

3 Metaphern und Widgets für interaktive 3D-Anwendungen

Benutzungsschnittstellen interaktiver 3D-Anwendungen für den Bereich Desktop-VR stehen im Mittelpunkt dieses Kapitels, dem die Analyse von etwa 240 wissenschaftlichen Arbeiten zu 3D-Benutzungsoberflächen und -Interaktionstechniken zugrunde liegt. Zunächst werden Begriffe und theoretische Grundlagen dreidimensionaler Benutzungsschnittstellen (*3D Graphical User Interfaces*, auch 3D-GUI) erörtert, und es wird ein kurzer Abriss über ihre Entwicklungsgeschichte gegeben. Eine Charakterisierung dieses Schnittstellentyps und die Vorstellung beispielhafter 3D-GUIs führen schließlich zu einer vergleichenden Analyse der Besonderheiten von 2D- und 3D-GUIs. Die frei nach Shakespeare formulierte Frage „*3D or not 3D?*“ [Shnei02] kann dabei nicht pauschal und abschließend beantwortet werden. Wohl aber kann die Notwendigkeit deutlich gemacht werden, sowohl auf konzeptionell-gestalterischer als auch auf Spezifikations- und Implementierungsseite deutliche Fortschritte bei der Entwicklung von 3D-GUIs anzustreben. Hier mangelt es – insbesondere im Bereich Desktop-VR – an Gestaltungsempfehlungen, Designrichtlinien und Standardisierungsbemühungen, bevor sich die aufgabenbezogene Effektivität und Effizienz von 3D-GUIs für bestimmte Anwendungsdomänen besser messen und überprüfen läßt.

Ein Ziel dieses Kapitels ist es deshalb zunächst, generelle Designrichtlinien für die Entwicklung von 3D-Applikationen aufzustellen. Eine Desktop-VR-Anwendung besteht typischerweise aus räumlichen Strukturen, in die sowohl die zu bearbeitenden bzw. zu betrachtenden 3D-Objekte als auch 3D-Interaktionselemente integriert sind. Für die räumliche Gliederung von 3D-Applikationen wurde das Konzept der *Action Spaces* entwickelt, das mit verschiedenen Metaphern vorgestellt wird. Der Hauptfokus des Kapitels liegt jedoch auf der systematischen Klassifikation und Spezifikation von 3D-Interaktionselementen (*3D-Widgets*), die als intermediäre Interaktionselemente zentraler Bestandteil von Desktop-VR 3D-GUIs sind. Dies geschieht in Abgrenzung zu Forschungsarbeiten auf dem Gebiet dreidimensionaler Interaktionstechniken in rein immersiven VR-Umgebungen. Es wird die erarbeitete Systematik von 3D-Widgets detailliert vorgestellt, wobei abschließend an einem ausgewählten Beispiel deutlich gemacht werden soll, wie diese Widgets spezifiziert werden können. Erst dadurch ist eine einheitliche Verwendung in verschiedenen Applikationen möglich, womit ein Beitrag zu Wiederverwendbarkeit und Standardisierung bei der Entwicklung von 3D-GUIs geleistet wird.

3.1 3D-Benutzungsoberflächen

3.1.1 Grundlagen: Mentales Modell und Metaphern

In vielfältiger Weise ist die Thematik der Metaphern in verschiedensten Fachgebieten bearbeitet worden, darunter Linguistik, Literatur-, Theater- und Filmwissenschaft. Auch innerhalb der Informatik und vor allem für Benutzungsschnittstellen haben Metaphern inzwischen einen wichtigen Platz eingenommen, da Computeranwendungen komplexer werden und immer mehr Alltagsnutzer mit Computersystemen interagieren. Für eine ausführliche und gelungene Darstellung von Metaphern in Benutzungsschnittstellen wird auf die Dissertation von Väänänen [Väänä95] verwiesen.

Menschen nutzen ihre Erfahrungen, um mentale Modelle aufzubauen und zu aktualisieren [Väänä95]. Ein *mentales Modell* ist laut [Marcu94] die geeignete Organisation und Repräsentation von Daten, Funktionen, Arbeitsaufgaben und Rollen, die Menschen innerhalb von Arbeits- oder anderen sozialen Situationen einnehmen. Dabei können Metaphern den Prozeß der ständigen Evaluierung und Aktualisierung mentaler Modelle erleichtern, indem sie für ungewohnte Erfahrungen vertraute Konzepte zur Verfügung stellen.

Metaphern lassen sich allgemein definieren als grundlegende Konzepte, Begriffe und Bilder, mit denen und durch die Informationen leicht erkannt, verstanden und erinnert werden können [Marcu94]. Metaphern erleichtern die Übertragung von Konzepten und Wissen aus einem bekannten Bereich in einen unbekanntem Zielbereich bzw. eine unvertraute Situation. Dabei setzen sie Ideen zueinander in Beziehung, bringen Erleichterungen für die Benutzung von Werkzeugen, Computerschnittstellen sowie für Informationssuche und Navigation. Sie dienen somit auch als Basis für Interaktionen mit dem Computer, dessen Welt aus Bits und Zeichen eine nur geringe Entsprechung in unseren alltäglichen Erfahrungsbereichen besitzt. *UI-Metaphern* im Besonderen werden in [Blume90] folgendermaßen definiert:

„A human interface metaphor involves importing features from the real world into a computer application to determine the presentation and behavior of the objects and operations that the computer application makes available to the user.“

Laut [Marcu94] lassen sich UI-Metaphern für die Objekte von Interaktionen (also *Substantive* der visuell-verbalen Kommunikation) sowie für Operationen, Prozesse, Handlungen und Algorithmen (also *Verben* der visuell-verbalen Kommunikation) unterscheiden. Metaphern der ersten Kategorie wären z.B. für den Begriff Dokument: Bücher, Kapitel, Papierstapel etc. Beispielmetaphern der zweiten Kategorie sind z.B. Berühren, Darauf-Zeigen, Einkreisen, Drücken oder In-die-Hand-Nehmen allein für den Begriff *Selektieren*. Eine (größtenteils erfolgreiche) Mischung von Metaphern aus verschiedenen Kontexten ist uns durch die von Xerox, dann Apple und später Microsoft verwendete Schreibtischmetapher moderner grafischer Benutzungsoberflächen geläufig. Bürobegriffe, wie Dokumente, Ordner und Schreibtisch werden mit architektonischen Metaphern, wie Fenster, Galerie oder Raum vermischt [Marcu94]. Auch Metaphern aus der Technik finden in heutigen 2D-GUIs Anwendung, zum Beispiel die des Steuerpultes für Dialogboxen, in denen Schaltknöpfe, Schiebe- und Drehregler etc. angeordnet sind.

UI-Metaphern bieten gerade für die Benutzung von inhärent abstrakten und hochgradig immateriellen Computersystemen einige Vorteile (teilweise aus [Väänä95]):

- Die Bekanntheit des Nutzers mit ihnen impliziert schnelleres Verständnis und Lernen durch die Übertragung bekannten Wissens auf die neue Domäne.
- Sie erhöhen den Vertrautheitsgrad mit der Benutzungsschnittstelle, erleichtern Wiedererkennung von Objekten oder Optionen und erleichtern den Dialogprozeß.

- Sie vermitteln dem Benutzer ein konsistentes mentales, konzeptuelles Modell von teilweise abstrakten Vorgängen, womit die kognitive Last und nötige Gedächtnisleistung reduziert wird.
- Sie unterstützen die menschlichen Fähigkeiten der räumlichen Orientierung und erlauben eine bessere Navigation des Nutzers durch eine Anwendungssoftware.
- Der Zugang zu einem System für verschiedene Nutzergruppen wird durch alternative Metaphern erleichtert. So können sie für Novizen besonders hilfreich sein, während „Abkürzungen“ oder alternative Metaphern für Experten ein schnelleres Arbeiten ermöglichen.
- Sie verbessern die Motivation der Benutzer und erhöhen die Attraktivität einer Benutzungsschnittstelle.

Die Anwendung von UI-Metaphern bringt jedoch auch Gefahren und Nachteile mit sich. Metaphern lassen sich oft schwer finden und manchmal nur schlecht miteinander kombinieren, sind technisch nicht immer einfach zu realisieren, konsumieren mitunter unnötig viel Systemleistung und sind relativ unflexibel gegenüber funktionalen Änderungen und Erweiterungen eines Systems. Auch kann die Abbildung zwischen Quell- und Zieldomäne problematisch sein, wenn Benutzer durch unangebrachte, unvollständige oder übertriebene Übertragung der Konzepte verwirrt werden. Dabei erzeugen vertraute Metaphern auch eine sehr hohe Erwartungskonformität bei Nutzern. Bei der Entwicklung von Metaphern und Interaktionskonzepten für Benutzungsschnittstellen sind deshalb folgende Kriterien und Anforderungen zu beachten:

- *Nützlichkeit und Anwendbarkeit* im Bezug auf einen bestimmten Anwendungskontext;
- *Verständlichkeit und Klarheit* des zugrundeliegenden Konzeptes;
- *Eignung* für die beabsichtigten Zielgruppen;
- *Erweiterbarkeit* des metaphorischen Konzeptes, z.B. für alternative Nutzerbedürfnisse und funktionale Änderungen;
- *Performance* für schnelles Arbeiten, evtl. Abwägen zwischen Verständlichkeit und Effizienz.

Zum Abschluß soll der Begriff *3D-Metapher* als Konkretisierung von UI-Metaphern eingeführt werden. In einer Arbeit zur mathematisch-informationstheoretischen Untersuchung von 3D-Metaphern [Kettn93] wird er als Ausführung einer dreidimensionalen Operation mit einem zweidimensionalen Eingabegerät erklärt. Aus der Notwendigkeit dieser Abbildung bei Desktop-VR-Anwendungen läßt sich unmittelbar die Bedeutung von Metaphern für diesen Bereich ableiten. 3D-Metaphern bestehen laut [Kettn93] aus zwei Teilen. Einerseits aus dem metaphorischen Konzept, das die strukturelle Abbildung des auf existierenden Erfahrungen beruhenden mentalen Modells des Benutzers in die Benutzungsschnittstelle ermöglicht. Andererseits aus der Implementierung, die eine Abbildung von Ereignissen der Nutzerschnittstelle auf das zugrundeliegende informationstheoretische Modell gestattet.

3.1.2 Historie interaktiver Benutzungsoberflächen

In diesem kurzen Abriss soll gezeigt werden, aus welcher Entwicklungslinie sich heutige und künftige Benutzungsschnittstellen ableiten und wie 3D-GUIs sich dabei einordnen lassen.

3.1.2.1 50er-80er Jahre: Batchsysteme, Kommandozeilen und Parameter

Die frühen Computersysteme der 50er und 60er Jahre des letzten Jahrhunderts sind dadurch gekennzeichnet, daß noch keine direkte interaktive Nutzung stattgefunden hat und Computer im Batch-Modus arbeiteten. Spätere Systeme waren ab den 60er Jahren durch einen alphanumerischen Volltextbildschirm geprägt und tastaturorientiert. Die Kommandozeile, bei der Befehle mit bestimmten Parametern eingegeben werden, bildete die zentrale Interaktionsform.

Obwohl für Expertennutzer effizient benutzbar, stellten diese Systeme jedoch hohe Anforderungen an das Erinnerungsvermögen und die Aufmerksamkeit der Benutzer. Man spricht dabei auch vom *Konversationellen Paradigma* mit klaren Sequenzen von syntaktisch festgelegten Eingaben und Systemausgaben. In [Shnei98] (S. 65 ff.) wird erläutert, welche Probleme die Abhängigkeit von exakter und fehlerintoleranter Syntax insbesondere für Novizen oder Gelegenheitsnutzer mit sich brachte. Der – immer noch textbasierte – Interaktionsstil auf Basis von Menüs stellte bereits eine starke Entlastung des Nutzergedächtnisses dar [Ziegl96] und war ein wichtiger Schritt für die Entwicklung grafischer Benutzungsoberflächen.

3.1.2.2 80er Jahre bis heute: Grafische Benutzungsschnittstellen (WIMP)

Mit Rastergrafik arbeitende, vernetzte Computer und auf der *point-and-click* Metapher basierende grafische Benutzungsschnittstellen (*Graphical User Interfaces* – GUIs) sind das Erbe der Entwicklungen aus den späten 70er Jahren [Dam97]. Wenn auch das berühmte, 1981 im *Xerox Palo Alto Research Center (PARC)* entwickelte GUI *Xerox Star* und sein unmittelbarer Nachfolger *Apple Lisa* kommerziell nicht erfolgreich waren, wurde die Idee jedoch 1984 durch den *Apple Macintosh* populär gemacht. Die grafische Betriebssystemoberfläche des Apples bildet die Basis für alle folgenden GUI-Entwicklungen, was zu einem sehr hohen Grad an Konsistenz und Standardisierung geführt hat. Bei der Entwicklung der ersten GUIs stand die elektronische Nachbildung einer gewohnten Schreibtisch- und Büroumgebung im Vordergrund. Dieses als *Desktop-Metapher* häufig beschriebene und bis heute dominante Schnittstellenkonzept beinhaltet auch die Benutzung einer Maus als Eingabegerät, mit dem Dokumente geöffnet, verschoben und auf vielfältige Art und Weise manipuliert werden. Daher wurde für dieses 1981 erstmalig implementierte Konzept auch der Name *WIMP* = *Windows, Icons, Menus, Pointing Device* eingeführt [Liu00]. Eine alternative Erklärung des Akronyms als *Windows, Icons, Mouse, Pull-down menu* wird in [Shnei98] gegeben.

Die direkte Manipulation ist eine grundlegende Funktionalität grafischer Benutzungsoberflächen ([Shnei98], [Ziegl96]). Dabei versteht Shneiderman unter „direkter Manipulation“ die unmittelbare Interaktion mit sichtbaren Informationsobjekten, wobei Hilfsmittel zum Zeigen, Selektieren und Ziehen ebenso wie unmittelbares Feedback als Konsequenz einer Handlung zur Verfügung stehen müssen. Damit weisen direktmanipulative Systeme eine höhere Fehler-toleranz auf und eliminieren das Konzept der Syntax. Weitere Konzepte und Techniken, wie *drag and drop*, hierarchische Menüsysteme, Werkzeugpaletten oder multimediale Erweiterungen gehören seit langem zum Standardrepertoire von WIMP-GUIs. Prinzipiell hat sich jedoch nur wenig seit den frühen fensterbasierten Systemen geändert. Nach wie vor wird insbesondere von unerfahrenen Nutzern die Verwendung von Maus und Tastatur nicht als natürlich empfunden. WIMP-GUIs arbeiten sequentiell und basieren auf diskreter Ereignisverarbeitung und einem Interaktionskanal. Es werden keine multiplen und simultanen Ein- und Ausgabeströme von mehreren Nutzern unterstützt.

3.1.2.3 Ab 1990: Post-WIMP GUIs

Schon 1992 wird in [Dam92] für die 90er Jahre eine rapide Zunahme an interaktiven 3D-Applikationen sowohl für traditionelle 3D-Domänen (z.B. CAD) als auch vormals durch 2D-Schnittstellen bestimmte Applikationen (z.B. Lehr-/Lernanwendungen) prognostiziert. Bereits 1990 wurden auf einem SIGGRAPH-Workshop Softwarearchitekturen und Metaphern für sogenannte *Non-WIMP User Interfaces* diskutiert [Green91], wobei auch erstmals der Begriff *Post-WIMP GUI* erwähnt wurde. Van Dam definiert dann in [Dam97] die künftige vierte Generation von Benutzungsschnittstellen als Ära der sogenannten *Post-WIMP User Interfaces*, die durch natürlichere Präsentationsformen (z.B. 3D-Grafik) und Interaktionstechniken (z.B. Gesten und Sprachsteuerung) gekennzeichnet sind. In [Green91] werden vier unterschiedliche *Non-WIMP-Styles* beschrieben, darunter auch Anwendungen der virtuellen Realität.

In den 90er Jahren wurden zahlreiche Forschungsarbeiten auf diesem Gebiet begonnen, um die an Papierdokumenten orientierte Desktop-Metapher durch intuitivere und den großen multimedialen Informationsmengen sowie neuartigen Ein- und Ausgabegeräten angepasste Schnittstellenkonzepte zu ersetzen. Dazu zählen zum Beispiel sogenannte *Zooming User Interfaces (ZUI)* [Liu00] oder dreidimensionale Räume. Durch die perspektivische Verkürzung besitzen sie eine natürliche Detail- und Kontext-Wirkung und eignen sich zum Beispiel zur Darstellung hierarchisch strukturierter Daten wie bei den Xerox PARC - Entwicklungen *Cone Trees* [Rober91] oder *Perspective Walls* [Macki91]. Vollständig dreidimensionale Benutzungsschnittstellen, wie die Task Gallery von Microsoft Research [Rober00], sind eine Weiterentwicklung unter konsequenter Ausnutzung der Möglichkeiten von interaktiver 3D-Grafik (s.a. weitere Beispiele im Abschnitt 3.1.4).

Eine parallele Entwicklung bzw. Weiterentwicklung dreidimensionaler Benutzungsoberflächen (3D-GUIs) stellen multimodale Benutzungsschnittstellen, wie z.B. mUltimo3D [Multimo3D@], dar. Sie stellen hohe Anforderungen durch Nutzung multipler paralleler, breitbandiger Ein- und Ausgabekanäle, die statt ausschließlich diskreter Signale auch kontinuierliche Signalströme verarbeiten müssen [Dam97]. VR-Applikationen und multimodale Benutzungsschnittstellen sind im Gegensatz zu WIMP-GUIs durch teilweise parallele Handlungen geprägt, z.B. die zeitgleiche Kombination von Stimmeingabe und Blicksteuerung. Die im Fokus dieser Arbeit stehenden 3D-GUIs und -Applikationen sind im Bezug auf Eingabegeräte und Sequentialität der Interaktion noch den klassischen WIMP-Anwendungen verpflichtet, während die bewußte Einbeziehung und Gestaltung des 3D-Interaktionsraumes schon auf Post-WIMP-Interfaces hinweist.

Die Dominanz des WIMP-Paradigmas in gegenwärtigen Computersystemen birgt neben dem großen Vorteil der Vertrautheit und des gewohnten Umganges aber auch die Gefahr in sich, daß alternative und innovative Ansätze kaum betrachtet und zugelassen werden [Pierc01]. Allein deshalb sollten dreidimensionale Benutzungsoberflächen – zwar nicht als substituierendes Paradigma, jedoch als lohnenswerte Alternative für zahlreiche Anwendungsbereiche – näher untersucht werden. Sie stellen also nur einen möglichen Aspekt künftiger GUIs dar und lassen sich bei entsprechender Reife der Konzepte und Technologien mit anderen Post-WIMP-Techniken kombinieren.

3.1.3 Definition und Charakteristik von 3D-Benutzungsschnittstellen

Nachdem in den vorausgegangenen Abschnitten ein historischer Abriß von Benutzungsschnittstellen vorgenommen wurde, soll hier nun eine genauere Begriffsbestimmung und Charakterisierung von dreidimensionalen Benutzungsschnittstellen erfolgen. Wir definieren 3D-GUIs wie folgt:

Eine dreidimensionale Benutzungsschnittstelle charakterisiert eine Applikation, die aus einem oder mehreren Action Spaces besteht, zwischen denen der Benutzer navigieren kann und in denen 3D-Dokumente und Dokumentbestandteile mit Hilfe von interaktiven, dreidimensionalen Widgets selektiert und manipuliert werden können.

Die Abbildung 4 zeigt eine dreidimensionale Benutzungsschnittstelle, deren Hauptbestandteile bezeichnet sind. Dabei wird die anzustrebende enge Integration des 3D-Dokumenttyps mit dreidimensionalen Schnittstellenelementen im virtuellen Raum deutlich. 3D-Dokumente können typischerweise virtuelle 3D-Objekte, z.B. Produkte – wie im Bild der Teil eines Motors – sein, ebenso aber auch abstrakte Hierarchie- und Informationsvisualisierungen oder andere Formen der wissenschaftlichen Visualisierung. Die eigentliche Benutzungsschnittstelle einer 3D-Applikation besteht aus den *Action Spaces* und *3D-Widgets*.

3.1.3.3 Interaktionstechniken

Laut der Definition von Conner et al. fallen Interaktionselemente ohne geometrische Repräsentation nicht unter den Begriff der 3D-Widgets, obwohl sie in [Conne92] nicht explizit ausgeschlossen werden. Die Entwicklung von Benutzungsschnittstellen ist auch eng mit der Entwicklung von spezifischen Interaktionstechniken verbunden [Zieg196]. So wird in [Bowma01b] der Begriff *Interaktionstechnik* wie folgt definiert:

Eine Interaktionstechnik ist Teil der Benutzungsschnittstelle. Dabei handelt es sich um eine Methode, die es dem Nutzer erlaubt, eine bestimmte Aufgabe im System zu erfüllen, wobei sowohl Hardware (Eingabegeräte) als auch Softwarebestandteile eingeschlossen sind.

Eine Interaktionstechnik kann ein einfacher Mausklick oder eine *drag and drop* - Interaktion sein, bei immersiven virtuellen Umgebungen aber auch komplexe Gesten mit einem Datenhandschuh einschließen. Obwohl Interaktionstechniken von Eingabegeräten abhängig sind, lassen sie sich unabhängig davon betrachten, weil ein und dieselbe Aufgabe mit verschiedenen Interaktionstechniken unter Nutzung verschiedener Eingabegeräte ausgeführt werden kann. Interaktionstechniken ließen sich auch als Widgets mit einem zugehörigen Verhalten – in den meisten Fällen jedoch ohne geometrische Repräsentation – bezeichnen [Hinz01].

3.1.3.4 Wesentliche Eigenschaften von 3D-Benutzungsschnittstellen

- Der 3D-Raum hat inhärent mehr Freiheitsgrade als der zweidimensionale Raum [Conne92]. Neben den aus dem Zweidimensionalen bekannten Parametern (wie Farbe, Anordnung etc.) stehen im 3D-Raum weitere Parameter zur Verfügung, darunter Perspektive, Blickwinkel oder Entfernung.
- Die Komplexität des Designs von 3D-GUIs ist aufgrund der erweiterten Freiheitsgrade sehr hoch. Unter Forschern und Praktikern gleichermaßen besteht Einigkeit darüber, daß 3D-Grafikanwendungen deutlich schwieriger zu entwerfen und zu implementieren sind als zweidimensionale Anwendungen, was sich auch bei der Gestaltung und Realisierung von 3D-Widgets zeigt [Leine97].
- Da 3D-Oberflächen häufig für Spezialaufgaben in spezialisierten Anwendungsdomänen entwickelt werden, sind die Schnittstellenbestandteile größtenteils auch auf diese Aufgaben zugeschnitten und nicht allgemein anwendbar, während 2D-GUIs hauptsächlich aus standardisierten Widgets aufgebaut sind.
- Trotz einer Forschungsdekade auf dem Gebiet von 3D-GUIs gibt es nur vergleichsweise geringe Erfahrungen mit dem Entwurf dreidimensionaler Schnittstellen, geschweige denn Designrichtlinien oder -standards. Das liegt unter anderem daran, daß 3D-GUIs mit einer Vielzahl von verschiedenen Toolkits und Formaten entwickelt werden, was die Wiederverwendbarkeit von Interaktionstechniken und Widgets erschwert.
- 3D-GUIs, insbesondere für Anwendungen der Virtuellen Realität, sind häufig von speziellen Ein- und Ausgabegeräten, also spezialisierter VR-Hardware, abhängig. Eingabegeräte besitzen in der Regel eine höhere Zahl von Freiheitsgraden als traditionelle Eingabegeräte. Ausgabegeräte unterstützen häufig stereoskopische Wiedergabe und dienen der Erzeugung eines Immersionsgefühls.

Da der Fokus dieser Arbeit nicht auf immersiven VR-Applikationen liegt, sondern auf Desktop-VR-Anwendungen, die mit standardisierten Ein- und Ausgabegeräten und 3D-Widgets als intermediären Interaktionselementen arbeiten, soll hier nicht im Detail auf die verschiedensten Geräte für 3D-GUIs eingegangen werden. Für eine ausführliche Klassifikation von Ein- und Ausgabegeräten wird z.B. auf [Hinz01] (Kap. 2.3 und 2.4) oder [Shnei98] (Kap. 9) verwiesen.

3.1.4 Beispiele für 3D-Benutzungsschnittstellen

Anhand der folgenden Beispiele soll das Potential von 3D-GUIs für verschiedene Bereiche aufgezeigt werden, wobei natürlich nicht alle Anwendungstypen für die Realisierung im 3D-Raum geeignet sind. Naheliegend ist die Nutzung von 3D-Applikationen für die Anzeige und Interaktion mit dreidimensionalen geometrischen Informationen, z.B. bei CAD-Anwendungen. Ein erfolgreiches Beispiel dafür ist das kommerzielle 3D-Modellierungs- und Animationsprogramm Caligari TrueSpace [TrueSpace@], bei dem eine Vielzahl kontextbasierter, direktmanipulativer 3D-Widgets zum geometrischen Modellieren eingesetzt wird. Auch beim wichtigen Bereich der 3D-Computerspiele könnte man im weiteren Sinne von 3D-GUIs sprechen, da neben der Navigation auch die Interaktion eine wesentliche Rolle spielt.

Die folgenden Beispiele von komplexeren dreidimensionalen Benutzungsschnittstellen zeigen jedoch auch andere erfolgreiche Anwendungsdomänen. Allen Beispielen ist gemeinsam, daß – etwa im Gegensatz zu 3D-Produktpräsentationen innerhalb von 2D-Webseiten – die gesamte Schnittstelle als 3D-Raum ausgeführt ist und zahlreiche Widgets zur Interaktion mit den Anwendungsobjekten enthält. Eine weitere Gemeinsamkeit der angeführten Beispiele ist ihre primäre Eignung für den Bereich Desktop-VR.

3.1.4.1 3D-Betriebssystemoberflächen

Naturgemäß handelt es sich bei dieser Gruppe um die wichtigste Anwendungsdomäne für 3D-GUIs, weshalb hier auch eine Reihe Arbeiten existieren. Darin wird versucht, die Desktop-Metapher von WIMP-GUIs durch eine dreidimensionale Betriebssystemoberfläche zu ersetzen. Bisher haben all diese Versuche noch keinen kommerziellen Durchbruch erzielt, was unter anderem am Fehlen eines allgemeinen mentalen Modells und an der primär zweidimensionalen Natur von Desktop-Anwendungen liegt (Bsp. Textverarbeitung oder Tabellenkalkulation). Die folgenden Beispiele zeigen jedoch die Stärke der dritten Dimension in der räumlichen Anordnung von Dokumenten / Applikationen und ihrem schnellen Wiederfinden auf. Abbildung 5 zeigt Bildschrimschnappschüsse von drei in den folgenden Absätzen beschriebenen 3D-Anwendungen.



Abbildung 5: Beispiele für 3D-Betriebssystemoberflächen. Von links nach rechts: Microsoft TaskGallery (Abb. aus [TaskGallery@]), Win3D, 3DNA

Task Gallery

Bei dieser von Microsoft Research entwickelten dreidimensionalen Betriebssystemoberfläche [TaskGallery@] für gewöhnliche Windows-Programme ist der Bildschirm als 3D-Raum gestaltet, in dem Tasks in Form von Bildern an den Wänden hängen. Ein Task kann dabei mehrere normale Windows-Anwendungsfenster umfassen, wobei Ein- und Ausgaben perspektivisch verzerrt umgelenkt werden. Ausgewählte Tasks werden in den Vordergrund gebracht, und man kann wie gewohnt in einem oder mehreren Fenstern damit arbeiten. Die *Task Gallery* besteht aus mehreren virtuellen Räumen, die mehrere Tasks enthalten können, wobei aktuell nicht genutzte an den Seitenwänden plazierte sind. Der Nutzer kann Tasks beliebig an den Wänden arrangieren, dabei wird sein räumliches Erinnerungsvermögen ausgenutzt [Rober00].

Die kommerziellen Oberflächen Win3D und 3DNA

Bei Win3D [Win3D@] der Firma ClockWise und 3DNA [3DNA@] der Firma 3DNA Inc. handelt es sich um dreidimensionale Oberflächen für Windows-Betriebssysteme. Bei Win3D enthalten die miteinander verbundenen Themenräume *Office*, *Internet* und *Games* in visueller Form alle wichtigen Programme, Zugriffs- und Steuerinformationen der Windows Plattform. Dabei werden die meisten Programme über Icons gestartet und erscheinen als 2D-Fenster vor dem Hintergrund. Auch 3DNA [3DNA@] ist eine 3D-Desktop-Oberfläche, die sich vielfältig konfigurieren läßt und ebenfalls thematisch gruppierte Start- und Einsprungpunkte für diverse Windows-Applikationen bietet. Dabei wird ein videospielartiger Ansatz verfolgt, so daß sich mit dieser Software weniger effizient arbeiten läßt als mit dem raumorientierten Win3D.

Visual 3D Operating System / mUltimo3D

Der am Heinrich Hertz Institut für Nachrichtentechnik in Berlin entwickelte autostereoskopische Bildschirm bildet das Herzstück einer multimodalen GUI mit Blick-, Hand- und Sprachsteuerung im Projekt *mUltimo3D* [Multimo3D@]. In [Skerj97] werden verschiedene 3D-Anwendungen für diese Schnittstelle vorgestellt, die in der Betriebssystemoberfläche *Visual Operating System* eingebettet sind. Anwendungen werden im Raum angeordnet und können durch Blicksteuerung und gesprochene Kommandos in den Vordergrund gebracht und bedient werden [Liu00], wobei auch klassische Maus- und Tastatursteuerung unterstützt wird.

3.1.4.2 3D-Dateibrowser und -Dokumentenmanager

Die beschriebenen 3D-Betriebssystemoberflächen stellen an sich ebenfalls räumliche Datei- und Dokumentmanager dar, während die hier vorgestellten Projekte ausschließlich auf diese Aufgaben fokussiert sind. In der Abbildung 6 ist dafür jeweils ein Beispielbild zu sehen.



Abbildung 6: 3D-Dateibrowser und -Dokumentenmanager. Von links nach rechts: 3D-FSN [SGI-FSN@], 3DOSX [3DOSX@], WebForager [Card96]

SGI 3D File System Navigator und 3D Filesystembrowser für Macintosh

1992 wurde von Silicon Graphics die Metapher der dreidimensionalen Informationslandschaften eingeführt und als Beispiel der *File System Navigator (FSN)* [SGI-FSN@] für UNIX-Dateisysteme entwickelt. Verzeichnisse werden durch Zellen oder Container repräsentiert, die Dateien in Form von Quadern enthalten. Ihr Volumen repräsentiert die Größe der jeweiligen Datei oder der Unterverzeichnishierarchie. Zwischen den Zellen symbolisieren Linien die Verzeichnishierarchie, verschiedene Navigationstechniken stehen zur Verfügung.

Bei 3DOSX [3DOSX@] handelt es sich um einen 3D-Dateibrowser für das Macintosh Betriebssystem Mac OS X [Chin02]. Dabei wird direkt auf das Dateisystem zugegriffen und die volle Funktionalität des *Mac OS X finders* nachempfunden. Verzeichnisse werden als flache „Platten“ dargestellt, die ringförmig angeordnete Icons für Dateien und Unterordner enthalten. Durch Anklicken bzw. Tastaturnavigation werden Verzeichnisse ausgewählt, die dann im Vordergrund erscheinen, während andere im Hintergrund simultan zu sehen sind. Dateien und

Ordner mit ihren typischen Icons statt als Blöcke oder Container anzuzeigen, erwies sich als vorteilhaft und erleichterte den Zugang für die Benutzer [Chin02].

Web Forager und Data Mountain

1996 wurde am Xerox PARC das prototypische System *WebForager* entwickelt, wobei es sich um einen dreidimensionalen Informationsraum zur Interaktion mit dem World Wide Web handelt [Card96]. Dabei werden beim *WebForager* das *WebBook* und andere Visualisierungsobjekte in einen hierarchischen 3D-Arbeitsraum integriert, der verschiedene Ablageplätze für aktive, gerade benutzte und archivierte Informationsobjekte enthält [Card96].

Als weitere wichtige Arbeit auf diesem Gebiet ist der sogenannte *Data Mountain* [Rober98] von Microsoft Research zu erwähnen. Dieses Dokumentenmanagementsystem erlaubt es dem Nutzer, Dokumente an einer beliebigen Stelle auf einer geeigneten Ebene innerhalb einer Desktop-VR-Anwendung zu plazieren, wobei eine effiziente 2D-Interaktionstechnik zum Einsatz kommt. Eine Nutzbarkeitsstudie mit Webseiten demonstrierte dabei, daß diese 3D-Technik statistisch relevante Vorzüge gegenüber dem Favoriten-Konzept im Internet Explorer aufweist [Rober98].

3.1.4.3 Kollaborative Arbeitsumgebungen

Spin3D [Dumas99] ist eine 3D-Multiuser-Plattform für synchrones, kollaboratives Arbeiten, die von der France Telecom und dem *Laboratoire d'Informatique Fondamentale de Lille* entwickelt wurde. Unterstützt werden Treffen kleinerer Gruppen, deren durch Avatare repräsentierte Teilnehmer gleichzeitig an mehreren, um einen zentralen Arbeitsraum angeordneten Dokumenten arbeiten können. Das 3D-GUI für kooperatives Arbeiten basiert auf der Metapher eines *Virtual Workspace*. Ziel ist die Betrachtung und Manipulation dreidimensionaler Objekte auf einem virtuellen Konferenztisch. Bei *Argonaute 3D* [Argonaute3D@] (s. Abbildung 7 links) handelt es sich um ein Projekt auf der Basis von Spin3D. Physikalisch entfernten Medizinern ist es damit möglich, eine medizinische Fallbesprechung und Diskussion an einem virtuellen Patienten durchzuführen.



Abbildung 7: Kollaborative 3D-Umgebung *Argonaute 3D* [Argonaute3D@] (links) und 3D-Produktpräsentationsumgebung *Implantorium* [Kolbe98] (Mitte und rechts)

3.1.4.4 3D-Produktpräsentationen

Zu dieser Kategorie zählen rein dreidimensionale Systeme, mit denen Produkte nicht nur betrachtet und konfiguriert, sondern auch in ihrer Handhabung virtuell erprobt werden können. Ein Beispiel ist das Forschungsprojekt *Implantorium* [Kolbe98] der Hochschule für Kunst und Design Burg Giebichenstein Halle genannt. Dabei handelt es sich um ein Laboratorium für die stomatologische Implantologie. Produkte, wie OP-Werkzeuge, Aufbauten oder Implantate, können selektiert und in einem Präsentationsraum im Detail betrachtet werden. Im OP-Modus lassen sich Operationsszenarien konfigurieren, wobei im Sinne einer Lehr-/Lernanwendung Operationsschritte virtuell durchgeführt werden können [Dachs99b].

3.1.5 Vergleich von 2D- und 3D-Benutzungsoberflächen

In der letzten Dekade wurde der Nutzen von 3D-GUIs im Vergleich zu traditionellen WIMP-Interfaces sehr kontrovers diskutiert. Zwischen optimistischer Verfechtung (z.B. in [Dam92] oder [Dumas99]) und kritischer Ablehnung (z.B. [Niels98]) finden sich auch differenzierte Meinungen (z.B. [Shnei02]), die zu weiterer Forschung auf diesem Gebiet motivieren. Nach einem Überblick zu Vergleichsstudien 2D vs. 3D sollen hier deshalb wesentliche Vor- und Nachteile dreidimensionaler Benutzungsschnittstellen zusammengetragen werden.

3.1.5.1 Vergleichsstudien

In [Shnei02] sprach sich Shneiderman für eine ehrliche und differenzierte Betrachtung der Vor- und Nachteile von 3D-GUIs aus und forderte systematische, empirische Benutzbarkeitsstudien, um Anwendungsvorteile für 3D-Grafik zu ermitteln. Es existieren inzwischen mehrere Forschungsarbeiten, die sich mit der Effektivität von 2D- vs. 3D-Benutzungsoberflächen beschäftigen. Dabei wurde die Rolle des räumlichen Gedächtnisses als zentraler Vorteil für Performance untersucht. In [Ark98] wird eine zweidimensionale, ikonische Benutzungsschnittstelle mit Rasterlayout mit einer dreidimensionalen, „ökologischen“, also natürlicheren 3D-Benutzungsoberfläche verglichen. Für Objektsuch- und Akquisitionsaufgaben ergab sich eine klare Präferenz für die realistische 3D-Oberfläche mit natürlicher Anordnung gegenüber der zweidimensionalen ikonischen Schnittstelle. Es ist wichtig zu beachten, daß ein Übergang von 2D- zu 3D-Oberflächen nicht zwangsläufig die Performance bei bestimmten Aufgaben erhöht. Evaluationsstudien wie [Cockb02] fanden heraus, daß die erhöhten Freiheitsgrade in der dritten Dimension zu einer Leistungsverringerung führen können, weil Probanden diese Schnittstellen als zu unübersichtlich und ineffizient empfanden.

Für das Gebiet der 3D-Informationsvisualisierung bemerkt Schönhage, daß Benutzer mit 3D-Grafik erst vertraut werden müssen, genau wie das mit traditionellen grafischen Benutzungsschnittstellen der Fall war. Kurzzeitstudien mit unerfahrenen Nutzern genügen deshalb nicht, zumal erfahrene Computernutzer mit 3D-Visualisierungen generell besser zurecht kommen als Novizen, was verschiedene Studien, darunter [Schön01], belegen.

In einer Vergleichsstudie [Lee01] werden verschiedene Arbeiten zur Gegenüberstellung von 2D- und 3D-Oberflächen für bestimmte Aufgaben analysiert. Aus dieser Analyse und den Ergebnissen der Vergleichsstudie sollen hier einige Resultate wiedergegeben werden:

- Ziele können im klar strukturierten 3D-Raum besser und schneller gefunden werden als bei 2D-Icons in klassischen Layouts. Räumliche Wiedererkennung ist an den dreidimensionalen Raum gebunden. Eine leichte und einprägsame Navigation durch den 3D-Raum ist von entscheidender Bedeutung für die Bearbeitungsgeschwindigkeit einer Aufgabe.
- Innerhalb von 3D-Umgebungen erhöhen 3D-Icons, 3D-Vorschaubilder oder *3D-Landmarks* die Performance im Vergleich zu textuellen oder ikonischen Repräsentationen.
- Für Dokumentenmanagement-Systeme gibt es keinen statistisch signifikanten Unterschied zwischen 2D- und 3D-Fensteroberflächen, jedoch empfanden Probanden 3D-Oberflächen als attraktiver.
- 3D-Informationsvisualisierungen sind nur für bestimmte Datensätze aus bestimmten Anwendungsdomänen effizient nutzbar und somit schlecht generalisierbar.

Je nach konkreter Aufgabe, Anwendungsdomäne und eingesetzter Displaytechnologie resultierten weitere Vergleichsarbeiten in Präferenz aber auch Unterlegenheit dreidimensionaler Benutzungsoberflächen [Cockb02]. Aus diesem Umstand läßt sich schließen, daß zusätzliche grundlegende und verallgemeinernde Forschungsarbeiten notwendig sind, um allgemeingültigere Aussagen und Empfehlungen über den Einsatz solcher GUIs ableiten zu können. Einige Vor- und Nachteile lassen sich jedoch schon jetzt klar erkennen.

3.1.5.2 Nachteile von WIMP-Interfaces

- Die strikte Trennung von standardisierten (Allzweck-)Interfaces und Anwendungssemantik bei 2D-GUIs erlaubt zwar ein generalisiertes, aber für viele Anwendungen wenig intuitives Arbeiten, da anwendungsspezifische Interaktionstechniken und -elemente fehlen.
- WIMP-Anwendungen unterscheiden sich äußerlich häufig kaum voneinander und führen durch identische, rechteckige Fenster und einheitsgraue Menüsysteme leicht zur Verwirrung von Nutzern [Shnei98]. Außerdem wird durch stereotype *point-and-click* Aktionen menschliche Handlungseffizienz verschenkt. Adaptive Menüsysteme, die bestimmte Einträge je nach Nutzungsgrad aus- und einblenden, besitzen den Nachteil, die Orientierung zu erschweren und das Muskelgedächtnis des Nutzers nicht auszunutzen.
- Im 2D-Raum können große Datenmengen nicht adäquat repräsentiert werden, ebenso bieten diese keine ausreichende Möglichkeiten für realistische Visualisierungsformen und die Repräsentation bestimmter Metaphern [Dumas99].
- WIMP GUIs wurden mit ihren 2D-Widgets auch vor allem für 2D-Applikationen, wie Textverarbeitungsprogramme und Tabellenkalkulationen, entwickelt. Für diese funktionieren sie erwiesenermaßen gut, versagen jedoch häufig für 3D-Applikationen, da die Abbildung von 2D-Kontrollelementen auf 3D-Aufgaben wenig natürlich ist [Dam97].

3.1.5.3 Vorteile von 3D-GUIs

Menschen leben in einem dreidimensionalen Raum und sind darauf trainiert, sich in diesem zu orientieren, mit dreidimensionalen Gegenständen umzugehen und sich deren Positionen zu merken. Daraus leitet sich räumliches Orientierungs- und auch Handlungswissen ab, das primär auf der Leistungsfähigkeit unseres räumlichen Gedächtnisses beruht.

- 3D-Applikationen können von diesen Fähigkeiten eher Gebrauch machen als traditionelle Anwendungen [Pierc01]. Auch die Fähigkeit diffusen räumlichen Erinnerungsvermögens beim assoziativen Suchen kann bewußt genutzt werden. Somit können selbst ungenaue Erinnerungen zu schnelleren Suchergebnissen führen als die Suche eines bestimmten Menüeintrages in einem gleichaussehenden, komplexen Menüsystem. Gleiches gilt auch für die Suche eines Objektes in einem unbekanntem Raum [Dachs99b].
- Dreidimensionale virtuelle Werkzeuge (besonders mit Analogien in der realen Welt) bieten ein großes Potential, über die Gestalt und visuelle Ausprägung ihre Funktion bzw. Handhabbarkeit kommunizieren zu können.
- Für Anwendungen mit 3D-Dokumenttypen erweist sich die enge Integration von 3D-Objekten mit 3D-Schnittstellenelementen zu ihrer Bearbeitung als Vorteil, weil der Bruch zwischen 2D- und 3D-Grafik vermieden wird, der bei stereoskopischer Darstellung die Benutzungsschnittstelle sogar unbrauchbar machen kann.
- Zu den kognitiven Aspekten der Interaktion mit Benutzungsschnittstellen gehört die Aufmerksamkeitssteuerung. 3D-GUIs erlauben eine vielfältigere visuellere Differenzierung als typischerweise mit 2D-Oberflächen möglich ist, da sie Aufmerksamkeit mit den Aspekten Farbe, Form, Textur, Schattierung, Größe, Lage, Bewegung etc. hervorrufen können [BenHa99]. Der reichere Informationsraum und die erhöhten Freiheitsgrade eignen sich besser für die Präsentation mehrdimensionaler Datenmengen [Schön01] oder auch für Anpassungen an die Vorlieben von Nutzern.
- 3D-GUIs werden häufig als attraktiver und emotional anziehender empfunden. Damit erreichen dreidimensionale Anwendungen einen breiteren Nutzerkreis, zu dem auch Kinder oder technisch wenig versierte Menschen zählen.

3.1.5.4 Probleme mit 3D-GUIs

- Die Gestaltung von 3D-Räumen ist komplex und erfordert hohes Fachwissen zum sorgfältigen Umgang mit den Freiheitsgraden. Die exakte Nachbildung von Realität führt dabei nicht immer zu besseren Lösungen.
- Es existieren keine verbindlichen, übergreifenden Interaktionstechniken, sondern hauptsächlich projektspezifische Lösungen.
- Orientierungs- und Navigationsprobleme (z.B. in [Haik02] beschrieben) stellen eine Hauptschwierigkeit in virtuellen Umgebungen dar, Nutzer können leicht „verloren gehen“ [Niels98]. 3D-Welten erfordern oft vom Nutzer zu viel Anstrengung beim Navigieren und Interagieren, statt die Konzentration auf die eigentliche Arbeitsaufgabe in den Vordergrund zu rücken [Shnei02].
- 3D-Objekte, Interaktionselemente und Navigationshilfen können wichtige Teile der Szene verdecken. So warnt Nielsen davor, daß zusätzliche Kontrollelemente zur Verbesserung der Navigation dem Nutzer in den Weg geraten können und von der eigentlichen Aufgabe abhalten [Niels98].
- Die Darstellungsqualität insbesondere von 3D-Text und in der Ferne dargestellten Objekten ist qualitativ häufig nicht ausreichend.
- Trotz großer Performance-Steigerungen im Grafikbereich haben 3D-GUIs immer noch einen deutlich höheren Ressourcenverbrauch als 2D-GUIs und häufig inakzeptable Bildwiederholraten.
- Vielfältige Eingabegeräte erschweren eine Standardisierung. Bestimmte höherdimensionale Eingabegeräte (z.B. SpaceMouse) sind für 3D-Interaktionsaufgaben teils besser geeignet, aber nicht generell verfügbar.
- Bisher gibt es keine Standardisierung sowohl auf metaphorischer Seite (im Gegensatz zur Desktop-Metapher fensterbasierter 2D-GUIs) als auch auf softwaretechnischer Seite (im Gegensatz zu ausgereiften Klassenbibliotheken im 2D-Bereich, wie z.B. Microsoft Foundation Classes oder Swing).
- Den meisten Nutzern fehlt die Vertrautheit im Umgang mit 3D-GUIs. Allerdings war auch im 2D-Bereich ein langjähriger Gewöhnungs- und Lernprozeß bis zur Akzeptanz von WIMP-Interfaces notwendig. Hinzu kommt, daß 3D-Anwendungen häufig noch einen verspielten Eindruck machen und deshalb für geschäftliche Anwendungen kaum in Betracht gezogen werden [Schön01].

3.1.6 Fazit und Motivation der Systematisierung

An dieser Stelle soll die Dringlichkeit dieser grundlegenden Forschung zu 3D-GUIs und ihren Bestandteilen anhand verschiedener Auszüge aus der VR-Literatur der letzten Dekade weiter motiviert werden. 3D-Widgets und 3D-Interaktionstechniken gehören zu den bisher am wenigsten erforschten Gebieten innerhalb des Forschungsbereiches 3D-Grafik und stellen das Herzstück der 3D-GUI-Forschung dar [3DWM@]. Dabei sind gerade im Bereich der immersiven VE in den 90er Jahren mehrere Arbeiten zu verschiedenen Interaktionstechniken, Ansätzen ihrer Systematisierung und Anwendung in VR-Projekten durchgeführt worden. Im Bereich Desktop-VR, wo 3D-Widgets als intermediäre Interaktionselemente eine zentrale Rolle spielen, sind bisher nur wenige Resultate verfügbar. Meistens werden 3D-Widgets applikationsspezifisch neu entwickelt, Standards existieren so gut wie gar nicht [Geige98].

In [Snibb92] und [Dam92] wird 1992 das überraschend kleine Repertoire an existierenden 3D-Widgets angemahnt, das sich damals noch auf im Raum schwebende Menüs, Widgets zur Selektion, Translation und Rotation sowie *Cone Trees* [Rober91] und *Perspective Walls* [Macki91] zur Informationsvisualisierung beschränkte. Eine Vielzahl von Experimenten wird

noch nötig sein, um schließlich nützliche und allgemein verwendbare bzw. standardisierte 3D-Metaphern und Widgets zu etablieren [Snibb92]. Im Gegensatz zu 2D-GUIs steckt die Entwicklung von Metaphern für 3D-Benutzungsoberflächen noch in den Kinderschuhen [Conne92]. In den wissenschaftlichen Arbeiten der 90er Jahre wird mehrfach die Unreife des Forschungsgebietes erwähnt, so z.B. in [Dam92] und [Hernd94a]. In [Leine97] wird zu Recht festgestellt, daß es für 3D-Widgets keine Klassifizierungen und kaum Standardisierungsbestrebungen gibt. Weiterhin wird dort der Entwicklungsaufwand für 3D-Widgets als zu hoch eingeschätzt, die Portabilität dreidimensionaler Anwendungen jedoch als zu gering.

Auch in Arbeiten jüngerer Datums wird konstatiert, daß Forscher auf dem Gebiet der interaktiven 3D-Grafik immer noch im Stadium der Entwicklung und des Testens von Interaktionstechniken, Gesten und Metaphern für 3D Benutzungsschnittstellen sind [Myers00]. An gleicher Stelle wird vorausgesagt, daß erst eine Reihe etablierter 3D-Interaktionstechniken und 3D-Widgets vorhanden sein muß, bevor High-Level-Werkzeuge entworfen und sinnvolle 3D-Applikationen auf breiterer Basis entwickelt werden können. In [Pierc01] wird ebenfalls beklagt, daß sich trotz jahrelanger Forschung gerade 3D-Interaktionstechniken noch im Anfangsstadium befinden und noch kein Repertoire oder Lexikon von Ausdrucksmitteln und Techniken für diesen Bereich etabliert ist. Während Pierce diesen Mangel vor allem aus einer zu eng verstandenen 1:1 – Abbildung der realen auf die virtuelle Welt interpretiert, soll hier ergänzt werden, daß dafür vor allem auch die zu eng gesehene und häufig praktizierte Abbildung von 2D-Interaktionstechniken auf 3D-Applikationen verantwortlich ist. In seinem Überblicksartikel zu 3D-Interaktionstechniken weist Hand darauf hin, daß die Fortschritte bei 3D-Grafikhardware sowie bei Ein- und Ausgabegeräten bisher keine ausreichende Entsprechung im Bereich der Interaktion und Konstruktion von VE gefunden haben [Hand97].

3.2 Gestaltungsrichtlinien für 3D-Anwendungen

Der Entwurf und die Gestaltung dreidimensionaler Benutzungsschnittstellen ist aufgrund der Komplexität des Designs im dreidimensionalen Raum eine herausfordernde und schwierige Aufgabe. Designrichtlinien für WIMP-Interfaces sind dafür nur teilweise einsetzbar, so daß an dieser Stelle ein Beitrag zu künftigen Gestaltungsleitlinien für 3D-GUIs geleistet werden soll.

3.2.1 Beachtung genereller UI- und Designrichtlinien

Die meisten Grundregeln des traditionellen 2D-GUI-Designs gelten auch für 3D-GUIs. Dazu zählen beispielsweise die ergonomischen Anforderungen an Bildschirmarbeitsplätze (ISO 9241, Teil 10, 1996): *Aufgabenangemessenheit, Selbstbeschreibungsfähigkeit, Steuerbarkeit, Erwartungskonformität, Fehlertoleranz, Individualisierbarkeit* und *Lernförderlichkeit*. In [Shnei98] werden grundlegende – sich teilweise mit den ISO-Anforderungen deckende – Prinzipien zur Gestaltung interaktiver Systeme vorgestellt, darunter:

- *Konsistenz*: einheitliche Farbgebung, gleiche Button-Positionen, analoges Interaktionsverhalten, gleiche Terminologie etc.
- Informatives audiovisuelles (ggf. auch taktiles) *Feedback* in allen Phasen der Interaktion.
- *Nutzerkontrolle und Vorhersagbarkeit*: Der Nutzer kontrolliert das System, nicht umgekehrt.
- *Reduzieren der Belastung des Kurzzeitgedächtnis*, z.B. durch 7 +/- 2 gleichzeitig angebotene Optionen bzw. dargestellte Alternativen.

Für 3D-GUIs lassen sich auch Prinzipien und Richtlinien anwenden, die bei der Gestaltung multimedialer Präsentation und Webseiten zum Einsatz kommen. Zahlreiche Gestaltungsrichtlinien für diesen Bereich lassen sich z.B. in [Meißn99@] finden.

Weiterhin sollten auch verstärkt allgemeine Grundprinzipien der visuellen Gestaltung und des Kommunikationsdesigns für den Entwurf dreidimensionaler Anwendungen eingesetzt werden. Hier seien nur einige wichtige aufgeführt:

- *Visuelle Reduktion*: Zulassen von Weißräumen zur visuellen Balance, Entfernen unnötiger Dekoration und Ornamentik.
- *Gestaltungsraster*: Nutzung von Reihungs- und Gruppierungsprinzipien etc.
- *Typographie*: angemessener Einsatz von Schrift und Lesetypographie.
- *Farbharmonie*: Nutzung geeigneter Farbspektren zur Kommunikation von Emotionen, zur visuellen Unterscheidung, Hervorhebung, Verbesserung der Lesbarkeit etc.
- *Aufmerksamkeitssteuerung*: Schaffung einer visuellen Hierarchie durch Hervorhebung wichtiger Bestandteile, Einsatz von Figur-Grund-Prinzipien, unterscheidbaren Farben etc.

3.2.2 Nutzung von Metaphern

Die Rolle von Metaphern für Benutzungsschnittstellen allgemein ist bereits im Kapitel 3.1.1 behandelt worden. Ihr Einsatz für 3D-GUIs trägt ebenfalls dazu bei, den Einstieg in eine Anwendung zu erleichtern und das Prinzip des geringsten Erstaunens zu verwirklichen, indem auf die existierenden Erfahrungen der Nutzer gebaut wird. Der in der dritten Dimension mögliche Realismus verführt jedoch leicht zu stark an der Realität orientierten Metaphern, die wenig Flexibilität zulassen und eine zu hohe Erwartungskonformität voraussetzen. In [Norma88] wird deshalb darauf hingewiesen, daß für erfolgreiche Anwendungen zunächst ein gutes konzeptionelles Modell entwickelt werden muß.

Geeignete Metaphern müssen im 3D-Raum für die strukturelle Aufteilung des Raumes in *Action Spaces*, für die Navigation innerhalb der Anwendung und für Schnittstellenelemente zur

Interaktion und Manipulation von 3D-Dokumenten gefunden werden. Die Kapitel 3.3.2 und 3.4 stellen zahlreiche Metaphern für *Action Spaces* und 3D-Widgets vor. Metaphern für GUI-Elemente lassen sich nicht immer aus dem Zweidimensionalen übernehmen, weil damit die Möglichkeiten des dreidimensionalen Raumes ungenutzt bleiben und sie andererseits oft wenig dafür geeignet sind. Dreidimensionale Menüs, die mit schlecht lesbarer Schrift schräg im Raum schweben, sind ein Beispiel dafür. Vielmehr sollten Werkzeuge oder Objekte, die im Zweidimensionalen textuell aufgelistet sind, in (evtl. ikonischer) 3D-Repräsentation auszuwählen sein. Metaphern sollten bewußt neu für den 3D-Raum gesucht werden, wobei sie der realen Welt entlehnt werden, jedoch auch der Imagination entspringen und somit „magischen“ Charakter haben können. Literatur, Kunst und Filme bieten dafür zahlreiche Anregungen, denkt man z.B. an fliegende Teppiche in Märchen, Teleportierung in Science Fiction Filmen oder die – technisch gesprochen – nicht-uniforme Skalierung von Personen und Gegenständen im Buch „Alice im Wunderland“. Daß diese Metaphern auch in Forschungs- und industriellen Projekten umgesetzt worden sind, zeigen verschiedene VR-Themenparkattraktionen in Disneyland [Mine03] oder die sogenannten *Worlds in Miniature* als in der Hand gehaltene 3D-Minilandkarten inmitten von VR-Szenen [Stoak95].

3.2.3 Keine 1:1 – Abbildung der Realität, mehr Magie

Neben dem gerade erwähnten „magischen“ Charakter von Metaphern, der sich nicht für alle Anwendungen eignet, können aber auch bekannte Applikationen und Metaphern von weniger Realismus profitieren. Erste 3D-Shoppinglösungen für das Internet wurden z.B. zunächst in Form virtueller Supermärkte vorgestellt, die man durchwandern konnte und in deren Regalen jedes Produkt wie in einem realen Laden in mehrfacher Ausführung präsentiert wurde. Inzwischen sind diese Anwendungen beinahe komplett aus dem Web verschwunden, da die übermäßige Nachahmung von Realität neben Problemen bei der Navigation und der unnötigen Mehrfachpräsentation identischer Produkte keinerlei Mehrwert ergeben hatte. Aus diesem Beispiel leitet sich die wichtige Empfehlung ab, eine Balance zwischen Nachahmung von Realität (soviel wie nötig, um eine Metapher glaubhaft zu vermitteln) und dem Potential der computergenerierten 3D-Welten zu schaffen, in denen mit tradierten Vorstellungen auch gebrochen werden kann.

Von mehreren Forschern wurde der Aspekt der Nachbildung von Realität für erfolgreiche VR-Anwendungen in den 90er Jahren überbetont. Dabei machen mehrere Jahrhunderte der Kunstausübung und der Gestaltung von Landkarten oder anatomischen Atlanten deutlich, daß z.B. 2D-Grafiken und nicht-photorealistische Darstellungen wichtige Vereinfachungen, Abstraktionen und für Betrachter verständlichere Abbildungen der Realität erlauben [Deuss01]. Statt die Realität zu kopieren, ist es also wichtig, die besonderen Möglichkeiten interaktiver 3D-Grafik kreativ zu nutzen. So sprachen sich u.a. Shneiderman in [Shnei02] und Pierce in [Pierc01] gegen die Schaffung glaubwürdiger Realität aus und für Abstraktionen, Nicht-Linearitäten und „*something better than reality*“. Zum „Repertoire“ des virtuellen Raumes zählen außerdem Transparenz, multiple Maßstäbe, perspektivische oder hyperbolische Verzerrung und ähnliche Möglichkeiten. In [Pierc01] wird auf die Notwendigkeit und Chance hingewiesen, die Freiheitsgrade des computergenerierten virtuellen Raumes zu nutzen und bekannte Vorstellungen nicht 1:1 nachzubilden. Er stellt dabei interessante Annahmen auf, die für die Realität nicht gelten, jedoch für 3D-Applikationen zur Verfügung stehen.

3.2.4 Integration von 3D-Dokumenten und 3D-Widgets

Bei 3D-GUIs sollte ein 2D-/3D-Bruch zwischen Schnittstellen und bearbeiteten Dokumenten vermieden und darauf geachtet werden, daß Interaktionselemente möglichst auch dreidimensional und in den Raum integriert sind. So wird in [Thoma99] die Designempfehlung gegeben, Nutzer nicht dazu zu zwingen, in 3D-Umgebungen 2D-Interfaces zu verwenden. Es ist

eine natürliche Konsequenz, bei von Natur aus dreidimensionalen Daten die Schnittstellenelemente zu ihrer direkten Manipulation ebenfalls in die dritte Dimension zu übertragen und somit eine homogene Umgebung, intuitivere Manipulation und unmittelbare Feedbacks zu erlauben. Beispielsweise ist einer dreidimensionalen Objektmenüausprägung, z.B. einem Ringmenü, in den meisten Anwendungsfällen der Vorzug gegenüber textuellen, im Raum schwebenden 2D-Menüs zu geben. Die in [Dachs99b] geforderte Kopplung von Schnittstellenelementen und dargestellten Gegenständen in einem gemeinsamen 3D-Raum entspricht auch viel eher einer aus dem Alltagsleben bekannten situativen Bindung von Werkzeugen an den konkreten Kontext eines dreidimensionalen Raumes.

3.2.5 Produktsprachlicher Ansatz

Da Software rein informationellen Charakter hat, fehlt ihr das sensomotorische Gestalt- und Aktionspotential materieller Produkte, insbesondere deren Anschaulichkeit und Handhabbarkeit, wodurch sie von stereotyper Mausclick-Funktionalität dominiert sind [Kolbe95]. Während reale Produkte und Werkzeuge ihre Funktionalität in Form von Materialeigenschaften und Formausprägungen sichtbar machen, erschweren die gegenwärtigen uniformen Softwareoberflächen mit ihrer tabellarischen Ansammlung von Hunderten ähnlich aussehender Menüpunkte das Ablesen und unmittelbare Erleben von Funktionalität. Einen Großteil der Zeit verbringen insbesondere unerfahrene Nutzer mit dem Suchen in Menüsystemen, wobei Menschen sich letztlich die Positionen von Menüeinträgen einprägen. Im realen täglichen Umfeld sind unsere Suchräume viel komplexer, und unser Suchen nach Objekten und Werkzeugen ist viel effizienter, da nicht nur positions-, sondern auch gestaltorientiert.

Die dritte Dimension besitzt jedoch ein inhärentes Potential, Softwareoberflächen einen *produktsprachlichen Charakter* zu geben. Man sollte nicht erst eine Beschriftung lesen müssen, um die Funktion eines 3D-Widgets zu verstehen [Hernd94a]. Statt dessen sollten Widgets diese durch Form und Erscheinung kommunizieren [Rober93a]. Dabei ist trotz der größeren Möglichkeiten im 3D-Raum nicht unbedingt Realismus anzustreben, weil weniger Details im Design oft mehr sind und eine Vereinfachung der Werkzeuggestalt im ikonographischen Sinne der Interaktionsaufgabe nützt. Schließlich kann das bekannte, bereits 1896 vom amerikanischen Architekten Louis H. Sullivan geäußerte Credo „*form follows function*“ auch auf 3D-GUIs angewendet werden.

Auch wenn es sich bei der Gestaltung von Benutzungsoberflächen primär um Informationsdesign und nicht mehr um klassisches Produktdesign handelt, lassen sich Erkenntnisse aus dem Gebrauch von realen Werkzeugen auch in den virtuellen 3D-Raum übertragen, wobei Erfahrungen aus dem Gebiet des Industriedesigns einfließen [Dachs99b]. Viele der Forderungen, die an ein reales Erzeugnis oder den Prozeß seiner Nutzung gestellt werden, lassen sich auch auf virtuelle Produkte anwenden. Dazu zählen z.B. die sinnfällige Anordnung und ergonomisch günstige Gestaltung der wesentlichen Bedienelemente, die eindeutige Ablesbarkeit der Verwendungsart oder die standardisierte Erscheinungsform innerhalb eines Bereiches. In [Somme98] wird sogar von einer nötigen Konvergenz von Produktdesign und Kommunikationsdesign gesprochen.

Daß ein produktsprachlicher Ansatz erfolgversprechend ist, zeigen beispielsweise die Arbeiten von Serra u.a. zur Interaktion des Nutzers mit medizinischen Volumendaten an einer *Workbench*, für die in [Serra95] ein natürliches Schnittstellenparadigma vorgeschlagen wird. Dabei weist die virtuelle Repräsentation eines in der Hand gehaltenen Stiftes Werkzeugcharakter auf, z.B. mit Scherensymbol zum Abtrennen oder in Messerform für Schnitte. Durch diese sehr anschauliche Präsentation der Funktion an der visuellen Werkzeugoberfläche wird ein sehr produktives Arbeiten ermöglicht.

3.2.6 Reduktion im Szenenlayout, visuelle Klarheit und Bildschirmeffektivität

Während 2D-Layouts sich auf einen fixen Betrachtungsstandpunkt verlassen können, ist dies für 3D-Szenen nicht der Fall, da hier kontinuierliche Perspektiv- und Blickpunktänderungen stattfinden können und ein statisches Layout somit selten möglich ist. Jedoch bietet das Konzept der *Action Spaces* hier Abhilfe, indem konkrete vordefinierte Blickpunkte eingesetzt werden.

Der im Kommunikationsdesign geforderten visuellen Klarheit und Reduktion der Stilmittel kommt eine besondere Bedeutung zu, damit überladene 3D-GUIs, visuelle Unordnung und unbeabsichtigte Objektverdeckungen vermieden werden. Der Architekt Ludwig Mies van der Rohe prägte die bekannte Richtlinie „*less is more*“, womit Verzicht auf Ornamente und der sparsame Umgang mit gestalterischen Mitteln, wie Farben und Formen, gemeint sind. Gerade die vielfältigen visuellen Parameter (s. Abschnitt 3.1.5.3) des 3D-Raumes verführen häufig zu überladenen 3D-Szenen. Einige konkrete Empfehlungen sollen hier gegeben werden:

- Verbesserte Aufmerksamkeitssteuerung durch Begrenzung der Zahl gleichzeitig sichtbarer Elemente (7 +/- 2 – Regel beachten), Einsatz von Kontrasten (Figur/Grund, groß/klein, Komplementärfarben etc.), Nutzung von Transparenz (z.B., damit Widgets Objekte nicht verdecken), durch visuelle Staffelung, Ausnutzung von Perspektive und Raumtiefe sowie durch Gruppierung zusammengehöriger Informationen, Optionen und Werkzeuge (z.B. in *Action Spaces*).
- Reduzierung notwendiger Navigations- und Interaktionsschritte auf ein Minimum; immer daran denken, das Problem selbst zu lösen [Shnei02]. Zum Beispiel können Werkzeuge auf Anforderung eingeblendet werden, statt erst an anderer Stelle in der 3D-Szene aufgegriffen und „mitgenommen“ werden zu müssen.
- Widgets sollten so angeordnet sein, daß sie nicht von der Aufgabe ablenken, schnell zur Hand sind [Kaur98] und harmonisch in den Raum integriert werden. Untersuchungen ergaben, daß die „ökologische“ Anordnung realistischer 3D-Objekte im Raum, also ihre natürliche Platzierung (z.B. Bücher in einem Regal) vs. geometrisches Rasterlayout die Performance von Objektsuchaufgaben signifikant erhöhten [Ark98].
- Schaffung visuell abgeschlossener Räume, die Orientierung und das Gefühl von Vollständigkeit und Überblick vermitteln. In Zusammenhang mit blickpunktabhängiger Navigation läßt sich so ein Orientierungsgefühl aufbauen und der Eindruck vermitteln, nichts übersehen zu haben.
- Realisierung geeigneter Größenverhältnisse und Proportionen zwischen Raumelementen, 3D-Dokumenten und 3D-Widgets unter Beachtung perspektivischer Größenänderungen. So müssen aktivierbare 3D-Widgets so groß gestaltet werden, daß sie auch im Hintergrund noch bedienbar sind. Weiterhin sollte die Größe von Widgets auch an die systemtechnischen Gegebenheiten angepaßt werden, also z.B. einbezogen werden, ob sie mit einem *6-DOF-Tracker* oder einer Maus bedient werden sollen.
- Bei Desktop-VR-Anwendungen und vordefinierten Blickpunkten innerhalb der Szenen sollte bildschirmeffektiv gearbeitet und somit möglichst wenig Bildschirmplatz verschwendet werden [Shnei02]. Das heißt jedoch nicht, Bildschirme zu überfüllen, sondern nur aufgabenbestimmte Informationen und Werkzeuge anzuzeigen, also solche, die für eine konkrete Handlung in einer bestimmten Situation relevant sind. Neben der Reduzierung der Informationsmenge bedeutet Bildschirmeffektivität auch die geschickte Ausnutzung des ohnehin begrenzten Raumes, so z.B. die bewußte Nutzung des Querformates von Bildschirmen.

3.2.7 Vielfältige Medienintegration

Der dreidimensionale Raum ermöglicht – im Gegensatz zu Einzelmedien – die Integration und dynamische Änderung sämtlicher Medien. Dieses Potential sollte kreativ genutzt werden, um z.B. das von VR-Anwendungen erreichte Immersionsgefühl auch bei Desktop-VR-Anwendungen zu ermöglichen. Im Bereich der Spieleprogrammierung gibt es erfolgreiche Beispiele dafür, wie mit einfachen audiovisuellen Hinweisen Immersion und dadurch Bindung an die Anwendung (= das Spiel) erzeugt werden kann. Für kommerzielle Anwendungen, z.B. 3D-Produktpräsentationen, tragen Medien entscheidend zur Schaffung von Emotionalität und Kundenbindung bei ([Dachs99b], [Templ99b]). Einige Empfehlungen dazu:

- Auditive Aspekte, insbesondere Raumklang, sollten verstärkt in 3D-Anwendungen integriert werden, da sie das Immersionsgefühl verstärken und wertvolle Hinweise auf Raumpositionen sowie Interaktionshilfen bieten. In [Hoffm01] wird der Nutzen von Raumklang insbesondere für Benutzungsschnittstellen jenseits von reinen Feedbackgeräuschen ausführlich beschrieben. In [Meißn99@] wird neben anderen Empfehlungen zum Einsatz von Geräuschen und Klängen auch erläutert, wie Sound z.B. andere, möglicherweise aufwendigere Medien ergänzen oder sogar substituieren kann.
- Texturen und Videos können als Ersatz für komplexere Objektgeometrien im Sinne einer Level-of-Detail-Technik eingesetzt werden.
- Der sinnvolle Einsatz von Texturen kann eine enorme Steigerung des Realismus- und Immersionsgefühls zur Folge haben. Neben detaillierten Texturen bieten sich hier auch *Environment Textures* für metallische Oberflächeneffekte, Schatten- und Lichttexturen zur Simulation aufwendiger Beleuchtungseffekte, interaktives Bump-Mapping zur Simulation von Oberflächenrauigkeiten und natürlich der konsequente Einsatz von Mehrfachtexturen an. Texturen mit transparenten Bereichen und Videotexturen auf einfachen Objektgeometrien können zudem deutlich komplexere Objekte glaubhaft vortäuschen.
- Mit Hintergrundbildern bzw. –panoramen, ortstypischen Geräuschen und kurzen Videoclips (z.B. am Himmel treibende Wolken) läßt sich sehr einfach eine bestimmte Atmosphäre für eine 3D-Anwendung erzeugen.
- Bei der Verwendung von Text im 3D-Raum ist Vorsicht geboten. Dies hängt einerseits mit der nach wie vor ungenügenden Darstellungsqualität von 3D-Text – häufig sogar noch ohne Anti-Aliasing – zusammen. Andererseits sind perspektivisch verzerrte Schriften nur schlecht wahrzunehmen und im Sinne der Lesetypographie tabu.
- Multimediale Elemente sollten immer sparsam und zweckorientiert eingesetzt werden. Auch hier ist weniger oft mehr. Eine Überfrachtung des Bildschirms und damit der Wahrnehmung des Nutzers mit komplexen Animationen, Videos und Sounds lenkt von der Interaktionsaufgabe ab, kann Unzufriedenheit beim Nutzer erzeugen und beansprucht unnötige Systemressourcen.

3.2.8 Ästhetische Qualität, Interface-Identität und Benutzungsfreude

Es besteht Einigkeit darüber, daß nicht nur die ästhetische Qualität von Alltagsprodukten, sondern auch grafischen Benutzungsschnittstellen ein wesentlicher Faktor für Akzeptanz, Benutzbarkeit und Nutzerzufriedenheit darstellt. Trotzdem sahen die 2D-GUIs sehr unterschiedlicher Anwendungen jahrelang sehr ähnlich aus. Erst in den letzten Jahren haben sich sogenannte *Skins* als frei wählbare Gestaltungsvarianten für Schnittstellen von Softwareprodukten insbesondere aus der Unterhaltungs- und Medienbranche (Medienplayer, Schnittsysteme etc.) – aber auch für Betriebssystemoberflächen – etabliert. Grundlegende Interaktionsprinzipien und die Anordnung von Interaktionselementen bleiben trotz des veränderten Erscheinungsbildes dabei gewahrt. Die Individualität und Alleinstellungsmerkmale eines Softwareproduktes äußern sich auch wesentlich über sein Aussehen.

Im Gegensatz zu uniformen, standardisierten Schnittstellen soll hier deshalb in Anlehnung an die *Corporate Identity* eines Unternehmens für die Umsetzung von *Interface Identity* [Dachs99b] plädiert werden, einem charakteristischen gestalterischen Erscheinungsbild der Benutzungsoberfläche als Mittel des Corporate Designs eines Unternehmens. Dabei ermöglichen typische Farben, Formen, Schriften, grafische Symbole und multimediale Stilelemente die Identifizierbarkeit und Wiedererkennbarkeit eines Interfaces. Dafür ist eine Balance zwischen Konsistenz in der Bedienung, unverwechselbarer Charakteristik und intelligenten visuellen Freiheitsgraden nötig. Konzeptionelle Konstanten garantieren die leichte Bedienbarkeit (ein Schieberegler muß als solcher erkennbar und immer gleich zu bedienen sein), visuelle Variablen eine gestalterische Flexibilität (Schieberegler müssen nicht alle grau und rechteckig sein).

Freude an der Benutzung (*joy of use*), Emotionalität und Attraktivität auch jenseits der aufgabenbezogenen Benutzbarkeit, also z.B. durch „Exklusivität“ oder „Originalität“ von GUIs werden auch in [Burme02] als wichtiges Kriterium eingefordert. Dies spielt nicht nur im Marketingbereich eine Rolle, wo z.B. eine Produktpräsentation im Internet auch die Sinne des Nutzers ansprechen soll, sondern verbessert auch bei traditioneller Software die Akzeptanz, Arbeitsqualität und Motivation der Nutzer. In [Burme02] wird anschaulich dargestellt, wie neben dem pragmatischen Aspekt der Usability auch die hedonistische Qualität einer Benutzungsoberfläche von entscheidender Bedeutung ist. 3D-Grafik in Verbindung mit anderen Medien bietet hier ein überragendes Potential, um von einer reinen *Nutzungsqualität* einer Software zu tatsächlicher *Erlebnisqualität* zu gelangen.

3.2.9 Optimierung und Schonung von Ressourcen

Trotz hoher verfügbarer Grafik-Performance sind 3D-Anwendungen nach wie vor so zu entwickeln, daß sie ressourcenschonend arbeiten, was Polygonzahl, Texturgrößen, Einsatz von Videos, Verwendung von Lichtquellen, Level-of-Detail-Stufen etc. betrifft. Die Benutzbarkeit eines 3D-GUIs steht und fällt mit der erzielten Framerate, ab der ein echt interaktives Arbeiten möglich ist. Die für frühere VR-Anwendungen angegebene Mindestframerate von 10 fps genügt dabei den heutigen Anforderungen an eine ergonomische interaktive Benutzungsschnittstelle oft nicht mehr. In [Leine97] wird z.B. eine Minimierung des Darstellungsaufwandes für 3D-Interaktionselemente zugunsten der 3D-Dokumente gefordert, um Rechenleistung zu sparen. So sollte bei 3D-Widgets nicht nur aus Gründen der Gestaltung, Aufgabenangemessenheit und visuellen Klarheit auf unnötige und hyperrealistische Details und zu feine (z.B. abgerundete) Geometrien verzichtet werden. Die Optimierung von 3D-Objekten für interaktive 3D-Anwendungen beginnt bereits beim 3D-Modeling, muß weiterhin bei den Exportoptionen und schließlich innerhalb des konkreten Echtzeitgrafikformates beachtet werden. So werden z.B. in [Meißn03@] konkrete Optimierungsmöglichkeiten für VRML97 vorgestellt. Trotz einiger existierender Werkzeuge sind hier vorrangig „Handarbeit“ und Erfahrungen mit konkreten Formaten gefragt.

3.2.10 Interdisziplinäre Entwicklung

Im Zeitalter der hochgradigen Arbeitsteilung ist es ausgeschlossen, sämtliche Aufgaben bei der Erstellung eines Softwareproduktes in Personalunion abzuarbeiten. Auch ist das Ergebnis solcher Bemühungen fast immer von geringerer Qualität als ein Produkt, welches in interdisziplinärer Teamarbeit entstanden ist. Gerade die Wichtigkeit gestalterischer Anforderungen neben denen funktioneller und technischer Natur wird immer noch weitestgehend ignoriert. Obwohl die interdisziplinäre Entwicklung von Benutzungsschnittstellen und insbesondere VR-Applikationen bereits mehrfach eingefordert und diskutiert wurde (z.B. in [Encar00] und [Dam00]), ist sie nach wie vor nicht selbstverständlich. Experten aus mehreren Fachgebieten, z.B. Informatik, Industrie- und Kommunikationsdesign, Architektur, Ergonomie, Psychologie,

Akustik, Tontechnik etc., die üblicherweise bei Multimediaproduktionen schon in Teams zu finden sind, werden in 3D-Anwendungen aufgrund der Komplexität der dritten Dimension, des großen Gestaltungsspielraumes und der vielfältigen Medienintegration um so dringender benötigt. Hier ist auch ein deutlicher Unterschied zu traditionellen Softwareproduktionen zu sehen.

Gerade die Umsetzung der zuvor aufgestellten Gestaltungsrichtlinien erfordert Fachwissen jenseits der Informatik. Neben der Anwendung von elementaren Gestaltungsregeln und -prinzipien können z.B. 3D-GUI-Entwicklungen von Erfahrungen der Produktdesigner und Architekten mit dem dreidimensionalen Raum und seiner Gestaltung profitieren. Alle künstlerischen Bereiche können überdies einen Beitrag zu einem – nicht immer am Machbaren orientierten, wohl aber inspirierenden und frischen – kreativen Denken und Entdecken von neuen Metaphern und Interaktionstechniken leisten.

3.2.11 Spezielle Richtlinien für 3D-Widgets

Da 3D-Widgets als Interaktionselementen einer 3D-Anwendung eine besondere Bedeutung zukommt, werden hier einige konkrete Richtlinien für ihren Entwurf aufgeführt:

- Ein Widget sollte möglichst selbsterklärend sein und seine Funktionalität klar kommunizieren (*Self-disclosure*) [Snibb92]. Dazu muß es eine adäquate Geometrie und Erscheinung besitzen (sogenannte *Affordances* [Norma88]), die Rückschlüsse auf seine Handhabung erlauben [Hernd94b]. Insbesondere die bedienbaren Teile eines Widgets sollten sich klar vom Rest abheben [Kaur98].
- Die Geometrie eines Widgets darf keine wichtigen Teile des zu manipulierenden Objektes verbergen [Snibb92], sollte also bei direkter Plazierung am Objekt möglichst unauffällig sein, was im Widerspruch zum vorherigen Punkt stehen kann [Hernd94b]. Sie dürfen nicht zu komplex sein, weil dies sich auch negativ auf das Erlernen der Interaktion, die Bedienbarkeit sowie die Grafikleistung auswirken kann. Widgets sollten zudem von allen möglichen Blickpunkten klar unterscheidbar und identifizierbar sein [Kaur98]. Solche *omniviewable widgets* sind insbesondere auch für *Volumetric Displays* von Bedeutung [Balak01].
- Während der Manipulation sollte ein Widget *Feedback* [Norma88] über seinen Zustand und seine manipulierbaren Freiheitsgrade zur Verfügung stellen [Hernd94b]. Das Resultat der Manipulation sollte im Sinne eines 3D-WYSIWYG unmittelbar sichtbar sein. In [Barri01] werden grundlegende Feedback-Varianten für den 3D-Raum vorgestellt, dazu visuelle Attribute des Designs von Feedback-Elementen.
- Beschränkungen der Freiheitsgrade eines Widgets, *Constraints*, erleichtern die Manipulation und Aufgabenerfüllung [Snibb92]. Operationen mit zu vielen Freiheitsgraden sollten dekomponiert und auf niederdimensionale Teilaufgaben abgebildet werden [Zelez97]. Zu Constraints lassen sich auch Positionierungshilfen, die Ausrichtung an Linien oder Gittern sowie das Einschnappen an bestimmten Raumpositionen zählen. Sollen höherdimensionale Eingabegeräte eingesetzt werden, müssen diese eventuell ebenfalls beschränkt und mit den verwendeten Widgets koordiniert werden [Hernd94b].

Diese vier Anforderungen werden in beispielhafter Weise durch die im Open Inventor Toolkit [Strau92] zur Verfügung stehenden Manipulatoren berücksichtigt. In Abbildung 8 ist unter (a) der sogenannte *Universal Transformer* zu sehen, der das zu manipulierende 3D-Objekt nur minimal verdeckt und Griffe, sogenannte *Handles*, zur Manipulation bereitstellt. Wählt man die grüne Kugel oben, erhält man als unmittelbares Feedback die Einblendung zweier Rotationskreise (b). Wählt man den grauen Anfasser oben vorn, werden Linien zur Visualisierung der nun möglichen uniformen Skalierung eingeblendet (c). Rechts im Bild (d) ist das Feedback für die Translation des Objektes in der gelb markierten Ebene zu erkennen.

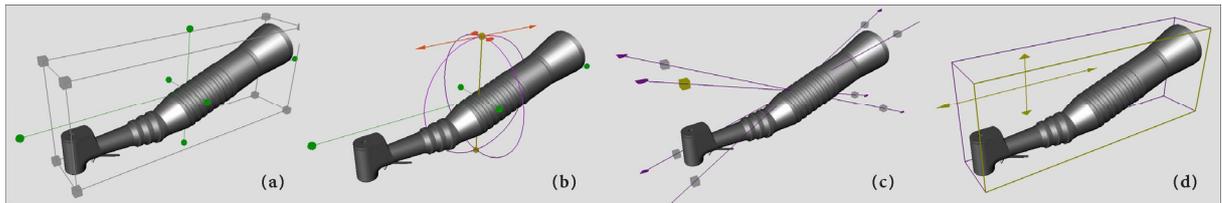


Abbildung 8: Open Inventor Manipulator *Universal Transformer* in verschiedenen Interaktionszuständen

- Die Aufgaben- und Zielgruppenangemessenheit eines Widgets spielt eine wichtige Rolle für dessen Entwurf [Snibb92]. So ist es ein Unterschied, ob beispielsweise exakte numerische Werte eingegeben werden müssen oder nur eine visuell zufriedenstellende Manipulation erfolgen soll. Der in Abbildung 8 gezeigte *Universal Transformer* erlaubt somit zwar eine sehr schnelle geometrische Manipulation, jedoch keine präzisen Werteangaben.
- Widgets müssen eine Balance zwischen Aufgabenbezogenheit und Allgemeingültigkeit eingehen. Hochspezialisierte Widgets, z.B. zur Manipulation von Stromlinien im Raum, stehen sehr allgemeinen, wie z.B. Schieberegler, gegenüber.
- Die flexible Anpassung der Parameter von 3D-Widget-Komponenten (z.B. Teilen der Geometrie und des Erscheinungsbildes) erlaubt jedoch einen Kompromiß, der einen bestimmten Widget-Typ immer als solchen erkennbar sein läßt und trotzdem an einen konkreten Aufgabenkontext gebunden ist [Dachs99a]. Zudem ist die Konfigurierbarkeit von Widgets innerhalb bestimmter Grenzen auch deswegen so wichtig, weil etablierte Standards auf diesem Gebiet noch nicht existieren und daher weitere Experimente nötig sind [Hand97].

Die Abbildung 9 zeigt als Beispiel für die letztgenannte Empfehlung einen Doppelschieberegler, der zur Steuerung der Transparenz von Objektbestandteilen dient [Dachs97]. Die Kugel im rechten Teil des Schiebereglers symbolisiert das gewählte 3D-Objekt und erlaubt dessen schrittweise Transparenzschaltung bis hin zur Unsichtbarkeit (linkes Bild). Die Schale im linken Teil symbolisiert alle restlichen, nicht ausgewählten Objekte, die sich ebenfalls schrittweise transparent schalten lassen bis hin zur Isolation des Einzelteiles (rechtes Bild). Im Gegensatz zu Allzweckschieberegler aus 3D-Widget-Bibliotheken, die gewöhnlich wie extrudierte 2D-Schieberegler oder eine Zusammenstellung geometrischer Primitive aussehen, ist dieser Transparenzregler in die Hemisphäre des Anwendungsraumes integriert und kommuniziert seine Funktion durch die Zunahme an Transparenz zu den Enden hin.



Abbildung 9: 3D-Schieberegler zur Transparenzschaltung von Objektteilen [Dachs97]

3.3 Action Spaces: Konzept und Metaphern

Action Spaces (AS) wurden bereits im Abschnitt 3.1.3.1 als blickpunktabhängige virtuelle Räume definiert, in denen 3D-Schnittstellenelemente zur Erfüllung einer Aufgabe angeordnet sind. Da Action Spaces neben 3D-Widgets ein wesentlicher Bestandteil von 3D-GUIs sind, sollen hier einige wichtige Ergebnisse aus [Dachs00] zusammengefaßt dargestellt werden. Action Spaces dienen einerseits der Strukturierung des virtuellen Raumes in verschiedene aufgabenorientierte Zonen, die realen Arbeitsräumen nachempfunden sind und zwischen denen Nutzer einfach navigieren können. Andererseits lösen sie das Problem der Aufteilung und Anordnung von 3D-Interaktionselementen, indem sie Raumstrukturen für ihre Integration bereitstellen. Sowohl die effektive Aufteilung des Bildschirms durch vordefinierte Anordnungen und Blickpunkte als auch die vereinfachte Navigation zwischen einzelnen Bereichen der 3D-Anwendung lösen wichtige Probleme gerade für Desktop-VR-Anwendungen.

3.3.1 Verwandte Arbeiten

Vorläufer dieses Konzeptes sind in zweidimensionalen GUIs zu sehen, für die multiple, aufgabenorientierte Desktops (*Virtual Workspaces*) zur Lösung des Platzproblems vorgestellt wurden. Ein Beispiel dafür ist das Ende der 80er Jahre im Xerox PARC entwickelte Rooms-Konzept für fensterbasierte GUIs [Hende86]. Anfang der 90er Jahre wurde dieses Idee im *Information Visualizer* ([Card91], [Rober93a]) auf *3D Rooms* als Metapher für Informationsräume zur animierten 3D-Informationsvisualisierung erweitert. Die für intuitive Handhabung bekannten Bildbearbeitungswerkzeuge der früheren Firma MetaCreations, z.B. die *PhotoFactory* [ScanSoft@], übernahmen das Raumkonzept und machten es für Alltagsnutzer populär. In verschiedenen 2.5D-Handlungsräumen wie „Sortieren“, „Bearbeiten“ etc. lassen sich Fotos mit realistisch anmutenden Werkzeugen bearbeiten.

Für das Problem des Ablegens und Wiederauffindens von 3D-Widgets und -Objekten sowie den Zugriff darauf innerhalb von Desktop-VR-Umgebungen wurden *Toolspaces & Glances* entwickelt [Pierc99]. Toolspaces sind Speicherorte für 3D-Objekte, die dem virtuellen Körper des Nutzers zugeordnet sind, und somit zwar – wie ein Rucksack – erreichbar, aber nicht sichtbar sind, bis sie gebraucht werden. Da Toolspaces Orte sind, wo nicht nur Objekte abgelegt, sondern auch Widgets aufgabenspezifisch angeordnet und manipuliert werden, ähneln sie dem hier vorgestellten Konzept, sind jedoch auf die aktuelle Nutzerposition bezogen.

Diese wie auch vergleichbare Arbeiten machen jedoch keine Vorschläge für Metaphern zur eigentlichen Strukturierung des Raumes. Im 1995 begonnenen IBM-Projekt *RealPlaces* [RealPlaces@] zur Untersuchung der Konvergenz von Geschäftsanwendungen und VR wird eine virtuelle Welt in Plätze eingeteilt, in denen (Interaktions-)Objekte jeweils strukturiert abgelegt sind. Diese Objekte können an speziellen Stellen im Sinne eines bestimmten Aufgabenkontextes gruppiert werden. Insofern ähnelt dieser Ansatz dem hier vorgestellten, wobei der Fokus auf einer realistisch anmutenden Welt liegt, durch die sich der Nutzer frei oder mit Punkt-zu-Punkt-Navigation bewegen kann.

Pierce betrachtet in seiner Dissertation ([Pierc01], Kap. 4 „*Places and Landmarks*“) zahlreiche Navigationstechniken zwischen verschiedenen Räumen einer virtuellen Anwendung, darunter auch Teleportierung und Punkt-zu-Punkt-Navigation. Er betont weiterhin die Notwendigkeit, virtuellen Welten, die nicht schon eine inhärente Strukturierung in verschiedene Räume bzw. Plätze besitzen, ein verständliches raumgliederndes Prinzip zu geben, um Nutzern den Aufbau eines mentalen Modells zu erleichtern. Die in [Pierc01] entwickelte Lösung in Form von *Place Hierarchies* bezieht sich jedoch nur auf immersive Umgebungen, die natürliche oder Fantasiewelten nachbilden.

3.3.2 Konzept und mentales Modell

Basis des Konzeptes sind die zahlreichen spezialisierten Arbeitsorte, die der Mensch sich für seine Aktivitäten geschaffen hat. In Labors, Werkstätten, Küchen, Büros, Tonstudios etc. können die jeweiligen Aufgaben unter anderem deshalb so effizient bearbeitet werden, weil benötigte Werkzeuge in optimaler Weise räumlich angeordnet sind. Dabei lassen sich vier abstrakte Ebenen identifizieren: 1. die metaphorische und visuelle Hauptstruktur (z.B. Haus), 2. die eigentlichen Handlungsräume (z.B. Werkstatt), 3. die Werkzeuge und ihre Ablageorte (z.B. Bohrer und Werkbank) und schließlich 4. die zu bearbeitenden Objekte (z.B. Werkstück). Wendet man dieses Modell auf 3D-GUIs an, benötigt man also für Action Spaces (Ebene 2) einen räumlichen Rahmen (Ebene 1) zur Orientierung. 3D-Dokumente (Ebene 4) werden schließlich mit bestimmten Werkzeugen (Ebene 3) bearbeitet.

Im Gegensatz zu den meisten realen Räumen und einigen Konzepten verwandter Arbeiten müssen Action Spaces keine (abgeschlossenen) Räume im geometrischen Sinne sein. Vielmehr sind sie durch konkrete Blickpunkte definiert, von denen aus man Werkzeuge und ihre möglichen Container bzw. Ablageorte zur Erfüllung bestimmter Aufgaben erreichen kann. Durch die Vordefinition der Plätze und *Constraints*, die Änderungen nur in bestimmten Grenzen zulassen, wird visuelle Unordnung vermieden und das Ortsgedächtnis von Nutzern verwendet. Da in Action Spaces nicht immer alle benötigten bzw. alle denkbaren 3D-Widgets gleichzeitig sichtbar sein können, ist es z.B. unter Nutzung des *Toolspaces*-Konzeptes [Pierc99] möglich, weitere Werkzeuge von temporär sichtbaren Paletten einzublenden.

Typische 3D-Anwendungen besitzen mehrere Action Spaces für verschiedene Aufgaben. Somit stellt sich das Problem der Navigation zwischen diesen, das ohne weiteres nicht – wie in immersiven Umgebungen häufig üblich – durch freie Nutzernavigation, z.B. *Gehen* oder *Fliegen*, gelöst werden kann. Um Orientierungsschwierigkeiten zu vermeiden und einen schnellen Wechsel zu ermöglichen, wird eine aufgabenzentrierte Navigation durch direktes Umschalten vorgeschlagen. Dies kann ausgelöst werden durch:

- 3D-Widgets, die den nächsten AS bzw. die Richtung dorthin symbolisieren,
- miniaturisierte Ansichten der anderen AS bzw. der Gesamtstruktur,
- sichtbare Teile der benachbarten AS (wie in Detail-und-Kontext-Techniken üblich),
- Nutzung von Spracheingabe, Gesten oder anderen Interaktionsformen.

Der Übergang zwischen einzelnen Räumen sollte animiert werden, um dem Nutzer die Orientierung zu erleichtern. Beim Wechsel zwischen zwei Action Spaces ist es möglich, daß sich der Blickpunkt des Nutzers gar nicht ändert, sondern nur der sichtbare Ausschnitt der Szene vor ihm. Dies ist z.B. bei der Theater-Metapher (s.u.) der Fall.

3.3.3 Metaphern

Die Integration von Action Spaces in einen konzeptionellen und visuellen Rahmen erfordert geeignete Metaphern, die über das naheliegende Konzept von Häusern mit Arbeitsräumen hinausgehen. Deshalb sollen im folgenden Strukturmetaphern für 3D-Anwendungen vorgestellt werden. Natürlich muß auch der Wechsel zwischen Räumen nachvollziehbar sein und durch Navigationsmetaphern unterstützt werden, die im Anschluß vorgestellt werden sollen.

Für die Klassifikation lassen sich zwei grundlegende Strukturmetaphern im Sinne des mentalen Modells unterscheiden, die in Abbildung 10 schematisch dargestellt sind. Die *Theater-Metapher (TM)* geht von einer statischen Nutzerposition (= Blickpunkt) aus, wobei sich der sichtbare Ausschnitt der virtuellen Welt ändert. Dieses Modell ähnelt dem von WIMP-Interfaces, bei dem der Nutzer eine feste Position vor dem Monitor einnimmt und sich die Darstellung wie auf einer Bühne ändert. Wie bei einem Bühnenportal könnte auch in einer 3D-Anwendung ein statischer Rahmen (evtl. mit allgemein benötigten 3D-Widgets) für alle

Action Spaces zur Verfügung stehen. 3D-Dokumente stehen im Mittelpunkt des Interesses, können jedoch auch ausgeblendet werden. Beim Wechsel zwischen Action Spaces ändert sich auch das „Bühnenbild“ und somit die 3D-Widgets und ergänzenden Objekte. 3D-Widgets, Werkzeugpaletten etc. können dabei auch hinter oder neben dem Nutzer abgelegt sein und bei Bedarf eingeblendet werden.

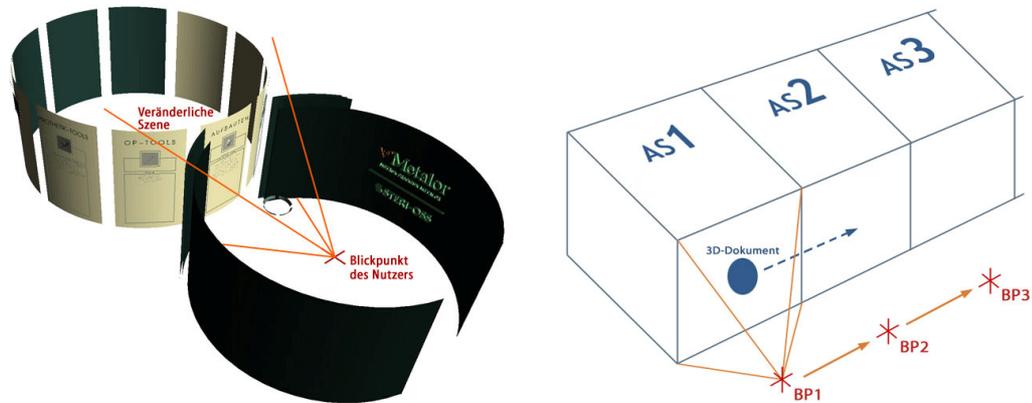


Abbildung 10: Grundlegende Strukturmetaphern für Action Spaces: Theater-Metapher (links) und Bewegungs-Metapher (rechts)

Die *Bewegungs-Metapher (BM)* geht von einer Änderung des Nutzer-Blickpunktes aus, was sich leicht mit dem Betreten verschiedener Räume von einem Flur aus illustrieren läßt. Die Action Spaces werden dabei vollständig geändert. Es ist auch möglich, daß 3D-Widgets in verschiedenen Räumen gemeinsam zur Verfügung stehen. 3D-Dokumente können entweder in einem Action Space bleiben oder in den gewünschten mitgenommen werden.

3.3.3.1 Raum- und Strukturmetaphern

Folgende grundlegende geometrische Strukturen werden zur Gliederung von 3D-Anwendungen vorgeschlagen:

- *Räume*: Grundlegende Struktur, die nicht rechteckig sein muß, sondern auch in Zell-, Blasen-, Kugel- oder anderer Form auftreten kann. Als Teil sowohl der TM und BM sind diese sehr vielfältig anwendbar.
- *Drehbühnen*: Komplexere Struktur, die mehrere durch Rotation erreichbare AS enthalten kann. Die gemeinsame Nutzung von 3D-Widgets wird gut unterstützt. Sie eignen sich für TM und Anwendungen aus den Bereichen 3D-Modellierung, Produktpräsentation, Training etc.
- *Gebäude mit Etagen und Fluren*: Räume, Drehbühnen oder ähnliche stapelbare Strukturen können vertikal wie bei Häusern oder Türmen und horizontal wie bei Arkaden oder Fluren gruppiert werden. Diese zumeist regulären und linearen Strukturen sind typische Beispiele für die BM, eignen sich besonders für sequentielle Aufgaben und sind vielseitig einsetzbar.
- *Raumstationen, Moleküle, Netze*: Erlauben größere geometrische Freiheit, da Zellen, Blasen, Räume, Kugeln o.ä. in einem möglicherweise nicht linienförmigen Raster, Ring oder Netzwerk angeordnet sind. Einzelne Zellen können durch Röhren oder Gänge miteinander verbunden werden, Subhierarchien mit mehreren gruppierten AS werden gut unterstützt. Sie eignen sich für die BM und zur Unterstützung von assoziativen, abhängigen oder anderweitig verbundenen Arbeitsaufgaben (bzw. AS), z.B. in den Bereichen Informationsvisualisierung, E-Commerce, Training, Unterhaltung.

- *Urbane Metaphern (Stadt, Straße, Platz etc.)*: Erlauben ein deutlich freieres Layout als die vorangegangenen Metaphern und erzeugen somit möglicherweise beim Nutzer die Erwartung freier Navigationsmöglichkeiten. Dafür sind diese Vertreter der BM sehr leicht verständlich und eignen sich für Anwendungen aus den Bereichen E-Commerce, Unterhaltung, Virtuelle Gemeinschaften und Virtuelle Demokratie.

3.3.3.2 Navigationsmetaphern

Strukturen, die dem Prinzip der Theatermetapher folgen, benötigen aufgrund des fixierten Blickpunktes keine Navigationsunterstützung. Für alle in die Kategorie der Bewegungsmetaphern gehörenden Strukturen sind jedoch Transportmetaphern nötig, die den Nutzer zu vordefinierten Plätzen bringen (auch *constraint travel* [Bowma98] genannt). Eine Unterscheidung zwischen strikt *sequentiellen* Metaphern, die eine vorgegebene Sequenz von Handlungen / Action Spaces unterstützen und *parallelen* Metaphern, die eine Auswahl zwischen Alternativen ermöglichen, hat sich dabei als hilfreich erwiesen. Für jede Metapher ist in Klammern der jeweilige Typ angegeben, wobei die folgende Zahl die Anzahl der Navigationsalternativen wiedergibt. Außerdem nimmt bei der folgenden Zusammenstellung die mögliche Bewegungsfreiheit immer mehr zu.

- *Fahrstuhl oder Hebebühne (sequ., 2)*: Eignung für vertikale Bewegung zwischen unterschiedlichen Ebenen, z.B. in einem Gebäude. Unterstützung sequentieller Aufgaben mit der Option des Vor- und Zurückgehens. Bei einem Fahrstuhl sind Anfang und Ziel fixiert, bei einem Paternoster Zyklen möglich.
- *Schienenfahrzeug (sequ., 2)*: „Geführte“ Tour zwischen horizontal angeordneten AS (z.B. Arkaden oder Passagen) wie auf einem Kamerawagen oder aus einem Zugabteil blickend. Lineare und komplexere Pfade sind möglich, sequentielle Aufgaben und zwei wählbare Richtungen werden unterstützt. Wie bei Weichen sind jedoch auch parallele Alternativen denkbar. Im Falle von *Achterbahnen* sind auch vertikale und komplexere Bewegungen möglich.
- *Seilbahn / Rutsche (sequ., 2/1)*: Auch mit diesen Metaphern sind „geführte Touren“ durch AS möglich. Freiere Formen, z.B. spiralartige, sind denkbar, wo an gewünschten AS einfach angehalten wird. Im Gegensatz zu Schienenfahrzeugen sind Rutschen Einbahn-Metaphern, die eine strikte Sequenz von AS unterstützen.
- *Fliegender Teppich oder Stuhl (par., n)*: Fliegende Stühle [Bowma98] oder Teppiche [Pausc96] erlauben freiere Bewegung ohne vorgegebenen Pfad und unterstützen so die parallele Auswahl von AS. Desorientierung kann zum Problem werden, wenn nicht wenigstens Teile des Transportmittels sichtbar sind und der zurückgelegte Weg durch Animation nachvollziehbar gemacht wird.
- *Teleportierung (par., n)*: Damit lassen sich beliebige, auch sehr entfernte und nicht miteinander verbundene AS aufsuchen. Nutzerorientierung ist nur äußerst schwer zu erreichen, wenn der aktuelle AS verschwindet und der neue eingeblendet wird. Die parallele Auswahl von AS wird gut unterstützt.

3.3.3.3 Kombination von Metaphern und Transport von Objekten

Generell ist festzustellen, daß Struktur- und Navigationsmetaphern sich in 3D-Anwendungen relativ frei miteinander kombinieren lassen. Auch eine Mischung aus Strukturen der Theater- und Bewegungsmetapher ist möglich. Bestimmte Kombinationen passen jedoch besser zusammen [Dachs00], was in der Tabelle 3 dargestellt ist. Je mehr Plus-Zeichen auftauchen, um so besser passen zwei Metaphern zusammen, Paare mit einem Minus sind nur sehr wenig oder gar nicht geeignet.

| | Fahrschein | Schiene-fahrzeug | Rutsche | Fliegender Teppich / Stuhl | Teleportierung |
|---------------|------------|------------------|---------|----------------------------|----------------|
| Raum | ++ | + | + | + | + |
| Drehbühne | ++ | - | + | + | + |
| Gebäude | +++ | ++ | + | - | - |
| Raumstation | ++ | ++ | - | ++ | + |
| Molekül | + | - | +++ | ++ | + |
| Stadtmetapher | - | +++ | ++ | +++ | ++ |

Tabelle 3: Bewertung der Kombinationsmöglichkeiten von Struktur- und Navigationsmetaphern

Bestimmte Werkzeuge müssen in mehreren Action Spaces verfügbar sein. Bei der Bewegungsmetapher können sie z.B. in miniaturisierter Form mitgeführt werden. Auch die Technik der *Toolspaces & Glances* [Pierc99] kann dafür eingesetzt werden. Bei der Theatermetapher lassen sich gemeinsam genutzte Werkzeuge leicht innerhalb der unveränderlichen visuellen Struktur unterbringen (z.B. Portal).

Auch 3D-Dokumente lassen sich zwischen Action Spaces transportieren, wofür sich besonders Fahrstühle und Hebebühnen eignen, da hier Dokumente in Originalgröße transportiert werden können. Die Abbildung 11 zeigt schematisch eine mögliche Hebebühne und drei übereinander angeordnete Action Spaces. Aber auch das temporäre „Schrumpfen“ eines 3D-Objektes zu Transportzwecken oder die halbtransparente Darstellung während des Action Spaces-Wechsels ist denkbar.

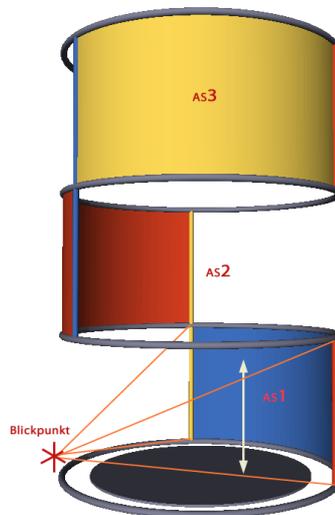


Abbildung 11: Hebebühnen-Metapher zur Verbindung von Action Spaces

3.4 3D-Widgets: Klassifikation und Systematisierung

Im Abschnitt 3.1.3.2 wurden 3D-Widgets als interaktive Bestandteile einer 3D-Anwendung, die Geometrie und Verhalten kapseln [Conne92], bereits eingeführt. Der 1992 von Forschern der Brown University geprägte Begriff wird auch synonym als *Control Object*, *Interactive Manipulator* [Strau92], *Interactor* [Kessl99] oder *3D-Gadget* [Schön99] bezeichnet. Dabei handelt es sich immer um 3D-Interaktionsobjekte, die wie Anwendungsobjekte gehandhabt und in den 3D-Raum integriert werden. Dafür wurden im Abschnitt 3.2.11 bereits spezielle Richtlinien und Gestaltungsempfehlungen aufgestellt.

In der VR-Literatur der letzten Dekade ist eine Vielzahl von häufig ad hoc und mit proprietärer Software entwickelten sowie in einem bestimmten Anwendungskontext entstandenen 3D-Interaktionstechniken und 3D-Widgets zu finden. Der allgemein beklagte Mangel an verfügbaren Widgets als Grundlage möglicher Standards ([Conne92], [Snibb92], [Hernd94a], [Geige98]) sowie nur wenige Ansätze zu ihrer Systematisierung und Klassifikation [Leine97] sind Indizien für eine immer noch fehlende Reife dieses Forschungsgebietes. Auch die in diesem Kapitel aufgestellte Systematisierung kann nur ein Beitrag auf dem eher langen Weg zu einer etablierten und erfolgreichen 3D-GUI-Entwicklung sein.

Daß bei der Entwicklung von VR-Applikationen 2D-GUIs oft Pate gestanden haben, sieht man an der zahlreichen Übernahme von 2D-Konzepten in 3D-Anwendungen. Da die meisten Nutzer mit 2D-Interaktionselementen und -techniken vertraut sind, ist die Integration von 2D-Menüs, Schiebereglern etc. naheliegend, insbesondere dann, wenn eine Projektion auf eine ebene Fläche vorgenommen wird. Mehrere Arbeiten berichten von einer erfolgreichen Übernahme von 2D-Widgets in den 3D-Raum, wenn diese auf einer virtuellen Repräsentation eines in der Hand gehaltenen Grafiktablets angeordnet sind und z.B. mit einem Stift bedient werden [Angu95]. Beispiele dafür sind *Hand-Held Windows* [Linde99] oder das *Personal Interaction Panel* innerhalb des AR-Forschungsprojektes *Studierstube* ([PIP@], [Szala97]).

Die Integration zweidimensionaler Widgets in 3D-Anwendungen weist jedoch auch eine Reihe von Nachteilen auf. Die Position dieser Widgets im Raum ist häufig ungeklärt oder kann sogar – z.B. bei Nutzung von Stereobildschirmen – störend und irritierend wirken [Hand97]. Auch die ungenügende Textlesbarkeit im 3D-Raum erweist sich als Problem, da 2D-Widgets einerseits sehr textlastig sind, und andererseits die Blick- und Interaktionsebene in 3D-Anwendungen nicht feststeht. Zudem sind spezialisierte 3D-Widgets häufig viel intuitiver, was am Beispiel der in [Snibb92] beschriebenen 3D-Deformationswidgets – verglichen mit einer komplexen Dialogbox mit mehreren Schiebereglern – deutlich wird. Auch wäre damit der direkte 3D-Bezug weniger gegeben und das Interaktionsgefühl durch den metaphorischen Bruch gestört. Schließlich muß noch erwähnt werden, daß bei der Nutzung von 3D-Eingabegeräten aus einer ursprünglich einfachen, eindimensionalen Aufgabe wie der Listenauswahl eine komplizierte und dreidimensionale Aufgabe wird, wobei sich die Abschätzung der Tiefe (und somit das „Treffen“ eines im Raum angeordneten 2D-Widgets) ohne stereoskopische Darstellung sehr schwierig gestaltet [Hand97].

Die nach einer Betrachtung verwandter Arbeiten und möglicher Klassifikationsaspekte folgende Systematisierung von Widgets bezieht jedoch sowohl aus 2D-GUIs übernommene als auch speziell für den 3D-Raum entwickelte Widgets ein. Verbindlich für die Klassifikation ist das Kriterium einer geometrischen Repräsentation, das auch bei [Conne92] zur Definition von 3D-Widgets herangezogen wird. Damit finden bestimmte VR-Interaktionstechniken hier keine Erwähnung, zumal der Fokus auf Desktop-VR-Anwendungen liegt, für die Widgets eine intermediäre Abbildungsfunktion besitzen. Erwähnt werden muß auch, daß die Klassifikation natürlich nicht vollständig sein kann und nicht jede Widget-Variante aufgenommen wurde. Die wesentliche Literatur und alle wichtigen 3D-Widgets wurden jedoch berücksichtigt.

3.4.1 Verwandte Arbeiten

An dieser Stelle werden nicht die zahlreichen anwendungsspezifischen VR-Projekte aufgeführt, in deren Kontext 3D-Widgets und Interaktionstechniken entstanden sind und die in Recherche und Analyse einbezogen wurden. Im folgenden finden jedoch wissenschaftliche Arbeiten Erwähnung, die sich explizit mit 3D-Widgets und ihrer Systematisierung beschäftigen.

3.4.1.1 Widget-Bibliotheken

Anfang der 90er Jahre wurde an der Brown University nicht nur der Begriff 3D-Widget geprägt [Conne92], sondern auch zahlreiche Widgets und Toolkits zu ihrer Konstruktion ([Zelez93], [Steve94]) entwickelt. Dabei standen jedoch noch eher grundlegende Fragen interaktiver 3D-Grafik im Vordergrund, so daß Widgets noch aus geometrischen Primitiven aufgebaut und nur wenig parametrisierbar waren.

Innerhalb der Forschungsplattform *3Dwm* [3DWM@] wurde ein 3D-Widget-Kit als Sammlung standardisierter Interaktionsbausteine entwickelt, darunter *Buttons*, *Sliders*, *Text Fields*, *Cone Trees* und diverse *Manipulation Handles*. Bei den Widgets handelt es sich um Subszenengraphen mit visuellen und Verhaltensknoten, die imperativ nachgenutzt werden können, jedoch keine High-Level-Parametrisierung oder Anpaßbarkeit besitzen.

Im Rahmen des Projektes *it3d (Interactive Toolkit library for 3D applications)* [Osawa02], einer interaktiven Werkzeugbibliothek zur Entwicklung von 3D-Applikationen unter Nutzung von AR-Technologien, wurde ebenfalls eine 3D-Widget-Bibliothek entwickelt. Mit den bereitgestellten 3D-Widgets *Button*, *Switch*, *Slider*, *Dial*, *Label*, *Text Box*, *List Box* und *Combo Box* handelt es sich dabei vor allem um ein aus 2D-GUIs übernommenes Repertoire. Widgets lassen sich über Attribute anpassen, wobei intern ein prototypbasiertes, objektorientiertes Delegationsmodell verwendet wird.

3.4.1.2 Klassifikationen

In [Kett93] werden 42 3D-Metaphern zur geometrischen Manipulation aus informationstheoretisch-mathematischer Sicht erörtert und klassifiziert. Mit dieser Arbeit wurde ein theoretisches Fundament für 3D-Widgets insbesondere im Bereich der direkten geometrischen Manipulation gelegt.

Die sogenannte *Widget Working Group* [WWG@] wurde 1997 von Seidman und Hand ins Leben gerufen, um Widgets zu klassifizieren und eine Sammlung wiederverwendbarer VRML-Widgets zu schaffen. Obwohl die damaligen Ziele denen dieser Arbeit sehr ähneln und interessante Beispiel-Widgets für VRML97 entstanden, schloßen die Aktivitäten dieser Arbeitsgruppe bald wieder ein.

Leiner, Preim und Ressel bieten in [Leine97] erstmalig einen Überblick über bis dahin existierende Widgets, erläutern Anwendungsfelder, Besonderheiten und Anforderungen und nehmen schließlich eine erste grobe Unterteilung von 3D-Widgets zur geometrischen Manipulation und solche zur Anwendungskontrolle vor. Diese erste nennenswerte Klassifikation vielfältiger Widgets ist jedoch eher als Sammlung und Grundlagenarbeit zu verstehen. Zudem erweist sich die Einteilung auch noch als zu wenig differenziert.

Einige wenige weitere Subklassifikationen existieren. In [Watt98] (Kap. 20.3) wird eine Klassifikation von Methoden zur direkten Manipulation von Objekten mittels 3D-Widgets vorgenommen. Die Unterteilung in lineare, nichtlineare und High-Level-Manipulation wurde für diesen Teilbereich übernommen. In [Encar02] werden 3D-Widgets zur Anwendungskontrolle in hand-orientierte Menüs, transformierte 2D-Widgets und direkt mit Objekten assoziierten 3D Widgets eingeteilt. In [Shnei98] (S. 238 ff.) wird – wenn auch auf 2D-GUIs bezogen –

eine Einteilung von Menütechniken vorgenommen, welche in die im Abschnitt 3.4.8.1 vorgestellte Menüklassifikation eingeflossen ist.

In [Braig00a] werden detaillierte Kriterien zur Analyse von 3D-Widgets und Interaktionstechniken aufgestellt. 22 Widgets und Interaktionstechniken werden nach diesen Kriterien systematisiert. Darauf aufbauend wird in [Hinz01] eine umfassendere Klassifikation nach den Einsatzbereichen Selektion, Manipulation, Navigation, 3D-Manipulation, Anwendungskontrolle, Informationsvisualisierung und Orientierung vorgenommen. Dabei sind etwa 80 verschiedene 3D-Widgets und 3D-Interaktionstechniken berücksichtigt worden. Ein großer Teil dieser Systematisierung floß auch in die unten vorgestellte Klassifikation ein, die jedoch modifiziert und um weitere Widgets erweitert wurde.

3.4.1.3 Interaktionstechniken

Interaktionstechniken, die im Abschnitt 3.1.3.3 bereits als Widgets mit einem zugehörigen Verhalten – in den meisten Fällen aber ohne geometrische Repräsentation – gekennzeichnet wurden, bilden einen wichtigen Schwerpunkt in der VR-Literatur der letzten Dekade. Forschergruppen um Bowman, Mine, LaViola, Popyrev u.a. entwickelten eine Fülle von 3D-Interaktionstechniken für virtuelle, vorrangig immersive Umgebungen. Diese Ergebnisse wurden ausführlich in [Bowma00] und [Bowma01b] dokumentiert. Spezielle VE-Interaktionstechniken wurden bereits Mitte der 90er Jahre in [Mine95] vorgestellt und in der Überblicksdarstellung [Hand97] durch weitere ergänzt. In [Bowma99a] und Bowmans Dissertation [Bowma99b] wird eine ausführliche Taxonomie von Interaktionstechniken vorgestellt und die inzwischen etablierte Grundeinteilung in Techniken zur Selektion, Manipulation und Navigation/Fortbewegung vorgenommen. Diese Klassifikation floß auch in dieses Kapitel ein, wobei in Abgrenzung zu allgemeinen VR-Interaktionstechniken nur jene Berücksichtigung fanden, in denen geometrische Widget-Repräsentationen eine Rolle spielen.

Klassifikationen existieren auch für einzelne Spezialaspekte, z.B. die in [LaVio00] und [Tan01] vorgestellten Taxonomien für Navigationstechniken oder die in [Popy98] vorgenommene Einteilung von Manipulationstechniken in exozentrische und egozentrische Metaphern. Jüngere Arbeiten beschäftigen sich mit einer praktikablen Spezifikation von Techniken, die auf verschiedene Technologien und Toolkits abgebildet werden kann. So nimmt Figueroa mit der XML-basierten Spezifikationssprache *InTML (Interaction Technique Markup Language)* [Figue02] z.B. eine Klassifikation und genaue deklarative Beschreibung von bekannten VR-Interaktionstechniken vor, die in einigen Punkten der hier vorgestellten ähnlich ist, bei der jedoch der Fokus auf Interaktionstechniken statt auf interaktiven Widgets mit geometrischer Ausprägung liegt.

3.4.2 Eine hierarchische Systematik von 3D-Widgets

Der Schwerpunkt dieser hier im Überblick und in den folgenden Abschnitten detaillierter dargestellten Klassifikation liegt auf interaktiven 3D-Widgets mit geometrischer Repräsentation und insbesondere auf Widgets zur Anwendungskontrolle. Ausschließlich für bestimmte Domänen oder Spezialanwendungen konzipierte Widgets wurden aufgrund der fehlenden Allgemeingültigkeit nicht berücksichtigt.

Zur Klassifikation von 3D-Widgets bieten sich zahlreiche Kriterien an. Da jedoch bestimmte Teilgruppen von Widgets unterschiedliche Aspekte besitzen, ist eine allgemeingültige und umfassende Klassifikation nicht einfach vorzunehmen. So kann z.B. die Zahl von Menüeinträgen, die mit einer konkreten Menüausprägung dargestellt werden können, von großer Bedeutung für diese Teilgruppe sein, für alle anderen jedoch bedeutungslos. Auch muß der Aspekt der potentiellen Nutzergruppe für eine Klassifikation gesehen werden. Eine sehr informationstheoretische oder mathematische Taxonomie wird für einen Anwendungsent-

wickler z.B. wenig hilfreich sein. Die in dieser Arbeit aufgestellte Systematik ist primär für die Sichtweise des Anwendungsentwicklers und Komponentenprogrammierers gedacht und trägt somit eher pragmatischen Charakter im Sinne einer tatsächlichen praktischen Verwendbarkeit. Damit unterscheidet sie sich z.B. signifikant von den bekannten Klassifikationen für Interaktionstechniken, die sich ausschließlich an Wissenschaftler richten. Zunächst sollen mögliche generelle Klassifikationskriterien erläutert werden:

- *Anwendungsgebiet*: Ein sehr allgemeines Kriterium, was aufgrund der Vielfalt von Gebieten ungeeignet ist und Mehrfacheinordnungen zwingend nötig machen würde.
- *Interaktionsaufgabe*: Ein nützliches Kriterium, jedoch einerseits zu ungenau (eine Selektion gehört z.B. fast immer zu einer Interaktion) und andererseits etwas zu technisch und für einen Anwendungsentwickler wenig hilfreich.
- *Einsatzbereich / Interaktionsziel*: Hiermit ist die generelle Teilaufgabe und das beabsichtigte Ziel einer Handlung gemeint, ohne auf technische Teilhandlungen (notwendige Interaktionstechniken) einzugehen. Dieses Kriterium entspricht am ehesten einer pragmatischen Herangehensweise, wo Ziele verfolgt werden, wie z.B.: die Auswahl eines Objektes, die Umschaltung zwischen drei Zuständen oder die schnelle Navigation zwischen Räumen.
- *Freiheitsgrade / Dimensionalität*: Das können einerseits die möglichen Freiheitsgrade der Widget-Interaktion und andererseits die der verwendeten Eingabegeräte sein.
- *Eingabegerät*: Dazu zählt die Eignung oder sogar Notwendigkeit bestimmter Eingabegeräte für konkrete Widgets. Auch eine Gruppierung in Widgets, die sich in der Hand halten bzw. auf einer mit einem Grafiktablett assoziierten Fläche bedienen lassen, ist denkbar.
- *VR-Eignung*: Hierunter ist zu verstehen, ob ein Widget sich für immersive und nicht-immersive 3D-Anwendungen eignet, ebenso für eine stereoskopische Ausgabe oder nicht.
- *Erscheinungsbild*: Verschiedene phänomenale Eigenschaften, wie Lage, Platzbedarf, Orientierung, Nutzung von Transparenz, Art der verwendeten Geometrien etc. können ebenfalls zur Klassifikation herangezogen werden.
- *Objektassoziation*: Für Widgets kann unterschieden werden, ob sie sich auf andere Objekte und Dokumente auswirken (wenn ja, auf welche) oder selbst ein Datenobjekt / 3D-Dokument darstellen (Bsp. Widgets zur Informationsvisualisierung).

Weitere Kriterien sind denkbar, z.B. die Ressourcenbeanspruchung, die Art des Feedbacks oder die Existenz von äquivalenten 2D-Widgets (s.a. [Hinz01]). Für eine brauchbare Klassifikation sind sie jedoch weitgehend ungeeignet. Das Kriterium „Einsatzbereich / Interaktionsziel“ wurde schließlich für die grundlegende Einteilung gewählt. Die weitere Unterklassifikation erfolgt dann flexibel je nach Widget-Gruppe mit unterschiedlichen Kriterien, wie z.B. Erscheinungsbild oder Art der manipulierten Daten, wobei der Fokus immer auf einer Auswahl im Sinne der Anwendungsentwicklung liegt. Für 3D-Menüs wird unter 3.4.8.1 auch eine alternative Einteilung vorgestellt. Verschiedene Widgets lassen sich auch mehreren Kategorien gleichzeitig zuordnen, werden jedoch nur an einer Stelle aufgeführt.

Die Tabelle 4 zeigt alle Klassifikationskategorien im Überblick, wobei die konkreten Widget-Varianten aus Gründen der Übersichtlichkeit weggelassen wurden. In den rechts angegebenen Unterkapiteln findet sich jeweils eine ausführliche Darstellung der entsprechenden Kategorie mit ihren zugehörigen Widgets (dann kursiv hervorgehoben). Neben der hier wiedergegebenen Darstellung ist unter [CONTIGRA@] auch eine aktuelle Fassung mit Verweisen auf konkrete Komponentenspezifikationen online zu finden. Diese Widget-Klassifikationshierarchie im Web erleichtert das Verständnis der folgenden Unterkapitel.

| | |
|--|--------------|
| Direkte 3D-Objektinteraktion | |
| Objektselektion | (s. 3.4.3) |
| Direkte Selektion | |
| Okklusionsselektion | |
| Distanzselektion | |
| Geometrische Manipulation | (s. 3.4.4) |
| Lineare Transformation | |
| Nichtlineare Transformation | |
| High-Level-Objektmanipulation | |
| Manipulation der 3D-Szene | |
| Orientierung und Navigation | (s. 3.4.5) |
| Direkte Blickpunktselektion | |
| Geführter Transport | |
| Miniaturdarstellungen | |
| Steuerung der Szenenpräsentation | (s. 3.4.6) |
| Lichtmanipulation | |
| Kameramanipulation | |
| Soundsteuerung | |
| Exploration und Visualisierung (s. 3.4.7) | |
| Geometrische Exploration | |
| Hierarchievisualisierungen | |
| Visualisierungen von Graphen | |
| Visualisierung von 2D-Daten und Dokumenten | |
| Wissenschaftliche Visualisierung | |
| Anwendungskontrolle (s. 3.4.8) | |
| Zustandsänderung / Diskrete Wertgeber | |
| Aktivierung | |
| Zwei Zustände | |
| Multiple Zustände | |
| Kontinuierliche Wertgeber | |
| Skalare Werte | |
| Multiple Werte | |
| Spezielle Werteeingaben | |
| Farbwähler | |
| Menüselektion | (s. 3.4.8.1) |
| Temporäre Optionsmenüs | |
| Einzelmenüs | |
| Menühierarchien | |
| Container | |

Tabelle 4: Kategorien der Klassifikation von 3D-Widgets nach Einsatzbereichen / Interaktionszielen

Wie in Tabelle 4 ersichtlich, wurde für 3D-Widgets eine Einteilung in folgende Hauptgruppen vorgenommen:

- *Direkte 3D-Objektinteraktion:* Hierzu zählen alle direktmanipulativen Verfahren zur Interaktion mit 3D-Objekten im Sinne von 3D-Dokumenten oder Dokumentbestandteilen. Dazu gehören Verfahren zur Selektion von frei in der Szene befindlichen Objekten sowie Verfahren zu ihrer direkten geometrischen Manipulation mittels Widgets, die mit 3D-Objekten assoziiert sind.
- *Manipulation der 3D-Szene:* Zu dieser Kategorie gehören Widgets, mit denen sich Parameter der 3D-Szene an sich manipulieren lassen. Dazu zählen solche zur Orientierung und Navigation, die letztlich den Blickpunkt verändern, und Widgets zur Beeinflussung der Szenenpräsentation, also zum direkten Manipulieren von Lichtquellen, Kamera und Sound.
- *Exploration und Visualisierung:* Bei dieser Gruppe handelt es sich einerseits um generische Explorationswidgets, die Teilaspekte geometrischer Objekte sichtbar machen. Andererseits zählen dazu Informationsvisualisierungs-Widgets zur Repräsentation zumeist nicht-geometrischer, also abstrakter Daten, wie Hierarchien, Graphen oder Stromlinien, aber auch zweidimensionaler Informationen und Dokumente.
- *Anwendungskontrolle:* In diese Kategorie fallen generische Widgets zur Zustandsänderung, Werteeingabe, Auswahl und Containerwidgets. All diese können für eine Vielzahl von allgemeinen Kontrollaufgaben eingesetzt werden. Dabei dienen sie der indirekten Manipulation von Parametern der verschiedensten Art und ähneln damit 2D-GUI-Konzepten. Das heißt nicht, daß es sich dabei nur um reine 3D-Varianten von 2D-Widgets handelt.

In den folgenden Unterkapiteln werden die einzelnen Kategorien zunächst näher erläutert und die existierenden Widget-Varianten tabellarisch zugeordnet. Danach folgt jeweils eine Tabelle, die alle wichtigen Widgets einer Kategorie mit ihren Eigenschaften im Kurzüberblick gegenüberstellt. Dabei wurden ähnliche Widgets aus verschiedenen Arbeiten bereits zusammengefaßt, in nur einer Spalte dargestellt und mit einem konkreten Beispiel vorgestellt. Eine ausführlichere Darstellung der einzelnen Widgets und ihrer Funktion findet sich in der jeweils angegebenen Originalquelle. Die Literaturangaben beziehen sich entweder auf die Erstveröffentlichung eines bestimmten Widgets oder auf einen wichtigen Repräsentanten dieser Gruppe. Wenn nicht anders gekennzeichnet, sind die Abbildungen auch dieser Quelle entnommen oder stammen vom Autor. Bei der Eignung für immersive / nicht-immersive Systeme bedeutet Fettdruck, daß ein Widget primär für diese Anwendungsart entwickelt wurde. Während die Subkategorien deutsch benannt werden, wurden konkrete Widget-Ausprägungen oft nicht extra übersetzt, da sie meistens unter ihrem Eigennamen bekannt sind oder sich nicht gut ins Deutsche übertragen lassen.

3.4.3 Widgets zur Objektselektion

Die allgemeine Selektion dreidimensionaler Objekte im Raum stellt eine zentrale Grundlage für jede Form der Manipulation und Interaktion dar. Im Gegensatz zu den im Abschnitt 3.4.8.1 vorgestellten spezialisierten Menütechniken zur Auswahl konkreter Objekte und Optionen aus einer vorgegebenen Menge handelt es sich bei den hier zusammengefaßten Verfahren um grundlegende Selektionstechniken, die üblicherweise aus immersiven VR-Anwendungen stammen. Bei Desktop-VR wird Selektion typischerweise mit Hilfe der Maus und einer vom zugrundeliegenden Toolkit bereitgestellten Strahlenprojektion der 2D-Koordinaten in den 3D-Raum realisiert. Da dafür keine Widget-Geometrie nötig ist und die Selektionstechnik bereits integriert ist, wird die 2D-Selektion hier nicht extra aufgeführt. Im immersiven Bereich sind aufgrund der höheren Freiheitsgrade der Eingabegeräte jedoch auf-

wendigere Techniken notwendig. Mehrere der VR-Selektionstechniken besitzen auch eine geometrische Repräsentation und werden deshalb an dieser Stelle erwähnt. Tabelle 5 zeigt die Klassifikation der Widgets zur Objektselektion im Überblick, Tabelle 6 beschreibt und vergleicht die wichtigsten Vertreter in Kurzform.

Angelehnt an das Konzept zweidimensionaler Cursor können zur **Direkten Selektion** 3D-Cursor (z.B. Punkte oder Fadenkreuze) bzw. eine virtuelle Hand zum Einsatz kommen, die üblicherweise mit einem 3- oder 6-DOF-Eingabegerät gesteuert werden. Da mit dieser Technik kleine Objekte oft nur schwer zu treffen sind und nur einzeln selektiert werden können, wurden als erweiterte Cursortechniken Methoden zur **Okklusionsselektion** eingeführt, bei denen Teile eines Objektes oder ein gesamtes Objekt zur Auswahl von der Selektionsgeometrie verdeckt bzw. eingeschlossen sein müssen. Beim Volumencursor [Zhai96] handelt es sich um einen halbtransparenten Quader, mit dem sich Geometrien selektieren lassen. Das Fischnetz-Werkzeug [Schma99] erlaubt durch eine größere Fläche, die mit Hilfe des *Personal Interaction Panels* [PIP@] wie ein Netz durch die Szene geführt wird, auch die gleichzeitige Auswahl vieler Objekte.

Techniken zur **Distanzselektion** wurden entwickelt, um die Probleme bei der Auswahl entfernter Objekte zu lösen. So läßt sich eine virtuelle Hand bzw. der Arm des VR-Nutzers mit Hilfe einer Armverlängerungsmethode, wie der Go-Go-Technik [Poupy96], nichtlinear virtuell ausstrecken. Auch *Virtual-Pointer*-Techniken [Poupy98] haben das gleiche Ziel, wobei einfache *Ray-Casting*-Techniken einen von der virtuellen Hand ausgehenden zylinderförmigen Strahl zur Objektselektion verwenden und Abwandlungen, wie z.B. die *Spotlight*-Auswahl, die Trefferquote vor allem bei kleinen Objekten erhöhen. Mit der Technik *Image Plane Selection* [Pierc97a] lassen sich auch sehr entfernte Objekte selektieren, da sich durch den aktuellen Blickpunkt des Nutzers und entsprechende Handgesten Projektionen berechnen lassen, in die entfernte Objekte eingeschlossen sind.

Es ist festzustellen, daß mit Ausnahme des Volumencursors alle hier vorgestellten Selektionstechniken von speziellen höherdimensionalen Eingabegeräten abhängig und kaum für Desktop-VR-Anwendungen geeignet sind.

| Direkte 3D-Objektinteraktion | |
|-------------------------------------|--|
| Objektselektion | |
| Direkte Selektion | |
| | <i>3D-Cursor</i> |
| | <i>Virtuelle Hand</i> |
| Okklusionsselektion | |
| | <i>Volumencursor</i> |
| | <i>Fischnetz</i> |
| Distanzselektion | |
| | <i>Armverlängerung (Go-Go Interaction)</i> |
| | <i>Virtueller Zeiger – Raycasting</i> |
| | <i>Virtueller Zeiger – Spotlight</i> |
| | <i>Projektive Selektion (Image Plane)</i> |
| Geometrische Manipulation | (s. 3.4.4) |

Tabelle 5: Klassifikation der Widgets zur Objektselektion

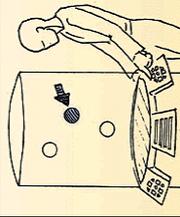
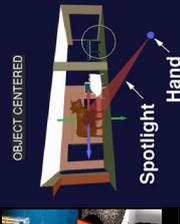
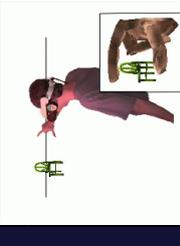
| Objektselektionswidgets | Direkte Selektion | | Okklussionsselektion | | Distanzselektion | | |
|---|---|---|---|--|---|---|---|
| | 3D-Cursor | Virtuelle Hand | Semitransparenter Volumencursor (Silk Cursor) | Fischnetz | Armverlängerung (Go-Go Interaction) | Virtuelle Zeiger (Spotlight, Raycasting) | Projektive Selektion (Image Plane) |
| Kurzbeschreibung | Selektion durch einen 3D-Punktcursor | Eine virtuelle Hand wird vom Nutzer zur direkten Objektselektion gesteuert | Selektion mit einem semitransparenten Volumenobjekt / 3D-Volumencursor | Auswahl mehrerer Objekte durch Bewegen eines Grafiktablets / Fischnetzes durch die Szene | Die Position der realen Hand wird nichtlinear auf die der virtuellen abgebildet | Objekt wird mit Laserpointer oder Kegelstrahl selektiert | Objekte werden mittels Handgesten selektiert und manipuliert |
| Bild |  |  |  |  |  |  |  |
| Referenz für immersive Systeme | z.B. [Balak01] | z.B. [Poupy96] | [Zhai96] | [Schma99] | [Poupy96] | z.B. [Mine95] | [Pierc97a] |
| für nicht-immersive Syst. | ja | ja | ja | ja | ja | ja | ja |
| Eingabegerät | eher nicht | eher nicht | ja | eher nicht | nein | eher nicht | nein |
| Dimensionalität der Interaktionsaufgabe | 3 DOF + 1 Trigger | 3 DOF + 1 Trigger | 3 DOF + 1 Trigger; 6 DOF, wenn mit Skalierung | Personal Interaction Panel (PIP) + Stift | 3 DOF + 1 Trigger | 6 DOF + 1 Trigger; Maus bei Desktop-VR | Datenhandschuh, Tracker |
| Geometrische Ausprägung / Widget | 3D | 3D (evtl. 6D) | 3D | 6D | 3D (evtl. 6D) | 2D (3D, falls Strahlursprung flexibel) | 3D + Gesten |
| Darstellungsaufwand für entfernte Objekte | Cursor (Kugel, Fadenkreuz, Punkt) | Virtuelle Hand | transparenter Quader | Halbtransparente Fläche („Fischnetz“) | Virtuelle Hand und Positionsmarker | Strahl und evtl. Hand | Silhouetten der Hände |
| mehrere Objekte selektierbar | gering | gering | mittel | mittel | gering | gering | gering |
| Kommentare / Besonderheiten | nein | nein | eher nicht | eher nicht | ja | ja | ja |
| | nein | nein | ja | ja, sehr gut | nein | nein, ja bei Kegelstrahl | nein, ja bei Technik framing hands |
| | Wenig angewendet, da unflexibel | Nur für lokale Selektion geeignet, intuitiv und verbreitet | Transparenz vermittelt Tiefengefühl, für Desktop-VR möglich | Effektives und intuitives Multiselektionstool, leider an PIP gebunden | Nichtlineare Abbildung sehr effektiv für entfernte Objekte | Intuitiv, jedoch auch bei Kegelstrahl Trefprobleme | Gute und intuitive Idee, aber an Datenhandschuhe und Gesten gebunden |

Tabelle 6: Übersicht der Widgets zur Objektselektion

3.4.4 Widgets zur Geometrischen Manipulation

Nach [Bowma99a] ist Manipulation die Aufgabe, Position und Orientierung eines zuvor selektierten Objektes zu ändern, möglicherweise auch seine Skalierung oder Form. Gerade die letzten beiden Aspekte, die auch komplexere geometrische Transformationen erfordern, spielen für nicht-triviale Manipulationen eine wichtige Rolle.

Für immersive Umgebungen wurde eine Vielzahl von Techniken zur direkten Manipulation von virtuellen Objekten entwickelt. Nach erfolgter Selektion kommen dazu zahlreiche einhändige und zweihändige Techniken, teils unter Nutzung weiterer physikalischer Eingabegeräte, wie z.B. Datenhandschuhe oder Tablett zum Einsatz. Häufig werden also Selektions- und Manipulationstechniken im Zusammenhang betrachtet, insbesondere, wenn 3-DOF- oder sogar 6-DOF-Eingabegeräte eingesetzt werden, bei denen eine 1:1 – Abbildung der Freiheitsgrade existiert. So sind mit virtuellen Händen beide Aufgaben möglich, ebenso mit der HOMER-Technik (*Hand-centered Object Manipulation Extending Ray-casting*) [Bowma99b], wobei eine Strahlselektion mit einer handbasierten Objektmanipulation kombiniert wird. Es existieren sogar noch komplexere Mischformen von Interaktionstechniken, bei denen zur Selektion und Manipulation auch eine Navigationskomponente hinzukommt, z.B. die Technik *Worlds In Miniature* [Stoak95], die den Nutzer mit einer kleinen Kopie der virtuellen Welt ausstattet, die in der Hand gehalten und zur Manipulation genutzt werden kann.

In [LaVio00] wird eine zweite Gruppe von Werkzeugen und Techniken zur Manipulation beschrieben, die der physikalischen Requisiten bzw. Gegenstände (*Props, Phicons*). Diese sind zu einer intuitiveren Manipulation geeignet, weil sie unserem täglichen Umgang mit realen Interaktionsaufgaben stark entsprechen. Dabei dienen einfache Gegenstände als Stellvertreter für virtuelle Objekte, geben dem Nutzer haptische Rückmeldungen und bilden eine kognitive Brücke zwischen virtuellem Objekt und seiner beabsichtigten Nutzung [LaVio00]. Gerade bei AR-Oberflächen nehmen solche Interaktionsformen an Bedeutung zu.

Beide Gruppen benötigen jedoch höherdimensionale Eingabegeräte oder spezielle Interaktionsgegenstände, weshalb sie in die folgende Aufstellung nicht aufgenommen werden. Die direkte Manipulation von Objekten ohne diese Geräte wird mit Hilfe von 3D-Widgets möglich, die zur Transformation der hohen Freiheitsgrade einer Interaktionsaufgabe auf die niederdimensionalen Eingabegeräte einer Desktop-VR-Umgebung dienen. Dies kann durch die Dekomposition komplexerer Transformationen in einzelne, meist über separate Bedienelemente (*Handles*) steuerbare Teilaufgaben oder die Verwendung von kontextabhängigen Beschränkungen (*Constraints*) ermöglicht werden. Daher spielen Widgets eine zentrale Rolle in Desktop-VR-Umgebungen, wo Datenhandschuhe, 6-DOF-Tracker und Tablett nicht zur Verfügung stehen. Die in der Tabelle 7 klassifizierten Widgets zur geometrischen Manipulation sind alle primär für nicht-immersive Systeme entwickelt worden und somit mit 2D-Eingabegeräten bedienbar. Tabelle 8 beschreibt und vergleicht die wichtigsten Vertreter in Kurzform.

Lineare Transformationen umfassen Translation, Rotation und Skalierung von Objekten. Sämtliche Widgets dieser Kategorie setzen *Constraints* ein, um die Transformationsbestandteile separat editieren zu können. So sind zahlreiche Widgets entwickelt worden, deren Funktionalität im Wesentlichen den mit *Constraining Primitives* [Kiyok00] aufgeführten Widgets gleicht. Hilfslinien zeigen Achsen bzw. Flächen an, auf denen in bestimmten Grenzen Translationen und Rotationen durchgeführt werden können, geometrische Primitive dienen als Anfasser zur Manipulation. Während diese Widgets primär dazu gedacht sind, mit Objekten verknüpft zu werden, um z.B. das Öffnen einer Schranktür zu beschränken, umschließen die folgenden beiden objektzentrierten Widgets im Sinne von *Bounding Volumes* die zu manipulierende Geometrie ganz und stellen zudem Kompositionen verschiedener einzelner Widgets dar.

Die *Virtual Sphere* [Chen88] erlaubt mit der Abbildung von 2D-Mauskoordinaten auf eine transparente Kugel die Steuerung der Rotation des eingeschlossenen Objektes. In der Open-Inventor-Ausprägung [Strau92] ist nicht nur die freie, sondern auch eine achsenbeschränkte Rotation möglich. *Handle Boxes* oder *Universal Transformers* [Strau92] sind primär zur flächenabhängigen Translation (durch Anfassen der Seiten des Quaders) und zur Skalierung mittels Anfassen an den Ecken gedacht, aber auch eine achsenabhängige Rotation ist über die Anfasser an den Achsenendpunkten möglich. In jedem Fall werden kontextabhängig Zusatzgeometrien eingeblendet, die z.B. die Ebenenbeschränkung anzeigen und die Manipulation erleichtern. Schließlich wurde mit den *Interactive Shadows* [Hernd92] eine interessante Technologie vorgestellt, bei der das Objekt nicht direkt transformiert wird, sondern seine Schattenprojektionen die interaktive Einstellung linearer Transformationen ermöglichen. Durch die orthogonalen Schatten werden sehr natürlich wirkende *Constraints* realisiert und im Gegensatz zu anderen Manipulationswidgets keine Objektteile verdeckt.

Nichtlineare Transformationen, wie Verdrehen, Biegen und Verjüngen eines Objektes, lassen sich z.B. mit einem *Rack-Widget* zur Deformation eines Objektes [Snibb92] erreichen, wobei einzelne Anfasser an einer Deformationsachse die Parameter dieser Transformationen steuern.

Werkzeuge, wie die *Freiformdeformation durch Krafteinwirkung* [Hsu92], zählen zur Gruppe der *High-Level-Objektmanipulation*, da sie die mathematischen Parameter (z.B. Kontrollpunkte von Oberflächen) einer komplexen geometrischen Repräsentation verbergen und statt dessen auf intuitivere Weise im Sinne eines geometrischen Modellierens visuell zugänglich machen. Mehrere visuelle Freiform-Operatoren für die Erstellung von Sweep-Objekten, das Zusammenfügen zweier Geometrien oder die Objektdeformation wurden in [Grimm95] vorgestellt. Da inzwischen eine Vielzahl an Widgets zum geometrischen Modellieren existiert, soll hier nur exemplarisch der *Magic Ring* des 3D-Modellierungsprogrammes *TrueSpace* [TrueSpace@] aufgeführt werden, mit dem sich die visuellen Eigenschaften eines Toroids editieren lassen.

| Direkte 3D-Objektinteraktion | |
|-------------------------------------|---|
| Objektselektion | (s. 3.4.3) |
| Geometrische Manipulation | |
| Lineare Transformation | |
| | <i>Eingeschränkte Transf. (Constraining Primitives)</i> |
| | <i>Rotationskugel (Virtual Sphere)</i> |
| | <i>Handle Box (Universal Transformer)</i> |
| | <i>Interactive Shadows</i> |
| Nichtlineare Transformation | |
| | <i>Deformation (Rack)</i> |
| High-Level-Objektmanipulation | |
| | <i>Freiformdeformation durch Krafteinwirkung</i> |
| | <i>Magic Ring; Sweep-, Warp-, Blend-Tool u.a.</i> |

Tabelle 7: Klassifikation der Widgets zur Geometrischen Manipulation

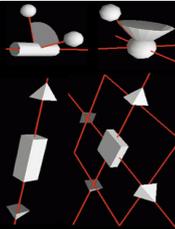
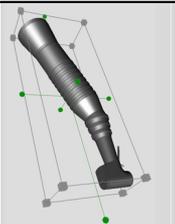
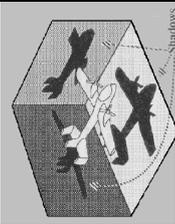
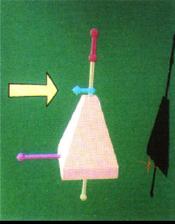
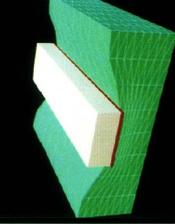
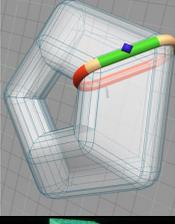
| Widgets zur Geometrischen Manipulation | Lineare Transformation | | | | Nichtlineare Transform. | | High-Level-Objektmanipulation | |
|--|---|--|--|--|--|--|--|--|
| | Eingeschränkte Transformation (Constraining Primitives) | Rotationskugel (Virtual Sphere) | Handlebox (Universal Transformer) | Interactive Shadows | Deformation (Rack Widget) | Freiformdeformation (Repelling Tool) | Parametermanipulationswidgets (z.B. MagicRing) | |
| Kurzbeschreibung | 1D- und 2D-Beschränkung von Translation und Rotation durch Interaktion mit den Primitiven | Eine virtuelle (unsichtbare) Kugel erlaubt die freie, Anfassergeometrien die achsenabhängige Rotation eines Objektes | Anfasser einer Bounding Box erlauben die Manipulation jeweils eines Transformationsbestandteiles | Transformation von Objekten mit Hilfe ihrer an Raumwände projizierten, interaktiven Schatten | Deformationsachse mit verschiedenen Anfassern zur nicht-linearen Transformation von Objektteilen | Nutzung einer Hilfsgeometrie zur Deformation einer Freiformoberfläche mittels Kraffteinwirkung | Manipulation sämtlicher Parameter eines Toroids oder allgemeiner parametrisierter Geometrien | |
| Bild |  |  |  |  |  |  |  | |
| Referenz | z.B. [Kiyok00] | Basis [Chen88], z.B. [Strau92] | z.B. [Strau92] | [Hernd92] | [Snibb92] | [Hsu92] | Basis [Grimm95] Bsp. [TrueSpace@] | |
| Geometrische Ausprägung / Widget | Primitive und Hilfsflächen für Achsen und Ebenen | Rotationsringe und transparente Kugel, kontextabhängige Zusatzgeometrien | Bounding Box und Anfasser für Rotation und Skalierung, kontextabhängige Zusatzgeometrien | Schattenprojektion des jeweiligen Objektes | Verschiedene Anfasser und Achsengeometrie | Objekt zur Kraffteinwirkung | Basisgeometrie mit verschiedenen Bereichen & Handles | |
| Darstellungsaufwand | gering | gering | gering | hoch | gering | höher | höher | |
| Unterstützte Transformationen | Translation, Rotation | Freie Rotation, achsenabhängige Rotation | Translation, Rotation, Skalierung | Translation x, y, z Rotation | Komplexe Deformationen: Verdrehen, Biegen, Verjüngen | Komplexe Änderung von Gitterrepräsentation und Kontrollpunkten | Alle Parameter der relevanten Oberfläche | |
| Kommentare / Besonderheiten | Lassen sich an beliebige Geometrien „andocken“ oder auch aus der Entfernung benutzen | Objektzentriertes Widget mit mehreren Arbeitsmodi und kontextabhängigen Hilfsgeometrien | Objektzentriertes Widget mit mehreren Arbeitsmodi und kontextabhängigen Hilfsgeometrien | Obwohl interessante Idee, selten im praktischen Einsatz zu finden, da Projektionsflächen nötig | Deformation möglichst achsenorientierter Objekte, sonst wenig sinnvoll | Berechnung komplex, Handhabung sehr intuitiv | Zur geometrischen Modellierung existieren zahlreiche Sweep-, Warp-, Blend- u.a. Tools | |

Tabelle 8: Übersicht der Widgets zur Geometrischen Manipulation

3.4.5 Widgets zur Orientierung und Navigation

Nach [LaVio00] lässt sich Navigation innerhalb von VE in drei Hauptkategorien unterteilen. Das sind die Exploration für entdeckendes Bewegen innerhalb einer VE ohne spezielles Ziel, das gezielte Aufsuchen eines Ortes vor allem für Suchaufgaben und schließlich präzise Bewegungen, um eine Aufgabe besser verrichten zu können. Dabei besteht die Navigationsaufgabe aus der motorischen Komponente der reinen Fortbewegung (*Travel*, auch *Viewpoint Motion Control*) und einer kognitiven Komponente des Findens eines Weges (*Wayfinding*). Für die Fortbewegung wurden zahlreiche Metaphern entwickelt, die sich nach [LaVio00] in fünf Kategorien einordnen lassen:

- Physikalische Bewegungen, wobei die Bewegung des Nutzers im Raum die Navigation ermöglicht.
- Manuelle Blickpunktmanipulation, z.B. durch „Ergreifen“ von Punkten im Raum.
- Steuerung und Lenkung, was durch Blicke, Kopf- oder Handpositionen geschehen kann.
- Zielgerichtete Navigation durch Auswahl eines Zielpunktes, wobei das System – zum Beispiel durch Teleportierung – die sprunghafte Änderung des Blickpunktes vornimmt.
- Routenplanung, wobei der Nutzer einen Navigationspfad spezifiziert, entlang dessen die Applikation ihn führt.

Für reine Desktop-VR-Umgebungen kommen als Navigationstechniken nur die beiden letztgenannten in Frage, weil immersive bzw. hochdimensionale Eingabegeräte nicht zur Verfügung stehen.

Für die aktive Navigation in virtuellen Umgebungen wurden verschiedene Interaktionstechniken, vor allem *Gehen* und *Fliegen* sowie abgeleitete Varianten, wie z.B. die beschleunigungsabhängige Szenenansicht beim Fliegen (*Speed-coupled flying*, [Tan01]) entwickelt. Auch das *Examinieren* (*Orbital Mode* [Mine95]) ist eine geläufige Navigationsvariante, wobei die Szene relativ zu einem fixen Punkt transformiert wird. Mit *Glances* [Pierc99] wurde eine flüchtige Navigationsmethode vorgestellt, wobei der virtuelle Kopf eines Nutzers an einer bestimmten Position in der virtuellen Umgebung temporär (so lange ein Eingabegerät betätigt wird) gedreht wird. Bei all diesen Varianten kommen jedoch keine Widget-Geometrien, sondern statt dessen 3- oder 6-DOF-Eingabegeräte zum Einsatz, weshalb sie hier nicht näher betrachtet werden sollen.

Gleiches gilt für das *Teleportieren* oder andere *Punkt-zu-Punkt-Navigationstechniken*, die zwar Widgets zur Zielpunktauswahl (*direction/target selection* [Bowma99a]) oder Bestimmung der Geschwindigkeit (*velocity/acceleration selection* [Bowma99a]) einsetzen, welche jedoch nicht speziell für diese Techniken entwickelt wurden, da es sich in der Regel um Schaltknöpfe, Wertgeber oder Auswahlmenüs handelt. Deshalb wird in der ersten innerhalb der Tabelle 9 aufgeführten Navigationskategorie **Direkte Blickpunktselektion** auch auf Menüwidgets verwiesen. Die in [Macki90] vorgestellte *Point-of-Interest-Navigation* erlaubt die Selektion eines entfernten Objektes / Szenenbestandteils mit der Maus, wobei eine Strahlgeometrie eingeblendet wird, die sich dann noch genauer auf der Oberfläche positionieren lässt, bevor der Blickpunkt mit einer logarithmischen Beschleunigungsfunktion gewechselt wird. Dieses und die im folgenden beschriebenen Widgets werden in Tabelle 10 gegenübergestellt.

Bei Navigationstechniken zum **Geführten Transport** (geführte Punkt-zu-Punkt-Verbindungen) kommen je nach Metapher verschiedene geometrische Ausprägungen zum Einsatz, z.B. ein Stuhl beim *Flying Chair* [Bowma98] oder ein Teppich bei einer VR-Attraktion in einem Themenpark [Pausc96]. Auch die unter 3.3.3.2 vorgestellten Metaphern Fahrstuhl, Hebebühne, Schienenfahrzeug oder Rutsche zählen zu dieser Kategorie.

Mit Erfolg wurden Miniaturdarstellungen zur Orientierung und Navigation in virtuellen Welten eingesetzt. Zur Navigation in VRML-Welten wird in [Elvin97] das Konzept der *Worldlets* vorgeschlagen. Hierbei handelt es sich um kleine, interaktive 3D-Vorschauansichten, durch die ein Nutzer schnell zu den entsprechenden Blickpunkten navigieren kann. Durch die Szenenvorschau sind die Blickpunktauswahltechniken auf Menübasis überlegen. *Worlds in Miniature (WIM)* [Stoak95] sind 3D-Miniaturlandkarten, die der Nutzer in der Hand hält, um innerhalb eines Ausschnittes der „großen“ virtuellen Welt mit der Gesamtdarstellung in Miniatur interagieren zu können. In [Pausc95] wurde die WIM-Idee erweitert, indem sich Nutzer von Ort zu Ort durch indirektes „Fliegen“ innerhalb der Miniaturszene fortbewegen können. Diesen Techniken ist gemeinsam, daß die Miniaturansichten zwar als Widget aufgefaßt werden können, sich jedoch keine Verallgemeinerungen für die Widget-Geometrien treffen lassen, da immer ein Ausschnitt oder die gesamte konkrete Szene angezeigt wird.

Anders ist dies mit den folgenden drei Techniken, die nicht nur zur Orientierung dienen, sondern teilweise auch explorativen Charakter haben oder sogar Manipulationen erlauben. Der in [Dachs97] vorgestellt *Navigator* ist ein virtueller Trackball mit einer Miniaturdarstellung eines Objektes oder einer Szene, der die aktuelle Szene überlagert und vorzugsweise am Bildschirmrand eingeblendet wird. Mit ihm lassen sich Objekte und Szenen rotieren und durch Bedienen der Anfasser Standardansichten wie hinten, vorn, oben etc. einstellen. Mit dem *Magic Mirror* [Grosj99] wurde ein intuitives Werkzeug für die Orientierung und Exploration in virtuellen Welten entwickelt, das mit der Metapher eines virtuellen Spiegels eine zweite Szenenansicht innerhalb der aktuellen Ansicht gestattet. Die Position des Spiegels läßt sich genauso wie der Zoomfaktor und die Umkehrung der Seite (Spiegelbild) einstellen. Als Erweiterung des magischen Spiegels wurde in [Stoev01] die *Through-the-Lens*-Technik vorgestellt, mit der über ein in die Szene eingeblendetes Fenster ein beliebiger anderer Szenenausschnitt dargestellt und manipuliert werden kann. Mit einem Grafiktablett läßt sich dieses Fenster in der Szene bewegen, ein Stift gestattet die Auswahl des Blickpunktes und diverse Manipulationen. Da freie Ausschnitte möglich sind, wird die Orientierung im Raum erleichtert, die Exploration von Objekten ergibt sich als weitere Anwendung.

| Manipulation der 3D-Szene | |
|--|--|
| Orientierung und Navigation | |
| Direkte Blickpunktselektion | |
| | <i>Direkte Blickpunktwahl (Point-Of-Interest)</i> |
| | Auswahl über Menüs → s. 3.4.8.1 |
| Geführter Transport | |
| | <i>Flying Chair / Carpet</i> |
| | <i>Fahrstuhl, Hebebühne, Schienenfahrzeug etc.</i> |
| Miniaturdarstellungen | |
| | <i>3D-Vorschau (Worldlets)</i> |
| | <i>Miniaturwelten (Worlds In Miniature)</i> |
| | <i>Navigator</i> |
| | <i>Magic Mirror</i> |
| | <i>Through-the-Lens</i> |
| Steuerung der Szenenpräsentation (s. 3.4.6) | |

Tabelle 9: Klassifikation der Widgets zur Orientierung und Navigation

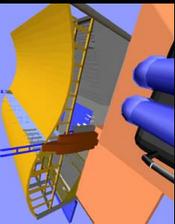
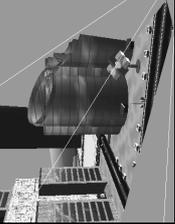
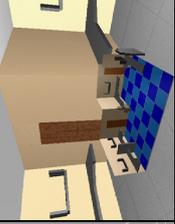
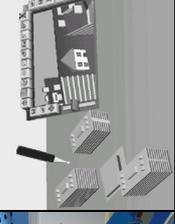
| Widgets zur Orientierung und Navigation | | Miniaturdarstellungen | | | | | | |
|---|---|---|---|---|---|--|---|------------------|
| | | Direkte Blickpunktselektion | Geführter Transport | 3D-Vorschau (Worldlets) | Miniaturlwelten (Worlds in Miniature) | Navigator | Magic Mirror | Through-the-Lens |
| | Direkte Blickpunktselektion (Point of Interest) | Geführter Transport (Flying Chair u.a.) | 3D-Vorschau (Worldlets) | Miniaturlwelten (Worlds in Miniature) | Navigator | Magic Mirror | Through-the-Lens | |
| Kurzbeschreibung | Auswahl eines Interessenpunktes mit Maus, Justierung der Orientierung und Logarithm. Bewegung dorthin | Geführte Punkt-zu-Punkt-Navigation mit einem (teilweise sichtbaren) Transportmittel | Kleine, interaktive 3D-Vorschauansichten zur Blickpunktnavigation | Nutzerinteraktion und Navigation mit eingebundenen Miniaturansichten der gesamten Szene | Indirekte freie & achsenabhängige Objekt-/Szenenrotation durch Miniaturansichten | Ein virtueller Handspiegel wird in der Szene bewegt und stellt eine zweite Ansicht dar | Darstellung eines zweiten Blickwinkels innerhalb eines mit einem Tabletgeführten Fensters | |
| Bild | kein Bild |  |  |  |  |  |  | |
| Referenz | [Mack190] | z.B. [Bowma98] | [Elvin97] | [Stoak95] | [Dachs97] | [Grosj99] | [Stoey01] | |
| für immersive Systeme | ja | ja | ja | ja | ja | ja | ja | |
| für nicht-immersive Syst. | ja | ja | ja | eher nicht | ja | ja | ja | |
| Eingabegerät | 2 DOF | 6 DOF | 2 DOF | 6 DOF | 2 DOF | 2 DOF, auch 6 DOF | 6 DOF Tablet | |
| Dimensionalität der Interaktionsaufgabe | 2D | 1D - 6D | 2D | 3D - 6D | 2D | 2D - 6D | 6D | |
| Geometrische Ausprägung / Widget | Auswahlstrahl, Feedback auf gewähltem Objekt (z.B. Kreis) | Transportmittel | Miniaturen der unterschiedlichen Blickpunkte (wechselnde Geometrie) | Szenenminiatur (wechselnde Geometrie) | Virtueller Trackball mit Anfassern für Standardansichten | Spiegel mit Handgriff | Fensterrahmen mit Buttons | |
| Darstellungsaufwand | gering | mittel | hoch | hoch | mittel | hoch | hoch | |
| Kommentare / Besonderheiten | Eigentlich keine echte Widget-sondern nur Feedbackgeometrie | Bei 6DOF-Gerät dient Transportmittel nur als Metapher, bei Blickpunktswahl (1D) auch keine Fkt. | Ließe sich auch der direkten Blickpunktselektion zuordnen (mit Vorschauzeilen) | Keine fixe Widget-Geometrie vorhanden | Auch als Spezialform der Virtual Sphere zu sehen, aber nicht objektzentriert | Ein Tablet wäre gut als Eingabegerät geeignet. Auch für Exploration. | Eignet sich auch für Manipulationstechniken und explorative Aufgaben | |

Tabelle 10: Übersicht der Widgets zur Orientierung und Navigation

3.4.6 Widgets zur Steuerung der Szenenpräsentation

Widgets zur Steuerung der Szenenrepräsentation werden üblicherweise in Autorenwerkzeugen zur Festlegung von Beleuchtungssituationen, Blickpunkten, Kamerafahrten und auditiven Szeneneigenschaften zu finden sein und eher selten in Anwendungsprogrammen für den Endnutzer. In der Tabelle 11 werden die Widgets klassifiziert, während Tabelle 12 die wichtigsten Vertreter in Kurzform beschreibt und vergleicht.

Zur Gruppe der **Lichtmanipulation** gehören Widgets, mit denen sich direktionale Lichtquellen, Punktlichtquellen und Spots definieren lassen. Innerhalb des *Open Inventor Toolkits* [Strau92] wurden entsprechende Widgets realisiert. Dabei wird die Position einer Lichtquelle über 2D-Schiebereglern (*Plane Sliders*) festgelegt, die dynamisch eingeblendet werden. Die Orientierung kann über direkte Bewegung von Richtungspfeilen vorgenommen werden, während bei Spots die Größe des Strahlkegels über einen entsprechenden Manipulator festgelegt wird.

In der Gruppe **Kameramanipulation** sind Widgets enthalten, mit denen explizit einfache Parameter einer virtuellen Kamera, aber auch komplexere Schwenks und Animationen festgelegt werden können. In [Kleme97] wird eine konfigurierbare, constraint-basierte Kamerasteuerung für 3D Visualisierungswerkzeuge vorgestellt. Obwohl es sich hierbei um eine komplexere Architektur handelt, die z.B. auch physikalische Gesetzmäßigkeiten berücksichtigt, entstand im Rahmen einer Designstudie auch eine visuelle Repräsentation einer Kamera mit ihren 7 Freiheitsgraden. So können über Widgets, die denen der geometrischen Manipulation ähneln, die Position (3D-Translation), Orientierung (3D-Rotation) und die Brennweite (Zoomfaktor) eingestellt werden. Ein anderer Ansatz wird in [Gleic92] beschrieben, wo Kameraparameter im Raum nicht nur direkt eingegeben werden können, sondern mit Hilfe von interaktiv gesteuerten, fixierbaren Bildpunkten Vorgaben für die resultierende Kamerabewegung festgelegt werden. So kann z.B. ein Kontrollpunkt einem Objekt zugeordnet werden. Ändert dieses seine Position in der Szene, folgt auch die Kamera diesem Objekt. Denkbar ist auch, daß die Kamera bei einer Bewegung automatisch gezoomt wird, um einen vorher mit Kontrollpunkten festgelegten Teil der Szene immer im Bild zu haben. Ebenfalls ein indirekter Weg wird in [Burtn01] gewählt, wobei sogar komplexe Kamerafahrten, Blickpunkttransitionen und sogenannte *Money-Shots* interaktiv festgelegt werden können. Interessant dabei ist, daß nicht nur vorgefertigte Animationen und Blickpunkte beschrieben werden können, sondern mit Hilfe der sogenannten Kameraoberflächen auch solche Bereiche, innerhalb derer Nutzer später den Blickpunkt selbst beschränkt verändern können.

Widgets zur **Soundsteuerung** werden in [Hoffm01] (S. 92 ff.) konzeptionell beschrieben. Damit lassen sich die verschiedensten Parameter für Soundquellen im Raum und für ihr Abstrahlverhalten ebenso einstellen, wie die Position des Hörers im Raum. Die Manipulationen lassen sich – wie bei Lichtwidgets auch – auf eine Komposition aus einfachen Widgets zur geometrischen Manipulation (Translation, Rotation, Winkel) zurückführen.

| Manipulation der 3D-Szene | |
|---|--|
| Orientierung und Navigation | (s. 3.4.5) |
| Steuerung der Szenenpräsentation | |
| Lichtmanipulation | |
| | <i>Spotlight Manipulator</i> |
| | <i>Directional Light Manipulator</i> |
| Kameramanipulation | |
| | <i>Kamerasteuerung</i> |
| | <i>Through-the-Lens Camera Control</i> |
| | <i>StyleCam</i> |
| Soundsteuerung | |
| | <i>Soundquellensteuerung</i> |
| | <i>Hörerpositionsmanipulator</i> |

Tabelle 11: Klassifikation der Widgets zur Steuerung der Szenenpräsentation

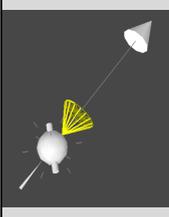
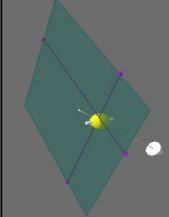
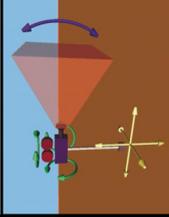
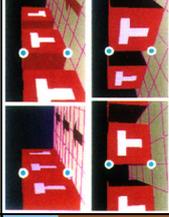
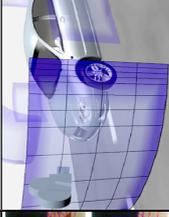
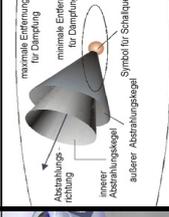
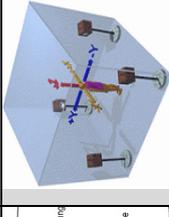
| Widgets zur Steuerung der Szenenpräsentation | Lichtmanipulation | | Kameramanipulation | | | Soundsteuerung | |
|--|---|---|--|--|--|--|---|
| | Spotlight Manipulator | Directional Light Manipulator | Kamerasteuerung | Through-the-Lens Camera Control | StyleCam | Soundquellensteuerung | Hörerpositionsmanipulator |
| Kurzbeschreibung | Verschiedene Anfassergeometrien erlauben die Positionierung, Ausrichtung und Parametrisierung der Lichtquelle | Wie Spotlight Manipulator, im Bild z.B. Positionierung innerhalb einer Ebene | Im Rahmen einer Kamerasteuerung mit Zusicherungen entstand als Designstudie eine visuelle Kamera | Manipulation der virtuellen Kamera durch direkte Manipulation des Bildes (wie durch Linse) | Gekrümmte Kameraoberflächen u. Transitionsfade erlauben vorgefertigte räumliche und zeitliche Kamerabewegungen | Interaktive Manipulation der Parameter einer Soundquelle | Positionierung des Hörers im Raum über Translationswidgets |
| Bild |  |  |  |  |  |  |  |
| Referenz | z.B. [Strau92] | z.B. [Strau92] | z.B. [Kleme97] | [Gleic92] | [Burtn01] | [Hoffm01] | z.B. [Hoffm01] |
| Geometrische Ausprägung / Widget | Lagekugel, Ausrichtungspfeil, Lichtkegel, Hilfsgeometrien | Lagekugel, Ausrichtungspfeil, Hilfsgeometrien | Kamera mit Manipulatoren für alle 7 Freiheitsgrade | Einfache Kontrollpunkte und Keyframing-Pfeile | Kameraoberflächen mit Geschwindigkeitsraster, Pfade für Transitionen | Lagekugel, Richtungs-pfeil, Abstrahlkegel, Dämpfungsringe | Translationswidgets für jede Richtung, Hörerfigur |
| Darstellungsaufwand | gering | gering | gering | gering | mittel | gering | gering |
| Änderbare Parameter | Position, Richtung, Strahlkegelgröße des Lichtes | Position, Richtung des Lichtes | Position, Blickrichtung, Brennweite der Kamera | Position, Distanz, Orientierung von Kontrollpunkten | Eingeschränkte Kameraoberflächen, Blickpunkttransitionen, Kamerafahrten | Position, Abstrahlrichtung, innerer und äußerer Abstrahlkegel, min. und max. Entfernung für Dämpfung | Position des Hörers |
| Kommentare / Besonderheiten | Unmittelbares Feedback in Form der Szenenbeleuchtung | Ist Bestandteil des <i>Spotlight Manipulators</i> , nur ohne Lichtkegeleinstellung | Klassische Kamerasteuerung | Constraint-basierte, indirekte Kamerasteuerung | Bisher noch als MAYA-Plugin realisiert und nicht als separate Widgets | Bisher noch konzeptionell, aber analog Lichtwidgets realisierbar | Bisher nur konzeptionell |

Tabelle 12: Übersicht der Widgets zur Steuerung der Szenenpräsentation

3.4.7 Widgets zur Exploration und Visualisierung

Diese Gruppe von 3D-Widgets ist relativ umfangreich, da sie Widgets zur Visualisierung wissenschaftlicher Daten und Informationsvisualisierungstechniken mit vielen Ausprägungen und Varianten umfaßt. Bei näherer Betrachtung erweisen sich jedoch viele Widgets als domänenspezifisch bzw. als reine 3D-Visualisierungen ohne direkte Manipulationsmöglichkeiten. Die in der Tabelle 13 klassifizierten Widgets zur Exploration und Visualisierung sind primär für nicht-immersive Systeme entwickelt worden und deshalb mit 2D-Eingabegeräten bedienbar. Tabelle 14 beschreibt und vergleicht die einzelnen Widget-Ausprägungen miteinander.

Zur Gruppe der Widgets zur Geometrischen Exploration gehören solche, die auf beliebige Objekt- oder Szenengeometrien anwendbar sind. *Cutting Plane* Widgets finden in zahlreichen Objekten Anwendung, insbesondere dort, wo es um Volumenvisualisierungen geht und das Innere eines Objektes betrachtet werden soll. Die *3D Magic Lenses* [Viega96] bieten noch mehr Freiheit, weil sie innerhalb eines kubischen Schnittvolumens das Innere eines Objektes oder auch abstrakte Ebenen (beim Menschen z.B. Skelett, Blutbahnen, Lymphsystem, Nerven etc.) darstellen können. Die schon unter 3.4.5 als Werkzeuge zur Orientierung erwähnten Techniken *Through-the-Lens* [Stoev01] und *Magic Mirror* [Grosj99] können ebenfalls als Explorationswerkzeug zur Objektinspektion verwendet werden, da sie z.B. Objekte, die sich vor dem Spiegel oder Linse befinden, in Teilen oder ganz ausblenden können. Dadurch kann auch in das Innere von Objekten gesehen werden.

Die folgenden drei Gruppen lassen sich auch als Widgets zur Informationsvisualisierung subsumieren und stellen in sich geschlossene Darstellungs-Widgets dar, die normalerweise keinen Einfluß auf andere Objekte der Szene haben. Diese Visualisierungen tragen natürlich auch explorativen Charakter und bieten entsprechende Interaktionsmöglichkeiten. Hier kann nur eine Auswahl wesentlicher Techniken vorgenommen werden, da es viele Abwandlungen der Techniken gibt. Ein guter Überblick zu Verfahren der dreidimensionalen Informationsvisualisierung wird z.B. in [Young96] gegeben.

Zur Gruppe der Hierarchievisualisierungen gehören die *Information Cubes* [Rekim93], bei denen die Metapher verschachtelter Boxen zur Darstellung genutzt wird. Dabei handelt es sich um eine geschachtelte Hierarchie semitransparenter dreidimensionaler Würfel, die jeweils beschriftet sind. Subhierarchien, also tiefer gelegene Würfel, lassen sich durch Strahlselektion herauszoomen. Beim *Cone Tree* [Rober91] (vertikal) bzw. *Cam Tree* (horizontal) handelt es sich um einen dreidimensionalen Baum, von dessen Wurzelement aus die jeweiligen Kindelemente auf einer Ebene kreisförmig angeordnet werden, woraus sich mehrere Kegel ergeben. Mit dieser Darstellung kann der Nutzer interagieren, gewählte Elemente werden in den Vordergrund gedreht, der Kontext innerhalb des Baumes bleibt durch die Perspektivwirkung und Transparenz der Kegel erhalten. Der innerhalb des Projektes *Lyberworld* [Lyberworld@] entwickelte Spiralbaum (auch *Visual Tree*) [Veit97] stellt eine Weiterentwicklung der *Cone Trees* dar, bei der sich eine höhere Zahl von Kindelementen durch spiralförmige Windung kompakter darstellen läßt. Anfassersorgen für das axiale und radiale Expandieren der Spirale, um Hierarchieelemente überhaupt wahrnehmen und ihre Texte besser lesen zu können. Während diese drei Widgets ihren Hauptzweck in der Exploration von Hierarchien sehen, wurde mit den *Collapsible Cylindrical Trees* [Dachs01c] ein Verfahren insbesondere zur schnellen Interaktion und Navigation innerhalb einer mittelgroßen Hierarchie vorgestellt. Zylinderförmige Menüs fahren bei Aktivierung von Untermenüs teleskopartig kleinere Subzylinder aus, während die anderen Menüzyylinder der Hauptebene axial skaliert werden. Damit bleiben alle Top-Hierarchieebenen gleichzeitig sichtbar und auswählbar. Mehrere der entwickelten 3D-Dateibrowser (s. Abschnitt 3.1.4.2) verfolgen ähnliche Prinzipien wie die der eben vorgestellten Hierarchievisualisierungstechniken, d.h. entweder die radiale (3DOSX für Mac OS X

[Chin02]) oder geschachtelte Anordnung (*SGI File System Navigator* [SGI-FSN@]) von Dateien und Verzeichnissen, zumeist auf einer Ebene.

Die an der *University of New Brunswick* entwickelten Systeme *GraphVisualizer3D* [GV3D@] und das Nachfolgesystem *NestedVision3D* [Parke98] sind prominente Beispiele für die Kategorie der dreidimensionalen Visualisierungen von Graphen. Dabei werden Netzwerkdiagramme mit vielfältigen Beziehungen und Interaktionsmöglichkeiten in Detail- und Kontext-Darstellung für die Beispieldomäne der Softwarevisualisierung eingesetzt. Mehrere Widgets erlauben die Einstellung diverser Parameter und unterstützen die Navigation innerhalb der Graphen. Für die dreidimensionale Visualisierung von Webstrukturen gibt der Übersichtsartikel [Benfo99] einen guten Überblick über existierende Visualisierungen von Webseitenhierarchien, Beziehungen zwischen diesen und ihre Verknüpfungsgeschichte. Da die meisten Methoden jedoch keinen Widget-Charakter tragen oder nur 2D-Widgets zur Steuerung verwenden, werden sie hier nicht extra aufgeführt. Aus reiner Visualisierungssicht wäre diese Gruppe jedoch beträchtlich größer.

Neben der Visualisierung von Hierarchien und Graphen machen sich auch Widgets zur Visualisierung von 2D-Daten und Dokumenten die Wirkung der Perspektive zur Darstellung abstrakter, nicht-geometrischer Daten zunutze. Auf der 3D-Informationswand *Perspective Wall* [Macki91] können beliebige, üblicherweise thematisch gruppierte, textuelle oder bildliche Informationen dargestellt werden. Gewählte Objekte werden automatisch ins Zentrum gebracht, die nach hinten gerichteten Seiten bieten Kontextinformationen. Ganz ähnlich ist es mit der in [Fairc93] beschriebene *Sphere Visualization*, nur daß die Nutzung einer Kugel eine noch bessere Perspektivwirkung erzeugt und sich auch mehrere Kugeln schachteln lassen. Mit der *Document Lens* [Rober93b] wurde eine 3D-Visualisierungstechnik für große rechteckige Präsentationen (z.B. Textdokumente) entwickelt, mit der es möglich ist, Details zu betrachten und trotzdem den Kontext zu erkennen.

Ebenfalls der Ausnutzung von Perspektive verpflichtet sind Technologien, die mehrere Webseiten (oder andere Dokumente) gleichzeitig darstellen. Beim *Web Book* [Card96] wurde dafür die Buchmetapher zur Präsentation von Webseiten genutzt. Der *CubicEye Viewer* [CubicEye@] der Firma 2ce, Inc. ist ein vollständiger Webbrowser, der Webseiten auf den 5 sichtbaren Innenseiten eines Würfels darstellt und diverse Navigationsmethoden zum schnellen Wechsel und Betrachten der Seiten bereitstellt. Obwohl beide Widget-Formen speziell für Webseiten entwickelt wurden, lassen sich sowohl die Buch- als auch die Würfelmetapher auf beliebige Dokumente verallgemeinern.

Schließlich gibt es innerhalb der Gruppe von Widgets zur Wissenschaftlichen Visualisierung mehrere Spezialentwicklungen für unterschiedliche Anwendungsdomänen. Hier werden exemplarisch Widgets zur Steuerung der Ergebnisdarstellung numerischer Strömungssimulationen vorgestellt, die sich in verschiedenen Anwendungen einsetzen ließen. An der Brown University wurde im Rahmen des Forschungsprojektes *Virtual Windtunnel* [VWT@] eine Bibliothek von Widgets [Brown@] entwickelt, darunter das *Rake*-Werkzeug zur Kontrolle der Platzierung mehrerer Stromlinien oder Partikelpfade, *Plane* als Steuerwidget für Emitterteilchen auf einer Ebene sowie *Hedgehog* zur Kontrolle eines 3D-Vektorfeldes. Diese Widgets enthalten jeweils Anfasser zur Positionierung, Skalierung und Orientierung sowie Schieberegler zur Einstellung verschiedener Werte, wie z.B. der Auflösung [Hernd94b].

| Exploration und Visualisierung | |
|--|---|
| Geometrische Exploration | |
| | <i>Interaktive Schnittflächen (Cutting Plane)</i> |
| | <i>3D Magic Lenses</i> |
| | <i>Through-the-Lens</i> → s. 3.4.5 |
| | <i>Magic Mirror</i> → s. 3.4.5 |
| Hierarchievisualisierungen | |
| | <i>Information Cube</i> |
| | <i>Cone Tree</i> |
| | <i>Spiralbaum (Visual Tree)</i> |
| | <i>Collapsible Cylindrical Trees</i> |
| | <i>Dateibrowser 3DOSX u. SGI-FSN</i> |
| Visualisierungen von Graphen | |
| | <i>GraphVisualizer3D, NestedVision3D u.a.</i> |
| Visualisierung von 2D-Daten und Dokumenten | |
| | <i>Perspective Wall</i> |
| | <i>Sphere Visualization</i> |
| | <i>Document Lens</i> |
| | <i>3D-Buch (Web Book)</i> |
| | <i>3D-Webbrowser (CubicEye Viewer)</i> |
| Wissenschaftliche Visualisierung | |
| | <i>Stromlinienwidget (Rake, Plane), Hedgehog u.a.</i> |

Tabelle 13: Klassifikation der Widgets zur Exploration und Visualisierung

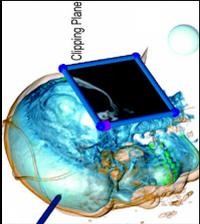
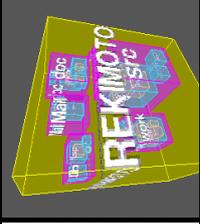
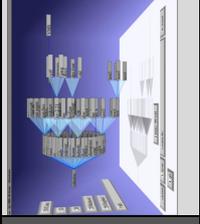
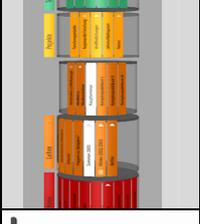
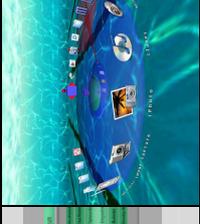
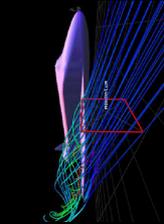
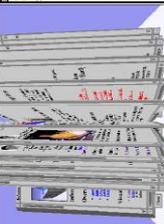
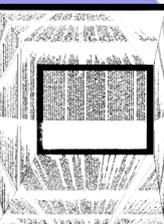
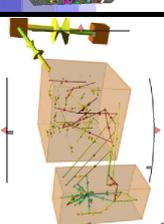
| Widgets zur Exploration und Visualisierung | Hierarchievisualisierungen | | | | | | |
|--|---|---|---|---|--|--|---|
| | Interaktive Schnittflächen (Cutting Plane) | 3D Magic Lenses | Information Cube | Cone Tree | Spiralbaum (Visual Tree) | Collapsible Cylindrical Trees | Dateisystem-Browser (Bsp. 3DOSX) |
| Kurzbeschreibung | Manipulierbare Schnittflächen erlauben Volumenschnitte auch mit abstrakten Daten | Transparente Geometrien für Skelettsichten, Falschfarben u.ä. | Schachtelung semi-transparenter 3D-Würfel mit Beschriftung. Subhierarchien lassen sich herauszoomen | Kegelförmige Baumvisualisierung, wobei selektierte Subbäume / Elemente nach vorn gedreht werden | Variante eines horizontalen Cone Trees mit änderbarer Spiraldarstellung der Kindelemente | Schnelle Navigation mit Zylindermenüs, die teleskopartig ausklappen, wobei Kontext sichtbar bleibt | Dateien einer Ebene als Icons auf Scheibe dargestellt, schnelle Navigation in untere Ebenen, alle bleiben immer sichtbar. |
| Bild |  |  |  |  |  |  |  |
| Referenz | z.B. [Kniss02] | [Vieg96] | [Rekim93] | [Rober91] | [Veit97] | [Dachs01c] | z.B. [Chin02] |
| Geometrische Ausprägung / Widget | Rahmen, Anfasser und evtl. Buttons | Halbtransparenter Quader mit Anfassern | Teiltransparente Würfel für jedes Hierarchieelement | Textplättchen u. transparente Kegel, Manipulationsbuttons | Kegel und Plättchen, Anfasser zur Steuerung der axialen u. radialen Expansion | Zylinder mit Plättchen | „Platten“ für Verzeichnisse, 3D-Icons für Dateien, Box bei vielen Elementen |
| Darstellungsaufwand | gering - mittel | mittel | mittel | hoch | hoch | mittel | mittel |
| Struktur und Art der visualisierten Daten | Objektgeometrien + evtl. abstrakte Sicht | Objektgeometrien + evtl. abstrakte Sicht | Hierarchien, Text | Hierarchien, Text | Hierarchien, Text | Hierarchien, Text+Icons | Hierarchien, Icons |
| Menge darstellb. Daten | - nicht anwendbar - | - nicht anwendbar - | mittel | groß | groß | mittel | mittel |
| Detail- und Kontext-Unterstützung | durch Clipping Plane werden innere Bestandteile sichtbar | durch Box lässt sich in das Innere eines Objektes sehen | durch Perspektive, Würfelskalierung u. Transparenzen | durch Perspektive | durch Perspektive und Spiraldrehung | durch Perspektive u. horizontale Zylinder-skalierung | durch Perspektive und Transparenzen |
| Kommentare / Besonderheiten | Volumenschnitte (Ebenen) können auch abstrakte / überlagerte Daten darstellen | Volumensegment kann auch abstrakte / überlagerte Daten darstellen | Kann schnell unübersichtlich werden, weniger klar. Transparenzen evtl. problematisch. | Horizontale und vertikale Varianten. Vorrangig Exploration | Einstellung der Spiralen bei unregelmäßigen Bäumen mühselig. Vorrangig Exploration. | Immer horizontal. Weniger Exploration, eher für schnelle Interaktionen. | Kann auch als verschachtelte Ringmenüs aufgefaßt werden |

Tabelle 14: Übersicht der Widgets zur Exploration und Visualisierung

| Widgets zur Exploration und Visualisierung (Forts.) | Visualisierung von 2D-Daten und Dokumenten | | | | | | Wissensch. Visualisierung |
|---|--|---|--|--|---|---|---|
| | Perspective Wall | Sphere Visualization | Document Lens | 3D-Buch (z.B. WebBook) | 3D-Webbrowser (z.B. Cubic Eye) | Stromlinien-Widgets | |
| Kurzbeschreibung | 3D-Graphen für Visualisierung von komplexer Software, zahlreiche Interaktions- und Visualisierungsoptionen | Multimediaobjekte und ihre Beziehungen werden auf Kugel dargestellt, Objekt des Interesses ist im Zentrum | Große rechteckige Informationen werden auf perspektivische Geometrie projiziert. Linse erlaubt Detaildarstellung | Interaktives 3D-Buch mit Webseiten, einfaches u. schnelles Durchblättern möglich | Webbrowser in Würfelform, auf dessen 5 Innenseiten Webseiten angezeigt werden. Diverse Interaktionsmöglichkeiten. | Mit den Widgets <i>Rake & Plane</i> lassen sich Emittierer im Raum festlegen und Stromlinienvisualisierungsparameter einstellen |  |
| Bild |  |  |  |  |  |  | |
| Referenz | [Parke98] | [Fairc93] | [Rober93b] | [Card96] | [CubicEye@] | z.B. [WWT@] | |
| Geometrische Ausprägung / Widget | Verschiedene Knoten, Kanten, Navigationshilfen etc. | Kugel | Rechteckiger Linsenrahmen + Wände | Buchseiten und Steuer-Widgets | Würfel ohne Vorderseite | Emittierzylinder bzw. -rechteck, dazu diverse Anfasser | |
| Darstellungsaufwand | hoch | gering - mittel | gering - mittel | hoch | gering, f. Seiten hoch | gering | |
| Struktur und Art der visualisierten Daten | Große verschachtelte Graphen und Netze, SW-Objekte als Knoten+Beziehungen | Gruppen, Objekticons | Rechteckfläche mit beliebigen Informationen (z.B. Text) | Kollektion von Webseiten | üblicherweise 5, max. 25 Webseiten | Stromlinien | |
| Menge darstellb. Daten | sehr groß | klein - mittel | groß | groß | mittel | gering | |
| Detail- und Kontext-Unterstützung | Perspektive, Transparenzen und zahlreiche Navigations-techniken | durch Perspektive der Kugel | durch Perspektive der Wände | durch Perspektive | durch Perspektive der Würfelseiten | - <i>nicht anwendbar</i> - | |
| Kommentare / Besonderheiten | Eigentlich kein Einzel-Widget, sondern hochentwickeltes, komplexes Visualisierungssystem | Natürliche Fischaugenansicht. Auch verschachtelte Kugeln für untergeordnete Objekte. | Setzt voraus, daß sich Dokumente auf einer großen Ebene befinden. Eher selten angewendet. | Mehrere gute Interaktionsmöglichkeiten erlauben das schnelle Browsen innerhalb einer Webseitenkollektion | Eigentlich komplettes Programm, aber auch als Widget realisierbar | Innerhalb des Virtualen Windtunnels gibt es mehrere Widgets zur wissenschaftlichen Visualisierung | |

3.4.8 Widgets zur Anwendungskontrolle

Widgets zur Anwendungskontrolle (auch Kontroll- oder Bedienelemente genannt) stellen eine wichtige Gruppe dar, da sie entscheidende Bausteine für dreidimensionale Anwendungen sind. Das liegt an ihrem Verallgemeinerungspotential und ihren vielfältigen Einsatzmöglichkeiten genauso wie an der Bekanntheit der meisten Konzepte aus 2D-GUIs. Während zur Objektmanipulation und Blickpunktänderung zahlreiche Forschungsarbeiten und Widgets existieren, stellen Interaktionselemente zur Anwendungskontrolle nach wie vor den am wenigsten erforschten Bereich für 3D-GUIs dar ([Hand97], [LaVio00], [Encar02]), sieht man einmal von zahlreichen Verwendungen von Schiebereglern, Schaltknöpfen und 3D-Menüs ab. Zu den typischen Aufgaben der Anwendungskontrolle zählen die Auswahl von Elementen aus einer vorgegebenen Menge, Zustandsänderungen und die Einstellung verschiedenartigster Parameter [LaVio00].

An diesen Zielen orientiert sich auch die in Tabelle 15 vorgenommene Klassifikation der Widgets, die alle primär für nicht-immersive Systeme entwickelt wurden und somit mit 2D-Eingabegeräten bedienbar sind. Da ihre optische Ausprägung und Funktionalität weitestgehend aus dem 2D-Bereich bekannt sind, wurde an dieser Stelle auf eine zusätzliche Tabelle verzichtet, welche die Widgets gegenüberstellt. Unter [CONTIGRA@] können jedoch die konkreten Komponentenspezifikationen dieser Kategorie online betrachtet werden. Der Gruppe der Widgets zur **Menüselektion** wurde ein extra Unterkapitel (3.4.8.1) gewidmet, da in dieser Rubrik interessante Ausprägungen zu finden sind, die stark von zweidimensionalen Konzepten abweichen.

Die Widgets der Kategorie **Zustandsänderung** lassen sich auch als diskrete Wertgeber verstehen, da ein Umschalten zwischen n verschiedenen Zuständen ebenso als eine Auswahl aus einer n -elementigen Wertemenge aufgefaßt werden kann. Einzelzustände können einfach nur aktiviert werden (Drucktaster, einzelne Menüeinträge), für zwei oder mehrere Zustände existieren verschiedene Schalterausprägungen.

Bei der Gruppe der **Kontinuierlichen Wertgeber** können aus vorgegebenen Wertebereichen entweder skalare Werte ausgewählt werden oder gleichzeitig multiple Werte, in der Regel zweidimensionale oder dreidimensionale Vektoren. So lassen sich Farbwähler auch zu dieser Kategorie zählen, auch wenn sie aufgrund der fehlenden Verallgemeinerung erst in der folgenden Kategorie der **Speziellen Werteeingaben** enthalten sind. Ein *Color Component Slider* (für 3D-Grafik erstmals in [Conne92] beschrieben) kann neben RGB-Komponenten natürlich auch für andere Farbmodelle, z.B. HSV oder CMYK eingesetzt werden. Das geometrische Modell des RGB-Farbraumes stand Pate für den *Color Cube* [VRMLWidgets@] und in Kugelform auch für die in [Kiyok00] beschriebene *Color Sphere*. Auch Texteingabe-Widgets zählen zu dieser Gruppe, wobei Desktop-VR-Systeme gegenüber immersiven VR-Umgebungen ohne Tastaturen hier deutliche Vorteile aufweisen.

Widgets der Kategorie **Container** lassen sich als Hilfsmittel zur Realisierung von *Action Spaces* einsetzen. Bisher existieren nur wenige Arbeiten und Ansätze zu komplexeren Widgets, die beispielsweise eine Containerfunktion für andere Widgets realisieren oder gar eine komplette Applikationsmetapher kommunizieren. Naheliegende Konzepte aus dem 2D-Bereich, wie Fenster, Paneele oder Werkzeugpaletten, zählen zu dieser Gruppe. In [Feine93] werden zweidimensionale Fenstertechniken erstmals für dreidimensionale Welten eingesetzt. In [Linde99] werden mit *hand-held windows* Fenster vorgestellt, die an ein Grafiktablett als Eingabegerät in der nicht-dominanten Hand gekoppelt sind. Mit *Toolspaces* werden in [Pierc99] Werkzeugcontainer beschrieben, die einen Behälter für 3D-Widgets und andere Objekte darstellen, der per *drag and drop* bestückt und entleert werden kann.

| Anwendungskontrolle | |
|--|--|
| Zustandsänderung / Diskrete Wertgeber | |
| Aktivierung | <i>Drucktaster (Push-Button)</i> Über Menüs → s. 3.4.8.1 |
| Zwei Zustände | <i>Druckschalter (Push Switch)</i> <i>Kippschalter (Toggle Switch & Rocker Switch)</i> <i>Schiebeschalter (Slide Switch)</i> <i>Check Box</i> |
| Multiple Zustände | <i>Radio Button</i> <i>Drehschalter (Dial Switch)</i> <i>Multiple Check Boxes</i> Über Menüs → s. 3.4.8.1 |
| Kontinuierliche Wertgeber | |
| Skalare Werte | <i>Drehrad (Knob), Drehknopf (Dial)</i> <i>Wippschalter (Up-Down Switch)</i> <i>Schieberegler (Slider)</i> <i>Rollbalken (Scroll Bar)</i> |
| Multiple Werte | <i>2D-Schieberegler (Plane Slider)</i> Farbwähler → s.u. |
| Spezielle Werteeingaben | |
| Texteingabe | |
| Farbwähler | <i>Farbkomponentenwähler</i> <i>Color Cube</i> <i>Color Sphere</i> |
| Menüselektion | (s. 3.4.8.1) |
| Container | |
| | <i>Fenster</i> <i>Panel</i> <i>Toolspaces</i> |

Tabelle 15: Klassifikation der Widgets zur Anwendungskontrolle – ohne Menüs

3.4.8.1 Widgets zur Menüselektion

Menüs sind eine Präsentationsform von Auswahlmöglichkeiten. Texte, Bilder, Objekte, Optionen, Befehle – so vielfältig wie die selektierbaren Varianten sind auch die geometrischen Ausprägungen im Dreidimensionalen, wobei jedoch einige Grundtypen erkennbar sind. Während verschiedene Widgets sich relativ gut aus dem Zweidimensionalen übernehmen und anpassen lassen, sind 2D-Menüs nicht einfach in den Raum zu transformieren [Hernd94a]. Das hat primär mit Textlesbarkeit und dem Ebenencharakter von Menüs zu tun, jedoch auch mit dem verschenkten ikonischen Potential. Warum sollte man ein Objekt aus einer textuellen Liste im 3D-Raum auswählen, wenn es in einer Miniaturrepräsentation bereits in der Szene zu sehen sein könnte? Als Antwort wurden zahlreiche 3D-Menütechniken entwickelt, die in der Tabelle 16 klassifiziert sind. Tabelle 18 beschreibt und vergleicht abschließend die wichtigsten Vertreter in Kurzform.

Bei der Gruppe der **Temporären Optionsmenüs** handelt es sich um einfache Werkzeug- und Optionsmenüs mit einer festgelegten und kleinen Zahl an Alternativen, die größtenteils nur kurzzeitig gezeigt werden und sich schnell aktivieren lassen. Die Auswahldimension ist – wie beim *Rotary Tool Chooser* [ISAAC@] – teilweise sogar nur 1D oder auch 2D, um eine schnelle Selektion zu erlauben. Hierarchien werden nicht unterstützt, durch die begrenzte Zahl der Punkte sind diese Menüs auch schwer zu verallgemeinern oder zu erweitern. Interessant ist, daß für diese Gruppe mit Ausnahme von Popup-Menüs ausschließlich neue 3D-Metaphern entwickelt wurden, die keine Äquivalenz im Zweidimensionalen besitzen.

Zur Kategorie der **Einzelmenüs** zählen zumeist permanent sichtbare Auswahlmenüs in Listenform, die eine feststehende, größere Anzahl von Auswahlmöglichkeiten in einer Hierarchiestufe anbieten. Sie sind damit relativ leicht generalisierbar und teilweise auch auf Menühierarchien erweiterbar. Dies gilt insbesondere für Listen- und Drop-Down-Menüs, deren Konzept aus dem Zweidimensionalen entlehnt wurde. Das *Rondell* [Preim97] beseitigt die großen Nachteile dieser Menüformen, nämlich die schlechte Textlesbarkeit und die freie Anordnung im Raum und ist trotzdem textuellen Einträgen für bestimmte Kategorien verpflichtet. *Ringmenü* [Liang94], *Chooser* [Abel95] und 3D-Paletten bieten konkrete 3D-Objektauswahl an und nutzen damit das Potential der dritten Dimension. Während sich alle Ausprägungen gut mit der Maus bedienen lassen, sind gerade 3D-Paletten als Form komplexer Auswahlpaneele häufig an die zweihändige Interaktion mit Tablett und Stift gebunden. Werkzeug- und Objektpaletten (z.B. in [Butte92], [ISAAC@], [PIP@], [Serra95], [Billi97], [Kiyok00], [Encar00]) spielen eine ähnliche Rolle wie Dialogboxen, da sie auch andere Interaktionselemente, Tools und 3D-Objekte zur Selektion enthalten.

Die 3D-Widgets für **Menühierarchien** sind einerseits klassischen zweidimensionalen hierarchischen Menüsystemen nachempfunden (*Hands-off Menu* [Darke94], *Hierarchical Pop-Up Menu* [VWT@] oder eines der ersten VR-Pull-Down-Menüsysteme in [Jacob92]). Das bedeutet Textorientierung, meistens bildschirmparallele Anzeige und gute Generalisierbarkeit. Andererseits sind die Menüformen *Tool Rack* [Serra95] und *3D Pie Menu* [Deeri95] eher hierarchische Ableitungen von dreidimensionalen Einzelmenüs, wobei das ikonische Potential besser ausgenutzt wird. Auch einige der unter 3.4.7 beschriebenen Hierarchievisualisierungen lassen sich zu dieser Widget-Kategorie zählen. Ihr primäres Ziel ist jedoch die Exploration einer Hierarchie, während Menütechniken auf den Vorgang der Auswahl fokussieren.

| Anwendungskontrolle | |
|----------------------------|---------------------------------------|
| Menüselektion | |
| Temporäre Optionsmenüs | |
| | <i>Rotary Tool Chooser</i> |
| | <i>Menükugel (Menu Ball)</i> |
| | <i>Command & Control Cube</i> |
| | <i>Popup Menu</i> |
| | <i>Tool Finger</i> |
| | <i>TULIP</i> |
| Einzelmenüs | |
| | <i>Ringmenü</i> |
| | <i>Listenmenü (Floating Menu)</i> |
| | <i>Drop-Down-Menü</i> |
| | <i>Rondell, Drehbühne mit Stelen</i> |
| | <i>Chooser Widget</i> |
| | <i>3D-Palette, Primitive Box u.a.</i> |
| Menühierarchien | |
| | <i>Hands-off Menu</i> |
| | <i>Hierarchical Pop-Up Menus</i> |
| | <i>Tool Rack</i> |
| | <i>Tortenmenü (3D Pie Menu)</i> |
| | Hierarchievisualisierungen → s. 3.4.7 |

Tabelle 16: Klassifikation der Widgets zur Menüselektion

Für die Klassifikation von Menüs sind – ebenso wie bei anderen Widgets – alternative Systematisierungen denkbar. So wurde auch eine alternative Einteilung entwickelt, die in Tabelle 17 vorgestellt wird. Sie ist nicht an der Zahl der Menüpunkte und Einsatzzweck orientiert, sondern stellt die geometrische Ausprägung und auch den Platzbedarf in den Vordergrund. Eine solche Einteilung kann hilfreich sein, wenn eine konkrete 3D-Szene bzw. ein *Action Space* bestimmte Gestaltungsanforderungen an die Menüselektionsform stellt.

Bei den **Flächenmenüs** können Menüeinträge auf im Raum schwebenden rechteckigen Flächen bzw. Paletten angeordnet sein oder aber kreisförmig in einer Ebene. Dabei sind im Bezug auf die Sichtachse des Nutzers vertikale, horizontale und auch gekippte Anordnungen in freiem Winkel möglich. *Overlay-Menüs* stellen eine vertikale, bildebeneparallele Spezialform dar, die in verschiedenen geometrischen Formen der Szene überlagert wird (*display-fixed*). Bei **Volumenmenüs** sind Menüeinträgen auf – evtl. gedachten – geometrischen Volumenkörpern angeordnet, wobei deren Seiten entweder gleich behandelt werden können (es werden permanent Menüeinträge verdeckt) oder aber bestimmte Seiten favorisiert werden, womit sich z.B. bei **Zylindermenüs** gleichzeitig mehr Menüelemente wahrnehmen lassen. **Freie Menüformen** ordnen schließlich Menüpunkte auf eine Weise an, die keinen geometrischen Grundformen folgt.

| Anwendungskontrolle | |
|----------------------------|---|
| Menüselektion | |
| Flächenmenüs | |
| Rechteckige | <ul style="list-style-type: none"> <i>Listenmenü (Floating Menu)</i> <i>Popup Menu</i> <i>Drop-Down-Menü</i> <i>Chooser Widget</i> <i>3D-Palette, Primitive Box u.a.</i> <i>Tool Rack</i> |
| Kreisförmige | <ul style="list-style-type: none"> <i>Ringmenü</i> <i>Rotary Tool Chooser</i> <i>Tortenmenü (3D Pie Menu)</i> |
| Overlay-Menüs | <ul style="list-style-type: none"> <i>Hands-off Menu</i> <i>Hierarchical Pop-Up Menus</i> |
| Volumenmenüs | |
| Gleichmäßige Verteilung | <ul style="list-style-type: none"> <i>Menükugel</i> <i>Platonische Körper</i> <i>Command & Control Cube</i> |
| Dominante Seiten | <ul style="list-style-type: none"> <i>Rondell</i> <i>Collapsible Cylindrical Trees → s. 3.4.7</i> <i>Tool Finger</i> |
| Freie Menüformen | |
| | <i>TULIP</i> |

Tabelle 17: Alternative Klassifikation der Widgets zur Menüselektion nach Erscheinungsbild

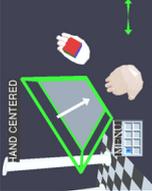
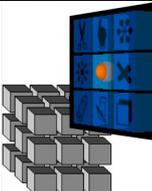
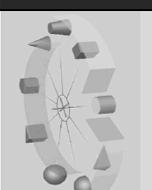
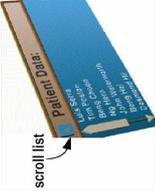
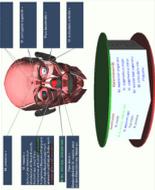
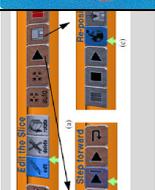
| Widgets zur Menüselektion | Temporäre Optionsmenüs | | | | | | Einzelmenüs | |
|---|---|---|---|--|---|---|---|---|
| | Rotary Tool Chooser | Menükugel (Menu Ball) | Command & Control Cube | Popup Menu | Tool Finger | TULIP | Ringmenü | Listenmenü (Floating Menu) |
| Kurzbeschreibung | Virtuelle Hand wird zur schnellen Objektselektion um eine Achse gedreht | Durch Handdrehung rastet Kugel in gewünschte Option (Schrift frontal sichtbar) | Schnelles 3D-Grafikmenü zur Optionsauswahl mit einer Kugel im Raum | Analog 2D, nur häufig im Raum, mit >3DOF-Gerät schwer zu bedienen | Stift, dessen Segmente Menüoptionen darstellen, Auswahl durch Schnittpkt. | Auswahl durch Berührung von Daumen und Fingerspitzen (mit Datenhandschuh) | Menüpunkte werden kreisförmig angeordnet, Rotation, gewähltes Objekt im Fokus | Klassisches, meist im Raum schwebendes Menü, Auswahl mit Zeigergeräten |
| Bild |  |  |  |  |  |  |  |  |
| Referenz | [ISAAC@] | [Boule@] | [Grosj01] | z.B. [Wloka95] | [Wesch03] | [Bowma01a] | [Liang94] u.a. | z.B. [Bowma01a] |
| für immersive Systeme | ja | ja | ja | ja | ja | ja | ja | ja |
| für nicht-immersive Syst. | nein | ja | nein | ja | nein | nein | ja | ja |
| Eingabegerät | 3-6 DOF | 6 DOF (Bat) | 6 DOF | 2-6 DOF, Sprache | 6 DOF | Pinch Gloves | 2-6 DOF (Bat) | 2-6 DOF, Sprache |
| Dimensionalität der Interaktionsaufgabe | 1 D | 2D | 3D | 2-3D | 3D | 8D | 3D | 2-3 D |
| Geometrische Ausprägung / Widget | Virtuelle Hand, Selektorgeometrie, Objekte | Kugel | Kubus mit einzelnen Halbwürfeln für Menüeinträge | Menüeinträge und evtl. Rahmengeometrie | Menüeinträge als Zylinderscheiben + temporärer Text | Einfache Textgeometrien + virtuelle Hände | Fokus-, Zwischen- und statische Geometrie | Menüeinträge und evtl. Rahmengeometrie |
| Darstellungsaufwand | gering | gering | mittel | gering | gering | gering | mittel | gering |
| Maximale Anzahl Items | ca. 5-8 | ca. 8 | 27 | ca. 7 | ca. 7 | ca. 16, evtl. mehr | ca. 16 | ca. 7-10 |
| Menühierarchien | nein | nein | nein | nein | nein | ja, 1 Stufe | nein | nein, erweiterbar |
| 3D-Objekte Icons Text | ja ja evtl. | evtl. ja ja | evtl. ja nein | evtl. evtl. ja | nein evtl. ja | nein evtl. ja | ja evtl. nein | evtl. evtl. ja |
| Erscheinung, Platzbedarf | 2½ D, gering | 3D, gering | 3D, größer | 2D-3D, gering | 3D, gering | eher 2D, gering | 3D, mittel | 2D-3D, gering |
| Äquivalenz in 2D | nein | nein | nein | ja | nein | nein | nein | ja |
| Kommentare / Besonderheiten | Ähneln Ringmenü, Variante <i>Sundial</i> | Nur für wenige Optionen und schnelle Interaktion geeignet | Selektionskugel vom Volumen begrenzt, sehr schnelle Auswahl | Typisches Optionsmenü in VR-Anwendungen | Enge Integration zw. Objekt- und Operationswahl + Ausführung | Sehr effektiv für Datenhandschuhe, sonst ungeeignet | In [Green96] auch als <i>Daisy Menu</i> . In [Dachs97] parallel zum Bildschirm. | Klassisches VR-Menü, ähnelt Popup, häufig jedoch komplexer |

Tabelle 18: Übersicht der Widgets zur Menüselektion

| Widgets zur Menüelektron (Forts.) | Einzelmenüs (Forts.) | | | | Menühierarchien | | | |
|---|---|---|---|--|---|---|---|---|
| | Drop-Down-Menü | Rondell | Chooser Widget | 3D-Palette | Hands-off Menu | Hierarchical Pop-Up Menus | Tool Rack | Tortenmenü (3D Pie Menu) |
| Kurzbeschreibung | Funktion wie in 2D, mit >3DOF-Gerät schwer zu bedienen | Rondellrotation, Menüeinträge auf Seitenflächen (Kategorien) | Objektpaletten mit Scrollfunktionalität, wie Dateimenü | Menü typischerweise an Tablett gekoppelt, mit Stift-Auswahl | 2D-Overlay-Menü. Ursprünglich mit Spracheingabe | Nach Aktivierung „klassische“ Bedienung, dem Nutzer immer zugewandt | Aktivierung mit Stylus, Auswahl von Optionen, Untermenüs möglich | Nach Aktivierung wird Szene aus- & Menü eingeblendet. Sub-Menüs |
| Bild |  |  |  |  |  |  |  |  |
| Referenz | [Serra95] u.a. | [Preim97] u.a. | [Abe195] | [Billi97] | [Darke94] | z.B. [VWT@] | [Serra95] | [Deeri95] |
| für immersive Systeme | ja | ja | ja | ja | ja | ja | ja | ja |
| für nicht-immersive Syst. | ja | ja | ja | ja | ja | ja | ja | ja |
| Eingabegerät | 2 - 6 DOF | 2 DOF | 2-16 DOF (Gloves) | 2-3 DOF + Tablett | Sprache + 2 DOF | 2 - 6 DOF | 6 DOF | 2 - 6 DOF |
| Dimensionalität der Interaktionsaufgabe | 2 - 3 D | 2D | 3D+ | 2D | 1D (Sprache) o. 2D (Position) | 2D | 3D | 2D |
| Geometrische Ausprägung / Widget | Überschrift, Menüeinträge, evtl. Rahmengenom. & Scrollbar | Rotationskörper mit 4-8 Seiten und Texteinträgen | Palettengeometrie, Scroll-Widgets + selektierbare Objekte | Palettengeometrie, optional Buttons + selektierbare Objekte | Nur Text | Menüeinträge & evtl. Rahmengeometrie | Icons für Optionen / Menüeinträge | Menüeinträge und evtl. Objekte |
| Darstellungsaufwand | gering | mittel | höher | höher | gering | gering | gering | gering |
| Maximale Anzahl Items | ca. 7-10, mit Scrollbar auch mehr | etwa 56, auch mehr möglich | 7 gleichzeitig, sonst beliebig | > 7-12 nicht sinnvoll | nahezu beliebig | nahezu beliebig | viele | nahezu beliebig |
| Menühierarchien | nein | 1 Stufe | nein | nein | ja | ja | ja | ja |
| 3D-Objekte Icons Text | nein evtl. ja | nein evtl. ja | ja ja nein | ja evtl. nein | nein evtl. ja | nein evtl. ja | evtl. ja ja | evtl. ja ja |
| Erscheinung, Platzbedarf | 2D - 3D, gering | 3D, höher | 3D, mittel | 3D, mittel | gering | 2D, mittel | 3D, gering-mittel | 2½ D, hoch |
| Äquivalenz in 2D | ja | nein | nein | ja, 2D-Paletten | ja, Menüsystem | ja, Menüsystem | nein | ja, 2D-Tortenmen. |
| Kommentare / Besonderheiten | Ähneln Listen- und Popupenü. Auch hier Problem Textlesbarkeit | Variante: Stelen auf einer Drehbühne [Kolbe98] | Älteres System, jedoch mit 3D-Paletten vergleichbar | Zahlreiche Varianten, auch 2-Hand-Bedienung und Sprachsteuerung | Ist durch Overlay dem Nutzer immer zugewandt | Je nach Realisierung können Objektverdeckungen auftreten | Wie hierarchische Toolpaletten, nötiger Rücksprung wenig elegant | Große Hierarchien, Kombination von 2D-Icons, Text und 3D-Objekten |

3.5 Spezifikation von 3D-Widgets: auf dem Weg zu Standards

Das dritte Kapitel soll mit einem Vorschlag zur Spezifikation von 3D-Widgets und einem sich anschließenden Ausblick abgerundet werden. Nach einer Einführung in die Thematik von 3D-Benutzungsoberflächen allgemein wurde im Unterkapitel 3.1.6 bereits anhand von Zitaten deutlich gemacht, wie wichtig eine Systematisierung für 3D-Interaktionstechniken und Widgets ist. Im Anschluß an die Vorstellung von Gestaltungsrichtlinien im Unterkapitel 3.2 und die Einführung von Action Spaces unter 3.3 wurde deshalb im Unterkapitel 3.4 eine Systematisierung von 3D-Widgets präsentiert. Aus der umfangreichen Recherche konnten über 70 verschiedene Widget-Typen identifiziert und klassifiziert werden, wobei wichtige Vertreter tabellarisch aufgeführt wurden.

Hierbei handelt es sich jedoch nur um einen ersten Schritt in Richtung Standardisierung für 3D-Benutzungsoberflächen. Am Beispiel des Ringmenü-Widgets (s. Abschnitt 3.4.8.1) soll dies deutlich gemacht werden. So zeigt die Abbildung 12, wie unterschiedlich noch Erscheinungsbild und teilweise auch die Funktionalität eines Ringmenüs in verschiedenen Anwendungen sein können.

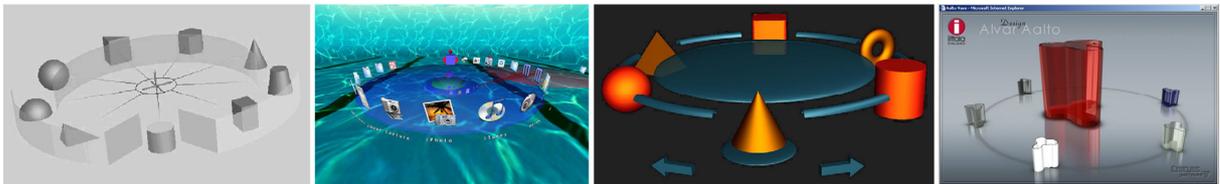
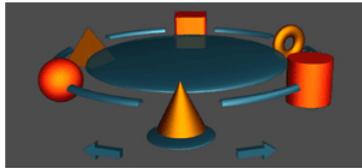


Abbildung 12: Verschiedene Ausprägungen eines Ringmenüs. Von links nach rechts: bei JDCAD [Liang94], 3DOSX-Filebrowser [3DOSX@], CONTIGRA [CONTIGRA@] und Cybelius-Showcase [Cybelius@]

Das Ziel sollte demzufolge eine exakte Spezifikation der Schnittstelle eines Ringmenüs sein, in der Parameter und Funktionalität unter Berücksichtigung verschiedener Freiheitsgrade verbindlich festgelegt sind. Für diesen Zweck wurde die CONTIGRA-Komponentenarchitektur (s. Kap. 4) entworfen, mit deren XML-Beschreibungssprache *CoComponent* (s. 5.3) sich Widgetschnittstellen verbindlich festlegen lassen. Neben diversen Metainformationen zu Version, Urheberschaft, Funktionsbeschreibung u.a. lassen sich vor allem die Parameter einer Komponente beschreiben. Dazu zählt die Kategorie eines Parameters (z.B. ob es sich um einen Geometrie- oder Verhaltensparameter handelt), sein Typ, seine Beschreibung, verschiedene Änderbarkeitsmodi und schließlich Informationen zur Unterstützung interdisziplinärer Autorenwerkzeuge. Unter 5.3.3 wird das CONTIGRA-Parameterkonzept detailliert vorgestellt. An dieser Stelle soll nur an einem Beispiel gezeigt werden, wie eine solche Ringmenü-Schnittstelle sich nach außen präsentiert. Dazu ist in Abbildung 13 die Webpräsentation für dieses Widgets zu sehen. Mit Hilfe einer solchen formellen Spezifikation eines jeden Widgets und der Kapselung in Form einer CONTIGRA-Komponente lassen sich sehr leicht für verschiedenste Projekte optisch angepaßte und trotzdem konsistente Varianten eines Ringmenüs erzeugen. Damit wird auch der unter 3.2.8 vorgestellten Gestaltungsrichtlinie „Ästhetische Qualität, Interface-Identität und Benutzungsfreude“ Rechnung getragen.

Ring Menu Component

[General Parameters](#) [Geometry Parameters](#) [Appearance Parameters](#) [Behavior Parameters](#) [Developer](#)



3D-Preview

A ringmenu component is composed of items arranged on a rotatable ring. In addition to the ring there is a fixed geometry, which doesn't rotate. In front of the menu a selection geometry highlights the current selection. When the mouse is moved over the geometry parts rotate left and rotate right, the ring is rotated accordingly.

Usage:
complement CoMenuComponent

Version: 1.0 date: 2001-10-11
Developer: [Michael Hinz](#)
Licence model: none

[Download Ring Menu Component](#)

| General Parameters | dataType | configurable | receivesEvents | generatesEvents | 👁 | 🔗 | 🎵 |
|---|-----------------|--------------|----------------|-----------------|---|---|---|
| ItemList List of menu entries | CoAnyURList | ✗ | | | ✗ | ✗ | |
| SelectedItem Index of currently selected item (zero based) | CoInteger | | ✗ | ✗ | ✗ | ✗ | |
| SelectedItemURL URL of currently selected item | CoAnyURI | | | ✗ | | ✗ | |
| Geometry Parameters | dataType | configurable | receivesEvents | generatesEvents | 👁 | 🔗 | 🎵 |
| FixedGeometry Fixed geometry does not rotate with the ring. It should be used as a frame. | CoGeometryGroup | ✗ | | | ✗ | | |
| InterItemGeometry Geometry between the items | CoAnyURI | ✗ | | | ✗ | | |
| RotateLeftGeometry Ring rotates left when mouse is moved over this geometry. | CoGeometryGroup | ✗ | | | ✗ | | |
| RotateRightGeometry Ring rotates right when mouse is moved over this geometry. | CoGeometryGroup | ✗ | | | ✗ | | |
| SelectionGeometry Selection geometry does not rotate with the ring. It should be used to highlight the selected item. | CoGeometryGroup | ✗ | | | ✗ | | |
| Appearance Parameters | dataType | configurable | receivesEvents | generatesEvents | 👁 | 🔗 | 🎵 |
| ItemRatio Ratio between items and in between geometry | CoFloat | ✗ | | | ✗ | | |
| RingRadius Radius of the ring | CoFloat | ✗ | | | ✗ | | |
| Behavior Parameters | dataType | configurable | receivesEvents | generatesEvents | 👁 | 🔗 | 🎵 |
| RotationSpeed Rotation speed of the ring (must be > 0) | CoFloat | ✗ | | | ✗ | ✗ | |

Michael Hinz
Dresden University of Technology
<http://www.contigra.com>

TU Dresden, Lehrstuhl Multimediotechnik
Dresden
Germany

Abbildung 13: Webdarstellung der CONTIGRA-Ringmenü-Spezifikation

Für eine Vielzahl der unter 3.4 klassifizierten Widgets wurde eine Beschreibung mithilfe der CONTIGRA-Beschreibungssprachen vorgenommen. Es fehlt jedoch noch ein einheitliches und standardisiertes „look and feel“ bei VE, was unter anderem an der Vielzahl wissenschaftlicher und proprietärer Arbeiten und Studien liegt, die aufgrund ihrer unterschiedlichen Anwendungsdomänen und maßgeschneiderten Lösungen bisher nur wenig Verallgemeinerung, weite Verbreitung und Evaluation zugelassen haben [Hand97]. Deshalb wurde im Rahmen dieser Arbeit zudem eine einheitliche geometrisch-optische Ausprägung angestrebt, um die Widgets leicht miteinander kombinieren zu können.

Es ist naheliegend, daß bisher nicht alle Widgets spezifiziert und vor allem implementiert werden konnten, da neben ihrer Beschreibung und geometrischen Modellierung auch die oft komplexe Realisierung der Funktionalität zu bewältigen ist. Das käme einer Re-Implementierung beinahe aller aus der Literatur bekannten Widgets gleich. Auf den

CONTIGRA-Projektwebseiten [CONTIGRA@] ist eine interaktiv navigierbare, der unter 3.4 vorgestellten Klassifikation folgende 3D-Widget-Hierarchie zu finden. Für alle bereits spezifizierten und implementierten Widgets sind Unterseiten im Stile der in Abbildung 13 für das Ringmenü gezeigten zu finden. Auf eine Auflistung innerhalb dieser Arbeit wurde aus Platzgründen verzichtet, zudem bietet die Online-Variante auch den aktuellen Stand der Realisierung und die Möglichkeit des interaktiven Ausprobierens einzelner Widgets. Die Webdarstellung der einzelnen Widgets könnte auch leicht zu einem *Online-Widget-Repository* erweitert werden, innerhalb dessen 3D-Widgets nicht nur präsentiert werden, sondern das auch Suchfunktionen, Download-Funktionalität, im Falle kommerzieller Anbieter Abrechnungsmöglichkeiten und andere Funktionen bereitstellt.

Mit der systematischen Analyse von Widgets und der Bereitstellung eines ersten Grundfundus an einheitlich spezifizierten und realisierten, frei miteinander kombinierbaren 3D-Widgets konnte eine Basis für eine künftige Weiterentwicklung und Standardisierung im Sektor der 3D-Benutzungsschnittstellen gelegt werden. Nach diesem theoretisch-konzeptionellen Teil wird in den folgenden Kapiteln dafür die komponentenbasierte Architektur bzw. technische Realisierung detailliert vorgestellt, die jenseits der Spezifikation von Widgets ihre Wiederverwendbarkeit in verschiedenen 3D-Applikationen erst möglich macht.