIPS, Mark: 3D palette: a virtual reality content creation tool. In: *VRST '97: Proceedings of the ACM symposium on Virtual reality software and technology*. New York, NY, USA : ACM Press, 1997, S. 155–156.

## **Bortz 2005**

BORTZ, Jürgen: *Statistik für Human- und Sozialwissenschaftler*. Heidelberg : Springer Medizin Verlag, 2005.

**Bowman u. Hodges 1997**

BOWMAN, Doug A. ; HODGES, Larry F.: An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. In: *SI3D '97: Proceedings of the 1997 symposium on Interactive 3D graphics*. New York, NY, USA : ACM Press, 1997, S. 35–ff..

**Bowman et al. 2001**

BOWMAN, Doug A. ; KRUIJFF, Ernst ; LAVIOLA, Joseph J. ; POUPYREV, Ivan: An Introduction to 3-D User Interface Design. In: *Presence: Teleoper. Virtual Environ.* 10 (2001), Nr. 1, S. 96–108. – ISSN 1054–7460.

**Bowman et al. 2005**

BOWMAN, Doug A. ; KRUIJFF, Ernst ; LAVIOLA, Joseph J. ; POUPYREV, Ivan: *3D User Interfaces – Theory and Practice*. Reading, MA : Addison-Wesley, 2005.

**Bowman u. Wingrave 2001**

BOWMAN, Doug A. ; WINGRAVE, Chadwick A.: Design and Evaluation of Menu Systems for Immersive Virtual Environments. In: *VR '01: Proceedings of the Virtual Reality 2001 Conference (VR'01)*. Washington, DC, USA : IEEE Computer Society, 2001, S. 149–ff..

**Bryson 1993**

BRYSON, Steve: The Virtual Windtunnel: A High-Performance Virtual Reality Application. In: *VR '93: Proceedings of the IEEE virtual reality annual international symposium*, 1993, S. 22–25.

**Chin 2002**

CHIN, Robert: Three-dimensional file system browser. In: *Crossroads* 9 (2002), Nr. 1, S. 16–18.

**Conner et al. 1992**

CONNER, Brookshire D. ; SNIBBE, Scott S. ; HERNDON, Kenneth P. ; ROBBINS, Daniel C. ; ZELEZNIK, Robert C. ; DAM, Andries van: Three-Dimensional Widgets. In: *SI3D '92: Proceedings of the 1992 symposium on Interactive 3D graphics*. New York, NY, USA : ACM Press, 1992, S. 183–188.

**Conrad 2005**

CONRAD, Stefan: System Control and Hybrid 3D/2D Manipulation in Virtual Environments Using Tracked Handheld Computers. In: *Tagungsband Zweiter Workshop Virtuelle und Erweiterte Realität der GI-Fachgruppe VR/AR*. Aachen, 2005.

**CONTIGRA Widgets**

*3D-Widget Klassifikation, Projekt CONTIGRA*. <http://www.3d-components.org>. – [Online, Version: 23. Juli 2007].

**Cruz-Neira et al. 1993**

CRUZ-NEIRA, Carolina ; SANDIN, Daniel J. ; DEFANTI, Thomas A.: Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE. In: *SIGGRAPH '93: Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*. New York, NY, USA : ACM Press, 1993, S. 135–142.

**Dachselt u. Ebert 2001**

DACHSELT, Raimund ; EBERT, Jürgen: Collapsible Cylindrical Trees: A Fast Hierarchical Navigation Technique. In: *INFOVIS '01: Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS'01)*. Washington, DC, USA : IEEE Computer Society, 2001, S. 79.

**Dachselt u. Hinz 2005**

DACHSELT, Raimund ; HINZ, Michael: Three-Dimensional Widgets Revisited - Towards Future Standardization. In: *New directions in 3D user interfaces* (2005), S. 89–92.

**Dachselt u. Hübner 2007**

DACHSELT, Raimund ; HÜBNER, Anett: Virtual Environments: Three-dimensional menus: A survey and taxonomy. In: *Computers and Graphics* 31 (2007), Nr. 1, S. 53–65.

**Darken u. Durost 2005**

DARKEN, Rudolph P. ; DUROST, Richard: Mixed-dimension interaction in virtual environments. In: *VRST '05: Proceedings of the ACM symposium on Virtual reality software and technology*. New York, NY, USA : ACM Press, 2005, S. 38–45.

**Eckel u. Jones 2000**

ECKEL, George ; JONES, Ken: *OpenGL Performer™ Programmer's Guide*. Silicon Graphics, Inc., 2000.

**Fakespace**

*Stereo-Projektionssysteme von Fakespace Systems*. <http://fakespace.com/displays.htm>. – [Online, Version: 14. Juli 2007].

**Fitts 1954**

FITTS, P. M.: The Information Capacity of the Human Motor System in Controlling Amplitude of Movement. In: *Journal of Experimental Psychology* 47 (1954), Nr. 6, S. 381–391. – Reprinted in *Journal of Experimental Psychology: General*, 121(3):262–269, 1992.

**Fröhlich et al. 2005**

FRÖHLICH, Bernd ; BLACH, Roland ; STEFANI, Oliver ; HOCHSTRATE, Jan ; HOFFMANN, Jörg ; KLÜGER, Karsten ; BUES, Matthias: Implementing Multi-Viewer Stereo Displays. In: *WSCG (Full Papers)*, 2005, S. 139–146.

**Gerber u. Bechmann 2004**

GERBER, Dominique ; BECHMANN, Dominique: Design and evaluation of the ring menu in virtual environments. In: *IPT2004: Immersive Projection Technology*. Ames, IA, USA, 2004.

**Gerber u. Bechmann 2005**

GERBER, Dominique ; BECHMANN, Dominique: The Spin Menu: A Menu System for Virtual Environments. In: *VR '05: Proceedings of the 2005 IEEE Conference 2005 on Virtual Reality*. Washington, DC, USA : IEEE Computer Society, 2005, S. 271–272.

**Grosjean et al. 2002**

GROSJEAN, Jérôme ; BURKHARDT, Jean-Marie ; COQUILLART, Sabine ; RICHARD, Paul: Evaluation of the Command and Control Cube. In: *ICMI '02: Proceedings of the 4th IEEE International Conference on Multimodal Interfaces*. Washington, DC, USA : IEEE Computer Society, 2002, S. 473.

**Grosjean u. Coquillart 2001**

GROSJEAN, Jérôme ; COQUILLART, Sabine: Command & Control Cube: a Shortcut Paradigm for Virtual Environments. In: *7th EG Workshop on Virtual Environments & 5th Immersive Projection Technology Workshop (IPT-EGVE01)*, 2001.

**Grossman u. Balakrishnan 2004**

GROSSMAN, Tovi ; BALAKRISHNAN, Ravin: Pointing at trivariate targets in 3D environments. In: *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA : ACM Press, 2004, S. 447–454.

**Häfner et al. 1999**

HÄFNER, Ulrich ; SIMON, Andreas ; DOULIS, Mario: Evaluation of Complex CAD Data in a Virtual Environment. In: *3. International Immersive Projection Technology Workshop*, Springer Verlag Berlin, 1999, S. 157–165.

**Häfner et al. 2000**

HÄFNER, Ulrich ; SIMON, Andreas ; DOULIS, Mario: Unencumbered Interaction in Display Environments with Extended Working Volume. In: *Proceedings of SPIE: Stereoscopic Displays and Virtual Reality Systems* Bd. 3957, SPIE, 2000, S. 473–480.

**Hand 1997**

HAND, Chris: A Survey of 3D Interaction Techniques. In: *Computer Graphics Forum* Bd. 16, 1997, S. 269–281.

**Herndon et al. 1994**

HERNDON, Kenneth P. ; DAM, Andries van ; GLEICHER, Michael: Workshop on the Challenges of 3D Interaction. In: *SIGCHI Bulletin* 26 (1994), Nr. 4.

**Holtkämper et al. 2007**

HOLTKÄMPER, Thorsten ; SCHOLZ, Sascha ; DRESSLER, Armin ; BOGEN, Manfred A.: Co-located Collaborative Use of Virtual Environments. In: *2007 AAPG Annual Convention and Exhibition (April 1 - 4, 2007) Technical Program*. Long Beach, CA USA, 2007.

**Infitec**

INFITEC GmbH, Ulm. <http://infitec.net>. – [Online, Version: 14. Juli 2007].

**Inside Reality**

Öl- und Gas-Software von Schlumberger Ltd., New York. <http://www.slb.com>. – [Online, Version: 14. Juli 2007].

**InterSense**

InterSense Inc. <http://www.isense.com>. – [Online, Version: 17. Juli 2007].

**Inventor**

SGI Open Inventor Homepage. <http://oss.sgi.com/projects/inventor/>. – [Online, Version: 9. September 2007].

**Jacoby u. Ellis 1992**

JACOBY, Richard H. ; ELLIS, Stephen R.: Using virtual menus in a virtual environment. In: *Proceedings of SPIE, Visual Data Interpretation* Bd. 1668. San Jose, CA, USA : SPIE, 1992, S. 39–48.

**Kim et al. 2000**

KIM, Namgyu ; KIM, Gerard J. ; PARK, Chan-Mo ; LEE, Inseok ; LIM, Sung H.: Multimodal Menu Presentation and Selection in Immersive Virtual Environments. In: *VR '00: Proceedings of the IEEE Virtual Reality 2000 Conference*. Washington, DC, USA : IEEE Computer Society, 2000, S. 281.

**Krüger et al. 1995**

KRÜGER, Wolfgang ; BOHN, Christian-A. ; FRÖHLICH, Bernd ; SCHÜTH, Heinrich ; STRAUSS, Wolfgang ; WESCHE, Gerold: The Responsive Workbench: A Virtual Work Environment. In: *Computer* 28 (1995), Nr. 7, S. 42–48.

**Krüger u. Fröhlich 1994**

KRÜGER, Wolfgang ; FRÖHLICH, Bernd: The Responsive Workbench. In: *IEEE Comput. Graph. Appl.* 14 (1994), Nr. 3, S. 12–15.

**Kurtenbach u. Buxton 1993**

KURTENBACH, Gordon ; BUXTON, William: The limits of expert performance using hierarchic marking menus. In: *CHI '93: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA : ACM Press, 1993, S. 482–487.



**Kurtenbach et al. 1999**

KURTENBACH, Gordon ; FITZMAURICE, George W. ; OWEN, Russell N. ; BAUDEL, Thomas: The Hotbox: Efficient Access to a Large Number of Menu-items. In: *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA : ACM Press, 1999, S. 231–237.

**Leiner et al. 1997**

LEINER, Ulrich ; PREIM, Bernhard ; RESSEL, Stephan: Entwicklung von 3D-Widgets – Überblicksvortrag. In: *Simulation und Animation '97*. Delft, Belgium : SCS Europe, 1997, S. 171–188.

**Lemoine et al. 2003**

LEMOINE, Patrick ; VEXO, Frederic ; THALMANN, Daniel: Interaction Techniques: 3D menus-based paradigm. In: *Proceedings of First Research Workshop on Augmented Virtual Reality (AVIR2003)*. Geneva, Switzerland, 2003.

**LEO**

*dict.leo.org Online Wörterbuch der LEO GmbH*. <http://dict.leo.org>. – [Online, Version: 21. Juli 2007].

**Liang u. Green 1994**

LIANG, Jiandong ; GREEN, Mark: JDCAD: a highly interactive 3D modeling system. In: *Computers & Graphics: Proceedings of the Conference on CAD and Computer Graphics*. Beijing, China, 1994, S. 499–506.

**MacKenzie u. Buxton 1992**

MACKENZIE, I. S. ; BUXTON, William: Extending Fitts' law to two-dimensional tasks. In: *CHI '92: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA : ACM Press, 1992, S. 219–226.

**Mine 1995**

MINE, Mark R.: Virtual Environment Interaction Techniques / University of North Carolina at Chapel Hill. Chapel Hill, NC, USA : University of North Carolina at Chapel Hill, 1995 (TR95-018).

**Möller u. Haines 1999**

MÖLLER, Tomas ; HAINES, Eric: *Real-Time Rendering*. A. K. Peters Limited, 1999.

**NASA VWT**

*NASA Virtual Windtunnel Widgets*. <http://www.nas.nasa.gov/Software/VWT/widgets.html>. – [Online, Version: 02. August 2007].

**Performer**

*SGI OpenGL/Performer Homepage*. <http://www.sgi.com/products/software/performer/>. – [Online, Version: 9. September 2007].

**Perlin 1998**

PERLIN, Ken: Quikwriting: continuous stylus-based text entry. In: *UIST '98: Proceedings of the 11th annual ACM symposium on User interface software and technology*. New York, NY, USA : ACM Press, 1998, S. 215–216.

**Polhemus Fastrack**

*Elektromagnetisches Tracking-System von Polhemus*. [http://polhemus.com/?page=Motion\\_Fastrak](http://polhemus.com/?page=Motion_Fastrak). – [Online, Version: 16. Juli 2007].

**Poupyrev et al. 1996**

POUPYREV, Ivan ; BILLINGHURST, Mark ; WEGHORST, Suzanne ; ICHIKAWA, Tadao: The Go-Go Interaction Technique: Non-Linear Mapping for Direct Manipulation in VR. In: *UIST '96*:

*Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA : ACM Press, 1996, S. 79–80.

**Preim et al. 1997**

PREIM, Bernhard ; RAAB, Andreas ; STROTHOTTE, Thomas: Coherent zooming of illustrations with 3D-graphics and text. In: *Proceedings of the conference on Graphics interface '97*. Toronto, Ont., Canada, Canada : Canadian Information Processing Society, 1997, S. 105–113.

**Projekt CONTIGRA**

*Projekt CONTIGRA - COmponent-orieNted Three-dimensional Interactive GRaphical Applications*. <http://www.contigra.de>. – [Online, Version: 21. Juli 2007].

**Quikwriting**

*Quikwriting Texteingabemethode*. <http://mrl.nyu.edu/~perlin/experiments/quikwriting/>. – [Online, Version: 30. August 2007].

**Riege et al. 2006**

RIEGE, Kai ; HOLTKÄMPER, Thorsten ; WESCHE, Gerold ; FRÖHLICH, Bernd: The Bent Pick Ray: An Extended Pointing Technique for Multi-User Interaction. In: *3DUI '06: Proceedings of the 3D User Interfaces*. Washington, DC, USA : IEEE Computer Society, 2006, S. 62–65.

**Robertson et al. 1997**

ROBERTSON, George ; CZERWINSKI, Mary ; DANTZICH, Maarten van: Immersion in desktop virtual reality. In: *UIST '97: Proceedings of the 10th annual ACM symposium on User interface software and technology*. New York, NY, USA : ACM Press, 1997, S. 11–19.

**Sherman u. Craig 2003**

SHERMAN, William R. ; CRAIG, Alan B.: *Understanding Virtual Reality*. San Francisco, CA, USA : Elsevier Science, 2003.

**Simon et al. 2005**

SIMON, Andreas ; DRESSLER, Armin ; KRÜGER, Hans-Peter ; SCHOLZ, Sascha ; WIND, Jürgen: Interaction and Co-located Collaboration in Large Projection-Based Virtual Environments. In: *INTERACT*, 2005, S. 364–376.

**Simon u. Göbel 2002**

SIMON, Andreas ; GÖBEL, Martin: The i-Cone – A Panoramic Display System for Virtual Environments. In: *PG '02: Proceedings of the 10th Pacific Conference on Computer Graphics and Applications*. Washington, DC, USA : IEEE Computer Society, 2002, S. 3.

**Simon u. Scholz 2005**

SIMON, Andreas ; SCHOLZ, Sascha: Multi-Viewpoint Images for Multi-User Interaction. In: *VR '05: Proceedings of the 2005 IEEE Conference 2005 on Virtual Reality*. Washington DC, USA : IEEE Computer Society, 2005, S. 107–113.

**Simon et al. 2004**

SIMON, Andreas ; SMITH, Randall C. ; PAWLICKI, Richard R.: OmniStereo for Panoramic Virtual Environment Display Systems. In: *VR '04: Proceedings of the IEEE Virtual Reality 2004*. Washington, DC, USA : IEEE Computer Society, 2004, S. 67.

**Springer et al. 2007a**

SPRINGER, Jan P. ; BECK, Stephan ; WEISZIG, Felix ; FRÖHLICH, Bernd: Multi-Frame Rate Rendering for Standalone Graphics Systems. In: *Proceedings 4. Workshop „Virtuelle und Erweiterte Realität“ der GI-Fachgruppe VR/AR*, Gesellschaft für Informatik, 2007, S. 55–64.

**Springer et al. 2007b**

SPRINGER, Jan P. ; BECK, Stephan ; WEISZIG, Felix ; REINERS, Dirk ; FRÖHLICH, Bernd: Multi-Frame Rate Rendering and Display. In: *Proceedings IEEE Virtual Reality 2007 Conference*, IEEE Computer Society Press, 2007, S. 195–202.

**Tramberend 1999**

TRAMBEREND, Henrik: Avocado: A Distributed Virtual Reality Framework. In: *IEEE Virtual Reality 1999 Conference (VR'99)*, 1999, S. 14–21.

**Tramberend 2003**

TRAMBEREND, Henrik: *Avocado: A Distributed Virtual Environment Framework*, Universität Bielefeld, Diss., 2003.

**VRGeo**

VRGeo Konsortium (*Virtual Reality for the Geosciences*). <http://vrgeo.org>. – [Online, Version: 14. Juli 2007].

**Wang et al. 2005**

WANG, Shuo ; POTURALSKI, Marcin ; VRONAY, David: Designing a generalized 3D carousel view. In: *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*. New York, NY, USA : ACM Press, 2005, S. 2017–2020.

**Wesche 2004a**

WESCHE, Gerold: *Conceptual Free-Form Styling in Virtual Environments*, Universität des Saarlandes, Diss., 2004).

**Wesche 2004b**

WESCHE, Gerold: A User Interface for Free-Form Shape Design at the Responsive Workbench. In: *Journal of Computing and Information Science in Engineering (Transactions of the ASME)* 4 (2004), Nr. 3, S. 178–185.

**Wloka u. Greenfield 1995**

WLOKA, Matthias M. ; GREENFIELD, Eliot: The virtual tricorder: a uniform interface for virtual reality. In: *UIST '95: Proceedings of the 8th annual ACM symposium on User interface and software technology*. New York, NY, USA : ACM Press, 1995, S. 39–40.

**Zhai 1998**

ZHAI, Shumin: User performance in relation to 3D input device design. In: *SIGGRAPH Comput. Graph.* 32 (1998), Nr. 4, S. 50–54.

**Zhai et al. 1996**

ZHAI, Shumin ; MILGRAM, Paul ; BUXTON, William: The influence of muscle groups on performance of multiple degree-of-freedom input. In: *CHI '96: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA : ACM Press, 1996, S. 308–315.



# **A Anhang**



## Fragebogen zur Benutzerstudie

Vielen Dank für die Geduld beim praktischen Teil dieses Usertests. Im Anschluss habe ich nun noch einige Fragen zu Deinen Eindrücken.

- 1) **Wie empfandest Du folgende Aspekte der verschiedenen Arten von Schiebereglern?**  
**Bitte vergleiche und bewerte die Schwierigkeitsgrade der drei Phasen „Regler greifen“, „Reglerwert einstellen“ und „Zielwert erreichen“.**  
**Kreuze an, welche Aussage am ehesten auf Dich zutrifft:**

• **Das Greifen des Sliders nach dem Aktivieren des Panels empfand ich als...**

	sehr schwierig	schwierig	weder noch	einfach	sehr einfach
Block (nah)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Block (fern)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
WIMP (nah)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
WIMP (fern)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
PDA	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Tablet-PC	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

• **Das Verändern des Wertes empfand ich als...**

	sehr schwierig	schwierig	weder noch	einfach	sehr einfach
Block (nah)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Block (fern)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
WIMP (nah)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
WIMP (fern)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
PDA	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Tablet-PC	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

• **Das Erreichen des Zielwertes empfand ich als...**

	sehr schwierig	schwierig	weder noch	einfach	sehr einfach
Block (nah)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Block (fern)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
WIMP (nah)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
WIMP (fern)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
PDA	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Tablet-PC	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

2) **Vergleiche die 3D-Regler miteinander.**

**Vergleiche die Optionen und kreuze an, wie sehr Du zu einer der jeweiligen Aussage tendierst:**

• **Als Entfernung des Reglers bevorzuge ich die Darstellung...**

weit entfernt in der Szene

nah am Körper

--------------------

• **Bei den weit entfernt dargestellten Reglern bevorzuge ich...**

das Greifen des Anfassers

das Greifen des kompletten Regler-Blocks

--------------------

das Schieben des Anfassers mit dem Strahl

die Werteingabe mittels Handrotation

--------------------

• **Bei den nah dargestellten Reglern bevorzuge ich...**

das Greifen des Anfassers

das Greifen des kompletten Regler-Blocks

--------------------

das Schieben des Anfassers mit dem Strahl

die Werteingabe mittels Handrotation

--------------------

3) **Vergleiche die 3D-Regler mit Tablet-PC und PDA**

**Kreuze an, wie sehr Du eine der angegebenen Optionen bevorzugst.**

Regler auf Tablet-PC

3D-Regler in Virtual Reality

--------------------

Regler auf PDA

3D-Regler in Virtual Reality

--------------------

Regler auf Tablet-PC

Regler auf PDA

--------------------



User Nr.:

Name:

Tablet:  Pult  Arm

---

4) Bitte bewerte das jeweilige Interaktions-„Gesamtpaket“ mit Schulnoten von 1 bis 5.  
Eine 1 bedeutet „sehr gut“, eine 5 „mangelhaft“

\_\_\_\_\_ PDA

\_\_\_\_\_ Tablet-PC

\_\_\_\_\_ 3D-Block-Regler mit Werteingabe mittels Handrotation

\_\_\_\_\_ 3D-Regler mit Anfasser und Werteingabe über Strahlinteraktion

---

5) Zum Abschluss benötige ich noch Angaben zu Deiner Erfahrung:

• *Ich benutze Virtual Reality...*

regelmäßig

ab und zu

sehr selten / bisher noch nie

• *Ich benutze Geräte mit Touchscreen und Stifteingabe wie PDA oder Tablet-PC...*

regelmäßig

ab und zu

sehr selten / bisher noch nie

---

6) Hier ist Platz für weitere Anmerkungen oder Anregungen: Hattest Du Probleme mit den genutzten Displays, mit den Geräten, der 3D-Brille oder mit der Interaktion? Hast Du Vorschläge oder Ideen? Was fandest Du besonders gut, schlecht oder bemerkenswert? War die Trainingszeit ausreichend?

Vielen Dank für die Teilnahme am Usertest!



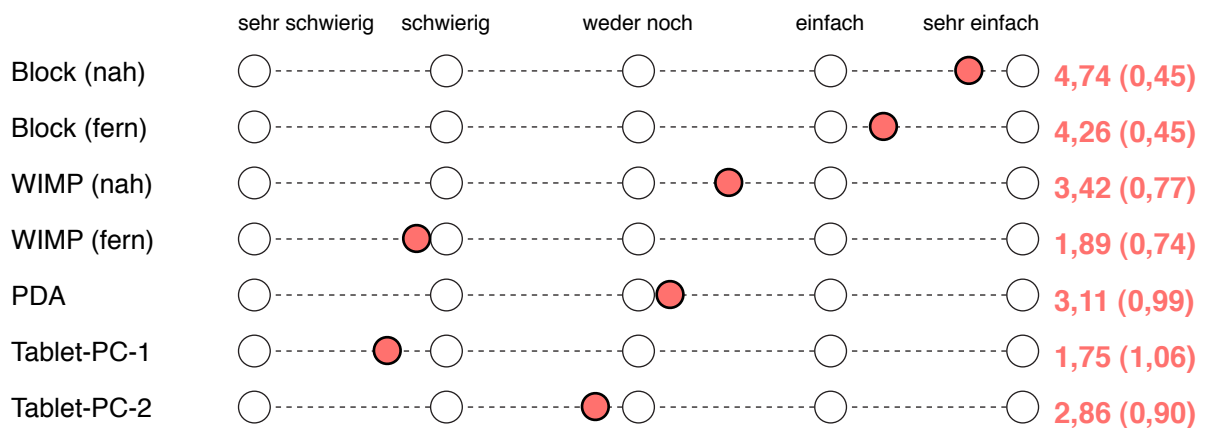
Block, WIMP, PDA: N = 19  
 Tablet-PC-1: N = 12  
 Tablet-PC-2: N = 7

Tablet: ○ Pult 9x ○ Arm 10x

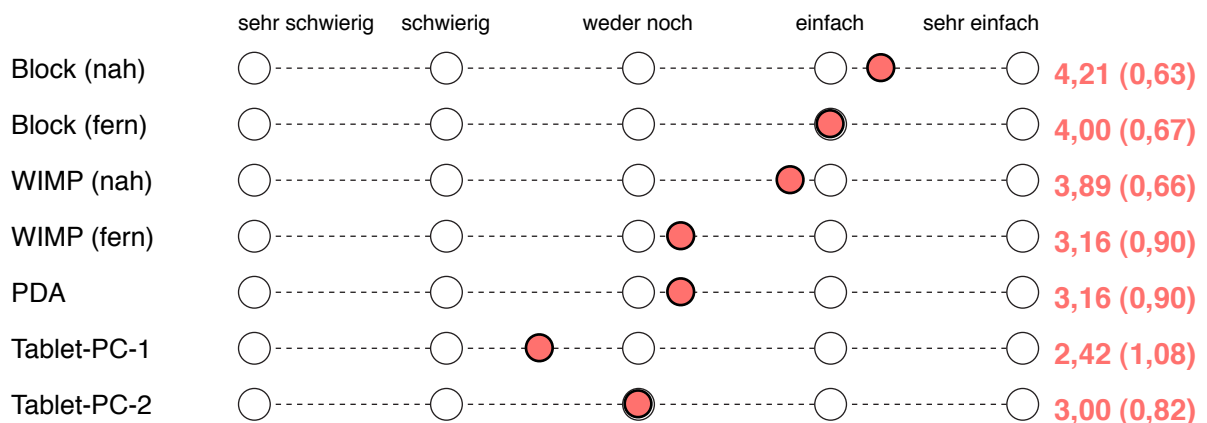
## Fragebogen zur Benutzerstudie (Mittelwerte)

Die roten Zahlen und Kreise geben die Mittelwerte aller Antworten an. In Klammern dahinter steht die Standardabweichung. Die Tablet-PC Fragen sind jeweils auf Tablet-PC-1 und Tablet-PC-2 aufgeteilt.

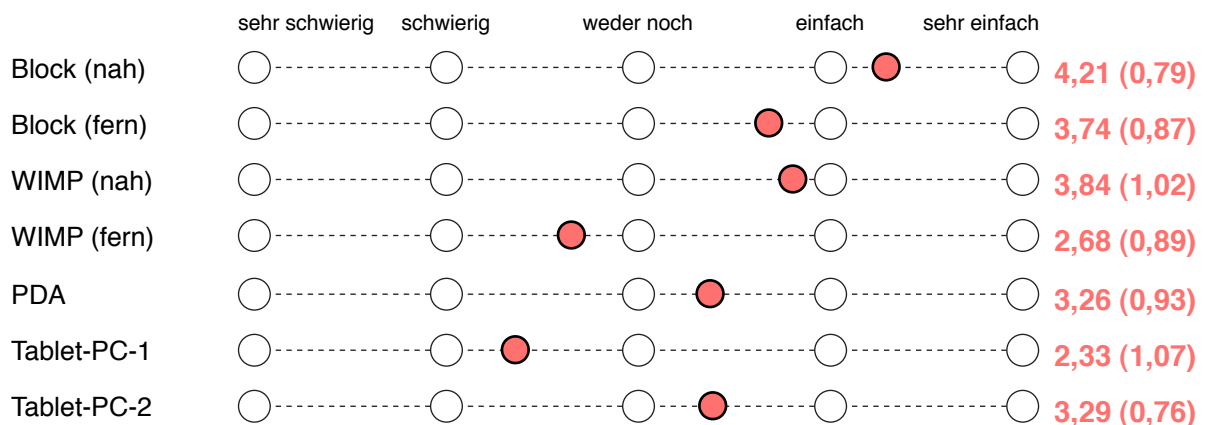
### • Das Greifen des Sliders nach dem Aktivieren des Panels empfand ich als...



### • Das Verändern des Wertes empfand ich als...



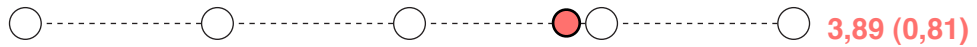
### • Das Erreichen des Zielwertes empfand ich als...



• Als Entfernung des Reglers bevorzuge ich die Darstellung...

weit entfernt in der Szene

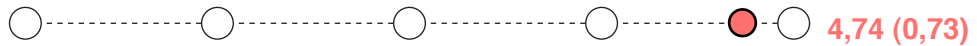
nah am Körper



• Bei den weit entfernt dargestellten Reglern bevorzuge ich...

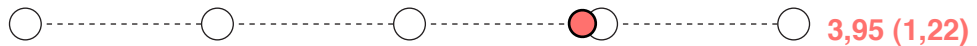
das Greifen des Anfassers

das Greifen des kompletten Regler-Blocks



das Schieben des Anfassers mit dem Strahl

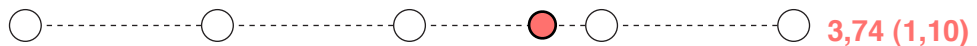
die Werteingabe mittels Handrotation



• Bei den nah dargestellten Reglern bevorzuge ich...

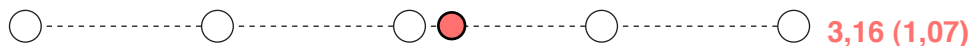
das Greifen des Anfassers

das Greifen des kompletten Regler-Blocks



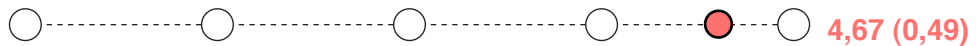
das Schieben des Anfassers mit dem Strahl

die Werteingabe mittels Handrotation



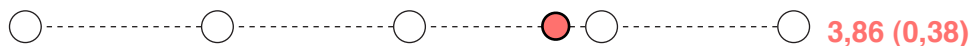
Regler auf Tablet-PC-1

3D-Regler in Virtual Reality



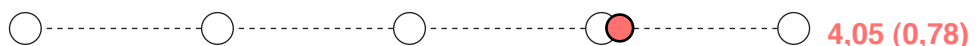
Regler auf Tablet-PC-2

3D-Regler in Virtual Reality



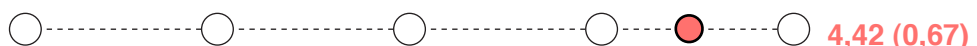
Regler auf PDA

3D-Regler in Virtual Reality



Regler auf Tablet-PC-1

Regler auf PDA



Regler auf Tablet-PC-2

Regler auf PDA



Bitte bewerte das jeweilige Interaktions-„Gesamtpaket“ mit Schulnoten von 1 bis 5.  
Eine 1 bedeutet „sehr gut“, eine 5 „mangelhaft“

3,16 (0,83) PDA

4,50 (0,67) Tablet-PC-1

3,29 (0,76) Tablet-PC-2

1,47 (0,51) 3D-Block-Regler mit Werteingabe mittels Handrotation

2,45 (0,76) 3D-Regler mit Anfasser und Werteingabe über Strahlinteraktion

---

Zum Abschluss benötige ich noch Angaben zu Deiner Erfahrung:

• *Ich benutze Virtual Reality...*

- regelmäßig 10x
- ab und zu 4x
- sehr selten / bisher noch nie 5x

• *Ich benutze Geräte mit Touchscreen und Stifteingabe wie PDA oder Tablet-PC...*

- regelmäßig 7x
- ab und zu 7x
- sehr selten / bisher noch nie 5x

---

6) Hier ist Platz für weitere Anmerkungen oder Anregungen: Hattest Du Probleme mit den genutzten Displays, mit den Geräten, der 3D-Brille oder mit der Interaktion? Hast Du Vorschläge oder Ideen? Was fandest Du besonders gut, schlecht oder bemerkenswert? War die Trainingszeit ausreichend?

Eine Auflistung der genannten Anmerkungen folgt auf den nächsten Seiten

Vielen Dank für die Teilnahme am Usertest!



---

## Benutzerstudie: Beobachtungen, Kommentare und Vorschläge

Am Ende des Fragebogens durften die Probanden Kritik und Vorschläge äußern. Hier ist eine Zusammenfassung (Anzahl der Erwähnungen in Klammern). Auf der nächsten Seite folgen Notizen zur Benutzerstudie.

### Kritik

- Der Blickwechsel zwischen VR-Display und PDA/Tablet-PC ist störend (7x)
- Die 3D-Brille ist schwer, klobig und rutscht, wenn man nach unten auf ein Geräte-Display guckt (2x)
- Durch die Shutterbrille auf ein Geräte-Display zu gucken ist unangenehm
- Die fehlende Handflächenerkennung des Tablet-PC wurde bemängelt (2x)
- Tablet-PC Touchscreen zu unempfindlich/schlecht zu bedienen (2x)
- Tablet-PC unhandlich
- Text in 3D schwierig zu lesen und führt zu Augenermüdung
- Strahlinteraktion bei WIMP-fern erfordert sehr hohe Konzentration und eine ruhige Hand
- Block-fern: Wenn ein Wert verfehlt wurde, ist der Pickray oft weit vom Block entfernt, so dass für ein Neugreifen eine Handbewegung nötig ist
- Usertest ist zu speziell
- Im Vergleich zur Stifteingabe fehlt in VR das haptische Feedback
- Block-nah und WIMP-nah war etwas zu nah am Benutzer
- Stylus zu groß: verdeckt Sicht auf Tablet-Display
- Steuerung aus der Hand heraus zwar bequem, jedoch Armhaltung auf Dauer unkomfortabel (Arm unbewegt, nur Handinteraktion)

### Vorschläge

- Es wäre hilfreich, die Regler- und Zielwerte auch auf dem PDA- und Tablet-Display anzuzeigen (3x)
- Optimierung der Slider-Darstellung/Interaktion auf dem Tablet-PC nötig, z.B. größerer Regler (2x)
- Rasterung der Slider wäre hilfreich
- Darstellung einer Skala am Slider wäre hilfreich
- Shortcut für kleine Wertänderung wäre hilfreich, wenn man den Zielwert nur um ein oder zwei Einheiten verfehlt hatte (vor allem auf dem Tablet, da hier das Greifen und Einstellen besonders schwierig war)
- Größere Anfasser für die WIMP-Slider
- Interaktion mittels Handrotation mit Beschleunigung: Größere Werte schneller ändern, dafür genauer bei kleinen Bewegungen
- Reglerwert mit einem Scrollrad am Input-Device verändern
- Slider vertikal darstellen anstelle horizontal
- Eingabe per Stift ist angenehmer, allerdings stört der Blickwechsel auf Tablet/PDA
- akustisches Feedback bei Erreichen des Zielwertes

## Notizen

Während der Usertest-Durchführung wurden Notizen zu den Gesprächen gemacht, aber auch zu Dingen, die während der Tests aufgefallen sind:

- Touchscreen vom Tablet verliert immer wieder den Slider, außerdem muss der User die Hand auflegen dürfen
- Die Brille schränkt das Gesichtsfeld stark ein, daher ist ein Blickwechsel zwischen VR-Display und PDA/Tablet nur mit Kopfbewegung möglich. Ein kurzes Hinab- oder Aufschauen ist daher sehr aufwändig
- Brille rutscht beim Blick nach unten
- Tablet-PC wird von vielen Usern freihändig bedient, also ohne die Hand oder den Arm auf der Tablet-Kante abzustützen. Dies geschieht auch, wenn nicht explizit erwähnt wurde, dass man die Hand nicht auflegen darf
- Der Anfasser des Tablet-Sliders ist viel zu klein. Mit dem klobigen Stylus in der Hand ist die Bedienung extrem schwierig.
- Viele User haben sich kaum Zeit für das Training genommen. Demnach ist die Interaktion sehr intuitiv. Dies kann aber auch ein Hinweis darauf sein, dass die Aufgabe unangenehm oder ermüdend ist. Besonders kurz fällt die Trainingszeit bei den kompliziertesten Aufgaben aus: Tablet-PC und WIMP-fern
- WLAN-Performance problematisch. Immer wieder hakt der Wert, besonders beim PDA. Aber auch beim Tablet wird der Wert oft im letzten Moment noch verrissen, da eine kurze Verzögerung der Werteingabe vorliegt
- Einzelne Nutzer hatten schon nach wenigen Minuten Probleme mit ihren Augen. Der vorgegebene Standpunkt vor dem Screen ist offenbar zu nah gewählt, denn viele Nutzer sind zwischendurch etwa einen halben Schritt zurückgewichen (ca 20-30cm).
- Die kurze Verzögerung, bis der Button-Click in Avango registriert wird, sorgt dafür, dass viele User zu schnell nach dem vermeintlichen Greifen des Sliders diesen schon verschieben wollen, ohne den Slider tatsächlich gegriffen zu haben.
- Der PDA wird praktisch nicht bewegt, sondern verharrt den kompletten Test lang in gleicher Lage. So kann ohne Bewegung für die nächste Runde das Context-Control aufgerufen werden.
- Verständnis-Problem: Zwei Nutzer musste ich ausdrücklich darauf hinweisen, dass man den Block-Slider nicht an der Grenze zwischen der blauen und weißen Fläche des Balkens greifen muss
- WIMP-fern: Einige Nutzer stützen ihre Interaktionshand mit der anderen Hand, um genauer zielen zu können. Besonders wenn WIMP-fern erst zum Ende des Usertests getestet wurde, war die Handermüdung wohl so hoch, dass die Hand zittriger wurde.
- Ein User fand das Verschwinden des Strahls während der Block-Interaktion irritierend



# Erklärung

Hiermit erkläre ich, dass ich diese Arbeit selbständig durchgeführt und abgefasst habe. Quellen, Literatur und Hilfsmittel, die von mir benutzt wurden, sind als solche gekennzeichnet.

Armin Dressler

Ilmenau, den 21. Oktober 2007



# Geheimhaltungsvereinbarung

Geheimhaltungsvereinbarung Platzhalter 1



# Benutzergerechte Menügestaltung für Virtuelle Umgebungen im Expertenkontext

Diplomarbeit von Armin Dressler

## Forschungsfrage

- Wie kann ein 3D-Menü für Virtual-Reality-Anwendungen implementiert werden, damit es im gegebenen Anwendungskontext alle Anforderungen erfüllt?

## Subfragen

- Wie ist der Anwendungskontext definiert? Welche Anforderungen gibt es?
- Welche Lösungen existieren bereits? Wie können diese zur Lösung beitragen?
- Wie kann ein Menü gestaltet werden, damit es intuitiv bedient werden kann?
- Welche Virtual-Reality-spezifischen Herausforderungen gilt es zu beachten?
- Welche Lösungsmöglichkeiten für die Probleme gibt es?

## Thesen

- Durch ein für Virtuelle Umgebungen optimiertes Menü-Layout lässt sich die Effizienz der Menübedienung gegenüber anderen Menüformen verbessern und die Fehlerrate senken. Durch eine Block-artige Listenstruktur lassen sich die Menüeinträge per Strahlinteraktion zuverlässig und schnell aktivieren.
- Das „Context-Control“-Konzept beschreibt die Zusammenfassung von Menüs zur Systemsteuerung, Kontextmenüs und Control-Paletten zur Parameteränderung in einer einzigen Menüform. Durch Vereinheitlichung von Darstellung und Interaktion aller Menüelemente ist die Bedienung intuitiv und auch für Laien leicht verständlich.
- Verdeckungsprobleme durch störende 3D-Objekte lassen sich durch eine regelbasierte Positionierung des Menüs vermeiden. Ein halbtransparenter Menühintergrund verhindert außerdem die vollständige Verdeckung der 3D-Szene durch das Menü.
- Eine situationsabhängige Skalierung des Menüs sichert eine gleichbleibende Interaktionsqualität auf unterschiedliche Distanzen. Die Skalierung beachtet außerdem die Darstellungsqualität der Menübeschriftung und gewährleistet die Lesbarkeit.



# Verwertungsrechte

Verwertungsrechte Platzhalter 1





Verwertungsrechte Platzhalter 2



# CD-Rom mit Originalmaterial

Auf der CD-Rom befinden sich folgende Daten:

- Diplomarbeit
- Fragebogen
- Originalmaterialien
- Quellen