

HAPTISCHE 3D-BENUTZUNGSOBERFLÄCHE

von

Rolf Theisinger

Matrikel Nr. 1227358

Diplomarbeit entsprechend der
Diplom-Prüfungsordnung des
Fachbereichs Biologie und Informatik
zur Erlangung des 'Titels

Diplom-Informatiker



Johann Wolfgang Goethe-Universität
Frankfurt am Main

2002

Prüfer: Prof. Dr.-Ing. Detlev Krömker
Betreuer: Dipl.-Inform. Christian Seiler
Dipl.-Designer (FH) Wolfgang Baier

Erklärung

Ich bestätige, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen Quellen und Hilfsmittel als die angegebenen benutzt habe.

Frankfurt am Main, den 2. September 2002

Rolf Theisinger

INHALTSVERZEICHNIS

1	Einleitung	1
1.1	Motivation	1
1.2	Gliederung	2
2	Grundlagen	4
2.1	<i>Virtuelle Realität</i>	5
2.1.1	Immersion	5
2.1.1.1	Räumliche Wahrnehmung	7
2.1.1.2	Sprache	8
2.2	<i>Der Mensch hat zwei Hände</i>	9
2.2.1	Ist der Einsatz der Hände sinnvoll?	10
2.2.2	Geste und Postur	11
2.2.3	Klassifizierung der Hand-Aktionen	12
2.2.4	Bezugsrahmen und kinematische Ketten	15
2.2.5	Input-Performanz	17
2.3	<i>Haptik</i>	20
2.3.1	Passive Haptische Wahrnehmung	21
2.3.2	Oberflächenbeschaffenheit	22
2.3.3	Aktive Haptische Wahrnehmung	22
2.3.4	Sehen mit den Händen	23
2.3.5	Psychomotorische Prozesse	23
2.3.6	Interaktion der Hände mit Objekten	24
2.4	<i>Steuerung von 3D-Simulationen</i>	26
2.4.1	3D-Interaktion	28
2.4.2	Psychologie der virtuellen Realität	29
2.4.2.1	Navigation	29
2.4.2.2	Merkmale virtueller Objekte	30
2.4.2.3	3D-Interaktionen	32
2.4.2.4	Ein- und Ausgabegeräte	32
2.4.2.5	Benutzungsoberflächen, Design und der Benutzer	33
2.4.2.6	Das konzeptionelle Modell	34
2.4.2.7	Mapping	36
2.4.2.8	Feedback	37
2.4.2.9	Human Action Cycle	38
2.4.2.10	Beschränken der möglichen Alternativen: Constraints	40
2.5	<i>Lernen aus der realen Welt</i>	42
2.5.1	Greifarten	42
2.5.2	Stellteile	45
3	Stand der Technik	50
3.1	<i>2D-Benutzungsschnittstellen</i>	51
3.1.1	WIMP: Windows, Icons, Mouse, Pointer	51
3.1.2	3D-Anwendungen im 2D-Arbeitsbereich	56
3.2	<i>3D-Benutzungsschnittstellen</i>	58

3.2.1	Maus-basierte 6DOF-Eingabegeräte	58
3.2.1.1	Cubic Mouse	58
3.2.1.2	Magellan/Logicad Spacemouse	60
3.2.2	Geräte mit haptischem Feedback	62
3.2.2.1	Logitech Driving Force	62
3.2.2.2	Phantom	64
3.2.2.3	Immersion CyberGrasp	65
3.2.3	Haptische Graphische Benutzungsschnittstellen	67
3.2.3.1	e-Touch	67
3.2.3.2	Studierstube	69
3.2.4	Real-World Benutzungsschnittstellen	72
3.2.4.1	Texas Instruments TI-92 Plus	72
3.2.4.2	Barthel Chemcode	73
3.2.4.3	BOSS BR-532 4-Kanal Aufnahmegerät	75
3.2.4.4	Canon Powershot S40	76
4	Entwicklung einer Haptischen 3D Benutzungsoberfläche	79
4.1	<i>Vorraussetzungen für die natürliche Mensch-Maschine Interaktion</i>	80
4.1.1	Möglichkeiten von virtuellen Benutzungsschnittstellen	83
4.2	<i>Konfiguration der Projekthardware</i>	84
4.2.1	Funktionsweise und Einschränkungen der Fremd-Bibliotheken und der Hardware	86
4.2.1.1	Die visuelle und haptische Repräsentation virtueller Objekte	87
4.2.1.2	Collision Detection	90
4.2.1.3	Die virtuelle Hand	90
4.2.1.4	Dokumentation	91
4.2.1.5	Die Szenegraph API: Cosmo	91
4.3	<i>Der Ansatz</i>	92
4.4	<i>EASY: a graspable GUI</i>	94
4.4.1	Das Konzept	94
4.4.1.1	Der Plan	95
4.4.1.1.1	Multithreaded Rendering	95
4.4.1.1.2	Berechnung der Kameraposition	95
4.4.1.1.3	Einfügen von Objekten in die Szene	96
4.4.1.1.4	Kollisionsberechnung	96
4.4.1.1.5	Einbinden der Hardware	97
4.4.1.1.6	Einbinden der haptischen Komponenten	97
4.4.1.1.7	Die Hände und die Werkzeuge	97
4.4.2	Implementation	97
4.4.2.1	Die Haptische Komponente	98
4.4.2.1.1	HapticComponentBase	98
4.4.2.1.2	HapticComponent	100
4.4.2.2	ComponentManager	103
4.4.2.2.1	Registrieren der Komponenten	103
4.4.2.3	Simulations-Schleife	103
4.4.2.4	Haptisches Feedback	104
4.4.2.5	Programmablauf	105
4.5	<i>Der Einsatz von EASY</i>	108
4.5.1	Wiederverwendbare Benutzungsschnittstellenelemente	109
4.5.1.1	Schalter und Regler	109
4.5.1.1.1	Schalter	110

4.5.1.1.2	Regler	111
4.5.2	Ein Kontrollpanel	111
4.6	<i>Bewertung</i>	<i>114</i>
5	Ausblick	116

DANKSAGUNG

Ich möchte mich bei allen bedanken, die mich während dieser Arbeit mit Rat und Tat unterstützt haben. Besonderer Dank sei meinen Betreuern Christian Seiler und Wolfgang Baier ausgesprochen, welche zu jeder Zeit ein offenes Ohr für mich hatten und mich mit Lob und wohl dosierter Kritik auf Kurs hielten.

Auch Christian Petrak sei gedankt, der sich trotz seines vollen Terminplans auf die Suche nach den letzten Rechtschreibfehlern gemacht hat.

AUFGABENSTELLUNG

HAPTISCHE 3D-BENUTZUNGSOBERFLÄCHE

An der Professur für Graphische Datenverarbeitung wird derzeit ein Projekt mit dem Namen „Virtual Glove Box“ bearbeitet. Eine „Glove Box“ ist ein Apparat, in welchem chemische Versuche in abgeschlossener Atmosphäre durchgeführt werden können. Die „Virtual Glove Box“ setzt dieses Konzept für Virtual Reality Anwendungen um. Der Betrachter schaut von außen auf zum Beispiel computergenerierte Objekte, welche er ähnlich dem realen Vorbild mit den Händen mit Hilfe haptischer Eingabegeräte manipulieren kann (mechanische Exoskelette, die wie Handschuhe an den Händen getragen werden). Um einen realistischen Eindruck zu vermitteln, müssen Aussehen, Verhalten und haptische Eigenschaften der virtuellen Gegenstände geeignet simuliert werden. Hierzu sollen komponentenbasierte Simulationsmethoden eingesetzt werden.

Derzeit existieren für die „Virtual Glove Box“ einzelne Komponenten und Anwendungen, die auf einer Benutzerebene zusammengeführt werden sollen. Dazu bedarf es einer 3D-Benutzungsfläche, die völlig neue Möglichkeiten durch ein haptisches Feedback bietet.

In dieser Arbeit sollen sowohl aktuelle 3D-GUI Konzepte als auch Benutzungselemente der realen menschlichen Umwelt recherchiert und diskutiert werden.

Hiervon ausgehend sollen eigene konzeptionelle Vorschläge für ein haptisches 3D-User-Interface (UI) erstellt werden. Dabei soll auch die Wiederverwendbarkeit und Kombinierbarkeit der konzipierten UI-Elemente beachtet werden. Das entwickelte Konzept soll prototypisch umgesetzt werden.

1 Einleitung

Ziel dieser Arbeit ist die Entwicklung einer flexiblen Benutzungsoberfläche für die Virtual Glove Box, welche wiederverwendbare Komponenten zur Anwendungssteuerung bereitstellt. Besondere Eigenschaften dieser Komponenten sind die Benutzbarkeit unter Anwendung zweier virtueller Hände und die Vermittlung von haptischem Feedback mit Hilfe von an den Händen getragenen Exoskeletten, abgestimmt auf die Anforderungen der Virtual Glove Box. Vorbereitend werden aktuelle 3D-GUI (Graphical User Interface) Konzepte als auch Bedienelemente der realen Welt diskutiert, sowie die für diese Arbeit nötigen dargestellt.

1.1 Motivation

Die Benutzungsoberfläche stellt den Kontakt zwischen Mensch und den Methoden einer Anwendung her. Entsprechend wichtig ist ihr Design, um eine möglichst intuitive Bedienung der Anwendung zu ermöglichen. Allerdings sind dem Designer bisher enge Grenzen gesetzt, bedingt durch Eingabegeräte mit nur wenigen Freiheitsgraden. Dies führt zwangsläufig zu Inkonsistenzen zwischen Gesehenem und Gefühltem, da konventionelle Eingabegeräte keine Möglichkeiten eines dynamischen Feedbacks bieten.

Es ist nötig, mit den beschränkten Möglichkeiten der Eingabegeräte eine Vielzahl von sehr unterschiedlichen, insbesondere abstrakten Vorgängen zu steuern, was in den meisten Fällen zu Verwirrung führt. Besonders bei Menschen, welche nicht im Umgang mit Computern geübt sind, tritt dieser Effekt ein. Es existieren sehr gute Ansätze, um mit Hilfe gut ausgewählter Visualisierungen die Verständlichkeit von Computer-Werkzeugen zu erleichtern. Der Benutzer ist jedoch gezwungen, seine Eingaben in Bewegungen der Maus oder Bedienung der Tastatur umzusetzen.

Die Grenzen dieser inzwischen etwas „angegrauten“ Technik liegen ganz klar im Bereich der 3D-Anwendungen. Möchte man eine Rotation eines Objekts in einer 3D-Umgebung durchführen, ist es für den Designer einer Benutzungsoberfläche nötig, für das jeweilige Eingabegerät eine Interaktions-Metapher zu finden, welche möglichst auf das Wissen und die Fähigkeiten des potentiellen Benutzers zugeschnitten ist. Deren Einsatz erfordert oft das Studium dicker (und oft schlechter) Handbücher.

Eine herkömmliche Maus mit 2 Freiheitsgraden stellt den Entwickler in vielen Fällen vor nur schlecht lösbare Probleme. Das Ziel sollte sein, die Fähigkeiten der Eingabegeräte zu verbessern, um den Umgang mit Computern zu erleichtern. Es können dann bessere Interaktions-Metaphern gefunden werden. Dass diese Forderung nicht sinnlos ist, im Gegenteil die Suche nach guten Schnittstellen einer der Forschungsschwerpunkte der Softwareindustrie ist, zeigen die vielen Projekte aus diesem Bereich.

1.2 Gliederung

Die Steuerung des Computers unter Einsatz zweier virtueller Hände in Verbindung mit Force-Feedback Geräten zur Vermittlung haptischer Informationen ist eine der Grundvoraussetzungen für die Durchführung dieser Arbeit. Die bisher gewonnenen und für diese Arbeit relevanten Erkenntnisse werden in Kapitel 2 beschrieben und sollen gleichzeitig als Anforderungen an die hier entwickelte Benutzungsoberfläche verstanden werden.

In Kapitel 3 beschreibt kurz den Weg von der 2D-Benutzungsoberfläche in die dritte Dimension und zeigt einige Beispiele der aktuell verfügbaren Hardware für 3D-Interaktionen. Ebenso werden aktuelle 3D-Benutzungsoberflächen betrachtet und mit Hilfe der in Kapitel 2 dargelegten Forderungen an eine solche Schnittstelle bewertet. Entsprechend der Aufgabenstellung werden ebenso einige thematisch passende Real-World Benutzungsoberflächen beschrieben und ebenfalls bewertet.

Kapitel 4 zeigt das Ergebnis dieser Arbeit, erklärt den Aufbau und die Anwendung der entstandenen API und enthält eine Bewertung unter Berücksichtigung der in Kapitel 2 gewonnenen Ergebnisse.

Die Entwicklung von Schnittstellen zur Interaktion mit Computern wird von den meisten Forschern als interdisziplinäre Aufgabe angesehen. Speziell für Informatiker bedeutet dies die Einnahme einer Vermittlerrolle zwischen Psychologen, Designern und den Benutzern der verschiedenen Anwendergruppen, für welche die Schnittstelle entwickelt werden soll. Viele der hier beschriebenen Grundlagen kommen aus dem Bereich der Psychologie. Ich habe mich bemüht, alle Quellen in ein Gesamtkonzept einzuarbeiten, wie es meiner Meinung für dieses Thema sinnvoll ist.

2 Grundlagen

Dieses Kapitel beschreibt die Anforderungen an eine natürlich bedienbare Benutzungsschnittstelle, welche möglichst alle Sinne des Benutzer stimulieren sollte und somit die höchstmögliche Bandbreite an Informationsübertragung ausnutzt. Die besonderen Eigenschaften einer virtuellen Umgebung und deren Einfluss auf die menschliche Wahrnehmung werden aufgezeigt.

Die Virtual Glove Box ist ein Gerät, mit dessen Hilfe immersive Virtual Reality Anwendungen bedient und beobachtet werden sollen. Eine Benutzungsoberfläche für eine solches Gerät erfordert aufgrund der speziellen Ein- und Ausgabegeräte spezielle Anpassungen für die Anwendungsbedienung. Diese trägt entscheidend zur Ausprägung des Gefühls der Immersion bei, dem Gefühl ein Teil der Szene sein. Dessen Ausprägung ist ein Indikator für die Benutzbarkeit der Anwendung auf der einen und der Qualität der Ein-Ausgabegeräte auf der anderen Seite. Als optimal wird hierbei die Direktmanipulation der virtuellen Objekte bei minimaler Kluft zwischen den Absichten des Benutzers und den Anforderungen der physikalischen Steuerung des Systems angesehen [2]. Erreicht werden kann dies durch Bereitstellung von sensorischem Input für den Menschen, durch Bedienen aller Sinne mit kohärentem Feedback auf die Eingaben des Benutzers. Die notwendige Technik steckt allerdings noch in den Kinderschuhen und entwickelt sich eher in Richtung spezialisierter Eingabegeräte für verschiedene Einsatzbereiche, wie im folgenden zu sehen ist.

2.1 Virtuelle Realität

Der Sinn mit der höchsten Bandbreite für verarbeitbare Informationen ist der visuelle [3]. Die bereitgestellte Information sollte möglichst realistisch dargeboten werden, denn die wahrgenommenen Informationen werden nicht in bezug auf die Objekte der virtuellen Szene ausgewertet. Vielmehr werden Rückschlüsse auf die eigene Körperlage und Bewegungen geführt. Widersprüchliche Informationen können im einfachsten Fall Unbehagen, in einigen Fällen sogar Übelkeit auslösen [2]. Man denke an die Simulation einer Achterbahnfahrt, bei welcher man jedoch nur vor einer großen Leinwand steht. Dabei können für die Anwendung unwichtige Informationen wie die Simulation einer natürlichen Umgebung sehr hilfreich für die Orientierung und das Wohlbefinden in der virtuellen Umgebung sein. Denn der wahrgenommene Unterschied zwischen realer und künstlicher Umgebung bestimmt, als wie realistisch die Simulation wahrgenommen wird [2].

Das Ziel der Entwicklung müssen Geräte sein, welche die menschlichen Kommunikationskanäle voll ausnutzen und eine natürliche Benutzung des Computers ermöglichen [2]. Nur so kann die kognitive Belastung während der Benutzung reduziert werden. Die Interaktion zwischen Mensch und Computer kann dann von einer diskreten, kommandoorientierten Form in eine kontinuierliche, natürliche übergehen. Die Rotation eines Objekts müsste nicht mehr durch Aufrufen verschiedener Befehle oder graphischer Eingabemetaphern durchgeführt werden, man könnte das Objekt ganz einfach in die Hand nehmen und drehen. Der normale Benutzer eines Computers muss dann nicht mehr umdenken oder geschult werden, bereits erworbene Fähigkeiten wie Greifen, Zeigen, Laufen oder auch berufsspezifische Abläufe könnten für die Interaktion mit der Maschine ausreichen sein.

2.1.1 Immersion

„Haben Sie sich jemals geduckt, weil ein Objekt auf dem Bildschirm schnell auf Sie zugeflogen kam?“ Die meisten Leute werden diese Frage mit einem

klaren „Nein!“ beantworten. Doch genau das ist ein großes Problem. Es ist leider nicht ausreichend, nur einen Sinn mit Input zu versorgen. Die bisher erwähnten Ansprüche an ein Virtual-Reality-System bedürfen einer gründlich gewählten Kombination von Reizen, welche im Kopf des Benutzers zu einem sinnvollen Ganzen verwoben werden können. Die erzielbare Performanz innerhalb einer künstlichen Umgebung hängt davon ab, wie sehr sich der Benutzer in diese Szene eingebunden fühlt [16]. Eine virtuelle Hand beispielsweise wird nur als die eigene wahrgenommen, wenn sie in Form und Position der eigenen entspricht (was auch für alle anderen Teile des Körpers gilt). Der Benutzer sollte das Gefühl verspüren, einen Ort zu besuchen als nur ein Bild zu betrachten, er sollte sich als Teil der Umgebung fühlen. In [16] wird ein Versuch beschrieben, bei welchem ein 3D-Schach in einer virtuellen Umgebung gespielt werden sollte. Die Ergebnisse zeigen einen Zusammenhang von Performanz-Gewinn und der Darbietung einer möglichst realistischen Umgebung in der Szene, nicht nur die eigentliche Aufgabe betreffend. So spielte eine Gruppe in einer leeren Umgebung, die andere in einer realistisch gestalteten. Letztere erzielte eine höhere Performanz bei der Lösung der Aufgabe, ganz allgemein war die Performanz vom Gefühl der Präsenz in der Szene abhängig. Dieser Versuch zeigt, dass die Präsentation der Aufgabe über die relevanten Informationen hinaus die menschlichen Sinne bedienen muss, wenn der Benutzer sich als Teil der virtuellen Welt fühlen soll (Abbildung 1).



Abbildung 1: Tele-Immersion Projekt

2.1.1.1 Räumliche Wahrnehmung

Im Zusammenhang mit Immersion steht die Fähigkeit des zugrundeliegenden Systems, räumliche Informationen zu vermitteln. Ohne Tiefenwahrnehmung fällt die Orientierung in einer virtuellen Szene deutlich schwerer, da Entfernungen nicht abgeschätzt werden können. Die räumliche Orientierung dient auch dem Erwerb von Routenwissen, was in Verbindung mit Landmarken zum Erlernen kognitiver Landkarten führt¹. Das Routenwissen wird durch Erkundung der Umgebung erlangt, die Landmarken sind hervortretende Merkmale auf diesen Routen. Es entsteht ein Wissen über den globalen Zusammenhang der besuchten Orte, was die Navigation in der Umgebung erlaubt. Diese Art von Wissen ist prozedural gespeichert, demnach ohne bewussten Einsatz des Gedächtnisses abrufbar und nicht konzentrationsstörend.

¹ Einen umfassenden Überblick über die Grundlagen menschlicher Wahrnehmung, insbesondere der visuellen Informationsverarbeitung und die Speicherung von Wissen, findet der Leser in [11]

2.1.1.2 Sprache

Jeder Gedanke kann ausgedrückt und jedes Gefühl kann beschrieben werden. Eine Steuerung des Computers durch Sprache versetzt den Benutzer in die Lage, abstrakte Aufgaben zu steuern, ohne dass hierfür eine Beschreibung durch Gesten, Eingabegeräte oder Metaphern jeglicher Art nötig ist. Eine Aussage wie „Addiere zwei und drei!“ könnte sofort zur Anzeige des Ergebnisses führen, ohne Aufruf einer Rechenanwendung und der Eingabe des Terms über eine Tastatur. Auch diskrete Aktionen lassen sich sehr leicht und fehlerfrei durch simple Sätze beschreiben. Eine Lampe könnte mit dem Satz „Lampe 1 an!“ eingeschaltet werden

Zum jetzigen Stand der Technik ist eine Erkennung der Sprache fast schon fehlerfrei möglich, der Sinn einer Aussage kann vom Computer jedoch nicht ermittelt werden. Der Benutzer müsste sich an eine Grammatik gewöhnen oder die Befehle auf einzelne Worte beschränkt werden. Diese Befehle können jedoch unabhängig von der Position der Hände und der ausgeführten Aktion ausgesprochen werden. Beispielsweise könnte man sich ein Werkzeug in die Hand geben oder sich für ein Objekt eine Liste von Eigenschaften anzeigen lassen. Denkbar sind auch Manipulationen von Objekten, welche außerhalb der Reichweite der Hände liegen. In [8] wird über Versuche mit Spracheingabe berichtet, was in allen Fällen zu einer deutlichen Steigerung der Performanz führte.

2.2 Der Mensch hat zwei Hände

Die Forderung nach einem System, welches eine direkte Manipulation der virtuellen Objekte ermöglicht, bedeutet auch, dass der Einsatz der Hände des Benutzers auf natürliche Art unterstützt werden muss. Kommt eine virtuelle Hand zum Einsatz, ist dieses Problem sofort gelöst. Jedes andere Eingabegerät als der Datenhandschuh stellt eine Reduktion der Freiheitsgrade der Hand dar. Dass dies nicht immer schlechter ist, zeigt dieser Abschnitt.

Wenn man eine Benutzungsschnittstelle konzipieren möchte, welche mit den Händen bedient werden soll, ist natürlich auch eine Untersuchung der Möglichkeiten der Hand und der für sie ausführbaren Aufgaben nötig. Leider sind für den Bereich der Computertechnik nicht sehr viele Untersuchungen durchgeführt worden, so dass an dieser Stelle mehr Fragen offen bleiben als beantwortet werden. Möglicherweise liegt dies an schlechter oder fehlender Hardware, welche den Einsatz der Hände in Virtual-Reality Anwendungen nicht sinnvoll erscheinen lässt, so dass Entwickler und Forscher ihre Arbeit in andere Projekte investierten. Vielleicht mangelt es auch an haptischem Feedback bei reinen Datenhandschuhen, was die Interaktion mit virtuellen Objekten genau so schwierig macht, wie es mit konventionellen Eingabegeräten der Fall ist.

Die Hände bieten in der realen Welt noch immer eine unerreichte Zahl von Einsatzmöglichkeiten, und es gibt keinen Grund, sich diese nicht auch in virtuellen Umgebungen zu nutze zu machen. Der hohe Aufwand macht allerdings Prüfung des zu erwartenden Vorteils notwendig. Dieser kann durch Einsparungen materieller Art, eine erhöhte Performanz beim Bearbeiten von Aufgaben oder eine geringere Einarbeitungszeit bei neuen Anwendungen erreicht werden.

Die für diese Arbeit wichtigsten Untersuchungen und Erkenntnisse, welche in Zusammenhang mit Interaktion in virtuellen Welten stehen, möchte ich hier vorstellen. Der aktuelle Wissensstand erlaubt leider keine Vorgabe einer

einheitlichen Vorgehensweise bei der Planung von graphischen oder gar haptischen Benutzungsoberflächen, daher sind die folgenden Ausführungen als Richtlinie zu betrachten.

2.2.1 Ist der Einsatz der Hände sinnvoll?

Die Hände bieten ein deutlich breiteres Spektrum an Einsatzmöglichkeiten als jedes andere Eingabegerät. Zudem sind sie ein Teil unseres Körpers, was eine intuitive, nicht kognitiv belastende Benutzung erlaubt. Stellt man die Frage, ob wirklich jede Aufgabe durch den Einsatz der Hände als Eingabegerät beschleunigt oder verbessert werden kann, so lautet die Antwort klar „Nein!“. Die vielleicht häufigste Aufgabe am Computer, das Eingeben von Buchstaben mit Hilfe der Tastatur, ist mit heutigen Mitteln effizient durchführbar. Das Eingabegerät (die Tastatur) ist der Aufgabe gut angepasst, auch wenn es etwas Übung bedarf. Eine Umstellung auf den Einsatz virtueller Hände macht keinen Sinn, da eine ebenbürtige virtuelle Alternative für Texteingaben nicht existiert. Ein anderes Beispiel ist das Zeichnen von Linien im 2D. Mit Hilfe der Maus und einem Raster für die genaue Positionierung ist diese Aufgabe gut und schnell auszuführen.

Was wäre jedoch, wenn die Aufgabe komplizierter ist: Translation eines Objekts, verbunden mit einer Rotation um neunzig Grad, zusätzlich soll das Objekt noch verformt und die Oberfläche abgetastet werden. Mit dieser in der realen Welt ganz natürlichen Aktion überfordert man jedes bekannte Eingabegerät (allein durch das Abtasten der Oberfläche). Was aber macht diesen Unterschied aus? Den steigenden Anforderungen der Anwendungen in bezug auf deren Steuerung sind konventionelle Eingabegeräte nicht gewachsen, da sie die gleichzeitige Ausführung der Manipulationen nicht ermöglichen. Vor allem im Bereich der 3D-Simulationen wird der Einsatz von Eingabegeräten mit vielen Freiheitsgraden erforderlich, falls dem späteren Benutzer ein einfacher Zugang ermöglicht werden soll.

2.2.2 Geste und Postur

Gesten bieten eine Möglichkeit zur Kommunikation ohne Sprache und Schrift. Jeder Ausdruck kann durch entsprechende Gesten deutlich gemacht werden. Die Gebärdensprache hörgeschädigter Menschen zeigt dies. Die folgenden Punkte beschreiben die Probleme beim Einsatz ([1][8][14][24]):

- Gesten werden für die Ausführung von Aktionen eingesetzt, was auf eine natürliche (oder natürlichere) Art die Interaktion mit der Anwendung erlauben soll. Aktionen wie Navigation oder Selektion sind jedoch keineswegs durch die verwendeten Gesten auf natürliche Weise zu steuern. Die begrenzte Anzahl möglicher Gesten schränkt zusätzlich stark ein. Man denke an eine Zeige-Geste: den Zeigefinger nach vorne zu strecken ist für die meisten Menschen kein Problem, den Mittelfinger wird man für eine solche Aufgabe nicht benutzen (erst recht nicht in Multi-User Umgebungen), den Ringfinger werden die meisten Menschen nicht unabhängig von den anderen Fingern strecken können, und den kleinen Finger setzen die wenigsten Menschen für diese Geste ein (es wäre keine natürliche Aktion).
- Die Erfassung von Gesten durch die Software stellt ein großes Problem dar. Die Frage ist, ab wann der Benutzer die Geste zeigt und nicht seine Hand für andere Aufgaben benutzt oder sie nicht der Spezifikation der Geste entsprechend hält, obwohl er der Meinung ist, dass er klare Eingaben produziert. Der Einsatz von Pinch-Gloves zur Erkennung von Posturen bieten zwar eine relativ sichere Erkennung, unbeabsichtigte Eingaben können auch sie nicht verhindern.
- Die Verfügbarkeit von Technologien zur Vermittlung von haptischem Feedback und die in naher Zukunft verfügbare Steuerung durch Sprache bieten einen leichteren und natürlicheren Zugang, wengleich Gesten durchaus zum alltäglichen Repertoire menschlicher

Kommunikation gehören. Die ausgedrückte Information ist meist nur eine ungenaue Untermalung gesprochener Worte.



Abbildung 2: "Put-That-There"

Eine Ausnahme gibt es jedoch: die einzig wirklich hilfreiche Postur, welche einen echten Informationswert besitzt, ist die eben erwähnte Zeige-Postur. Sogar ein Finger wurde nach ihr benannt. Zudem ist sie die erste koordinierte Handbewegung des Menschen und wird wohl fast jeden Tag in Verbindung mit Sprachinformationen eingesetzt: „Put that there“ ([27], Abbildung 2).

2.2.3 Klassifizierung der Hand-Aktionen

Ich habe bereits Gesten als Möglichkeit der Interaktion erwähnt. Für genauere Betrachtungen ist jedoch ein Modell der möglichen Aktionen der Hand nötig, wobei zwei Hauptkategorien aufgestellt werden können:

- Kontinuierliche Aktionen: basieren auf der Anzahl der Freiheitsgrade der Hand und beinhalten kontinuierliche Größen wie Position der Fingerspitzen, Drehgeschwindigkeit der Gelenke oder der

einwirkenden Kraft auf die Hand. Gesten fallen in diesen Bereich, aber auch das Bewegen der Maus oder des Joysticks.

- Diskrete Aktionen: basieren auf statischen Werten der Eigenschaften der Hand. Ein Beispiel ist das Ballen zur Faust oder das Zeigen auf ein Objekt mit dem Zeigefinger, oder das Drücken der Maustaste.

Beide Kategorien können auf drei verschiedene Arten interpretiert werden (direkt, abgebildet (mapped), symbolisch), so dass Hand-Aktionen insgesamt durch sechs Kategorien beschrieben werden [14]:

- Kontinuierlich/direkt: Die Daten einer virtuellen Hand werden kontinuierlich den realen, kinematischen Werten angepasst.
- Kontinuierlich/abgebildet: Die kontinuierlichen Daten der realen Hand werden auf ein Eingabegerät abgebildet. Das Bedienen der Computer-Maus ist eine Abbildung der Bewegungen der Hand.
- Kontinuierlich/symbolisch: Die Anwendung interpretiert die kontinuierlichen Hand-Daten und interpretiert die Intention des Benutzers. Eine Geste in einer virtuellen Umgebung fällt in diese Kategorie.
- Diskret/direkt: Diskrete Hand-Posturen werden direkt in eine manipulierende Aktion überführt.
- Diskret/abgebildet: Diskrete Hand-Posturen aktivieren eine diskrete Aktion. Beispielsweise bewegt sich die virtuelle Kamera in Richtung eines ausgestreckten Zeigefingers.
- Diskret/symbolisch: Natürliche diskrete Hand-Posturen werden benutzt, um Kommandos für eine Anwendung zu implementieren,

wie das Ausführen einer Stop-Postur (Handrücken in Richtung Schulter, alle Finger nach oben gerichtet), um ein Objekt zu stoppen.

Demnach ist das Benutzen der Maus in einer Window-Umgebung eine Kombination aus kontinuierlichen und diskreten Aktionen. Diskrete Operationen finden sich allgemein im Bereich künstlich erzeugter Dinge, insbesondere im Bereich von Benutzungsschnittstellen. Kontinuierliche Operationen sind jedoch die natürliche Art der Benutzung der Hände. Man kann ganz allgemein sagen, dass der Mensch von Natur aus keine diskreten Abläufe kennt. Er hat sie erfunden, da sie die Kommunikation in einigen Bereichen vereinfachen, zum Beispiel im Straßenverkehr oder generell im Bereich von Technik und Mathematik. Für die Informationsaufnahme hat diese Methode sicher ihre Berechtigung, eine diskrete Steuerung einer Maschine mit Hilfe von Tasten und Reglern ist bei sinnvoller Anordnung und entsprechendem Feedback oft noch sinnvoll, die Steuerung eines Computerprogramms mit Maus und Maustaste bedeutet für den Benutzer jedoch eine hohe kognitive Belastung, da schon die Positionierung des Zeigers volle Aufmerksamkeit verlangt. Die Bewegung ist obiger Taxonomie entsprechend kontinuierlich, jedoch so schwer zu steuern, dass sie nicht intuitiv ausgeführt werden kann.

Durch den Einsatz virtueller Hände kann zumindest der Stand der Technik der realen Welt der Benutzungsschnittstellen auf die Software-Technologie angewendet werden. Wie schon erwähnt, ist besonders der Bereich der 3D-Simulationen in diesem Zusammenhang von Bedeutung, wo die Steuerung einzelner Objekte mit Hilfe der Hände stark vereinfacht werden und eine direkte Manipulation erfolgen kann, sofern ein haptisches Feedback vom System geliefert wird.

Ohne den Einsatz von haptischem Feedback ist die Steuerung einer Anwendung durch virtuelle Hände nur mit Hilfe weiterer Eingabemetaphern möglich. Diese können aber auch durch herkömmliche Eingabegeräte

gesteuert werden. Möchte man beispielsweise ein Objekt greifen, wird der Moment der Berührung vor allem taktil wahrgenommen. Die erforderliche Hand-Postur ergibt sich aus der Form des Objekts. In einer virtuellen Umgebung ohne haptisches Feedback kann dieser Mechanismus nicht funktionieren, so dass der Zustand „Gegriffen²“ nicht bestimmt werden kann. Der Benutzer muss eine weitere Aktion zusätzlich zum Positionieren der Hand und der Finger ausführen, um der Anwendung zu signalisieren, dass er gegriffen hat. Die eigentlich abgeschlossene Aktion muss also zusätzlich bestätigt werden.

2.2.4 Bezugsrahmen und kinematische Ketten

Wir haben nun ein einfaches Modell der Einsatzmöglichkeiten der Hand. Es stellt sich jedoch die Frage, welche Einschränkungen es gibt und welche Voraussetzungen erfüllt sein müssen, damit die Hand sinnvoll eingesetzt werden kann. Es existiert zur Zeit kein Modell, welches eine genaue Klassifizierung zweihändiger Aktionen ermöglicht.

Die Hand ist Teil einer kinematischen Kette, eines Systems, welches aus starren, in Serie platzierten Teilen besteht ([8][18][28]). Für den menschlichen Arm bedeutet dies, dass der Oberarm mit der Schulter verbunden ist, der Unterarm mit dem Oberarm, die Hand mit dem Unterarm, die Finger mit der Hand. Jedes Teil der Kette wird in seinem Einflussbereich von den hierarchisch über ihm stehenden Teilen beeinflusst: wird der Oberarm bewegt, hat dies Einfluss auf die Position des Unterarms und dies wiederum hat Einfluss auf die Hand. Der Einsatzbereich der Hand wird durch diese Vorgaben auf das Gesichtsfeld beschränkt.

Die Hand ist jedoch nicht nur durch die Anatomie des Menschen beschränkt. Die beiden Hände erledigen unterschiedliche Aufgaben. Man unterscheidet hier zwischen der dominanten und der nicht-dominanten Hand: bei einem

² Dieser Begriff soll nicht als diskreter Zustand verstanden werden, sondern umschreibt den kontinuierlichen Prozess des Anpassens der von der Hand aufzubringenden Kräfte, so dass das Objekt gehalten werden kann.

Rechtshänder ist die rechte Hand die dominante. Im Zusammenspiel scheint die nicht-dominante Hand der dominanten als eine Art „Wegbereiter“ zu dienen, so dass die dominante bei der Ausführung einer Aufgabe unterstützt wird. Somit gibt die nicht-dominante Hand den Bezugsrahmen für die dominante vor. Das Schreiben auf ein Stück Papier verdeutlicht diesen Ablauf:

- Räumlicher Bezugsrahmen von dominant zu nicht-dominant: die nicht-dominante Hand hält und positioniert das Blatt, welches von der dominanten Hand beschrieben wird.
- Ungleichgewicht der Bewegungen: die dominante Hand wird deutlich häufiger und schneller eingesetzt als die nicht-dominante, die Hände arbeiten asymmetrisch. Das Blatt wird deutlich seltener durch die nicht-dominante Hand bewegt als der Stift durch die dominante.
- Vorrang der nicht-dominanten Hand: die nicht-dominante Hand beginnt normalerweise den Ablauf des asymmetrischen Ablaufs. Das Blatt wird erst positioniert, dann erst beginnt die dominante Hand das Schreiben.

Dieses Modell gilt nur für die Ausführung von Aufgaben, bei denen auch beide Hände zum Einsatz kommen und die in Teilaufgaben unterteilt werden können. Das Fangen eines Tennisballs wird nur von einer Hand unabhängig von der anderen durchgeführt (normalerweise der dominanten). Das Fangen eines Fußballs erfordert den Einsatz beider Hände, hier sind beide Hände in gleichem Maß beteiligt, die Aufgabe an sich ist jedoch nicht von abstrakter Natur.

Direkte Manipulation von Objekten in virtuellen Welten erfordert aber genau das oben beschriebene Zusammenspiel der Hände, so dass für den Bereich der Benutzungsschnittstellen davon ausgegangen werden kann, dass sich der Bezugsrahmen für die Interaktion mit virtuellen Objekten zwischen den

Händen befindet, dynamisch bestimmt durch die Position der nicht-dominanten Hand. Die nicht-dominante Hand führt normalerweise intuitive Aufgaben aus, welche demnach nicht von der eigentlichen, abstrakten Aufgabe der dominanten Hand ablenken und ist somit eine unersetzliche Hilfe, deren Fehlen sich auch bei scheinbar „einhändigen“ Aufgaben wie Schreiben sofort unangenehm und performanzreduzierend auswirkt.

2.2.5 Input-Performanz

Ob der Einsatz der Hände allgemein einen echten Performanzgewinn erzielt, ist ebenfalls kaum erforscht. Das Problem bei Aussagen über die Performanz besteht darin, dass bisher kein Modell existiert, das Aussagen über die Lösbarkeit von Aufgaben mit Hilfe der Hand zulässt ([8][10][14][15][18][21]). Dies wird auch nur äußerst schwierig aufzustellen sein, da zum einen kontinuierliche Aktionen qualitativ bewertet werden müssten (wobei unterschiedliche Vorgehensweisen zum gleichen Ergebnis führen können), zum anderen wäre eine Untersuchung aller möglichen Kombinationen von Hand-Aktionen erforderlich, so dass ein Gesamtbild entstehen könnte. Die folgenden Aussagen zeigen einen Rahmen für die Fähigkeiten der Hand:

- Eine perfekte Linie kann nicht gezeichnet werden, sei es mit einer oder mit beiden Händen. Für solche Aufgaben funktioniert die Hand nicht genau genug. Das Zeichnen von Bildern jedoch kann von geübten Menschen fast foto-realistisch durchgeführt werden. Der Unterschied besteht darin, dass beim Zeichnen kontinuierliches visuelles Feedback dargeboten wird, so dass der Zeichner kontinuierlich die nächste Bewegung an den Gesamteindruck anpassen kann. Das Ziehen einer Linie bietet keinen Spielraum für Fehler.
- Der Einsatz von Werkzeugen erlaubt das Ausführen von Aufgaben, welche durch die „nackte“ Hand nicht durchführbar wären. So kann ein Nagel nicht ohne Hammer in die Wand gehauen oder ein Faden

nicht ohne Nadel in Stoff eingewebt werden. Eine perfekte Linie kann mit Hilfe eines Lineals gezeichnet werden. Die Hand kann in diesem Fall das Werkzeug auf durchaus unterschiedliche Arten benutzen, so dass es noch besser an die jeweilige Aufgabe angepasst werden kann. Auch können Werkzeuge für Aufgaben eingesetzt werden, für die sie nicht konzipiert wurden. So kann mit einem Messer eine Schraube gedreht oder mit einem Glas ein Teig ausgerollt werden.

- Manche Aufgaben lassen sich mit Hilfe der Hände oder mit Hilfe von Werkzeugen lösen. Die Wahl der passenden Methode hängt in diesem Fall von der Übung und den Fähigkeiten des Menschen ab.

Wenn man nun die Anforderungen an eine Benutzungsoberfläche betrachtet, kann man das Benutzen von Eingabegeräten durchaus als das Benutzen von Werkzeugen betrachten. Die Performanz und Benutzbarkeit muss dann im Einzelfall getestet werden. Wenn die Aufgabe nur zwei Freiheitsgrade erfordert, beispielsweise das Positionieren des Zeigers, so wird der Einsatz einer virtuellen Hand zur Direktmanipulation des Zeigers kaum einen Performanzvorteil bringen, da die vielen Freiheitsgrade der Hand nicht benötigt werden.

Aufgrund obiger Aussagen könnte man zu der Ansicht gelangen, dass die rein anatomisch gesehen identischen Hände des Menschen rein funktional gesehen völlig unterschiedlich sind. Dies ist nur bedingt richtig. Der Mensch kann sich nur auf eine schwierige Aufgabe gleichzeitig konzentrieren. Schwierig bedeutet in diesem Zusammenhang, dass die volle Aufmerksamkeit auf diese Aufgabe gerichtet ist. Das Bearbeiten zweier schwieriger Aufgaben bedeutet ein geistiges Umschalten zwischen diesen, ein paralleler Ablauf ist nicht möglich. Ist nun diese Aufgabe durch den Einsatz einer einzelnen Hand durchführbar, so wird die dominante Hand eingesetzt. Die nicht-dominante Hand könnte diese Aufgaben bei Bedarf in vollem Umfang übernehmen,

allerdings ohne die feinmotorische Abstimmung der dominanten Hand, was an mangelnder Übung der nicht-dominanten Hand liegt. Sogar die Füße können viele Aufgaben der Hand übernehmen, wenn sie entsprechend trainiert werden. Im Normalfall wird dennoch die dominante Hand die feinmotorischen Aufgaben übernehmen. Um einen Performanzgewinn zu erzielen und intuitive Bedienung zu gewährleisten, sollten dem Benutzer nicht zwei schwierige Aufgaben gleichzeitig zugemutet werden.

2.3 Haptik

Taktile Wahrnehmung ist die Voraussetzung für jede Art von koordinierter Bewegung. Sie liefert die nötige Rückmeldung über Position und Haltung unseres Körpers. Fehlen diese Informationen sind besonders feinmotorische Aufgaben kaum noch zu erfüllen. Das Greifen eines Objekts mit einer virtuellen Hand wäre allein von der Qualität der räumlichen Darstellung und der räumlichen Wahrnehmung des Benutzers abhängig. Das Fehlen taktiler Informationen fällt auf, sobald sie fehlen. Simulieren lässt sich dies bei Kälte, da das haptische System dann nicht mehr funktioniert [2].

Der visuelle Sinn ist zwar der dominanteste, es treten jedoch bei mangelnder Versorgung mit verwertbarer Information die anderen Sinne immer weiter in den Vordergrund. Man denke an einen schlecht beleuchteten Raum, in dem man durch Tasten den Weg zur Tür oder zum Lichtschalter sucht. Dieses Beispiel veranschaulicht auch die Mächtigkeit dieses Sinns: allein durch aktives Fühlen kann ein kognitives Bild der gesamten Umgebung erstellt werden, inklusive Informationen über die Oberflächenbeschaffenheit von Objekten [3]. Allerdings ist die verfügbare Auflösung begrenzt: an den Stellen mit der höchsten Auflösung, den Fingerspitzen, müssen zwei Objekte mindestens 2,5 Millimeter voneinander entfernt liegen, um nicht als nur ein Objekt gefühlt zu werden. Die Höhe eines Objekts relativ zum Untergrund braucht hingegen nur wenige tausendstel Millimeter zu betragen, um wahrgenommen zu werden. In [3] wird sogar ein Weg aufgezeigt, wie aus graphischen Daten in Echtzeit taktile Information für ein dynamisches taktiler Display gewonnen werden können.

Der Einsatz von Haptik in virtuellen Realitäten ist zudem eine Möglichkeit, die Bandbreite der vermittelbaren Informationen zu erhöhen, ohne kognitive Ressourcen zu verbrauchen. Durch die Zugabe von haptischem Feedback wird auch die Direktmanipulation von Objekten mit Hilfe von virtuellen Händen ermöglicht. Ohne Feedback ist es nicht möglich, ein Objekt zu greifen oder zu drehen, da die Form nicht erfühlt werden kann, man greift

hindurch. Dies führt zu einem Widerspruch zwischen in der realen Welt Erlerntem und der Wahrnehmung in der Simulation, die Bedienung wird unnatürlich und es entsteht der Bedarf an Eingabemetaphern, was den Vorteil der virtuellen Hand als Eingabegerät zunichte gemacht.

2.3.1 Passive Haptische Wahrnehmung

Haptisches Feedback ist wie sensorische Wahrnehmung im allgemeinen ein kontinuierlicher Prozess. Wir nehmen viele Dinge nur unbewusst wahr und werten die gewonnenen Informationen nicht aus, solange sie nicht unsere Aufmerksamkeit erregen. So fühlen wir die Temperatur, Oberflächen, Formen oder auch auf die Hand einwirkende Kräfte beim Händeschütteln. Weicht unsere Wahrnehmung jedoch von gewöhnlichen Mustern ab, wird sich die Konzentration auf diese Abweichung lenken. Passives Feedback hilft uns, außergewöhnliche Zustände zu erkennen. Die Wahrnehmung von Schmerz dient gar unserer Sicherheit. Dies macht einen nicht offensichtlichen Aspekt der Wahrnehmung deutlich: wenn wir denken, dass wir keine Schmerzen haben, sind die von den Nerven transportierten Signale nicht stark genug sind, um Schmerz empfinden zu lassen. Ein Signal wird dennoch übermittelt.



Abbildung 3: Passive haptische Wahrnehmung findet auch im Sitzen statt

2.3.2 Oberflächenbeschaffenheit

Oberflächenstrukturen können durch Darbietung von Vibrationen an den Fingerspitzen vermittelt werden, da die Sensoren in der Haut des Menschen durch Streichen über eine Oberfläche in Schwingungen versetzt werden. Dieser Mechanismus durch künstliche Vibrationen getäuscht werden kann [2]. Diese Art der Stimulation erlaubt ebenfalls ein realistischeres Erleben der virtuellen Umgebung.

2.3.3 Aktive Haptische Wahrnehmung

Zusätzlich zu den passiv erworbenen Informationen kann haptische Information auch bewusst aktiv erworben werden. In [4] werden folgende Arten der haptischen Exploration unterschieden:

- Seitliche Bewegung: Beurteilung der Oberflächenbeschaffenheit
- Ausüben von Druck: Beurteilung der Härte, normalerweise im rechten Winkel zur Oberfläche
- Statischer Kontakt: Beurteilen der Temperatur
- Freier Halt: Beurteilung des Gewichts
- Umgreifen: Bestimmung von globaler Form und Volumen
- Verfolgen von Konturen: Bestimmung von globaler Form und exakter Form
- Testen der Funktionalität: Erforschen der speziellen Funktion
- Prüfen der Beweglichkeit einzelner Teile

Der Mensch benutzt diese Arten der Exploration sehr systematisch. Die gewonnenen Informationen gehen dabei über das allein durch Sehen Wahrnehmbare hinaus. Die Sinne ergänzen sich und sind keineswegs

voneinander unabhängig. Die Performanz einer solchen Aktion ist von den Erfahrungen des Menschen abhängig, da Wissen über taktile Information implizit gespeichert ist und keine bewusste Verarbeitung erfordert. Eine unbekannte Form verlangt daher mehr Aufmerksamkeit als ein Objekt, welches sich in nur einem Merkmal von einem bekannten Gegenstand unterscheidet.

2.3.4 Sehen mit den Händen

Wie bereits erwähnt ergänzen sich die Sinne, um ein möglichst vollständiges Bild eines wahrgenommenen Objekts zu gewinnen. In [17] geht man sogar einen Schritt weiter. Im dort aufgestellten Modell wird die Wahrnehmung der Hand als analog zum Sehen eines Objekts von der Rückseite definiert. Desweiteren wird aus den gewonnenen Ergebnissen eine Blickpunkt-Abhängigkeit der haptischen Wahrnehmung abgeleitet, genau entgegen der visuellen Blickrichtung.

Möglicherweise versucht dieses Modell eine zu strikte Beschreibung der Zusammenhänge zwischen haptischer und visueller Wahrnehmung. Die Wichtigkeit der haptischen Wahrnehmung wird jedoch deutlich.

Unabhängig vom Zusammenspiel der einzelnen Sinne macht folgendes Beispiel die Mächtigkeit dieses Sinns deutlich: ein blinder Mensch kann allein durch aktives Fühlen und Erkunden mit den Händen seine gesamte Umgebung wahrnehmen und im Detail beschreiben, abgesehen von der Farbe eines Objekts. Allerdings dauert dies deutlich länger als die visuelle Wahrnehmung, der Mensch ist und bleibt ein Augentier.

2.3.5 Psychomotorische Prozesse

Durch Erfahrung im Umgang mit dem eigenen Körper können Handlungen beschleunigt ausgeführt werden, da Aktionen bei entsprechender Darbietung von Reizen antizipiert werden können und den Körper auf die nächste Aktion vorbereiten. Ein Sportler erkennt durch eine Bewegung seines Gegners dessen Vorhaben, ein Autofahrer bewegt die Hand in Richtung des

Schalthebels, wenn der Motor zu hoch dreht. Dies ist nur möglich durch die taktilen Rückmeldungen des Körpers, wodurch die eigene Körperhaltung ermittelt werden und der Körper in einer ähnlichen Situation wieder in eine ähnliche Haltung gebracht werden kann.

2.3.6 Interaktion der Hände mit Objekten

Das Ergreifen von Objekten mit den Händen ist ein hochkomplexer Vorgang, welcher von Seiten der Hardware möglichst genau simuliert werden muss. Dabei kommen alle Fähigkeiten der menschlichen Hand zum Einsatz. In [5] werden folgende Phasen der Interaktion unterschieden:

- **Kontaktphase:** Der Ort einer Berührung lässt sich aufgrund der hohen Rezeptordichte in den Fingern genau bestimmen. Leichte Berührungen der Haut führen nur zu einer Verformung, stärkerer Kontakt führt auch zu Veränderungen der Steifigkeit des Muskelapparates der Hand. Eine Untersuchung der Druckverteilung bei Berührungen findet man in [6].
- **Ergreifen:** Die Erscheinungsform eines Objekts sowie die beabsichtigte Art der Benutzung ist entscheidend für die Stellung der Finger während des Greifvorgangs. Man kann dabei zwei Arten des Greifens unterscheiden: Power-Grasping mit den Zielen Stabilität, Sicherheit und Umschließen mit der ganzen Hand. Precision-Grasping mit den Zielen Empfindlichkeit, Geschicklichkeit und Ergreifen mit den Fingerspitzen. Die Hand wird dabei vom Arm in die jeweils günstigste Position zur Durchführung der Aufgabe gebracht.
- **Manipulieren:** Beim Power-Grasping werden die Hände oft von den Armen unterstützt, das Objekt wird fest gegriffen. Beim Precision-Grasping hingegen kann das Objekt unter Einsatz der Finger bewegt

oder abgetastet werden. Die aufgebrauchten Kräfte liegen dabei nur geringfügig über der zum Halten des Objekts notwendigen Schwelle.

- Mechanischer Widerstand: Die Hand reagiert auf vom Objekt ausgeübte Kräfte beispielsweise durch Erhöhung der Muskelspannung. Dieser Reflex, welcher eigentlich dem Schutz vor Verletzungen dient (Überdehnung des Muskels), kann bewusst zur Durchführung von Bewegungen eingesetzt werden.

Unterstützt wird das Benutzen der Hände wiederum vom visuellen System, welches vorab Informationen über die Lage des zu greifenden Objekts liefert, was wiederum eine Bereitstellung von räumlicher Information erforderlich macht.

2.4 Steuerung von 3D-Simulationen

Die Schnittstelle zu 3D-Anwendungen stellt den Entwickler vor deutlich größere Probleme als dies bei 2D-Anwendungen der Fall ist. In [7] werden 3D-Interfaces als im gleichen Raum existierend wie die zu steuernde Anwendung bezeichnet. Ebenfalls werden die für 3D-Interfaces notwendigen Voraussetzungen aufgezeigt. Fast alle Anwendungen aus den typischen Einsatzgebieten von 3D-Simulationen (CAD/CAM, Animationen, Medizin, Spiele, Visualisierung wissenschaftlicher Daten...) haben bestimmte Formen der Interaktion gemeinsam:

- Skalierung: Viele wissenschaftliche Anwendungen beschränken sich in der Darstellung der Szene auf einen Teil eines speziellen Problems. Kommerzielle Anwendungen oder für einen breiten Benutzerkreis konzipierte Anwendungen müssen jedoch in der Lage sein, eine große Menge von Daten für den Benutzer zugänglich zu machen. Die Visualisierung der Teile-Datenbank eines Flugzeugs, welche mehrere Gigabyte groß sein kann, darf kein Problem darstellen.
- Level of Detail: Die Darstellungsgenauigkeit muss variabel gestaltet sein, um die deutliche Wahrnehmung der relevanten Daten zu gewährleisten. So kann bei der Untersuchung von Luftverwirbelungen eines Flugzeugs die Ansicht des gesamten Rumpfs von Bedeutung sein, im nächsten Schritt interessiert vielleicht nur der Bereich um die Triebwerke.
- Interaktions-Modell: Herkömmliche WIMP-Interfaces³ sind für die meisten 3D-Anwendungen unpassend, besonders dann, wenn Ein- und Ausgabegeräte mit mehr als zwei Freiheitsgraden benutzt werden. Da die Kluft zwischen Anwendung und Interface deutlich geringer ausfällt als im 2D-Fall, werden auch folgende Punkte wichtig(er):

³ Window, Icon, Mouse, Pointer, siehe Kapitel 3.

- a) Eine höhere Bandbreite muss für Ein- und Ausgabe gewährleistet werden
 - b) Die Anzahl der Freiheitsgrade muss gesteigert werden
 - c) Keine Kluft zwischen Aktionen des Benutzers und der Reaktion des Systems, desweiteren müssen die Antworten des Systems kontinuierlich erfolgen
 - d) Probabilistischer Input, weg von diskreten Eingaben, hin zu kontinuierlicher Bedienung des Systems
 - e) Bei Mehrbenutzersystemen müssen simultane Ein- und Ausgabekanäle unterstützt werden
- Interaktionstechniken: Die Fähigkeiten der Anwendung bestimmen das Design der Benutzungsschnittstelle. Eine Schnittstelle zur Steuerung einer Architektur Anwendung muss andere Techniken zur Verfügung stellen als eine Rennsimulation. Einige Schnittstellen zur Interaktion scheinen jedoch fast alle Anwendungen gemeinsam zu haben:
 - a) Objekt-Erzeugung
 - b) Objekt-Selektion
 - c) Objekt-Plazierung und Manipulation
 - d) Kontrolle der Perspektive
 - e) Bereitstellung von Informationen über die Umgebung
 - f) Spezifikation von Beziehungen von Objekten untereinander



Abbildung 4: virtuelle Welten benötigen spezielle Interaktions-Metaphern

2.4.1 3D-Interaktion

Bisher existiert keine generell einsetzbare Sammlung von Interaktions-Komponenten für 3D-Anwendungen, obgleich einige Techniken dafür geeignet scheinen, beispielsweise das Ändern der Kameraposition. Eine der Schwierigkeiten hierbei ist das Implementieren der Techniken für Eingabegeräte wie die Maus, auch sind viele verschiedene Techniken für das gleiche Problem möglich.

Eingaben des Benutzers müssen mit herkömmlicher Hardware erst umgesetzt werden. Die Hand bewegt die Maus, der Zeiger steuert ein Widget, die Interaktion mit dem Widget manipuliert das entsprechende Objekt. Eine Direktmanipulation mit Hilfe entsprechender Hardware mit vielen Freiheitsgraden kann diese Art von Problemen umgehen. Manchmal kann jedoch auch eine Beschränkung des Benutzers dessen Aktionen in eine

sinnvolle Richtung leiten. Zum Zeichnen paralleler Linien ist der Einsatz eines Rasters beispielsweise eine gute Idee.

Navigation in der virtuellen Welt ist für den Benutzer oft schwierig, da die Orientierung schnell verloren geht. Der Einsatz von Orientierungspunkten ist daher empfehlenswert. Entsprechend einfach muss auch das Interface zur Steuerung der Kameraposition gestaltet sein, da die beste Orientierungshilfe nutzlos ist, wenn der Benutzer nicht in der Lage ist, an den gewünschten Punkt zu gelangen. Viele 3D-Interfaces werden in Form von Metaphern beschrieben, deren Aussehen und Form die gekapselte Funktionalität visualisieren sollen. Eingabemetaphern lassen sich in drei Klassen einteilen:

- Physikalische Metaphern: aus der realen Welt entliehen, wie Papierkorb, Ordner.
- Aktions-Metaphern: beschreiben eine Bewegung oder eine Form der Manipulation, wie Auswählen oder Greifen.
- Unrealistische Metaphern: nicht in der realen Welt möglich, wie Teleportieren.

2.4.2 Psychologie der virtuellen Realität

Für die Gestaltung von 3D-Schnittstellen sind einige Besonderheiten der menschlichen Wahrnehmung zu berücksichtigen [12].

2.4.2.1 Navigation

Um im Raum zu navigieren muss Wissen über die Umgebung, in welcher man sich bewegen möchte, vorhanden sein. Ob es sich dabei um die reale oder virtuelle Welt handelt, macht keinen Unterschied. Während einer Bewegung werden Informationen über Position, Geschwindigkeit und Beschleunigung ausgewertet. Positions- und Geschwindigkeitsindikatoren sind normalerweise Eingaben für das visuelle System, beispielsweise Fahrbahnmarkierungen oder Bäume. In einem Flugzeug sind in großer Höhe

daher keine Aussagen über die Geschwindigkeit möglich. In einer virtuellen Umgebung sind diese Informationen jedoch leicht mit Hilfe geeigneter Visualisierungen vermittelbar. Anders sieht es bei der Simulation von Beschleunigungen aus. Es existieren zwar für einige Anwendungen geeignete Systeme für die Simulation von Beschleunigungskräften auf den Körper (Flugsimulator, manche Freizeitparks bieten entsprechende Fahrsimulationen an), allerdings sind auch sie sehr teuer und kaum flexibel. Eine „normale“ 3D-Simulation muss ohne eine solche Vorrichtung auskommen. Die Auswirkungen sind, dass eine am Bildschirm dargestellte Bewegung nicht als solche wahrgenommen wird, vielmehr scheint sich die simulierte Szene um den Betrachter zu bewegen. Der Einsatz von Landmarken und anderer Orientierungshilfen ist in jedem Fall nötig, um dem Benutzer kontinuierlich Aussagen über die eigene Position zu ermöglichen.

2.4.2.2 Merkmale virtueller Objekte

3D-Benutzungsoberflächen sollten unter Berücksichtigung folgender Richtlinien konzipiert werden:

- Funktionelle Naturtreue (functional fidelity): Der Entwurf muss nicht einem realen Gegenstück fotorealistisch entsprechen. Das menschliche Gehirn ist in der Lage, die relevanten Merkmale eines Objekts zu extrahieren und sich darauf zu beschränken. Daher ist es ausreichend, nur die für eine bestimmte Aufgabe relevanten Merkmale eines Objekts darzustellen.

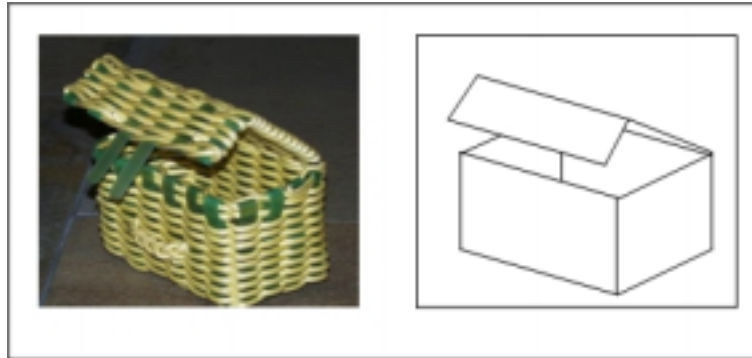


Abbildung 5: Reduktion auf wenige Eigenschaften

Abbildung 5 verdeutlicht dies: beide Objekte sind als Schachteln zu erkennen, auch die gezeichnete ist völlig ausreichend, um in ihr andere Objekte in einer Simulation zu verstauen. Man darf jedoch nicht vergessen, dass eine realistische Darstellung dem Gefühl der Präsenz in der Szene zuträglich ist. Die Darstellung einer quaderförmigen Kiste mit einer Holztextur wird aber das gleiche Ergebnis erzielen wie ein modellierter Weidekorb.

- Reaktionsfähigkeit: Jede Aktion des Benutzers muss sofort vom System beantwortet werden. Nur so kann ein natürlicher Umgang für den Benutzer gewährleistet werden, denn jede Verzögerung bedeutet für den Benutzer erhöhte Konzentration, die Simulation ist nicht mehr interaktiv. Daher sollte größter Wert auf die Synchronisation aller Komponenten der Ein-Ausgabe-Kette gelegt werden. Ebenfalls sollte auch Feedback für das Auslösen einer abstrakten Aktion gegeben werden, so dass der Benutzer immer darüber informiert ist, ob die Anwendung noch läuft oder die Berechnungen bereits beendet sind.

- Affordances⁴: Schon das Erscheinungsbild der Schnittstelle sollte dem Benutzer deren Funktionalität verdeutlichen. Ein Schalter, welcher wie eine Lampe modelliert ist, lässt vermuten, dass das Licht damit geschaltet werden kann. Ein rundes Objekt verleitet eher zum Drehen als ein eckiges. Auch die verwendeten Materialien tragen entscheidend zur Auswahl des Verhaltens bei. Ein Gegenstand aus Glas wird vorsichtiger behandelt als einer aus Metall, eine glatte Oberfläche bietet keinen guten Halt, Papier eignet sich zum Schreiben.
- Erzeugen mentaler Repräsentationen (appeal to mental projections): Eine Schnittstelle muss auch als solche erkennbar sein.
- Mehrkanalige Ein- und Ausgabe: Benutzungsoberflächen sollten nicht nur den visuellen Kanal zur Kommunikation benutzen, es sollten so viele Sinne wie möglich stimuliert werden.

2.4.2.3 3D-Interaktionen

Jeder Mensch ist ständig geistig und körperlich aktiv. Daraus folgt, dass auch die Interaktion mit der Schnittstelle einer Maschine ein kontinuierlicher Prozess ist, wenn auch nicht jede wahrgenommene Information zu einer Aktion des Benutzers führt. Zu nennen sind in diesem Zusammenhang Dinge wie Gravitation oder Reibung beim Manipulieren eines Objekts. Diese Informationen werden nicht bewusst verarbeitet, sie unterstützen jedoch in hohem Maß die Interaktion mit der Welt, sei sie nun real oder virtuell.

2.4.2.4 Ein- und Ausgabegeräte

Die Anzahl der Freiheitsgrade der Eingabegeräte muss den Anforderungen der jeweiligen Aufgabe angepasst sein. Sollte das Eingabegerät weniger Freiheitsgrade unterstützen als die Aufgabe erfordert, also der Task Space ([7]) zu groß sein, sind wieder Metaphern zur Vereinfachung nötig, da direkte

⁴ Es existiert keine Übersetzung für dieses Wort. An dieser Stelle wird es benutzt im Sinne von Wahrnehmung der Eigenschaften eines Objekts, welche einen Rückschluss auf die mögliche Verwendung zulassen oder als den Aufforderungscharakter von Objekten und Materialien.

Manipulation nicht mehr möglich ist. Auf der anderen Seite rechtfertigt diese Aussage absolut die Verwendung der Maus im 2D. Problematisch kann jedoch auch die Verwendung von Eingabegeräten mit vielen Freiheitsgraden bei Aufgaben mit kleinerem Task Space sein. Wie schon erwähnt ist es nicht möglich, eine exakte Linie mit der Hand zu zeichnen. Hier sind wiederum Hilfen in Form von Metaphern nötig, in diesem speziellen Fall wäre dies zum Beispiel ein Lineal.

2.4.2.5 Benutzungsoberflächen, Design und der Benutzer

Wenn die Eingabegeräte sich an der realen Welt orientieren, sollten sich auch die virtuellen Benutzungsoberflächen an dieser orientieren. Die bisher im Bereich des Schnittstellen-Designs gewonnenen Erkenntnisse könnten direkt von Entwicklern von Simulationen adaptiert werden, insbesondere bei der Gestaltung der Form und Handlichkeit der Objekte. Das zu erreichende Ziel ist die Gestaltung von Schnittstellen, für deren Benutzung keine Beschreibung und idealerweise noch nicht einmal eine Beschriftung notwendig ist. Sicherlich wird dies in den meisten Fällen nur schwer zu erreichen sein. Es existieren jedoch Techniken für den sinnvollen Einsatz von Formen, Farben und Ikonen, womit deutlich mehr Informationen codiert werden können als durch den Einsatz von Text.

Die verschiedenen Einsatzbereiche von Benutzungsschnittstellen verlangen auch eine Anpassung des Designs an die jeweilige Benutzergruppe. Desweiteren muss berücksichtigt werden, dass eine solche Schnittstelle von Anfängern und Experten einer Benutzergruppe benutzt werden wird, entsprechend vielseitig und doch schlicht und deterministisch muss sie gestaltet werden. Der goldene Weg scheint demnach nicht zu existieren. Die Grundlagen für ein erfolgreiches Design, zumindest aus technischer Sicht, möchte ich im folgenden darlegen.

2.4.2.6 Das konzeptionelle Modell

Von jedem Objekt, welches von uns wahrgenommen wird, erstellen wir ein mentales Modell, das durch unser Wissen und durch unsere Erfahrung geprägt wird. Dadurch sind wir in der Lage, Dinge in unsere Sicht der Welt einzuordnen, sie zu begreifen und ihnen Eigenschaften zuzuweisen, sie zu benutzen. Ist die Wahrnehmung unvollständig, so werden fehlende Teile durch gespeichertes Wissen ergänzt, was die mentale Repräsentation des Objekts zu einer individuellen Erfahrung macht⁵. Gleiches gilt für mentale Repräsentationen semantischer Zusammenhänge wie Handlungsabläufen, welche für die Steuerung eines Geräts erforderlich sein können. Unterscheidet sich die Benutzungsschnittstelle eines Geräts nicht von der eines anderen mit unterschiedlicher Funktionalität, ist eine Fehlbedienung so gut wie sicher, da der Mensch erlerntes Verhalten generalisiert, er ist ein Gewohnheitstier.



Abbildung 6: Carelmans Tandem, aus [12]

Das Tandem aus Abbildung 6 ist ein sehr gutes Beispiel für diese Fähigkeit. Kaum jemand wird versuchen, auf diesem Gefährt eine Probefahrt zu machen. Man weiß, dass zwei Lenker in der Mitte keinen Sinn machen, dass wir nicht in entgegengesetzte Richtungen fahren können, dass wir das Gleichgewicht nicht werden halten können. Dies ist jedoch nur möglich, weil wir die einzelnen Komponenten des Objekts deutlich sehen können, die

⁵ Dieses Phänomen ist im täglichen Leben häufig bei Zeugenaussagen anzutreffen. Eine Gruppe von Leuten kann einen Unfall beobachten und jeder wird eine andere Beschreibung der beteiligten Fahrzeuge und des Unfallhergangs abgeben. Der Vorgang an sich wird von allen ähnlich beschrieben werden.

Affordances sind in ausreichendem Maß wahrnehmbar. Das Design einer Schnittstelle unterliegt den gleichen Anforderungen: es sollte dem Benutzer die Möglichkeit geben, ein konzeptionelles Modell des Geräts oder der steuerbaren Funktionen zu erstellen.

Das Fehlen eines konzeptionellen, mentalen Modells führt zwangsläufig zu Fehlbedienungen, bedingt durch fehlendes Verständnis für die Funktionsweise des zu bedienenden Objekts. Man denke an das Konzept der Black-Box, wie es bei vielen elektronischen Geräten zwangsläufig zum Einsatz kommt: die Arbeitsweise solcher Geräte ist für Nicht-Techniker oft nicht nachvollziehbar. Der normale Benutzer sieht sich einem Kasten gegenüber, welcher auf „magische“ Weise die gewünschten Aufgaben ausführt. Wie dies im einzelnen geschieht, kann nicht durch Exploration der Form (haptisch oder visuell) ergründet werden. Steht nur die schriftliche Bedienungsanleitung zur Verfügung, wird das konzeptionelle Modell allein durch diese Informationen erstellt, was oftmals die Verwirrung vervollständigt. Stimmt dann die Beschreibung auch nicht mit der tatsächlichen Implementierung überein, oder wird nicht die gesamte Beschreibung gelesen, kann die Bedienung nicht mehr korrekt erfolgen. Das größte Problem ist jedoch, dass ohne ein generelles Verständnis für ein Objekt dessen Benutzung nicht generalisiert werden kann, da für nicht-triviale Situationen kein Lösungsweg aus der mentalen Repräsentation des Objekts abgeleitet werden kann.

Das mentale Modell steuert dabei auch die Bewegungsabläufe während der Benutzung von Geräten. Die Computertastatur besitzt knapp über 30 verschiedene Tasten, welche ständig für Eingaben verwendet werden. Ein Anfänger ist ca. 100-mal langsamer als ein erfahrener Benutzer, unter der Voraussetzung, dass der Anfänger 1-3 Sekunden pro Buchstabe benötigt und der Weltrekord bei über 500 fehlerfreien Anschlägen pro Minute liegt. Diese Leistung ist nur mit Hilfe des mentalen Modells erreichbar, welches zielgerichtete intuitive Bewegungen der Finger ermöglicht. Taktiler Feedback

unterstützt die Bildung des mentalen Modells, indem Informationen über die Soll-Stellung der Finger für das Drücken einer bestimmten Taste geliefert werden. Dieses gespeicherte oder auch erinnerbare Konzept besteht demnach nicht nur aus visuellen Informationen, sondern genau aus den für die Benutzung und das Verständnis der Funktionsweise relevanten Informationen. Äußerliche Merkmale wie die Höhe und Breite, die Farbe oder das Gewicht der Tastatur spielen in diesem speziellen Fall keine Rolle. Die Integration in unser Gedächtnis geht so weit, dass erfahrene „Tipper“ auf Kommando nicht sagen können, an welcher Stelle sich die Buchstaben befinden, beim Schreiben jedoch häufig benutzte Worte und Zeichenfolgen mit einer hohen Zahl von Anschlägen schreiben können.

2.4.2.7 Mapping

Im Sinne des Designs von Schnittstellen beschreibt der Begriff Mapping (Abbildung) die Beziehung zwischen den Bewegungen des Kontrollgeräts und deren Auswirkungen. Sinnvollerweise entspricht die Bewegung des Steuergeräts der Bewegung oder Aufgabe des zu steuernden Objekts. Eine seitliche Bewegung eines Objekts ist demnach durch eine seitliche Bewegung einer Kontrolleinheit am sinnvollsten umzusetzen. Bei räumlichen Bewegungen ist dies noch relativ einfach realisierbar. Problematisch wird es, sobald entweder das gesteuerte Objekt nur wenige Zustände hat, wie eine Lampe, oder die auszuführende Aufgabe von abstrakter Natur ist, so dass es keine natürliche Steuerung dafür geben kann. Eine Reihe von Lichtschaltern lässt normalerweise keinen Rückschluss darauf zu, welche Lampe mit welchem Schalter gesteuert wird. Ein Taschenrechner kann Zahlen addieren, eine Aufgabe, für die es keine natürliche Steuerung gibt, da es für sie keine Analogie in der realen Welt geben kann. Lichtschalter hingegen können so angeordnet werden, dass ihre Platzierung auf einer Montageplatte relativ zur Position der Lampen im Raum ist. Diese Montageplatte kann zusätzlich noch horizontal angebracht werden. So können Assoziationen zwischen der Steuerung und dem zugrunde liegenden Objekt geschaffen werden, welche die Bedienbarkeit vereinfachen, da ein mentales Bild geschaffen werden kann

und auch ein Mapping zwischen Steuerung und Gerät möglich ist. Dies basiert allerdings auf der logischen Einsicht, dass die Anordnung der Schalter nicht willkürlich geschehen ist. Die möglichen Hilfestellungen durch eine sinnvolle Repräsentation der Benutzungsschnittstelle sind mit Zunahme des Abstraktionsgrades der auszuführenden Aufgabe von Erfahrung und Wissen des Anwenders abhängig.

Ein weiterer Aspekt ist die Anzahl der Kontrollmöglichkeiten. Man kann Tasten mit mehreren Funktionen belegen, abhängig vom aktuellen Zustand des Geräts. Diese Tasten können zusätzlich die Dauer des Drückens registrieren und entsprechend Aktionen auslösen. Auf der anderen Seite kann man jeder Funktion eine eigene Taste zuordnen. Diese könnten durch ihr Erscheinungsbild Aufschluss über ihre Funktion geben. Leider ist durch den Trend zur Miniaturisierung auch der Platz für die Kontrollelemente knapp geworden, was die Bedienung in den meisten Fällen erschwert, da wenige Tasten kontextabhängig mit vielen Funktionen belegt sind. Ein natürliches Mapping ist hier nicht mehr möglich.

2.4.2.8 Feedback

Da elektronische Geräte nicht bei Ihrer Arbeit beobachtet werden können, fällt es Benutzern oft schwer, sich ein mentales Bild zu erzeugen. Durch gezieltes Feedback auf Aktionen des Benutzers kann jedoch der aktuelle Zustand des Geräts vermittelt werden. Dies kann ein simpler Ton beim Drücken einer Taste sein oder gar die permanente (und immer aktuelle) Anzeige des Zustands per Grafik-Display. Abbildung 7 verdeutlicht dieses Prinzip. Der Zustand von Akku, Signalgüte, Datum und die aktuelle Belegung der oberen Tasten ist immer zu erkennen. Aktionen des Benutzers lösen in fast allen Fällen visuelles Feedback aus. Zusätzlich kann jeder Tastendruck mit einem Ton quittiert werden. Im Übrigen sind Handys gute Beispiele für schlechtes Mapping, da jede Taste mit mehreren Funktionen belegt ist. Wer schon den Short-Message-Service (SMS) benutzt hat, konnte feststellen, wie ermüdend das Suchen der Buchstaben auf der Tastatur ist und dass sich die

Zeit für das Schreiben eines Texts um einen Faktor größer als zwei erhöht. Zum einen müssen die Tasten mehrfach gedrückt werden, bis der gesuchte Buchstabe erscheint, zum anderen werden SMS nur einhändig geschrieben. Zur ausführlichen Bewertung von Real-World-Benutzungsschnittstellen verweise ich auf Kapitel 3.



Abbildung 7: Siemens S45i im Standby-Betrieb

2.4.2.9 Human Action Cycle

Menschliches Verhalten kann auf verschiedene Weisen im Modell angenähert werden. Die meisten Ansätze gehen von der Vorstellung aus, dass unser Gehirn ähnlich wie eine Computer-Simulation funktioniert: in einer Art Informationsverarbeitungskreislauf. In [12] wird das Modell des Human Action Cycle definiert (Abbildung 8). Demnach werden sensorische Wahrnehmungen nach dem Empfang interpretiert, was wie der gesamte kognitive Prozess von den Erfahrungen und dem Wissen des Menschen abhängig ist. Die Interpretation der Welt zum aktuellen Zeitpunkt wird

bewertet, wobei objektiv gleiche Situationen von verschiedenen Menschen völlig unterschiedlich bewertet werden können (siehe [13]). Aus der subjektiven Sicht der Dinge werden neue Ziele (goals) formuliert, für deren Erreichen Aktionen notwendig sind, welche ausgeführt werden und den Einfluss des Menschen auf die Welt verkörpern.

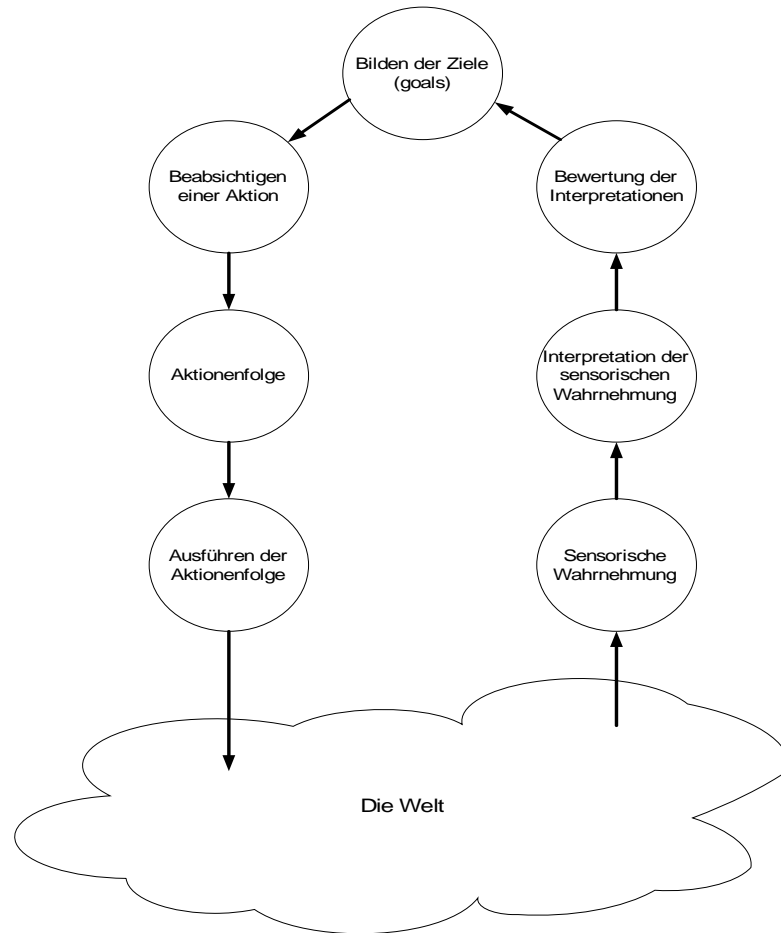


Abbildung 8: Human Action Cycle

Der Ablauf ist jedoch nicht als strikt sequentiell anzusehen. Es ist kein fester Zeitpunkt für die einzelnen Schritte des Modells nachweisbar, auch beschreibt das Modell nicht vollständig den Prozess der Kognition. Es ist vielmehr ein Prozess einer kontinuierlichen Informationsaufnahme und Verarbeitung, welcher im ganzen als Kreislauf angesehen werden kann, ohne zeitlichen Rahmen.

2.4.2.10 Beschränken der möglichen Alternativen: Constraints

Durch Beschränkung der möglichen Alternativen bei der Bedienung eines Geräts kann die kognitive Last deutlich reduziert werden, da kein oder nur ein minimaler Entscheidungsprozess durchlaufen werden muss. Ideal wäre die Beschränkung auf eine Alternative, was in der Praxis jedoch kaum möglich sein wird. Durch die Auswahl und Bereitstellung von Formen und Materialien mit der Aufgabe entsprechenden Affordances und Beschränkungen (constraints) kann der Benutzer jedoch in die gewünschte Richtung gelenkt werden.



Abbildung 9: „Babys erste Bausteine“ von Fisher-Price

Für den Bereich des Designs von virtuellen Objekteigenschaften kann man vier Klassen von constraints unterscheiden:

- Physikalisch: diese constraints basieren nur auf der wahrgenommenen Umgebung und sind nicht vom Wissen des Benutzers abhängig. Ein Gegenstand, welcher nicht getragen oder gehoben werden soll, kann festgeschraubt werden, der Benutzer kann die Aktion „Heben“ nicht

durchführen. Man trifft in seiner Umgebung ständig auf diese Art der Beschränkung: der Schlüssel passt nur in eine Richtung in das Schlüsselloch, eine Taste lässt sich nur drücken. Abbildung 9 zeigt eine Beschränkung der möglichen Alternativen durch die Form: das Runde passt nicht in das Eckige!

- Semantisch: nicht jede mögliche Aktion macht in jeder Situation Sinn. Ein Bild wird meistens an die Wand gehängt und nicht an das Fenster, eine heiße Pfanne stellt man besser auf einen Untersetzer und nicht auf den lackierten Holztisch. In der realen Welt sind semantische Constraints vom Wissen des Menschen abhängig, in einer virtuellen Welt können auch Objekte mit Wissen über sich selbst ausgestattet werden, was den Benutzer einer Virtual-Reality-Anwendung sehr gut unterstützen kann (zumindest in naher Zukunft), da Fehlbedienungen in gewissem Maß vom Objekt vermieden oder verhindert werden können.
- Logisch: Ein Kreuzworträtsel hat für manche Fragen mehrere richtige Antworten. Ist in einer kreuzenden Folge bereits eine Antwort eingetragen, wird die Anzahl der möglichen Antworten für die Frage mit mehreren Lösungen logisch beschränkt. Logisch sind auch die oben genannten Zusammenhänge beim Mapping. Diese Beschränkungen sind in hohem Maß vom Wissen des Menschen abhängig, da Entscheidungen allein aus dem bereits erworbenem Wissen über die Welt getroffen werden.
- Kulturell: Wenn es mehr als eine sinnvolle Alternative trotz physikalischer, semantischer und logischer Beschränkungen gibt, können kulturelle Erfahrungen des Menschen die Entscheidung beeinflussen. Dies gilt auch für Menschen mit verschiedenen Berufen. So kann für einen Techniker die Farbe Grün bedeuten, dass alles in Ordnung ist, für einen Mediziner kann es ein Warnsignal sein.

2.5 Lernen aus der realen Welt

Da das Ziel dieser Arbeit die Entwicklung natürlich bedienbarer Schnittstellen ist, kann auf die Erfahrungen und Entwicklungen von Real-World-Oberflächen zurückgegriffen werden. Im folgenden findet sich eine Übersicht der verschiedenen Steuerelemente und entsprechenden Greifarten ([26]).

2.5.1 Greifarten

Man kann menschliche Greifarten in drei Kategorien einteilen (Abbildung 10):

- Kontakt-Griff
- Zufassungs-Griff
- Umfassungs-Griff


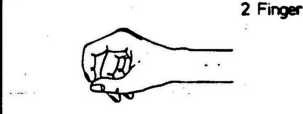
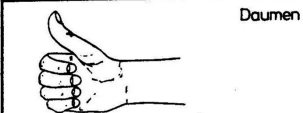

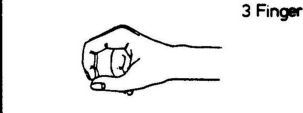

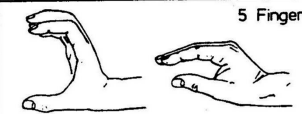
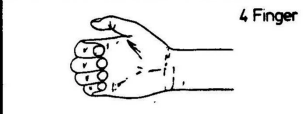
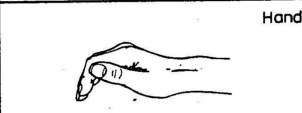
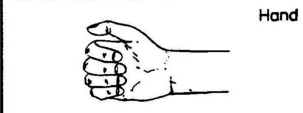
Kontakt - Griff	Zufassungs - Griff	Umfassungs - Griff
 <p>1 Finger</p>	 <p>2 Finger Daumen gegenübergestellt Daumen quer gestellt</p>	 <p>2 Finger</p>
 <p>Daumen</p>	 <p>3 Finger gleichverteilt Daumen gegenübergestellt</p>	 <p>3 Finger</p>
 <p>Hand</p>	 <p>5 Finger gleichverteilt Daumen gegenübergestellt</p>	 <p>4 Finger</p>
 <p>Handkamm</p>	 <p>Hand</p>	 <p>Hand</p>

Abbildung 10: Systematik der Greifarten

Abbildung 11 beschreibt die Eignung der Greifarten in bezug auf Genauigkeit, Zeitbedarf und der Größe der übertragbaren Kraft.

Arbeitsaufgabe Greifartgruppe	großer Arbeitswiderstand	kleiner Zeitbedarf	große Genauigkeit
Kontaktgriff	○	○ ○ ○	○ ○
Zufassunggriff	○ ○	○ ○	○ ○ ○
Umfassunggriff	○ ○ ○	○	○

○ ○ ○ zu bevorzugende Greifartgruppe

Abbildung 11: Gegenüberstellung von Greifarten und Eignung

Für große Arbeitswiderstände ist eine Eignungszunahme vom Kontaktgriff zum Umfassunggriff zu erkennen. Der Zufassunggriff erlaubt willkürliche Grifflagen innerhalb der Hand und ist daher unter dem Gesichtspunkt der Arbeitsqualität und Genauigkeit am sinnvollsten.

Die Abbildungen 12 und 13 zeigen formschlüssige und reibschlüssige Kopplungen für die Greifartgruppen. Formschlüssig ist ein Schluss zwischen zwei sich berührenden Wirkflächen fester Körper senkrecht zu den Berührungsflächen. Reibschlüssig ist ein Schluss, wenn Tangentialkräfte in allen Richtungen einer gemeinsamen Berührungsebene übertragen werden.

Kopplungsart: formschlüssig						
		Translation			Rotation	
		Zug	Druck	Zug-Druck	Drehwinkel < 180°	Drehwinkel > 180°
Kontakt-Griff	Finger					
	Hand					
Zufassungs-Griff	Finger					
	Hand					
Umfassungs-Griff	Finger					
	Hand					

Abbildung 12: Zusammenstellung der Grundformen für die Greifartgruppen bei formschlüssiger Kopplung

Kopplungsart: reibschlüssig						
		Translation			Rotation	
		Zug	Druck	Zug - Druck	Drehwinkel < 180°	Drehwinkel ≥ 180°
Kontakt-Griff	Finger					
	Hand					
Zufassungs-Griff	Finger					
	Hand					
Umfassungs-Griff	Finger					
	Hand					

Abbildung 13: Zusammenstellung der Grundformen für die Greifartgruppen bei reibschlüssiger Kopplung

2.5.2 Stellteile

Stellteile sind definiert als „Elemente an Arbeitsmitteln, die, durch menschliche Kraft bewegt, eine Veränderung des Informations-, Energie- und/oder Stoffflusses, bzw. einer Position bewirken. Sie können auch der Lageeinstellung anderer Bauteile dienen.“ ([26]). Die hier gezeigte Einteilung der Stellteile orientiert sich an den in der Praxis gebräuchlichsten Bewegungsformen Drehen, Schwenken, Ziehen, Schieben, Drücken (Abbildungen 14-17).

Drehen des Stellteils

Einsatzmöglichkeit von Stellteilen

Greifart	Kopplung			Lage der Stellnuten
	formschlüssig	reibschlüssig	Beispiele	
1 Finger Kontakt	formschlüssig	reibschlüssig		
	formschlüssig	reibschlüssig		
2 Finger Zufassung	formschlüssig	reibschlüssig		
	formschlüssig	reibschlüssig		
3 Finger Zufassung	formschlüssig	reibschlüssig		
	formschlüssig	reibschlüssig		
Hand Zufassung	formschlüssig	reibschlüssig		
	formschlüssig	reibschlüssig		
Hand Umfassung	formschlüssig	reibschlüssig		
	formschlüssig	reibschlüssig		

Abbildung 14: Einsatzmöglichkeiten von Stellteilen zum Drehen

Schwenken des Stellteils Einsatzmöglichkeit von Stellteilen

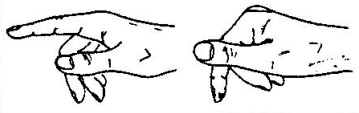













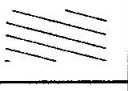

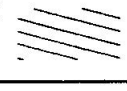

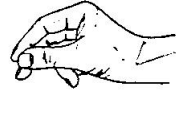
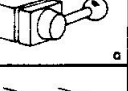
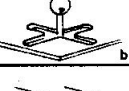



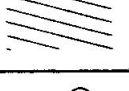
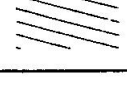

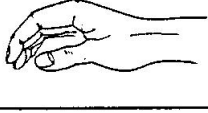
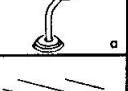
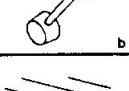


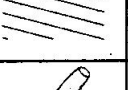

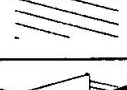

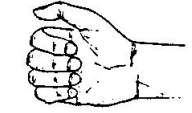

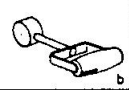
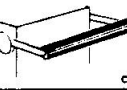


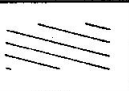


Greifart	Kopplung			Lage der Stellbochse	
	formschlüssig	Beispiele	reibschlüssig		
1 Finger Kontakt 	formschlüssig	 a	 b	 c	
	reibschlüssig	 a	 b	 c	
2 Finger Zufassung 	formschlüssig	 a	 b	 c	
	reibschlüssig	 a	 b	 c	
3 Finger Zufassung 	formschlüssig	 a	 b	 c	
	reibschlüssig	 a	 b	 c	
Hand Zufassung 	formschlüssig	 a	 b	 c	
	reibschlüssig	 a	 b	 c	
Hand Umfassung 	formschlüssig	 a	 b	 c	
	reibschlüssig	 a	 b	 c	

Abbildung 15: Einsatzmöglichkeiten von Stellteilen zum Schwenken

Ziehen - Schieben des Stellteils

Einsatzmöglichkeit von Stellteilen

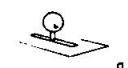







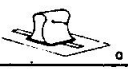

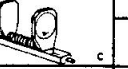










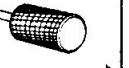















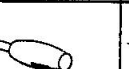


Greifart	Kopplung	Beispiele			Lage der Stellflachse
1 Finger Kontakt	formschlüssig				
	reibschlüssig				
2 Finger Zufassung	formschlüssig				
	reibschlüssig				
3 Finger Zufassung	formschlüssig				
	reibschlüssig				
Hand Zufassung	formschlüssig				
	reibschlüssig				
Hand Umfassung	formschlüssig				
	reibschlüssig				

Abbildung 16: Einsatzmöglichkeiten von Stellteilen zum Ziehen und Schieben

Drücken des Stellteils

Einsatzmöglichkeit von Stellteilen

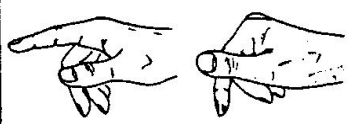


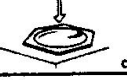

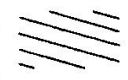








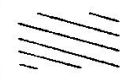

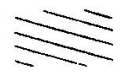

Greifart	Kopplung	Beispiele			Lage der Stellflachse
<p>1 Finger Kontakt</p> 	formschlüssig				
	reibschlüssig				
<p>Hand Kontakt</p> 	formschlüssig				
	reibschlüssig				

Abbildung 17: Einsatzmöglichkeiten von Stellteilen zum Drücken

Kapitel 3

3 Stand der Technik

Ich möchte in diesem Kapitel den Weg von 2D-WIMP bis zu aktuellen 3D-Benutzeroberflächen darstellen, inklusive einem Blick auf aktuelle Eingabegeräte und deren Fähigkeiten und Einsatzgebiete. Ebenso betrachte ich thematisch passende Real-World Schnittstellen und bewerte sie in bezug auf Benutzbarkeit.

3.1 2D-Benutzungsschnittstellen

Die ersten Benutzer von Benutzungsoberflächen sahen sich einem leeren Bildschirm gegenüber. Der einzige Hinweis auf die nächste auszuführende Aufgabe war ein blinkender Cursor nach dem Prompt.

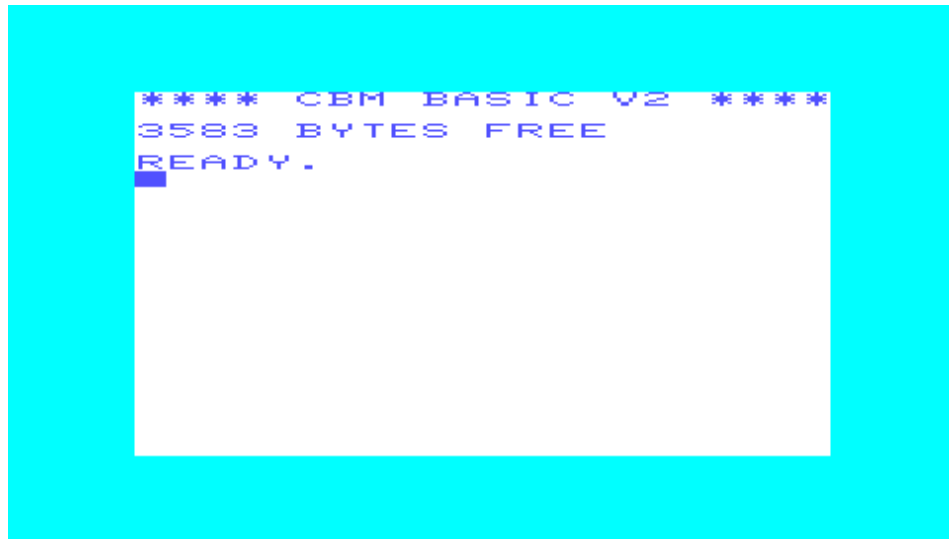


Abbildung 18: Commodore VC20-Prompt.
Ready – Aber wofür?

3.1.1 WIMP: Windows, Icons, Mouse, Pointer

Eine Lösung dieses Problems musste her. Es entstanden die ersten graphischen Benutzungsschnittstellen, kurz GUI. Die erste kommerzielle Verbreitung dieser Technik fand durch den Personal Computer LISA von Apple 1983 (Abbildung 19) statt, angelehnt an die Technik von Xerox.

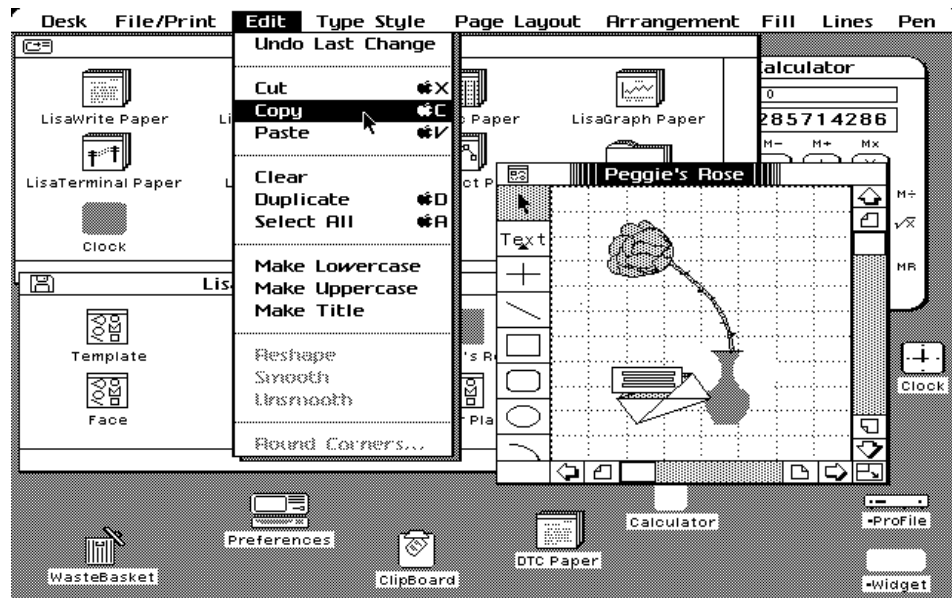


Abbildung 19: Apples GUI

Die Bedienung erfolgt mit der Maus als Eingabegerät. In verschiedenen Fenstern können verschiedene Anwendungen laufen, jederzeit durch den Benutzer nachvollziehbar. Dieser Ansatz stellt einen enormen Zuwachs der Kommunikationsfähigkeit des Computers mit dem Benutzer dar, weil die Bandbreite des visuellen Kanals sehr viel stärker ausgenutzt werden kann. Neu ist ebenfalls der Einsatz von Ikonen, welche die Art einer Anwendung oder einer Aufgabe sehr viel besser beschreiben können als eine reine Textinformation. Zusätzlich müssen Befehle nicht mehr auswendig gelernt werden, da ein weiteres Konzept eingebaut ist: Menüs. Zwar war diese Technik nicht mehr neu, unterstützt die Bedienung einer Anwendung jedoch ungemein. Fenster und Menüs wurden zum Bestandteil des Betriebssystems und der Benutzer musste sich nicht mehr an verschiedene Menüstrukturen von verschiedenen Anwendungen anpassen. Es gab einen übergreifenden Zugang zu jeder Anwendung.

Als Eingabemetapher wurde ein Pfeil-Symbol gewählt, welches das Zeigen auf Objekte im Anzeigebereich symbolisieren soll. Für das Klicken von 2D-Tasten und einfache Auswahlaufgaben ist dies eine gute Methode, da für die

eigentliche Aktion, das Positionieren und Klicken, ein sehr gutes Mapping möglich ist. Die Bedienung in einer anderen Ebene als der Anzeige abgesehen ist der einzige Schwachpunkt. Zudem erleichtert der Einsatz von Maus Zeiger die Einarbeitungszeit für Anfänger enorm, da die Maus relativ einfach zu bedienen ist..

Trotz aller Vorteile gegenüber der Kommandozeile bedarf diese Art von Schnittstelle einiger Gewohnheit und einem Mindestmaß an Verständnis für die zugrunde liegenden Aufgaben. Ein generelles Problem von Computern ist die Undurchschaubarkeit der internen Prozesse für den unerfahrenen Benutzer. Der Klick auf einen Schalter und die folgende Reaktion des Computers sind nicht immer leicht in Zusammenhang zu bringen und schrecken viele Leute ab. Die Führung des Benutzers durch Menüs ist eine fehlerträchtige Angelegenheit, da der Benutzer den Überlegungen des Entwicklers ausgeliefert ist. Da Windows und somit die zugehörigen Menüs keine feste Position und nicht einmal eine feste Größe haben, kann sich der Benutzer nur mühsam ein konzeptionelles Modell des Systems bilden. Desweiteren stehen gleichnamige Menüeinträge in verschiedenen Anwendungen für verschiedene Aktionen, alle undurchschaubar und von abstrakter Natur. Sogar erfahrene Informatiker finden sich nicht selbstverständlich zurecht. Dem Anspruch der Determiniertheit von Benutzungselementen wird WIMP nicht gerecht.

Der Einsatz von Metaphern verstärkt die vorhandene Verwirrung zusätzlich, da durch die Namensgebung und das Aussehen der Metapher eine Erwartungshaltung geweckt wird, welche jedoch die Maus und der Zeiger als Eingabegeräte nicht bedienen können. Ein Beispiel: während der Informatiker sich über die Verwahrung alter Dateien im Papierkorb freut, ist für den unerfahrenen Benutzer schon das abstrakte Konzept der Datei nicht zu verstehen. Ganz zu schweigen von der Speicherung von Dateien im Papierkorb, der mit einem Klick geöffnet und mit einem Klick geleert wird.

Damit sind wir bei der diskreten Methode der Bedienung: ein Klick reicht zum Auslösen einer Aktion, ganz gleich ob Formatieren der Festplatte oder Wechseln zum nächsten Fenster, wodurch Fehlbedienungen immer wieder vorkommen. In erster Linie liegt dies an der schwierig positionierbaren Maus und den eng beieinander platzierten Buttons. In zweiter Linie werden die Verhaltensweisen des Menschen außer Acht gelassen. Der Mensch generalisiert und liest nicht jede Warnung in jedem Fenster. Erscheint beispielsweise beim Surfen im Internet immer wieder die Frage, ob man ein Cookie zulassen möchte, wird man auf vertrauenswürdigen Sites ganz automatisch akzeptieren. Erscheint nun anstelle der Cookie-Warnung eine andere, so werden die meisten aus Gewohnheit ebenfalls akzeptieren. Eine gute Schnittstelle muss solche Fehlbedienungen von Grund auf verhindern, denn letztendlich soll sich der Computer dem Menschen anpassen und nicht umgekehrt.

Bei dieser diskreten Art der Steuerung sind noch weitere Fehlbedienungen zu bemängeln. Das Benutzen eines Werkzeugs in der realen Welt ist ein kontinuierlicher Prozess, über den man in den meisten Fällen die Kontrolle hat, sei es das Mischen von Farben oder das Drehen einer Schraube. Bei WIMP-Systemen ist Einflussname auf ausgelöste Aktionen oft nicht möglich und falls doch, ist es immer von der Arbeit der Entwickler abhängig, ein Logbuch der durchgeführten Aktionen zu führen, um diese rückgängig machen zu können. Von der Tatsache abgesehen, dass diese Methoden oft ihre Fehler haben, ist es doch eine Bekämpfung der Symptome. Auch die manchmal unausweichlichen „Klickorgien“ sind sehr konzentrationsstörend. Der Benutzer wird von der eigentlichen Aufgabe abgelenkt, weil die Ausrichtung des Zeigers explizit erfolgt. Die Bewegungen des Zeigers sind so ungenau, dass keine kognitive Karte der Oberfläche aufgebaut werden kann, welche eine intuitive Benutzung zulassen würde.

Auch das Feedback des Systems ist unzureichend. Zwar kann durch die bessere Ausnutzung der Bandbreite des visuellen Kanals deutlich mehr

Information vermittelt werden, das Feedback beschränkt sich im besten Fall aber auf einen Fortschrittsbalken, dessen Verweilen in einer bestimmten Position jedoch nicht bedeutet, dass keine Berechnungen mehr im Hintergrund stattfinden. Die meisten Aktionen liefern nur Feedback, indem ein Ergebnis präsentiert wird. Man erfährt nicht, ob die richtige Taste geklickt wurde, ob die Aktion rückgängig gemacht werden kann oder ob die Aktion noch gar nicht beendet ist und man nur ein Zwischenergebnis präsentiert bekommt.

Doch vergessen werden darf nicht, dass es sich hier um ein 2D-System handelt, für dessen Bedienung die Freiheitsgrade der Hand reduziert werden müssen auf die beschränkten Möglichkeiten des Window-Systems, erst durch die Maus, dann durch den Zeiger. Auch der Einsatz von Ikonen und Farben ist von großer Bedeutung, zumindest für erfahrene und unerschrockene Benutzer eines solchen Systems, da sehr viel Information durch gute Visualisierungen auf einen Blick vermittelt werden kann. Speziell die Möglichkeit der Gruppierung von Information trägt zur Verständlichkeit der Oberfläche bei. Die Performanz ist mit dieser Art von Schnittstelle deutlich höher und die Einarbeitungszeit für neue Benutzer ist deutlich geringer im Vergleich zur vorangegangenen Eingabeaufforderung, sofern die Anzahl der auf dem Bildschirm dargestellten Elemente den Benutzer nicht überfordert (siehe Abbildung 20). Je mehr Elemente dargestellt werden, umso weniger unterscheiden sich diese, was gegen die Forderung der Determiniertheit von Benutzungselementen spricht. Auch kann nicht von einem Gefühl der Immersion gesprochen werden.

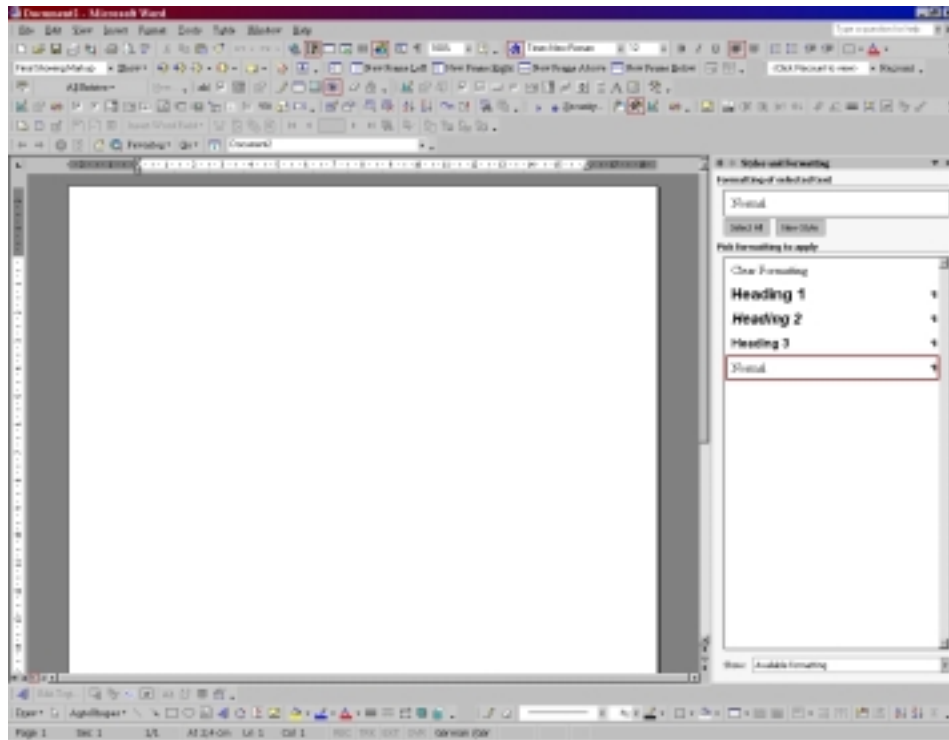


Abbildung 20: bekannte Textverarbeitung mit allen Toolbars, der Benutzer kann sich nicht mehr orientieren

3.1.2 3D-Anwendungen im 2D-Arbeitsbereich

Im Verlauf der Entwicklung der Computertechnologie und der rasanten Erhöhung der Rechengeschwindigkeit von Computern, welche in manchen Bereichen sogar Moore's Gesetz übertrifft, wurde es sehr schnell möglich, echte 3D-Inhalte zum Bestandteil von Anwendungen zu machen. Es gab wohl Entwicklungen, welche die Window-Oberflächen an die Anforderungen des Feedback und der Benutzbarkeit anpassen sollten (3D-Tasten, andere Formen des Zeigers), das eigentliche Problem des nicht natürlichen Mappings bei der Bedienung von 3D-Anwendungen konnte so aber nicht gelöst werden. Erst der Einsatz von Widgets war wieder ein Schritt in Richtung einer direkteren Manipulation.

3D-Widgets sind abstrakte graphische Elemente, welche direkt mit Objekten einer Szene in Verbindung stehen oder sie erweitern ([8]). Beispielsweise kann ein Objekt von einem Ring umgeben sein, um eine Rotation zu erlauben (siehe Abbildung 21). Ein Berühren des Rings mit dem Mauszeiger und anschließende Bewegung der Maus rotiert den Würfel um die zur Ebene des Rings orthogonale Achse.



Abbildung 21: Widget zur Unterstützung einer Rotation um eine Achse

Ein Nachteil dieser Technik besteht darin, dass die Objekte nicht mehr real erscheinen. Allerdings können Widgets sehr gut an die Eingabehardware angepasst werden, so dass alle gewünschten Aufgaben erfüllt werden können, wobei die Forderung an ein möglichst natürliches Mapping gut bedient werden kann. Widgets müssen jedoch meistens über zusätzliche Menüs oder Toolbars ausgewählt werden, was wieder eine Reduktion der Performanz darstellt. Auch ist die Funktionalität nicht immer aus der Form erkennbar. Ohne Widgets wären jedoch die meisten 3D-Anwendungen kaum noch bedienbar, da sonst eine Manipulation von Objekten nur sehr eingeschränkt möglich wäre (mit der Maus als Eingabegerät).

3.2 3D-Benutzungsschnittstellen

Der Schritt von 2D-Schnittstellen in Richtung 3D ist nur mit Hilfe von Eingabegeräten zu bewerkstelligen, welche viele Freiheitsgrade besitzen. Diese erlauben aber auch eine direkte oder besser kontrollierbare Interaktion mit den virtuellen Objekten, so dass weniger Notwendigkeit für Menüs, Toolbars oder Kontext-Menüs besteht. Die Window-Umgebung verliert so an Bedeutung. Ich möchte einige der aktuellen Geräte an dieser Stelle beschreiben.

3.2.1 Maus-basierte 6DOF-Eingabegeräte

Die hier beschriebenen Geräte basieren auf der Idee eines einfachen Geräts für die Interaktion mit dem Computer und sind für Steuerungen von 3D-Anwendungen konzipiert.

3.2.1.1 Cubic Mouse

Die Cubic Mouse bietet in ihrer höchsten Ausbaustufe 12 Freiheitsgrade: das Gerät kapselt einen Positions-Tracker mit 6 Freiheitsgraden. Durch das Gehäuse hindurch ist für jede Raumachse ein Stab montiert, jeweils senkrecht zu den Gehäuseseiten. Diese Stäbe können gedrückt und geschoben werden, zusätzlich ist auch die Rotation der Stäbe um die Längsachse möglich. An den Enden der Stäbe und auf einer Seite des Gehäuses befinden sich zudem Tasten für weitere Eingaben.

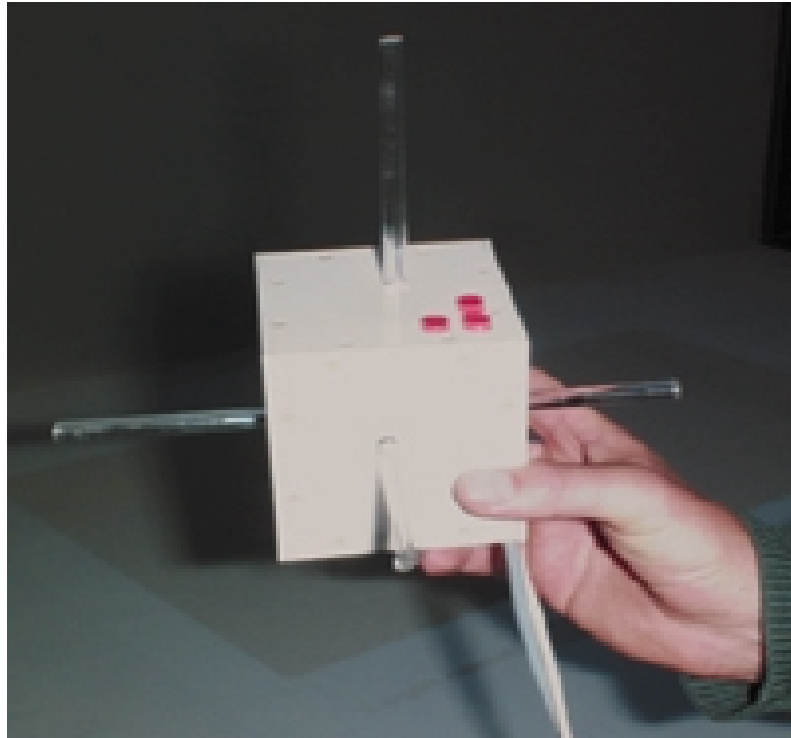


Abbildung 22: Cubic Mouse

Das Prinzip der Benutzung ist interessant: die Cubic Mouse repräsentiert die Referenz eines virtuellen Objekts. Dieses Objekt wird somit vom Benutzer in seinen Händen gehalten. Manipulationen finden auf direktem Weg statt. Die virtuellen Objekte sollten jedoch in ihren elementaren Eigenschaften der kubischen Form der Maus entsprechen, beispielsweise über sechs Seiten verfügen. Ein eingebauter Tracker wird zur Positionierung des Objekts im Raum eingesetzt, die Tasten auf der oberen Seite erlauben die genaue Steuerung der Bewegungen, wie das Einfrieren des Objekts an einer Zielposition. Diese Art der Eingabe liefert nebenbei auch kontinuierliches taktiler Feedback (man hält die Maus ständig in der Hand), was dem Benutzer ermöglicht, das Gerät ohne Sichtkontakt zu benutzen, wodurch die Performanz deutlich erhöht wird.

Unvermeidlich ist jedoch, dass solch ein Gerät nur für bestimmte Aufgaben geeignet ist. In diesem Fall sind es vor allem die Operationen Translation und

Rotation. Auch das Wechseln des Fokus vom aktuellen zu einem anderen Objekt kann Schwierigkeiten bereiten, da die Position der Maus auch die Position des kontrollierten Objekts darstellt. Clutching⁶ ist die Folge. Die Art der Objektselektion ist leider nicht in den Veröffentlichungen beschrieben, vermutlich gestaltet sich auch dies schwierig. Die herkömmlichen Techniken wie Picking sind mit der Cubic Mouse auch nur durch den Einsatz weiterer Metaphern zu verwirklichen.

3.2.1.2 Magellan/Logicad Spacemouse

Dieses Gerät verfolgt einen anderen Weg als die eben erwähnte Cubic Mouse. Die Spacemouse orientiert sich an der herkömmlichen, weit verbreiteten Maus in Form und Erscheinung. Sie steht fest an einer Stelle, manipuliert werden Objekte durch eine Steuereinheit, welche in der Mitte der Grundplatte montiert ist und sechs Freiheitsgrade bietet. Die Steuereinheit erlaubt ein feinfühliges Bestimmen der Rotations- und Translationsgeschwindigkeit der manipulierten Objekte. Zusätzlich stehen elf Tasten für weitere Eingaben zur Verfügung.

Auch hier dient der Steuerknopf als Referenz des zu manipulierenden Objekts. Nachteilig ist, dass keine Direktmanipulation wie im Fall der Cubic Mouse möglich ist, weil das Gerät statisch an einer Stelle steht. Es entsteht die oben erwähnte Kluft zwischen Gesehenem und Gefühltem. In [10] wurde die Performanz von Spacemouse, herkömmlicher Maus und einem Eigenbau verglichen. Der Eigenbau bestand aus einem Positions-Tracker, kombiniert mit einer Taste, kurz Tracked Button.

⁶ Von Clutching spricht man, wenn eine Maus über eine grössere Distanz bewegt werden soll, jedoch nicht genügend Platz auf der Arbeitsfläche zur Verfügung steht. Das Gerät muss in diesem Fall neu positioniert werden, bevor die Bewegung fortgesetzt werden kann.



Abbildung 23: Magellan/Logicad Spacemouse

Getestet wurde dabei die Effizienz, mit der Objekte positioniert werden können, wie genau diese Positionen erreicht werden kann und wie zufrieden der Benutzer mit der Bedienung des Geräts ist. Die gemessenen Zeiten führten zu der Erkenntnis, dass die Spacemouse nur etwa 20 Prozent schneller zu bedienen war als die herkömmliche Maus, bei der einfachen Translation einer Sphäre der herkömmlichen Maus sogar unterlegen war. Der Tracked Button hingegen erlaubte eine im Durchschnitt ca. 3-fach schnellere Ausführung der Aufgaben. Besonders interessant bei diesem Versuch ist, dass beim Tracked Button die Zeit für die einfache Translation der Sphäre und der kombinierten Rotation und Translation eines Konus fast gleich war. Die direkte Manipulation der Objekte brachte demnach eine riesige Erhöhung der Performanz mit sich.

Die Spacemouse ist aufgrund ihres statischen Auftretens wieder von guten Eingabemetaphern abhängig und somit keine große Bereicherung für den

Markt der Eingabegeräte, wenngleich einzelne Aufgaben durch den Einsatz dieses Geräts etwas schneller bearbeitet werden können. Vorteilhaft ist sicher das Ausbleiben des oben erwähnten Clutching-Effekts. Allerdings sind seit einiger Zeit Mäuse mit Trackball erhältlich, bei denen Clutching aufgrund der Bedienung mit dem Daumen fast nicht mehr wahrgenommen wird.

3.2.2 Geräte mit haptischem Feedback

Die Entwicklung von Force-Feedback Geräten ist eine schwierige Aufgabe, da entweder das Gerät auf einem Tisch steht, immobil und somit nur eingeschränkt nutzbar ist, oder es wird vom Benutzer getragen, was zum heutigen Stand der Technik ebenfalls problematisch ist, da die Aktoren sehr robust und entsprechend massiv gebaut sein müssen, weswegen sie nicht einfach an die Hand platziert werden können. Einige Geräte haben es trotzdem zu kommerzieller Verbreitung gebracht, insbesondere im Bereich der Computerspiele ist diese Technik kaum noch wegzudenken.

Ein gutes Feedback-Gerät sollte für jeden Freiheitsgrad, welcher dem Benutzer durch das gewählte Eingabegerät zur Verfügung steht, einen Aktor für entsprechendes Feedback zur Verfügung stellen. Zusätzlich können weitere haptische Informationen vermittelt werden, beispielsweise in Form von Vibrationen an den Fingerspitzen zur Vermittlung taktiler Informationen der Oberflächenbeschaffenheit.

Computerspiele sind häufig durch einfache Eingaben zu steuern, vor allem wenn es um Simulationen wie Autorennen oder Flugsimulatoren geht. Reale Cockpits können sehr leicht mit Hilfe von Force-Feedback Geräten nachgebaut werden, im Fall der Spiele natürlich stark vereinfacht, aber dennoch der realen Welt sehr nah.

3.2.2.1 Logitech Driving Force

Exemplarisch für Heim-Anwendungen von Force Feedback Geräten möchte ich auf ein Lenkrad für Rennsimulationen eingehen.



Abbildung 24: Logitech Driving Force

Das Lenkrad steht in der realen Welt schon für die Aktion Lenken. Wir sehen hier also ein perfektes Mapping der Aktion für die Eingabe in die virtuelle Welt. Die Rückmeldung der Kräfte unterstützt die Wahrnehmung beim Fahren zusätzlich. Dinge wie Schlaglöcher oder Unebenheiten im Fahrbahnbelag werden oft nicht gesehen aber gespürt. Es steht somit zusätzliche Information zur Verfügung, welche ohne geistige Anstrengung verarbeitet werden kann, was die Qualität der Interaktion deutlich erhöht, außerdem ist für die Benutzung noch nicht einmal eine Beschreibung notwendig.

Da nur ein Freiheitsgrad der Eingabe der Lenkbewegung dient und das Lenkrad fest montiert ist (auch im echten Auto), kann die Feedback-Funktionalität „leicht“ realisiert werden und ist somit auch kostengünstig. Diese Schnittstelle ist ein Paradebeispiel für Einfachheit und Benutzbarkeit.

3.2.2.2 Phantom

Etwas anders ist die Sachlage bei diesem Gerät. Es besitzt sechs Freiheitsgrade für Ein- und Ausgaben und kann somit für abstrakte Aufgaben im 3D-Bereich eingesetzt werden. Das Phantom ist außerdem in verschiedenen Ausbaustufen erhältlich. Das Design mit Stift, welchen der Benutzer in der Hand führt, schränkt den Einsatzbereich auf entsprechende Aufgaben ein, um ein gutes Mapping zwischen Eingabe und Aktion zu gewährleisten. Das Phantom bietet mit seinen Fähigkeiten jedoch einen schon deutlich größeren Einsatzbereich als das oben beschriebene Lenkrad. Eingesetzt wird das Gerät beispielsweise im Bereich der Neurochirurgie, da das Führen eines Skalpells dem Bedienen des Phantoms sehr ähnlich ist und sich daher sehr gute Ergebnisse erzielen lassen.



Abbildung 25: Phantom Desktop

Ein Nachteil ist, dass der Benutzer in seinem Bewegungsradius stark eingeschränkt ist, mobile Anwendung des Geräts ist nicht möglich. Desweiteren kann das Gerät bedingt durch seine Bauweise nur schlecht für andere abstrakte Aufgaben eingesetzt werden, eine Direktmanipulation in Form von Rotation, ähnlich wie mit der Cubic-Mouse, ist nicht möglich, da der Bewegungsradius des Stifts ebenfalls eingeschränkt ist. Das Gerät funktioniert dann nicht besser als eine 6-DOF-Maus.

3.2.2.3 Immersion CyberGrasp

Das CyberGrasp Exoskelett wird in Verbindung mit einem Datenhandschuh verwendet, dem CyberGlove der gleichen Firma, und stellt dem Benutzer Feedback an allen fünf Fingern bereit. Allerdings beschränkt sich dies auf das Schließen der Hand. Das Exoskelett dient der Führung von Seilzügen, welche mit Hilfe von Schlaufen mit den Fingerkuppen verbunden werden. Die Seilzüge haben eine Länge von einem Meter und bieten somit ausreichend Bewegungsspielraum. Unter Anwendung eines rucksackgroßen Tragegestells kann das System auch mobil eingesetzt werden. Zur Verfügung steht dem Benutzer dabei ein virtuelles Bild seiner Hand, was ein direktes Eintauchen in die Szene ermöglicht.



Abbildung 26: Immersion CyberGlove und CyberGrasp

Dieses System erlaubt die Direktmanipulation von virtuellen Objekten, es ist kein Eingabegerät dazwischen geschaltet, was ein Mapping der Hand-Aktionen auf das Eingabegerät und ein Mapping der Funktionen des Eingabegeräts auf die virtuellen Objekte unnötig macht. Dieses Prinzip ist somit erste Wahl für den Einsatz in Anwendungen, welche die Vorteile und Fähigkeiten der direkten Interaktion ausnutzen wollen. Anwendungen, wie sie speziell für Phantom zum Einsatz kommen, gehören nicht dazu, weil die natürliche Benutzung der Hand auch die Ungenauigkeit von Bewegungen auf die virtuelle Repräsentation weiterleitet. Ein anderer Mangel ist das Fehlen eines Widerstandes. So kann zwar ein von den Fingern gehaltenes Objekt wahrgenommen werden, sollte ein Objekt auf ein anderes stoßen, kann jedoch kein Feedback vermittelt werden, welches diese Situation auflösen würde. Dies führt zu der noch ungünstigeren Situation, in welcher die Hand

ein Objekt durchdringt. Solche unrealistischen Eindrücke werden als äußerst störend empfunden.

Interessant wäre meiner Meinung nach die Kombination von Phantom und CyberGrasp/CyberGlove. So könnte die virtuelle Hand dafür sorgen, dass virtuelle Objekte für die Bearbeitung mit dem vom Phantom repräsentierten Werkzeug in Position gebracht oder gehalten werden.

3.2.3 Haptische Graphische Benutzungsschnittstellen

Der Aufgabenstellung entsprechend sind wiederverwendbare Schnittstellenkomponenten zu entwickeln. Die einzigen beiden Systeme, welche ebenfalls diesen Ansatz verfolgen und zudem in der Lage sind, haptisches Feedback zu vermitteln, möchte ich hier vorstellen

3.2.3.1 e-Touch

Dieses System befindet sich noch in der Entwicklungsphase, stellt aber schon einige sehr interessante Möglichkeiten zum Bau der Steuerung einer Simulation und der passenden Benutzungsschnittstelle zur Verfügung. Der Entwickler kann unter einer Vielzahl von Elementen auswählen, welche stark an die herkömmlichen Window-Oberflächen angelehnt sind. Die Bedienung kann theoretisch über beliebige Eingabegeräte mit sechs Freiheitsgraden erfolgen, da e-Touch eine offene Schnittstelle für die Integration der Hardware bereitstellt. Ausgelegt ist die Entwicklung dennoch auf das oben beschriebene Phantom.

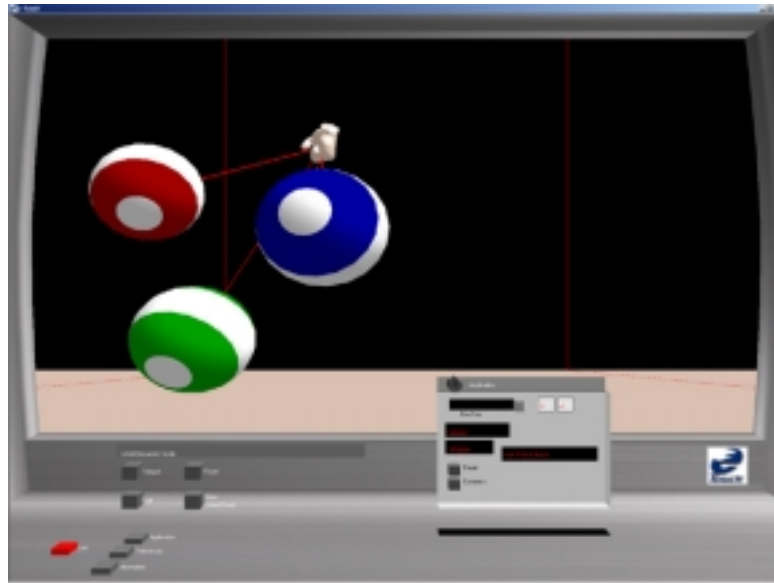


Abbildung 27: e-Touch
Benutzungsschnittstelle

Abbildung 27 zeigt eine solche Simulation. Die Transformationen der Hand-Metapher zur Steuerung der Anwendung sind relativ zu denen des Eingabegeräts.

Bei der Platzierung der Bedienelemente sind dem Entwickler keine Grenzen gesetzt. Einzige Bedingung ist die Erreichbarkeit durch den Zeiger. Ein Greifen der Elemente ist nicht möglich, lediglich die Oberfläche kann abgetastet und berührt werden. Es stehen ebenfalls Datenstrukturen zur Verfügung, welche die Implementation eigener Interaktions-Metaphern vereinfachen, so dass der Benutzer einer Anwendung nicht auf die Benutzung von Tasten beschränkt bleiben muss. Somit ist die universelle Anwendbarkeit der Komponenten sichergestellt.

Ob e-Touch die Erzeugung echter 3D-Ansichten unterstützt, konnte ich aus den Beschreibungen nicht entnehmen. Falls nicht, wird die fehlende Tiefenwahrnehmung sicher Schwierigkeiten bei der genauen Positionierung machen. Auch die Möglichkeiten zur direkten Interaktion mit Objekten beschränkt sich auf das Berühren mit der virtuellen Fingerspitze. Durch das

Aufstellen des Phantoms neben dem Bildschirm ist wahrscheinlich das Gefühl der Immersion nicht sehr ausgeprägt, da reale und virtuelle Hand sich nicht an der „gleichen“ Position befinden.

Insgesamt gesehen scheint es sich um eine wirklich gute Idee zu handeln, nur beschränkt durch die fehlende Möglichkeit zur direkten Interaktion, also dem Greifen von Objekten.

3.2.3.2 Studierstube

Das einzige mir bekannte System mit wiederverwendbaren haptischen Benutzungsoberflächenelementen für die zweihändige Interaktion ist Studierstube. Es handelt sich um ein Framework für die Entwicklung von Augmented-Reality-Anwendungen. Das speziell entwickelte Eingabegerät ist das Personal Interaction Panel (PIP, Abbildung 28). Ein Head-Mounted-Display sorgt für die Einblendung der virtuellen Objekte.

Der Benutzer hält das PIP mit der linken Hand. Die rechte Hand führt einen Stift, welcher die reale Repräsentation des virtuellen Zeigers darstellt. Mit Hilfe des Stifts werden die eingeblendeten Bedienelemente betätigt (Abbildung 29). Zur Verfügung stehen Schalter und Schieberegler.

Der Benutzer hat durch den Einsatz des Stifts die Möglichkeit zur direkten Interaktion mit virtuellen Objekten. Ebenso kann das PIP dynamisch an die aktuelle Aufgabe angepasst werden, indem kontextbezogene Bedienelemente präsentiert werden.



Abbildung 28: Personal Interaction Panel

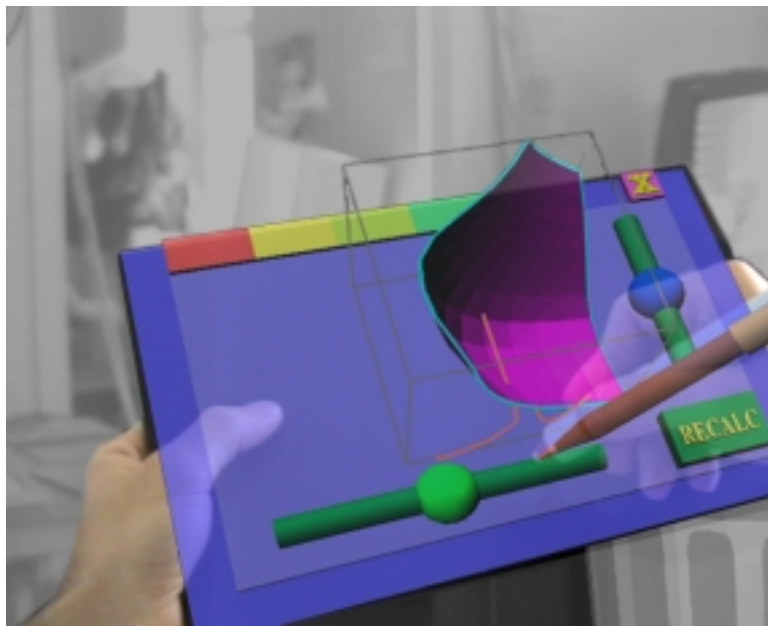


Abbildung 29: Virtuelle Erweiterung des
Personal Interaction Panels

Der Vorteil dieser Methode besteht darin, dass zumindest passives Feedback vermittelt wird. So kann der Benutzer spüren, dass er ein Bedienelement berührt hat. Da das Panel in Hand gehalten wird, kann es der Aufgabe entsprechend positioniert werden. Objekte im Fokus können durch

Bewegung des Panels oder Betätigen eines Bedienelements von allen Seiten betrachtet werden.

Desweiteren können beliebige andere Schnittstellenelemente eingefügt werden. Abbildung 30 zeigt eine Papierkorb-Metapher.

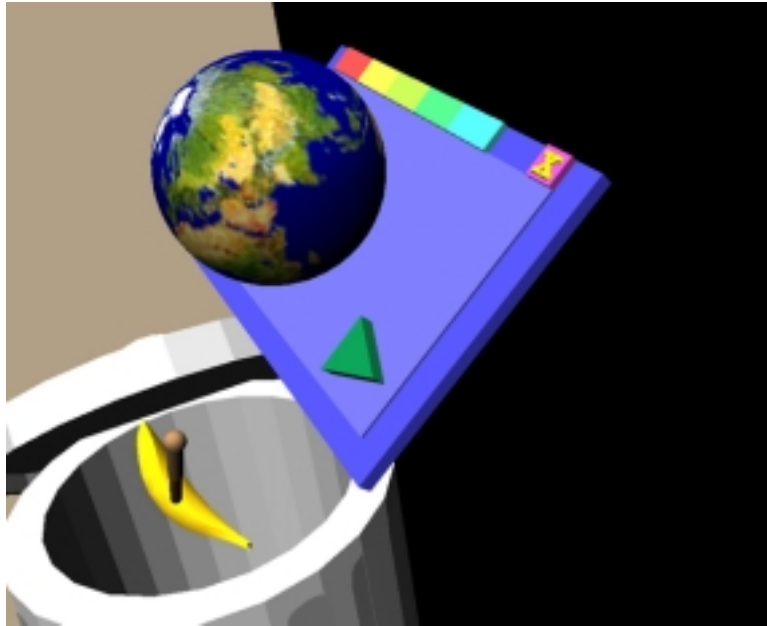


Abbildung 30: Zusätzliche Interaktions-Metaphern

Das Personal Interaction Panel wird als System für zweihändige Interaktion bezeichnet. Die Hände bedienen jedoch zwei Eingabegeräte und interagieren nicht direkt mit virtuellen Objekten. Genau genommen ist die einzige Eingabemöglichkeit die Bewegung des virtuellen Zeigers. Der Benutzer ist allerdings in der Szene mobil und das Prinzip der Augmented Reality Anwendungen erlaubt ein hohes Gefühl der Immersion, da der Benutzer ganz real ein Teil der Szene ist.

Die Steuerung komplexer Aufgaben erfolgt hier nur über das PIP. Manipulation mit Hilfe des Stifts beschränkt sich auf Objektselektion. Das Panel ist universell einsetzbar, reduziert damit die Steuerung der Anwendungen allerdings auf nur ein einzelnes Eingabegerät.

3.2.4 Real-World Benutzungsschnittstellen

Für die Steuerung abstrakter Aufgaben ist ein passendes Mapping für die Benutzungsschnittstelle nur schwierig zu finden. Die folgenden Beispiele zeigen Lösungen aus der realen Welt. Die vorgestellten Geräte haben gemeinsam, dass sie für die Ausführung abstrakter Aufgaben eingesetzt werden. Genau genommen steuern sie für bestimmte Aufgaben spezialisierte Rechner.

3.2.4.1 Texas Instruments TI-92 Plus

Die Möglichkeiten dieses Taschenrechners entsprechen fast denen eines Computers. Von der einfachen Addition bis zur Lösung von Gleichungssystemen reicht das Einsatzgebiet dieses Geräts. Zusätzlich können eigene Programme geladen und ausgeführt werden.



Abbildung 31: Texas Instruments TI-92 Plus

Man kann deutlich die Bemühungen der Designer erkennen, die Benutzungsoberfläche so übersichtlich wie möglich zu gestalten. Die meisten Tasten besitzen nur eine Funktion. Der Zahlenblock ist farblich von den anderen Tasten deutlich unterscheidbar. Die Anordnung der Buchstaben entspricht der normalen Computertastatur. Für die Bedienung der Menüs steht hingegen eine Wippe zur Verfügung, welche ein sinnvolles Mapping der

Cursor-Bewegung in der 2D-Display-Ansicht darstellt. Auch die anderen Tasten, welche für die Bedienung des Menüs notwendig sind, können durch ihre Farbe leicht der Funktionalität zugeordnet werden. Für visuelles Feedback steht ein Grafikdisplay zur Verfügung.

Wie bei allen Geräten zur Lösung mathematischer Aufgaben ist für die Bedienung das Verständnis für die Aufgabe notwendig. Die Benutzungsoberfläche kann nur dabei helfen, die notwendigen Schritte nicht noch durch die Suche nach den richtigen Tasten zu erschweren. Dies ist bei diesem Rechner gelungen.

3.2.4.2 Barthel Chemcode

Dieses Gerät ist ein Taschenrechner, welcher für Anwendungen in Chemie und Biologie konzipiert wurde. Die Tasten erlauben schnellen Zugriff auf die Eigenschaften aller bekannten Atome und den am häufigsten im Labor gebrauchten Berechnungen.

Es gibt zwei Arten von Tasten: die Auswahl der Atome erfolgt mit rechteckigen Tasten, Zahlen und Rechenarten sind rund ausgeführt. Die Bedeutung der Tasten wird fast ausschließlich durch Text und Symbole vermittelt. Die meisten Tasten sind mit mehr als einer Funktion belegt. Einige der Textinformationen sind farblich voneinander abgegrenzt. Über ein zweizeiliges Display erhält der Benutzer ein visuelles Feedback seiner Aktionen.



Abbildung 32: Chemcode

Auf den ersten Blick wirkt das Gerät völlig überladen. Die Zahlentastatur ist sehr klein ausgefallen und hebt sich optisch nicht aus der Gruppe der runden Tasten ab. Für die Gruppe der Elemente-Tasten hätte meiner Meinung nach ein Mapping von der Periodentafel der Elemente auf den Tastenblock die Verständlichkeit deutlich erhöht, weil jeder Chemiker und Biologe diese Tabelle auswendig kennt (Abbildung 33). Auch ein Gerät ohne Beschriftung könnte so bedient werden. Sicherlich ist die Größe des Geräts ein beschränkender Faktor. Wenn dadurch die Einarbeitungszeit steigt und die Benutzbarkeit leidet, sollte man sich die Frage stellen, ob der Bau eines solchen Geräts dann noch Sinn macht.

The image shows a standard periodic table of elements. The title is "Periodic Table of the Elements". The table is organized into groups (IA to VIIA) and periods (1 to 7). It includes the Lanthanide and Actinide series at the bottom. The elements are color-coded by groups: IA (yellow), IIA (orange), IIIA (light green), IVA (green), VA (light blue), VIA (blue), VIIA (purple), and VIII (red). The Lanthanide and Actinide series are shown in a separate box at the bottom, with the Lanthanide series in light blue and the Actinide series in light green.

Abbildung 33: Periodentafel der Elemente

3.2.4.3 BOSS BR-532 4-Kanal Aufnahmegerät

Bei der Aufnahme von Musik sind grundsätzlich andere Aufgaben zu erfüllen als das diskrete Tippen von Zahlen. Hier wird ein kontinuierlicher Datenstrom verändert oder durchlaufen. Der Unterschied zu den oben beschriebenen Taschenrechnern ist deutlich zu sehen: für alle kontinuierlichen Aufgaben wie das Einstellen der Lautstärke stehen Regler zur Verfügung. Diskrete Aufgaben werden mit Hilfe von Tasten gesteuert (Abbildung 34).

Die benutzten Drehregler auf der linken Seite werden mit 2-Finger Zufassunggriff bedient und erlauben daher eine hohe Genauigkeit bei der Einstellung der Werte. Die Schieberegler dieser Geräte sind so hochwertig, dass sie sich schnell und präzise bedienen lassen, obwohl ihre Form reibschlüssig ist. Der Jog-Shuttle auf der rechten Seite erlaubt sowohl den Umfassunggriff als auch den Kontaktgriff und bietet somit sicheren Griff und Präzision. Über die Stellungen der Regler ist der Benutzer mit einem Blick informiert. Diskrete Betriebszustände werden über das zweizeilige Display angezeigt. Wichtige Tasten besitzen zusätzlich zur Beschriftung eine farbliche Hervorhebung. Für jede Aufgabe wurde hier das passende Bedienelement eingebaut. Man kann sich ohne Kenntnis der Bedienungsanleitung ein Bild über die Funktionalität des Geräts machen.



Abbildung 34: BOSS: BR-532

Einen dicken Fehler haben sich die Designer allerdings erlaubt. Neben dem Jog-Shuttle wurde ein Undo/Redo-Button eingebaut. Zwei grundsätzlich verschiedene Aktionen werden mit der gleichen Taste ausgeführt. Man kann auf diese Weise nicht verhindern, dass der Benutzer nach einer Fehlbedienung auch wirklich Undo drückt. Solche Fehler dürfen besonders bei so wichtigsten Funktionen wie Undo nicht passieren.

3.2.4.4 Canon Powershot S40

Digitalkameras sind ein gutes Beispiel für den Einzug der Computertechnologie in das alltägliche Leben. Die konventionelle Kamera ist dem Prinzip der digitalen Bildbearbeitung schon jetzt zum Opfer gefallen. Trotz einer Vielzahl neuer Möglichkeiten scheint kaum jemand den Umgang mit diesen Geräten zu scheuen.

Wenn man sich die Kamera aus Abbildung 35 betrachtet, sieht man sofort, dass sich die grundlegende Bedienung nicht geändert hat. Es sind die von herkömmlichen Modellen gewohnten Bedienelemente verfügbar. Es gibt

einen Sucher und einen Auslöser, deren Funktion man aufgrund von Erfahrungen mit herkömmlichen Geräten erraten kann. Die Bedeutung des Displays erklärt sich sofort nach dem Einschalten des Geräts, was durch das Öffnen der Objektivklappe erfolgt.



Abbildung 35: Canon Powershot S40
Bedienelemente

Das Verbergen der technischen Neuerungen erlaubt offensichtlich die Weiterverwendung des mentalen Modells eines Fotoapparats.

Im Fall dieses Geräts sind auch die Elemente für die Steuerung der Software übersichtlich und leicht zu bedienen. Die Tasten haben höchstens zwei Funktionen, welche sich durch die Wahl der Kameramodi ergeben. Eine versehentliche Bedienung wird so weitgehend verhindert. Die Einstellung der Aufnahmeeigenschaften erfolgt über das einrastende Stellrad an der Oberseite des Geräts mit Hilfe aussagekräftiger Symbole. Die Bewegung des Cursors durch die Menüs (Abbildung 36) erfolgt durch einen Multicontroller, angebracht an der rechten oberen Ecke. Dieser bietet ein natürliches Mapping von der Eingabe zur Bewegung.



Abbildung 36: Auszug aus der Menüstruktur der Powershot S40

Das Display bietet im Aufnahmebetrieb kontinuierlich Informationen über die aktuellen Einstellungen. Die Unterstützung durch viele automatische Funktionen ermöglicht dem unerfahrenen Benutzer die Aufnahme qualitativ hochwertiger Bilder, der Profi kann mit wenigen Handgriffen Optimierungen vornehmen.

4 Entwicklung einer Haptischen 3D Benutzungsoberfläche

Im folgenden möchte ich meine Umsetzung einer haptischen 3D-Benutzungsoberfläche vorstellen. Sie unterstützt die natürliche Interaktion mit den Objekten der virtuellen Umgebung. Der Benutzer hat die Möglichkeit zur direkten Interaktion. Ebenso können durch den Einsatz von virtuellen Werkzeugen die Beschränkungen der virtuellen Hand umgangen werden.

Leider konnte aufgrund von Problemen mit der Force-Feedback Hardware (wahrscheinlich ein Treiber-Problem) nicht ein einziger Test durchgeführt werden. Daher sind auch keine Aussagen über die Benutzbarkeit der entstandenen API möglich.

4.1 Voraussetzungen für die natürliche Mensch-Maschine

Interaktion

Die vorgestellten Arbeiten und Systeme erlauben mir die Ableitung einiger Anforderungen an eine haptische 3D-Benutzungsoberfläche:

- Die beste und die natürlichste Art der Steuerung des Computers ist der Einsatz zweier virtueller Hände in Verbindung mit Force-Feedback Geräten zur Vermittlung haptischer Informationen. Die virtuelle Hand erlaubt den größtmöglichen Spielraum für die Interaktion des Benutzers mit der Szene.
- Die Gestaltung der Szene soll so realistisch wie möglich ausgeführt werden. Es erlaubt dem Benutzer ein tieferes Eintauchen in die virtuelle Welt und die Performanz steigt.
- Vernachlässigen kann man die Stimulation von Geschmacks- und Geruchssinn. Zum einen ist die übermittelbare Datenbandbreite sehr gering, zum anderen sind keine Anwendungen bekannt (außer Kochen), welche durch Stimulation der beiden Sinne einen Performanzschub erhalten könnten. Es existieren keine bekannten Ansätze für Ausgabegeräte, welche eine Anwendung unterstützen könnten. Aufgefallen ist mir, dass nicht einer der Autoren davon berichtet, dass eine seiner Versuchspersonen das Fehlen von entsprechenden Geräten erwähnt oder bemängelt habe. Auch die Autoren selbst erwähnten in ihren Ausblicken und Forderungen entsprechende Geräte oder Bedürfnisse mit keiner Silbe.
- Die räumliche Wahrnehmung muss durch Darbietung einer echten 3D-Ansicht möglichst gemacht werden. Grundvoraussetzung ist demnach ein entsprechendes Ausgabegerät.

- Die Möglichkeiten der Sprachsteuerung sind faszinierend. Mit Sicherheit wird bei entsprechender Verfügbarkeit entsprechender Software oder sogar Spracherkennungshardware diese Art der Eingabe rasanten Einzug in die alltägliche Umgebung der Computertechnologie halten. Zur Zeit verhindert jedoch die nötige Rechenleistung, welche erst recht bei Simulationen immer knapp ist, die intensive Nutzung dieser Funktionalität. Auch sieht die aktuelle Konfiguration der Virtual Glove Box keine Sprachbedienung vor. In Zukunft wird man nicht ohne auskommen.
- Den Einsatz von Gesten und Hand-Posturen als Eingabemetapher halte ich zum heutigen Zeitpunkt für überflüssig: Eine gut gewählte Eingabe-Metapher schafft mehr Klarheit und es kann ein einfacheres Eingabegerät gewählt werden als ein teurer Datenhandschuh. Gesten sind ebenfalls inkompatibel zu Anwendungen, welche den ständigen Einsatz der Hände in vielen verschiedenen Situationen benötigen. Der Einsatz von Werkzeugen ist ausreichend, um jede mit einer Geste möglichen Aufgabe auszuführen.
- Die Bereitstellung von aktivem haptischem Feedback erfordert eine genauere Beschreibung von Objekten der virtuellen Welt, welche in herkömmlichen Anwendungen allein durch ihre Form beschrieben werden. Eigenschaften wie Masse, Material, Beweglichkeit, Temperatur müssen in die Beschreibung des Objekts einfließen
- Die Gestaltung der virtuellen Welt sollte ebenfalls passives haptisches Feedback ermöglichen. Der Mensch ist kontinuierliche Versorgung des haptischen Sinns mit Informationen gewöhnt und wird das Fehlen dieser Stimulation bemerken.
- Die Erscheinungsbilder der Bedienelemente sollten sich so weit wie möglich voneinander unterscheiden. So kann der Benutzer mit einem

Blick erkennen, welche Möglichkeiten die aktuelle Benutzungsoberfläche bereitstellt. Die Benutzbarkeit der gesamten Benutzungsoberfläche steigt, je weniger sich der Benutzer auf das Suchen der Bedienelemente konzentrieren muss.

- Ziel der Entwicklung einer Interaktions-Metapher sollte es sein, einen maximalen Wissenstransfer von der realen in die virtuelle Welt zu ermöglichen. Dadurch wird der nötige Lernaufwand für die Benutzung der Anwendung so weit wie möglich reduziert. Allerdings wird der Benutzer von einer virtuellen Fernbedienung auch die Eigenschaften des realen Pendant erwarten.
- Wenn man den Human-Action-Cycle betrachtet, kann man ableiten, dass ein Objekt, welches von Menschen benutzt werden soll, zwei wichtige Kriterien erfüllen muss:
 - Explorierbarkeit: die Wahrnehmung eines Objekts ist nicht durch eine Konstante beschreibbar. Wahrgenommene Details formen sich durch Exploration mit Hilfe aller Sinne zu einem Ganzen, dem mentalen Modell. Ein Objekt muss explorierbar sein, sonst kann es nicht vom Menschen verstanden werden, sofern nicht weitere Informationen geliefert werden (beispielsweise in Form einer Bedienungsanleitung).
 - Interpretierbarkeit: das Objekt muss für den Menschen interpretierbar, seine Eigenschaften für den Betrachter offensichtlich sein. Nur wenn die Eigenschaften wahrnehmbar sind, werden sie durch Exploration dem mentalen Modell hinzugefügt.
- Eine Kombination von sinnvollen Affordances und Constraints sollten Eigenschaften einer guten Benutzungsschnittstelle sein.

Beispiele finden sich in der realen Welt zur Genüge, und es spricht nichts dagegen, sich an diesen zu orientieren oder sie nachzubilden.

- Einen optimalen Weg für den Bau einer Schnittstelle kann man nicht vorgeben. Auch in der realen Welt sind für das gleiche Produkt fast immer verschiedene Umsetzungen verfügbar. Die Entwicklung muss einen iterativen Prozess durchlaufen, welcher die Anforderungen des potentiellen Benutzers berücksichtigt.

4.1.1 Möglichkeiten von virtuellen Benutzungsschnittstellen

Das Nachbilden von Benutzungsschnittstellen aus der realen Welt bedeutet natürlich nicht, dass diese Konzepte nicht um Möglichkeiten der virtuellen Welt ergänzt werden können. Ein kleines Objekt könnte über einen Schalter verfügen, mit dessen Hilfe man es aufklappen könnte, um an normalerweise verdeckte Bedienelemente zu gelangen. Einem virtuellen Molekül beispielsweise könnte man auf diese direkte Art andere abstrakte Eigenschaften zuweisen, ohne einen Umweg über globale Menüs gehen zu müssen. Auch kann sich die Form während der Manipulation verändern, so dass physikalische Constraints dynamisch eingesetzt werden können. Ein Panel voller Schalter (was auch bei „gutem“ Design nicht immer vermieden werden kann) könnte Elemente, welche in einem bestimmten Zustand nicht betätigt werden sollen, verdecken oder deren Farbe ändern.

Das Anwendungsgebiet einer Benutzungsschnittstelle darf nicht beschränkt sein, wenn sie das Attribut „wiederverwendbar“ erhalten soll. Es ist demnach die Entwicklung einer allgemein gehaltenen Datenstruktur notwendig, welche den individuellen Bau und die Gestaltung der Bedienelemente zulässt.

4.2 Konfiguration der Projekthardware

Die vorhandene Konfiguration besteht aus folgenden Teilen:

- Dual Intel Xeon System mit je 933Mhz und 1GB RAM
- 3dLabs Wildcat 4210 Grafikkarte
- 2 lichtstarke Projektoren mit einer Auflösung von 1280x1024 Punkten im Inneren der Virtual Glove Box
- 2 Immersion CyberGlove/CyberGrasp Kombinationen, angeschlossen an externe Controller-Hardware, inklusive Polhemus Fastrak Positionstracker

Abbildung 37 zeigt ein Schema der Virtual Glove Box. Der Benutzer schaut auf eine Leinwand mit den Maßen 150cm x 90cm. Der Abstand zur Leinwand beträgt nur wenige Zentimeter, so dass das Blickfeld fast vollständig von der Projektion ausgefüllt ist, welche von hinten auf die Leinwand geworfen wird. Durch die Öffnungen in der Gehäusefront greifen die Hände in Glove-Box, entsprechend der Bedienung der realen Glove-Box. Der Benutzer wird durch die Position der Öffnungen derart beschränkt, dass die Hände nicht in den Projektionsstrahl gehalten werden können.

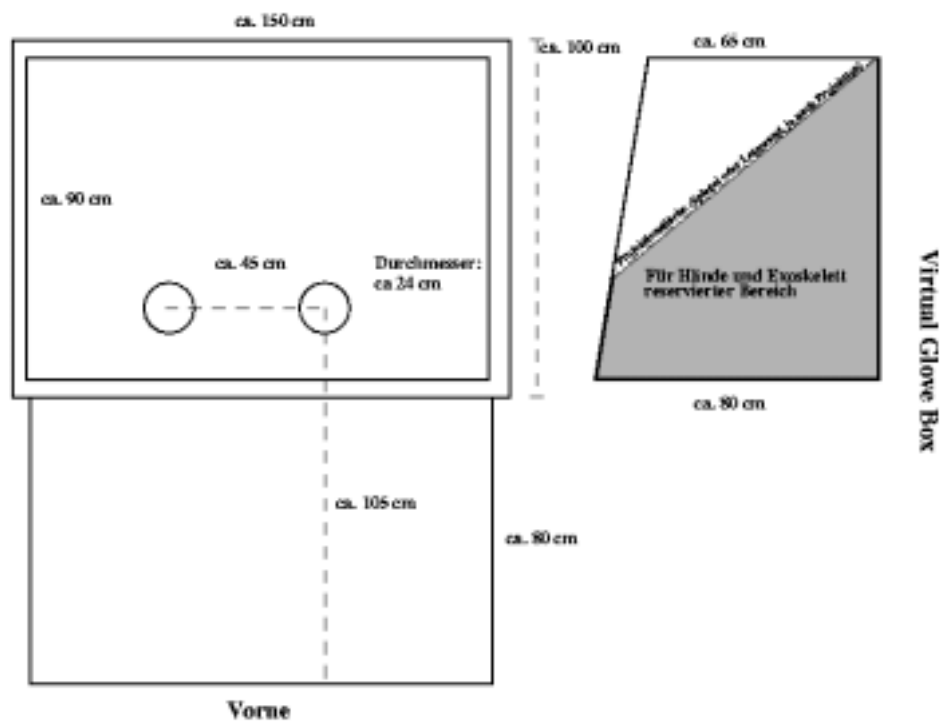


Abbildung 37: Schema der Virtual Glove Box

Durch die Verwendung von Polarisationsfiltern vor den Linsen der Projektoren ist es möglich, ohne den Einsatz unbequem zu tragender Geräte ein echtes 3D-Bild zu erzeugen. Der Benutzer trägt lediglich eine einfache Brille aus Pappe, deren Folien-Gläser wiederum Polarisationsfilter sind. So kann für beide Augen ein separates Bild auf die gleiche Leinwand geworfen werden. Abbildung 38 zeigt die Virtual Glove Box bei der Ausführung einer Simulation.



Abbildung 38: Virtual Glove Box in Betrieb

4.2.1 Funktionsweise und Einschränkungen der Fremd-Bibliotheken und der Hardware

Für die Entwicklung einer Anwendung zur Vermittlung von haptischem Feedback ist eine Datenstruktur nötig, welche die Beschreibung der haptischen Eigenschaften für jedes Objekt ermöglicht. Auch muss der Abgleich zwischen visueller und haptischer Repräsentation gewährleistet sein. Für die hier benutzten Ein- und Ausgabegeräte steht eine umfangreiche Software-Bibliothek zur Verfügung, welche auf der einen Seite die Kommunikation zwischen Simulation und Hardware stark vereinfacht, zum

anderen alle nötigen Routinen für die haptische Interaktion bereitstellt. Da diese Funktionalität stark in meine Arbeit eingeflossen ist und einfließen musste, ist es nötig, alle beschränkenden Vorgaben darzustellen:

4.2.1.1 Die visuelle und haptische Repräsentation virtueller Objekte

Die visuelle Beschreibung einer Szene kann leicht mit Hilfe eines Szenegraphen umgesetzt werden. Auf genau diesem Ansatz baut die Erweiterung um die haptische Repräsentation der Szene auf. Für jedes Objekt aus dem Szenegraphen kann automatisch eine haptische Repräsentation erzeugt werden. Dies geschieht wie folgt: nach dem Laden eines Objekts in den visuellen Szenegraph, welches aus einer oder mehreren Geometrien bestehen kann, wird eine konvexe Hülle⁷ erzeugt, welche als greifbare Repräsentation der Geometrie in den „Haptischen Szenegraph“ eingefügt wird. Der Ermittlung von Berührungspunkten liegt dabei die Berechnung von Kollisionen unter Objekten des haptischen Szenegraphen zu Grunde. Der Einsatz einer dritten Datenstruktur, dem „Data-Neutral Graph“, gewährleistet dabei die Unabhängigkeit von einer speziellen Szenegraph-Bibliothek. Sie dient als Referenz zwischen visueller und haptischen Repräsentation eines virtuellen Objekts (Abbildung 39). Jeder Knoten des Data-Neutral Graphen besitzt Zeiger auf die Vaterknoten der beiden Repräsentationsformen eines Objekts.

Durch die Trennung von visueller und haptischer Repräsentation ist es jedoch nötig, alle Änderungen während der Ausführung der Anwendung immer in beiden Graphen vorzunehmen.

⁷ Unter Anwendung des qhull-Algorithmus

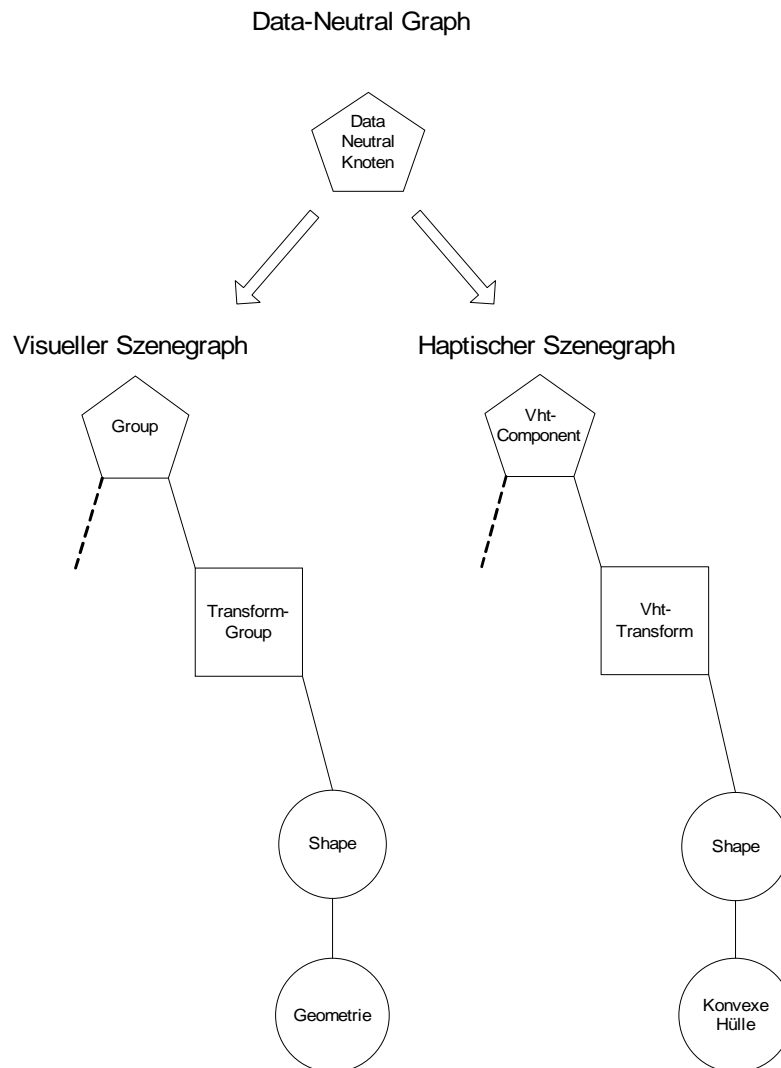


Abbildung 39: Zusammenhang von visuellem und haptischem Szenegraph

Eine andere bedeutende Einschränkung besteht darin, dass Objekte nicht an beliebiger Stelle im haptischen Szenegraph eingefügt werden können. Wie in Abbildung 39 zu sehen ist, besitzt ein greifbares Objekt einen übergeordneten Knoten vom Typ vhtComponent. Alle untergeordneten Objekte werden als Teile eines einzigen Objekts betrachtet und können daher nicht untereinander kollidieren. Die Hierarchie des visuellen Szenegraphen, welche den leichten Aufbau einer Szene erst ermöglicht, ist somit nicht auf die haptische Repräsentation der Szene übertragbar! Vielmehr müssen alle Objekte, welche

kollidierbar sein sollen, direkt an die Wurzel des haptischen Szenegraphen angehängt werden (Abbildung 40).

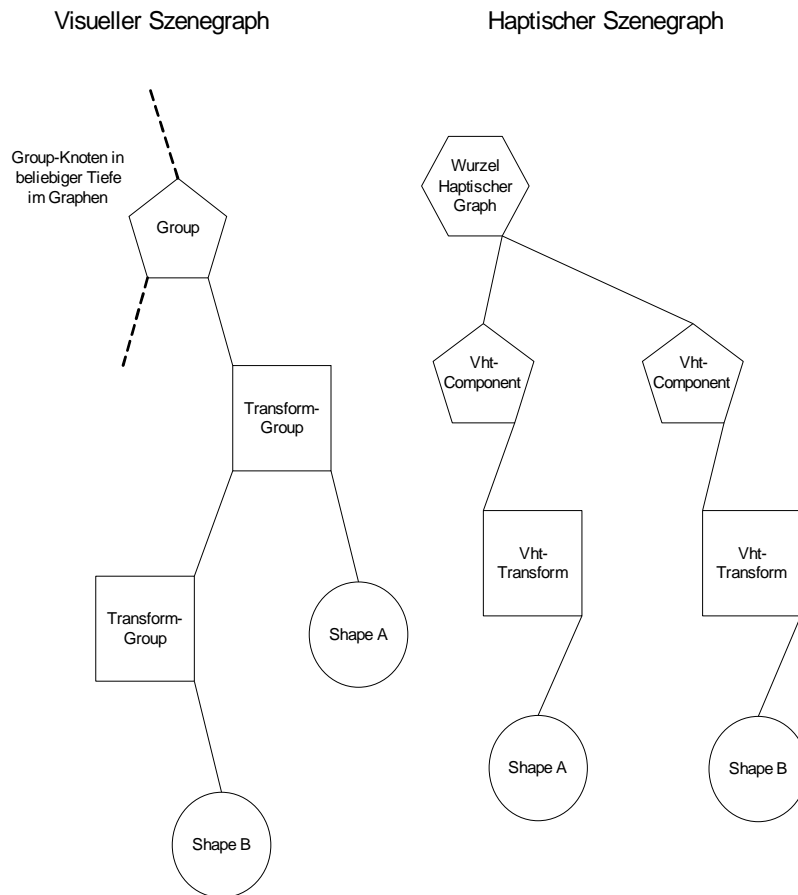


Abbildung 40: visuelle und haptische Repräsentationen befinden sich in unterschiedlicher Tiefe im Visuellen- und Haptischen Szenegraphen

Aufgrund dieser Tatsache ist es sinnvoll, die Simulation von Bewegungen vollständig mit Hilfe des visuellen Szenegraphen durchzuführen, da sonst Bewegungen von Objekten relativ zueinander nur mit viel Programmieraufwand bewerkstelligt werden können.

Soll nun ein Objekt mit der virtuellen Hand gegriffen werden, geschieht dies mit Hilfe der im haptischem Szenegraphen gespeicherten Informationen. Sobald ein Objekt von einer einstellbaren Anzahl Finger berührt wird, gilt es

als gegriffen. Bewegt man die Hand, so wird die Position des Objekts im haptischen Szenegraphen der Bewegung der Hand entsprechend angepasst. In diesem Fall muss die aktuelle Position im visuellen Szenegraphen aktualisiert werden. Für den sinnvollen Datenaustausch zwischen den beiden Szenegraphen ist jedoch allein der Entwickler verantwortlich.

4.2.1.2 Collision Detection

Um ein haptisches Feedback vermitteln zu können, muss der Punkt der Berührung genau bestimmt werden. Hierfür steht eine Engine zur Verfügung, welche bei jedem Durchlauf der Simulation die aktuellen Kollisionspaare aus den verfügbaren Objekten im haptischen Szenegraph ermittelt. Die Form der gelieferten Paare erlaubt außer der Bestimmung der kollidierten Objekte eine Aussage über den Berührungspunkt. Daraus lassen sich die Normalen an den Kontaktstellen der beiden kollidierten Objekte ableiten.

Für die Übermittlung von Kräften an die CyberGrasp Force-Feedback Geräte ist es zusätzlich nötig, für alle Kollisionspaare von einer virtuellen Hand und einem Objekt „Contact-Patches“ zu errechnen, welche eine Tangentialfläche zum Berührungspunkt darstellen. Diese werden von der Controller-Hardware ausgewertet und entsprechende Kräfte an den Seilzügen aufgebaut (genau ist dieser Vorgang nicht dokumentiert).

Die Collision Engine ist integraler Bestandteil der Bibliothek, kann nicht ausgetauscht werden und lässt keine individuelle Skalierung zu, so dass immer alle Objekte des haptischen Szenegraphen untereinander auf Kollision getestet werden. Allerdings ist sie als eigener Thread implementiert und unterstützt so den Entwurf von Multithread-Anwendungen.

4.2.1.3 Die virtuelle Hand

Auf die virtuelle Repräsentation der Hand ist kein Zugriff möglich. Es ist somit ausgeschlossen, Objekte direkt mit dem Modell zu verknüpfen, die Form zu ändern oder Zustände der Hand zu visualisieren. Beispielsweise könnte man die fehlenden Temperaturinformationen durch

Farbveränderungen an der Hand darstellen, wenn in der virtuellen Welt ein „warmer“ Gegenstand berührt wird.

4.2.1.4 Dokumentation

Die Dokumentation ist äußerst dürftig ausgefallen und in einigen Punkten falsch oder unvollständig. Viele Methoden mussten erst ausprobiert werden, um deren Funktionsweise verstehen zu können. Beispielsweise sind Matrizen der Virtual-Hand-API nicht zeilenweise sondern spaltenweise aufgebaut. Bei jeder Aktualisierung zwischen visuellem und haptischem Szenegraph ist demnach eine Umrechnung von $A_{\text{visuell}} = (a_{ik})_{(4,4)}$ nach $A_{\text{haptisch}} = (a_{ki})_{(4,4)}$ oder umgekehrt nötig. In der Dokumentation wird darauf nicht hingewiesen, nur in einem mitgelieferten Demoprogramm ist diese Umrechnung zu finden.

4.2.1.5 Die Szenegraph API: Cosmo

Für die Erstellung des visuellen Szenegraphen dient die Szenegraph Bibliothek Cosmo von Silicon Graphics, welche auf OpenGL basiert. Für das Laden virtueller Objekte in den haptischen Szenegraph steht eine Schnittstelle für diese Bibliothek zur Verfügung. Desweiteren ist sie leicht zu benutzen und eignet sich in Verbindung mit einer Erweiterung namens Optimizer auch für die Visualisierung großer Datenmengen, da viele Methoden zur Optimierung der graphischen Daten zur Verfügung gestellt werden. Ebenfalls wird Multithreading unterstützt, was auf einem Dual-Prozessor System natürlich von Bedeutung ist.

4.3 Der Ansatz

Die Aufgabe ist der Bau einer Benutzungsschnittstelle für Virtual-Glove-Box spezifische Anwendungen. Da dem Benutzer virtuelle Hände als Eingabegeräte zur Verfügung stehen, muss demzufolge jedes Schnittstellenelement mit diesen bedient werden können. Ein Blick zurück auf die in dieser Arbeit beschriebenen Virtual-Reality Anwendungen zeigt zwei grundsätzliche Methoden zur Anwendungssteuerung:

- I. Benutzen der Hände zum Steuern von Eingabegeräten, welche mit Hilfe spezieller Interaktionsmetaphern die Steuerung der Anwendung ermöglichen.
- II. Benutzen virtueller Hände, welche durch Einsatz von Gesten die Steuerung einer Anwendung ermöglichen.

Mein Ansatz soll die Vorteile beider Methoden vereinen: die Universalität der Eingabegeräte und das Einbinden des Benutzers in die Szene mit Hilfe virtueller Hände.

Durch den Einsatz der virtuellen Hände ergibt sich für die Bedienung eine Kombination von direkter Manipulation und dem Einsatz virtueller Werkzeuge, welche wiederum direkt durch die Hände manipuliert werden können. Der Benutzer soll das jeweils passende Werkzeug für die anfallenden Aufgaben zur Verfügung gestellt bekommen.

Durch diesen Ansatz, welcher nur durch den Einsatz von Force-Feedback möglich ist, wird die Anwendungssteuerung nicht auf hoch-spezialisierte Aufgaben beschränkt. Es ist sowohl die natürliche Interaktion mit Objekten möglich (wie in der realen Welt) als auch die Ausführung abstrakter Aufgaben wie das Einschalten einer Lampe über einen virtuellen Schalter oder auch der Einsatz eines virtuellen Hammers zur Zerstörung eines Objekts. Auch die Simulation einer herkömmlichen Maus ist denkbar. Ebenso ist die zweihändige Bedienung von Werkzeugen ist möglich. Die Werkzeuge, welche

einem realen Pendant entsprechen können oder speziell für eine Aufgabe entworfen sind, können zusätzlich mit in der realen Welt unmöglichen Fähigkeiten ausgestattet sein. So kann ein Hammer über das Wissen verfügen, dass er keine Objekte aus Glas zerstören darf. Er kann sich auch seinen Aufbewahrungsort merken und sich nach dem Loslassen dorthin begeben.

4.4 EASY: a graspable GUI

Aufgrund der vorangegangenen Überlegungen entstand ein Konzept, welches den dargelegten Möglichkeiten gerecht werden soll. Die Benutzungsoberfläche bietet die Möglichkeit zur zweihändigen und direkten Interaktion. Die Entwicklung der Datenstrukturen ist von den Vorgaben der benutzten Bibliotheken gekennzeichnet.

4.4.1 Das Konzept

Die Entwicklung wiederverwendbarer Oberflächenelemente impliziert die Erstellung einer Bibliothek zur Speicherung aller notwendigen Komponenten. So kann der Entwickler einer Anwendung die Komponenten und die nötige virtuelle Umgebung ohne Quellcode-Kenntnisse leicht in sein Projekt einfügen. Die statische Konfiguration der Virtual-Glove-Box (es wird immer die gleiche Hardware benutzt) spricht für die Bereitstellung eines vorgefertigten Simulationsablaufs, so dass sich der Entwickler ausschließlich um das Verhalten der Komponenten seiner Simulation zu kümmern braucht.

Benötigt wird also eine Grund-Komponente, welche der Entwickler einer Simulation nach seinen Bedürfnissen erweitern und anpassen kann. Sie sollte Wissen über ihre visuelle und haptische Repräsentation kapseln, so dass die Aktualisierung der beiden Szenegraphen automatisch erfolgen kann oder vom Entwickler nur angestoßen werden muss. Im Idealfall sollten sich aus ihr alle Bedien- und Simulationselemente ableiten lassen. Desweiteren sollte sie über eine Methode verfügen, welche bei jedem Durchlauf der Simulation aufgerufen wird und die Ausführung beliebiger Methoden ermöglicht.

Für die Steuerung von Anwendungen ist in irgendeiner Form immer der Einsatz von Schaltern oder Tasten notwendig. Auch Schiebe- oder Drehregler finden sich an vielen Geräten. Da ich keine Alternative kenne, ist demnach die Implementation solcher Elemente erforderlich.

4.4.1.1 Der Plan

Die eben beschriebene Idee liest sich schnell, die Implementierung ist aber deutlich komplexer.

4.4.1.1.1 Multithreaded Rendering

Die eingesetzte Grafikkarte ermöglicht den parallelen Anschluss der beiden Projektoren. Es stehen zwei Prozessoren zur Verfügung. Die Ansteuerung der Monitorausgänge erfolgt abhängig von der gewählten Auflösung: Ist eine Auflösung von 1280x1024 Punkten gewählt, so werden alle Inhalte, welche sich innerhalb dieser Grenzen befinden, auf dem ersten Monitor angezeigt. Alle Inhalte, die vom zweiten dargestellt werden sollen, müssen sich dann im horizontalen Bereich zwischen 1281 und 2560 Punkten befinden. Da die Anwendung bildschirmfüllend laufen soll, ist demnach die Erzeugung zweier Fenster mit den Eckpunkten $(0;0),(1280;1024)$ und $(1281;0),(2560;1024)$ notwendig.

Die Cosmo-API stellt für solche Fälle eine Lösung zur Verfügung, mit deren Hilfe man den beiden Fenstern einen eigenen Thread zuordnen kann. Beiden Threads kann eine Frame-Funktion übergeben werden, welche bei jedem Loop des Threads aufgerufen wird. Diese kann, bei entsprechender Berücksichtigung der Datenintegrität, für beide Threads dieselbe sein. So lassen sich die Berechnungen der Simulation in diese übergebene Funktion auslagern und müssen nicht von beiden Threads ausgeführt werden. Das Rendern der Szene, welches für die Erzeugung der 3D-Ansicht aus verschiedenen Kameraperspektiven erfolgen muss, kann und sollte parallelisiert werden. Es ist nur darauf zu achten, dass die Fenster synchron aktualisiert werden.

4.4.1.1.2 Berechnung der Kameraposition

Die visuellen Repräsentationen der virtuellen Hand lassen sich nicht in den Szenegraph einbinden und benötigen zur korrekten Darstellung einen OpenGL-Kontext. Obwohl Cosmo auf OpenGL basiert, ist der Einsatz

zweier Kameras pro Thread notwendig (eine für den Szenegraph, eine für die virtuellen Hände). Dies erfordert auf der einen Seite die Synchronisation der Kamerapositionen und Blickrichtungen, auf der anderen aber auch die Erzeugung identischer Frustums.

Die Kamerapositionen müssen auch an die Größe des Benutzers anpassbar sein, um einen bestmöglichen Blick auf die Szene sicherzustellen. Ebenfalls ist der Abstand der Kameras der beiden Ansichten variabel zu gestalten, so dass der Augenabstand des Benutzers berücksichtigt werden kann.

4.4.1.1.3 Einfügen von Objekten in die Szene

Der Aufbau der Szene ist dem Entwickler einer Simulation nicht abzunehmen. Das Erzeugen der haptischen Repräsentation ist es sehr wohl. Es besteht aus einer Abfolge von immer gleichen Schritten, deren Ergebnis ein Data-Neutral Knoten ist, welcher die haptische Repräsentation eines Objekts im haptischen Szenegraphen referenziert. Eine Komponente könnte diesen leicht speichern und dem Entwickler einer Simulation durch automatische Updates der beider Repräsentationen des Objekts viel Arbeit ersparen.

4.4.1.1.4 Kollisionsberechnung

Die Collision-Engine läuft ebenfalls als eigener Thread. Um sie einzusetzen muss die Klasse `vhtSimulation` abgeleitet und die Methode `handleConstraints()` überschrieben werden. Diese Methode wird in jedem Loop des Threads aufgerufen und muss Code für den Start der Kollisionsberechnung enthalten. Während der Kollisionsberechnung darf der haptische Szenegraph nicht verändert werden, was durch das Setzen eines Locks realisiert werden kann. Problematisch ist jedoch, dass ohne weiteren Abgleich des Programmablaufs Situationen auftreten können, in denen während eines Loops der Collision-Engine mehrere Loops der Simulation durchlaufen werden können (oder umgekehrt). Der Einsatz eines Locks zur

abwechselnden Berechnung von Simulation und Kollisionsdaten ist die meiner Meinung nach beste Lösung für dieses Problem.

4.4.1.1.5 Einbinden der Hardware

Die Anpassung der Hardware erfolgt mit Hilfe eines Programms namens DCU, welches die leichte Einrichtung der Kommunikationsparameter ermöglicht. Für jedes angeschlossene Gerät lässt sich ein Name eingeben, über welchen es aus einer Anwendung heraus angesprochen werden kann, solange DCU läuft. So kann mit wenigen Zeilen Code die komplette Hardware eingebunden werden.

4.4.1.1.6 Einbinden der haptischen Komponenten

Für die Koordination der Komponenten der Simulation ist eine übergeordnete Instanz nötig, welche das Ausführen der Methoden jeder eingebundenen Komponente auslöst. Diese Instanz kann gleichzeitig für das Auswerten von Kollisionspaaren benutzt werden und diese an die beteiligten Komponenten übermitteln, welche selbst über die Weiterverarbeitung entscheiden können. Auch sollte eine zentrale Stelle eine Übersicht über die eingebundenen Komponenten verwalten.

4.4.1.1.7 Die Hände und die Werkzeuge

Die direkte Manipulation von Objekten ist bereits durch die Benutzung der bereitgestellten Bibliotheken möglich. Für den Einsatz von Werkzeugen muss jedoch ein Verfahren gefunden werden, welches die Kommunikation zwischen jeder Art von Komponenten erlaubt. Wenn ein Hammer auf ein anderes Objekt trifft, reicht es möglicherweise, wenn das Objekt in seiner Liste von Kollisionspaaren den Hammer entdeckt. Wenn ein Pinsel die Farbe eines Objekts verändern soll, reicht die einfache Bestimmung einer Kollision nicht mehr aus. Es ist zusätzlicher Datenaustausch erforderlich

4.4.2 Implementation

Die Entwicklung der Bibliothek bestand zu einem Großteil aus der Bereitstellung eines Frameworks für den Einsatz der durch die

Aufgabenstellung geforderten wiederverwendbaren Oberflächen-Komponenten. Der Vorteil dieser Methode besteht darin, dass ein Entwickler sich allein auf die Definition des Verhaltens der Komponenten beschränken kann. Das entwickelte Grundgerüst einer haptischen Komponente kann zum einen für die Erstellung von Benutzungsoberflächenelementen benutzt werden, zum anderen können andere abstrakte Konzepte für die Erstellung einer Simulation realisiert werden.

4.4.2.1 Die Haptische Komponente

Ziel der Entwicklung war eine Komponente, welche sowohl als Gerüst für ein Werkzeug als auch für eine allgemeine Simulationskomponente eingesetzt werden kann. Hierfür war es nötig, die Gemeinsamkeiten zu finden und in einer Datenstruktur zu vereinen. Die Komponenten sind nur in Verbindung mit dem unten beschriebenen Komponenten-Manager lauffähig.

4.4.2.1.1 HapticComponentBase

Die Basis bildet die Klasse `HapticComponentBase`. Sie stellt die folgenden Möglichkeiten bereit:

- Laden beliebiger Geometrien
- Automatischer Abgleich von visueller und haptischer Repräsentation
- Speicherung und Abfrage des Data Neutral Knoten für den Zugriff auf die haptische Repräsentation
- Speicherung der jeweils aktuellen Kollisionspaare
- Zyklisch aufgerufene Methode zur Realisierung eines beliebigen Verhaltens
- Möglichkeit zur Blockierung der Erstellung einer haptischen Repräsentation

- Flags für die Abfrage der Zustände Gegriffen, Nicht Gegriffen, Losgelassen

Diese Struktur ermöglicht den einfachen Bau eines visuellen Szenegraphen und kapselt die gesamte Aktualisierung und Repräsentation des haptischen Teils der Komponente. Abbildung 41 verdeutlicht dies.

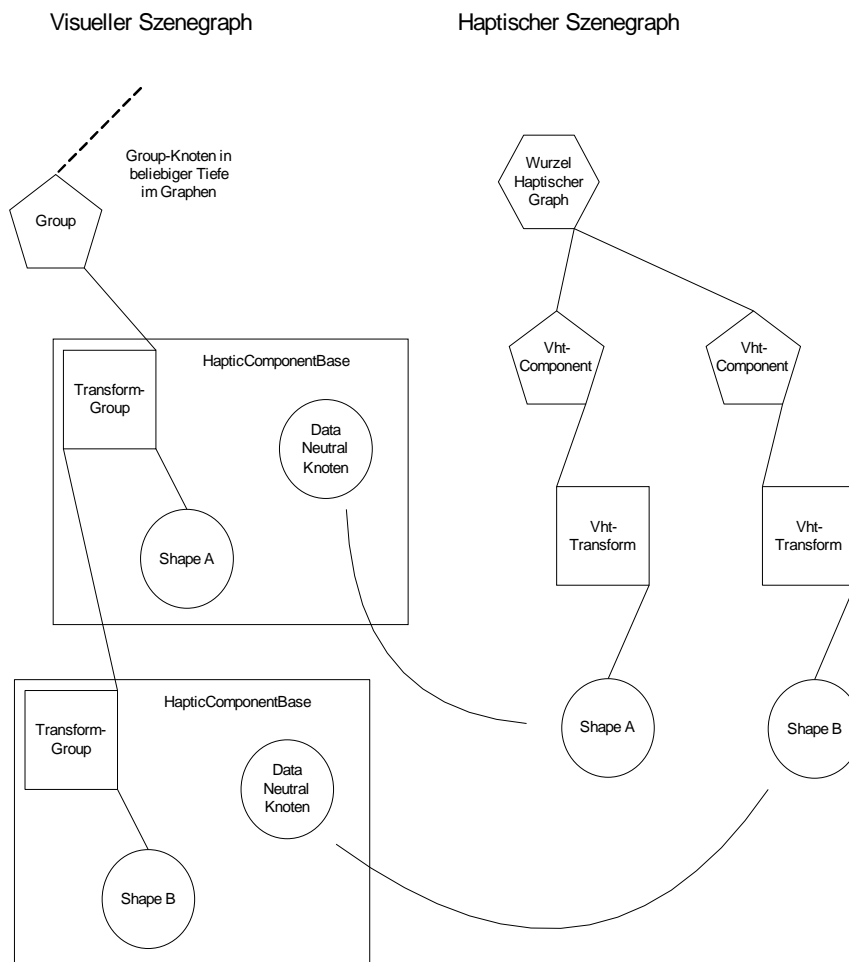


Abbildung 41: Aufbau der Szene unter Verwendung der Klasse HapticComponentBase

Der Szenegraph muss nicht ausschließlich aus haptischen Komponenten bestehen. Er kann den Anforderungen der Anwendung entsprechend konstruiert werden. Nur Objekte, welche direkt von der virtuellen Hand oder

indirekt von Werkzeugen manipuliert werden sollen, müssen von diesem Typ sein.

Die Komponente besitzt eine Methode namens `updateLoop(ticks)`, welche in jedem Frame vom Komponenten-Manager aufgerufen wird. Sie enthält drei Abfragen über den aktuellen Zustand, welche als `virtual` deklarierte Methoden aufrufen:

1. Wurde die Komponente von der virtuellen Hand gegriffen, wird automatisch die im haptischen Szenegraphen geänderte Position mit der im visuellen abgeglichen.
2. Wurde die Komponente losgelassen, wird ebenfalls die Position aus dem haptischen Szenegraphen übernommen.
3. Wenn die Komponente nicht gegriffen oder losgelassen wurde, wird die Methode `mySimulation(ticks)` aufgerufen.

In abgeleiteten Klassen können sie überschrieben werden, um individuelles Verhalten implementieren zu können. Der Parameter `ticks` ist die Zeit in Millisekunden, die für den letzten Simulationsdurchlauf benötigt wurden. Natürlich sind auch Komponenten denkbar, welche sich nicht bewegen oder sonstige Berechnungen durchführen. Dann kann diese Klasse selbst instantiiert werden.

4.4.2.1.2 HapticComponent

Für den Bau eines virtuellen Werkzeugs ist die Klasse `HapticComponentBase` jedoch nicht ausreichend. Ein Werkzeug besteht normalerweise aus einem Griff und einem oder mehreren Teilen zur Manipulation der Werkstücke. Ein Hammer hat einen Stiel und eine spitze und eine flache Seite, um Nägel zu bearbeiten. Die Geometrien der virtuellen Werkzeuge müssen demnach einen Rückschluss darauf zulassen, welches Teil an einer Kollision beteiligt ist.

Die Klasse `Place` dient der Einbindung beliebiger Geometrien zusätzlich zum Hauptobjekt. Der hierfür benötigte Typ von Komponente wurde aus der Klasse `HapticComponentBase` abgeleitet. Es entstand die Klasse `HapticComponent`. Diese besitzt die nötige Logik zur Verwaltung der zusätzlichen Instanzen von `Place`. Damit `Place`-Instanzen auch kollidieren können, müssen auch sie in den haptischen Szenegraphen eingefügt werden. Auch hier besteht für den Entwickler kein weiterer Handlungsbedarf. Er muss lediglich die `Place`-Instanzen unter den Root-Knoten der zugehörigen Komponente hängen, das Erzeugen der haptischen Repräsentation erfolgt automatisch bei der Anmeldung am Komponenten-Manager. Abbildung 42 zeigt die Beziehung zwischen visueller und haptischer Repräsentation in den beiden Szenegraphen.

Die Klasse `Place` ist für dafür gedacht, die haptischen Komponenten um kollidierende Geometrien zu erweitern. So kann die Spitze eines Hammers vom Rest der Geometrie klar unterschieden werden. Ebenso können semantische Constraints durch beliebige Erweiterungen um `Place`-Instanzen simuliert werden. `Place`-Instanzen sind nicht greifbar und nur in Verbindung mit der Klasse `HapticComponent` sinnvoll einsetzbar.

Eine visuelle Repräsentation dieser Erweiterungen ist nicht immer sinnvoll, da sie das Erscheinungsbild des eigentlichen Objekts verändert. Für manche Anwendungen kann dies aber durchaus gewünscht sein. Beispielsweise könnte eine `Place`-Instanz ein Verbindungsstück zwischen zwei Komponenten symbolisieren. Es besteht auf jeden Fall die Möglichkeit, durch Setzen eines Flags die visuelle Repräsentation ebenfalls einzubinden.

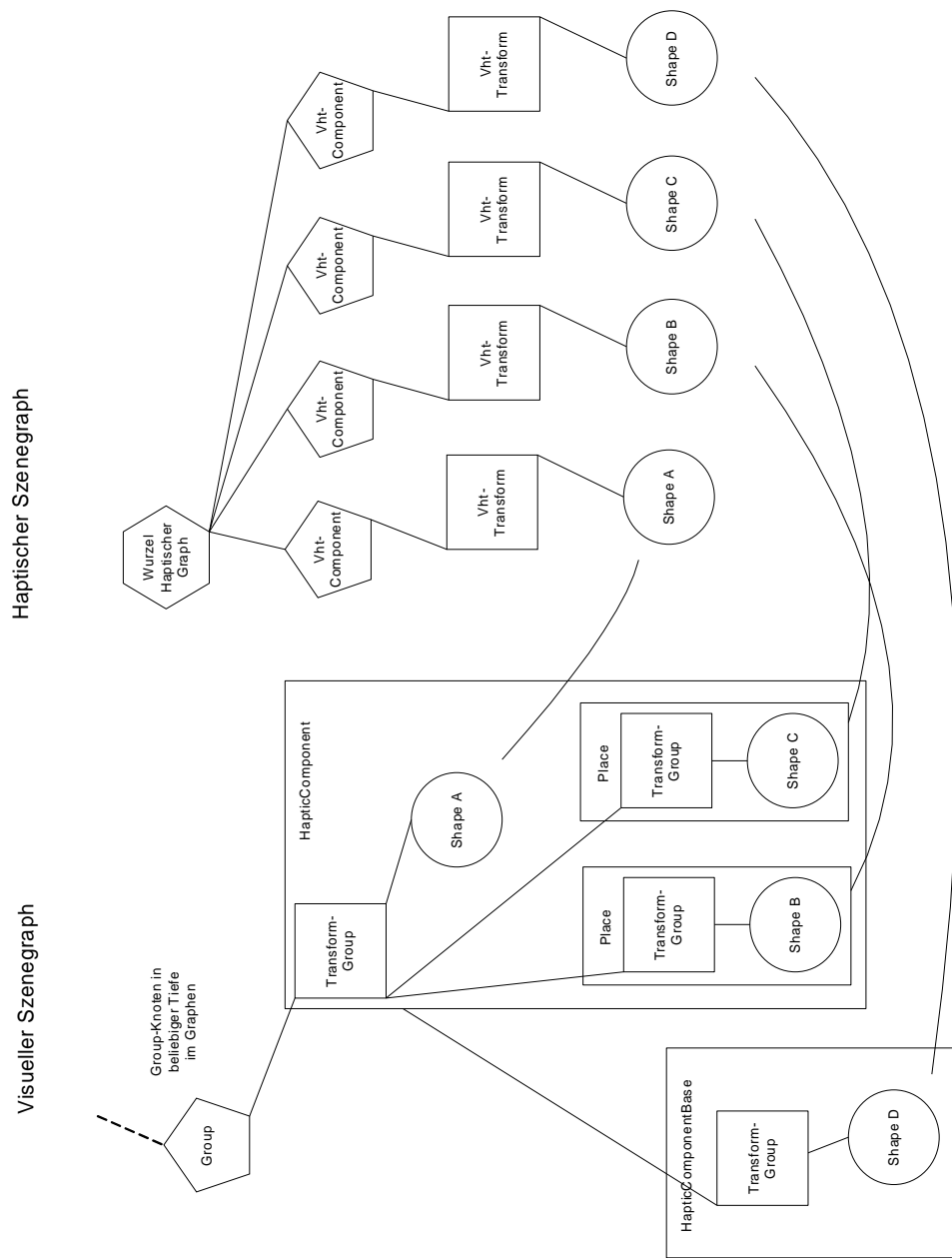


Abbildung 42: Komponenten im visuellen und haptischen Szenegraph

4.4.2.2 ComponentManager

Die Klasse `ComponentManager` ist der zentrale Punkt des Simulations-Frameworks. Die bereitgestellten Methoden koordinieren den Ablauf von Hand-Interaktion, Kollisionserkennung und Aufruf des `updateLoop` aller registrierten Komponenten.

4.4.2.2.1 Registrieren der Komponenten

Jede Komponente muss sich beim Komponenten-Manager anmelden, um im haptischen Szenegraph registriert zu werden. Durch den Aufruf der Methode `registerComponent(aComponent)` wird dieser Vorgang gestartet. Die Methode ist überladen und erlaubt das Registrieren von Instanzen der Klassen `HapticComponentBase` und `HapticComponent`. Für alle Geometrien der Komponenten wird zunächst die konvexe Hülle errechnet, anschließend werden diese als haptische Repräsentation der Geometrien in den haptischen Szenegraph eingefügt. Die registrierte Komponente erhält dabei Zeiger auf die zugehörigen Data-Neutral Knoten, so dass im weiteren Verlauf der Simulation die einfache Identifizierung gewährleistet ist.

Jede registrierte Komponente wird ebenfalls in eine Liste eingetragen, so dass zu jeder Zeit eine Suche über alle Komponenten durchgeführt werden kann. Ein Werkzeug könnte nach einem passenden Aufbewahrungsort für sich suchen. Auch Objekte, welche keine haptische Repräsentation besitzen sollen, können sich registrieren lassen. Hierfür muss nur eine Komponente ohne bereits zugewiesene Geometrie beim Komponenten-Manager registriert werden. Die visuelle Repräsentation kann später zugefügt und separat in den visuellen Szenegraphen eingebunden werden.

4.4.2.3 Simulations-Schleife

Die Methode `simulationLoop(ticks)` wird in jedem Frame durchlaufen und koordiniert den Ablauf aller Teile der Simulation, welche mit den

Berechnungen im haptischen Szenegraph zusammenhängen. Die einzelnen Schritte sind folgende:

- Alle Komponenten werden auf ihren aktuellen Zustand bezüglich der Interaktion mit den virtuellen Händen geprüft. Das Ergebnis der Berechnung ist später für die Komponenten leicht verfügbar.
- Die Liste der Kollisionspaare, welche vorab von der Collision-Engine an den Komponenten-Manager übergeben wurde, wird durchlaufen und zum einen auf sinnlose Kollisionen untersucht (zum Beispiel Kollisionen zwischen einer Instanz vom Typ `Place` und einer Hand), zum anderen werden bei Kollisionen zwischen Hand und haptischen Komponenten die für die Controller-Hardware notwendigen Kontaktflächen berechnet und an die entsprechende Instanz der virtuellen Hand übermittelt.
- Die übrigen Kollisionspaare werden an die beteiligten Kollisionspartner, also die Komponenten mit haptischer Repräsentation in der Szene übermittelt.
- Als letztes folgt der Aufruf der Simulations-Methoden aller registrierten Komponenten in der Reihenfolge ihrer Registrierung.

4.4.2.4 Haptisches Feedback

Die gesamte Funktionalität für die Vermittlung des haptischen Feedbacks durch das CyberGrasp-Gerät wird von den Fremdbibliotheken bereitgestellt. Es stehen drei Modi zur Verfügung:

- Impedance-Mode: Aus den vom Komponenten-Manager bereitgestellten Kontaktflächen werden ohne weitere Programmierarbeit die Kräfte entsprechend der relativen Position eines Objekts zum berührenden Finger ermittelt und auf die Seilzüge des CyberGrasp übertragen. Sofern keine Berührungen erkannt

werden, also keine Kollision zwischen Hand und Objekt erkannt wurde, führt das Gerät die Seilzüge entsprechend der Stellung der Finger nach, so dass die Schlaufen am Ende der Seilzüge nicht von den Fingern rutschen.

- Force-Mode: Der Entwickler kann jeden Seilzug direkt ansprechen und der Anforderung entsprechend Kräfte auf die Finger übertragen.
- Force-Effect-Mode: Es können beliebige Feedback-Effekte erstellt und gespeichert werden. Beispielsweise können Vibrationen geringer Frequenz oder auch sinusförmige Kraftzuwächse simuliert und ohne weiteren Programmieraufwand an das CyberGrasp übertragen werden.

Für den Einsatz in der Virtual-Glove-Box ist der Einsatz des Impedance-Modus sinnvoll, da er das Greifen und Fühlen, welches für die direkte Interaktion mit virtuellen Objekten nötig ist, leicht benutzbar bereitstellt.

Um den Gebrauch von Effekten zu ermöglichen, wurde die Übermittlung der Kollisionsdaten an die Hardware in eine eigene Klasse namens `GraspForceManager` gekapselt. Eine gegriffene Komponente kann so den Modus des zur Greifhand gehörigen `CyberGrasp` ändern und einen Effekt starten. Allerdings ist unbekannt, wie sich ein Wechsel des Modus während der Laufzeit der Anwendung auswirkt, so dass an dieser Stelle der alleinige Betrieb im Impedance-Modus empfohlen wird. Dies ist auch die Voreinstellung beim Starten des Frameworks.

4.4.2.5 Programmablauf

Das Computersystem, welches für das Virtual-Glove-Box Projekt zur Verfügung steht, stellt zwei Prozessoren bereit. Der Ablauf wurde durch Einsatz der Thread-Technologie soweit wie möglich parallelisiert. Dies unterstützt auf der einen Seite das Rendern der beiden Ansichten, auf der anderen Seite kann die aufwendige Kollisionserkennung direkt nach dem

Durchlauf der Frame-Funktion erneut gestartet werden. Abbildung 43 zeigt das grundsätzliche Schema der Schleife des Simulations-Frameworks.

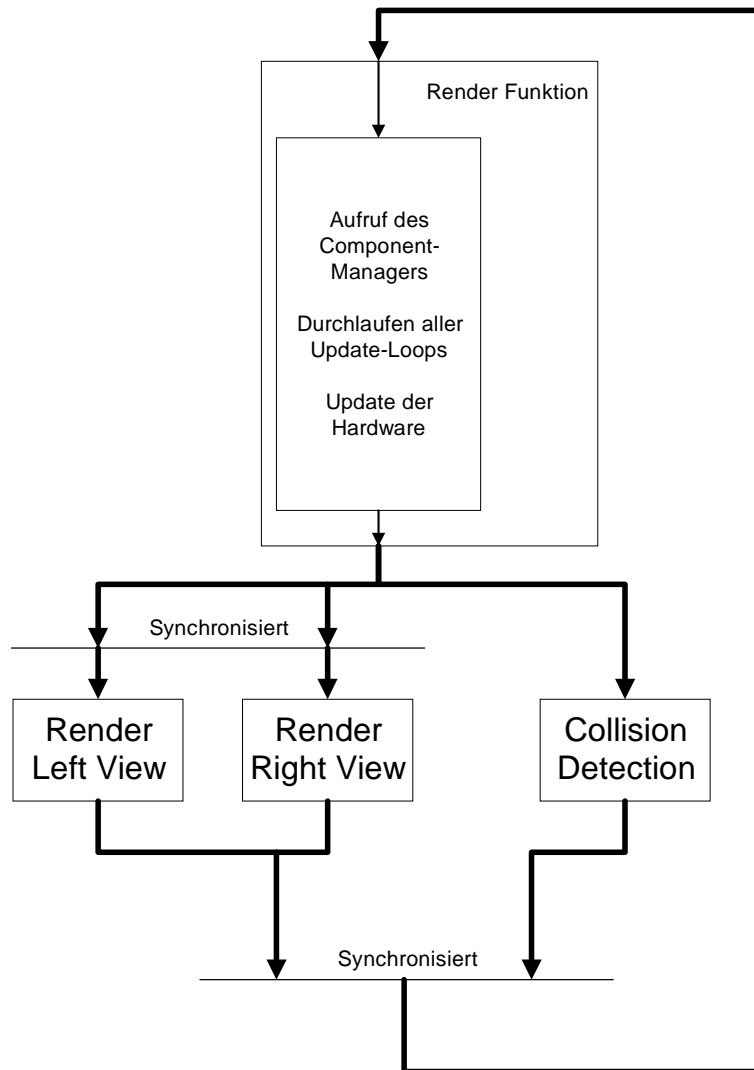


Abbildung 43: Programmablauf des Simulations-Frameworks

Das Rendern der beiden Ansichten erfolgt erst nach den Berechnungen der Simulation. So ist sichergestellt, dass beide Ansichten die Szene zum gleichen Zeitpunkt darstellen. Das Wechseln der Window-Buffer geschieht ebenfalls synchron, damit beide Augen zur gleichen Zeit die aktualisierte Ansicht der Szene präsentiert bekommen. Die Collision-Engine, welche auch als eigener Thread läuft, kann parallel zum Rendern laufen, da sie selbst nicht auf die visuelle Repräsentation der Szene zugreift.

4.5 Der Einsatz von EASY

Um die Anwendung der eben beschriebenen Strukturen zu verdeutlichen, möchte ich an dieser Stelle den Code für den Aufruf des Frameworks und das Einfügen einer haptischen Komponente beschreiben⁸.

```
#include "easyOpenGL.h"

int
main(int argc, char *argv[])
{
    // Framework erzeugen durch Instanzierung eines Objekts
    // vom Typ easyOpenGL
    easyOpenGL *easy = new easyOpenGL;
    easy->initClasses();

    // eine Komponente erzeugen
    HapticComponent *aComponent = new HapticComponent;

    // ein Vrlm-File Laden
    // diese Methode ist durch Cosmo-API vorgegeben
    opGenLoader *loader = new opGenLoader;
    csTransform *aTransform = new csTransform;
    aTransform->addChild(loader->load("einModell.wrl"));

    // das Vrlm-Modell der Komponente uebergeben
    aComponent->addVisual(aTransform);

    // Die Komponente registrieren
    easy->getComponentManager()->registerComponent(aComponent);

    // den Simulations-Loop starten
    easy->startLoops();
}
```

Dieser Code führt zu dem in Abbildung 44 dargestellten Ergebnis. Die eingefügte Komponente ist greifbar und verfügt über eine einfache, mit einem 3D-Modeller erstellte Geometrie. Wie man sehen kann beschränkt sich die Aufgabe des Entwicklers hauptsächlich auf das Einbinden der Geometrie und das Erstellen der Logik für die eingebundenen Komponenten. Die komplette Dokumentation steht separat im HTML-Format zur Verfügung.

⁸ Die gezeigten Szenen wurden für den Ausdruck dieses Dokuments angepasst.

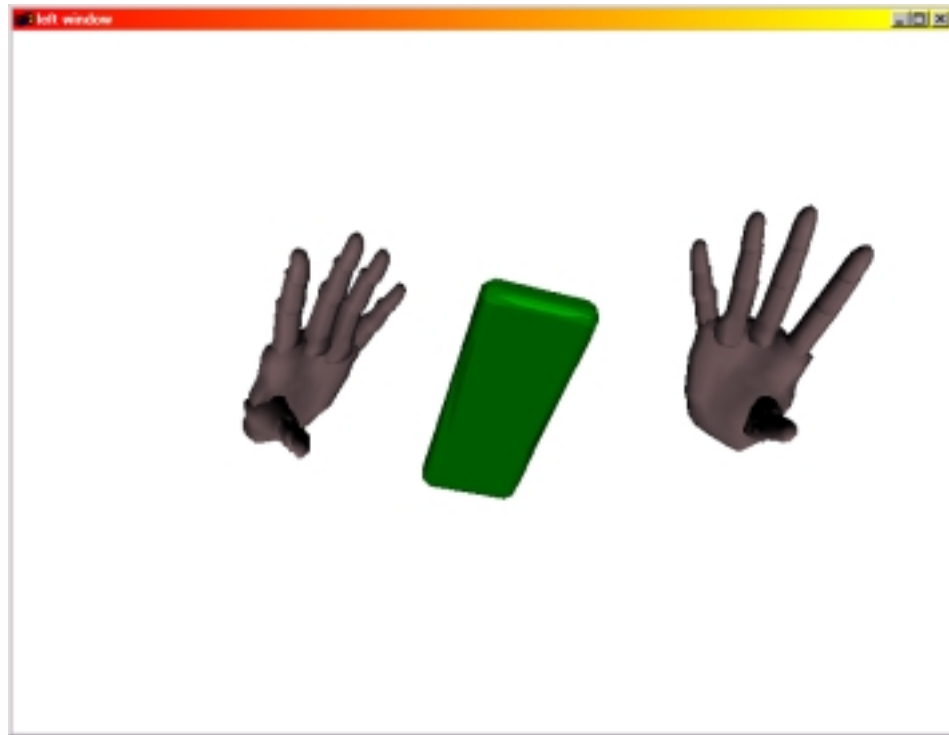


Abbildung 44: EASY -- Ergebnis der Ausführung des Codes

4.5.1 Wiederverwendbare Benutzungsschnittstellenelemente

Die bereitgestellten Klassen erlauben den individuellen Bau von Schnittstellenelementen. So kann der Entwickler allen Aufgaben, welche nicht durch direkte Interaktion zu lösen sind, ein angepasstes Bedienelement zur Verfügung stellen. Auf diese Weise wird auch die universelle Verwendbarkeit dieser Bibliothek sichergestellt.

4.5.1.1 Schalter und Regler

Für viele abstrakte Aufgaben bei der Steuerung von Geräten werden Schalter und Regler benutzt. Dies widerspricht zwar der Forderung nach einer kontinuierlichen Bedienung, in Ermangelung einer Alternative ist es nötig, genau diese Elemente zur Verfügung zu stellen. Mir scheint die Benutzung von Schaltern und Reglern innerhalb dieses Rahmens jedoch deutlich weniger anfällig für Fehlbedienung zu sein. Zum einen werden die Elemente mit der Hand bedient, über welche man perfekte Kontrolle besitzt, zum anderen

können die Elemente an andere Komponenten angehängt werden, wodurch ihr Bezugsrahmen sofort ersichtlich ist.

Da ich durch die Schwierigkeiten mit der Hardware keinerlei Erfahrung in der Anwendung des Impedance-Modus und seiner Einschränkungen besitze, kann ich hier nur ein Konzept für den Bau dieser beiden Elemente beschreiben.

4.5.1.1.1 Schalter

Die Klasse `HapticComponentBase` kann abgeleitet und um Elemente zur Speicherung des Zustands eines Schalters ergänzt werden. Ein erster Ansatz könnte eine Boolesche Variable sein, welche den Zustand Ein/Aus speichert. Problematisch wird es bei der Bedienung des Schalters. Sobald ein Objekt von der virtuellen Hand gegriffen wurde, wird seine Position relativ zur Position der Hand gesetzt. Ein Schalter sollte sich aber nicht aus seinem Rahmen heraus bewegen (sofern er einen besitzt). Die Position sollte sich entsprechend der wahrgenommenen Affordances verändern.



Abbildung 45: Lichtschalter.

Die Affordances des Schalters aus Abbildung 45 lassen erwarten, dass der Schalter in Form einer Wippe realisiert wurde und offensichtlich der obere Teil der Wippe gedrückt werden muss, um den Schaltzustand zu ändern.

Sollte ein funktionierender Impedance-Modus diese Umsetzung nicht unterstützen, wäre eine Alternative der Bau von Schaltern, welche keine beweglichen Elemente besitzen und allein auf Berührung reagieren.

4.5.1.1.2 Regler

Für den Bau von Reglern gelten die gleichen Einschränkungen wie für Schalter. Zwar sind auch sie mit Hilfe der Klasse `HapticComponentBase` zu erstellen, doch wie der Schalter sollte der Regler in seiner Bedienung den Erwartungen entsprechen.

Sollte der Impedance-Modus in Zukunft funktionieren, könnte der Entwickler beispielsweise ein Regler als Stellrad realisieren. Position und Stellwinkel könnten leicht aus den zugehörigen Transformation ausgelesen werden. Der eingestellte Wert kann dann für die weitere Verarbeitung zur Verfügung gestellt werden oder die Komponente selbst aktiv werden lassen.

4.5.2 Ein Kontrollpanel

Bedienelemente jeder Art könnten aus den bereitgestellten Datenstrukturen gebaut und benutzt werden, sofern haptisches Feedback vermittelt werden kann. Abbildung 46 zeigt ein solches Beispiel.

Eine Szene mit mehreren Elementen zur Anwendungssteuerung zeigt Abbildung 47. Es ist eine Desktop-Metapher zu sehen, welche den anderen Elementen als Ablage dient. So wird verhindert, dass Objekte als frei in der Luft hängend wahrgenommen werden. Die Bedienelemente können beliebig angeordnet werden. Auf dem Desktop befindet sich ein Werkzeug, welches jederzeit von der virtuellen Hand gegriffen werden kann. Die „schwebenden“ Moleküle stellen die eigentliche Simulation dar.



Abbildung 46: EASY -- Beispiel für ein haptisches Kontrollpanel als Benutzungsoberfläche

Der Gestaltung der Benutzungsoberfläche sind mit EASY keine Grenzen gesetzt. Die einzigen Beschränkungen liegen in der begrenzten Reichweite der Hände durch das Konstruktionsprinzip der Virtual Glove Box. Eine realistisch wirkende Umgebung ist hier nicht zum Einsatz gekommen. Die einzufügenden Elemente würden die Performanz noch weiter drücken.

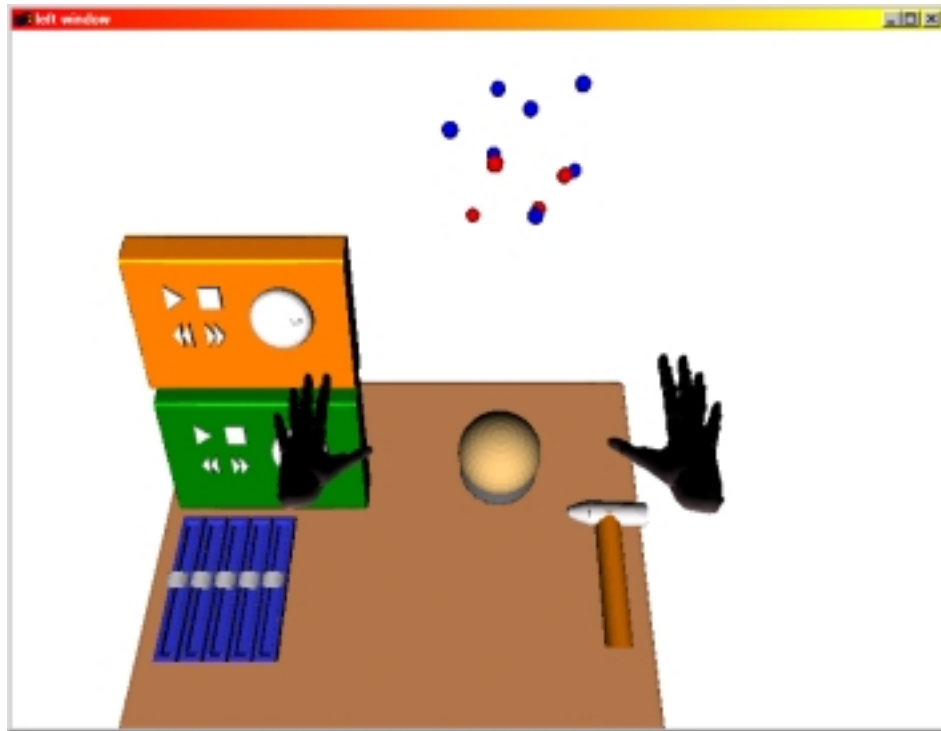


Abbildung 47: EASY -- komplette Szene

4.6 Bewertung

Ich habe gezeigt, dass mit den von mir entwickelten Komponenten eine komplexe haptische 3D-Benutzeroberfläche gebaut werden kann. Das notwendigerweise entstandene Framework bietet zudem die einfache Entwicklung jeder Art von Simulationen.

Leider konnte nicht ein Probelauf mit haptischem Feedback durchgeführt werden, so dass die Benutzbarkeit im Praxiseinsatz nicht beurteilt werden kann. Einige Eindrücke konnten dennoch gewonnen werden:

- Die Größe der Objekte sollte das Greifen mit den Fingerspitzen ermöglichen. Durch die Konstruktion des Exoskeletts kann nicht verhindert werden, dass Objekte in der Hand diese augenscheinlich durchdringen. Diesem Effekt kann durch die Benutzung kleinerer Objekte entgegen gewirkt werden.
- Die Geometrien der in den haptischen Szenegraph eingefügten Objekte sollten einen möglichst geringen Durchmesser haben. So werden die Eigenschaften der bereitgestellten Collision-Engine am effizientesten ausgenutzt.
- Die Collision-Engine ist der aufwendigste Teil der Berechnungen. Ab etwa 200 Kollisionen pro Frame sinkt die Framerate auf unter 10 Bilder pro Sekunde.
- Die eingefügten Elemente nehmen sehr schnell die zentrale Rolle im Blickfeld ein. Der Designer einer Simulation muss darauf achten, dass die Benutzeroberfläche sich dezent in die Simulation einfügt.
- Das Tragen der Datenhandschuhe in Verbindung mit den Exoskeletten ist nicht viel länger als eine Stunde zu ertragen. Sind die Stellschrauben zur Anpassung an die Hand des Benutzers zu locker, rutschen entweder die Schlaufen von den Fingern oder das ganze

Exoskelett rutscht. Dreht man die Stellschrauben etwas fester, fühlt man nach einiger Zeit nicht mehr jeden Teil seiner Hand.

- die Qualität der Hardware ist nicht sehr hoch. Beispielsweise sind die Schlaufen, welche um die Fingerkuppen gelegt werden, so miserabel an den Seilzügen befestigt, dass schon einige abgerissen sind und wieder befestigt werden mussten.

Besonders aufgefallen ist mir folgendes: einige Kommilitonen trugen teilweise die Hardware. Sie sollten versuchen, einige der virtuellen Objekte zu ergreifen. Ich konnte sie mit dem Satz „Nimm’ mal das Grüne Ding in die Hand!“ genau dazu bringen. Es gab kein Nachfragen oder Missverständnisse. Die Interaktion durch die virtuelle Hand ist demnach nicht nur leicht zu erlernen, sie ändert auch die Beschreibungsform der auszuführenden Aufgaben in Richtung der alltäglichen Kommunikation.

5 Ausblick

Ich habe ein System für den Bau individueller Benutzungsoberflächen entwickelt. Es besteht die Möglichkeit zur direkten Manipulation von Objekten einer virtuellen Umgebung. Der Einsatz von virtuellen Werkzeugen zur Erweiterung der Fähigkeiten der virtuellen Hände ist in dieser Kombination neu.

Der nächste Schritt muss natürlich das Finden einer Lösung für die Probleme mit der Ausgabehardware sein. Möglicherweise kann ein eigener Impedance-Modus implementiert werden. Genauso wichtig ist meiner Meinung nach die Entwicklung einer offenen Schnittstelle für die Einbindung beliebiger Hardware, so wie es bei den bekannten Window-Systemen der Fall ist.

Ich habe in dieser Arbeit meine Aufmerksamkeit vor allem der Implementierung des Frameworks und der Stabilität geschenkt. Für den praxisgerechten Einsatz des Systems fehlt eine Plattform für die effiziente Kommunikation der Komponenten. Hierfür ist ebenfalls ein Mindestmaß an Wissen der Komponenten über virtuelle Welt nötig. Wenn beispielsweise ein Hammer einen virtuellen Gegenstand trifft, muss dieser sich entsprechend Verhalten. Die Entwicklung der Komponenten könnte von einer bereitgestellten Wissensbasis profitieren.

Sollte EASY weiterentwickelt werden (was ich hoffe), müsste man eine Untersuchung der ergonomischen Eigenschaften der Hardware durchführen. Schließlich trägt der Benutzer unhandliche Geräte an den Händen, wodurch diese schnell ermüden. Ebenso ist die Aufstellung einer Richtlinie für den Bau des virtuellen Arbeitsplatzes zu erstellen. Wenn der Mensch mehrere Stunden mit diesen Systemen verbringt, spielen auch Faktoren wie die Beleuchtung in der virtuellen Umgebung eine Rolle.

Ich glaube, dass mit der Entwicklung besserer und vor allem billigerer Hardware die Verbreitung haptischer graphischer Oberflächen nicht aufzuhalten ist. Dafür sind die vorstellbaren Möglichkeiten zu groß.

BIBLIOGRAPHY

- [1] Mundell, *Towards a Virtual Operating Environment*.
<http://citeseer.nj.nec.com/mundell99towards.html>, 1999
- [2] Zwisler, Rainer. *Virtuelle Realität und die Rolle von Haptic*,
<http://www.zwisler.de/scripts/haptics/haptics.html>, Juli 1998
- [3] Way, Barner. *Automatic Visual To Tactile Translation, Part 1: Human Factors, Access Methods and Image Manipulation*, IEEE Transactions on Rehabilitation Engineering, March, 1997.
- [4] Lederman, Klatzky. *Hand movements: a window into haptic object recognition*, Cognitive Psychology 19(3), 1987
- [5] Hajian, Howe. *Biomechanics of Manipulation: Grasping the Task at Hand*, in J. Winters and P. Crago, eds, Neural Control of Posture and Movement, Springer-Verlag, 2000, pp.382-389.
- [6] Pawluk, Howe. *Dynamic Contact of the Human Fingerpad Against a Flat Surface*, Journal of Biomechanical Engineering, Volume 121, pp. 605-611, 1999
- [7] Herndon, van Dam, Gleicher. *Workshop on the Challenges of 3D Interaction*, SIGCHI Bulletin, October 1994, Vol. 26
- [8] Hinckley, K. (1997): *Haptic Issues for Virtual Manipulation*, PhD-thesis, University of Virginia
<http://citeseer.nj.nec.com/hinckley97haptic.html>
- [9] Fröhlich, Platte, Wind, Wesche, Göbel. *Cubic-Mouse-Based Interaction in Virtual Environments*, IEEE Computer Graphics and Applications, Juli 2000, Vol. 20
- [10] Roessler, Grantz. *Performance Evaluation of Input Devices in Virtual Environments*,
<http://vr.iao.fhg.de/papers/3d-input-devices.pdf>, kein Jahr
- [11] Anderson, *Kognitive Psychologie – Eine Einführung*. Spektrum Verlag, 1996
- [12] Norman, *The Design of Everyday Things*, MIT Press, 1988
- [13] Weiner, *Motivationspsychologie*. Beltz Psychologie Verlags Union, 1994
- [14] LaViola, *Whole-Hand and Speech Input in Virtual Environments*,
<http://citeseer.nj.nec.com/laviola99wholehand.html>, 1999
- [15] Balakrishnan, Hickley. *The Role of Kinesthetic Reference Frames in Two Handed Input Performance*, CHI-Letters--UIST'99, 171-178
- [16] Slater, Linakis, Usoh, Kooper. *Immersion, Presence and Performance in Virtual Environments: An Experiment with Tri-dimensional Chess*. ACM Virtual Reality

- Software and Technology (VRST), Mark Green (ed.), ISBN: 0-89791-825-8, p 163-172
- [17] Newell, Ernst, Tjan, Bültboff, *Viewpoint Dependence in Visual and Haptic Object Recognition*. Psychological Science Vol.12, No.1 Januar 2001, 37-42
- [18] Balakrishnan, Hickley, *Symmetric Bimanual Interaction*, CHI 2000, 1-6 April 2000, 33-40
- [19] Smith, Stuerzlinger, *Integration of Constraints into a VR Environment*. Virtual Reality International Conference, Lava Virtual 2001, May 16-18
- [20] Vroubel, Markopoulos, Bekker, *FRIDGE: exploring intuitive interaction styles for home information appliances*. <http://citeseer.nj.nec.com/vroubel01fridge.html>, 2001, noch nicht gedruckt
- [21] Gribnau, Hennessey, *Comparing single- and two-handed 3D input for a 3D object assembly task*. CHI'98, Summary, ACM, New York, 1998, pp. 233-234
- [22] Fitzmaurice, Ishii, Buxton, *Bricks: Laying the Foundations for Graspable User Interfaces*. CHI'95, ACM Proceedings, 442-449
- [23] Ware, Rose, *Rotating Virtual Objects with Real Handles*. ACM Transactions on CHI, 6(2) 162-180
- [24] Bowman, Kruijff, LaViola, Poupyrev, *An Introduction to 3D User Interface Design*. Presence Vol.10, Number 1, 2001, 96-108
- [25] MacLean, *Designing with Haptic Feedback*. Proceedings of IEEE Robotics and Automation (ICRA'2000), San Francisco, CA, April 22-28
- [26] Habermann, *Kompodium 2/3-dimensionales Gestalten*, WS96/97 – SS97.
- [27] Bolt, *“put-that-there” - Voice and Gesture at the Graphics Interface*. SIGGRAPH '80 Proceedings, Vol.14, 262-270, July 1980
- [28] Mackay, *Is Paper Safer? The Role of Paper Flight Strips in Air Traffic Control*. ACM Transactions on Computer-Human Interaction, Vol.6, No.4, 1999.
- [29] Weiser, *The Computer For The 21st Century*. <http://www.ubiq.com/hypertext/weiser/SciAmDraft3.html>, 1989.
- [30] Szalavari, Gervautz, *The Personal Interaction Panel – a two Handed Interface for Augmented Reality*. Proceedings of EuroGraphics'97, pp.335-345, September 1997
- [31] Pouprey, Weghorst, Billinghamorst, Ichikawa. *A Study of techniques for selecting and positioning object in immersive VEs: effects of distance, size, and visual feedback*. Proceedings of ACM CHI '98

- [32]Arsenault, Ware, *Eye-Hand-Coordination with Force-Feedback*. CHI-Letters Vol.2, issue 1
- [33]Shaw, Green, THRED: *A Two-Handed Design System*. Multimedia Systems Journal, 3(6), November 1995
- [34]Cutler, Fröhlich, Hanrahan. *Two Handed Direct Manipulation on the Responsive Workbench*. 1997 Symp. on Interactive 3D Graphics, Providence, RI, (1997), pp. 107-114
- [35] Lindeman, Silbert, Hahn. *Towards Usable VR: An Empirical Study of User Interfaces for Immersive Virtual Environments*. <http://citeseer.nj.nec.com/26648.html>, kein Jahr
- [36]Hickley, Pausch, Goble, Kassell, *A Survey of Design Issues in Spatial Input*. In Proceedings of the ACM Symposium on User Interface Software and Technology, pages 213-222, 1994
- [37]Burdea, *Haptic Issues in Virtual Environments*. Proceedings of Computer Graphics Interantional 2000, Geneva Switzerland, Juni 2000

