

Johann Wolfgang Goethe-Universität  
Frankfurt am Main



Diplomarbeit

---

# **Navigations- und Interaktionstechniken**

durchgeführt am Fachbereich Biologie und Informatik  
von Thomas Paes

## ***Prüfer***

Prof. Dr.-Ing. Detlef Krömker  
Professur für Graphische Datenverarbeitung

## ***Betreuer***

Dipl.-Wirtsch.-Inform. Daniel F. Abawi  
Dipl.-Wirtsch.-Inform. Silvan Reinhold

Dezember 2002



## **Eidesstattliche Erklärung**

Hiermit erkläre ich an Eides statt, dass die vorliegende Diplomarbeit ohne unzulässige Hilfe und ausschließlich unter Verwendung der angegebenen Literatur angefertigt wurde.

Frankfurt am Main, den 19. Dezember 2002

---

Thomas Paes



## **Aufgabenstellung**

Die Möglichkeiten, die aktuelle Systeme zur Navigation im dreidimensionalen Raum bieten, sowie die Interaktionsmetaphern, die diese Systeme bei der Interaktion mit Objekten im Raum verwenden, erscheinen vielen Anwendern nicht intuitiv genug.

Die erhöhte Zahl der Freiheitsgrade möglicher Bewegungen oder Interaktionen machen daher eine Neu- oder Weiterentwicklung von Navigations- und Interaktionsmethoden notwendig.

Ziel der Arbeit ist es, vorhandene Interaktions- und Navigationsmethoden zu untersuchen, sowie ihre Stärken und Schwächen zu analysieren, um daraus Ideen und Verbesserungspotentiale für neuartige Konzepte zu gewinnen.

Die erstellten Methoden sollen anhand einer prototypischen Implementierung ihre Handhabbarkeit in Bezug auf Navigation und Interaktion mit einem dreidimensionalen Zauberwürfel (Rubik's Cube) unter Beweis stellen.

# Inhalt

<b>Eidesstattliche Erklärung .....</b>	<b>iii</b>
<b>Aufgabenstellung .....</b>	<b>v</b>
<b>Abbildungsverzeichnis .....</b>	<b>x</b>
<b>Tabellenverzeichnis .....</b>	<b>xiii</b>
<b>1. Einleitung.....</b>	<b>1</b>
1.1 Motivation.....	1
1.2 Ziele .....	2
1.3 Aufbau der Arbeit .....	3
<b>2. Grundlagen.....</b>	<b>5</b>
2.1 Begriff der Metapher.....	5
2.1.1 Definition .....	5
2.1.2 Metaphern in der menschlichen Kommunikation.....	6
2.2 Rubik's Cube.....	7
2.2.1 Aufbau.....	7
2.2.2 Geschichte .....	8
2.2.3 Begriffsdefinition .....	9
<b>3. Menschliche Faktoren im UI-Design.....</b>	<b>11</b>
3.1 Aspekte der Wahrnehmungspsychologie.....	12
3.1.1 Die menschlichen Sinne.....	12
3.1.2 Informationsverarbeitung im Gehirn.....	13
3.1.3 Modellbildung als Grundlage des Lernprozesses .....	16
3.2 Metaphern für die Benutzungsschnittstelle.....	16
3.2.1 Metaphern zur Unterstützung der Modellbildung.....	16
3.2.2 Anforderungen an den Metaphern-Entwurf.....	18
3.2.3 Probleme beim Einsatz von Metaphern .....	20
3.3 Entwurfsprinzipien der Human Computer Interaction.....	22
<b>4. Untersuchung von Interaktionstechniken für den 3D-Raum .....</b>	<b>27</b>

4.1	Interaktion in virtuellen 3D-Welten .....	27
4.1.1	Geschichtliche Entwicklung .....	27
4.1.2	3D-Eingabegeräte .....	28
4.1.3	Eigenschaften der 3D-Interaktion.....	29
4.2	Besondere Bedingungen der Desktop-VR.....	30
4.2.1	Freiheitsgrade der 3D-Interaktion .....	30
4.2.2	Desktop-VR .....	32
4.2.3	Desktop-VR und Freiheitsgrade .....	32
4.2.4	Funktion der 3D-Metapher in der Desktop-VR.....	34
4.3	Grundlegende Interaktionsaufgaben im 3D-Raum.....	36
4.3.1	Navigation .....	36
4.3.2	Manipulation.....	42
<b>5.</b>	<b>Interaktion mit dem virtuellen Rubik's Cube .....</b>	<b>47</b>
5.1	Interaktionsformen am Rubik's Cube.....	47
5.1.1	Inspektion der Würfelseiten.....	47
5.1.2	Manipulation am Rubik's Cube.....	48
5.2	Der Rubik's Cube: Ein geeignetes Versuchsobjekt?.....	48
5.2.1	Allgemeinheit, Übertragbarkeit .....	48
5.2.2	Angemessene Komplexität .....	49
5.2.3	Vergleich von realer und virtueller Schnittstelle.....	50
5.2.4	Einschränkungen .....	51
5.3	Inspektionsmetaphern .....	52
5.3.1	Virtual Sphere.....	52
5.3.2	Eckpunkt-Metapher .....	54
5.3.3	ColorMap-Metapher .....	58
5.4	Rotationsmetaphern .....	61
5.4.1	Rubik's Online-Metapher .....	61
5.4.2	Zweipunkt-Metapher .....	63
5.4.3	Spotlight-Metapher.....	66
5.4.4	Bezugsrahmen-Metapher .....	71
<b>6.</b>	<b>Implementierung .....</b>	<b>75</b>
6.1	Java3D-API .....	75
6.2	Darstellung einzelner Konzepte.....	76
6.2.1	Behavior-Konzept von Java-3D .....	76
6.2.2	Picking-Mechanismen .....	77
6.3	Implementierung der Interaktionskonzepte .....	78
6.4	Darstellung ausgewählter Klassen der vorliegenden Implementierung .....	78

<b>7.</b>	<b>Empirische Untersuchung</b> .....	<b>81</b>
7.1	Vorbereitungen.....	81
	7.1.1 Berücksichtigung der Komplexität des Rubik's Cube.....	81
	7.1.2 Unterteilung der Benutzerstudie.....	82
	7.1.3 Wahl geeigneter Metaphern-Kombinationen.....	83
7.2	Durchführung.....	85
	7.2.1 Einteilung der Benutzergruppe.....	85
	7.2.2 Erläuterung der Metaphern und Training.....	87
	7.2.3 Durchführen der Repeater-Durchgänge.....	87
	7.2.4 Subjektive Bewertung der Metaphern.....	88
	7.2.5 Vergleich zwischen realem und virtuellem Zauberwürfel.....	89
7.3	Messwerte Repeater-Versuche.....	89
	7.3.1 Inspektion: Intuitivität.....	90
	7.3.2 Inspektion: Effizienz.....	91
	7.3.3 Inspektion: Erlernbarkeit.....	92
	7.3.4 Rotation: Intuitivität.....	93
	7.3.5 Rotation: Effizienz.....	94
	7.3.6 Rotation: Erlernbarkeit.....	95
	7.3.7 Zeitliche Performance.....	96
	7.3.8 Anzahl Inspektionen.....	97
	7.3.9 Anzahl Fehler.....	97
7.4	Messwerte „Lösen einer Würfelseite“.....	98
	7.4.1 Teilnehmer 2.....	99
	7.4.2 Teilnehmer 3.....	100
	7.4.3 Teilnehmer 8.....	101
7.5	Auswertung.....	102
	7.5.1 Virtual Sphere.....	102
	7.5.2 Eckpunkte-Metapher.....	103
	7.5.3 ColorMap-Metapher.....	104
	7.5.4 Rubik's Online-Metapher.....	105
	7.5.5 Zweipunkt-Metapher.....	106
	7.5.6 Spotlight-Metapher.....	107
	7.5.7 Bezugsrahmen-Metapher.....	108
	7.5.8 Lösen einer Würfelseite.....	109
<b>8.</b>	<b>Zusammenfassung und Ausblick</b> .....	<b>111</b>
8.1	Zusammenfassung.....	111
8.2	Ausblick und Erweiterungsmöglichkeiten.....	112
	<b>Anhang A: Glossar</b> .....	<b>115</b>



<b>Anhang B: Fragebogen .....</b>	<b>117</b>
<b>Anhang C: Kommentiertes Literaturverzeichnis .....</b>	<b>123</b>

## Abbildungsverzeichnis

Abbildung 4.1: Freiheitsgrade der Rotation [hand93].....	31
Abbildung 4.2: Die Funktion der 3D-Metapher [kettner93] .....	35
Abbildung 4.3: Head-butt Zoom-Verfahren [mine97] .....	42
Abbildung 4.4: Die Virtual Sphere-Metapher [kettner93] .....	44
Abbildung 5.1: Rotationen am Rubik's Cube [rubik02] .....	50
Abbildung 5.2: Mausbewegung bei der Virtual Sphere-Metapher .....	53
Abbildung 5.3: Unterschiedliche Ansichten des Rubik's Cube .....	55
Abbildung 5.4: Funktion der Eckpunkt-Metapher .....	56
Abbildung 5.5: Funktionsweise der ColorMap-Metapher.....	60
Abbildung 5.6: Die Rubik's Online-Metapher .....	62
Abbildung 5.7: Varianten der 3-Punkt-Rotationsmetapher [kettner93] .....	64
Abbildung 5.8: Die einzelnen Phasen der Zweipunkt-Metapher .....	66
Abbildung 5.9: Spotlight-Metapher, Rotation der unteren Würfelscheibe .....	67
Abbildung 5.10: Spotlight-Metapher, Rotation einer seitlichen Würfelscheibe ...	68
Abbildung 5.11: Erweiterung der Spotlight-Metapher.....	70
Abbildung 5.12: Bezugsrahmen-Metapher: Bewegen der Fingerpunkte .....	72
Abbildung 5.13: Bezugsrahmen-Metapher: Rotation einer Teilscheibe .....	73
Abbildung 5.14: Problem der Operatorzuordnung.....	74
Abbildung 7.1: gemittelter Intuitivitätswert mit Standardabweichung .....	90
Abbildung 7.2: gemittelter Effizienzwert mit Standardabweichung.....	91
Abbildung 7.3: gemittelter Wert für die Erlernbarkeit mit Standardabweichung .	92
Abbildung 7.4: gemittelter Intuitivitätswert mit Standardabweichung .....	93
Abbildung 7.5: gemittelter Effizienzwert mit Standardabweichung.....	94
Abbildung 7.6: gemittelter Wert für die Erlernbarkeit mit Standardabweichung .	95
Abbildung 7.7: zeitliche Performance gemittelt mit Standardabweichung.....	96

Abbildung 7.8: Zeit für die Lösung einer Seite, Teilnehmer 2 .....	99
Abbildung 7.9: benötigte Anzahl von Rotationen, Teilnehmer 2 .....	99
Abbildung 7.10: Zeit für die Lösung einer Seite, Teilnehmer 3 .....	100
Abbildung 7.11: benötigte Anzahl von Rotationen, Teilnehmer 3 .....	100
Abbildung 7.12: Zeit für die Lösung einer Seite, Teilnehmer 8 .....	101
Abbildung 7.13: benötigte Anzahl von Rotationen, Teilnehmer 8 .....	101



## Tabellenverzeichnis

Tabelle 4.1: kontrollierte Freiheitsgrade bei 3D-Operationen [chen88].....	33
Tabelle 7.1: Intuitivitätsbewertung der Inspektionsmetaphern.....	90
Tabelle 7.2: Effizienzbewertung der Inspektionsmetaphern.....	91
Tabelle 7.3: Bewertung der Erlernbarkeit der Inspektionsmetaphern .....	92
Tabelle 7.4: Intuitivitätsbewertung der Rotationsmetaphern.....	93
Tabelle 7.5: Effizienzbewertung der Rotationsmetaphern.....	94
Tabelle 7.6: Bewertung der Erlernbarkeit der Rotationsmetaphern .....	95
Tabelle 7.7: zeitliche Performance der Metapher-Kombinationen .....	96
Tabelle 7.8: Anzahl der Inspektionsoperationen während 20 Rotationen .....	97
Tabelle 7.9: Anzahl der fehlerhaften Rotationen bei 20 Durchgängen.....	97



# 1. Einleitung

Die Leistungsfähigkeit moderner Grafikhardware, mit der heute selbst PCs der Einstiegsklasse ausgestattet sind, erreicht ein Niveau, auf dem sich aufwändig gestaltete 3D-Szenen in kürzester Zeit berechnen lassen.

Neben ihrem Einsatz in Wissenschaft und Technik beschränkt sich die Anwendung dieser leistungsfähigen Grafikprozessoren aber vor allem auf den Bereich von 3D-Computerspielen, die mit immer neuen Spezialeffekten um Käufer werben. Auf dem Gebiet der 3D-Grafik ist in den letzten Jahren ein heißumkämpfter Markt entstanden und ein Ende der Entwicklung ist nicht abzusehen.

Den mit jeder neuen Chip-Generation einhergehenden Leistungssteigerungen auf Hardware-Ebene steht auf dem Gebiet des User Interface-Designs keine vergleichbare Entwicklung gegenüber. [pierce00] spricht in diesem Zusammenhang vom sogenannten *WIMP-Plateau*, auf dem sich die Entwicklung von Benutzungsschnittstellen in den letzten 20 Jahren bewegt. Seit der Vorstellung des Xerox Star im Jahre 1981, der Geburtsstunde moderner graphischer Benutzeroberflächen, wird die Bedienung von Computeranwendungen im wesentlichen von zweidimensionalen, windowbasierten Mauszeiger-Systemen dominiert [preim99]<sup>1</sup>.

## 1.1 Motivation

Der Mensch lebt und interagiert in einer dreidimensionalen Welt und so sehen viele Autoren den natürlichen nächsten Schritt bei der Entwicklung moderner Benutzungsschnittstellen in einer Erweiterung um genau diese dritte Dimension (siehe zum Beispiel [pierce00] [bowman01]). Gerade bei der Darstellung von umfangreichen multimedialen Daten in unterschiedlichen Fenstern einer Windows-Oberfläche werden die Grenzen einer solchen zweidimensionalen Schnittstelle mit ihren vielstufigen Menüs und Scrollbars ganz unmittelbar deutlich.

Auf dem Weg zu graphischen 3D-User Interfaces muss jedoch eine entscheidende Frage beantwortet werden, auf die klassische Window-Systeme keine Antwort

---

<sup>1</sup> die erwähnten Interaktionselemente standen daher auch Pate für das Akronym *WIMP*: die Anfangsbuchstaben stehen für *Window*, *Icons* und *MousePointer*.

bieten: Wie ist die Interaktion mit einer 3D-Applikation zu gestalten, sodass die Bedienung vom Anwender als intuitiv und effizient empfunden wird? Die Bedienung einer räumlichen Anwendung wird nur dann als nützlich und angenehm angesehen werden, wenn der Anwender in die Lage versetzt wird, sich im virtuellen Raum angemessen zu orientieren und mit den im Raum enthaltenen Objekten zu interagieren [tan01].

In den letzten Jahren wurde eine Vielzahl von Systemen entwickelt, die es dem Anwender erlauben, in die virtuelle Welt gleichsam einzutauchen und ohne die Schnittstelle Maus oder Bildschirm ganz unmittelbar mit der dargestellten Szene zu interagieren. Trotz einzelner Entwicklungen der letzten Zeit, die zu der Hoffnung Anlass geben, dass sich diese Systeme in Zukunft weiter verbreiten werden, steht einer solchen Verbreitung heute vor allem noch ihr hoher Preis und Administrationsaufwand entgegen<sup>2</sup>.

3D-Anwendungen, die auf herkömmlichen Desktop-Rechnern ablaufen können und sich intuitiv über Maus und Tastatur bedienen lassen, stellen vor diesem Hintergrund eine ökonomische Alternative zu aufwändigen Systemen zur Erzeugung von immersiven virtuellen Welten dar [kettner94].

Beim Entwurf solcher Systeme kommt es insbesondere darauf an, die spezifischen Eigenschaften der menschlichen Wahrnehmung zu berücksichtigen, sodass sich das System dem Anwender anpasst und nicht umgekehrt. Ein Prinzip, das bereits beim Entwurf der ersten WIMP-Systeme erfolgreich angewendet wurde, erscheint hier besonders vielversprechend: der gezielte Einsatz von *metaphorischen Konzepten* in der Benutzungsschnittstelle. Gerade bei der Interaktion mit räumlichen Welten über klassische 2D-Eingabegeräte wird der Einsatz von Metaphern als geeignetes Verfahren zur Überbrückung der Kluft zwischen 2D-Eingabe und 3D-Interaktion angesehen [kettner93] [preim99].

## 1.2 Ziele

Im Rahmen dieser Arbeit sollen 3D-Metaphern zur Navigation und Interaktion in einer dreidimensionalen Umgebung entworfen und anhand einer prototypischen Implementierung auf ihre Tauglichkeit hin untersucht werden.

---

<sup>2</sup> ein Beispiel für kostengünstige immersive VR-Systeme findet sich in [ix02]



Im Rahmen des Prototypen sind weiterhin bereits bestehende Referenzmetaphern zur Navigation und Interaktion zu implementieren, um auf diese Weise eine Vergleichbarkeit der eigenen Konzepte zu gewährleisten.

Die gefundenen Interaktionswerkzeuge sollen verallgemeinerbar sein und sich einfach auf andere Anwendungszusammenhänge übertragen lassen.

### **1.3 Aufbau der Arbeit**

Kapitel 2 führt den für diese Arbeit grundlegenden Begriff der Metapher ein und beschreibt den prinzipiellen Aufbau des Rubik's Cube, der die Grundlage für eine spätere prototypische Umsetzung der gefundenen Konzepte darstellt.

In Kapitel 3 werden zunächst einzelne Aspekte der Wahrnehmungspsychologie beschrieben, bevor auf dieser Grundlage die besondere Bedeutung von Metaphern beim Einsatz in modernen Benutzungsschnittstellen erläutert wird. Abschließend werden wichtige Prinzipien der Human Computer Interaction erläutert.

Kapitel 4 beschäftigt sich mit den besonderen Bedingungen der 3D-Interaktion und geht dabei insbesondere auf die Eigenschaften der Desktop-VR ein, welche als Zielplattform für die Implementierung dient. Ausgewählte Metaphern für die 3D-Interaktion werden dargestellt und bilden die Grundlage für die Konzeption der Bedienmetaphern im folgenden Kapitel.

In Kapitel 5 wird zunächst untersucht, inwieweit sich der Rubik's Cube für eine Untersuchung von räumlichen Interaktionstechniken eignet, die abzubildenden Bedienhandlungen analysiert und anschließend auf Grundlage der in Kapitel 3 und 4 gewonnenen Erkenntnisse eigene Interaktionsmetaphern entwickelt. Hierbei wird insbesondere überprüft, welches Potential die einzelnen Konzepte bei einer Übertragung auf andere Anwendungsbereiche bieten.

Das Kapitel 6 erläutert grundlegende Aspekte der Implementierung; ausgewählte Konzepte werden beschrieben und wichtige Klassen der Implementierung in Kürze dargestellt.

Kapitel 7 beschreibt die Planung und Durchführung der Benutzerstudie. Die im Rahmen der Studie gewonnenen Ergebnisse werden dargestellt und ausführlich diskutiert.

Abgeschlossen wird die Arbeit in Kapitel 8 mit einer Zusammenfassung der wichtigsten Ergebnisse und einer Darstellung einzelner Erweiterungsansätze, auf denen nachfolgende Arbeiten aufbauen können.

## 2. Grundlagen

### 2.1 Begriff der Metapher

Der Begriff der Metapher steht im Rahmen dieser Arbeit an einer zentralen Position; im Folgenden sollen daher die grundlegenden Funktionsprinzipien dieses Konzepts erläutert werden. Wir beschränken uns an dieser Stelle zunächst auf die Beschreibung der *verbalen Metapher* und geben Definitionen des Metaphern-Begriffs aus der Sprachwissenschaft wieder. In Abschnitt 3.2 wird näher auf die Bedeutung von *Interface-Metaphern* [preim99] beim Entwurf von Benutzungsschnittstellen eingegangen werden.

#### 2.1.1 Definition

Bei der Metapher handelt es sich um eine aus der Sprachwissenschaft bekannte rhetorische Figur, „die einen bildhaften Ausdruck für einen Gegenstand oder einen abstrakten Begriff darstellt“ [encarta01]. Da es sich beim Begriff der Metapher um ein Konzept mit einer Reihe von unterschiedlichen Aspekten handelt, seien einer zusammenfassenden Diskussion zunächst weitere gängige Definitionen vorangestellt:

[dtv80] definiert die Metapher als eine „Redewendung, bei der statt der eigentlichen Bezeichnung eine uneigentliche oder übertragene gebraucht wird“ und gibt zwei Beispiele: *Hafen* zur Bezeichnung einer Zuflucht und *kalt* zur Charakterisierung einer gefühllosen Person.

Ergänzend hierzu wird in [bertelsmann92] darauf hingewiesen, dass beim metaphorischen Vergleich „das Gemeinte durch eine Vorstellung (meist ein Bild) zum Ausdruck gebracht wird, die aus einem anderen Bereich stammt und keine reale Beziehung zum Gemeinten hat“.

Schließlich beschreibt [duden74] die Metapher als eine „bildliche Übertragung, besonders eines konkreten Begriffs auf einen abstrakten, auf Grund eines Vergleichs“.

Die Metapher setzt verschiedene Bedeutungszusammenhänge zueinander in *Beziehung* – ein Konzept, das bereits in der etymologischen Wurzel des Wortes angedeutet wird: *Metaphora* bezeichnet im Griechischen eine *Übertragung*.

Durch die Metapher wird die Bedeutung eines Begriffs auf einen anderen bildhaft übertragen. Die Gebiete, aus denen diese beiden in Beziehung gestellten Begriffe stammen, werden oft als Quell- und Zieldomäne bezeichnet [väänänen95]. Der Begriff der Zieldomäne wird auf diese Weise durch Zusammenhänge in der Quelldomäne veranschaulicht.

Eine Funktion der Metapher kann es hierbei sein, einen abstrakten oder bislang unbekanntem Begriff einer Zieldomäne durch einen vertrauten Zusammenhang in einer konkreten Quelldomäne zu erläutern.

### **2.1.2 Metaphern in der menschlichen Kommunikation**

Auf Grund ihrer veranschaulichenden Funktion spielen Metaphern eine wichtige Rolle beim Erlernen neuer Wissensbereiche [kettner93]. So werden Metaphern beispielsweise eingesetzt, um schwer zu beobachtende Zusammenhänge anschaulich zu machen [preim99]: Die Bewegung von Elektronen in einem elektrischen Leiter unter dem Einfluss einer wirkenden Potentialdifferenz wird oft über das *Strömen von Wasser* in einem Fluss veranschaulicht oder der Aufbau eines Atoms mit der Struktur unseres Sonnensystems verglichen<sup>3</sup>.

Im Gegensatz zur Analogie ist es bei der Metapher dabei nicht entscheidend, dass alle Aspekte der Quell-Domäne mit jenen der Ziel-Domäne übereinstimmen. So sind die um den Atomkern kreisenden Elektronen nicht ihrerseits wieder von weiteren Objekten umgeben, so wie im Sonnensystem einzelne Planeten von ihren Monden umlaufen werden.

Metaphern erklären daher unter Umständen nur *Teilaspekte* eines komplexeren Zusammenhangs und wollen nicht wörtlich genommen werden [kettner93]; solange die Metapher dem Ziel dienlich ist, einen abstrakten Zusammenhang plausibel

---

<sup>3</sup> In [tsialtas01] werden am Beispiel des Begriffs *Computervirus* die Zusammenhänge der Quelldomäne jenen der Zieldomäne ausführlich gegenüber gestellt und dabei Gemeinsamkeiten beziehungsweise Unterschiede herausgearbeitet. Für eine Zusammenstellung weiterer rhetorischer Figuren siehe ebenfalls dort.

zu machen, werden diese Unterschiede jedoch billigend in Kauf genommen und wirken sich nicht störend auf das Verständnis aus [kuhn95].

Metaphern sind ein integraler Bestandteil der menschlichen Kommunikation [preim99] [marcus02a]. Wie tief die Verankerung von Metaphern in unserer Alltagssprache ist, lässt sich anschaulich am Begriff der *dominierenden Metaphorik* erläutern, der besagt, dass sich ein großer Teil der zu einer bestimmten Zeit gebräuchlichen Metaphern an den Themengebieten orientiert, die sich in der jeweiligen Periode im Mittelpunkt des öffentlichen Interesses befanden [brockhaus84]. Herrschte beispielsweise Anfang des 20. Jahrhunderts noch eine dem Militärischen entlehnte Metaphorik vor („eine Schlacht schlagen“), so entstammen Metaphern der jüngeren Zeit oftmals den Bereichen Sport oder Technik („sich an die Spitze setzen“, „unter Strom stehen“) [debatin97].

Erkenntnisse der Kognitionsforschung aus den vergangenen Jahren legen darüber hinaus nahe, dass weite Teile unseres Denkens metaphorisch ablaufen und Metaphern damit eine Grundlage für das menschliche Verstehen an sich darstellen [väänänen95]<sup>4</sup>.

## 2.2 Rubik’s Cube

Anhand des Rubik’s Cube [rubik02] – oder Zauberwürfels, wie er im Deutschen genannt wird – sollen die im Rahmen dieser Arbeit konzipierten Bedienmetaphern prototypisch umgesetzt und auf ihre Tauglichkeit hin überprüft werden.

Im Folgenden werden daher zunächst der prinzipielle Aufbau sowie grundlegende Eigenschaften des Würfels beschrieben und ein kurzer geschichtlicher Abriss gegeben. Vor der Konzeption neuer Metaphern wird in Abschnitt 5.2 zusätzlich untersucht, welche Eigenschaften den Cube als geeignetes Versuchsobjekt qualifizieren.

### 2.2.1 Aufbau

Der Rubik’s Cube besteht aus 26 Teilwürfeln, die über ein Gelenkstück im Inneren zu einem regelmäßigen 3x3x3-Quader zusammengesetzt sind. Jeder Teilwür-

---

<sup>4</sup> Für eine ausführliche Erläuterung der verschiedenen Theorien zum Metaphernbegriff – unter anderem Lakoff und Johnsons Theorie über Metaphern und das menschliche Denken – siehe ebenfalls in [väänänen95].

fel des Cubes lässt sich, gemeinsam mit den übrigen acht Teilwürfeln der jeweiligen Würfelscheibe, um jede der drei Raumachsen rotieren.

Die neun Teilflächen einer Würfelseite besitzen im geordneten Zustand des Rubik's Cube jeweils die gleiche Farbe; auf Grund der Mechanik im Inneren bleibt die Position von Mittelwürfeln sich gegenüberliegender Würfelseiten bei allen Drehbewegungen unverändert. Auf diese Weise definiert die Farbe des Mittelwürfels im ungeordneten Zustand die Farbe, welche die ihn umgebenden Teilwürfel besitzen müssen, um diese Seite zu lösen<sup>5</sup>.

Die sich aus den möglichen Drehbewegungen ergebenden vielfältigen Freiheitsgrade machen den Rubik's Cube zu einem anspruchsvollen 3D-Puzzle: Sobald man den Würfel durch eine Reihe von zufälligen Rotationsbewegungen ungeordnet hat, ist es äußerst schwierig, alle sechs Seiten des Würfels zu lösen und damit die ursprüngliche Konfiguration wiederherzustellen.

Allein die astronomisch große Zahl von über  $4,3 \cdot 10^{19}$  verschiedenen Konfigurationen, in die der Würfel durch Drehbewegung der Teilscheiben gebracht werden kann, lässt die hinter der einfachen geometrischen Struktur verborgene mathematische Komplexität erahnen [rubik02].

### 2.2.2 Geschichte

Erfunden wurde der Rubik's Cube im Jahre 1974 von *Ernö Rubik*, einem Dozenten am Department of Interior Design an der Academy of Applied Arts and Graphics in Budapest.

Zu seinen Vorlesungen in diesen Jahren fertigte Rubik dreidimensionale Modelle an, über die er seinen Studenten die behandelten Themen veranschaulichte. Sein besonderes Interesse galt hierbei einfachen Grundformen, die durch Vervielfältigung und Manipulation eine Fülle neuer und komplexer Formen hervorbringen [rubik98]<sup>6</sup>.

---

<sup>5</sup> Die Verteilung der Farbseiten wurde im Laufe der Jahre verändert; beim aktuellen Zauberwürfel liegt die gelbe Seite immer gegenüber der weißen, Blau gegenüber von Grün und Orange gegenüber von Rot.

<sup>6</sup> Ein anschauliches Beispiel für dieses Grundprinzip in *zwei Dimensionen* stellt das alte chinesische Legespiel Tangram dar, bei dem ein Quadrat in sieben geometrische Grundformen geteilt wird (fünf Dreiecke, ein Parallelogramm und ein kleines Quadrat). Durch Umlegen dieser Grund-

„*We turn the Cube and it twists us...*“ – Ernő Rubik

Schnell wurde klar, welche Faszination sein neues Modell auf Freunde und Studenten ausübt und im Jahre 1977 lagen die ersten Zauberwürfel in den Regalen der lokalen Spieläden. Verglichen mit seinem späteren weltweiten Erfolg, verbreitete sich der Rubik's Cube zunächst jedoch relativ langsam<sup>7</sup>. Erst als zwei Jahre nach Markteinführung die ersten Würfel in den Straßen und Cafés Ungarns auftauchten, übernahm ein amerikanischer Investor den internationalen Vertrieb.

Der Rubik's Cube startete seinen Siegeszug um die Welt: Im März 1981 erschien er auf dem Titelbild von *Scientific American*, verkaufte sich in drei Jahren mehr als 100 Millionen mal und wurde zu einer Ikone der 80er Jahre. Die starke Nachfrage führte zur Verbreitung einer Vielzahl von Raubkopien; ab 1983 brach der Markt für Zauberwürfel komplett zusammen.

Erst in der zweiten Hälfte der 90er Jahre erlebte der Rubik's Cube ein Comeback, 1996 wurden allein in den USA wieder mehr als 300.000 Zauberwürfel verkauft [rubik98] – die Faszination für Erno Rubiks Erfindung ist bis heute ungebrochen.

### 2.2.3 Begriffsdefinition

Im Abschnitt über den Aufbau des Zauberwürfels wurde bereits auf die besondere Bedeutung des *Mittelwürfels* hingewiesen, durch den die Farbe der jeweiligen Seite definiert wird. An dieser Stelle sollen zwei weitere Begriffe eingeführt werden, um die Bezeichnung von einzelnen Teilwürfeln im weiteren Verlauf der Arbeit zu erleichtern:

- *Eckwürfel*

Ein Teilwürfel des Rubik's Cube heißt *Eckwürfel*, wenn er – wie der Name bereits andeutet – an einer der acht *Ecken* platziert ist. Ein Eckwürfel besitzt drei farblich markierte Außenflächen.

---

formen ergibt sich eine Fülle von neuen Formen (eine Online-Version findet sich unter [www.fonline.de/home/fo0126/spiele/tangram/denk12.htm](http://www.fonline.de/home/fo0126/spiele/tangram/denk12.htm)).

<sup>7</sup> Bereits die lange Zeitspanne von drei Jahren zwischen Erfindung und Produktion des Würfels lässt erahnen, welche Hürden Erno Rubik mit seinem Würfel im damals noch kommunistischen Ungarn zu überwinden hatte – Hindernisse, die auch in den Folgejahren die Verbreitung des Würfels in der westlichen Welt zunächst stark behindert haben.

- *Kantenwürfel*

Ein Teilwürfel des Rubik's Cube heißt *Kantenwürfel*, wenn er sich zwischen zwei Eckwürfeln befindet. Ein Kantenwürfel besitzt zwei farblich markierte Außenflächen.

Neben diesen Bezeichnungen für Teilelemente des Zauberwürfels sollen im Folgenden Begriffe für einzelne Interaktionshandlungen definiert werden.

Bei der Interaktion mit dem virtuellen Modell des Rubik's Cube kann man zwischen zwei grundlegenden Interaktionsmodi unterscheiden: Vor der Durchführung einer Rotationsbewegung muss sich der Spieler einen Überblick über die aktuelle Konfiguration des Würfels machen; dieser Vorgang kann mit Navigationsoperationen im virtuellen Raum verglichen werden – die besondere Bedingung beim Rubik's Cube besteht darin, dass hier nicht zwischen *räumlich verteilten Objekten* in einer 3D-Szene navigiert wird, sondern vielmehr die *verschiedenen Seiten* des Würfels *inspiziert* werden müssen.

Nach Analyse der Würfel-Konfiguration müssen durch geeignete Rotationsbewegungen Teilwürfel neu positioniert oder ausgerichtet werden. Diese *Manipulation* des Würfels kennzeichnet den zweiten Interaktionsmodus. Vergleichbar mit einer Manipulation von beliebigen Objekten einer 3D-Szene werden hierbei einzelne Objekt-Parameter von einem Anfangs- in einen Endzustand transformiert<sup>8</sup>.

Im Rahmen dieser Arbeit bezeichnet der Begriff *Inspektion* die Navigation um den Würfel und der Begriff *Manipulation* die Rotation einzelner Teilscheiben des Rubik's Cube.

---

<sup>8</sup> Die Manipulation des Rubik's Cube überführt den Würfel also von einer Teilflächen-Konfiguration in die nächste; als einfaches Beispiel einer grundlegenden Manipulation von 3D-Objekten sei hier auf die Skalierungsoperationen verwiesen, bei denen die *ursprüngliche Größe* eines Objekts über einen Skalierungsfaktor auf die *Zielgröße* abgebildet wird.



### 3. Menschliche Faktoren im UI-Design

Im vorangegangenen Kapitel wurden die für diese Arbeit grundlegenden Begriffe und Definitionen eingeführt. Bevor in Kapitel 4 die spezifischen Bedingungen von 3D-Benutzungsschnittstellen erläutert werden, soll an dieser Stelle zunächst auf den menschlichen Faktor in der Mensch-Computer-Interaktion eingegangen werden.

In der Literatur lassen sich viele Hinweise darauf finden, wie entscheidend eine besondere Berücksichtigung dieses „Faktors Mensch“ für den Erfolg von Computer-Anwendungen ist und wie oft auch heute noch genau dieser Faktor beim Entwurf neuer Systeme vernachlässigt wird [vDam97] [bowman01].

Die Beschäftigung mit dieser menschlichen Seite der Mensch-Computer-Interaktion ist jedoch nicht neu: Als eine sichere Bedienung der immer komplexer werdenden Maschinen und Apparate durch den Menschen nicht mehr sichergestellt war, begann bereits in den 40er Jahren eine intensive Forschung auf diesem Gebiet [butler95].

Im Laufe der Zeit wurde dem Gesichtspunkt der *Usability* einer Anwendung, also die Frage, wie benutzerfreundlich das User Interface gestaltet wurde, ein immer größeres Gewicht gegeben [butler95]. Über die Benutzungsschnittstelle werden die Daten und die Funktionen einer Anwendung bereitgestellt – kleine Nachlässigkeiten bei ihrem Design können einen entscheidenden Unterschied für die Intuitivität der Bedienung machen. Bei einem zunehmenden Konkurrenzdruck auf dem Software-Markt wird gutes UI-Design zu einem entscheidenden Erfolgskriterium [barrileaux01].

Intelligentes UI-Design, das sich an den spezifischen Merkmalen der menschlichen Wahrnehmung orientiert, kann die besonderen Fähigkeiten des Menschen unterstützen und Einschränkungen berücksichtigen, um auf diese Weise dem angestrebten Idealzustand näher zu kommen, bei dem der Benutzer weder über- noch unterfordert wird und sich voll auf die Lösung der gestellten Aufgaben konzentrieren kann [dix95] [kettner93].

Im Folgenden werden zunächst für das UI-Design relevante Erkenntnisse auf dem Gebiet der Wahrnehmungspsychologie beschrieben, anschließend die besondere Bedeutung der Metapher in diesem Kontext untersucht und darauf aufbauend grundlegende Entwurfsprinzipien für Benutzungsschnittstellen eingeführt.

### 3.1 Aspekte der Wahrnehmungspsychologie

#### 3.1.1 Die menschlichen Sinne

Im Alltag orientiert sich der Mensch vor allem mit seinem Sehsinn: Über 80% der vom Gehirn ausgewerteten Sinneseindrücke entfallen auf Informationen, die vom Auge geliefert werden. Das menschliche Gehirn ist darauf trainiert, ihm angebotene Bilder auszuwerten; Kommunikation erfolgt aus diesem Grund häufig effizienter, wenn Informationen in geeigneter Form visualisiert werden [balaguer91].

Der über die Augen vermittelte primäre Sinneseindruck wird ergänzt durch den Hör- und den Tastsinn, welche vor allem dann in den Vordergrund treten, wenn vom Auge – zum Beispiel bei Dunkelheit – keine hinreichenden Informationen geliefert werden<sup>9</sup>.

Um dem Anwender die Möglichkeit zu geben, neben dem visuellen Kanal auch diese zusätzlichen Sinne für der Interaktion mit der Anwendung zu nutzen, werden zunehmend sogenannte „multimodale Benutzungsschnittstellen“ entwickelt. Bei diesen Anwendungen besteht die Möglichkeit der Spracheingabe, auf Benutzereingaben wird Audio-Feedback erzeugt und über Force Feedback-Datenhandschuhe werden Gesten erfasst und haptische Eindrücke wiedergegeben<sup>10</sup> [hernon94b].

Auffällig bei der Interaktion mit einer graphischen Computer-Anwendung ist eine starke Asymmetrie in der Bandbreite zwischen Ein- und Ausgabe-Kanal. So können komplexe visuelle Darstellungen „mit einem Blick“ wahrgenommen werden: Der Anwender kann die angebotenen Informationen sehr schnell erfassen. Die

---

<sup>9</sup> für eine ausführliche Betrachtung des Tastsinns und seiner Bedeutung für Benutzungsschnittstellen siehe [theisinger02]

<sup>10</sup> In einer solchen multimodalen Schnittstelle müssen nicht alle genannten Features realisiert sein; die Betonung liegt hierbei vielmehr auf der Ergänzung der rein visuellen Kommunikation durch weitere Sinneseindrücke und -äußerungen – ein Verfahren, dass sich am natürlichen Weg des Menschen orientiert, mit der Welt zu interagieren [bowman01].

Interaktion mit der Anwendung erfolgt im Gegensatz dazu über einen Kanal mit relativ geringer Bandbreite. Um diesen Zusammenhang zu verdeutlichen, stelle man sich beispielsweise eine Hypermedia-Anwendungen vor, in der ein einzelner Mausklick zur Darstellung umfangreicher neuer Informationen führen kann [ware99].

### 3.1.2 Informationsverarbeitung im Gehirn

Die von den menschlichen Sinnen wahrgenommenen sensorischen Daten werden im Gehirn weiterverarbeitet und interpretiert. Neben den bereits beschriebenen Zusammenhängen des visuellen Systems kommen beim UI-Design weitere Eigenschaften des menschlichen Gehirns zum Tragen, die im Folgenden beschrieben werden<sup>11</sup>.

Ein wichtiger Zusammenhang der Hand-Auge-Koordination wurde von Fitts im Jahre 1954 beschrieben: Das nach ihm benannte *Fitts' Law* setzt die Zeit für eine motorische Bewegung in einen linearen Zusammenhang zur Aufgabenschwierigkeit. Im Experiment wurden die Zeiten gemessen, die Probanden benötigten, um mit einem Stift wechselseitig in zwei schmale Streifen zu zeigen [kettner93]. Mathematisch ausgedrückt lautet der Zusammenhang:

$$SelectionTime = a + b \log_2(D/W + 1.0)$$

Der logarithmische Term gibt die Aufgabenschwierigkeit wieder ( $D$  steht für *distance*, die Entfernung zum anzusteuern Ziel und  $W$  für *width*, die Ausdehnung des Zielbereichs),  $a$  und  $b$  sind empirisch bestimmte Konstanten. Die Erkenntnisse aus Fitts' Versuchen lassen sich auf Positionierungs- und Selektionsaufgaben mit einer Computer-Maus übertragen, um auf diese Weise Erkenntnisse für das Design von Benutzungsschnittstellen abzuleiten (grundlegende Prinzipien des UI-Designs werden in Abschnitt 3.3 beschrieben).

Ein wichtiger Faktor, der bei der Anwendung von Fitts' Gesetz auf die Interaktion mit Computersystemen zu berücksichtigen ist, stellt die Verzögerungszeit der Sy-

---

<sup>11</sup> Der menschlichen Wahrnehmung liegen hochkomplizierte Vorgänge zugrunde, die von der Wissenschaft zum Teil noch nicht erschöpfend erklärt werden können. Da eine ausführliche Beschreibung den Rahmen dieser Arbeit sprengen würde, wird an dieser Stelle lediglich ein kurzer Überblick über die für das Thema der Arbeit relevanten Aspekte gegeben.

stemantwort dar: Es kann gezeigt werden, dass die Zeit, die für Positionierungs- oder Selektionsaufgaben benötigt wird, im Sekundenbereich ansteigt, sobald die Reaktion des Systems auf eine Benutzereingabe (Mausbewegung) um Werte im Millisekundenbereich verzögert wird. Dramatisch wird dieser Effekt vor allem bei der Auswahl relativ kleiner Objekte [ware99].

Im Experiment wurde ein störender Einfluss ab Verzögerungszeiten von ca. 200ms ermittelt, was sich direkt in Verbindung setzen lässt zur Geschwindigkeit, mit der unser Gehirn visuelle Signale verarbeitet (135 - 290ms). Bei einer direkten Manipulation erwartet der Benutzer spätestens nach dieser Zeit eine Systemantwort auf seine – in der Regel kontinuierliche – Eingabe [kettner93].

Neben diesen Aspekten der Verarbeitungsgeschwindigkeit und Reaktionszeit spielen beim Entwurf von Benutzungsschnittstellen Überlegungen zur *kognitiven Belastung* des Anwenders eine besondere Rolle. Anschaulich gesprochen wird die für einen Anwender wirksame kognitive Belastung umso größer sein, je mehr er zu einem bestimmten Zeitpunkt *im Kopf behalten* muss [kettner93] – ein Aspekt der vor allem auf die begrenzte Kapazität des Arbeits- bzw. Kurzzeitgedächtnisses verweist.

Eine Erkenntnis der Kognitionsforschung besagt, dass das Kurzzeitgedächtnis nur  $7(\pm 2)$  einzelne Informationen (Ziffern, Wörter, Namen, ...) speichern kann [dix95]<sup>12</sup>. Erfordert das Durchführen einer Bedienhandlung, dass der Benutzer eine größere Zahl von Informationen behalten muss, so wird die Bedienung als schwierig empfunden; der Benutzer muss sich in jedem Teilschritt aktiv an die für die jeweilige Aktion notwendigen Informationen erinnern, eine flüssige Bedienung wird dadurch verhindert [kuhn93] [kettner93].

---

<sup>12</sup> Eine in diesem Zusammenhang interessante Beobachtung kann bei einer Analyse der Kanalkapazität für die menschliche Sinneswahrnehmung gemacht werden: Experimentell wurde gezeigt, dass der Mensch bei vielen eindimensionalen Größen bis zu sieben verschiedene Werte unterscheiden kann. Eine Erkenntnis, die sich auch in unserer Alltagserfahrung widerspiegelt: So ist die Woche in sieben Tage eingeteilt, die Tonleiter unterscheidet sieben verschiedene Tonhöhen, im kontinuierlichen Farbspektrum werden sieben Primärfarben definiert u. ä. [kettner93].

Diese Einschränkung des menschlichen Kurzzeitgedächtnisses wird durch *Lernprozesse* überwunden, indem kleinere Informationseinheiten zu größeren rekodiert werden<sup>13</sup>.

Abschließend soll auf den Begriff der *kinästhetischen Übereinstimmung* eingegangen werden. Die Kinästhesie bezeichnet den Muskelsinn des Menschen, das heisst, das Empfinden eigener körperlicher Bewegungen [brockhaus84].

Kinästhetische Übereinstimmung bei der Interaktion mit einem Computersystem ist dann gegeben, wenn die Eingabebewegung des Benutzers mit der resultierenden Objektbewegung auf dem Bildschirm korrespondiert. Dieser Zusammenhang wird oft auch als Reiz-Reaktionsverträglichkeit bezeichnet (stimulus-response-compatibility) [kettner93].

Die grundlegende Forderung nach einer guten kinästhetischen Übereinstimmung bedeutet nicht, dass die Eingabebewegung unmittelbar und ohne Änderung in die Objektbewegung umgesetzt werden muss. So kann es beispielsweise wünschenswert sein, dass der Faktor, über den eine Bewegung der Maus in eine Veränderung der Zeigerposition am Bildschirm übersetzt wird, vom Benutzer individuell eingestellt werden kann – und in der Folge zwischen Eingabe- und Ausgabe-geschwindigkeit kein absoluter, sondern nur noch ein relativer Zusammenhang besteht, die sogenannte *control-to-display-ratio* [ware99] [kettner93].

Neben diesen einfachen Erweiterungen kann es durch den Einsatz von Metaphern und der durch sie induzierten mentalen Modelle zu Abweichungen zwischen Eingabe- und Ausgabebewegung kommen. Von zentraler Bedeutung ist aber auch in diesem Zusammenhang, dass eine von der Eingabebewegung abweichende Objektbewegung von der angebotenen Metapher hinreichend erklärt wird<sup>14</sup>.

---

<sup>13</sup> Der Prozess des Lernens im Zusammenhang der Mensch-Computer-Interaktion wird in Abschnitt 3.1.3 näher beschrieben.

<sup>14</sup> Ist die Maus-Bewegung über die angebotene Metapher beispielsweise an eine Bewegung der Kamera gebunden, würden sich die Objekte der Szene bei einer *Mausbewegung nach rechts* auf die *linke* Seite bewegen; ein Verhalten, das dem durch die Metapher implizierten mentalen Modell des Benutzers entspricht und daher durchaus erwünscht wäre.

### 3.1.3 Modellbildung als Grundlage des Lernprozesses

Wie bereits im vorangegangenen Abschnitt erläutert wurde, spielt der Prozess des Lernens beim Aneignen von Techniken zur Interaktion mit einer Computeranwendung eine bedeutende Rolle. An dieser Stelle soll kurz auf die im Rahmen der Arbeit relevanten Zusammenhänge eingegangen werden.

Zunächst sei hier das rein mechanische *Auswendiglernen* genannt, das ohne tieferes Verständnis der zugrunde liegenden Konzepte abläuft. Das hierbei gewonnene Wissen ist rein willkürlich, mit der Aneignung des Wissens ist ein hoher Arbeits- und Zeitaufwand verbunden; kommt es bei der Anwendung des so gewonnenen Wissens zu Fehlern, lässt sich aus den gelernten Informationen keine Lösungsstrategie ableiten [norman89].

Im Gegensatz hierzu steht der Prozess des *Chunking*, bei dem bereits Gelerntes in größere Einheiten zusammengefasst wird. Dieser Prozess geht mit der Bildung von Konzepten auf einer höherer Ebene einher; der Lernende sucht und findet Strukturen im zunächst Ungeordneten. Dieser *Prozess der Modellbildung* beschreibt einen ganz wesentlichen Aspekt des menschlichen Lernens und Verstehens [ware99] [kettner93].

Metaphern, die dem Benutzer für die Bedienung einer Anwendung zur Verfügung gestellt werden, können diese Modellbildung unterstützen; auf ihre zentrale Bedeutung beim UI-Design wird im nun folgenden Abschnitt eingegangen.

## 3.2 Metaphern für die Benutzungsschnittstelle

In Abschnitt 2.1.2 wurde bereits auf die erklärende Kraft von verbalen Metaphern in der menschlichen Kommunikation hingewiesen. Im Folgenden wird beschrieben, wie der Einsatz von *Interface-Metaphern* dazu beitragen kann, die Bedienung von interaktiven Computersystemen zu vereinfachen.

### 3.2.1 Metaphern zur Unterstützung der Modellbildung

Der Benutzer einer Computeranwendung wird diese umso erfolgreicher bedienen können, je besser das der Anwendung zugrunde liegende Modell vom Anwender verstanden wurde. Metaphern an der Schnittstelle zwischen Benutzer und Anwendung – sogenannte Interface- oder Bedien-Metaphern – können, sofern sie geeignet gewählt und eingesetzt werden, dem Benutzer beim Verständnis dieses Modells behilflich sein [preim99].

Diese *mentalen Modelle* werden vom Anwender auch dann entwickelt, wenn die Anwendung keine entsprechenden Metaphern bereitstellt [väänänen95]. Diese Modelle sind aber unter Umständen nicht befriedigend, weil sie zum Beispiel relevante Bedienaspekte nicht abdecken oder nur unzureichend erklären [norman89]. Bedienmetaphern werden in diesem Zusammenhang oft auch als *sinnstiftende Aspekte* (englisch: *sense makers*) eines interaktiven Systems bezeichnet [kuhn95].

Wie die sprachliche Metapher überträgt auch die Interface-Metapher Zusammenhänge aus einer Quell- auf eine Zieldomäne, wobei sich die Zieldomäne aus möglichen Benutzeroperationen konstituiert, die von der Anwendung zur Verfügung gestellt werden.

Auf Grund der angebotenen Metaphern hat der Benutzer die Möglichkeit, Auswirkungen von möglichen Eingaben bereits im Vorfeld abzuschätzen: Durch die bekannten Zusammenhänge in der Quelldomäne wird ein bestimmtes Verhalten der über die Metapher gesteuerten Anwendung impliziert.

[preim99] weist darauf hin, dass der Einsatz von Metaphern Einstiegshürden bei der Bedienung von interaktiven Systemen verringern kann und den Vorteil einer steilen Lernkurve beim Aneignen der Handhabung einer Anwendung bietet. Ist das Konzept, auf dem die Bedienung einer Anwendung beruht, erst einmal verstanden, kann die Funktion einzelner Bedienelemente bei einer späteren Benutzung leichter erinnert werden; die sogenannte *re-learning time* des Systems kann deutlich reduziert werden [kuhn95] [väänänen95].

Im Zusammenhang mit einer leichteren (Wieder-)Erlernbarkeit soll an dieser Stelle auf einen weiteren bedeutenden Punkt beim Einsatz von Metaphern in interaktiven Systemen hingewiesen werden:

Anwendungen, bei denen der Benutzer zum Erreichen seiner Ziele eine Reihe von textbasierten Befehlen eingeben muss, die sich stark an der Programmlogik der Applikation orientieren und keine weitergehenden Konzepte für das Verständnis anbieten, erfordern von Benutzer im Wesentlichen ein Auswendiglernen einer der Anwendung zugrunde liegenden Syntax [marcus02a]. Ein Prozess, der – wie bereits in Abschnitt 3.1.2 beschrieben wurde – mit einer Reihe von Problemen behaftet ist.

An dieser Stelle weist der Einsatz von Bedienmetaphern den Weg von einer maschinenzentrierten Sicht hin zu einer Entwicklung, die sich an den spezifischen Anforderungen des menschlichen Anwenders orientiert.

### 3.2.2 Anforderungen an den Metaphern-Entwurf

Nachdem beschrieben wurde, wie der Einsatz von Metaphern das Erlernen und die Bedienung einer Anwendung erleichtern kann, sollen nun Kriterien untersucht werden, die solche Metaphern erfüllen müssen, sodass die beschriebenen Vorteile in einer konkreten Umsetzung auch wirkungsvoll zum Tragen kommen.

#### *Vertrautheit der Quelldomäne*

Der Einsatz einer Bedien-Metapher kann nur dann erfolgreich sein, wenn dem Benutzer die der Quelldomäne zugrunde liegenden Konzepte vertraut sind [kuhn95]. Nur auf diese Weise können diese Zusammenhänge auf einen neuen Anwendungsfall übertragen werden. Vor der Entwicklung von Bedien-Metaphern ist also insbesondere sicherzustellen, dass die verwendeten Bilder, Zusammenhänge und Konzepte im Kreis der voraussichtlichen Anwender gut bekannt sind<sup>15</sup>.

#### *Anbieten eines adäquaten Modells*

Das von der Metapher implizierte mentale Modell muss dem konkreten Anwendungsfall angemessen sein [ware99]. Dies bedeutet insbesondere, dass die durch die Metapher bereitgestellte Struktur über einen angemessenen Komplexitätsgrad verfügen sollte. Ein Interaktionswerkzeug für einen relativ einfachen Bedienvorgang, dem eine komplexe metaphorische Vorstellung zugrunde liegt, ist ebenso ungeeignet wie eine eher einfache Metapher, mit der eine komplexe Interaktion gesteuert werden soll.

---

<sup>15</sup> Unter anderem in [kuhn95] und [kettner93] wird auf die kulturelle Abhängigkeit von Bedienmetaphern in interaktiven Systemen hingewiesen. Eine Metapher, die von Benutzern eines Kulturkreises ohne Probleme verstanden wird, kann in einem anderen kulturellen Kontext anderes, konträr oder überhaupt nicht verstanden werden. So weist [kettner93] darauf hin, dass die Übertragung von räumlichen auf zeitliche Zusammenhänge zu kulturspezifischen Schwierigkeiten führen kann. So liegt die Zukunft für Anwender der westlichen Welt *vor ihnen*, während sie sich in anderen Kulturkreisen durchaus *hinter* einem befinden kann (und den Menschen gleichsam einholt).



An dieser Stelle sei auf die Erörterungen zum Metapher-Begriff in Abschnitt 2.1.2 verwiesen: Ein angemessenes metaphorisches Modell muss nicht zu 100 Prozent die Zusammenhänge der Zieldomäne abbilden, sondern vor allem das Verständnis und die intuitive Bedienung erleichtern<sup>16</sup>. Dennoch können unter bestimmten Umständen Probleme durch *negative Analogien* auftreten, die weiter unten beschrieben werden.

Die angebotenen Bedien-Metaphern müssen nicht nur dem Bedienvorgang angemessen sein, sie müssen sich auch auf die Erfahrung und besonderen Voraussetzungen des Anwenders einstellen: Bei einem erfahrenen Benutzer wird die Erwartung an ein effizientes Interaktionswerkzeug im Vordergrund stehen, während ein Neuling vor allem an einer leichten Erlernbarkeit interessiert sein wird. Aspekte, die sich gegebenenfalls widersprechen und weiter unten diskutiert werden.

### *Erweiterbarkeit*

Einzelne Metaphern erscheinen unter Umständen ideal für einen spezifischen Anwendungsfall, lassen sich aber schlecht oder gar nicht erweitern. Auf der anderen Seite wird sich der Funktionsumfang einer Anwendung in aller Regel mit der Zeit erweitern, einzelne Funktionen werden zusätzliche Optionen bieten während andere wegfallen.

Probleme ergeben sich, wenn durch neue Features das von einer Metapher implizierte mentale Modell inkonsistent wird und komplett neue Metaphern entwickelt werden müssen. Bei der Entwicklung von Interaktionsmetaphern sollte der Aspekt der Erweiterbarkeit daher gesondert berücksichtigt werden, um die natürliche Evolution einer Anwendung nicht zu beeinträchtigen [väänänen95].

### *Berücksichtigung unterschiedlicher Benutzergruppen*

Die intuitive Bedienung durch eine Interaktionsmetapher wird unter Umständen durch zusätzliche Interaktionsschritte erkauft, die bei einer Schnittstelle, welche rein nach Effizienz Gesichtspunkten gestaltet wurde, zum Lösen derselben Aufgabe nicht benötigt würden.

---

<sup>16</sup> [paelke02] weist beispielsweise darauf hin, dass das Doppelklicken auf ein Dokument zum Öffnen desselben nicht durch die zugrunde liegende Desktop-Metapher erklärt wird – ein Umstand, der die intuitive Bedienung jedoch keineswegs behindert.

Ein anschauliches Beispiel für diesen Zusammenhang erhält man, wenn man sich den Gelegenheitsanwender einer typischen Office-Anwendung vorstellt, wie er sich zum Lösen einer bestimmten Aufgabe an einen erfahrenen Benutzer wendet: In der Regel wird der Anfänger in kürzester Zeit von Maus und Tastatur getrennt, um bei der anschließenden Demonstration des effizientesten Lösungsweges vollständig den Überblick zu verlieren.

An diesem Beispiel zeigt sich der entscheidende Unterschied der beiden Benutzergruppen: Im Gegensatz zum Anfänger hat beim Experten im Laufe der Zeit ein Lernprozess eingesetzt, wie er in Abschnitt 3.1.3 beschrieben wurde – eine Reihe einzelner Bedienschritte wurde zu größeren Einheiten zusammengefasst, die Auswahl von Unterpunkten in einem verzweigten Menüsystem werden durch Tastaturkürzel (*short cuts*) ersetzt, Ausgaben des Systems werden während der Bedienung bereits antizipiert ohne dabei auf reine Statusmeldungen Rücksicht zu nehmen.

Die Durchführung von Aufgaben, die in einer solchen Weise internalisiert wurden, läuft rein automatisch ab – der erfahrene Anwender greift nicht mehr auf die vom System gegebenenfalls angebotenen metaphorischen Konzepte zurück. Im Gegensatz zum Gelegenheitsanwender, der sich an jeden einzelnen Teilschritt einer Bedienhandlung erinnern muss, bedeutet dieses Vorgehen für den Experten keine höhere kognitive Belastung [norman89].

### **3.2.3 Probleme beim Einsatz von Metaphern**

Im vorangegangenen Abschnitt wurde bereits ein mögliches Problem bei der Verwendung von Metaphern erwähnt: Fortgeschrittene Benutzer erwarten vom System die Möglichkeit, Aufgaben möglichst effizient durchzuführen; ein Problem, dem auf unterschiedliche Weise begegnet werden kann: Zum einen können für gängige Aufgaben Tastaturkürzel angeboten werden, zum anderen können die Bedienmetaphern konfigurierbar gestaltet werden, so dass der Anwender das Verhalten der Metapher seinen Ansprüchen entsprechend einstellen kann (zum Beispiel um Statusmeldungen zu reduzieren oder Teilabläufe zu automatisieren).

Neben solchen Effizienzgesichtspunkten können sich durch den Einsatz von Metaphern weitere Probleme ergeben, die im Folgenden beschrieben werden.

### *Zusammengesetzte Metaphern*

[kuhn95] weist beispielsweise darauf hin, dass eine einzelne Metapher oftmals nicht ausreicht, um den gesamten Funktionsumfang einer komplexen Applikation zu beschreiben. Als ein möglicher Lösungsansatz wird die Kombination von mehreren Metaphern genannt, wobei sich Probleme ergeben können, falls die Teilmaphern nicht miteinander kompatibel sind [väänänen95]. In verschiedenen Quellen wird jedoch darauf hingewiesen, dass ein Anwender die Kombination von unterschiedlichen metaphorischen Konzepten in einer Benutzungsschnittstelle nicht zwangsläufig als hinderlich empfinden wird; sofern die unterschiedlichen Konzepte bereits beim Entwurf berücksichtigt und sinnvoll zusammengeschaltet werden, können sie in einer schlüssigen Oberfläche aufgehen [preim99] [väänänen95] [kuhn95].

### *Negative Analogien*

Ebenfalls in [preim99] wird auf Probleme hingewiesen, die sich durch sogenannte *negative Analogien* ergeben. So kann es beispielsweise vorkommen, dass durch die Anwendung bestimmte Funktionen implementiert sind, für die im Bereich der Quelldomäne der Bedienmetapher kein Äquivalent existiert. Hier besteht also die Gefahr, dass dem Anwender diese zusätzlichen Funktionen nicht bewusst werden und daher nur ein Teil der Möglichkeiten, die in der Anwendung zur Verfügung stehen, auch tatsächlich genutzt wird.

Auch der umgekehrte Fall birgt Gefahren: Die Applikation stellt bestimmte Funktionen nicht zur Verfügung, die der Anwender auf Grund von Teilaspekten der verwendeten Metapher jedoch erwartet. In diesem Fall hat die Metapher „zu viel versprochen“ [väänänen95], der Anwender wird vergeblich nach einer Möglichkeit suchen, die gewünschte Operation durchzuführen<sup>17</sup>.

In [ware99] wird beschrieben, dass eine Bedienmetapher unter Umständen einzelne Operationen sehr gut, andere hingegen nur unvollkommen unterstützt. Dies wird dann kein Problem sein, wenn für die Bedienung nur die gut unterstützen

---

<sup>17</sup> In [preim99] wird zur Veranschaulichung der oben beschriebenen negativen Analogien das Beispiel einer Textverarbeitung und das zugrunde liegende Konzept einer Schreibmaschine herangezogen: Durch die Metapher werden wesentliche Funktionen der Anwendung nicht erklärt (Inhaltsverzeichnis erstellen, Zeichen formatieren); auf der anderen Seite stehen andere Funktionen nicht zur Verfügung (Formulare ausfüllen).

Operationen benötigt werden. Sobald eine Metapher jedoch auf andere Anwendungszusammenhänge übertragen werden soll, ist der Aspekt der Übertragbarkeit und Erweiterbarkeit gesondert zu berücksichtigen.

### **3.3 Entwurfsprinzipien der Human Computer Interaction**

#### *Analyse der durchzuführenden Aufgaben*

Am Beginn des Entwurfs einer Benutzungsschnittstelle muss eine eingehende Analyse der Aufgaben stehen, die über diese Schnittstelle durchgeführt werden sollen. Nur auf diese Weise ist es möglich, ein konsistentes Konzept zu entwickeln (siehe unten), das dem Anwender Werkzeuge an die Hand gibt, die optimal auf seine Arbeit abgestimmt sind [preim99].

Bei einer solchen Analyse werden grundlegende Aufgaben identifiziert, die ihrerseits wieder aus Teilaufgaben zusammengesetzt sein können. Teile einer Bedienhandlung, die der Benutzer mit einer einzigen Aktion ausführen kann (zum Beispiel die Eingabe eines einzelnen Kommandos oder die Wahl eines Menübefehls) heißen *atomar*.

[shneiderm98] weist darauf hin, wie wichtig die angemessene Wahl der Granularität dieser atomaren Bedienhandlungen ist: Ist diese Granularität zu groß gewählt, werden für die jeweiligen Aktionen viele unterschiedliche Werkzeuge mit einer Vielzahl von Optionen entworfen – ist sie zu klein gewählt, kann es leicht zu Frustrationserscheinungen auf Seiten des Benutzers kommen, der selbst für einfache Aufgaben eine Fülle von Einzelschritten ausführen muss.

In [preim99] wird darüber hinaus auf die sogenannte 80:20-Regel bei der Benutzerinteraktion hingewiesen: Sie besagt, dass in 80% aller Anwendungsfälle nur 20% der insgesamt zur Verfügung stehenden Interaktionsmöglichkeiten genutzt werden. Eine Regel, die deutlich macht, wie wichtig ein sorgfältiger Entwurf gerade für die Interaktionselemente ist, über die die meisten Bedienhandlungen ablaufen.

### *Entwurf eines konsistenten Gesamtsystems*

Eine weitere grundlegende Anforderung an den Entwurf von Benutzungsschnittstellen ist das Gewährleisten eines schlüssigen, konsistenten Gesamtsystems. Für dieses Konsistenzkriterium sind eine Reihe von Faktoren entscheidend (für diese und weitere Faktoren siehe [shneiderm98]):

Bei der Darstellung von Interaktionselementen ist beispielsweise darauf zu achten, dass gleiche Farben immer gleiche Systemzustände kodieren, Statusmeldungen sollten jeweils am gleichen logischen Ort ausgegeben werden und einzelne Menüpunkte dürfen ihre Position nicht ändern [foley96]. Gerade der letzte Punkt verweist auf den bereits eingeführten Begriff der Kinästhesie, den Muskelsinn des Menschen: Ein Anwender, der mit dem Aufbau des Menüsystems einer Anwendung vertraut ist, ist nicht mehr darauf angewiesen, jeden einzelnen Menüpunkt wahrzunehmen, sondern vertraut – neben seiner räumlichen Orientierung – auf ihm bekannte Bewegungsabläufe, um den Mauszeiger zum gewünschten Unterpunkt zu bewegen.

Im Zusammenhang mit der Systemaneignung durch den Anwender weist [foley96] darauf hin, dass der Mensch nur in einem konsistenten Gesamtsystem die Möglichkeit hat, vorhandenes Wissen auf unbekannte Zusammenhänge zu übertragen. Ein Kriterium für Konsistenz ist also insbesondere, dass gleiche Aufgaben in verschiedenem Kontext durch die gleiche Abfolge von Operationen zu erledigen sind – ein Zustand, der dann gegeben ist, wenn das System wenigen, einfachen Regeln gehorcht und Ausnahmestände auf ein unvermeidbares Minimum reduziert worden sind [shneiderm98].

### *Bereitstellen einer adäquaten Systemantwort*

Beim Erstellen eines Dokuments mit einem Textverarbeitungsprogramm wird vom Benutzer erwartet, dass ein Tastendruck zu einer unmittelbaren Antwort des Systems führt; genauso wie beim Bewegen der Maus eine entsprechende Bewegung des Mauszeigers auf dem Bildschirm erwartet wird. Dieses Feedback auf Hardware-Ebene ist jedoch nur der unterste Level, auf dem eine adäquate Systemantwort sicherzustellen ist.

Auf der Ebene der Applikationslogik ist zu gewährleisten, dass der Benutzer über die jeweiligen *Systemzustände* informiert wird [foley96]. Hierzu gehört zum Beispiel das Kenntlichmachen von selektierten Objekten, die Information, dass das

System aktuell beschäftigt ist und gegebenenfalls die Anzeige des Arbeitsfortschrittes.

Ebenfalls in [foley96] wird auf das Problem der *visuellen Kontinuität* hingewiesen: Werden Statusmeldungen in einem eigenen Bildschirmbereich ausgegeben, wird dieser unter Umständen nicht beachtet, wenn der Anwender einen anderen Fokus gesetzt hat. Abhilfe können hier beispielsweise akustische Signale für kritische Fehler schaffen.

### *Verringern der Fehlermöglichkeiten*

Um Bedienfehler erst gar nicht aufkommen zu lassen, sollte bereits beim Entwurf der Benutzungsschnittstelle darauf geachtet werden, dass mögliche Fehlerquellen von vornherein ausgeschlossen werden.

Ein Aspekt hierbei ist das Bereitstellen klar unterscheidbarer Bedienelemente [shneiderm98], bei denen dem Anwender unmittelbar einsichtig ist, welche Aktionen durch sie ausgelöst werden. In diesem Zusammenhang spricht man von *Affordances*, ein Begriff, der die Möglichkeiten und Konsequenzen eines Bedienelementes bezeichnet [herndon94b].

In einem bestimmten Systemzustand sollten darüber hinaus nur diejenigen Befehle aktiv sein, die eine für diesen Zustand zulässige Aktion auslösen [foley96]; ein einfaches Beispiel für solche kontextsensitiven Betrachtungen ist das Zahlenfeld, in das kein Text eingegeben werden darf oder das Unterdrücken des Löschen-Befehls, sofern zuvor kein zu löschendes Objekt ausgewählt wurde [shneiderm98].

Ist dem Anwender dennoch ein Fehler unterlaufen, sollte die Applikation Möglichkeiten zur Verfügung stellen, diesen auf einfache Weise zu beheben. So macht der *Undo-Befehl* beispielsweise die letzte Benutzeraktion rückgängig; in einer Erweiterung kann dieser Befehl auch in einer mehrstufigen Variante (*multilevel undo*) angeboten werden, um auf diese Weise ganze Befehlsfolgen rückgängig zu machen und einen konsistenten Systemzustand wiederherstellen [foley96].

[preim99] weist darauf hin, dass neben der Möglichkeit, Aktionen abubrechen, gerade das Angebot, Eingaben rückgängig machen zu können, beim Anwender dazu führt, sich das System unbefangen und explorativ anzueignen. Das Trial-and-Error-Prinzip führt hier zu einem kreativen Problemlösungsprozess.

### *Verringern der kognitiven Last*

Bereits in Abschnitt 3.1.2 wurde auf die begrenzte Kapazität des menschlichen Kurzzeitgedächtnisses hingewiesen. Diesem Problem kann unter anderem durch eine sinnvolle modale Strukturierung einer Operationenfolge erreicht werden: Einzelne Aktionen sollten dafür in klar unterscheidbaren Phasen ablaufen (Parameterspezifikation, Durchführung, Ende/Statusmeldung), sodass der Anwender Dinge, die er sich für eine Operation merken muss, unmittelbar anwenden – und wieder vergessen – kann, um sich nach erfolgreichem Abschluss dieser Aktion auf die nächste Teilaufgabe konzentrieren zu können [shneiderm98].

Um zu verhindern, dass sich der Anwender komplexe (Syntax-)Details einzelner Operationen behalten muss, sollte die Anwendung geeignete Hilfe anbieten und – idealerweise kontextbezogene – Hinweise für die Bedienung geben [broy00] [shneiderm98].





## **4. Untersuchung von Interaktionstechniken für den 3D-Raum**

In Kapitel 3 wurden Grundlagen für den Entwurf von Benutzungsschnittstellen dargestellt. Hierbei wurde ein Schwerpunkt auf die Berücksichtigung menschlicher Faktoren und insbesondere auf die Darstellung von Bedien-Metaphern gelegt. Es wurde verdeutlicht, wie sich durch den Einsatz von metaphorischen Konzepten intuitive Benutzungsschnittstellen entwerfen lassen.

Im folgenden Kapitel wird auf die speziellen Bedingungen von 3D-Anwendungen eingegangen und dargestellt, welche Rolle 3D-Metaphern beim Entwurf von räumlichen Interaktionswerkzeugen und insbesondere für die *Desktop Virtual Reality* (Desktop-VR) spielen. In Kapitel 5 werden auf Grundlage der hier vorgestellten Konzepte neue Interaktionsmetaphern für eine Anwendung der Desktop-VR entwickelt.

### **4.1 Interaktion in virtuellen 3D-Welten**

#### **4.1.1 Geschichtliche Entwicklung**

Der Mensch lebt und interagiert in einer dreidimensionalen Welt und so erscheint es nur natürlich, dass schon frühzeitig Computer-Anwendungen für eine Interaktion mit virtuellen 3D-Räumen entworfen wurden.

So hat Sutherland bereits im Jahre 1965 grundlegende Konzepte für die Immersion in virtuellen Welten vorgestellt. 1983 wurde am MIT das Virtual Workspace-System entwickelt, bei dem der Benutzer über die Position seiner Hand räumlich mit 3D-Objekten interagieren konnte.

Am NASA Ames Research Center wurde 1984 ein Projekt gestartet, in dessen Fokus die Fernsteuerung von Maschinenoperationen stand (*telepresence control*); Operationen, die es erforderlich machen, dass der Benutzer die volle räumliche Kontrolle über das System besitzt.

In den Blickpunkt einer breiteren Öffentlichkeit tritt die Idee der Virtual Reality erstmals im Jahre 1989 anlässlich einer Trade Show der Firmen VPL und Autodesk, die ihre speziell für virtuelle Räume entwickelten Interaktionsgeräte präsentieren<sup>18</sup>.

### 4.1.2 3D-Eingabegeräte

Zur Interaktion in virtuellen Umgebungen (englisch *virtual environments*, *VEs*) wurden im Laufe der Jahre buchstäblich hunderte unterschiedliche Ein- und Ausgabegeräte entwickelt [bowman01]<sup>19</sup>. Im Folgenden soll ein kurzer Überblick der gängigsten Technologien für 3D-Eingabegeräte gegeben werden (für detaillierte Darstellungen und eine Beschreibung von 3D-Ausgabegeräten siehe [kettner93] [hand93] [stork00]).

#### - *Polhemus-Sensor*

Der Polhemus-Sensor besteht aus einem Sender und einem Empfänger, die über ein elektromagnetisches Feld miteinander gekoppelt sind. Der Sender ist stationär im Raum angebracht, die räumliche Position und Orientierung des Empfängers kann auf Grund seiner Lage im vom Sender ausgestrahlten elektromagnetischen Feld bestimmt werden.

#### - *Datenhandschuh*

Mit dem Datenhandschuh werden Hand- und Fingerbewegungen des Benutzers erfasst, um die so gewonnenen Daten in ein entsprechendes 3D-Modell der Hand in die 3D-Szene zu integrieren. Eine Möglichkeit, die Stellung der Fingerglieder zu bestimmen, besteht in der Verwendung von in den Handschuh eingelassenen Glasfasern, deren Lichtdurchlässigkeit mit der Biegung der Lichtleiters variiert.

Mit einem auf dem Handschuh angebrachten Polhemus-Sensor kann zusätzlich die räumliche Lage der Hand bestimmt werden, was eine Verwendung des Datenhandschuhs als intuitives 3D-Eingabegerät ermöglicht: Die Bewegungen der eigenen Hand werden in Bewegungen des virtuellen Modells in der Szene umgesetzt, um so beispielsweise Objekte zu greifen und zu bewegen.

---

<sup>18</sup> Für diese und weitere Beispiele der historischen Entwicklung siehe [balaguer91].

<sup>19</sup> Ein Zustand, der [kettner93] dazu veranlasst, vom *Zoo der höherdimensionalen Eingabegeräte* zu sprechen...

- *Space-Mouse*

Im Gegensatz zum Datenhandschuh ist die Space-Mouse ein tischgebundenes 3D-Eingabegerät; der Benutzer kann bei der Interaktion mit einer 3D-Szene seinen Unterarm ablegen, was ein ermüdungsfreies Arbeiten auch über einen längeren Zeitraum ermöglicht [hand93].

Im Mittelpunkt der Space-Mouse steht eine bewegliche Sensorkappe, die sich in alle Raumrichtungen bewegen und um jede der drei Raumachsen verdrehen lässt. Für die Auslenkung der Sensorkappe muss eine von der Auslenkung abhängige Kraft aufgewendet werden, sodass der Benutzer ein Feedback darüber erhält, wie weit der Sensor bereits bewegt wurde.

Hervorgegangen ist die Space-Mouse aus einer Steuerkugel, mit der während der Spacelab D2-Mission erstmals ein Roboter von der Erde ferngesteuert wurde [techtrans02].

### 4.1.3 Eigenschaften der 3D-Interaktion

Eine Besonderheit der 3D-Interaktion ist ihre enge Beziehung zur Interaktion in der realen Welt: [haik02] weist darauf hin, dass dieses Merkmal zu einer intuitiven Bedienung der 3D-Anwendung beitragen kann, deutet aber bereits Probleme an, die sich für die Unterstützung der räumlichen Orientierung in einem virtuellen 3D-Raum ergeben.

So bestehen nach [bowman01] wesentliche Unterschiede zwischen der realen und einer vom Computer erzeugten virtuellen Welt: Die reale Welt bietet uns eine Vielzahl von Anhaltspunkten (englisch *depth cues*), die uns bei einer räumlichen Auswertung der wahrgenommenen Szene unterstützen (für eine ausführliche Beschreibung siehe auch [krömker02]).

Durch den Einsatz geeigneter Darstellungstechniken versucht man, dem Betrachter einer virtuellen 3D-Szene Anhaltspunkte für die Lage der Objekte im Raum zu geben. Beispielsweise kann die Texturierung von Gegenständen in der Szene dabei helfen, ein Gefühl für Entfernungen und Plastizität zu vermitteln, die Manipulation von Farbwerten in Abhängigkeit der Tiefe eines Objekts in der Szene kann, vergleichbar mit Nebeneffekten der realen Welt, ebenfalls zu einer verbesserten räumlichen Wahrnehmung beitragen [kettner93].

Wirksame *Constraints* sind ein Unterschied zwischen realer und virtueller Welt, auf den in [bowman01] hingewiesen wird. Unter einem *Constraint* versteht man beispielsweise die Beschränkung einer Bewegung auf bestimmte Koordinatenachsen<sup>20</sup>. So bewegt sich der Mensch im täglichen Leben zumeist nur entlang von zwei Richtungen im Raum; eine Beschränkung, welche den Traum vom Fliegen plausibel macht, von den meisten aber zunächst als durchaus beruhigend empfunden wird.

In [hanson97] wird beschrieben, dass gerade die freie Kontrolle aller Freiheitsgrade bei der Navigation zu dem oft zitierten Gefühl des *lost-in-space* beiträgt, bei dem der Anwender die Orientierung im Raum verliert<sup>21</sup>. [smith01] weist darauf hin, dass die Interaktion mit Objekten der realen Welt oft nicht über die Steuerung von drei Freiheitsgraden hinausgeht und gibt als Beispiel Möbelstücke, die üblicherweise entlang zweier Raumachsen verschoben und um eine Raumachse gedreht werden können.

Durch ihre enge Beziehung sind interaktive Aufgaben in virtuellen Räumen vergleichbar mit jenen der realen Welt, können jedoch auf vielfältige Weise umgesetzt werden. Als ein wichtiges Kriterium beim Entwurf von Interaktionswerkzeugen für 3D-Anwendungen wird daher häufig eine möglichst natürliche Übertragung von Bedingungen der realen auf die der virtuellen Welt genannt [haik02] [hannema01].

## 4.2 Besondere Bedingungen der Desktop-VR

### 4.2.1 Freiheitsgrade der 3D-Interaktion

Sowohl der Polhemus-Sensor als auch die Space-Mouse stellen dem Anwender die Kontrolle von *sechs Freiheitsgraden* zur Verfügung. Der Begriff Freiheitsgrad (oder englisch *degree of freedom*, häufig abgekürzt als *DOF*) bezeichnet nach [encarta01]:

---

<sup>20</sup> Für eine ausführliche Beschreibung von semantischen Constraints in der 3D-Interaktion siehe [schulze02].

<sup>21</sup> Solche Mechanismen werden beispielsweise von Navigationswerkzeugen angeboten, die sich an der Steuerung eines Helikopters orientieren. Eine Beschreibung der grundlegenden Interaktionsaufgaben des 3D-Raums – unter anderem der Navigation – findet sich in Abschnitt 4.3.

die frei wählbaren, voneinander unabhängigen Parameter eines physikalischen Systems, die je nach Zusammenhang die räumliche Lage, die Bewegungsmöglichkeiten oder den Zustand des Systems eindeutig festlegen.

Ein starrer Körper besitzt *drei Freiheitsgrade der Translation*: Er kann unabhängig in jede der drei Raumrichtung verschoben werden. Zusätzlich lässt er sich um jede einzelne der drei Raum-Achsen rotieren: der Körper verfügt über *drei Freiheitsgrade der Rotation*.

Die *Position* eines Objektes im Raum wird demnach über seine Translationsparameter definiert, während die *Orientierung* durch seine Rotationsparameter bestimmt wird. Abbildung 4.1 visualisiert die drei Freiheitsgrade der Rotation<sup>22</sup>.

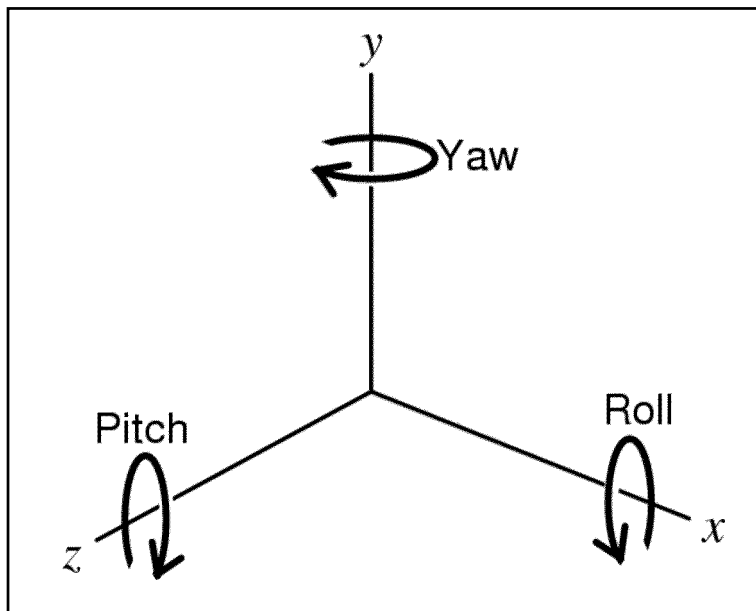


Abbildung 4.1: Freiheitsgrade der Rotation [hand93]

---

<sup>22</sup> Für die in der Abbildung aufgeführten Begriffe *pitch*, *roll* und *yaw* existieren im Deutschen keine adäquate Entsprechungen, beziehungsweise existieren *nicht mehr*: So wird als mögliche Übersetzung des Verbs *to yaw* das deutsche *gieren* vorgeschlagen. Eine *Gier* bezeichnet in der Schifffahrt das Abweichen eines Schiffes vom gesteuerten Kurs (also eine auf den Schiffskörper wirkende Rotationskomponente um die y-Achse); diese Bedeutungsvariante ist in der heutigen Alltagssprache jedoch nicht mehr geläufig.

## 4.2.2 Desktop-VR

Die 3D-Eingabegeräte der virtuellen Realität erlauben dem Anwender die unmittelbare Kontrolle der einzelnen Freiheitsgrade des 3D-Raums – bedeuten aber gleichzeitig hohe Anschaffungskosten und sind in der Regel relativ administrativ aufwändig [merten92].

Neben dem weiten Feld der immersiven VEs bestehen Systeme der sogenannten *Desktop-Virtual Reality (Desktop-VR)*; auch *Fishtank-VR* bzw. *Non-immersive VEs* genannt [slater02]. Bei diesen Systemen erfolgt die Interaktion mit der 3D-Szene über eine herkömmliche Computer-(2D-)Maus und die Eingabe-Tastatur. Die 3D-Szene wird auf einem traditionellen 2D-Display dargestellt<sup>23</sup>. Auf Grund der zunehmenden Leistungsfähigkeit moderner Grafikkarten auf dem Gebiet des 3D-Renderings bieten aktuelle Desktop-Computer selbst bei der Animation aufwändiger 3D-Szenen Frame-Raten, für die vor einigen Jahren noch teure Grafik-Workstations nötig waren<sup>24</sup>.

Die Umsetzung der im Rahmen dieser Arbeit entwickelten Bedienkonzepte erfolgt auf einem solchen Desktop-VR-System, aus diesem Grund sollen im Folgenden die besonderen Bedingungen dieser Architektur untersucht werden.

## 4.2.3 Desktop-VR und Freiheitsgrade

Eine Computer-Maus erlaubt dem Anwender – im Gegensatz zu den oben beschriebenen spezialisierten 3D-Eingabe-Geräten – die Kontrolle von bis zu zwei Freiheitsgraden; die Maus lässt sich in zwei unabhängige Richtungen bewegen: vor/zurück (ein DOF) und von einer Seite zur anderen (ein weiterer DOF).

---

<sup>23</sup> Eine Konfiguration, die [barrileaux01] mit dem Akronym *POCS* (für *Plain Old Computer System*) bezeichnet; in Anlehnung an *POTS (Plain Old Telephone System)*, das sich seit der Einführung von modernen digitalen Telekommunikationsdiensten auf das herkömmliche analoge Telefon bezieht.

<sup>24</sup> In [iX02] wird eine Entwicklung der TU Braunschweig beschrieben, die sogar bei einer klassischen VE-Anwendung, der *CAVE (Cave Automatic Virtual Environment)*, auf den Einsatz von herkömmlichen PCs setzt: Das Projekt *DAVE (Definitely Affordable Virtual Environment)* benutzt einen PC-Cluster auf Linux-Basis zur Erzeugung der mehrseitigen Raumansichten. Als besonderes Merkmal wird – neben dem Kostenfaktor – auf die im Gegensatz zu High-End-Workstations deutlich kürzeren Innovationszyklen bei Standard-Prozessoren und -Grafikkarten hingewiesen.

Bei der Steuerung von Position beziehungsweise Orientierung eines 3D-Objektes mit Hilfe der Maus kann der Anwender daher nur zwei Variablen gleichzeitig steuern: zwei Translations- oder Rotationsparameter<sup>25</sup>.

Der Vergleich von realer und virtueller Welt legt nahe (siehe oben), dass eine solche Beschränkung nicht unbedingt von Nachteil sein muss. [chen88] beschreibt in seiner Arbeit zur Entwicklung der Virtual-Sphere-Metapher beispielsweise, wie sich der Mensch bei der Durchführung unterschiedlicher 3D-Operationen verhält<sup>26</sup>: Versuchsteilnehmer wurden gebeten, verschiedene Aufgaben mit 3D-Eingabegeräten durchzuführen und dabei die Anzahl der von ihnen gleichzeitig gesteuerten Freiheitsgrade erfasst. Ein aufschlussreiches Ergebnis dieser Arbeit ist in der folgenden Tabelle dargestellt:

Operation	kontrollierte DOF
Rotation	1 DOF
Translation	2 DOF
Skalierung	3 DOF

Tabelle 4.1: kontrollierte Freiheitsgrade bei 3D-Operationen  
[chen88]

Die Versuchsergebnisse zeigen, dass die meisten Versuchspersonen bei der Durchführung einer Rotationsoperation lediglich eine einzige Rotationsachse kontrollieren. Einige wenige Versuchsteilnehmer haben bei Rotationsoperationen zwei Rotationsachsen simultan gesteuert; bei diesen beiden Achsen hat es sich jedoch immer um die x- und y-Achse gehandelt – eine Kombination, die den Teilnehmern vertrauter zu sein schien als die xz- oder yz-Achsenkombination.

[chen88] folgert aus diesen Ergebnissen, dass es den Teilnehmern offenbar schwerfällt, ein verlässliches mentales Modell zu entwickeln, das es ihnen ermöglichen würde, die Auswirkungen einer Operation mit gleichzeitiger Veränderung aller zur Verfügung stehenden Freiheitsgrade abzuschätzen. Eine Ausnahme stellen lediglich Skalierungsoperationen dar, die von den Teilnehmern offenbar als weniger komplex empfunden werden.

---

<sup>25</sup> Die Steuerung eines 3D-Objektes kann sich dabei sowohl auf beliebige Gegenstände in der Szene als auch auf die Kontrolle der *Kamera* zur Steuerung des Blickwinkels beziehen.

<sup>26</sup> Die Virtual-Sphere-Metapher wird ausführlich in Abschnitt 4.3.2 beschrieben.

Neben dem Aspekt einer hohen kognitiven Last sieht [hannema01] im motorischen System des Menschen ein weiteres Hindernis für die gleichzeitige Kontrolle mehrerer Freiheitsgrade: Das exakte Platzieren oder Ausrichten eines Objekts ohne wirksame Constraints wird als schwierig empfunden, weil die menschliche Motorik darauf trainiert ist, *natürliche Constraints* zu suchen und auszunutzen. Soll ein Stift beispielsweise auf einer Tischoberfläche abgelegt werden, wird man ihn nicht zunächst sorgfältig im freien Raum parallel zur Oberfläche ausrichten, um ihn dann behutsam abzusenken; vielmehr benutzt man die Ablagefläche ganz intuitiv als ein die Bewegung beschränkendes Element.

### 4.2.4 Funktion der 3D-Metapher in der Desktop-VR

„*WYDIWYS – What You Do Is What You See*“ [chen88]

Bei der 3D-Interaktion über Computer-Maus und Tastatur und der daraus resultierenden Beschränkung auf zwei Freiheitsgrade spielt der Begriff der 3D-Metapher eine bedeutende Rolle. Bereits in Abschnitt 3.2 wurde die Funktion von Bedienmetaphern bei der Gestaltung von allgemeinen Benutzungsschnittstellen hervorgehoben; im Folgenden wird die Bedeutung von 3D-Metaphern für die Interaktion in 3D-Räumen erläutert.

[kettner93] definiert den Begriff der 3D-Metapher wie folgt:

*Eine 3D-Metapher erklärt die Bedienung einer dreidimensionalen Operation durch ein zweidimensionales Eingabegerät. Sie besteht aus zwei Teilen: dem metaphorischen Konzept, das die Strukturabbildung vom mentalen Modell des Anwenders in die Mensch-Maschine-Schnittstelle beschreibt, und der Implementierung, die die Abbildung der zweidimensionalen Eingabe in das dreidimensionale, mathematische Modell der Applikation bezeichnet. Um den Aufbau eines mentalen Modells zu erleichtern, bezieht sich das metaphorische Konzept auf den dreidimensionalen Erfahrungsbereich des Anwenders.*

In dieser Definition finden sich die Eigenschaften wieder, welche bereits im Abschnitt über Metaphern in allgemeinen Benutzungsschnittstellen beschrieben wurden; hier soll nun auf die spezifischen Bedingungen der 3D-Metaphern eingegangen werden:



- *erklärender Charakter*  
Die 3D-Metapher macht die Funktion einer dreidimensionalen Operation anschaulich. Die Bedienmöglichkeiten, die ein Interaktionswerkzeug bereitstellt, die sogenannten Affordances (siehe Abschnitt 3.3) werden durch das von der 3D-Metapher angebotene mentale Modell deutlich gemacht.
- *Abbildung Quell- nach Zieldomäne*  
Konzepte aus einer Quelldomäne werden von der 3D-Metapher auf eine Zieldomäne übertragen. Die Zieldomäne besteht dabei aus Operationen im 3D-Raum, die vom Benutzer gesteuert werden sollen. Diese Abbildung erfolgt nicht direkt, sondern über den Umweg der zweidimensionalen Benutzungsschnittstelle. Die Funktion der 3D-Metapher in diesem spezifischen Kontext wird anschließend genauer erläutert.
- *Vertrautheit der Quelldomäne*  
Um die Funktion der 3D-Metapher zu gewährleisten, müssen dem Anwender die Konzepte der Quelldomäne vertraut sein beziehungsweise vor der ersten Anwendung plausibel gemacht werden. Die Quelldomäne der 3D-Metapher bezieht ihre Konzepte dabei aus dem Gebiet der räumlichen Interaktion in der realen Welt.

Gemäß der obigen Definition vermittelt die 3D-Metapher zwischen der Steuerung über die zweidimensionale Benutzungsschnittstelle (2D-Maus) und der Durchführung einer dreidimensionalen Operation – ein Zusammenhang, der über die folgende Abbildung anschaulich gemacht wird.

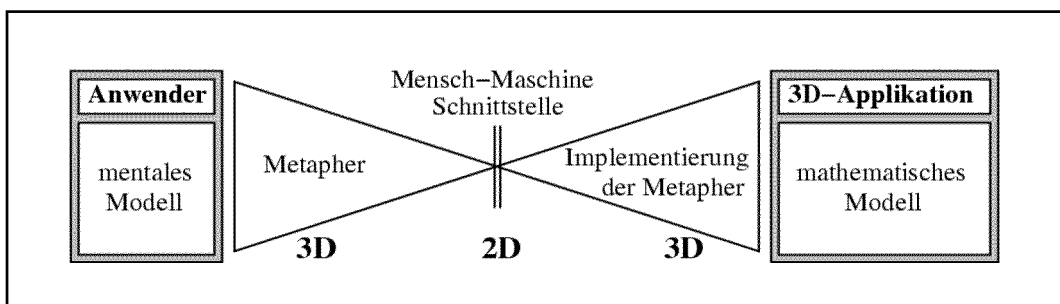


Abbildung 4.2: Die Funktion der 3D-Metapher [kettner93]

In der Mitte der Abbildung ist die zweidimensionale Mensch-Maschine-Schnittstelle dargestellt; hier erfolgt die eigentliche Interaktion mit der Anwendung. Zur Steuerung räumlicher Operationen werden dem Anwender von der Schnittstelle 3D-Metaphern bereitgestellt; das von einer solchen Metapher induzierte mentale

Modell erklärt, wie die zweidimensionalen Eingaben zur Steuerung der räumlichen Interaktionsaufgabe umgesetzt werden. Die 3D-Applikation setzt dieses Modell mathematisch um; das heißt, hier werden die Benutzereingaben ausgewertet und entsprechende 3D-Operationen berechnet (beispielsweise die Rotationskomponenten eines Objekts verändert oder die Position der Kamera aktualisiert).

Entspricht die 3D-Metapher den in Abschnitt 3.3 erläuterten Entwurfskriterien, gibt sie dem Anwender ein Werkzeug an die Hand, mit dem ihm eine intuitive Bedienung von 3D-Anwendungen ermöglicht wird [preim99]. [kettner93] weist zusätzlich darauf hin, dass der Begriff der 3D-Metapher von anderen Autoren nicht auf die Desktop-VR beschränkt wird; die Übertragung des mentalen Modells erfolgt dann nicht über eine 2D-Schnittstelle, sondern direkt über die 3D-Eingabegeräte.

Im nun folgenden Abschnitt 4.3 werden Basisoperationen des dreidimensionalen Raums beschrieben und ausgewählte Beispiele für 3D-Metaphern dargestellt.

### 4.3 Grundlegende Interaktionsaufgaben im 3D-Raum

Navigation, Objekt-Selektion und -Manipulation sind die Basisoperationen, die von dreidimensionalen Anwendungen umgesetzt werden müssen. Im Folgenden werden die besonderen Aufgaben und Schwierigkeiten hierbei beschrieben und Beispiele für die Umsetzung gegeben.

#### 4.3.1 Navigation

In der 3D-Interaktion verstehen wir unter Navigation die Bewegung des Benutzers in der Umgebung der 3D-Szene – in der Regel gesteuert durch den Prozess einer kontinuierlichen Spezifikation von unterschiedlichen Kamera-Parametern [hanson97]<sup>27</sup>.

Häufig ist die Navigation durch eine Szene die Grundlage zur Durchführung der eigentlichen Interaktionsaufgaben und nicht die Aufgabe an sich. [balaguer91] und [bowman01] weisen darauf hin, dass die Bedienung der Navigationswerkzeu-

---

<sup>27</sup> An dieser Stelle sei kurz darauf hingewiesen, dass der Begriff *Navigation* schon an sich eine (verbale) Metapher darstellt – und wie bereits weiter oben liegt auch hier der Ursprung auf dem Gebiet der Schifffahrt. In der Informationstechnik wird der Vorgang einer Bewegung durch ein bestimmtes Medium auf zunehmend abstrakte Zusammenhänge übertragen: Die Navigation durch virtuelle 3D-Räume oder auch die Hypertext-Navigation zwischen verschiedenen Internetseiten.

ge in einem solchen Fall so leichtgewichtig wie möglich zu gestalten ist, um den Benutzer nicht von der Durchführung der eigentlichen Aufgaben abzulenken.

Damit die Bewegung durch eine 3D-Szene möglichst effizient und natürlich ablaufen kann, sind neben den spezifischen Gegebenheiten des Raums, in dem die Bewegung stattfindet, auch die durchzuführenden Interaktionsaufgaben zu berücksichtigen: An Navigationswerkzeuge für weitläufige Szenen mit relativ wenigen Objekten sind daher andere Anforderungen zu stellen (schnelle Bewegung zu einem gewählten Objekt) als an Tools, die eine Bewegung in engen Räumen mit einer Vielzahl von kleinen Gegenständen unterstützen sollen (exakte Positionierung und Ausrichtung der Kamera).

In diesem Zusammenhang können folgende Modi der Navigation unterschieden werden [bowman01] [tan01]:

- *Exploration*  
Der Benutzer erkundet die Szene ohne festgesetztes Ziel. Die Navigation dient nicht einer übergeordneten Interaktionsaufgabe, sondern ist vielmehr Mittel zum Zweck.
- *Search*  
Hierbei handelt es sich um eine gezielte Suche des Benutzers nach bestimmten Objekten oder Viewpoints der Szene, um dort weitere Operationen durchzuführen.
- *Maneuvering, Inspection*  
Beim *Maneuvering* geht es dem Benutzer um eine *präzise Steuerung* seiner Position und Blickrichtung im Raum, um zur Durchführung einer bestimmten Aufgabe eine möglichst vorteilhafte Position einzunehmen. Bei der *Inspektion* bezieht sich die Ausrichtung im Raum auf ein zu untersuchendes Objekt.

Ist die Bewegung im virtuellen Raum nicht durch wirksame Constraints beschränkt, sind vom Anwender eine Vielzahl von Freiheitsgraden zu steuern: Neben der räumlichen Position (3 DOF) und Orientierung (3 DOF) können bei einer Bewegung durch den Raum weitere Freiheitsgrade hinzukommen (Geschwindigkeit oder Beschleunigung der Bewegung). Zwei weit verbreitete Navigationsmetaphern beschränken gezielt die Anzahl der Freiheitsgrade und machen eine Steuerung auch mit 2D-Eingabegeräten möglich:

### - *Walking*

Der Anwender bewegt sich auf einem zweidimensionalen Untergrund entlang der durch seine Blickrichtung definierten Achse und steuert dabei zwei unterschiedliche Freiheitsgrade. Zum einen kann die Bewegung entlang dieser Achse gesteuert werden (vor und zurück), zum anderen lässt sich die Blickrichtung durch eine Rotation der Kamera anpassen (Drehung nach links oder rechts).

Häufig erfolgt die Veränderung des Blickwinkels dabei nicht im lokalen Koordinatensystem der Kamera, sondern richtet sich nach der y-Achse des Weltkoordinatensystems.

Die Walking-Metapher erfordert eine *kontinuierliche Eingabe* des Benutzers; wird diese unterbrochen, bricht auch die Bewegung durch die Szene ab. Ein Umstand, der den Einsatz dieser Metapher gegebenenfalls problematisch erscheinen lässt: Steht die Navigation nicht im Mittelpunkt der durchzuführenden Interaktionsaufgabe, kann die Notwendigkeit einer solchen kontinuierlichen Steuerung vom Anwender als unerwünschter Overhead empfunden werden [igarashi98]. Vor allem in dünn besetzten und weitläufigen Szenen wird der Einsatz einer Walking-Metapher zur Bewegung in der Szene nicht angemessen sein [slater02] [elcacho01].

Als einen weiteren Nachteil führt [elcacho01] die Schwierigkeit in Feld, sich in der Gesamtszene zu orientieren: Die vertikale Ausrichtung der Kamera ist fix und erlaubt es dem Anwender nicht, sich durch eine erhöhte Perspektive einen Überblick zu verschaffen.

Gut geeignet erscheint die Navigationssteuerung mittels Walking dagegen in eher engen Räumen, wenn es im Wesentlichen darauf ankommt, die Umgebung zu erkunden oder bestimmte Standorte exakt anzusteuern: Der ideale Standpunkt für die jeweilige Interaktionsaufgabe kann sukzessive angenähert und Bewegungen in eine falsche Richtung leicht rückgängig gemacht werden [slater02]<sup>28</sup>.

---

<sup>28</sup> Die Walking-Metapher wird beispielsweise erfolgreich in 3D-Computerspielen eingesetzt, um einen vom Spieler gesteuerten Charakter durch die 3D-Szene zu bewegen [igarashi98]. Die Reduktion auf zwei steuerbare Freiheitsgrade wird hier in der Regel nicht als Beschränkung wahrgenommen, sondern verringert vielmehr die Gefahr eines Orientierungsverlustes, indem zum Beispiel unerwünschte *roll*- und *yaw*-Rotationskomponenten von vornherein unterbunden werden.

- *Flying*

Die Flying-Metapher kann – vor allem in Anbetracht der Vor- und Nachteile bei der Bewegungssteuerung – gewissermaßen als Gegenstück der Walking-Metapher angesehen werden.

Bei der Anwendung dieser Metapher bewegt sich der Benutzer kontinuierlich entlang einer geraden Flugbahn durch die ihn umgebende Szene. Die aktuelle Richtung dieser Flugbahn wird üblicherweise durch die Spezifizierung eines Ziels gesteuert. Bei der Verwendung einer 2D-Maus geschieht dies durch die Auswertung eines Mausklicks in die Szene, der als eine Richtungsangabe entlang der x- und y-Achse interpretiert wird.

Zusätzlich zur Flugrichtung kann der aktuelle Geschwindigkeitsvektor der Bewegung variiert werden. In einer Desktop-VR-Umgebung kann dies beispielsweise über die Pfeiltasten des Keyboards erfolgen (nach oben: schneller, nach unten: langsamer, ggf. auch mit der Möglichkeit einer negativen Geschwindigkeit → Bewegung *aus der Szene heraus*).

Im Gegensatz zum Walking erlaubt die Flying-Metapher also die Steuerung von insgesamt drei Freiheitsgraden (2 DOF für das Ziel plus 1 DOF für den Geschwindigkeitsvektor) und ermöglicht dem Benutzer so eine freie Navigation durch die Szene – die Bewegung ist nicht mehr an eine zugrunde liegende Ebene gebunden.

Ein Nachteil der Flying-Metapher wird bei einer Navigation durch dichtbesetzte oder abgeschlossene Räume deutlich, in denen die freie Wahl der Flugbahn häufig durch andere Objekte oder Wände beschränkt wird. Bei der Interaktion muss sich der Anwender dann vor allem darauf konzentrieren, die Geschwindigkeit der Bewegung und das aktuelle Flugziel möglichst exakt zu kontrollieren, um auf diese Weise mögliche Singularitäten zu vermeiden (wie

---

Steht bei einem Computerspiel zudem der Explorationsaspekt einer fremden 3D-Welt im Vordergrund – der Spieler begibt sich auf die Suche nach Werkzeugen, Waffen und Zaubersprüchen, mit deren Hilfe sich seine Gegner ausschalten lassen – erklärt sich in diesem Punkt zusätzlich die hohe Akzeptanz der Walking-Metapher in einer solchen Umgebung: Ein großer Teil des Spielerlebnisses besteht ja gerade darin, sich *keinen* Überblick verschaffen zu können, sondern sich von immer neuen Gefahren und Verlockungen überraschen zu lassen.

beispielsweise den Flug *durch* ein Objekt oder den Abbruch der Bewegung vor einer Wand) [ousland98] [vBallegooij01].

Zusätzlich weist [igarashi98] auf die fehlende Expressivität der Flying-Metapher hin: Die Bewegung erfolgt immer entlang einer Geraden ohne sich am spezifischen Kontext der Szene zu orientieren, außerdem lässt sich die Orientierung der Kamera nicht unabhängig von der Flugbahn verändern<sup>29</sup>.

Neben diesen beiden grundlegenden Navigationsmetaphern werden in der Literatur eine Vielzahl weiterer Konzepte beschrieben. Im Folgenden werden ausgewählte Verfahren dargestellt, um einzelne Aspekte exemplarisch zu beleuchten.

Die Bewegung durch die Szene mit Hilfe einer Flying- oder Walking-Metapher ermöglicht dem Benutzer eine freie Erkundung der Umgebung, ohne dass dabei einzelne Standpunkte oder Blickwinkel besonders ausgezeichnet sind. Geht es in einer Anwendung jedoch darum, die spezifischen Eigenschaften einer Szene ohne umfangreiche Erkundungen zu veranschaulichen, bieten sich Erweiterungen dieser Navigationsmetaphern an.

### *Viewpoint Navigation und Navigation by Query*

Ein *Viewpoint* ist eine von der Anwendung definierte Position in der Szene verbunden mit Informationen zu Blickrichtung und Weite des Sichtfeldes. Indem die Applikation solche Viewpoints an ausgezeichneten Punkten in der Szene definiert, erlaubt sie dem Benutzer, sich durch gezielte Navigation zu diesen Viewpoints schnell einen Überblick von der ihm dargebotenen Szene zu verschaffen [elcacho01].

In diesem Sinne ist ein Viewpoint also nichts anderes als die Übertragung von ausgezeichneten touristischen Aussichtspunkten der realen Welt, die es dem Reisenden ermöglichen, sich beispielsweise von einer landschaftlichen Attraktion ein Bild aus einem besonders geeigneten Blickwinkel zu machen.

[vBallegooij01] erweitert dieses Verfahren der szenenspezifischen Viewpoints und berücksichtigt bei dem von ihm vorgestellten Konzept der *Navigation by*

---

<sup>29</sup> Eine Lösung für dieses Problem stellt [igarashi98] mit seiner Path-Drawing-Metapher vor, die in Abschnitt 5.4.2 bei der Herleitung der Zweipunkt-Metapher für die Manipulation am Rubik's Cube besprochen wird.

*Query* weitere Elemente aus der virtuellen Umgebung, wie zum Beispiel sogenannte *areas of interest* (Teilräume mit besonderen Eigenschaften), einzelne Objekte der Szene oder – in Multi-User-Umgebungen – weitere in der Szene anwesende Personen.

Ein Problem haben Viewpoint-Verfahren und Navigation by Query gemein: Auf welche Weise wird der Anwender vom aktuellen Standpunkt zum angewählten Ziel geführt? Ein direkter Sprung zum Ziel und der damit verbundene abrupte Wechsel der Szenendarstellung kann leicht zur Desorientierung des Benutzers führen. Je nach Art der Szene wird man für solche Überführungsaufgaben in der Regel zwischen angepassten Flying- und Walking-Metaphern wählen [vBallegooij01].

#### *Head-butt Zoom, Frame of Reference*

Beim von [mine97] eingeführten Verfahren des *Head-butt Zoom* handelt es sich um eine Technik zur schnellen Annäherung an einen ausgewählten Bereich der Szene (beziehungsweise zum Ändern des aktuellen Zoom-Faktors der Kamera)<sup>30</sup>.

Hierbei markiert der Anwender einen rechteckigen Bereich der Szene, an den er sich annähern möchte. Über Bezugsrahmen (engl. *frame of reference*) wird sowohl der Annäherungskoeffizient als auch ein neuer Blickwinkel definiert und die Kameraposition entsprechend verändert. Die Abbildung 4.3 veranschaulicht das diesem Verfahren zugrunde liegende Prinzip.

---

<sup>30</sup> Dieses Verfahren wurde für immersive virtuelle Umgebungen entwickelt, wobei der modale Wechsel zwischen Zoom- und Original-Ansicht durch eine Kopfbewegung nach vorn oder hinten bewirkt werden konnte. In diesem Zusammenhang erklärt sich auch der etwas ungewöhnliche Name des Verfahrens.

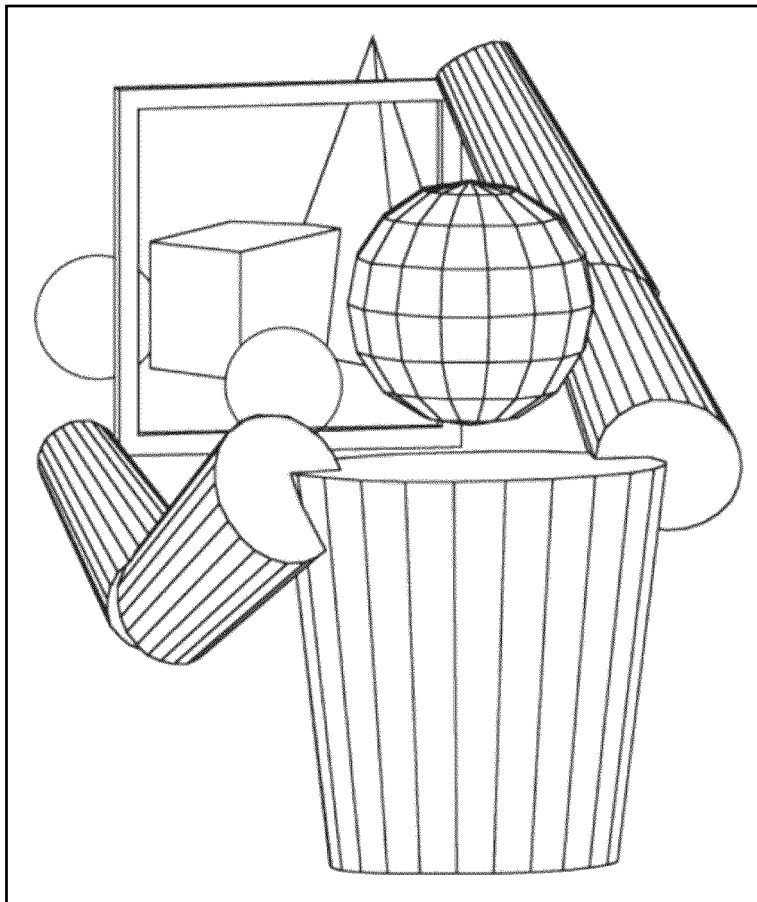


Abbildung 4.3: Head-butt Zoom-Verfahren [mine97]

### 4.3.2 Manipulation

Bei der Interaktion mit dem virtuellen Modell des Rubik's Cube stehen Rotationsoperationen im Mittelpunkt; Skalierungs- und Translationsaufgaben müssen hierbei nicht abgebildet werden. Im Folgenden soll daher ein Schwerpunkt auf Rotationsmetaphern gelegt werden und hier insbesondere auf das von Chen im Jahre 1988 entwickelte Virtual Sphere-Konzept [chen88]. Diese Metapher wird im Rahmen der prototypischen Umsetzung zudem als Referenz-Konzept für Inspektion und Manipulation implementiert.

Im Rahmen seiner Arbeit stellt Chen grundlegende Verfahren zur Parameterspezifikation für Rotationsaufgaben dar, die eine 3D-Interaktion mit herkömmlichen 2D-Eingabegeräten unterstützen:



- *Sliders*  
Hierbei steuert der Benutzer die Objektrotation durch eine Wertgabe mit Hilfe von drei graphischen Schieberegler, wobei ein Regler die Rotationskomponente um jeweils eine Raumachse definiert.
  
- *Menu Selection*  
Nach der Spezifikation einer Rotationsachse über ein Menü steuert der Anwender den Rotationswinkel um eine Raumachse durch Auslenkung der Maus in eine Richtung.
  
- *Button Press*  
Bei diesem Verfahren wird die Rotationsachse über eine von drei Maus- oder Keyboard-Tasten gewählt. Wie bei der *Menu Selection* wird eine anschließende Auslenkung der Maus als Wertgabe für die durchzuführende Rotation interpretiert.

Den beschriebenen Verfahren ist ein Nachteil gemeinsam: Die gleiche Mausebewegung steuert – je nach spezifizierter Rotationsachse – eine andere Rotationsbewegung des Objektes. Die Forderung nach einer guten Reiz-Reaktionsverträglichkeit (vgl. Abschnitt 3.1.2 ) wird auf diese Weise nicht – oder nur unzureichend – erfüllt: Nur wenn die aktuelle Ausrichtung des Objekts (und damit des lokalen Koordinatensystems) mit der gewählten Rotationsachse korrespondiert, wird die Mausebewegung in eine konforme Rotationsoperation umgesetzt.

Während den oben beschriebenen Rotationsmetaphern noch relativ simple Annahmen zugrunde liegen (direkte Umsetzung einer eindimensionalen Steuergröße in eine Rotationskomponente um eine Achse), impliziert der Virtual Sphere-Controller ein komplexeres metaphorisches Konzept: Wie der Name bereits andeutet, beruht die Vorstellung hierbei auf einer *virtuellen Glaskugel*, die das zu bewegendes Objekt umgibt und deren Mittelpunkt mit dem Mittelpunkt des Objekts übereinstimmt.

Bewegt der Anwender die Maus bei gedrückter linker Maustaste über dem Objekt (*mouse dragging*), so wird diese Bewegung als Drehung der virtuellen Sphäre (und des in ihr enthaltenen Objekts) interpretiert (siehe untenstehende Abbildung). Die Assoziation einer freigelagerten Kugel, die unter den Bewegungen der Maus *wegrollt*, macht die im Deutschen übliche Bezeichnung *Rollkugel-Metapher* plausibel [kettner93].

Die Virtual Sphere-Metapher ermöglicht es dem Benutzer, ein selektiertes Objekt frei um die x- und die y-Achse zu rotieren. Wird die Maus dabei diagonal sowohl in x- als auch in y-Richtung der Eingabeebene bewegt, resultiert diese Bewegung in einer zusammengesetzten Drehung um beide Rotationsachsen – wobei die vertikale Translationskomponente der Maus den Betrag der Rotation um die x-Achse und die horizontale Translationskomponente die Rotation um die y-Achse spezifiziert.

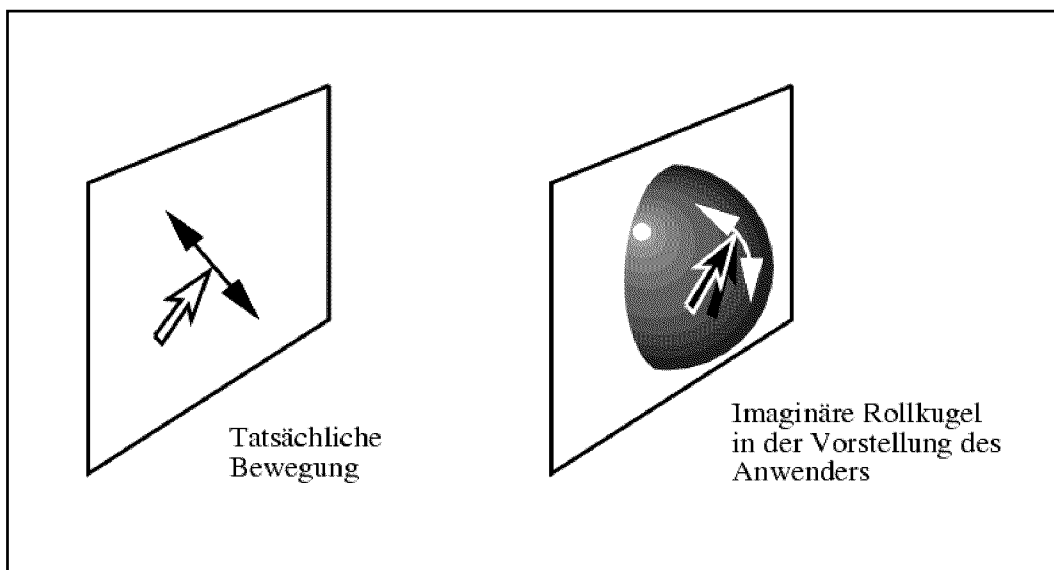


Abbildung 4.4: Die Virtual Sphere-Metapher [kettner93]

Die Virtual Sphere-Metapher hat in dieser oder einer ähnlichen Form im Laufe der letzten Jahre eine weite Verbreitung gefunden, unter anderem beruht das in der Java3D-API [sun02] enthaltene MouseRotate-Behavior auf dem hier beschriebenen Konzept. Zwischen der Virtual Sphere-Konzeption von Chen und der Sun-Implementierung bestehen jedoch einige Unterschiede, auf die hier kurz eingegangen werden soll:

- *Visualisierung und Skalierungsfaktor*

In der Implementierung von Chen wird die virtuelle Sphäre optisch durch einen das Objekt umgebenden Kreis angedeutet und die der Rotation zugrunde liegende mathematische Auswertung so ausgerichtet, dass ein komplettes Abfahren des Kreisdurchmessers zu einer 180°-Drehung des zu rotierenden Objektes führt. In der Sun-Implementierung wird auf diese Visualisierung mit der ihr verbundenen Semantik verzichtet; hier kann explizit ein Skalierungsfaktor an-

gegeben werden, über den die Umsetzung einer Mausbewegung in eine Rotationskomponente gesteuert werden kann.

- *Rotation um die z-Achse*

In der ursprünglichen Konzeption der virtuellen Sphäre durch Chen ist neben der freien Drehung um x- und y-Achse ebenfalls eine Rotation um die senkrecht zur Projektionsfläche stehenden z-Achse vorgesehen. Für den modalen Wechsel zwischen xy- und z-Rotation wird zwischen Bewegungen innerhalb und außerhalb der angedeuteten virtuellen Sphäre unterschieden. Eine Bewegung innerhalb wird in ein xy-Rotationsverhalten übersetzt, eine Kreisbewegung außerhalb der virtuellen Sphäre führt zu einer Rotation um die z-Achse. Auch hier wird von Chen der Skalierungsfaktor vorgegeben, da bei einem vollständigen Umfahren der Virtual Sphere-Außenhülle eine z-Rotation um 360° postuliert wird.

In der Implementierung von Sun ist dieser zusätzliche Rotationsmodus nicht vorgesehen; hier muss eine Rotation um die z-Achse durch eine Komposition von xy-Rotationen erzeugt werden (Drehung der z-Achse auf die xy-Ebene, Durchführen der gewünschten Rotation und Drehung der z-Achse in ihre ursprüngliche Lage).



## 5. Interaktion mit dem virtuellen Rubik's Cube

Im vorangegangenen Kapitel wurden allgemeine Techniken der Interaktion in virtuellen dreidimensionalen Räumen beschrieben und auf die besonderen Bedingungen der Desktop-VR eingegangen. Basierend auf den hierbei gewonnenen Erkenntnissen werden im folgenden Kapitel Interaktionsmetaphern entwickelt, um diese in einer prototypischen Umsetzung an einem virtuellen Modell des Rubik's Cube eingehend zu untersuchen.

### 5.1 Interaktionsformen am Rubik's Cube

#### 5.1.1 Inspektion der Würfelseiten

Beim Spiel mit dem Rubik's Cube, also beim Versuch einer zielgerichteten Lösung des Würfels, kommt es entscheidend darauf an, sich ein Bild von der Verteilung der einzelnen Farbflächen zu machen. Auf diese Weise kann der im nächsten Schritt zu bewegende Teilwürfel identifiziert werden; auf Grund der räumlichen Lage dieses Teilwürfels zu seiner anvisierten Zielposition werden die durchzuführenden Rotationsbewegungen der Würfelscheiben geplant.

Jede der sechs Würfelseiten besteht aus neun Teilelementen – der Rubik's Cube setzt sich also aus insgesamt 54 verschiedenen Farbflächen zusammen. Die im Abschnitt 3.1.2 beschriebene Beschränkung des menschlichen Kurzzeitgedächtnis lässt die Komplexität der Inspektionsaufgabe am Rubik's Cube erahnen: Der ungeübte Spieler wird nicht in der Lage sein, die Gesamtkonfiguration des Würfels – also die Verteilung der einzelnen Farbflächen über die sechs Seiten – komplett zu erfassen (geschweige denn, diese Konfiguration entsprechend der durchgeführten Rotationsbewegungen anzupassen)<sup>31</sup>.

---

<sup>31</sup> Die Zeiten, welche in der inoffiziellen Weltrekordliste der sogenannten *Speedcuber* für die Lösung eines 3x3x3-Würfels angegeben sind, lassen jedoch erahnen, zu welchen kognitiven Leistungen das menschliche Gehirn mit dem entsprechenden Training in der Lage ist: Der aktuelle Rekord für die Lösung eines gemischten Würfels liegt bei unglaublichen 11 Sekunden [speed02].

Der Inspektion von einzelnen Würfelseiten kommt also eine besondere Bedeutung zu und Metaphern für diese Interaktionsoperation müssen den Anwender möglichst optimal unterstützen, sodass er sich auf die eigentliche Aufgabe konzentrieren kann: Die Lösung des Zauberwürfels.

### **5.1.2 Manipulation am Rubik's Cube**

Über ein speziell konstruiertes Innenelement lassen sich die äußeren Würfel des Rubik's Cube um alle drei Raumachsen rotieren. Interaktionswerkzeuge für den virtuellen Cube müssen dem Anwender daher die Möglichkeit zur Verfügung stellen, bestimmte Drehscheiben auszuwählen und die Richtung einer durchzuführenden Drehbewegung zu spezifizieren.

Ein einzelner Teilwürfel gehört zu jedem Zeitpunkt drei verschiedenen Drehscheiben an, über die sich seine Position und Orientierung im Gesamtwürfel verändern lässt. Es genügt daher nicht, die Selektion einer Drehscheibe über einen einfachen Picking-Mechanismus für einen Teilwürfel zu realisieren: Um eine Drehscheibe eindeutig auszuwählen, muss zusätzlich zur Wahl des zu bewegenden Teilwürfels die gewünschte Rotationsachse angegeben werden.

## **5.2 Der Rubik's Cube: Ein geeignetes Versuchsobjekt?**

Vor der Konzeption von Bedien-Metaphern für die oben beschriebenen Interaktionsaufgaben soll hier zunächst erläutert werden, was den Rubik's Cube als geeignetes Modell für die Umsetzung und zum Test dieser Metaphern qualifiziert.

### **5.2.1 Allgemeinheit, Übertragbarkeit**

Die im Rahmen dieser Arbeit zu entwickelnden Interaktionsmetaphern sollen auf allgemeine Anwendungsfälle übertragbar sein. Es ist also wünschenswert, dass sich die Interaktionsaufgaben leicht auf Grundoperationen für allgemeine Objekte in virtuellen 3D-Räumen zurückführen lassen (beispielsweise Selektion, Rotation und Translation).

Gute Metaphern für sehr spezifische und an ein bestimmtes Modell gebundene Interaktionsaufgaben mögen die Interaktion mit diesem einen Modell vorbildlich unterstützen, lassen sich in der Regel aber nur schwer auf andere Anwendungszusammenhänge übertragen.

In diesem Sinne genügen die oben beschriebenen Interaktionsformen am Rubik's Cube dem Anspruch der Übertragbarkeit: Inspektionsoperationen sind in der beschriebenen oder einer ähnlichen Form relevant für viele Bereiche der 3D-Interaktion (beispielsweise im Fahrzeugbau, der wissenschaftliche Datenanalyse, oder auf dem Gebiet der Molekularbiologie).

Da die Inspektion als eine Sonderform der Navigation durch allgemeine 3D-Räume aufgefasst werden kann, ergeben sich bei geeigneten Metaphern außerdem Möglichkeiten, die im Rahmen der Benutzerstudie gefundenen Ergebnisse auf andere Anwendungsfälle zu übertragen.

Ähnliches gilt für die Manipulation am Rubik's Cube: Die Rotation von Teilen eines 3D-Objekts um verschiedene Raumachsen ist ein so grundlegender Vorgang, dass Metaphern hierfür leicht auf andere Anwendungen übertragbar sein werden. Die Manipulationsaufgabe am Würfel setzt sich zudem aus unterschiedlichen Selektionstasks zusammen (unter anderem der Wahl von Teilwürfeln, Drehscheiben oder Rotationsachsen). Da Metaphern für die Würfelmanipulation auch diese Selektionsaufgaben abdecken müssen, sollten sich Erkenntnisse für diese Grundaufgaben der 3D-Interaktion ableiten lassen.

### **5.2.2 Angemessene Komplexität**

Ein sehr allgemeines Modell läuft leicht Gefahr, zu einfach und zu wenig aussagekräftig zu sein, wohingegen ein komplexes und sehr spezifisches Modell Gefahr läuft, dem oben formulierten Anspruch an Übertragbarkeit nicht gerecht zu werden.

Aus der Beschreibung der einzelnen Interaktionsformen am Rubik's Cube geht hervor, dass die einzelnen Aufgaben nicht durch triviale Interaktionstools abzubilden sind. So wird die Manipulation am Rubik's Cube zum Beispiel durch die Notwendigkeit, verschiedene Parameter während eines Interaktionszyklus zu definieren, zu einem durchaus komplexen Vorgang, der nach geeigneten metaphorischen Konzepten verlangt, um eine angemessene Bedienung zu ermöglichen (die untenstehende Abbildung 5.1 veranschaulicht die vielfältigen Möglichkeiten der Rotation von Teilscheiben am Zauberwürfel; die dargestellte Rotationsfolge bewegt einen Eckwürfel auf eine definierte Zielposition).

Ein weiterer Komplexitätsgrad ergibt sich durch den typische Ablauf beim Spiel mit dem Zauberwürfel. Dieser besteht aus einem fortgesetzten Wechsel zwischen

den beiden grundlegenden Interaktionsmodi Inspektion und Manipulation: Nach der Analyse der aktuellen Konfiguration wird der nächste Zug geplant und durchgeführt, anschließend die sich ergebene neue Konfiguration untersucht, um sich wiederum durch einen geeigneten Zug der Lösung des Würfels zu nähern.

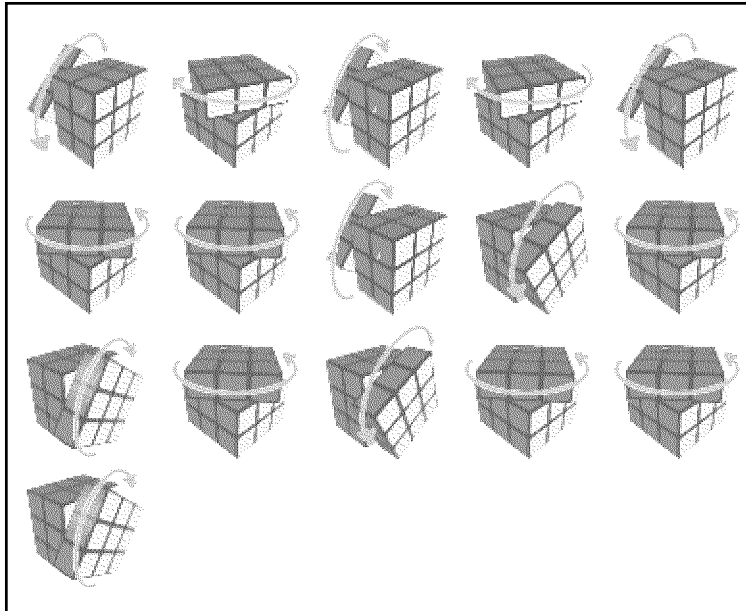


Abbildung 5.1: Rotationen am Rubik's Cube [rubik02]

Dieser fortgesetzte Wechsel zwischen unterschiedlichen Interaktionsmodi macht es erforderlich, während des Interaktionsdesigns eine sinnvolle Verschaltung dieser beiden Modi zu konzipieren und so eine flüssige Bedienung der Anwendung zu gewährleisten.

### 5.2.3 Vergleich von realer und virtueller Schnittstelle

Gerade durch die unmittelbare Vergleichsmöglichkeit zwischen realem und virtuellem Objekt wird sich die Güte der einzelnen Interaktionsmetaphern beweisen müssen: Erfolgt die Interaktion mit dem virtuellen Modell über die Schnittstelle Maus/Tastatur/Display, so geschieht die Interaktion mit dem realen Objekt über eine Schnittstelle, die unmittelbarer kaum sein kann: mit den eigenen zwei Händen.

Beim Spiel mit dem realen Würfel konzentriert sich der Spieler vor allem darauf, geeignete Züge zum Platzieren einzelner Teilwürfel zu finden. Um die für einen Zug notwendigen Drehbewegungen durchzuführen, muss keine weitere geistige



Arbeit aufgewendet werden: Eine Hand richtet den Würfel geeignet aus, die andere greift an der Drehscheibe an und führt die gewünschte Rotation aus.

Die Bedienung des virtuellen Rubik's Cube sollte ähnlichen Bedingungen genügen – das bedeutet, der Spieler soll sich auch hier in erster Linie auf die Planung seiner Züge konzentrieren, die Rotation von Drehscheiben sollte möglichst intuitiv durchzuführen sein. Je intuitiver dem Anwender die vom System angebotene Interaktionsmetapher erscheint, umso weniger stark wird die Behinderung durch die 2D-Schnittstelle empfunden werden.

An dieser Stelle soll bereits auf eine wichtige Unterscheidung aufmerksam gemacht werden: Die Durchführung einer *einzelnen* Rotationsbewegung am realen Würfel ist – wie soeben beschrieben – sehr intuitiv; die zielgerichtete Ausführung einer *Abfolge* von Rotationsbewegungen<sup>32</sup> erfordert dagegen strategische Planung. Dieser Umstand ist vor allem bei der Durchführung der empirischen Untersuchung zu berücksichtigen und wird im Abschnitt 7.1.1 gesondert behandelt.

#### **5.2.4 Einschränkungen**

Neben den beschriebenen Vorteilen, die den Zauberwürfel als geeignetes Versuchsobjekt auszeichnen, sei an dieser Stelle auf zwei Einschränkungen hingewiesen:

Der Reiz beim Spiel mit dem Zauberwürfel liegt gerade darin, dass es für den Laien eine große Herausforderung darstellt, einen ungeordneten Würfel komplett zu lösen: Einer solchen Lösung des Zauberwürfels liegt keine triviale Lösungsstrategie zugrunde.

Bei der Untersuchung der entwickelten Metaphern besteht daher die Gefahr, dass sich ein ungeübter Anwender sehr stark auf die Entwicklung einer geeigneten Lösungsstrategie konzentriert und die Anwendung der eigentlichen Metaphern in den Hintergrund gerät. Diese besondere Bedingung wird bei der Konzeption der Benutzerstudie zu berücksichtigen sein.

---

<sup>32</sup> beispielsweise, um einen bestimmten Teilwürfel mit der korrekten Orientierung an seine Zielposition zu bewegen – ohne dass hierbei bereits ausgerichtete Teilwürfel aus ihrer Ordnung gebracht werden

Die Umsetzung der hier entwickelten Metaphern am Zauberwürfel führt zwangsläufig dazu, dass besondere Bedingungen der Navigation, wie sie zum Beispiel in weitläufigen 3D-Welten mit vielen unterschiedlichen Objekten auftreten, nicht untersucht werden können. Im Mittelpunkt bei Rubik's Cube steht also nicht die Bewegung des Anwenders *in einer weitläufigen Szene*, sondern vielmehr die Inspektion eines *einzelnen Objektes* als Sonderform der Navigation. Ergebnisse sind daher auch im wesentlichen für diese spezielle Navigationsform zu erwarten.

### 5.3 Inspektionsmetaphern

#### 5.3.1 Virtual Sphere

Die in Abschnitt 4.3.2 beschriebene Virtual Sphere-Metapher dient in Form der Implementierung von Java3D als Referenz für die Inspektion der Würfelseiten.

Mit Hilfe dieses Interaktionstools kann der Zauberwürfel frei um die x- und die y-Achse gedreht werden; eine Rotation des Würfels um die z-Achse wird erreicht, indem die z-Achse zunächst in die xy-Ebene gedreht, die gewünschte Operation in dieser Ebene durchgeführt und die z-Achse abschließend wieder in die ursprüngliche Lage rotiert wird. Durch Angabe eines Skalierungsfaktors kann die Geschwindigkeit der Rotation gesteuert werden.

Die Virtual Sphere als Inspektionsmetapher erlaubt es dem Anwender, den Zauberwürfel beliebig im Raum zu orientieren, um auf diese Weise einen Überblick der aktuellen Würfelkonfiguration zu erhalten; eine bestimmte Würfelorientierung wird nicht gesondert ausgewiesen.

Durch ihre weite Verbreitung sind viele Anwender von 3D-Anwendungen mit dem Virtual Sphere Konzept vertraut; es ist daher davon auszugehen, dass die Teilnehmer der Benutzerstudie in der Bedienung dieser Referenzmetapher geschult sind und die virtuelle Sphäre routiniert zur Inspektion des Zauberwürfels einsetzen können.

Neben den hier geschilderten Vorteilen sind bei dem Virtual Sphere-Konzept auch einzelne negative Aspekte erkennbar, die im Folgenden beschrieben werden sollen.

Eine erste Beobachtung betrifft die Position des Mauszeigers im Laufe eines längeren Interaktionszyklus: Soll der Würfel in einer einzelnen Interaktion fortgesetzt

in eine bestimmte Richtung gedreht werden, entfernt sich der Mauszeiger immer weiter vom Rotationszentrum<sup>33</sup>.

Dies führt beispielsweise dazu, dass bei einer längeren Inspektionsinteraktion die Rotation des Würfels durch entfernte Bewegungen des Mauszeigers gesteuert wird (siehe Abbildung 5.2); ein Umstand, der dem zugrunde liegenden metaphorischen Konzept zuwider läuft – die mit dem Mauszeiger zu rotierende virtuelle Kugel bewegt sich gleichsam mit der Maus vom eigentlich zu rotierenden Objekt weg. Problematisch wird dies vor allem dann, wenn der Anwender mit dem Mauszeiger den Rand des Bildschirms erreicht und eine weitere Rotation in der gewünschten Richtung erst durch eine Neupositionierung der Maus möglich ist.

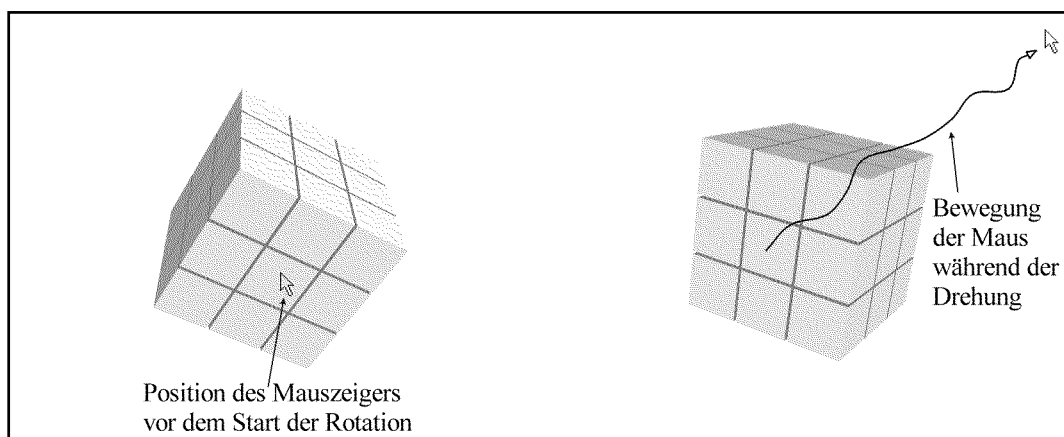


Abbildung 5.2: Mausbewegung bei der Virtual Sphere-Metapher

Eine weitere Beobachtung betrifft den prinzipiellen Vorteil der Virtual Sphere: Die Möglichkeit der freien Rotation. Gerade ungeübten Anwendern fällt es schwer, durch gezielte Rotationsbewegungen um x- und y-Achse zu einer *bestimmten Orientierung* des Zielobjekt zu gelangen. In diesem Zusammenhang sei auf die bereits zitierte Beobachtung Chens verwiesen, dass Probanden bei Rotationsoperationen im Wesentlichen einen einzigen Freiheitsgrad steuern (vgl. Abschnitt 4.3.2) und zusammengesetzte Rotationen eine erhöhte Konzentration des Anwenders erfordern<sup>34</sup>.

---

<sup>33</sup> die Maus muss für eine konstante Rotation um eine Achse immer weiter in eine bestimmte Richtung bewegt werden

<sup>34</sup> insbesondere dann, wenn wie in den üblichen Implementierungen der Virtual Sphere, Rotationen um die z-Achse durch drei aufeinanderfolgende Rotationen um verschiedene Raumachsen erfolgen müssen

Abschließend sei auf die eingeschränkte Undo-Funktionalität der Virtual Sphere-Metapher hingewiesen: Wenn eine längere Rotationsoperation nicht zum gewünschten Ergebnis geführt hat, ist in der Regel eine ebenso aufwändige Rotationsbewegung notwendig, um die ursprüngliche Ausrichtung des Objekts – annähernd – wiederherzustellen<sup>35</sup>.

### 5.3.2 Eckpunkt-Metapher

In Abschnitt 4.3.1 wurde das Verfahren der Viewpoint-Navigation beschrieben, bei dem der Anwender gezielt zu bestimmten Punkten der Szene geführt und die Kamera mit einem definierten Winkel auf die Szene gerichtet wird, um auf diese Weise mit minimalem Aufwand möglichst optimale Ansichten des umgebenden 3D-Raums zu erhalten.

Die Vorteile dieses Navigationsverfahrens sollen im Folgenden auf eine Inspektionsmetapher für den Zauberwürfel übertragen werden. Zu diesem Zweck werden hier zunächst die unterschiedlichen Ansichten auf das Cube-Objekt beschrieben:

---

<sup>35</sup> Ein Effekt, der auch bei einer unachtsamen, relativ schnellen Mausbewegung auftritt, bei der das rotierte Objekt gewissermaßen „durchdreht“ und keine kontrollierte Rotation erfolgt.

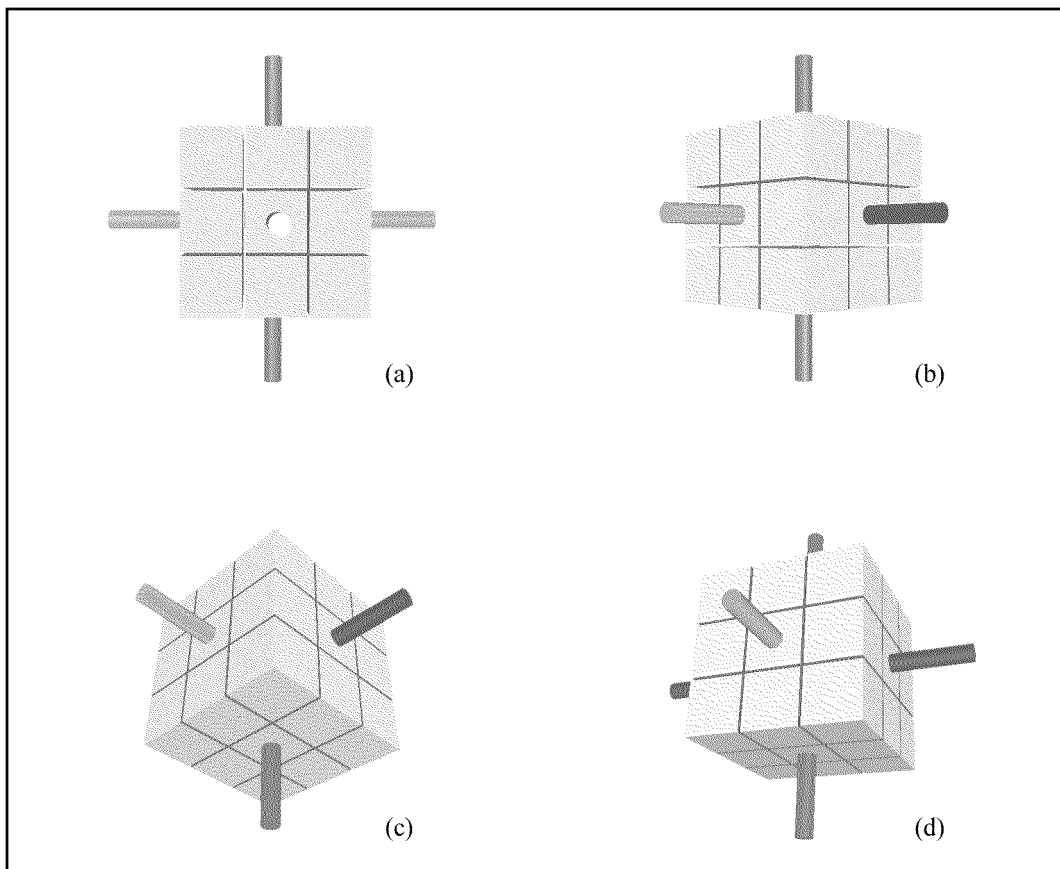


Abbildung 5.3: Unterschiedliche Ansichten des Rubik's Cube

In Abbildung 5.3 (a) ist eine frontale Ansicht auf den Zauberwürfel dargestellt. Der Anwender erkennt allein die neun Farbflächen der Seite, die ihm zugewandt ist. Abbildung 5.3 (b) und (c) zeigen den gleichmäßig angeschnittenen Blick auf zwei beziehungsweise drei Würfelseiten; der Anwender erhält eine Übersicht der 18 (b) beziehungsweise 27 (c) Farbflächen auf den zu ihm hin ausgerichteten Farbflächen.

Teil (c) der Abbildung stellt insofern also eine ausgewiesene Orientierung des Würfel-Objekts dar, als dass der Anwender hier maximal viele Teilflächen gleichzeitig im Blick hat.

In Abbildung 5.3 (d) ist eine Variation der Dreiseiten-Ansicht dargestellt, in der die Seiten nicht mehr gleichmäßig angeschnitten sind; vielmehr existiert hier eine

ausgewiesene Vorderseite und eine relativ stark angeschnittene Seiten- beziehungsweise Unterfläche<sup>36</sup>.

Löst man sich von dem speziellen Anwendungsfall des Rubik's Cube, so lässt sich ganz allgemein feststellen, dass verschiedene Ausrichtungen eines dreidimensionalen Objekts im Raum und damit verbunden unterschiedliche Ansichten dieses Objekts einen unterschiedlichen Informationsgehalt aufweisen oder bestimmte Informationen mehr oder weniger gut erkennbar machen. Um diesen Zusammenhang anschaulich zu machen, stelle man sich beispielsweise die molekulare Darstellung eines medizinischen Wirkstoffes vor, bei dem sich spezifische Andockpunkte an verschiedenen Stellen auf den Außenseiten des Moleküls befinden und Ansichten auf diese Punkte dadurch besonders ausgezeichnet werden.

Der Eckpunkt-Metapher liegt genau dieser Ansatz von *ausgewiesenen Ansichten* zugrunde: Beim Rubik's Cube scheint es vorteilhaft, möglichst viele Teilflächen des Würfels gleichzeitig im Blick zu haben; außerdem möchte der Anwender zwischen Ansichten auf unterschiedliche Seiten hin- und herschalten können, um sich über die aktuelle Konfiguration des Würfels zu vergewissern.

Konzeptionell werden bei der Eckpunkt-Metapher die Viewpoints der Szenen-Navigation *auf das Objekt* übertragen und auf diese Weise eine Inspektion ermöglicht. In Abbildung 5.4 ist dieses Prinzip veranschaulicht.

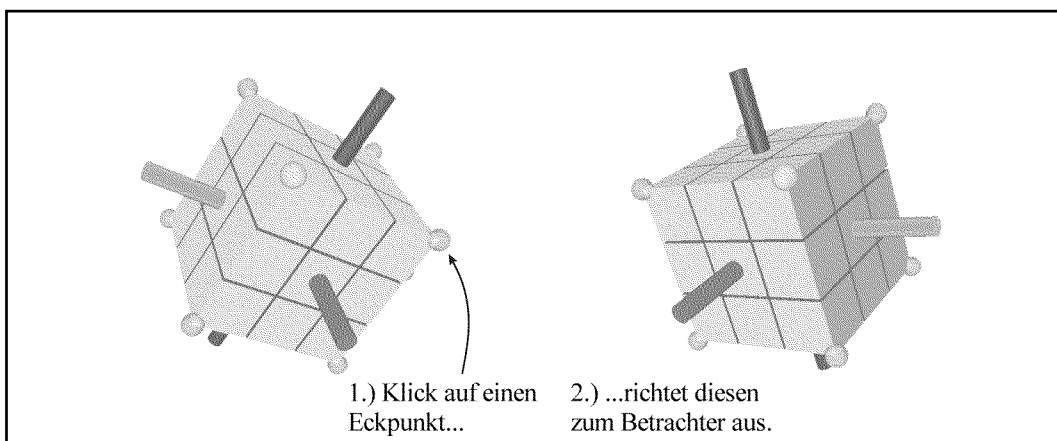


Abbildung 5.4: Funktion der Eckpunkt-Metapher

---

<sup>36</sup> Diese Ansicht spielt für Interaktionen eine besondere Rolle, bei denen es auf eindeutige Raumrichtungen (oben/unten – links/rechts) ankommt. Als Beispiel sei hier auf die Frame of Reference-Metapher verwiesen, die weiter unten eingeführt wird.

Die Eckpunkt-Metapher erlaubt es dem Anwender, auf einfache Weise zwischen verschiedenen Ansichten des Zauberwürfels zu wechseln: Wird durch einen Klick auf einen entfernten Eckpunkt dieser nach vorn rotiert, stellt ein weiterer Klick auf den gerade wegrotierten Eckpunkt die alte Ansicht wieder her.

Anhand der Eckpunkt-Metapher lässt sich der in Abschnitt 4.2.4 eingeführte Begriff der 3D-Metapher noch einmal anschaulich erläutern: Dem Anwender kann die Funktion dieses Inspektionswerkzeugs plausibel gemacht werden, indem darauf verwiesen wird, dass Eckpunkte des Würfels mit Hilfe des Mauszeigers „geangelt“ und dadurch nach vorn gezogen werden können. Die Steuerung dieser Metapher erfolgt intuitiv im Rahmen der zwei von der Computer-Maus bereitgestellten Freiheitsgrade, die Anwendung wertet die Benutzeraktion aus und berechnet gemäß des zugrunde liegenden metaphorischen Konzepts die entsprechende Rotationsmatrix.

Im Gegensatz zum Verfahren, auf dem die Umsetzung der virtuellen Sphäre beruht, muss sich der Anwender bei der Bedienung der Eckpunkte-Metapher nicht länger auf die Wahl unterschiedlicher Rotationsachsen konzentrieren, um zu einer bestimmten Würfelansicht zu gelangen. Durch eine entsprechend Abbildung 5.4 ausgerichtete Grundstellung des Würfels sind in jeder Ansicht sieben der insgesamt acht verschiedenen Eckpunkte sichtbar und können für eine Inspektionsoperation angeklickt werden. Lediglich für eine Drehung des Würfels um  $180^\circ$  werden zwei hintereinandergeschaltete Inspektionen benötigt.

Neben den genannten Vorteilen seien an dieser Stelle auch kritische Punkte nicht verschwiegen: Entscheidend für die leichte Bedienung ist die einfache Selektionsmöglichkeit der Sphären-Objekte an den Ecken des Zauberwürfels. Gemäß der in Abschnitt 3.1.2 beschriebenen Grundlagen im Zusammenhang mit Fitts' Law dürfen diese Kugeln nicht zu klein gestaltet werden, um eine akzeptable Erreichbarkeit zu gewährleisten. Werden die Interaktionselemente jedoch zu großzügig dimensioniert, verdecken sie einen unangemessen großen Teil der eigentlichen Szene. Hier ist ein adäquater Kompromiss zwischen diesen gegensätzlichen Anforderungen zu finden.

Die Kugeln an den Eckpunkten des Würfels wurden halbtransparent gestaltet und mit einer Farbe versehen, die in der übrigen Szene nicht auftritt. Auf diese Weise werden diese Objekte als Interaktionselemente kenntlich gemacht und vom eigentlichen Zauberwürfel abgehoben.

Die Eckpunkte-Metapher lässt sich konzeptionell relativ leicht auf allgemeine 3D-Objekte übertragen. Ideal eignen sich hierfür relativ *gleichmäßig aufgebaute, konvexe Körper*, auf denen die Rotationspunkte so angeordnet werden können, dass sie aus unterschiedlichen Perspektiven gut erreichbar sind. Im allgemeinen Fall wird es sich bei diesen Rotationspunkten nicht mehr unbedingt um Eckpunkte des einzelnen Objektes handeln – wodurch die Funktion der Metapher jedoch keineswegs eingeschränkt wird.

Auch bei komplexen Geometrien kann die Eckpunkte-Metapher eingesetzt werden. Sollte die Platzierung der Rotationspunkte *auf dem Objekt* jedoch wegen konkaven Elementen oder einem stark verwinkelten Aufbau nicht vorteilhaft sein, bietet sich hier die Integration einer *halbtransparenten Hülle* an, auf der die Rotationspunkte positioniert werden können und die zu Beginn einer Inspektionsinteraktion eingeblendet wird.

### 5.3.3 ColorMap-Metapher

Die soeben beschriebene Eckpunkte-Metapher ermöglicht eine Navigation zwischen unterschiedlichen Würfelansichten, jedoch werden bei der Inspektion keine objektspezifischen Informationen und Gegebenheiten ausgenutzt. An dieser Stelle setzt die ColorMap-Metapher an, indem sie *Grundannahmen über die Objektstruktur* mit in den Navigationsprozess einbezieht.

Grundlage dieser Metapher ist ein Kartenkonzept, bei dem neben der eigentlichen Szene zusätzlich eine Übersichtskarte über die im Raum vorhandenen Objekte angezeigt wird<sup>37</sup>. Die ColorMap-Metapher erweitert dieses Konzept der Übersichtskarte und integriert semantischen Informationen des zu inspizierenden Objektes.

---

<sup>37</sup> So werden Übersichtskarten beispielsweise in 3D-Computerspielen dazu eingesetzt, dem Spieler seine aktuelle Position – und die Position seiner Gegner – anzuzeigen. In [haik02] wird dieser Ansatz, der sich zunächst nur auf die grundlegende Orientierung des Spielers bezieht, auf Zwecke der Navigation ausgedehnt: In dem dort vorgestellten System kann sich der Anwender zwischen unterschiedlichen, miteinander verbundenen Räumen einer virtuellen räumlichen Umgebung hin- und herbewegen, indem er den gewünschten Raum auf einer Miniaturansicht der Gesamtszene anklickt.



Im speziellen Fall des Rubik's Cube lassen sich den jeweiligen Seiten des Würfels bestimmte Farben zuordnen, die durch die Farbe des Mittelwürfels der jeweiligen Seite eindeutig definiert sind<sup>38</sup>.

Die sechs Einzelfarben des Zauberwürfels werden bei der ColorMap-Metapher in Form einer 2x3-Matrix in der unteren rechten Bildschirmecke angeordnet, wobei einander gegenüberliegende Farbseiten übereinander angeordnet sind. Ein Klick auf eine der Farbflächen rotiert das Würfelobjekt so, dass die zugehörige Würfel-seite zum Betrachter hin gedreht wird.

Der Würfel selbst ist so ausgerichtet, dass eine Seite der Kamera zugewandt ist und zwei weitere Seiten in einem schrägen Anschnitt zu sehen sind. Da sich der Würfel im Laufe der einzelnen Rotationsbewegungen nur um 90° beziehungsweise 180° um jeweils eine der drei Rotationsachsen dreht, bleibt diese prinzipielle Ausrichtung des Würfels während der gesamten Interaktion unverändert (in der untenstehenden Abbildung 5.5 wird dieses Prinzip verdeutlicht).

Bei der Navigation zu einer bestimmten Farbseite lassen sich zwei verschiedene Fälle unterscheiden: Im ersten Fall ist die Farbseite, die der Anwender nach vorn ausrichten möchte, bereits in der Szene sichtbar (es handelt sich also um eine der beiden schräg angeschnittenen Würfel-seiten). In diesem Fall genügt ein Klick auf die entsprechende Fläche in der Farbmatrix, um die Inspektionsoperation abzuschließen.

Im zweiten Fall ist die gewünschte Farbseite noch nicht in der Szene sichtbar, das heißt, es handelt sich um eine der drei verdeckten Würfel-flächen. Hier muss der Anwender zunächst die Farbe der gegenüberliegenden – und damit sichtbaren – Seite ermitteln und kann im zweiten Schritt durch einen Klick auf die sich über beziehungsweise unter dieser Farbe befindlichen Farbfläche in der 2x3-Matrix die gewünschte Seite des Würfels nach vorne ausrichten.

---

<sup>38</sup> In Abschnitt 2.2.1 wurde die grundlegende Struktur des Zauberwürfels beschrieben, aus der sich die besondere Bedeutung der Mittelsteine einer Würfel-seite bestimmt.

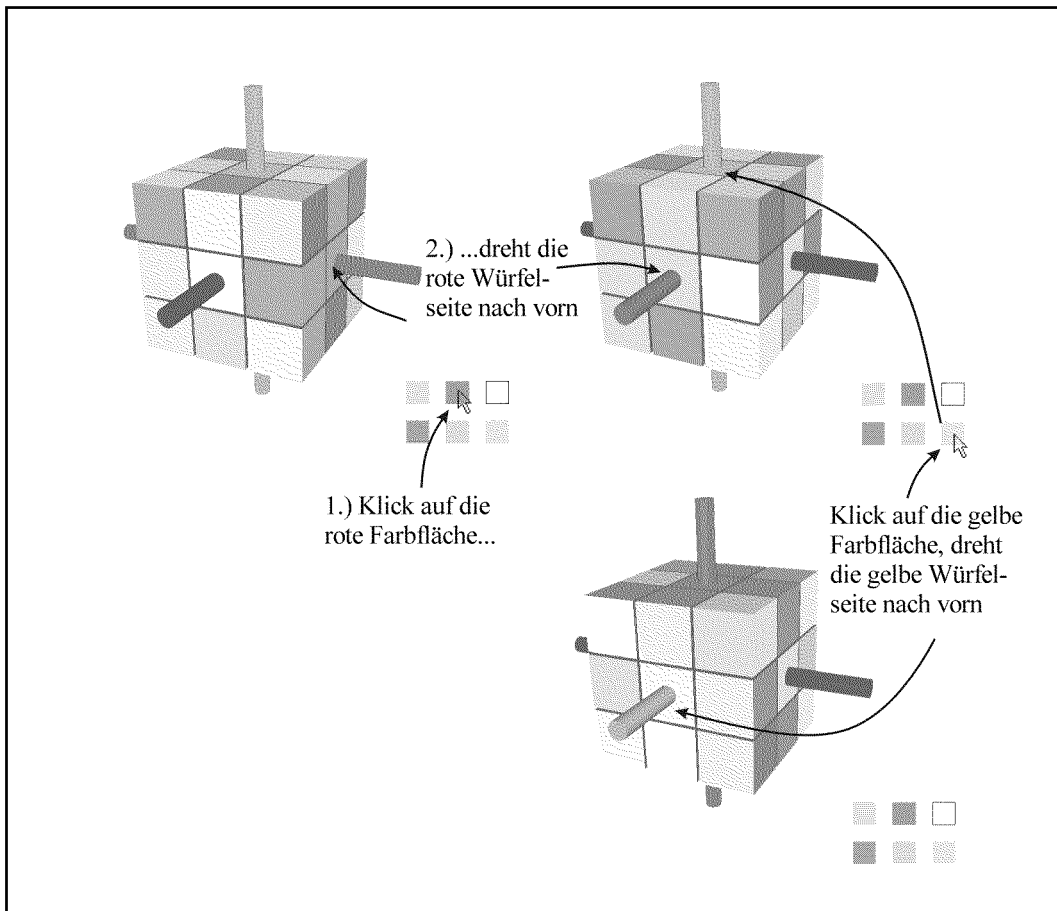


Abbildung 5.5: Funktionsweise der ColorMap-Metapher

Dieses Prinzip, einzelnen Ansichten eines Objekts eine bestimmte Semantik zuzuordnen, lässt sich auf andere Anwendungszusammenhänge übertragen. Bei allgemeinen Objekten muss es sich jedoch nicht mehr zwangsläufig um eine Farbsemantik handeln. Auf einfache Weise lassen sich beispielsweise geometrische Zuordnungen finden (Rück- und Unterseite, linke/rechte Seite); bei strukturell komplexeren Objekten sind weitergehende *inhaltliche Abbildungen* auf die Auswahlmatrix vorstellbar<sup>39</sup>. Im Unterschied zur Eckpunkte-Metapher ist es hierfür nicht erforderlich, dass die Elemente, zu denen navigiert werden soll, in der aktuellen Ansicht bereits direkt am Objekt zu selektieren sind.

<sup>39</sup> beispielsweise die Wahl einer Ansicht auf spezifische Bereiche eines komplexen Moleküls oder auf einzelne Komponenten eines vierteiligen Modells im Fahrzeug- und Maschinenbau

## 5.4 Rotationsmetaphern

### 5.4.1 Rubik's Online-Metapher

Auf der Internet-Seite der Herstellerfirma des Rubik's Cube wird ein Java-Applet bereitgestellt, das ein Spiel mit einem virtuellen Zauberwürfel ermöglicht [rubik02]. Das in dieser Anwendung eingesetzte Verfahren zur Würfelmanipulation dient im Rahmen dieser Arbeit als Referenzmetapher für die Rotation einzelner Drehscheiben. Dieses Verfahren wurde für die in der Benutzerstudie eingesetzte Testanwendung nachimplementiert und soll im Folgenden beschrieben werden.

Die Rubik's Online-Metapher überträgt das Konzept der virtuellen Sphäre für die Würfelinspektion auf die Rotation einzelner Teilscheiben des Würfels. Die zugrunde liegende Intuition legt hierbei nahe, dass jeder einzelne Teilwürfel von einer eigenen virtuellen Sphäre umgeben ist und eine Ziehbewegung mit der Maus auf dieser Sphäre in eine Rotationsbewegung der entsprechenden Teilscheibe des Gesamtwürfels umgesetzt wird.

Den einzelnen Würfelseiten kommt bei diesem Verfahren eine besondere Bedeutung für die Wahl der Rotationsachse zu: Gedreht wird jeweils um die Achse, welche senkrecht zu der Ebene steht, in der die Ziehbewegung der Maus erfolgt ist – Abbildung 5.6 veranschaulicht dieses Prinzip.

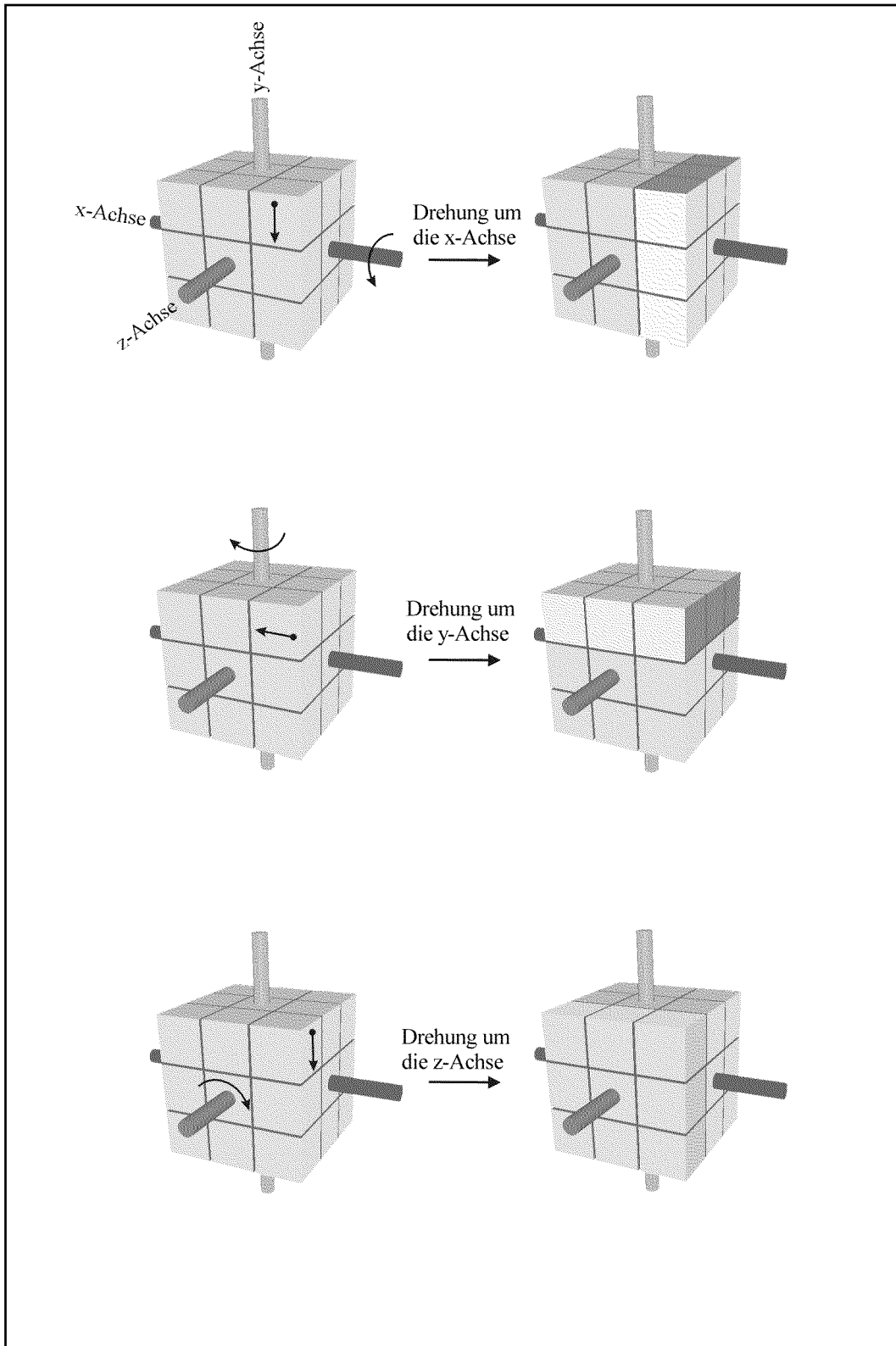


Abbildung 5.6: Die Rubik's Online-Metapher

Um das allgemeine Prinzip der virtuellen Sphäre zur Rotation beliebiger Körper für die Durchführung von Manipulationsoperationen am Rubik’s Cube benutzen zu können, werden bei der Rubik’s Online-Metapher weitgehende Annahmen über die Struktur des zugrunde liegenden Körpers verwendet: Bei der Umsetzung des Mouse-Dragging in die jeweilige Rotationsbewegung einer Teilscheibe wird beispielsweise ausgenutzt, dass ein Teilwürfel zu jedem Zeitpunkt Element von genau drei Teilscheiben ist und auf Grund der Würfelkonstruktion jede dieser Scheiben nur um genau eine Raumachse rotiert werden kann.

Eine weitere Annahme bezieht sich auf die kubische Geometrie der Teilwürfel: Erst durch die Semantik von unterschiedlichen Würfelseiten ist es möglich, durch *zwei* unterschiedliche Dragging-Gesten auf zwei verschiedenen Seiten des Teilwürfels *drei* mögliche Rotationsbewegungen zu unterscheiden.

Diese Annahmen über die innere Struktur des manipulierten Körpers ermöglichen auf der einen Seite eine einfache und bequeme Steuerung des Zauberwürfels, auf der anderen Seite ergeben sich Probleme, sobald man diese Metapher auf andere Objekte übertragen möchte: Im allgemeinen Fall können in der Regel weniger strukturelle Constraints bei der Implementierung dieser Metapher benutzt werden und eine umfangreiche Anpassung des vorliegenden metaphorischen Konzepts wird erforderlich.

Auf Grund ihrer hohen Intuitivität erscheint die Rubik’s Online-Metapher jedoch geeignet als Referenzimplementierung für Manipulationsaufgaben am Zauberwürfel: Alle weiteren Metaphern werden sich an dieser Implementierung messen lassen müssen; Vor- und Nachteile einzelner Konzepte können im Vergleich zu dieser Metapher exemplarisch untersucht werden.

### 5.4.2 Zweipunkt-Metapher

Funktioniert die Wertgabe für die Benutzerinteraktion bei der virtuellen Sphäre und dem Rubik’s Online-Verhalten noch über die Auswertung einer Mausbewegung, so erfolgt bei der Eckpunkte- und ColorMap-Inspektion diese Parameterspezifikation über eine Selektion von Elementen in der dargestellten Szene<sup>40</sup>.

---

<sup>40</sup> Die Wahl eines spezifischen Eckpunktes oder eines Farbfeldes der ColorMap legt fest, um welche Achse und welchen Winkel der Zauberwürfel rotiert werden soll.

Solche Selektionsverfahren für die 3D-Interaktion eignen sich für vielfältige Anwendungsbereiche. So wird in [igarashi98] beispielsweise die Path Drawing-Metapher zur gesteuerten Navigation durch eine 3D-Szene beschrieben, welche auf der allgemeinen Walking-Metapher beruht und diese um spezifische Weginformationen erweitert: Der Benutzer kann zu Beginn der Bewegung den zurückzulegenden Weg mit Hilfe der Maus in der Szene andeuten und die Anwendung berechnet über eine Projektion dieses Weges auf die Grundfläche der Szene den Pfad, auf dem die Bewegung durchgeführt wird.

Die in [kettner93] beschriebene 3-Punkt-Rotationsmetapher ermittelt die für Operation benötigten Parameter ebenfalls mittels Selektionstechniken aus der dargestellten Szene, wobei zwei unterschiedliche Verfahren beschrieben werden:

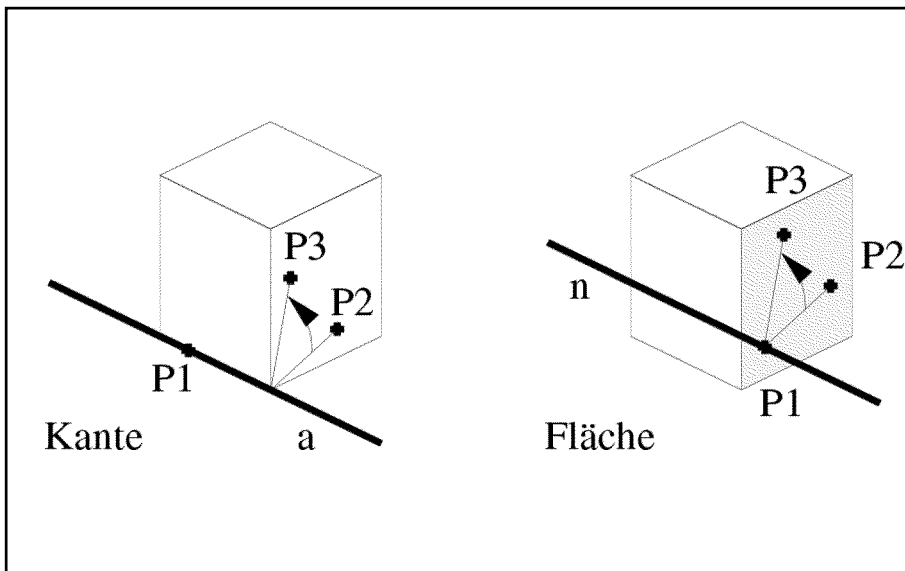


Abbildung 5.7: Varianten der 3-Punkt-Rotationsmetapher [kettner93]

In der in Abbildung 5.7 links dargestellten ersten Variante wird zunächst eine Kante des zu rotierenden Würfels als Rotationsachse selektiert und anschließend ein Start- und ein Zielpunkt für diese Rotation angegeben, welche auf die Seitenfläche des Würfels projiziert werden.

Bei der anschließenden Rotation wird der Startpunkt  $P2$  solange um die spezifizierte Rotationsachse  $a$  gedreht, bis die Ebene durch  $a$  und  $P2$  den Zielpunkt  $P3$  enthält.

Indem die Wahl der Rotationsachse ebenfalls über eine Fläche des zu rotierenden Körpers erfolgt (in der Abbildung 5.7 rechts wird die Rotationsachse durch die Flächennormale der selektierten Außenseite definiert), ist dieses Verfahren auch für Objekte mit allgemeinen Geometrien anwendbar.

Die Zweipunkt-Metapher zur Manipulation am Zauberwürfel folgt diesem Konzept der Wertgabe durch Szenenselektion und wird im Folgenden beschrieben.

Die Zweipunkt-Metapher erlaubt es dem Anwender, zwei Kantenwürfel des Rubik's Cube zu selektieren. Die Position des zweiten selektierten Würfels wird als *Zielposition* für den ersten interpretiert, wodurch eine Rotationsoperation vollständig und eindeutig definiert wird<sup>41</sup>.

Sobald der Anwender einen Kantenwürfel mit der Maus auswählt, wird ihm ein entsprechendes Feedback gegeben: Auf den Außenseiten erscheint eine *Selektionsmarke* in Form eines schwarzen Kreuzes. Die Anwendung erwartet nun die Wahl der Zielposition für den soeben selektierten Startwürfel. Bewegt der Anwender die Maus über andere Kantenwürfel, erscheinen weitere Selektionsmarken, sobald sich der Mauszeiger über einer gültigen Zielposition befindet. Sobald die gewünschte Zielposition markiert ist, löst ein Doppelklick die Rotationsbewegung aus (siehe Abbildung 5.8).

Wurde versehentlich ein Startwürfel selektiert, über den nicht die gewünschte Rotationsoperation ausgeführt werden kann, kann der Benutzer durch einen Klick auf die rechte Maustaste in den Grundzustand der Zweipunkt-Rotation zurückkehren; vorhandene Markierungen auf Teilwürfeln werden gelöscht.

Durch die in der Szene angezeigten Selektionsmarken ist dem Anwender zu jedem Zeitpunkt der aktuelle Zustand der übermittelten Parameter ersichtlich. Auf diese Weise wird eine hohe Kontrolle über die durchzuführenden Operationen erreicht – für die Zweipunkt-Metapher ist daher eine eher geringe Fehlerwahrscheinlichkeit zu erwarten<sup>42</sup>. Dieser Vorteil wird jedoch durch eine größere An-

---

<sup>41</sup> Dies würde beispielsweise bei der Wahl von zwei benachbarten *Eckwürfeln* nicht der Fall sein: Hier existieren zwei in der Drehachse verschiedene Rotationen, durch die der erste Eckwürfel auf die Position des zweiten gedreht werden kann.

<sup>42</sup> Indem bei der Zweipunkt-Metapher alle Zwischenschritte der Interaktion visualisiert werden, unterscheidet sie sich grundlegend vom Grundprinzip der Rubik's Online-Metapher: In der zugrunde liegenden Implementierung dieser Metapher startet die Rotation, sobald aus der Eingabe-

zahl von Interaktionsschritten erkauft, was in der Folge gegebenenfalls zu einer schlechteren zeitlichen Performanz führen kann.

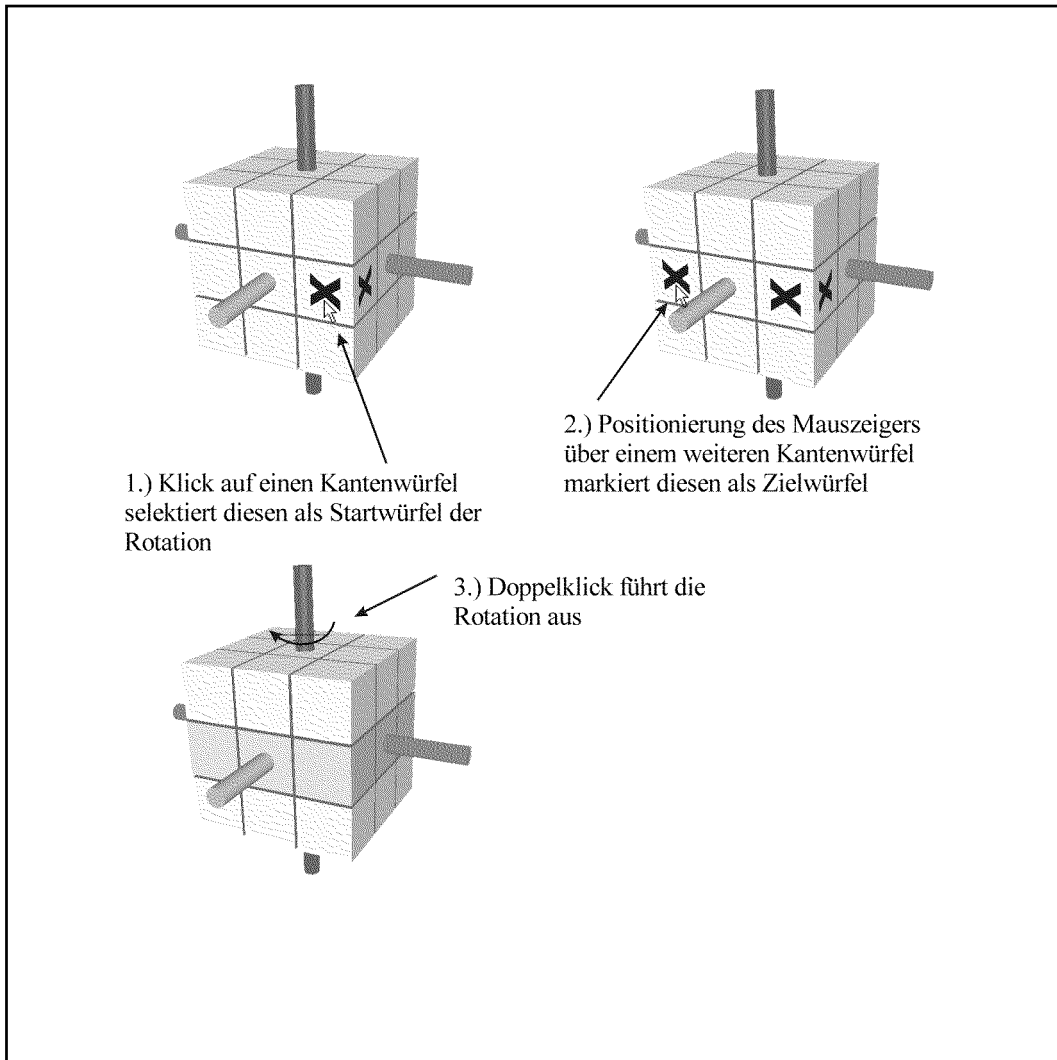


Abbildung 5.8: Die einzelnen Phasen der Zweipunkt-Metapher

### 5.4.3 Spotlight-Metapher

Sind bei der Zweipunkt-Metapher noch drei aufeinanderfolgende Interaktionsschritte nötig, um eine Rotation von Teilscheiben des Rubik's Cube auszulösen,

---

bewegung eindeutige Parameter errechnet werden können. Startet der Anwender zum Beispiel auf der falschen Seite eines Teilwürfels oder führt er die Maus nicht genau genug, resultiert dies sofort in einer unerwünschten Rotationsbewegung.



verfolgt die Spotlight-Metapher das gegensätzliche Konzept: Sie ermöglicht dem Anwender die Definition und den Start einer Rotationsbewegung mit einem einzelnen Mausklick.

Grundlage dieses Interaktionswerkzeugs ist das metaphorische Konzept eines Lichtkegels oder *Spots*, der von der jeweiligen Mausposition in Richtung des Würfelmittelpunkts strahlt. Befindet sich die Maus über dem Szenenhintergrund, wird ihre Position dreidimensional ausgewertet, indem die Mauskoordinaten als Punkt auf der xy-Ebene des globalen Koordinatensystems interpretiert werden<sup>43</sup>.

Auf diese Weise ist es möglich, bis zu vier Würfelseiten des Rubik's Cube senkrecht anzustrahlen, wobei die jeweils angestrahlte Seite als selektiert gilt: Nachfolgende Rotationsoperationen beziehen sich damit auf die so angestrahlte Seite.

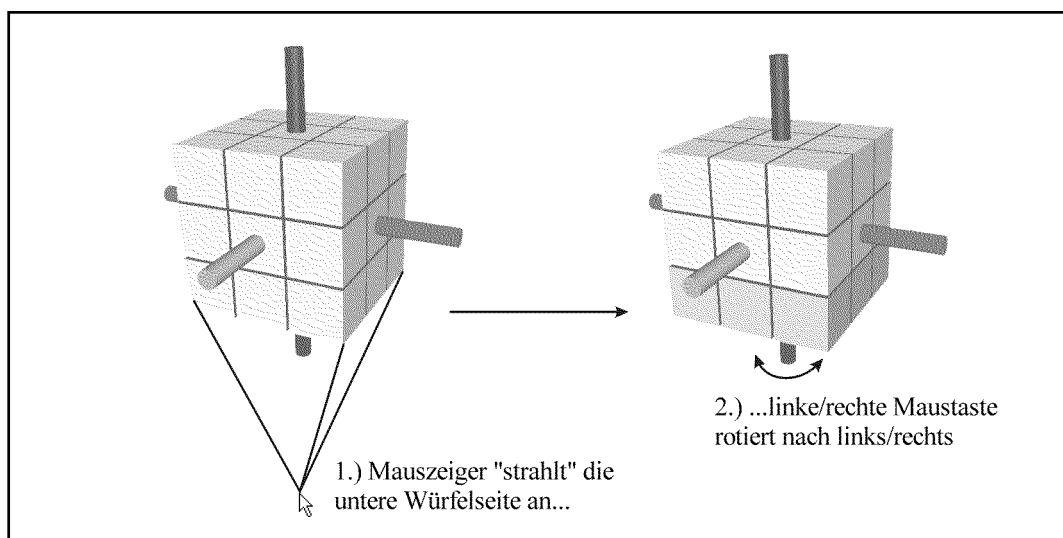


Abbildung 5.9: Spotlight-Metapher, Rotation der unteren Würfelseite

In Abbildung 5.9 ist die Situation für die Rotation der unteren Würfelseite schematisch dargestellt: Der Mauszeiger befindet sich unterhalb des Rubik's Cube und strahlt gemäß dem metaphorischen Konzept in Richtung des Würfelmittelpunkts nach oben. Dadurch wird die untere Würfelseite selektiert und ein Klick auf die linke Maustaste rotiert diese Scheibe nach links (Klick auf rechts rotiert entsprechend nach rechts).

<sup>43</sup> die Bildschirmkoordinate  $(x, y)$  wird dabei also in die Weltkoordinate  $(x', y', 0)$  transformiert

Äquivalent zu dieser Anordnung verhält sich die Metapher, wenn der Mauszeiger *über* dem Würfel positioniert wird: Die obere Seite wird angestrahlt, ein Klick auf die linke/rechte Maustaste rotiert diese Scheibe nach links beziehungsweise nach rechts.

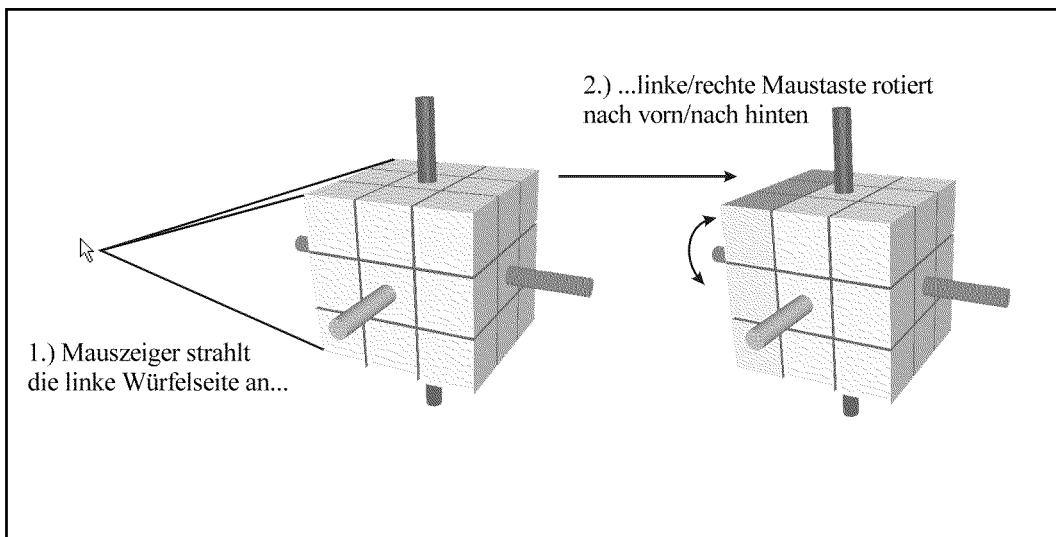


Abbildung 5.10: Spotlight-Metapher, Rotation einer seitlichen Würfelscheibe

Die Situation im Fall einer Position des Mauszeigers *neben* dem Würfel-Objekt veranschaulicht Abbildung 5.10: Im Sinne der Spotlight-Metapher wird die Mauszeiger-Position auf die xy-Ebene des globalen Koordinaten-Systems transformiert und von dieser Koordinate aus ein virtueller Lichtstrahl zum Würfelmittelpunkt hin ausgerichtet.

Im dargestellten Fall wird durch die Mausposition links vom Zauberwürfel die linke Teilscheibe selektiert. Ein Klick mit der linken Maustaste rotiert diese Scheibe *zum Anwender hin*, ein Klick mit rechts rotiert *vom Anwender weg*.

Symmetrisch verhält sich die Situation, wenn der Mauszeiger rechts vom Würfel positioniert wird: Auch hier rotiert die linke Maustaste nach vorn und die rechte Maustaste nach hinten.

Das hier dargestellte Konzept der *impliziten Selektion* über die Position des Eingabezeigers lässt sich unmittelbar auf andere Anwendungszusammenhänge übertragen und kann als verallgemeinerte Form der klassischen Picking-Mechanismen für die Desktop-VR verstanden werden.

So wird beim *Raycasting-Verfahren* zur Selektion von Objekten im Raum ein *virtueller Strahl* vom Augpunkt des Benutzers in die Szene gerichtet, wobei üblicherweise das vom Strahl zuerst getroffene Objekt selektiert wird.

In [bowman01] wird dieses und weitere Verfahren zur Objektselektion für *immersive* virtuelle Umgebungen beschrieben und mit der dort dargestellten Spotlight-Technik bereits eine Erweiterung des Raycasting-Verfahrens erwähnt, mit der das Problem der Selektion kleiner beziehungsweise weit entfernter Objekte überwunden wird.

Die Spotlight-Metapher zur Rotation von Teilscheiben des Rubik’s Cube überführt die in [bowman01] beschriebene Spotlight-Technik auf die speziellen Anforderungen der Desktop-VR: Durch die Einschränkung der 2D-Ein-/Ausgabe ist hier eine freie Positionierung des Eingabezeigers im Raum nur über Umwege möglich. Die Abbildung der Zeigerposition auf 3D-Koordinaten und die Ausrichtung des Spots an festen Bezugspunkten des Objekts (Mittelpunkt des Zauberwürfels) erlaubt jedoch auch im Kontext der Desktop-VR die Anwendung eines vergleichbaren Prinzips.

An dieser Stelle sei auf eine weitere Besonderheit beim Einsatz der Spotlight-Metapher hingewiesen: Dieses Verfahren ermöglicht es insbesondere, mit Teilen eines 3D-Objekts zu interagieren, die in der 2D-Projektion der Szene nicht oder nur stark angeschnitten dargestellt werden. Von der in Abbildung 5.9 selektierten unteren Drehscheibe werden beispielsweise nur sechs Seitenflächen dargestellt, von der in Abbildung 5.10 ausgewählten linken Scheibe sind in der vorliegenden Orientierung des Würfels sogar nur drei Außenflächen sichtbar.

Da es sich bei der Spotlight-Metapher im Wesentlichen um eine Selektionstechnik handelt, bleibt das Feld möglicher Einsatzgebiete nicht auf Rotationsaufgaben beschränkt. Neben der Manipulation beliebiger Parameter der durch den Spot selektierten Objekte (beispielsweise Farbe oder Textur), ist ebenso vorstellbar, die Richtungskomponente des Spots für Translations- oder Skalierungsoperationen des angestrahlten Objekts zu verwenden<sup>44</sup>.

---

<sup>44</sup> So könnte die Spotlight-Metapher beispielsweise auf einfache Weise so erweitert werden, dass sie auch als Inspektionsmetapher für den Rubik’s Cube verwendet werden kann, indem die jeweils angestrahlte Fläche bei einem Mausklick zum Benutzer hin ausgerichtet wird.

Ein Nachteil der Spotlight-Metapher im spezifischen Zusammenhang der Interaktion mit dem Rubik's Cube ist darin zu sehen, dass in einer Würfelorientierung nur *maximal vier Teilscheiben* rotiert werden können. Um die Metapher auf die speziellen Erfordernisse in diesem Anwendungszusammenhang hin anzupassen, wurde im Rahmen der prototypischen Umsetzung folgende Erweiterung implementiert:

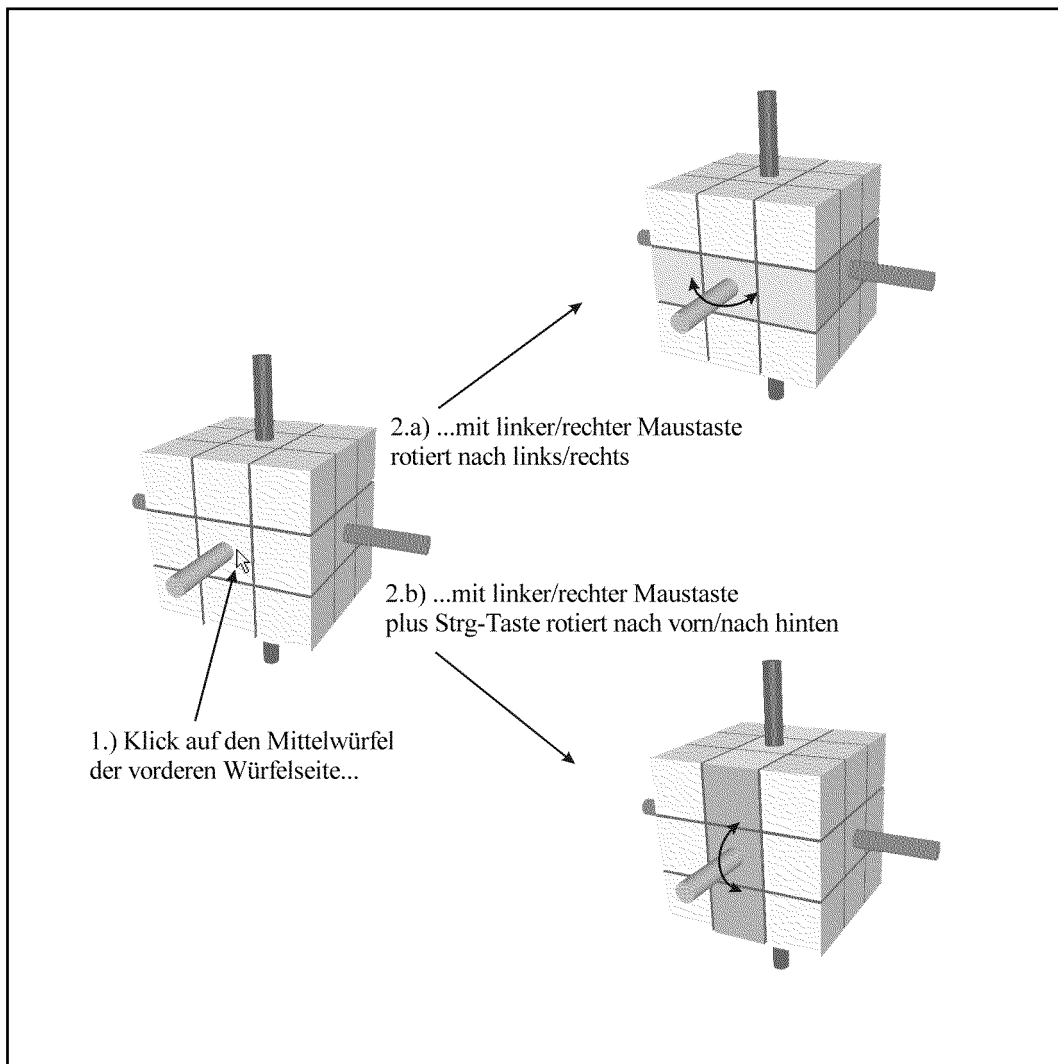


Abbildung 5.11: Erweiterung der Spotlight-Metapher

Um die horizontale und vertikale Mittelscheibe rotieren zu können, wurde der Mittelwürfel der vorderen Würfelseite mit einer speziellen Funktionalität versehen. Vergleichbar dem in Abbildung 5.9 dargestellten Verfahren rotiert ein Klick mit der linken Maustaste auf diesen Teilwürfel die horizontale Mittelscheibe nach links und ein Klick mit der rechten Maustaste nach rechts.

Eine dem Verhalten aus Abbildung 5.10 vergleichbare Funktion wird über eine Modifier-Taste erreicht: Der Klick mit der linken Maustaste bei gedrückter Steuerungstaste rotiert die vertikale Mittelscheibe nach vorn, ein Klick mit der rechten Maustaste rotiert diese Scheibe nach hinten.

#### 5.4.4 Bezugsrahmen-Metapher

Im Gegensatz zu den bisher vorgestellten Metaphern handelt es sich bei der Bezugsrahmen-Metapher um ein rein tastaturgesteuertes Interaktionswerkzeug.

Grundlage für diese Metapher bildet die Art und Weise, wie ein Spieler mit dem realen Zauberwürfel interagiert: Für die Rotation einer Drehscheibe wird diese in der Regel von den Fingern der führenden Hand umfasst, während die andere Hand die übrigen Würfelscheiben fixiert. Die Rotation wird ausgeführt, indem die Drehscheibe mit der sie umfassenden Hand in die gewünschte Zielposition gedreht wird.

Indem eine Hand die zu rotierende Teilscheibe umfasst, wird die Achse, um die diese Scheibe gedreht werden soll, bereits durch die *Position der einzelnen Finger* definiert. Die Fingerpositionen stellen also einen *Bezugsrahmen* dar (*frame of reference*), über den Teile der Operationsparameter bestimmt werden können.

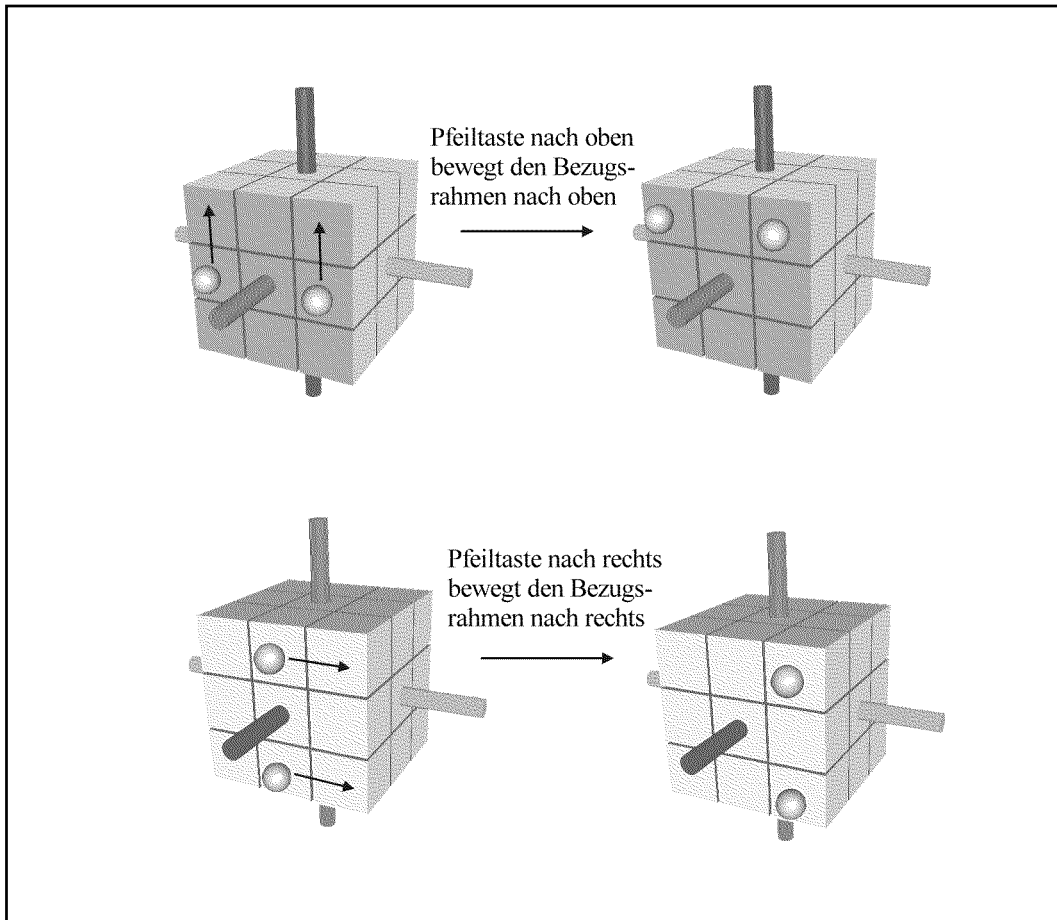


Abbildung 5.12: Bezugsrahmen-Metapher: Bewegen der Fingerpunkte

Die Bezugsrahmen-Metapher basiert auf der beschriebenen Vorstellung einer am Würfel angreifenden Hand und visualisiert diesen Rahmen über die Darstellung abstrakter „Finger“ in Form von vier Kugeln.

Am Realwürfel kann ein minimaler Bezugsrahmen mit Hilfe von drei Fingern einer Hand definiert werden: Zeige- und Mittelfinger greifen an der Oberseite von zwei benachbarten Eckwürfeln an und der Daumen greift die Unterseite des gegenüberliegenden Kantenwürfels. Da beim virtuellen Modell die einfache räumliche Orientierung ein entscheidendes Kriterium für die intuitive Bedienung darstellt, werden hier anstelle *einer* Kugel für den virtuellen Daumen ebenfalls zwei Fingerpunkte angezeigt. Auf diese Weise kann die räumliche Lage des Bezugsrahmens aus jeder Perspektive eindeutig erkannt werden.

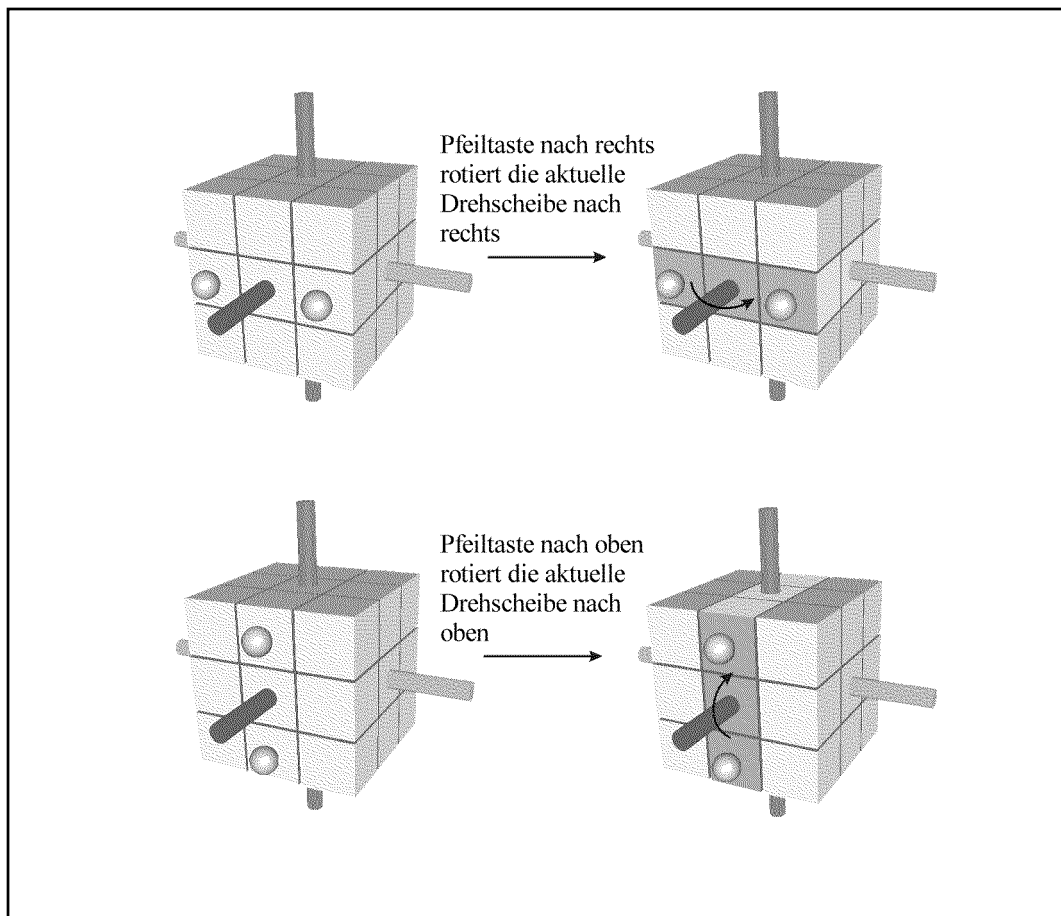


Abbildung 5.13: Bezugsrahmen-Metapher: Rotation einer Teilscheibe

Abbildung 5.12 und Abbildung 5.13 veranschaulichen die Funktionsweise der Bezugsrahmen-Metapher: Die Fingerpunkte können mit Hilfe der Pfeiltasten horizontal und vertikal bewegt werden, sodass die drei Drehscheiben einer Rotationsachse gezielt selektiert werden können.

Je nach Orientierung des Würfels erfolgt die Verschiebung des Bezugsrahmens über das Pfeiltastenpaar „links/recht“ oder „hoch/runter“. Das jeweils andere Pfeiltastenpaar bewirkt die Rotation einer Teilscheibe in die entsprechende Richtung.

Diese Zuordnung der Pfeiltastenfunktion über die Orientierung des Würfels setzt eine geeignete Grundausrichtung des Würfels voraus, sodass eine eindeutige Abbildung auf die zu steuernden Einzeloperationen gewährleistet werden kann.

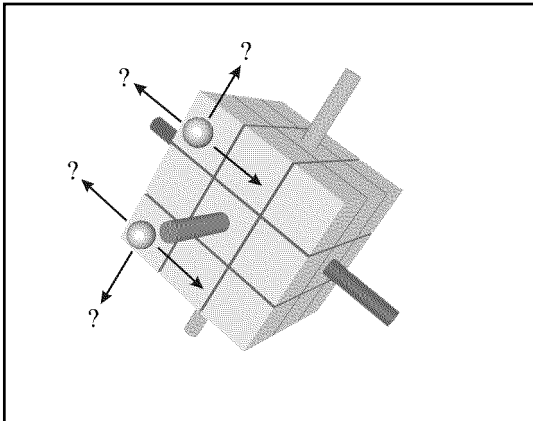


Abbildung 5.14: Problem der Operatorzuordnung

Abbildung 5.14 veranschaulicht dieses Problem: In der vorliegenden Würfelorientierung kann nicht mehr eindeutig bestimmt werden, ob beispielsweise die Pfeiltaste „runter“ den Bezugsrahmen verschieben soll oder ob eine Rotation der aktuellen Drehscheibe durchzuführen ist.



## 6. Implementierung

Die im vorangegangenen Kapitel entwickelten Interaktionsmetaphern wurden im Rahmen einer prototypischen Anwendung implementiert, um in einer durchzuführenden Benutzerstudie ihre Tauglichkeit für die Interaktion mit dem virtuellen Rubik's Cube zu untersuchen.

Im Folgenden werden kurz die eingesetzten Technologien dargestellt, spezielle Techniken bei der Implementierung erläutert und ein Überblick über das Objektmodell der Anwendung gegeben.

### 6.1 Java3D-API

Als Zielplattform für die Implementierung der entwickelten Interaktionskonzepte wurde die Java3D-API von Sun gewählt [sun02]. Die erste Version dieser Entwicklungsumgebung wurde 1998 veröffentlicht und wird oft als Vertreter einer 4. Generation von 3D-Grafik-APIs bezeichnet [mohan98] [barrilleaux01]<sup>45</sup>.

Zu dieser neuen Generation von 3D-APIs werden insbesondere Entwicklungsumgebungen gezählt, die auf einem Szenengraphen-Konzept basieren – einer Baumstruktur, in der alle Objekte der dargestellten Szene verwaltet werden (wie beispielsweise 3D-Objekte, Transformationskomponenten, Licht und Sound), und gleichzeitig eine schnelle Manipulation und schnelles Rendering ermöglichen [mohan98].

Durch ihren objektorientierten Ansatz werden Rendering-Details bei der Programmierung zunehmend transparent, ohne dass hierbei Flexibilität und Effizienz verloren gehen [barrilleaux01].

Ein besonderer Vorteil der Java3D-API besteht zudem in ihrer Verfügbarkeit auf unterschiedlichen Plattformen und in weitgehenden Skalierungsmöglichkeiten für Host-Systeme mit verschiedenen Hardware-Voraussetzungen. So werden neben den klassischen Ein- und Ausgabegeräten auch gängige Sensor- und Display-

---

<sup>45</sup> so benannt auf Grund der drei vorangegangenen 3D-API-Generationen (in chronologischer Reihenfolge): Siggraph CORE, PIGHS und OPEN GL

Verfahren der immersiven VR wie Head Mounted Displays, CAVE-Umgebungen und die Anbindung einer Space Mouse unterstützt [barrilleaux01]. Die hier entwickelten Verfahren sollten sich daher ohne großen Entwicklungsaufwand auch auf immersive Interaktionstechniken erweitern lassen.

## 6.2 Darstellung einzelner Konzepte

### 6.2.1 Behavior-Konzept von Java-3D

Zur Implementierung von Interaktionselementen stellt die Java3D-API mit der abstrakten Behavior-Klasse einen Mechanismus zur Verfügung, über den mit angepasstem Code auf Benutzereingaben reagiert werden kann<sup>46</sup>.

Die Java3D-Klasse *Behavior* stellt die Schnittstelle für diesen Mechanismus zur Verfügung; in ihr werden die folgenden abstrakten Methoden definiert:

- *initialize()*

Mit Hilfe dieser Methode wird der initiale Trigger für das Behavior-Objekt gesetzt. Dies geschieht mit Hilfe eines von der abstrakten *WakeupCriterion*-Klasse abgeleiteten *WakeupOnX*-Objekts, wobei das *X* im Klassennamen für die entsprechende Trigger-Bedingung steht.

Mögliche Trigger können aus AWT-Events bestehen (für Maus- und Tastatureingaben) oder durch ein Kollisionsereignis in der Szene ausgelöst werden. Trigger auf Grund von zeitlichen Bedingungen oder einer internen Kommunikation zwischen verschiedenen Behavior-Objekten sind ebenso möglich und erlauben die Implementierung komplexer Verhaltensmerkmale.

- *processStimulus(java.util.Enumeration criteria)*

In Rahmen der *processStimulus*-Methode erfolgt die eigentliche Berechnung der Reaktion auf eine erfolgte Benutzerinteraktion. Diese Methode wird aufgerufen, sobald die in der *initialize*-Methode gesetzte Trigger-Bedingung erfüllt wurde. In der Aufzählung *criteria* werden alle getriggerten *WakeupCriterion*-

---

<sup>46</sup> Dieses Behavior-Konzept bezieht sich nicht allein auf die Abfrage von Benutzereingaben, sondern kann auch weitere Ereignisse während des Programmablaufs behandeln. Unter anderem werden über die Behavior-Klasse auch Animationsaufgaben in der Szene realisiert. Wir konzentrieren uns bei der weiteren Beschreibung jedoch auf die interaktiven Aspekte des Behavior-Konzepts.

Objekte übergeben und können im weiteren Verlauf der Methode ausgewertet werden.

Im Rahmen der *processStimulus*-Methode können vielfältige Systemantworten generiert werden: So ist es beispielsweise möglich, Objekte des Szenengraphen zu entfernen oder neue Objekte hinzuzufügen; ebenso können Attribute von bestehenden Objekten verändert und Transformationskomponenten angepasst werden<sup>47</sup>.

Indem Behavior-Objekte individuell an einzelne Elemente der 3D-Szene gebunden werden können, wird es möglich, komplexe interaktive Systeme zu entwickeln, in denen sich Benutzereingaben direkt auf Teile des zugrunde liegenden Szenengraphen auswirken.

### 6.2.2 Picking-Mechanismen

Ergänzend zum Behavior-Konzept werden von Java3D Picking-Mechanismen zur Verfügung gestellt, über die es beispielsweise möglich ist, direkt auf das vom Benutzer selektierte Objekt der 3D-Szene zuzugreifen. Diese Verfahren beruhen auf einem sogenannten *PickRay*, also einem Strahl, dessen Verlauf vom Augpunkt des Betrachters in die Szene verfolgt wird und dabei auftretende Kollisionen mit Objekten der Szene der Anwendung zurückgeliefert werden.

Java3D erlaubt darüber hinaus die Definition von beliebigen Picking-Geometrien und verallgemeinert das Picking-Konzept auf unterschiedliche Startpunkte und Ausrichtungen des Pick-Strahls. Dieses Verfahren wurde beispielsweise eingesetzt, um mit Hilfe eines kegelförmigen Picking-Objekts das in Abschnitt 5.4.3 beschriebene Spotlight-Rotationsverfahren zu implementieren. Durch eine ausreichend abstrakte High-Level 3D-API war so die Umsetzung der zugrunde liegenden 3D-Metapher nahezu 1:1 möglich<sup>48</sup>.

---

<sup>47</sup> Das Markieren von Teilwürfeln bei der Selektion von Start- und Zielwürfel im Rahmen des Zweipunkt-Verhaltens nutzt beispielsweise die Möglichkeit aus, einzelne Attribute von Objekten im Szenengraph zu verändern: Das Anzeigen beziehungsweise Ausblenden der Markierung auf den Außenseiten des selektierten Teilwürfels wird auf einfache Weise durch das An- und Ausschalten einer vorbereiteten Textur realisiert.

<sup>48</sup> Das Spotlight des mentalen Modells kann ganz unmittelbar über einen ebenso geformten Pick-Ray im Code realisiert werden.

### 6.3 Implementierung der Interaktionskonzepte

Jedes der im Kapitel 5 entwickelten Interaktionskonzepte wurde in einem eigenen Behavior-Objekt realisiert. Dabei wurde insbesondere darauf geachtet, dass die einzelnen Verfahren beliebig miteinander kombiniert werden können. So können aus Anwendungssicht beispielsweise verschiedene Inspektionsmetaphern völlig unproblematisch zusammenschaltet werden<sup>49</sup>.

Auch bei der Verschaltung von Inspektions- mit unterschiedlichen Rotationsmetaphern wurde auf eine ausreichende *modale Differenzierung* geachtet, so dass Ambivalenzen in der Auswertung von Benutzereingaben vollständig vermieden werden konnten.

Jedes Inspektions- und Rotationsverhalten verfügt gemäß der API-Spezifikation über eigene Trigger-Kriterien; es werden also nur die Ereignisse an die jeweiligen Behavior-Objekte weitergeleitet, die für eine Behandlung der vorliegenden Benutzereingabe relevant ist.

Sofern für einzelne Interaktionsmetaphern die Darstellung von objektgebundenen 3D-Elementen nötig war<sup>50</sup>, wurden diese getrennt von der eigentlichen Behavior-Logik implementiert und können bei der Übertragung auf andere Szenen leicht angepasst oder durch extern geladene Geometrien ersetzt werden.

### 6.4 Darstellung ausgewählter Klassen der vorliegenden Implementierung

Im Folgenden werden ausgewählte Klassen der Rubik's Cube-Implementierung kurz dargestellt und wesentliche Methoden beschrieben:

- *Game.java*

Die *Game*-Klasse steuert den gesamten Spielablauf. Hier wird die 3D-Szene vorbereitet, der Szenengraph aufgebaut und die Wahl von verschiedenen Inter-

---

<sup>49</sup> Auf diese Weise ist eine Kombination von Virtual Sphere- und Eckpunkt-Metapher, wie sie von einzelnen Teilnehmern im Rahmen der Benutzerstudie gewünscht wurde (siehe dort), bereits in der jetzigen Ausbaustufe der Applikation problemlos möglich.

<sup>50</sup> beispielsweise die RotationHandles der Eckpunkte-Metapher oder die Fingerpunkte der Bezugsrahmen-Metapher

aktionsmetaphern verwaltet. Außerdem übernimmt die *Game*-Klasse die Aufzeichnung der Log-Dateien während der Durchführung der Benutzerstudie.

- *RubiksCube.java*

In dieser Klasse wird die Geometrie und das Verhalten des Zauberwürfels gekapselt. Über die *RubiksCube*-Klasse kann ein Zauberwürfel-Objekt in den Szenengraph gehängt werden und durch den Aufruf der *startRotation*-Methode mit den entsprechenden Parametern einzelne Drehscheiben des Würfels rotiert werden. Über bereitgestellte Methoden kann zudem der Repeater-Modus initiiert oder der Zauberwürfel gemischt werden.

Die *RubiksCube*-Klasse ist prinzipiell bereits für beliebige Würfelgrößen ausgelegt (neben dem Standard aus 3x3x3 Teilwürfeln beispielsweise auch für die 2x2x2- oder 4x4x4-Variante).

- *SmallCube.java*

Repräsentiert einen Teilwürfel des Rubik's Cube. Die *SmallCube*-Klasse verwaltet unter anderem die optische Markierung und die Markierung auf Grund einer Spotlight-Selektion. Für eine kontrollierte Manipulation des Szenengraphen wird in jedem *SmallCube*-Objekt zusätzlich ein Index-Feld geführt und mit einer *compareTo*-Methode die Vergleichbarkeit von *SmallCube*-Paaren sichergestellt.

- *CubeMode.java*

Hilfsklasse für die Auswertung der Ausrichtung des Zauberwürfel-Objekts. Um die einzelnen Würfelorientierungen unterscheiden zu können, wird die Position von zwei Teilwürfeln ausgewertet, deren Ortsvektoren linear unabhängig sind. Die genaue Kenntnis einer solchen Würfelausrichtung wird beispielsweise bei der Abbildung der Pfeiltasten-Funktion im Rahmen der Bezugsrahmen-Metapher benötigt.

- *MetaphorSelectionPanel.java*

Verwaltet das Panel, über das die einzelnen Interaktionsmetaphern aktiviert oder deaktiviert werden können. Hier sind außerdem Buttons für das Mischen des Würfels und für den Start der Aufzeichnung während der Repeater-Durchgänge integriert.

- *RotationRepeater.java*  
Steuert einen Repeater-Durchgang. Von der Anwendung vorgegebene Rotationen sind vom Anwender zu reproduzieren und werden auf Identität überprüft.
- *CubeMouseRotate, RotationHandlesBehavior und ColorChooserBehavior*  
Behavior-Klassen für die Inspektionsmetaphern (Virtual Sphere-, Eckpunkte- und ColorMap-Metapher)
- *RubiksOnlineBehavior, SpotRotateBehavior und FrameOfReferenceBehavior.java*  
Behavior-Klassen für die Rotationsmetaphern (Rubik's Online-, Spotlight- und Bezugsrahmen-Metapher)

## **7. Empirische Untersuchung**

Die im vorangegangenen Kapitel vorgestellte prototypische Implementierung der konzipierten Interaktionsmetaphern dient als Grundlage für die Durchführung einer Benutzerstudie.

Im Rahmen dieser Studie soll untersucht werden, wie die einzelnen Interaktionswerkzeuge im Praxistest abschneiden und wie ihre Bedienung und Handhabbarkeit von den Versuchsteilnehmern eingeschätzt wird.

Die erfassten Messergebnisse werden ausgewertet und mit den in Kapitel 3 und 1 beschriebenen Grundlagen in Beziehung gesetzt, um auf diese Weise Gründe für das jeweilige Abschneiden der einzelnen Metaphern angeben zu können.

### **7.1 Vorbereitungen**

#### **7.1.1 Berücksichtigung der Komplexität des Rubik's Cube**

Im Rahmen der Untersuchung in Kapitel 5.2, inwieweit sich der Rubik's Cube als Versuchsobjekt für eine empirische Untersuchung der entwickelten Interaktionsmetaphern eignen würde, wurde bereits darauf hingewiesen, dass eine solche Untersuchung sorgfältig vorzubereiten sein würde und als Grund das Problem einer nicht-trivialen Lösungsstrategie für den Zauberwürfel genannt.

Dieser Umstand führt beim realen Würfel beispielsweise dazu, dass der Spieler am sorgfältig gemischten Würfel zwar anfangs noch relativ einfach eine Reihe von ordnenden Zügen durchführen kann, dann aber relativ schnell vor dem Dilemma steht, zur Platzierung weiterer Teilwürfel bereits geordnete Bereiche des Rubik's Cube zeitweise umordnen zu müssen.

Mit einer gewissen Praxis – und gegebenenfalls dem Blick auf diverse Internet-Seiten, die Tipps und Tricks für die Lösung des Zauberwürfels geben<sup>51</sup> – eignet sich der Spieler zunehmend Strategien für die Lösung von Teilaufgaben an.

Mit der entsprechenden Übung kann zum Beispiel eine einzelne Würfelseite bereits relativ zügig gelöst werden; die für diese Aufgabe erforderlichen Rotationsabfolgen, die ein Würfelement in die gewünschte Position drehen, bestehen hier noch aus wenigen Teilschritten und können relativ leicht gelernt werden.

Die Komplexität des Rubik's Cube wird deutlich, wenn man versucht, auf einer Teillösung aufbauend, weitere Steine an ihre endgültige Stelle im Würfel zu bewegen. Die Anzahl der für eine einzelne Operation erforderlichen Teilschritte steigt stark an und bedeuten für den ungeübten Spieler eine besondere Konzentrationsanstrengung bei der Durchführung<sup>52</sup>.

Es scheint vor diesem Hintergrund daher nicht angemessen, die Teilnehmer der Benutzerstudie mit der Lösung des gesamten Zauberwürfels zu konfrontieren, um auf diese Weise Aussagen über die Performance der einzelnen Interaktionsmetaphern zu treffen: Statt sich auf die Besonderheiten der angebotenen Metaphern zu konzentrieren, wären die Teilnehmer vor allem mit ihrer jeweiligen Lösungsstrategie beschäftigt.

### **7.1.2 Unterteilung der Benutzerstudie**

Um einen angemessenen Komplexitätsgrad für die Untersuchung zu gewährleisten, wurde die Benutzerstudie in zwei Durchgänge unterteilt: Im sogenannten Repeater-Durchgang geht es zunächst darum, Rotationsoperationen, die von der Anwendung vorgegeben werden, zu reproduzieren. Auf diese Weise können die einzelnen Interaktionsmetaphern völlig unabhängig von der jeweiligen Expertise des Teilnehmers am Zauberwürfel untersucht werden.

---

<sup>51</sup> stellvertretend für eine ganze Reihe ähnlicher Seiten sei an dieser Stelle auf [marshall01], [dhonline02] und [reimann02] verwiesen

<sup>52</sup> Um einen Kantenwürfel bei der Lösung der zweiten Ebene des Rubik's Cube korrekt auszurichten, sind beispielsweise bis zu 14 Einzeloperationen nötig.



Im zweiten Schritt wird den Teilnehmern die Aufgabe gestellt, eine einzelne Würfel­seite des Rubik's Cube zu lösen, wobei es hierbei lediglich um die Lösung *ei­ner Seite* und nicht um eine *komplette Ebene* des Zauberwürfels gehen soll<sup>53</sup>.

Um ein objektives Maß für die Expertise am Zauberwürfel an der Hand zu haben, wird vor der Durchführung dieses zweiten Versuchsteils zunächst die Zeit ermit­telt, die ein Teilnehmer für die Lösung einer Seite am realen Zauberwürfel benö­tigt. Es wird erwartet, dass der Unterschied zwischen einer Interaktion mit den ei­genen Händen und einer Interaktion über den Umweg der 2D-Schnittstelle, gerade durch diese unmittelbare Gegenüberstellung von realem und virtuellen Zauber­würfel besonders deutlich wird.

### 7.1.3 Wahl geeigneter Metaphern-Kombinationen

Bereits bei der Beschreibung der Bezugsrahmen-Metapher in Abschnitt 5.4.4 wurde auf mögliche Einschränkungen bei der Kombination einzelner Rotations­ mit bestimmten Inspektionsmetaphern hingewiesen<sup>54</sup>.

Im Rahmen der Benutzerstudie sollen die einzelnen Metaphern möglichst unbe­ einflusst durch Seiteneffekte einer solchen möglicherweise nachteiligen Meta­ phern-Kombination untersucht werden, gleichzeitig sollen die einzelnen Ver­ suchsteile zu möglichst aussagekräftigen Ergebnissen führen.

Durch vorgelagerte Entwicklungstests wurden die folgenden Metapher-Kombina­ tionen in die engere Wahl genommen:

- *Virtual Sphere und Rubik's Online*

In diesem Versuchsteil werden die beiden Referenz-Metaphern für Inspektion und Rotation miteinander kombiniert. Indem diese beiden Metaphern in beson­ derer Weise auf die Anforderungen bei der Interaktion mit dem Rubik's Cube zugeschnitten wurden, wird hier der Maßstab festgesetzt, an dem sich die übrigen Metaphern werden messen lassen müssen.

---

<sup>53</sup> Das Ziel ist also, neun einheitliche Farbflächen auf einer Würfel­seite zu versammeln – ohne, dass dabei die übrigen Seitenflächen der Teilwürfel dieser Seite bereits korrekt zu ihren jeweiligen Farbseiten ausgerichtet sein müssen.

<sup>54</sup> Im konkreten Beispiel ergeben sich gegebenenfalls Schwierigkeiten, sobald das Würfelobjekt mittels der Virtual Sphere-Metapher frei im Raum ausgerichtet wurde, in der Folge dann aber kei­ ne eindeutigen Richtungsangaben für die Steuerung der Rotationsoperationen mehr abzuleiten sind.

### - *Eckpunkt- und Zweipunkt-Metapher*

Diese beiden Konzepte für Inspektion und Rotation lassen sich kombinieren, ohne dass sich durch die speziellen Ausformungen der Einzelmetaphern nachteilige Effekte ergeben.

Indem die Eckpunkte-Metapher den Rubik's Cube bereits so ausrichtet, dass drei Seiten des Würfels gleichzeitig zu sehen sind, ist zu erwarten, dass eine Reihe von Rotationsbewegungen bereits in dieser Grundausrichtung des Würfels durchzuführen sind.

Da einzelne Kantenwürfel, die für die Definition einer Rotation zu selektieren sind, sich gegebenenfalls an einer entfernten Stelle des Würfels befinden oder relativ stark angeschnitten dargestellt werden, können in einer solchen Situation zielgerichtete Inspektionsoperationen jedoch durchaus sinnvoll erscheinen<sup>55</sup>.

### - *ColorMap- und Spotlight-Metapher*

Ähnlich wie die Bezugsrahmen-Metapher beruht auch das Spotlight-Konzept zur Rotation von Drehscheiben auf einer angemessenen Ausrichtung des Gesamtwürfels. Sobald die einzelnen Würfelseiten durch eine freie Rotation mit den durch das Koordinatensystem aufgespannten Raumebenen einen Winkel von 45 Grad bilden, werden durch den Spot mehrere Seitenflächen angestrahlt und eine Zuordnung der einzelnen Rotationsparameter ist nicht mehr eindeutig möglich<sup>56</sup>.

Indem die ColorMap-Metapher die Grundausrichtung des Würfels unverändert beibehält, wird erwartet, dass bei der Kombination dieser Metapher mit dem Spotlight-Konzept keine störenden Seiteneffekte auftreten.

---

<sup>55</sup> um ganz im Sinne von Fitts' Law (siehe Abschnitt 3.1) die für die Selektion anzusteuern Fläche zu vergrößern

<sup>56</sup> Da im Konzept der Virtual Sphere jede Ausrichtung des Würfels gleichberechtigt ist und daher keine besonderen Würfelorientierungen ausgewiesen werden, ist diese Metapher beispielsweise schlecht für die Kombination mit der Spotlight-Rotation geeignet.

- *ColorMap- und Bezugsrahmen-Metapher*

Bei der Wahl der ColorMap-Metapher für diesen Versuchteil kommen ähnliche Überlegungen zum Tragen, wie sie im Teil ColorMap/SpotLight bereits beschrieben wurden.

Ergänzend dazu sei hier angemerkt, dass durch die spezielle Konzeption der Bezugsrahmenmetapher eine eher geringe Anzahl von Inspektionsoperationen zu erwarten ist: Die Visualisierung des Bezugsrahmens erfolgt in einer Form, die in der Grundausrichtung des Würfels jede der 18 Rotationsoperationen<sup>57</sup> prinzipiell möglich macht.

## 7.2 Durchführung

### 7.2.1 Einteilung der Benutzergruppe

Die Benutzerstudie wurde mit insgesamt neun Teilnehmern (zwei weiblich, sieben männlich) im Alter zwischen 23 und 29 Jahren durchgeführt. Die Teilnehmer wurden vor Versuchsbeginn gebeten, ihre Erfahrung auf folgenden Gebieten einzuschätzen (0: keine Erfahrung, 4: Experte):

- *WIMP-Systeme*

Hierbei soll die Erfahrung im Umgang mit mausbasierten Window-Systemen eingeschätzt werden. Da die entwickelten Metaphern zum großen Teil über die Maus gesteuert werden, wird hier eine Beeinflussung der Ergebnisse erwartet, sollten sich einzelne Teilnehmer in der Bedienung der Maus unsicher fühlen.

Alle neun Teilnehmer bewerten ihre eigene Erfahrung auf diesem Gebiet mit 4 (Experte).

- *3D-Anwendungen*

Die Teilnehmer sollen ihre Erfahrung mit desktopbasierten 3D-Anwendungen einschätzen. An dieser Stelle wird nicht zwischen Anwendungen verschiedener Bereiche unterschieden: Wissenschaftliche 3D-Visualisierungen sollen genauso berücksichtigt werden wie beispielsweise 3D-Spiele. Ausschlaggebend ist hier die Erfahrung mit virtuellen 3D-Räumen.

---

<sup>57</sup> Rotation von 3 Drehscheiben um 3 Achsen in 2 Richtungen:  $3 \cdot 3 \cdot 2 = 18$

Die Teilnehmer schätzen ihre Erfahrung auf diesem Gebiet im Mittel mit  $2,67 \pm 0,94$  ein<sup>58</sup>.

### - *3D-Eingabe-Geräte*

Hier geht es um die Erfahrung mit 3D-Eingabe-Geräten wie Datenhandschuh oder Space-Mouse. Teilnehmer mit Erfahrung auf diesem Gebiet werden gegebenenfalls vergleichende Aussagen zum Effekt der auf zwei Freiheitsgraden beschränkten Interaktion machen können.

Die Teilnehmer der Studie schätzen ihre Erfahrung auf diesem Gebiet jedoch mit  $0,78 \pm 1,23$  ein, sind also eher unerfahren im Umgang mit diesen höherdimensionalen Eingabegeräten. Außer einem Teilnehmer, der seine Erfahrung mit 4 einschätzt, geben sich alle übrigen Probanden Werte zwischen 0 und 1.

### - *Virtual Sphere-Metapher*

Bei der Virtual Sphere-Metapher handelt es sich um ein auf dem Gebiet der 3D-Anwendungen recht weit verbreitete Interaktionstechnik. Teilnehmer der Studie, die in der Bedienung dieser Metapher bereits geübt sind, werden routinierter im Umgang mit der Referenzimplementierung sein.

Die Teilnehmer schätzen ihre Expertise mit der Virtual Sphere-Metapher durchschnittlich mit  $2,33 \pm 1,15$  ein.

### - *Zauberwürfel*

Da es im zweiten Versuchsteil um das Lösen einer Würfelseite gehen soll, wird hier gesondert die Erfahrung mit dem Zauberwürfel abgefragt. Beteiligt sich der Teilnehmer an diesem zweiten Versuchsteil, wird der hier angegebene Wert durch Lösungsversuche am realen Würfel ergänzt.

Leider schätzen die Teilnehmer ihre Erfahrung am Zauberwürfel nur unterdurchschnittlich ein, nämlich mit  $1,67 \pm 1,05$ . Nur ein Teilnehmer bewertet seine Erfahrung auf diesem Gebiet mit 4.

Abgesehen von der Erfahrung mit dem Zauberwürfel lassen die hier aufgenommenen Werte auf ein relativ homogenes Teilnehmerfeld schließen, das mit klassi-

---

<sup>58</sup> hier und bei allen folgenden Werten gibt die zweite Zahl die Standardabweichung an

schen Desktop-Anwendungen wohlvertraut ist, auf dem Gebiet der immersiven virtuellen Realität jedoch eher unerfahren ist.

### **7.2.2 Erläuterung der Metaphern und Training**

Vor Durchführung der Repeater-Durchgänge mit den ausgewählten Metapher-Kombinationen wird den Versuchsteilnehmern die am nächsten Repeater-Durchgang beteiligte Inspektions- und Rotationsmetapher erläutert. Anschließend erhalten die Teilnehmer Gelegenheit, sich die Bedienung soweit anzueignen, bis sie sich bei der Interaktion mit den Metaphern sicher fühlen.

In dieser Trainingsphase wird darauf geachtet, dass die Repeater-Durchgänge von den einzelnen Teilnehmern auf einem vergleichbaren Niveau durchgeführt werden. Werden während des Trainings individuelle Fehlbedienungen identifiziert, wird der Teilnehmer auf diese hingewiesen, um mögliche Missverständnisse vor Versuchsstart auszuschließen.

### **7.2.3 Durchführen der Repeater-Durchgänge**

Sobald sich der Teilnehmer mit der Bedienung der jeweiligen Metapher-Kombination vertraut ist, startet er den Repeater-Durchgang für dieses Metaphern-Paar. Hierbei werden von der Anwendung insgesamt 20 einzelne Rotationen vorgegeben, wobei vor dem Start der jeweils nächsten Rotation abgewartet wird, bis der Teilnehmer die aktuelle Rotation reproduziert hat.

Wurde die vorgegebene Rotationsbewegung korrekt nachgeahmt, erscheint ein grünes Licht am oberen Bildschirmrand, ist dem Teilnehmer ein Fehler unterlaufen, wird ein rotes Licht eingeblendet. Die nächste Rotation, die von der Anwendung vorgegeben wird, startet unmittelbar nach Abschluss der vom Teilnehmer durchgeführten Rotationsinteraktion.

Die Teilnehmer werden vor dem Start des ersten Repeater-Durchgangs darauf hingewiesen, dass bei den folgenden Versuchen sowohl die *Schnelligkeit* als auch die *Korrektheit* der Einzelaktionen im Vordergrund stehen.

Die Anwendung protokolliert die Zeit, die für den gesamten Repeater-Durchgang benötigt wurde. Zusätzlich werden die durchgeführten Inspektionsoperationen gezählt und für jede Rotation vermerkt, ob diese der von der Anwendung vorgegebenen Rotation entsprochen hat.

### 7.2.4 Subjektive Bewertung der Metaphern

Nach jedem Repeater-Durchgang werden die Teilnehmer gebeten, ihren individuellen Eindruck bei der Bedienung der einzelnen Interaktionsmetaphern wiederzugeben. Hierbei geht es zunächst um eine umgangssprachliche Beschreibung des ersten Eindrucks. Zusätzlich werden vom Versuchsleiter besondere Beobachtungen notiert (beispielsweise charakteristische Fehler oder individuelle Strategien bei der Bedienung einzelner Metaphern).

Nach Abschluss des letzten Repeater-Durchgangs sollen die Teilnehmer die *Einzelmetaphern* nach folgenden Kriterien bewerten:

- *Intuitivität*

Der Teilnehmer soll einschätzen, wie intuitiv er die Bedienung einer einzelnen Metapher empfunden hat. Bei diesem Kriterium spielt die kognitive Last, die durch die Handhabung der Metapher erzeugt wird, eine besondere Rolle: Je stärker sich der Teilnehmer auf die Bedienung konzentrieren muss, desto weniger intuitiv ist die jeweilige Metapher.

- *Effizienz*

Bei diesem Kriterium geht es darum abzuschätzen, wie schnell die angebotene Metapher die gewünschte Interaktion unterstützt. Eine solche Effizienz-Betrachtung muss nicht zwangsläufig mit dem Kriterium der Intuitivität zusammenfallen: So kann man sich vorstellen, dass ein Anwender sich bei einer Interaktion zwar stark konzentrieren muss, diese sich dann aber auch relativ schnell durchführen lässt.

- *Erlernbarkeit*

Beim diesem letzten Punkt soll eingeschätzt werden, wie aufwändig eine Metapher zu erlernen ist. Dabei soll der Teilnehmer insbesondere auch seine eigene Erfahrung während der Trainingsphase berücksichtigen.

Wie bereits bei der Einschätzung des eigenen Erfahrungsprofils sind auch bei der subjektiven Bewertung der Einzelmetaphern Antwortmöglichkeiten von 0 (sehr schlecht) bis 4 (sehr gut) vorgegeben.

### **7.2.5 Vergleich zwischen realem und virtuellem Zauberwürfel**

Teilnehmer mit entsprechender Expertise beim Lösen des Zauberwürfels wurden gebeten, auch am zweiten Teil der Benutzerstudie teilzunehmen, in dem eine Würfelseite sowohl am realen als auch am virtuellen Modell zu lösen war.

In diesem Versuchsteil war zunächst eine Seite des realen Würfels zu ordnen. Hierfür wurde der Zauberwürfel mit einer Reihe von zufälligen Rotationen gemischt; zusätzlich wurde sichergestellt, dass auf der zu lösenden Seite außer der Mittelfläche keine weitere Farbfläche in der Zielfarbe vorhanden war. Der Mischvorgang war für den Teilnehmer nicht sichtbar.

Der Zauberwürfel wurde dem Teilnehmer übergeben, wobei die zu lösende Fläche nach oben ausgerichtet worden ist. Es wurde sowohl die Zeit als auch die benötigte Anzahl von Rotationen für die Lösung dieser Seite protokolliert. Insgesamt wurde dieser Versuchsteil dreimal wiederholt.

Im Anschluss wurde den Teilnehmern eine vergleichbare Aufgabe am virtuellen Modell gestellt: Auch hier ging es darum, eine Würfelseite beim zuvor gemischten Würfel zu lösen, wobei hier drei Durchgänge mit drei unterschiedlichen Metapher-Kombinationen durchgeführt worden sind (Virtual Sphere und Rubik's Online, Eckpunkt und Zweipunkt beziehungsweise ColorMap und Bezugsrahmen).

Die benötigte Zeit und die Anzahl der Rotationsoperationen bis zur Lösung der Teilfläche wurden von der Anwendung in einer Log-Datei protokolliert.

## **7.3 Messwerte Repeater-Versuche**

Auf den folgenden Seiten werden die im Rahmen der Benutzerstudie erfassten Messwerte zusammengestellt und graphisch ausgewertet. Auf den Seiten 90 bis 92 wird die subjektive Einschätzung der Inspektionsmetaphern durch die Teilnehmer dargestellt, auf den Seiten 93 bis 95 folgen die Einschätzungen der Rotationsmetaphern.

Auf Seite 96 werden die benötigten Zeiten für die einzelnen Repeater-Durchgänge zusammengestellt und auf Seite 97 folgen die Angaben zur Anzahl der benötigten Inspektionsoperationen und zur Fehlerzahl in diesem Versuchsteil.

In die Diagramme ist die jeweilige Standardabweichung als Fehlerindikator mit aufgenommen worden.

### 7.3.1 Inspektion: Intuitivität

<i>Teilnehmer</i>	<i>Virtual Sphere</i>	<i>Eckpunkt</i>	<i>ColorMap</i>
1	4	4	0
2	3	1	2
3	3	2	1
4	4	1	3
5	4	4	1
6	4	4	3
7	4	3	1
8	4	2	1
9	3	4	1
Mittelwert:	3,67	2,78	1,44
Std.-Abw.:	0,47	1,23	0,96

Tabelle 7.1: Intuitivitätsbewertung der Inspektionsmetaphern

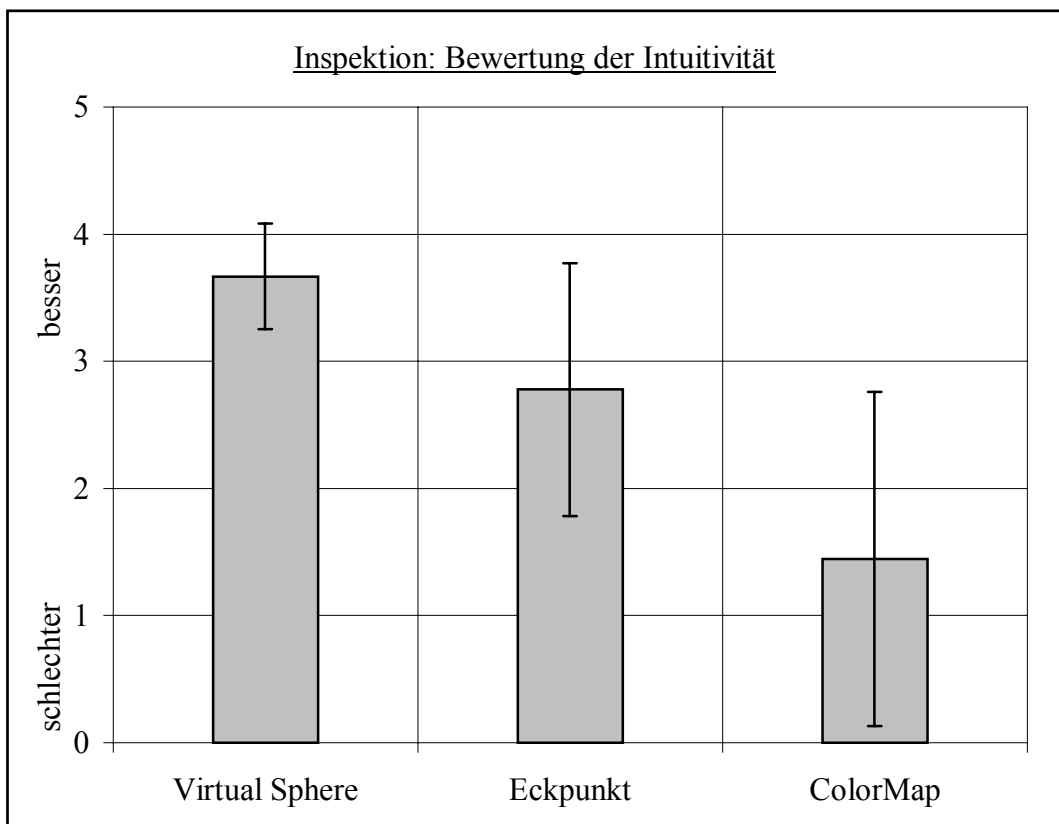


Abbildung 7.1: gemittelter Intuitivitätswert mit Standardabweichung



### 7.3.2 Inspektion: Effizienz

<i>Teilnehmer</i>	<i>Virtual Sphere</i>	<i>Eckpunkt</i>	<i>ColorMap</i>
1	4	4	2
2	3	2	1
3	3	2	0
4	2	2	3
5	4	2	0
6	2	4	3
7	3	2	1
8	4	3	2
9	3	4	2
Mittelwert:	3,11	2,78	1,56
Std.-Abw.:	0,74	0,92	1,07

Tabelle 7.2: Effizienzbewertung der Inspektionsmetaphern

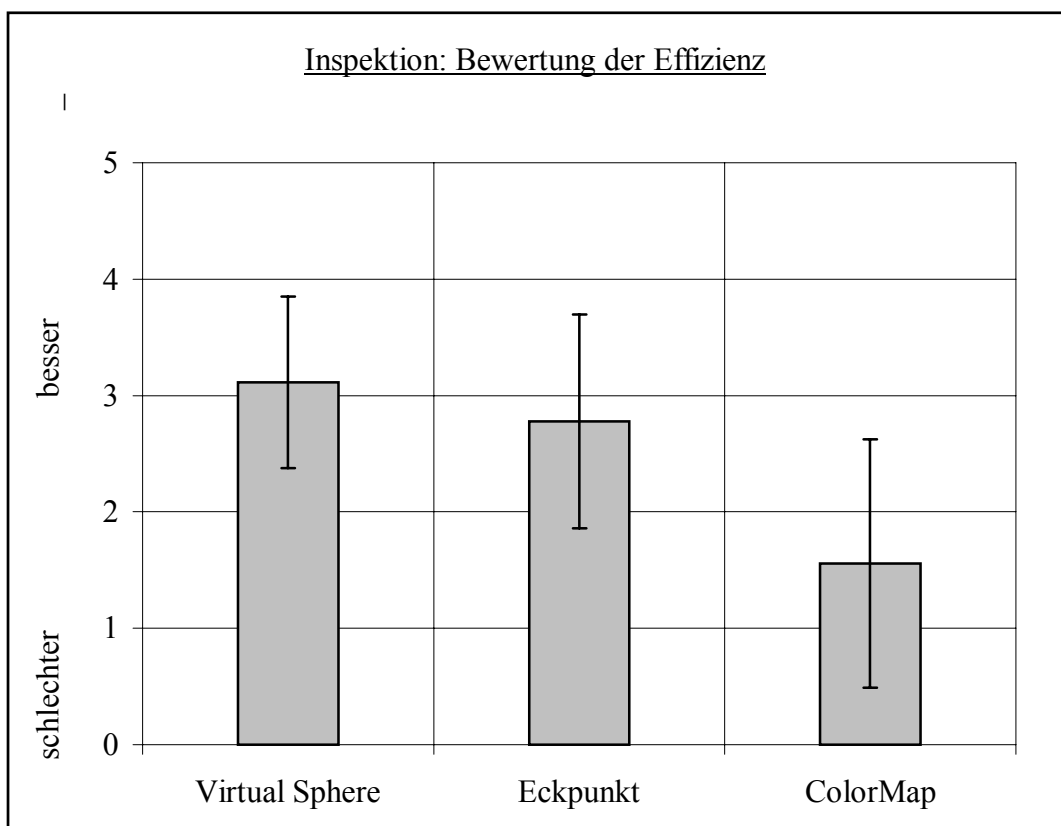


Abbildung 7.2: gemittelter Effizienzwert mit Standardabweichung

### 7.3.3 Inspektion: Erlernbarkeit

<i>Teilnehmer</i>	<i>Virtual Sphere</i>	<i>Eckpunkt</i>	<i>ColorMap</i>
1	4	3	2
2	4	3	2
3	3	1	1
4	4	2	3
5	4	3	0
6	4	4	4
7	3	2	1
8	4	4	3
9	4	4	4
Mittelwert:	3,78	2,89	2,22
Std.-Abw.:	0,42	0,99	1,31

Tabelle 7.3: Bewertung der Erlernbarkeit der Inspektionsmetaphern

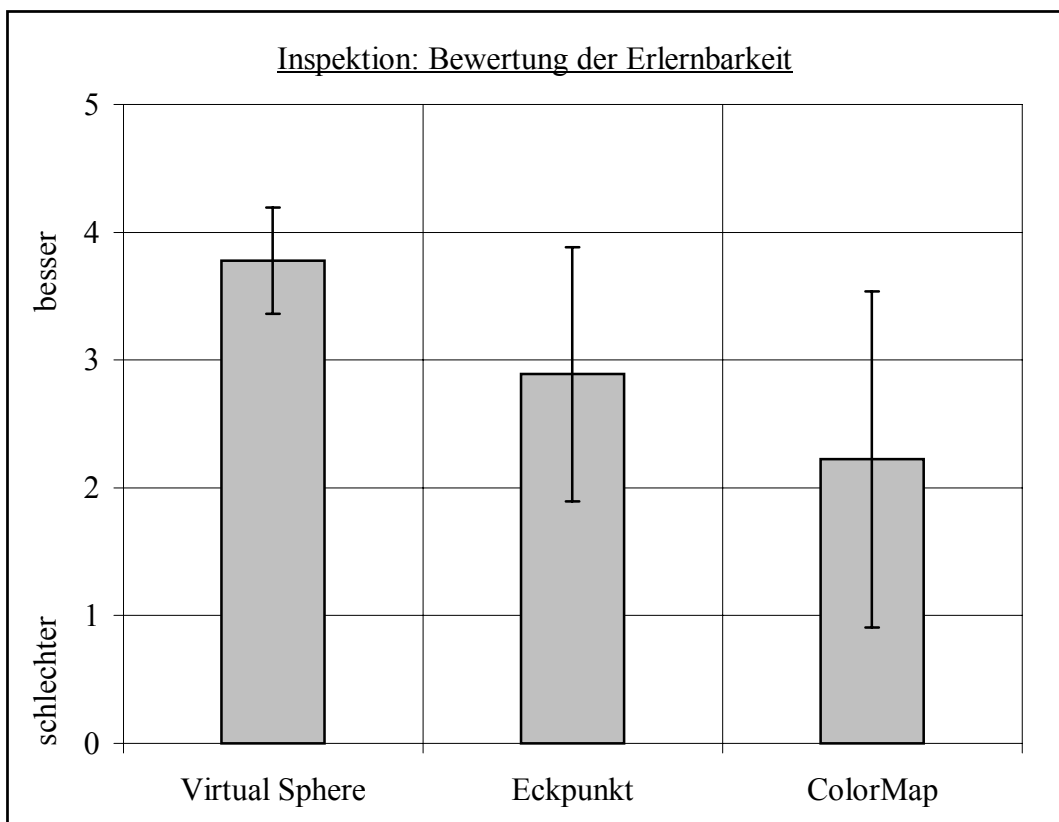


Abbildung 7.3: gemittelter Wert für die Erlernbarkeit mit Standardabweichung

### 7.3.4 Rotation: Intuitivität

<i>Teilnehmer</i>	<i>Rubik's</i>	<i>Zweipunkt</i>	<i>Spotlight</i>	<i>Rahmen</i>
1	4	3	1	2
2	4	3	1	2
3	4	3	3	2
4	4	3	0	3
5	4	2	0	3
6	4	3	3	3
7	3	2	1	2
8	3	3	2	4
9	4	3	2	3
Mittelwert:	3,78	2,78	1,44	2,67
Std.-Abw.:	0,42	0,42	1,07	0,67

Tabelle 7.4: Intuitivitätsbewertung der Rotationsmetaphern

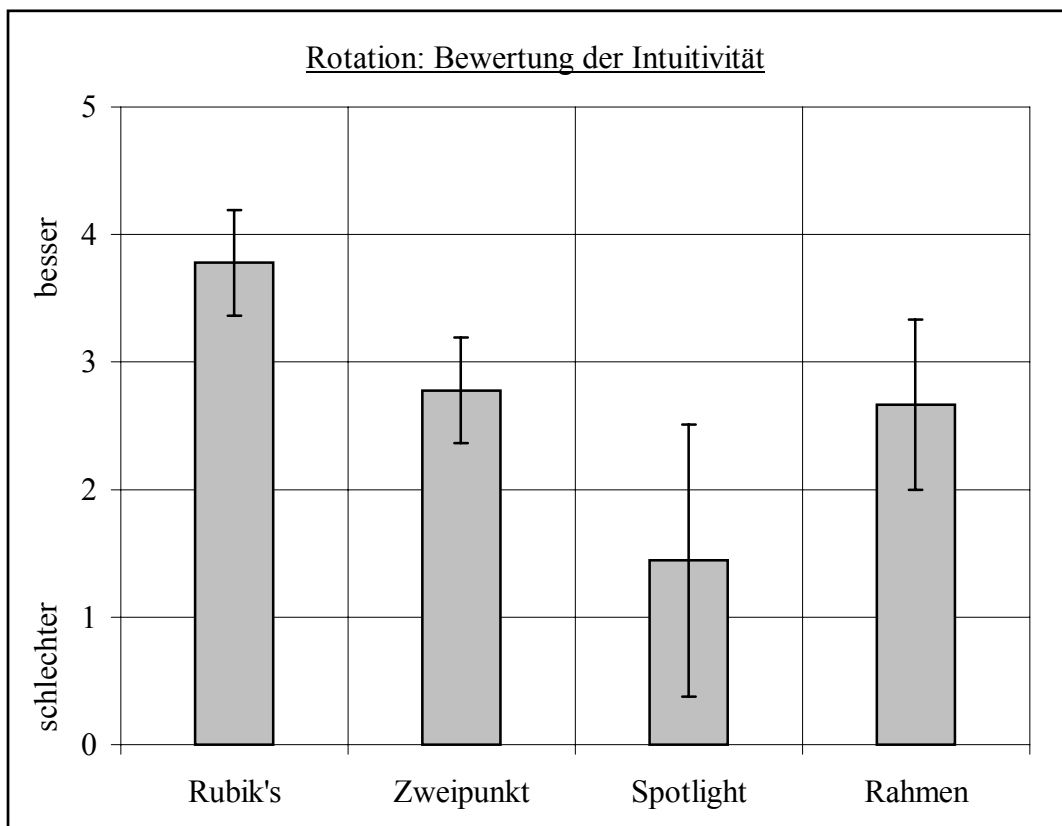


Abbildung 7.4: gemittelter Intuitivitätswert mit Standardabweichung

### 7.3.5 Rotation: Effizienz

<i>Teilnehmer</i>	<i>Rubik's</i>	<i>Zweipunkt</i>	<i>Spotlight</i>	<i>Rahmen</i>
1	4	4	3	2
2	4	3	1	2
3	4	4	2	2
4	4	3	1	2
5	4	2	0	3
6	4	2	1	3
7	3	2	1	2
8	3	2	3	4
9	4	3	2	2
Mittelwert:	3,78	2,78	1,56	2,44
Std.-Abw.:	0,42	0,79	0,96	0,68

Tabelle 7.5: Effizienzbewertung der Rotationsmetaphern

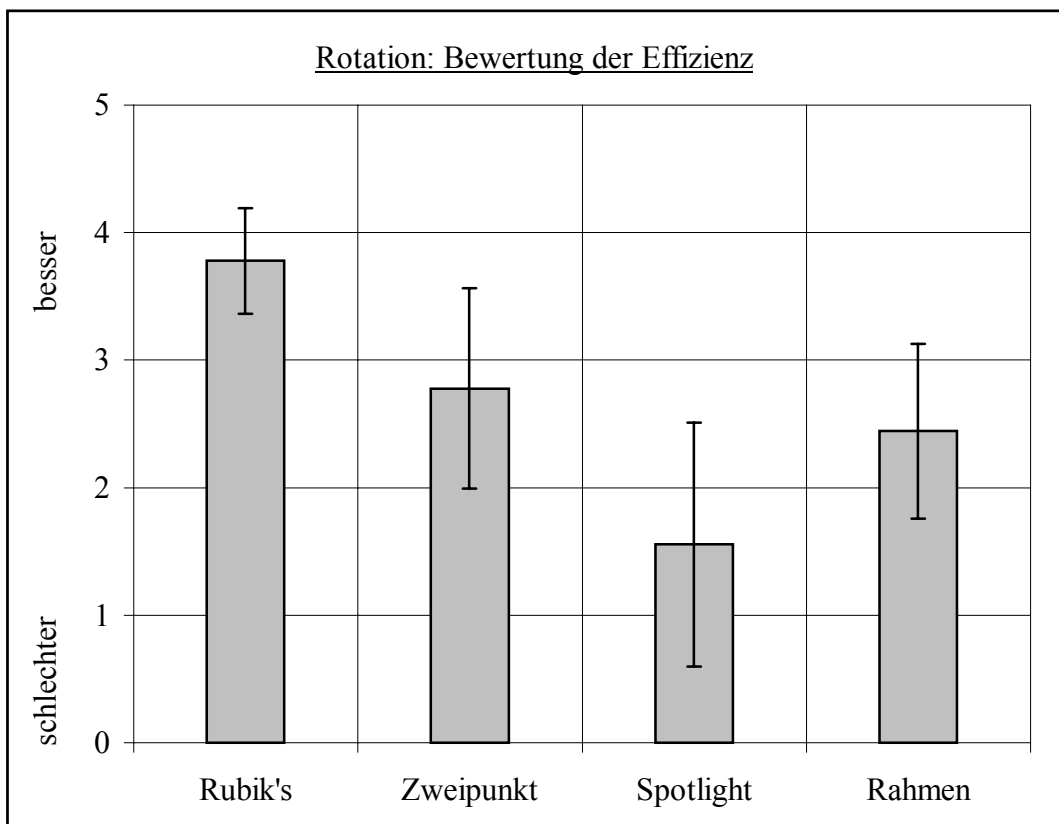


Abbildung 7.5: gemittelter Effizienzwert mit Standardabweichung

### 7.3.6 Rotation: Erlernbarkeit

<i>Teilnehmer</i>	<i>Rubik's</i>	<i>Zweipunkt</i>	<i>Spotlight</i>	<i>Rahmen</i>
1	4	4	1	3
2	3	4	1	2
3	4	4	3	3
4	4	3	0	3
5	4	1	0	2
6	4	2	1	0
7	3	3	1	3
8	3	3	3	3
9	4	3	1	2
Mittelwert:	3,67	3,00	1,22	2,33
Std.-Abw.:	0,47	0,94	1,03	0,94

Tabelle 7.6: Bewertung der Erlernbarkeit der Rotationsmetaphern

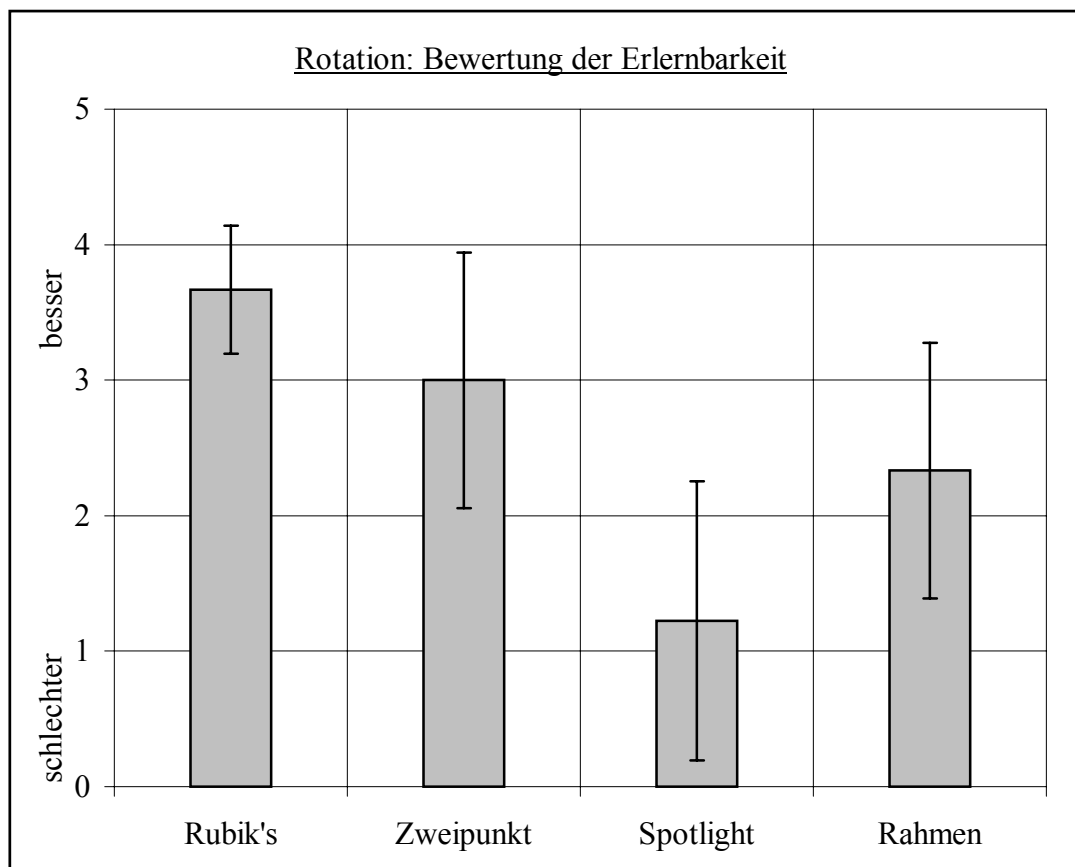


Abbildung 7.6: gemittelter Wert für die Erlernbarkeit mit Standardabweichung

### 7.3.7 Zeitliche Performance

Teilnehmer	Virt. Sphere + Rubik's Online	Eckpunkte + Zweipunkt	ColorMap + Spotlight	ColorMap + Bezugsrahmen
1	30,99	64,62	84,57	85,40
2	65,61	75,09	200,59	119,70
3	59,13	72,99	133,56	54,73
4	52,37	77,60	144,25	73,52
5	42,97	78,68	166,36	84,75
6	37,34	69,59	127,17	71,88
7	49,16	81,00	368,79	63,13
8	42,99	82,63	74,59	58,36
9	69,27	61,89	129,76	68,50
Mittelwert:	49,98	73,79	158,85	75,55
Std.-Abw.:	12,13	6,78	82,46	18,50

Tabelle 7.7: zeitliche Performance der Metapher-Kombinationen

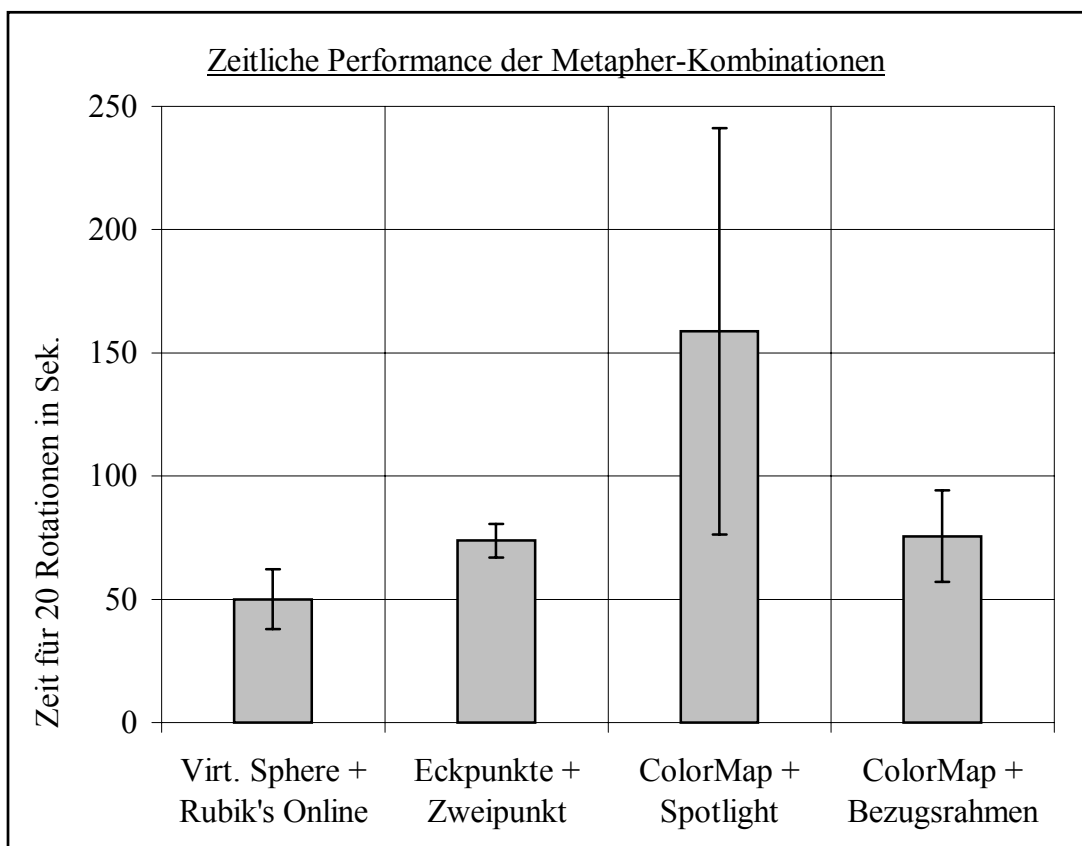


Abbildung 7.7: zeitliche Performance gemittelt mit Standardabweichung

### 7.3.8 Anzahl Inspektionen

<i>Teilnehmer</i>	<i>Virt. Sphere + Rubik's Online</i>	<i>Eckpunkte + Zweipunkt</i>	<i>ColorMap + Spotlight</i>	<i>ColorMap + Bezugsrahmen</i>
1	0	0	5	0
2	4	0	6	1
3	1	3	10	0
4	9	0	4	0
5	0	0	16	0
6	0	0	12	0
7	0	0	12	0
8	8	5	7	0
9	21	0	6	0
Mittelwert:	4,78	0,89	8,67	0,11
Std.-Abw.:	6,65	1,73	3,80	0,31

Tabelle 7.8: Anzahl der Inspektionsoperationen während 20 Rotationen

### 7.3.9 Anzahl Fehler

<i>Teilnehmer</i>	<i>Virt. Sphere + Rubik's Online</i>	<i>Eckpunkte + Zweipunkt</i>	<i>ColorMap + Spotlight</i>	<i>ColorMap + Bezugsrahmen</i>
1	0	0	7	1
2	3	0	4	2
3	2	0	3	1
4	0	0	10	2
5	2	0	5	3
6	1	0	2	5
7	6	0	5	2
8	1	0	3	2
9	3	0	5	0
Mittelwert:	2,00	0,00	4,89	2,00
Std.-Abw.:	1,76	0,00	2,28	1,33

Tabelle 7.9: Anzahl der fehlerhaften Rotationen bei 20 Durchgängen

## 7.4 Messwerte „Lösen einer Würfelseite“

Drei Teilnehmer haben sich auch dem zweiten Teil der Benutzerstudie gestellt und eine Seite des Rubik's Cube sowohl am realen Würfel als auch am virtuellen Modell gelöst.

In diesem zweiten Teil ging es weniger um eine möglichst breite Datengrundlage sondern vielmehr um den individuellen Vergleich zwischen der Performance am realen und am virtuellen Objekt. Auf den folgenden Seiten werden die in diesem Teil gesammelten Messergebnisse daher für jeden Teilnehmer einzeln dargestellt.

Im oberen Diagramm werden die Zeiten gegenübergestellt, die der Teilnehmer für die Lösung einer Würfelseite benötigt hat. Die Zeit für den realen Würfel ergibt sich als Mittelwert der drei wiederholten Lösungsversuche.

In den Diagrammen des unteren Seitenbereichs sind zusätzlich die benötigten Rotationen bis zur Lösung dargestellt. Auch hier ergibt sich der Wert für den realen Würfel aus dem Mittel der drei durchgeführten Wiederholungen.



### 7.4.1 Teilnehmer 2

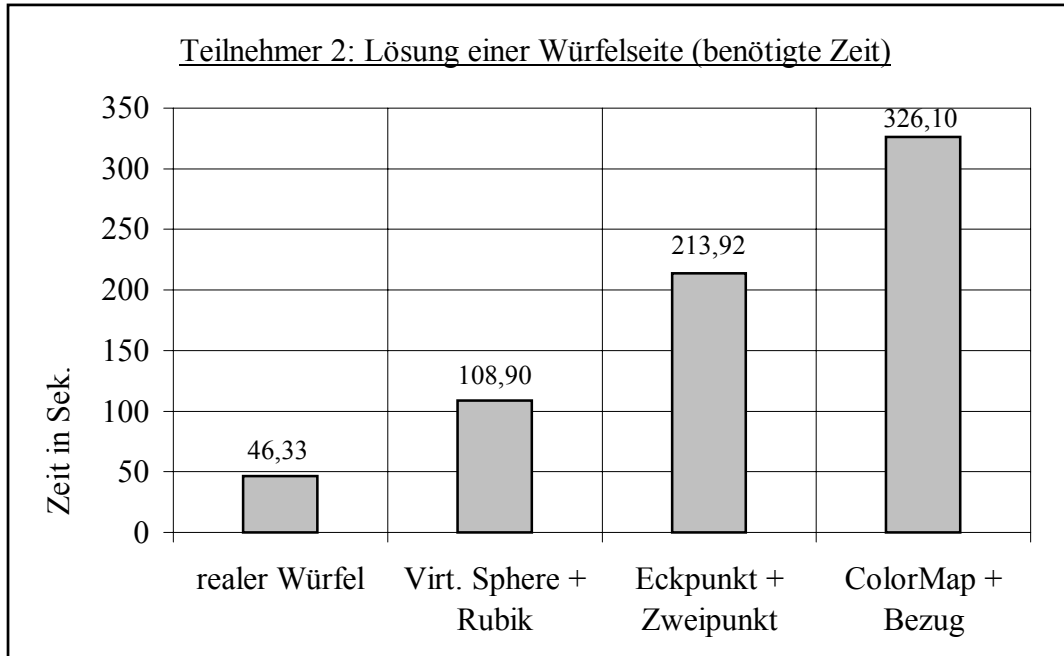


Abbildung 7.8: Zeit für die Lösung einer Seite, Teilnehmer 2

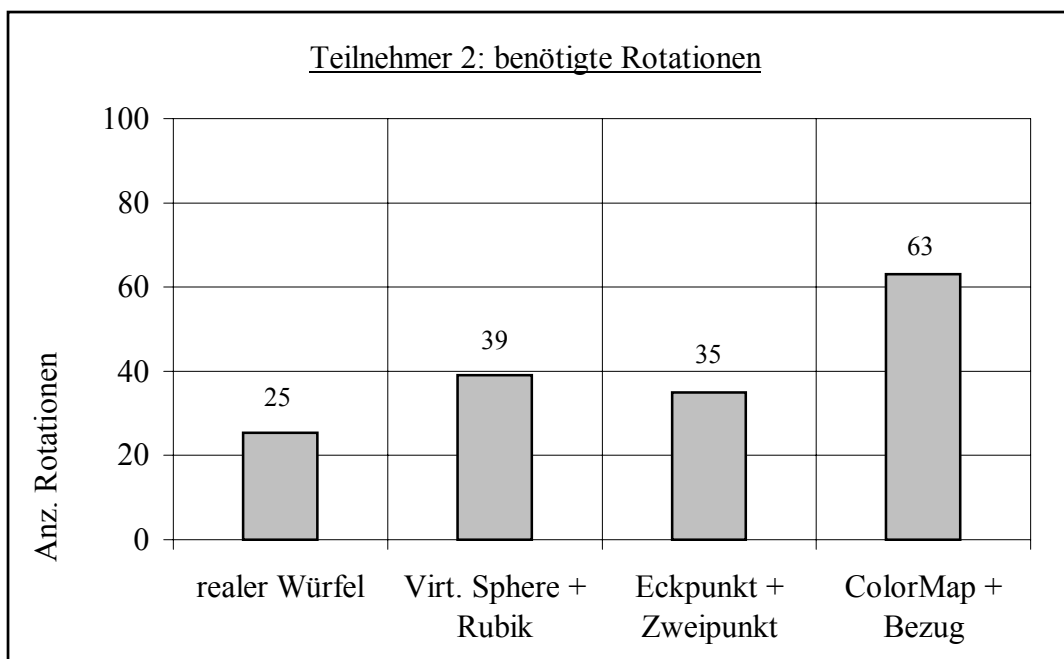


Abbildung 7.9: benötigte Anzahl von Rotationen, Teilnehmer 2

### 7.4.2 Teilnehmer 3

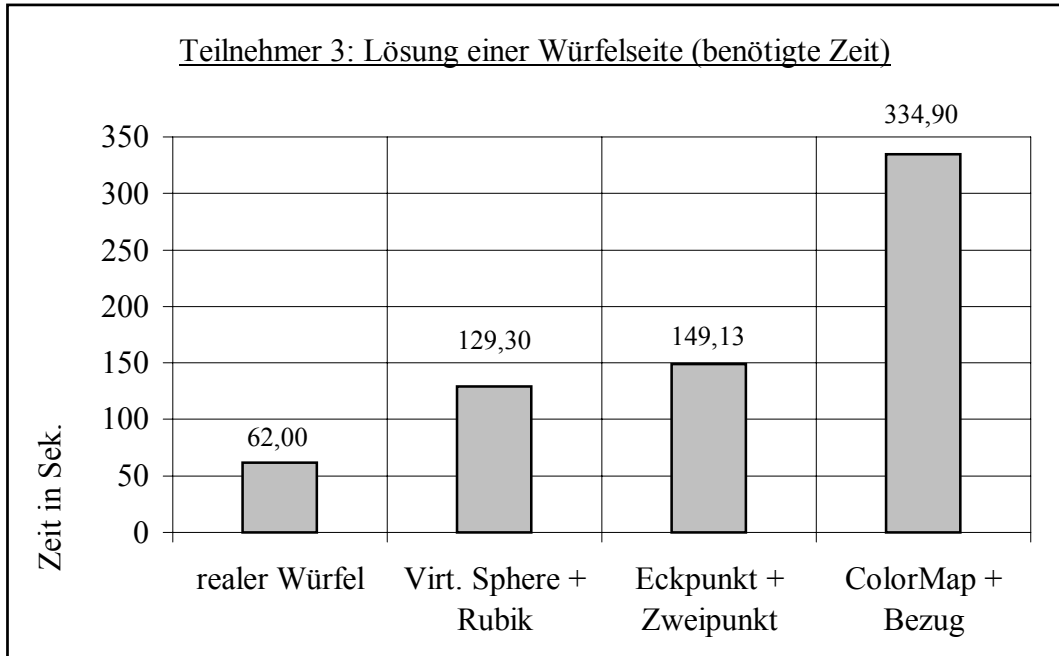


Abbildung 7.10: Zeit für die Lösung einer Seite, Teilnehmer 3

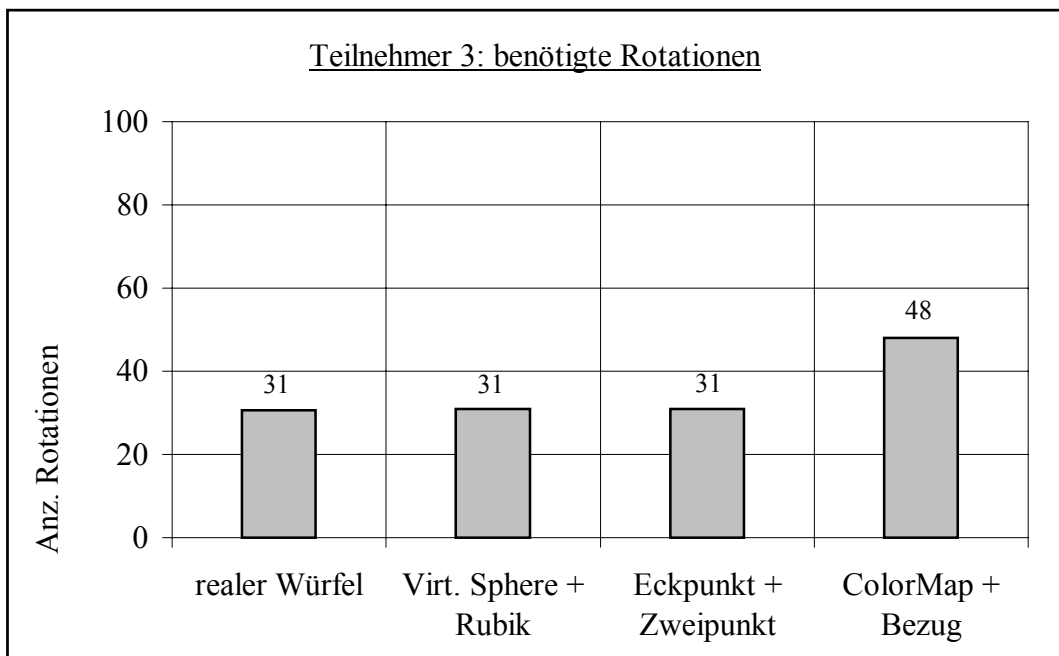


Abbildung 7.11: benötigte Anzahl von Rotationen, Teilnehmer 3

### 7.4.3 Teilnehmer 8

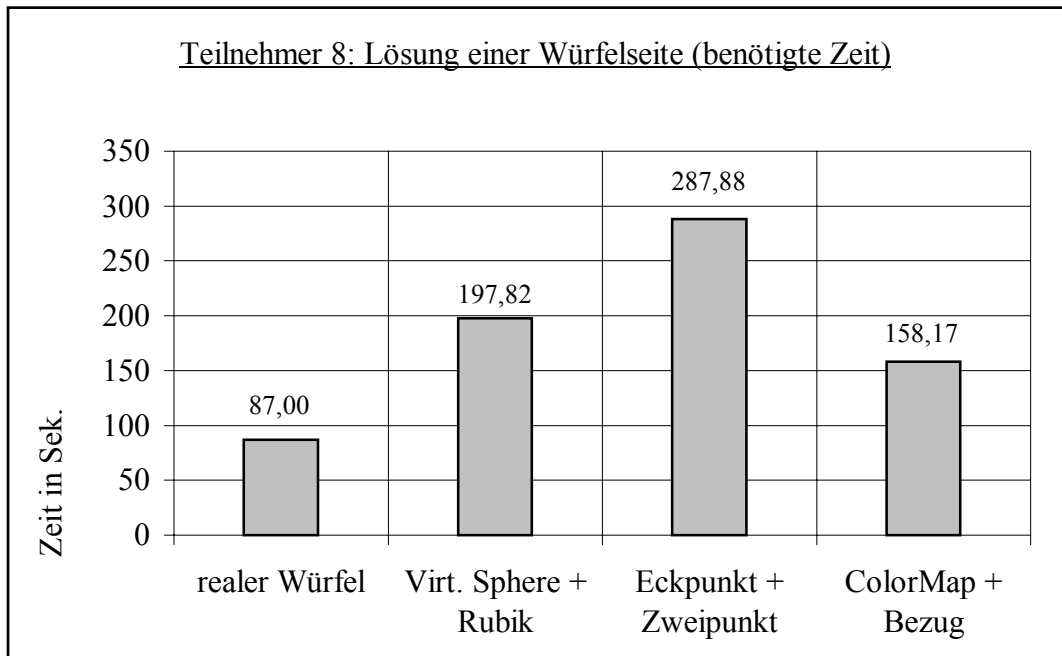


Abbildung 7.12: Zeit für die Lösung einer Seite, Teilnehmer 8

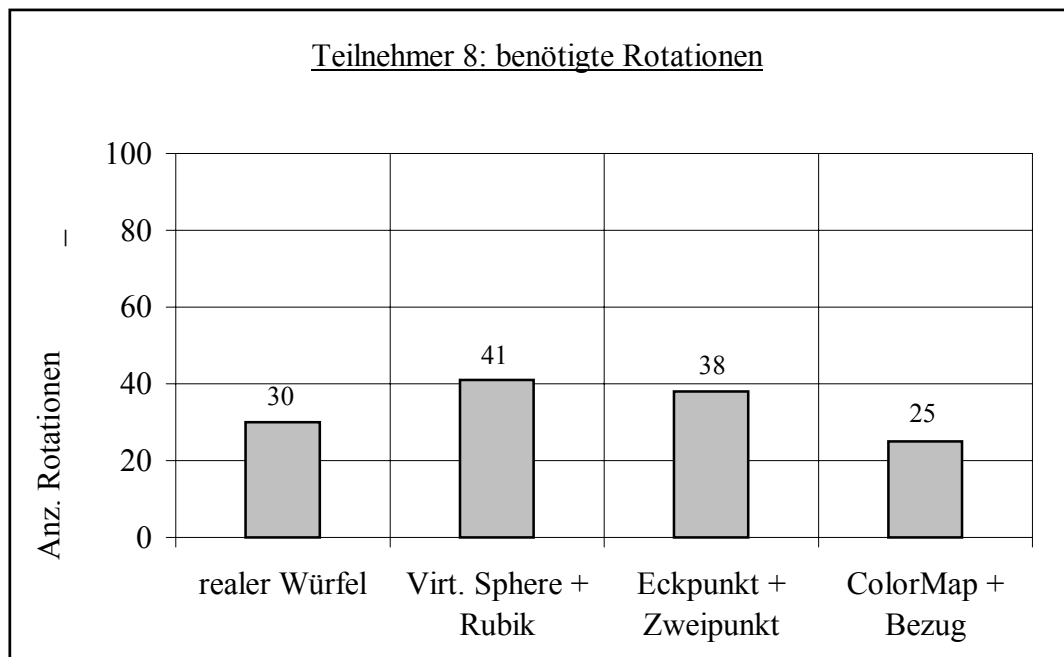


Abbildung 7.13: benötigte Anzahl von Rotationen, Teilnehmer 8

## 7.5 Auswertung

Im folgenden Abschnitt sollen die während der Benutzerstudie gewonnenen Daten ausgewertet und ihre Bedeutung für die einzelnen Interaktionskonzepte diskutiert werden. An dieser Stelle werden zur Beschreibung einzelner Aspekte ebenso die subjektiven Beurteilungen der Versuchsteilnehmer mit einbezogen.

### 7.5.1 Virtual Sphere

Die Bedienung der Virtual Sphere-Metapher wurde von einem Großteil der Teilnehmer als sehr intuitiv empfunden ( $3,67 \pm 0,47$  von maximal 4 Punkten). Die Rotation des Zauberwürfels ist in der Regel ohne großen Konzentrationsaufwand möglich.

Einige Teilnehmer merkten jedoch kritisch an, dass eine gezielte Rotation um die z-Achse nur sehr schwierig oder nur mit ausreichender Übung durchzuführen ist. Da im konkreten Anwendungsfall aber hintereinander ausgeführte Rotationen um die x- und y-Achse in der Regel ausgereicht haben, um zu einer bestimmten Würfelorientierung zu gelangen, wurde diese Einschränkung als nicht besonders hinderlich eingestuft.

Die Möglichkeit der freien Rotation hat bei einigen Teilnehmern dazu geführt, dass der Würfel während der Repeater-Durchgänge zunächst nicht optimal ausgerichtet wurde<sup>59</sup>. Eindrucksvoll lässt sich dieser Effekt in Tabelle 7.8 nachvollziehen: Fünf der Teilnehmer haben während des Repeater-Durchgangs keine oder kaum Inspektionsoperationen ausgeführt, während vier Teilnehmer sich erst über eine Reihe von Würfelrotationen (bis zu 21) einer vorteilhaften Ausrichtung angenähert haben.

Auch bei den Kriterien Effizienz und Erlernbarkeit erhält die Virtual Sphere-Metapher bessere Bewertungen als die übrigen Inspektionsmetaphern, wobei aber darauf hinzuweisen ist, dass mehr als die Hälfte der Teilnehmer ihre Erfahrung mit dieser Metapher vor Start der Benutzerstudie als gut oder sehr gut einge-

---

<sup>59</sup> So wurde die Seite des Würfels, auf der eine Rotation stattgefunden hat, beispielsweise frontal zum Betrachter hin ausgerichtet, was dazu führt, dass folgende Rotationen um andere Achsen nur noch schwer zu beobachten waren. Der Teilnehmer musste also zunächst lernen, dass bestimmte Würfelorientierungen für die durchzuführende Aufgabe besser geeignet waren als andere.

schätzt haben. Durch ihren hohen Bekanntheitsgrad ist die Virtual Sphere-Metapher also gewissermaßen bereits mit einem leichten Vorsprung ins Rennen gegangen.

### 7.5.2 Eckpunkte-Metapher

Die Eckpunkte-Metapher bekommt wie die Virtual Sphere-Metapher bei allen abgefragten Bewertungskriterien überdurchschnittliche Noten – bei der Effizienz-Bewertung liegt sie sogar fast gleichauf mit der Virtual Sphere-Metapher (2,78 zu 3,11 von maximal 4 Punkten).

Dass die Eckpunkte-Metapher gerade auf dem Gebiet einer effizienten Bedienung besonders gut abschneidet, bestätigt die Überlegungen der Konzeptionsphase (siehe Abschnitt 5.3.2): Indem die Anwendung die Berechnung der notwendigen Rotationskomponenten übernimmt und diese nicht durch eine freie Rotation manuell umgesetzt werden müssen, ist ein einfacher Wechsel zwischen verschiedenen Ansichten des Würfels möglich.

Ein weiterer Vorteil der Eckpunkt-Metapher wird deutlich, wenn man die Anzahl der während des Repeater-Durchgangs benötigten Inspektionsrotationen hinzuzieht (im Mittel 0,89): Durch eine jederzeit optimale Ausrichtung des Würfels (gleichzeitige Ansicht von drei Seiten) sind vergleichsweise wenig Neuausrichtungen des Würfels nötig.

Bei der Aufnahme einer ersten subjektiven Einschätzung direkt nach Abschluss eines Repeater-Durchgangs wurde die Interaktion mit dieser Metapher von einzelnen Teilnehmern anfangs als ungewöhnlich und neuartig beschrieben, die Bedienung dann aber als nahezu ebenso einfach wie die Bedienung der Virtual Sphere-Metapher empfunden.

Einzelne Teilnehmer empfanden als Nachteil, dass sich der Rubik's Cube nicht in einem Interaktionsschritt um 180 Grad drehen lässt; hier sind in der Regel zwei Inspektionsoperationen nötig. Außerdem wurde stellenweise kritisch angemerkt, dass die Kontaktpunkte für die Inspektion (Kugeln an den Eckpunkten des Würfels) nicht immer gut zu treffen waren – insbesondere, wenn sie durch die Zentralperspektive der Darstellung zusätzlich kleiner dargestellt wurden. Hier ist eine geeignete Balance zwischen Erreichbarkeit der Interaktionselemente auf der einen und einer größtmöglichen Darstellungsfläche für die eigentlichen Objekte der Szene zu erreichen.

### 7.5.3 ColorMap-Metapher

Von allen drei Inspektionsmetaphern bekam die ColorMap-Metapher bei den Kriterien Intuitivität, Effizienz und Erlernbarkeit die schlechtesten Noten von den Teilnehmern.

Ein Grund für dieses schlechte Abschneiden wurde besonders häufig genannt: Um eine bestimmte Seite des Würfels nach vorn zu rotieren, muss zunächst die Verteilung der Farbflächen am Zauberwürfel analysiert werden und anschließend ist die Farbe der gewünschten Seite in der ColorMap zu lokalisieren. Zusätzlich erschwert wird dieser Vorgang vor allem dann, wenn die Würfelseite, die nach vorn gedreht werden soll, in der aktuellen Ansicht nicht zu sehen ist und die zugehörige Farbe also nur über die Anordnung der ColorMap ermittelt werden kann.

Interessanterweise wurde die Idee der ColorMap von einzelnen Teilnehmern bei der Vorstellung noch begeistert aufgenommen, da man bei Inspektionsoperationen eine beliebige Farbseite mit einem Mausklick nach vorne ausrichten kann (im Gegensatz zur Eckpunkte- und Virtual Sphere-Metapher also ein semantischer Bezug zum Interaktionsobjekt besteht). Dieser Vorteil scheint durch die genannten Nachteile jedoch aufgehoben zu werden.

An dieser Stelle wünschten sich viele Teilnehmer die Möglichkeit, einzelne Inspektionsmetaphern miteinander kombinieren zu können: So würde beispielsweise eine Kombination der Virtual Sphere- mit der ColorMap-Metapher weiterhin die freie Rotation des Würfels ermöglichen und gleichzeitig könnte die ColorMap-Metapher ihre Vorteile voll ausspielen, sobald gezielt zu einer bestimmten Farbseite navigiert werden soll.

Weiterhin sei darauf hingewiesen, dass ein Grund für das schlechte Abschneiden der ColorMap-Metapher gegebenenfalls auch in der besonderen Situation beim Rubik's Cube zu sehen ist. Der Zauberwürfel besitzt ja gerade keine ausgewiesenen Seiten, einzelne Flächen lassen sich nur durch die Farbe ihres Mittelwürfels identifizieren. Aber die Informationen über die räumliche Verteilung dieser sechs Flächen zueinander übersteigt offenbar die Kapazität des menschlichen Kurzzeitgedächtnisses und muss sich bei jeder Inspektionsoperation buchstäblich vor Augen geführt werden.

In einem Anwendungszusammenhang, bei dem das zu steuernde Objekt über ausgezeichnete Ansichten verfügt<sup>60</sup>, scheint das der ColorMap zugrunde liegende Konzept der Kartennavigation daher eher angemessen zu sein.

#### 7.5.4 Rubik's Online-Metapher

An dieser Stelle folgt die Auswertung der einzelnen Rotationsmetaphern und wie bei der Inspektion wird auch hier mit der Referenzimplementierung begonnen:

Ähnlich wie die Virtual Sphere-Metapher schneidet auch die Rubik's Online-Metapher für die Rotation von Teilscheiben des Zauberwürfels in den Disziplinen Intuitivität, Effizienz und Erlernbarkeit am besten ab, nur bei der aufgetretenen Fehleranzahl muss sie sich der Eckpunkte-Metapher geschlagen geben<sup>61</sup>. An diesem Punkt zeigt sich, dass die Differenzierung zwischen einzelnen Seiten eines Teilwürfels nicht unkritisch ist und gerade bei einer schnellen Interaktionsabfolge oftmals unterschätzt wurde.

Ein weitere Beobachtung, die man während der Versuchsdurchführung machen konnte, betrifft die Rotation einer Drehscheibe, deren Seitenflächen diagonal zur Projektionsfläche ausgerichtet waren: Für die Rotation einer solchen Drehscheibe musste die Maus *diagonal* mit einem bestimmten Winkel in die gewünschte Rotationsrichtung bewegt werden; eine Bewegung, die von einzelnen Teilnehmern als relativ schwierig eingeschätzt wurde<sup>62</sup>.

Außerdem konnte bei einigen Teilnehmern beobachtet werden, dass die Mausbewegung zur Initiierung des Rotationsverhaltens schnell und mit einer hohen Auslenkung durchgeführt wurde – was häufig zu Ungenauigkeiten führte oder dazu, dass die Rotation von der Anwendung nicht mehr eindeutig erkannt werden konnte und in der Folge keine Aktion ausgeführt wurde.

---

<sup>60</sup> beispielsweise Vorder- und Rückseite, linke und rechte Seite, oben und unten

<sup>61</sup> Die Vergleichbarkeit der Ergebnisse mag kein Zufall sein – beruht doch auch die Rubik's Online-Metapher im Prinzip auf dem Konzept der virtuellen Sphäre.

<sup>62</sup> Als eine Alternative zu dieser Diagonalebewegung hätte es sich angeboten, den Würfel vor der Rotation geeignet auszurichten, sodass wiederum nur eine Mausbewegung entlang einer Raumachse nötig gewesen wäre, um eine Drehscheibe zu rotieren; eine Möglichkeit, die von den meisten Teilnehmern aber nicht wahrgenommen wurde.

Diesem Verhalten scheint eine sehr starke Intuition zugrunde zu liegen, denn auch nach entsprechenden Hinweisen fiel es den Teilnehmern schwer, sich auf die erforderliche Interaktionsform einzustellen. Scheinbar wirkt sich hier die Erfahrung am realen Würfel sehr stark aus, bei dem eine Rotation auch am besten mit entsprechendem „Schwung“ durchgeführt werden kann.

### 7.5.5 Zweipunkt-Metapher

Nach der Rubik's Online-Metapher zeichnet sich die Zweipunkt-Metapher durch die zweitbeste zeitliche Performance aus (siehe Abbildung 7.7). Trotz der höheren Anzahl von Interaktionsschritten konnten die vorgegebenen Rotationen recht zügig reproduziert werden.

Hervorgehoben werden sollte vor allem die geringe Fehleranfälligkeit dieser Rotationsmetapher: Hier schneidet die Zweipunkt-Rotation unter allen untersuchten Metaphern am besten ab (0 Fehler bei allen Teilnehmern, siehe Tabelle 7.9).

Ein Grund für dieses gute Abschneiden ist sicherlich im Punkt der einzelnen, aufeinanderfolgenden Interaktionsschritte zu sehen, der von einzelnen Teilnehmern zunächst kritisch gesehen wurde (Wahl des Startwürfels, Wahl des Zielwürfels, Doppelklick): Erfolgt die Rotation einer Teilscheibe bei der Rubik's Online-Metapher bereits nach einer kurzen Dragging-Bewegung der Maus, so müssen bei der Zweipunkt-Metapher die genannten drei Interaktionsschritte kontrolliert hintereinander ausgeführt werden. Sobald der Anwender einen Fehler bemerkt, kann der Interaktionszyklus mit Hilfe der rechten Maustaste auf den Anfangszustand zurückgesetzt werden.

Positiv angemerkt wurde von den meisten Teilnehmern zudem die gewählte Form der Visualisierung der selektierten Teilwürfel. Auf diese Weise waren sich die Probanden zu jedem Zeitpunkt über den aktuellen Interaktionsmodus im Klaren.

Eine häufig gemachte Beobachtung während der Studie betrifft die Wahl des Startwürfels: Viele Teilnehmer klickten hierbei zunächst auf einen *Eckwürfel* und realisierten erst, wenn dieser nicht markiert wurde, dass sie zur Definition der Rotation einen *Kantenwürfel* auswählen müssen.

Ähnlich wie bei der Rubik's Online-Metapher scheinen sich hier die physikalischen Bedingungen der wirklichen Welt sehr deutlich auszuwirken: Um eine Drehscheibe am realen Objekt zu bewegen, wird man in der Regel an dem Punkt



mit der besten Kraftübertragung angreifen und damit an einer Außenseite beziehungsweise an einem Eckwürfel. Dieses implizite Verständnis der realen Welt wird unbewusst auf das virtuelle Modell übertragen und steht einer unvoreingenommenen Bedienung der angebotenen Metapher im Wege<sup>63</sup>.

### 7.5.6 Spotlight-Metapher

Die Spotlight-Metapher bildet das Schlusslicht der untersuchten Rotationsmetaphern. Dieses negative Abschneiden lässt sich auf verschiedene Eigenschaften dieses Interaktionswerkzeugs zurückführen, die zum Teil mit der besonderen Struktur des Rubik's Cube begründet werden können und sich in einem anderen Anwendungszusammenhang gegebenenfalls nicht derart gravierend auf die Performance auswirken würden.

Eine prinzipielle Kritik der Teilnehmer bezieht sich auf die Schwierigkeit, einzelne Interaktionsmodi zu unterscheiden. Das metaphorische Konzept des selektierenden Spots wurde in der Regel noch gut verstanden und relativ sicher angewendet. Der aus der funktionalen Unterlegung des frontalen Mittelwürfels resultierende zusätzliche Interaktionsmodus zur Rotation der beiden Mittelscheiben konnte jedoch nur schwer in dieses metaphorische Verständnis eingebettet werden. Der Unterschied zwischen einer entfernten Manipulation auf der einen und einer Interaktion direkt am Objekt auf der anderen Seite wird als nicht intuitiv empfunden. Die zusätzliche modale Unterscheidung über die Steuerungstaste zur Differenzierung zwischen horizontaler und vertikaler Rotation erschwert die Bedienung zusätzlich.

Ein weiterer Kritikpunkt der Teilnehmer betraf den Umstand, dass bestimmte Rotationsoperationen in der jeweiligen Würfelausrichtung prinzipiell nicht durchführbar waren<sup>64</sup>. Auch bei den übrigen Metapher-Kombinationen treten Fälle auf, in denen eine Neuausrichtung des Würfels vorteilhaft erscheint, jedoch wäre die gewünschte Rotation oftmals auch in der ursprünglichen Orientierung – wenn auch schwieriger – prinzipiell möglich gewesen.

---

<sup>63</sup> Genau dieses „Angreifen an den Außenseiten“ wird von der Bezugsrahmen-Metapher umgesetzt.

<sup>64</sup> beispielsweise die vertikale Rotation der von der Vorderscheibe verdeckten mittleren Drehscheibe

Auch eine *konzeptionelle Schwäche* der Spotlight-Metapher ist im Rahmen der Benutzerstudie deutlich geworden: Die zum Benutzer ausgerichtete Vorderseite muss, um rotiert werden zu können, zunächst vom Anwender *weggedreht* werden. Ein Umstand, der sowohl Intuitivitäts- als auch Effizienzkriterien entgegensteht<sup>65</sup>.

Bei der Analyse der Log-Dateien für diesen Versuchsteil fällt ein Punkt besonders auf: Bei den Zeiten für eine einzelne Rotation treten eklatante Unterschiede auf. Während für einige Rotationen Zeiten im Bereich von circa 1 Sekunde gemessen wurden, dauerten andere Rotationsoperationen unverhältnismäßig länger (10 Sekunden und mehr).

Hier wird noch einmal der Unterschied zwischen den unterschiedlichen Interaktionsmodi anschaulich: Eine Rotation durch Anstrahlen einer Würfelseite kann – sofern der Würfel entsprechend ausgerichtet ist – sehr schnell und ohne besonderen Konzentrationsaufwand umgesetzt werden. Soll jedoch eine Würfelscheibe rotiert werden, die nicht auf diese einfache Art ausgewählt werden kann, benötigen die Teilnehmer deutlich mehr Zeit und es treten unverhältnismäßig viele Fehler auf.

Wie oben bereits erwähnt wurde, scheint die Spotlight-Metapher daher für die Vielzahl der unterschiedlichen Rotationsvarianten am Rubik's Cube nicht geeignet. In einem der bei der Konzeption in Abschnitt 5.4.3 beschriebenen Anwendungsfälle sollten die prinzipiellen Vorteile dieser Metapher jedoch deutlicher zum Tragen kommen.

### 7.5.7 Bezugsrahmen-Metapher

Als besonderes Merkmal der Bezugsrahmen-Metapher ist während der Studie aufgefallen, dass bei der Anwendung dieses Rotationstools kaum Inspektionsoperationen durchgeführt worden sind<sup>66</sup>. Da mindestens zwei Fingerpunkte des Bezugsrahmens bei der jeweiligen Würfelausrichtung zu sehen waren, waren Rotationen um beliebige Achsen bereits in der Grundausrichtung durchführbar.

---

<sup>65</sup> Insbesondere der Intuition, dass die Interaktion mit naheliegenden Objekten einfacher möglich ist als die mit entfernten; in diesem Punkt ist es interessant anzumerken, dass sich dieselbe Kritik auch auf die Rotation der *Rückseite* des Rubik's Cube anwenden lässt, dies von den Teilnehmern aber nicht in ähnlicher Form als besonders nachteilig angesehen wurde.

<sup>66</sup> Acht von neun Teilnehmern kamen ganz ohne Inspektion aus, ein Teilnehmer hat die angebotene Inspektionsmetapher einmal benutzt (siehe Tabelle 7.8).

Gerade bei dieser tastaturgesteuerten Metapher sind individuelle Unterschiede besonders deutlich aufgefallen: Teilnehmer, die auch sonst eine Tastatursteuerung von Anwendungen der mausgestützten Interaktion vorziehen, waren hier klar im Vorteil<sup>67</sup>.

Von einigen Teilnehmern wurden Rotationen um die senkrecht zur Projektionsebene liegende Drehachse als schwierig empfunden – in diesen Fällen muss der Bezugsrahmen entlang der schräg angeschnittenen Würfelseite bewegt werden. Hier wird die bereits in der Konzeption angedeutete Problematik der eindeutigen Richtungsabbildung deutlich: Unter Umständen ist nicht sofort ersichtlich, welche Pfeiltasten die *Rotation* und welche die *Ausrichtung* des Bezugsrahmens veranlassen. In einem solchen Fall würde eine Neuausrichtung des Würfels Abhilfe schaffen.

### 7.5.8 Lösen einer Würfelseite

Zunächst ist festzustellen, dass alle Teilnehmer bei der Lösung der virtuellen Würfelseite mehr als doppelt so viel Zeit benötigen wie am realen Rubik's Cube. Selbst der während der Repeater-Durchgänge bravourös abgeschnittenen Kombination Virtual Sphere und Rubik's Online gelingt es in diesem zweiten Durchgang nicht „die Realität zu schlagen“.

Weiterhin fällt auf, dass sich die Anzahl der benötigten Rotationen – bis auf einzelne Ausreißer – nicht wesentlich von der am realen Objekt benötigten Anzahl unterscheidet. Da die Zeit für das Durchführen einer Rotation auch beim virtuellen Modell im einstelligen Sekundenbereich liegt, lässt sich der zeitliche Blow-Up beim virtuellen Modell insbesondere durch zeitintensive Inspektionsoperationen erklären: Beim realen Rubik's Cube ist der Würfel wörtlich im „Handumdrehen“ neu ausgerichtet und auf Grund des kinästhetischen Sinns weiß der Spieler intuitiv, wo sich die ehemalige Vorderseite nach einer Verdrehung befindet.

---

<sup>67</sup> So hat der schnellste Teilnehmer für den Repeater-Durchgang nur die Hälfte der Zeit des langsamsten Teilnehmers benötigt. Interessanterweise haben diese beiden Teilnehmer auch am zweiten Teil der Studie teilgenommen und dort wurden diese Ergebnisse noch einmal bestätigt (siehe Abschnitt 7.4).

Im Gegensatz zu den Repeater-Durchgängen fällt bei diesem zweiten Versuchsteil auf, dass die Performance-Unterschiede der einzelnen Metaphern insbesondere bei zwei der Teilnehmer bei weitem nicht mehr so stark ausfallen.

So zeigt sich bei Teilnehmer 8 beispielsweise die ColorMap- und Bezugsrahmen-Kombination in der zeitlichen Performance sogar der Referenzimplementierung überlegen und bei Teilnehmer 3 liegt die Eckpunkt/Zweipunkt-Kombination mit der virtuellen Sphäre nahezu gleich auf (siehe Abbildung 7.10 und Abbildung 7.12). In diesem Punkt zeigt sich sehr deutlich, wie stark sich individuelle Vorlieben (beispielsweise eine bevorzugte und daher gut eingeübte Tastatursteuerung) auf die Bedienung der einzelnen Metaphern auswirken können.

Ein Aspekt, der gerade bei der ColorMap-Navigation erst während dieses zweiten Versuchsteils festgestellt wurde, betrifft die Umkehrbarkeit einer Inspektionshandlung: Bei dieser Metapher ist das Wiederherstellen der Ausgangsorientierung nach einer Inspektionshandlung in der Regel ebenso aufwändig wie die ursprüngliche Inspektionsinteraktion (jeweils Identifizieren der gewünschten Farbseite und Wahl auf der ColorMap). In diesem Punkt scheint vor allem die Eckpunkt-Metapher den anderen Verfahren überlegen, denn hier genügt ein Klick auf die ursprünglich nach vorn gerichtete Würfecke, um die alte Orientierung des Würfels wiederherzustellen.

## 8. Zusammenfassung und Ausblick

Das folgende Kapitel fasst den Inhalt dieser Arbeit zusammen und gibt einen Ausblick auf mögliche Erweiterungen, die auf Grund der erhaltenen Ergebnisse sinnvoll erscheinen.

### 8.1 Zusammenfassung

Das Ziel der vorliegenden Arbeit war es, neue *Navigations- und Interaktionstechniken* für räumliche Anwendungen zu entwerfen und anhand einer *prototypischen Implementierung* die Eignung dieser Techniken für die Interaktion mit einem virtuellen Modell des Rubik's Cube zu untersuchen.

Da die entwickelten Verfahren ihre Tauglichkeit insbesondere bei der Interaktion über klassische Ein- und Ausgabegeräte unter Beweis stellen sollten (Maus, Tastatur und 2D-Display), waren geeignete Abbildungen der zu beherrschenden Freiheitsgrade zu konzipieren.

Die Beschreibung grundlegender Aspekte der menschlichen Wahrnehmung führte zum Konzept der *3D-Metapher*, welche die Durchführung einer dreidimensionalen Operation mit Hilfe von 2D-Eingabegeräten erklärt. Wichtige Prinzipien und Designkriterien beim Entwurf von 3D-Metaphern wurden erläutert.

Im Anschluss wurden einzelne Interaktionsaufgaben des 3D-Raums dargestellt und Beispiele für metaphorische Konzepte bei der Implementierung der beschriebenen Operationen gegeben.

Nach der Darstellung der am Rubik's Cube auftretenden Interaktionsformen wurde untersucht, welche Eigenschaften den Zauberwürfel als ein geeignetes Versuchsobjekt qualifizieren. Nachfolgend wurden für die Interaktionsaufgaben *Inspektion* und *Rotation* unterschiedliche metaphorische Konzepte entworfen und ihre besonderen Eigenschaften beschrieben; hierbei wurde ebenfalls auf grundlegende Verfahren eingegangen, die bei der Implementierung der Metaphern eingesetzt worden sind.

Im Rahmen einer *Benutzerstudie* wurde die Bedienung der konzipierten Interaktionsmetaphern im praktischen Einsatz getestet. Hierbei wurden insbesondere die Kriterien *Intuitivität*, *Effizienz* und *Erlernbarkeit* untersucht sowie die zeitliche Performance und Fehlerhäufigkeiten beim Einsatz der unterschiedlichen Werkzeuge analysiert.

## 8.2 Ausblick und Erweiterungsmöglichkeiten

Die vorliegende Arbeit bietet eine Reihe von Ansätzen für künftige Erweiterungen; im Folgenden werden ausgewählte Aspekte beschrieben.

### *Entwicklung einer Autorenumgebung für interaktive Anwendungen*

Das bei der Implementierung der Interaktionsmetaphern eingesetzte Behavior-Konzept der Java3D-API erlaubt eine weitgehende Kapselung der Programmlogik, über die das Verhalten der jeweiligen Interaktionskomponente gesteuert wird.

Vor diesem Hintergrund bietet es sich an, auf Grundlage der hier entwickelten Konzepte eine *Autorenumgebung* zu entwerfen, mit deren Hilfe interaktive Anwendungen auch ohne Programmierkenntnisse erstellt werden können.

Im Rahmen einer solchen Umgebung wäre es dann auf einfache Weise möglich, einzelnen Objekten einer bestehenden 3D-Szene bestimmte Interaktionsmerkmale hinzuzufügen und verschiedene Attribute einer Metapher zu konfigurieren, um so passend eingestellte Werkzeuge für spezifische Aufgaben zu erhalten.

Bei dem Entwurf dieser Autorenumgebung wäre ebenfalls zu prüfen, inwieweit die Erstellung *neuer Interaktionsmetaphern* – gegebenenfalls auf Grundlage von atomaren Interaktionselementen – sinnvoll und wünschenswert erscheint.

### *Konzeption eines Kommunikationskanals zwischen den Interaktionsmetaphern*

Wie in Abschnitt 6.3 beschrieben wurde, lassen sich die einzelnen Interaktionsmetaphern aus Implementierungssicht beliebig miteinander kombinieren. Ein vielversprechender Ansatz für eine Erweiterung der hier vorgestellten Konzepte könnte in der Entwicklung eines *Kommunikationskanals* zwischen den einzelnen Interaktionswerkzeugen bestehen; auf diese Weise könnten kontextspezifische Daten ausgetauscht und mit einem angepassten Verhalten reagiert werden.

So stehen auch in der Realität einzelne Interaktionsaufgaben meist nicht isoliert nebeneinander; vielmehr impliziert eine erste durchgeführte Aktion oftmals eine Reihe von möglichen Folgeaktionen.

So stehen beispielsweise beim realen Zauberwürfel Inspektions- und Rotationsaufgaben in einem engen Zusammenhang. Indem der Würfel auf eine bestimmte Weise in der Hand gehalten wird – und sich dadurch also eine ausgewählte Inspektionsansicht definiert – können bereits Aussagen über eine möglicherweise folgende Rotationsoperation gemacht werden. Diese wird in der Regel an der Würfel­seite initiiert, von der maximal viele Seitenflächen sichtbar sind, da der Spieler auf diese Weise die Auswirkung des jeweiligen Zuges optimal abschätzen kann.

Solche kontextabhängigen Informationen, bei denen eine spezifische Konfiguration der einen Komponente auf naheliegende Aktionen einer anderen verweist, können über einen geeignet konstruierten Kommunikationskanal zwischen den Interaktionskomponenten ausgetauscht werden. Auf diese Weise wird der Anwender von Aufgaben entbunden, die durch eine Interaktion der Komponenten untereinander selbständig ausgeführt werden können.

Im speziellen Fall des Zauberwürfels könnte beispielsweise die Ausrichtung der Fingerpunkte bei der Bezugsrahmen-Metapher auf Grund der jeweiligen Würfelorientierung bestimmt werden. Bei einer folgenden Drehscheiben-Rotation würde sich der Bezugsrahmen dann mit einer guten Wahrscheinlichkeit bereits an der richtigen Position befinden, die mittlere Anzahl der benötigten Interaktionsschritte könnte so reduziert werden.

### *Transfer auf immersive VR-Anwendungen*

Die in dieser Arbeit entworfenen Interaktionsmetaphern wurden in Hinblick eines Einsatzes auf dem Gebiet der Desktop-VR konzipiert. Prinzipiell lassen sich die hier vorgestellten Verfahren aber auch in einer immersiven Umgebung einsetzen.

Bei einem solchen Transfer wäre insbesondere zu untersuchen, ob die Beschränkung der gleichzeitig kontrollierten Freiheitsgrade von den Anwendern als Einschränkung empfunden wird oder im Gegenteil sogar mit zu einer effektiven Interaktion beitragen kann. An dieser Stelle sei auf die Erläuterungen in Abschnitt 4.2.3 verwiesen, die beschreiben, dass auch bei Interaktionen in der realen Welt

ganz intuitiv geeignete Constraints gesucht werden, um so die Anzahl der gleichzeitig kontrollierten Freiheitsgrade zu beschränken.



## **Anhang A: Glossar**

### **3D-Eingabegeräte**

Eingabegeräte wie Datenhandschuh oder Space-Mouse, mit deren Hilfe die Beschränkung auf zwei Freiheitsgrade aufgehoben wird, wie sie bei der Interaktion über eine klassische Computermaus besteht.

### **3D-Metapher**

Die 3D-Metapher kann dazu eingesetzt werden, dem Anwender ein → mentales Modell für die Durchführung einer Interaktionshandlung zu vermitteln. Auf diese Weise können beispielsweise intuitive Schnittstellen auf dem Gebiet der → Desktop-VR entworfen werden.

### **Desktop-VR**

Spezialfall der virtuellen Realität, bei der die Interaktion mit der räumlichen Anwendung über herkömmliche Ein- und Ausgabegeräte erfolgt, wie sie bei Desktop-Systemen üblich sind (Maus, Tastatur, Bildschirm).

### **kognitive Last**

Auf dem Gebiet der Mensch-Computer-Interaktion bezeichnet der Begriff kognitive Last die Beanspruchung des Anwenders bei der Durchführung einer bestimmten Interaktionsaufgabe. Je intuitiver ein Interaktionswerkzeug gestaltet wurde, desto geringer ist die kognitive Last bei seiner Benutzung.

### **mentales Modell**

Mentale Modelle repräsentieren das Verständnis des Menschen von einem abstrakten System. Diese Modelle sind in der Regel nicht vollständig und werden ergänzt oder angepasst, sobald sich Widersprüche bei der Interaktion mit dem System ergeben.



## **Anhang B: Fragebogen**

Auf den folgenden Seiten ist der Fragebogen wiedergegeben, auf dem das Erfahrungsprofil der Teilnehmer an der Benutzerstudie erfasst und ihre jeweilige Einschätzung der Interaktionsmetaphern verzeichnet wurde.

# Fragebogen

zur Benutzerstudie, durchgeführt im Rahmen der Diplomarbeit  
„Navigations- und Interaktionstechniken“ an der Professur für Graphische  
Datenverarbeitung, Johann-Wolfgang Goethe Universität Frankfurt

## Teilnehmer-Daten

Name: \_\_\_\_\_ Benutzer-ID: \_\_\_\_\_

Alter: \_\_\_\_\_ Geschlecht: m w

e-mail: \_\_\_\_\_

Erfahrung/Expertise      0   1   2   3   4

- WIMP-Systeme:     

- 3D-Anwendungen:     

- 3D-Eingabe-Geräte:     

- Virtual Sphere Metapher:     

- Zauberwürfel:     

## Erläuterung und Training

Rotation

- RubiksOnline: \_\_\_\_\_

- Bezugsrahmen: \_\_\_\_\_

- Spotlight: \_\_\_\_\_

- Start/Ziel: \_\_\_\_\_

Inspektion

- Virtual Sphere: \_\_\_\_\_

- Eckpunkte: \_\_\_\_\_

- Farbnavigation: \_\_\_\_\_

Durchführung

Lösen einer Seite am Realwürfel [Zeit + #Rotationen]

- 1. Versuch: \_\_\_\_\_ - 3. Versuch: \_\_\_\_\_

- 2. Versuch: \_\_\_\_\_

Bewertung der Inspektionsmetaphern

kompliziert      0      1      2      3      4      intuitiv

Virtual Sphere:                             

Eckpunkte:                             

Farbnavigation:                             

langwierig      0      1      2      3      4      effizient

Virtual Sphere:                             

Eckpunkte:                             

Farbnavigation:                             

schwer erlernbar      0      1      2      3      4      leicht

Virtual Sphere:                                    erlernbar

Eckpunkte:                             

Farbnavigation:

<u>Bewertung der Rotationsmetaphern</u>						
<u>kompliziert</u>	<u>0</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>intuitiv</u>
Rubiks Online:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Start/Ziel:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Spotlight:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Bezugsrahmen:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<u>langwierig</u>	<u>0</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>effizient</u>
Rubiks Online:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Start/Ziel:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Spotlight:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Bezugsrahmen:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<u>schwer erlernbar</u>	<u>0</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>leicht</u>
Rubiks Online:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	erlernbar
Start/Ziel:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Spotlight:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Bezugsrahmen:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Anmerkungen nach einzelnen Versuchsteilen:

Repeater 1:

---

---

Repeater 2:

---

---

Repeater 3:

---

---

Repeater 4:

---

---

Farbseite 1:

---

---

Farbseite 2:

---

---

Farbseite 3:

---

---

Kombinationen mehrerer Inspektions- bzw.  
Rotationsmetaphern

---

---

---

---

---

Anregung und Kommentare

---

---

---

---

---

---

---

---

---

---



## Anhang C: Kommentiertes Literaturverzeichnis

- [balaguer91] Balaguer, F.; Mangili, A.: *Virtual Environments*, in: Thalmann, D. (Hrsg.): *New Trends in Animation and Visualization*, S. 91-106, Wiley, 1991  
<http://citeseer.nj.nec.com/379199.html>  
*umfassende Übersicht über Anwendungen der virtuellen Realität, Darstellung von 3D-Interaktionsgeräten und -techniken (unter anderem viewpoint specification, object manipulation)*
- [barrilleaux01] Barrilleaux, J.: *3D User Interfaces with Java 3D*, Manning, 2001  
*beschreibt grundlegende Interface-Techniken (control, feedback, visualization, navigation, manipulation, access), erläutert die Konzepte der Java3D-API und entwickelt ein eigenes Java3D-Framework für interaktive Anwendungen*
- [bertelsmann92] *Bertelsmann Lexikon*, Bertelsmann Lexikon Verlag, 1992
- [bowman01] Bowman, D.; Kruijff, E.; La Viola, J.; Poupyrev, I.: *An Introduction to 3-D User Interface Design*, (online), 2001  
<http://citeseer.nj.nec.com/bowman01introduction.html>  
*einführende Übersicht über Ein- und Ausgabe-Geräte der virtuellen Realität, Beschreibung grundlegender Interaktionstechniken (Navigation, Selektion, Manipulation)*
- [brockhaus84] *Der große Brockhaus*, 18. Auflage, F. A. Brockhaus, 1984
- [broy00] Broy, V.: *Grundlagen für das Design von Benutzungsschnittstellen*, Seminararbeit im Rahmen des Proseminars *Design von Benutzungsoberflächen* im SS 2000, TU München, 2000  
*Einführung in klassische Modellierungstechniken und Besonderheiten der Mensch-Maschine-Kommunikation*

- [butler95] Butler, K.; Jacob, R.; John, B.: *Introduction & Overview to Human-Computer Interaction*, CHI '95 Tutorial, 1995  
*Darstellung der Ursprünge der HCI, Erläuterung psychologischer Aspekte*
- [chen88] Chen, M.; Mountford, S.; Sellen, A.: *A Study in Interactive 3-D Rotation Using 2-D Control Devices*, in: *Computer Graphics*, Vol. 22, 1988  
*beschreibt die Konzeption der Virtual Sphere-Metapher und erläutert die Ergebnisse der Benutzerstudie*
- [debatin97] Debatin, B.: *Metaphern und Mythen des Internet*, (online), 1997  
<http://www.uni-leipzig.de/~debatin/German/NetMet.htm>  
*Darstellung von gesellschaftspolitischen Zusammenhängen moderner Technikanwendungen (Technikbild, Strukturmetaphern, Funktionsmetaphern)*
- [dhonline02] Lösungsweg zur Ordnung des Magischen Würfels, (online), 2002  
<http://www.dh-online-net.de/zaub.htm>
- [dix95] Dix, A.; Finlay, J.; Abowd, G.; Beale, R.: *Human-Computer Interaction*, Prentice Hall, 1995  
*Strategien der Problemlösung, Abläufe beim Lernen neuer Zusammenhänge (Gestalt-Theorie, Problembereich-Theorie, gedankliche Modelle)*
- [dtv80] *dtv Lexikon*, Deutscher Taschenbuch Verlag, 1980
- [duden74] *Duden Fremdwörterbuch*, Bibliographisches Institut, 3. Auflage, 1974
- [encarta01] *Encarta 2001*, Microsoft Corporation, 2001
- [foley96] Foley, J.: *Computer graphics: principles and practice*, Addison-Wesley, 1996  
*in Kapitel 9 erfolgt eine ausführliche Beschreibung der Grundlagen des Dialogdesigns (modes and syntax, visual design, design methodology)*

- [haik02] Haik, E.; Barker, T.; Sapsford, J.; Trainis, S.: *Investigation into Effective Navigation in Desktop Virtual Interfaces*, Proceedings of the 7th International Conference on 3D Web Technology, 2002  
*Beschreibung von Navigationsstrategien in sogenannten non-immersive virtual environments (NIVEs), Identifizierung von relevanten Variablen (navigation, mouse, task, orientation, finding, freedom, preference, time, remembering), ausführliche Beschreibung der durchgeführten Benutzerstudie*
- [hand93] Hand, C.: *A Survey of 3-D Input Devices*, Department of Computing Science, De Montfort University, 1993  
*Analyse von Besonderheiten der 3D-Interaktion, Beschreibung unterschiedlicher Tracking-Technologien (unter anderem mechanisch, elektromagnetisch, optisch), vergleichende Darstellung der von den einzelnen Geräten bereitgestellten Freiheitsgraden*
- [hannema01] Hannema, D.: *Interaction in Virtual Reality*, Dissertation, Universiteit van Amsterdam, 2001  
*in Kapitel 2 Beschreibung grundlegender Interaktionstechniken der immersiven virtuellen Realität (arm-extension, ray-casting, multi-modal interaction)*
- [hanson97] Hanson, A.; Wernert, E.: *Constrained 3D Navigation with 2D Controllers*, Computer Science Department, Indiana University, 1997  
<http://www.cs.indiana.edu/~hanson/papers/vis97.pdf>  
*Darstellung der prinzipiellen Schwierigkeiten bei der Durchführung von Navigationsaufgaben im 3D-Raum, Berücksichtigung von 2D-Eingabegeräten, aufwändige mathematische Konzeption eines Systems zur constraint-basierten Navigation*
- [herndon94] Herndon, K.; van Dam, A.; Gleicher, M.: *Workshop on the Challenges of 3D Interaction*, SIGCHI Bulletin 26(4), 1994  
<http://www.cs.wisc.edu/graphics/Papers/Gleicher/CMU/workshop-report.pdf>  
*Beschreibung der besonderen Aufgaben auf dem Gebiet der 3D-Interaktion, Berücksichtigung von psychologischen Aspekten, Darstellung der Freiheitsgrade verschiedener Eingabegeräte, Übertragung von Prinzipien des 2D-Designs*

- [igarashi98] Igarashi, T.; Kadobayashi, R.; Mase, K.; Tanaka, H.: *Path Drawing for 3D Walkthrough*, (online), 1998  
<http://www.mtl.t.u-tokyo.ac.jp/~takeo/papers/uist98.pdf>  
*Konzeption einer Navigationsmetapher, bei der Pfadangaben des Anwenders gesondert berücksichtigt werden, Erweiterung der Flying-Metapher*
- [ix02] *Höhlenzeitalter*, iX November 2002, S. 99-103, Heise, 2002  
*Darstellung des DAVE-Projekts, das immersive virtuelle Anwendungen auf Grundlage eines Linux-Systems realisiert*
- [kettner93] Kettner, L.: *Mathematisch-informationstechnische Untersuchung von 3D-Metaphern*, Diplomarbeit, Institut für Betriebs- und Dialogsysteme, Universität Karlsruhe, (online), 1993  
<ftp://ftp.ira.uka.de/pub/uni-karlsruhe/papers/da.3d-metaphors.ps.gz>  
*Umfangreiche Untersuchung der 3D-Metapher und ihrer Funktion, Beschreibung einzelner Aspekte der menschlichen Wahrnehmung (unter anderem Arbeitsgedächtnis, Kinästhetik, Fitts' Gesetz) und Darstellung bestehender 3D-Metaphern für verschiedene Operationen*
- [kettner94] Kettner, L.: *Theoretical Foundations of 3D-Metaphors*, Institut für Betriebs- und Dialogsysteme, Universität Karlsruhe, (online), 1994  
<http://www.inf.ethz.ch/personal/kettner/pub/3d-metaphors.workshop.ps.gz>  
*Konzeption eines Klassifizierungsschemas für 3D-Metaphern*
- [krömker02] Krömker, D.: *Skript zur Vorlesung Virtual Reality und Augmented Reality*, Professur für Graphische Datenverarbeitung, Universität Frankfurt, 2002  
*unter anderem: Beschreibung der Tiefen- und Raumwahrnehmung, Bedeutung von depth cues und damit verbundene Probleme bei interaktiven MR-Systemen*
- [kuhn95] Kuhn, W.: *7+/-2 Questions and Answers About Metaphors for GIS User Interfaces*, Department of Geoinformation, Technical University Vienna, (online), 1995  
[ftp://ftp.geoinfo.tuwien.ac.at/kuhn/1563\\_prn\\_NatoARW\\_Metaph.ps](ftp://ftp.geoinfo.tuwien.ac.at/kuhn/1563_prn_NatoARW_Metaph.ps)  
*Untersuchung der Bedeutung von Interface-Metaphern, Beschreibung grundlegender Konzepte und Erläuterung von Verfahren für den Entwurf neuer Metaphern*

- [marcus02a] Marcus, A., *Metaphors and User Interfaces in the 21th Century*, Interactions, S. 7-9, March + April 2002  
*Beschreibung der Bedeutung von metaphorischen Konzepten für die menschliche Kommunikation*
- [marcus02b] Marcus, A.: *Dare We Define User-Interface Design?*, Interactions, S. 19-24, September + October 2002  
*kurze Zusammenstellung wichtiger Begriffe des UI-Designs*
- [marshall01] The Ultimate Solution to Rubik's Cube, (online), 2001  
<http://www.olympus.net/personal/prmhem/>
- [mine97] Mine, M.; Brooks, F.; Sequin, C.: *Moving Objects in Space: Exploiting Proprioception in Virtual-Environment Interaction*, Department of Computer Science, University of North Carolina, 1997  
*Beschreibung verschiedener Interaktionstechniken des 3D-Raums, vor allem Verfahren der immersiven virtuellen Realität, Darstellung der Head-butt Zoom-Metapher*
- [mohan98] Mohan, S.: *The Fourth Generation of 3D Graphics APIs has arrived!*, (online), 1998  
[http://java.sun.com/products/java-media/3D/collateral/wp\\_mktg/j3d\\_wp.pdf](http://java.sun.com/products/java-media/3D/collateral/wp_mktg/j3d_wp.pdf)  
*Darstellung grundlegender Konzepte der Java3D-API, Vergleich mit bestehenden 3D-Programmierungsumgebungen*
- [norman89] Norman, D.: *Dinge des Alltags: gutes Design und Psychologie für Gebrauchsgegenstände*, Campus, 1989  
*in Kapitel 3 Beschreibung des menschlichen Gedächtnisses (unter anderem Gedächtnis für Willkürliches, Gedächtnis für sinnvolle Beziehungen), Darstellung der Funktion natürlicher Mappings*
- [ousland98] Ousland, A.; Turcato, H.: *Navigating 3D Environments*, European Institute for Research and Strategic Studies in Telecommunications, 1998  
*Erläuterung der Walking- und Flying-Metapher zur Navigation, ausführliche Beschreibung der durchgeführten Benutzerstudie*

- [pierce00] Pierce, J.: *Expanding the Interaction Lexicon For 3D Graphics*, Thesis Proposal, School of Computer Science, Carnegie Mellon University, 2000  
*kurzer historischer Überblick der Entwicklung von interaktiven Systemen von den Anfängen bis heute, Kategorisierung bestehender Interaktionskonzepte, Beschreibung eines generischen Ansatzes, der darauf beruht, metaphorische Konzepte zu entwickeln, die mit Grundannahmen der Realität brechen*
- [preim99] Preim, B.: *Entwicklung interaktiver Systeme*, Springer, 1999  
*ausführliche Erläuterung von Entwurfsprinzipien für interaktive Systeme, Beschreibung der Bedeutung von Metaphern in der Mensch-Computer-Interaktion*
- [reimann02] *Anleitung für den Zauberwürfel*, (online), 2002  
<http://keks.de/wuerfel/index.html>
- [rubik02] Rubik/Seven Towns: *Rubik's Online – Online Cube*, 2002  
[http://www.rubiks.com/cube\\_online.html](http://www.rubiks.com/cube_online.html)  
*eine Online-Version des Rubik's Cube; die hier eingesetzten Metaphern zur Inspektion und Navigation dienen als Referenz für die Benutzerstudie*
- [rubik98] Rubik/Seven Towns: *Brief History of the Cube*, (online), 1998  
<http://www.rubiks.com/cubehistory.html>
- [schulze02] Schulze, D.: *Visualisierung semantischer Constraints*, Diplomarbeit, Professur für Graphische Datenverarbeitung, Universität Frankfurt, 2002  
*Konzeption eines Systems zur Visualisierung von Constraints, die sich aus semantischen Beziehungen der realen Welt ergeben; Implementierung einer prototypischen Anwendung*
- [shneiderm98] Shneiderman, B.: *Designing the user interface: strategies for effective human-computer-interaction*, Addison Wesley Longman, 1998  
*Beschreibung von Entwurfsprinzipien des User Interface-Designs, Gegenüberstellung Anfänger/fortgeschrittener Benutzer, Darstellung gängiger Verfahren für Benutzerbefragungen und Akzeptanztests*

- [slater02] Slater, M.; Steed, A.; Chrysanthou, Y.: *Computer Graphics and virtual environments: from realism to real-time*, Pearson Education, 2002  
*Gegenüberstellung unterschiedlicher Eingabegeräte, in Kapitel 21*  
*Beschreibung von besonderen Verfahren bei der 3D-Interaktion mit Hilfe von 2D-Eingabegeräten*
- [speed02] Chris Hardwick's Rubik's Cube Page: *Rubik's Cube 3x3x3: Fastest Times to Solve*, (online), 2002  
[http://www.speedcubing.com/chris/recs\\_cube\\_333.html](http://www.speedcubing.com/chris/recs_cube_333.html)
- [stork00] Stork, A.: *Effiziente 3D-Interaktions- und Visualisierungstechniken für benutzer-zentrierte Modellierungssysteme*, Dissertation, Fachbereich Informatik an der Technischen Hochschule Darmstadt, 2000  
<http://homepages.fh-giessen.de/~hg11831/seminar/diss.pdf>  
*in Kapitel 2 Beschreibung von gängigen 3D-Eingabegeräten (unter anderem SpaceMouse, Polhemus-Sensor und Datenhandschuh)*
- [sun01] Sun: *The Java3D Tutorial*, (online), 2001  
<http://java.sun.com/products/java-media/3D/collateral/>
- [sun02] Sun: *Java3D API Documentation*, (online), 2002  
[http://java.sun.com/products/java-media/3D/forDevelopers/J3D\\_1\\_2\\_API/j3dapi/](http://java.sun.com/products/java-media/3D/forDevelopers/J3D_1_2_API/j3dapi/)
- [tan01] Tan, D.; Robertson, G.; Czerwinski, M.: *Exploring 3D Navigation: Combining Speed-coupled Flying with Orbiting*, Microsoft Research, 2001  
<http://citeseer.nj.nec.com/527495.html>  
*Konzeption einer Navigationsmetapher, bei der sich die Bewegungsgeschwindigkeit auf die Flughöhe auswirkt; in der Umgebung von Objekten der Szene kann in einen Orbiting-Modus zur Inspektion gewechselt werden; ausführliche Beschreibung der durchgeführten Benutzerstudie*
- [techtrans02] Techtrans: *Space Mouse*, (online), 2002  
<http://www.techtrans.de/html/bereich/computer/t14.htm>

- [theisinger02] Theisinger, R.: Haptische 3D-Benutzungsflächen, Diplomarbeit, Professur für Graphische Datenverarbeitung, Frankfurt, 2002  
*ausführliche Betrachtung des Tastsinns und seiner Bedeutung für Benutzungsschnittstellen*
- [tsialtas01] Tsialtas, A.: *3D-Metaphern und Interaktion*, Diplomarbeit, Professur für Graphische Datenverarbeitung, Frankfurt, 2001  
*Beschreibung einer Studie, in der verschiedene Interaktionsmetaphern untersucht und Kriterien zu ihrer Bewertung aufgestellt werden; Entwurf eines Konzepts für einen Verhaltenseditor*
- [väänänen95] Väänänen, K.: *Metaphor-Based User Interfaces for Information Authoring, Visualization and Navigation in Multimedia Environments*, Dissertation, Fachbereich Informatik an der Technischen Hochschule Darmstadt, 1995  
*in Kapitel 5 ausführliche Beschreibung von User Interface-Metaphern, besondere Berücksichtigung von Problemen, die sich durch den Einsatz von Metaphern ergeben können*
- [vballegooij01] van Ballegooij, A.; Eliens, A.: *Navigation by Query in Virtual Worlds*, (online), 2001  
<http://portal.acm.org/citation.cfm?id=363361.363380>  
*Beschreibung eines Konzepts zur Navigation durch 3D-Umgebungen mit Hilfe einer interaktiven Auswahl des gewünschten Zielorts*
- [vdam97] van Dam, A.: *Post-WIMP User Interfaces*, in Earnshaw, R. (Hrsg.): *Frontiers of human-centred computing, online communities and virtual environments*, pp. 163-178, Springer, 1997  
*Darstellung von möglichen Eigenschaften zukünftiger Benutzungsschnittstellen, unter anderem 3D Interfaces*
- [ware99] Ware, C.: *Information visualization*, Academic Press, 1999  
*in Kapitel 10 Beschreibung von grundlegenden Eigenschaften der menschlichen Wahrnehmung (visual-manual control loop, view refinement and navigation loop, problem-solving loop)*