

Entwicklung von 3D-Widgets - Überblicksvortrag

Ulrich Leiner¹, Bernhard Preim², Stephan Ressel²

¹Zusammenfassung

Der Entwurf von 3D-Interaktionselementen (kurz 3D-Widgets) für Benutzungsschnittstellen wird immer wichtiger, um komplexe 3D-Anwendungen, wie CAD-Systeme, zu bedienen. Insbesondere kann das Potential der direkten Manipulation bei 3D-Anwendungen nur genutzt werden, wenn die modellbildenden Objekte Ein- und Ausgabefunktionalität besitzen. 3D-Widgets werden zunehmend auch für Aufgaben, die traditionell mit 2D-Widgets gelöst werden angewendet. Da 3D-Widgets zudem attraktiv sind, ist die Beschäftigung mit diesem Thema reizvoll. Dieser Beitrag bildet eine Übersicht über die Entwicklung von 3D-Widgets – einschließlich eigener Ergebnisse – und über offene Fragestellungen.

Schlüsselwörter: 3D-Interaktion, Direkte Manipulation von 3D-Daten

1 Einleitung

Anwendungen, in denen 3D-Daten manipuliert werden, haben sich in den letzten Jahren rasant verbreitet. Beispiele dafür sind Anwendungen in den Bereichen CAD, Scientific Visualization, Medizin und Unterhaltung. Diese Anwendungen werden zumeist mit herkömmlichen Interaktionselementen (Widgets) bedient, die für die Manipulation von 2D-Daten entwickelt worden sind. Mit diesen Widgets können 3D-Daten aber nur indirekt bearbeitet werden, oft über umständliche Angaben in langen Parameterlisten. Die Trennung zwischen der Darstellung der 3D-Daten und ihrer Manipulation in einem anderen Bildschirmbereich ist ein ernsthaftes Hindernis. Einen Ausweg versprechen dabei 3D-Widgets, mit denen 3D-Daten direkt bearbeitet werden können und die zu einer schnellen visuellen Rückkopplung führen. Der Begriff 3D-Widget ist von Conner *et al.* [1992] eingeführt worden und wird in der Literatur einheitlich verwendet. Ein 3D-Widget ist demnach eine Kombination einer 3D-Geometrie mit dem dazugehörigem Verhalten zur Steuerung oder Anzeige von Informationen.

Dreidimensionale Bedienoberflächen werden neuerdings auch für Büroanwendungen oder Informationszugang entwickelt. Dabei besteht das Ziel darin, eine Auswahl aus sehr großen Datenbeständen zu erleichtern.

Obwohl das Potential von 3D-Widgets zumindest für interaktive 3D-Graphikanwendungen unbestritten ist, sind erfolgreiche Umsetzungen schwierig. Hierfür sind mehrere Ursachen identifizierbar, aus denen sich Forschungs- und Entwicklungsaufgaben ableiten:

¹ SIEMENS AG, ZT

Otto-Hahn-Ring 6, D-81739 München
e-mail: ulrich.leiner@zfe.siemens.de

² Institut für Simulation und Graphik, Otto-von-Guericke-Universität Magdeburg
Universitätsplatz 2, D-39106 Magdeburg
e-mail: {bernhard, sressel}@isg.cs.uni-magdeburg.de

- Die kognitiven Fähigkeiten des Menschen im Umgang mit 3D-Widgets sind relativ unerforscht (Isaacs *et al.* [1995]). Wie weit Erfahrungen von realen auf virtuelle Räume übertragen werden können, ist noch unklar.
- Da die Aufgaben, für die 3D-Widgets geeignet sind, nicht hinreichend klassifiziert sind und ständig anwachsen, ist es kaum zu Standardisierungen gekommen. Dies macht die Entwicklung von 3D-Widgets aufwendig und führt zu Portabilitätsproblemen. Es führt auch dazu, daß der Lernaufwand für den Benutzer durch das Fehlen wiedererkennbarer Elemente relativ hoch ist.
- Die Anwendungen, für die 3D-Interfaces prädestiniert sind, sind so rechenintensiv, daß es inakzeptabel ist, wenn die Bedienelemente Rechenleistung größeren Ausmaßes erfordern. Dies schränkt die Entwurfsmöglichkeiten ein und führt dazu, daß sie bzgl. ihres Darstellungsaufwands minimiert werden.
- Bei der Anwendung von 3D-Widgets sind ferner folgende Probleme zu lösen: Sie müssen leicht als Interaktionselemente erkennbar sein und sich deutlich von nichtbedienbaren Objekten einer 3D-Szene unterscheiden. Ihre Bedienung setzt voraus, daß sie sichtbar sind und hinreichend groß dargestellt werden. Diese Anforderungen führen dazu, daß bei der Entwicklung von 3D-Widgets verschiedene Koordinatensysteme, Kamera- und Lichteinstellungen berücksichtigt werden müssen.

Diese Aufzählung zeigt, daß der Einsatz von 3D-Widgets durch technologische und konzeptuelle Schwierigkeiten beeinträchtigt ist. In diesem Beitrag wird davon ausgegangen, daß konzeptuelle Schwierigkeiten (wofür ist welche Interaktion geeignet?) das Hauptproblem sind und dementsprechend liegt darauf der Schwerpunkt der Betrachtung. Behandelt werden typische PC- und Workstation-Anwendungen, die keine zusätzliche Interaktionshardware erfordern, wie dies bei immersiven VR-Systemen der Fall ist.

2 Anwendungsfelder

Interaktionselemente implizieren Applikationen, mit denen sie kommunizieren. Mehrere typische Anwendungsfelder sind dabei erkennbar und kategorisierbar:

- Anwendungen, die 3D-Objekte erstellen oder konfigurieren, wie CA(D)-Systeme im Maschinenbau, in der Architektur, im Anlagenbau, Anwendungen in der Medizin und bei der Visualisierung wissenschaftlicher Berechnungen. Mit dem Einsatz von 3D-Widgets wird die Grenze aufgehoben, die durch eine Trennung von 3D-Darstellung und Bearbeitungsoptionen in verschiedene Bildschirmbereiche entstanden ist. Mit 3D-Widgets kann außerdem die zu Grunde liegende Mathematik (z.B. Kontrollpunkte von Splineflächen, Transformationsmatrizen) vor dem Benutzer verborgen werden.
- Anwendungen, denen kein räumliches Problem zugrunde liegt: Dieser jüngere Typ von Anwendungen gewinnt an Bedeutung, da leistungsfähige Erstellungswerkzeuge und gestiegene Grafikleistung Zugang und Einsatz wesentlich erleichtern. Hierzu gehören Büroanwendungen wie Dokumentenablage, Telekooperation, aber auch Daten- und Informationsaufbereitung.

Für solche Anwendungen erscheint es zunächst nicht plausibel, 3D-Metaphern und 3D-Widgets zu entwickeln, da auch mit 2D-Widgets direkt-manipulatives Arbeiten möglich ist (z.B. für die Navigation in einem Dateisystem). Dennoch zeigt sich, daß die zusätzliche Dimension Vorteile in der Übersichtlichkeit mit sich bringt und das Navigieren in großen Datenbeständen erleichtern kann (Robertson *et al.* [1991]). Außerdem wird argumentiert, daß diese Art der Navigation der natürlichen Selektion, z. B. beim Einkaufen, nahekommt.

- Interaktionen, die nicht die Anwendung selbst betreffen, sondern „nur“ ihre Präsentation am Bildschirm oder auf anderen Ausgabegeräten. Dazu gehört Zoomen und Navigieren mittels einer virtuellen Kamera, Manipulation von Lichtquellen, Auflösung der Darstellung (*Level of Detail*), Textur- und Farbeinstellungen.

Für diese Interaktionen bringen 3D-Bedienelemente erhebliche Vorteile: Sie sind einfacher, direkter bedienbar und verständlicher als vergleichbare Nicht-3D-Elemente.

3 3D-Widgets zur Manipulationen von 3D-Daten

Ausgehend von den Anwendungsfeldern und typischen Interaktionsaufgaben beschreibt dieser Abschnitt den Entwurf von 3D-Widgets für die Manipulation von 3D-Daten.

Die Bedeutung der Entwicklung von 3D-Widgets zur Manipulation von 3D-Daten hat zur Durchführung eines CHI-Workshops mit dem Thema „Herausforderungen der 3D-Interaktion“ geführt (Herndon *et al.* [1994]). Auch die Durchführung eines SIGGRAPH-Tutorials „Praktischer Entwurf von 3D-Widgets“ (Isaacs *et al.* [1995]) zeigt das gestiegene Interesse an diesem Thema. Dabei wird vor allem eine Herangehensweise an den Entwurf, die Implementierung und die Evaluierung vermittelt.

Dieser Abschnitt stützt sich maßgeblich auf diese Quellen. Allerdings wird die Betrachtung hier dahingehend strukturiert, daß von den typischen Interaktionsaufgaben ausgegangen wird. Außerdem werden erstmals Anwendungen aus der Medizin in einen Überblick integriert. Dies erscheint nötig, da medizinische Anwendungen, wie virtuelle Operationen, eine 3D-Interaktion erfordern und in diesem Zusammenhang Konzepte entwickelt worden, die auch auf andere Bereiche übertragbar sind.

3D-Widgets ermöglichen die direkt-manipulative Handhabung von 3D-Daten und können so die Distanz zwischen Anwendung und Bedienung verringern. Naheliegenderes Ziel der Überlegungen ist, für typische Aufgaben wie Selektion und Transformation von Objekten, Kamerasteuerung und Lichtquellendefinition 3D-Widgets zu etablieren. Dieser Abschnitt ist diesen Aufgaben entsprechend gegliedert.

3.1 Selektion von 3D-Objekten

Die Selektion von Objekten im 3D-Raum ist schwierig, insbesondere dann, wenn in komplexen Modellen selektiert wird. Aus einer bestimmten Sicht ist nur ein Teil der Objekte sichtbar. Einzelne Objekte sind sogar derart von anderen Objekten umgeben, daß man die Kamera ins Innere des Modells richten muß, um sie zu sehen. Eine exakte und eindeutige Selektion ist nur dadurch möglich, daß auf symbolischer Ebene Objekte über ihre Namen oder Attribute ausgewählt werden, was allerdings die Kenntnis der Namen voraussetzt.

Die direkte Selektion über Zeigen kann jedoch wesentlich erleichtert werden, wenn Teile des 3D-Modells semi-transparent dargestellt werden, so daß man auch verdeckte Objekte erkennen kann. Für die Selektion ist es erforderlich, daß Semi-Transparenz nicht nur zur Visualisierung eingesetzt wird, sondern daß ein semi-transparentes Interaktionsobjekt zur Selektion durch die Szene bewegt werden kann.

Diese Interaktionsform ist systematisch von Zhai *et al.* [1996] untersucht worden. Die Untersuchung klärt, ob die Reihenfolge der Objekte vor und hinter einem semitransparenten Objekt korrekt wahrgenommen wird (qualitative Tiefenverhältnisse). Bei dem Experiment wird ein Fisch zufällig in einem Aquarium bewegt, und ein Benutzer muß eine 3D-Box so positionieren, daß sich der Fisch vollständig in dieser Box befindet (siehe **Fehler! Verweisquelle konnte nicht gefunden werden.**). Es wurde verglichen, wie diese Aufgabe mit einer 3D-Box mit bzw. ohne semi-transparente Flächen gelöst wurde.

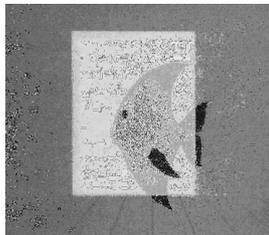


Abbildung 1: Eine semitransparente 3D-Box zur Lokalisation. Im Moment ist der Cursor noch nicht an der richtigen Stelle.

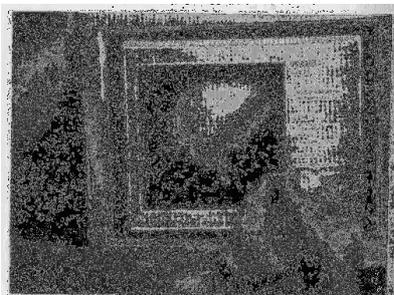


Abbildung 2: Erkundung eines Gehirns durch einen Neurochirurgen, der ein reales Objekt manipuliert. Die semi-transparente Darstellung unterstützt die räumliche Vorstellung

Mit statistischer Signifikanz konnte nachgewiesen werden, daß semitransparente Flächen die Genauigkeit und die Geschwindigkeit der Positionierung im 3D-Raum verbessern. (59% weniger Fehler, 48% schneller). Es konnte auch gezeigt werden, daß Semitransparenz für die 3D-Interaktion einen größeren Effekt hat als stereoskopisches Sehen.

Eine semitransparente Schnittfläche ist von Hinckley *et al.* [1994] eingesetzt worden, um neurochirurgische Eingriffe zu planen (Operationen am Gehirn). Der Benutzer hält ein Dummy-Objekt, das die Form eines Gehirns hat und eine Schnittfläche in den Händen. Die Bewegung der Schnittfläche relativ zum Gehirn wird am Bildschirm veranschaulicht, indem sich eine semi-transparente Schnittfläche durch das 3D-Modell des Gehirns bewegt (Abbildung 2). Dabei werden jeweils alle Bestandteile vor der Schnittfläche semi-transparent dargestellt. So können Schnitte geplant und am Modell durchgeführt werden.

Der Einsatz eines realen Objektes ist neu an dem System von Hinckley *et al.*, die interaktive Erkundung durch eine semi-transparente Scheibe ist dagegen in der Medizin verbreitet. Diese Technik ist auch in anderen Bereichen anwendbar, um ein komplexes 3D-Modell, z.B. einen Motor zu erkunden.

3.2 Transformation von 3D-Objekten

Die Transformation von 3D-Objekten setzt deren Selektion voraus. Für die Transformation existieren weitverbreitete Widgets. So stehen im 3D-Toolkit Open Inventor (siehe Strauß, Carey [1992]) vorgefertigte Klassen zur Verfügung, die sowohl Grafik, als auch Verhalten auf Mauseingabe vordefiniert anbieten.

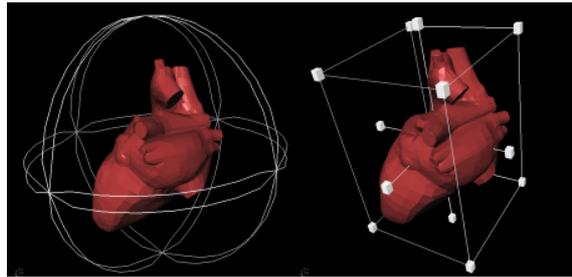


Abbildung 3: Interaktionselemente für die Transformation von 3D-Objekten
Trackball für die Rotation (oft auch als Virtuelle Kugel bezeichnet); 6 Handles
Handlebox für Skalierungen/Translationen; 12 Handles für 6 Freiheitsgrade (2*3 für Translation, 2*3 für Skalierung jeweils an den Eckpunkten des Quaders)

3D-Widgets haben mehr Freiheitsgrade als 2D-Widgets. Dies erlaubt dem Benutzer einerseits eine mächtige Interaktion, macht sie aber gerade für Anfänger schwerer beherrschbar. Daher ist es üblich, solche Widgets mit verschiedenen Handles (Griffen) zu versehen, wobei jedes Handle eine Teilfunktionalität der angebotenen Interaktionsfunktionalität auswählt und aktiviert (z.B. Skalierung in einer Richtung, Rotation um eine Achse). Diese Technik ist in Abbildung 3 zu sehen und macht die Freiheitsgrade für den Benutzer beherrschbar. Der vom Benutzer aktivierte Teil wird hervorgehoben und solange die Maus gedrückt bleibt, wird die entsprechende Manipulation durchgeführt.

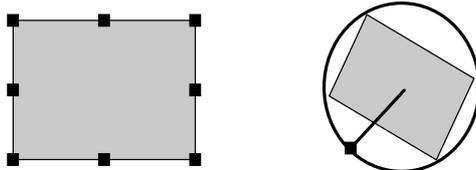


Abbildung 4: 2D-Widgets zur Translation, Rotation und Skalierung grafischer Objekte

Diese 3D-Widgets besitzen keine direkte Entsprechung in der Realwelt, sehr wohl aber gibt es vergleichbare 2D-Eingabemöglichkeiten (siehe Abbildung 4): Skalier- und Verschiebeoptionen für Fenster eines Multiwindow-Systems oder für grafische Einzelobjekte in einem Werkzeug für den Entwurf von Benutzungsschnittstellen. Allerdings erhöht die zusätzliche Dimension die Komplexität dieser Widgets wesentlich.

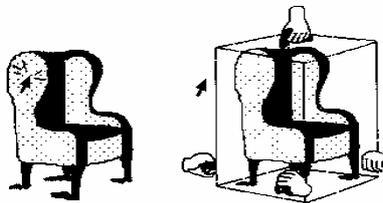


Abbildung 5: Bei der Selektion eines Objektes werden Griffe zur Skalierung angezeigt

Interaktionsmöglichkeiten stehen. Die Interaktion mit einem Handcursor kann mit einer Handlebox kombiniert werden, indem an der Handlebox Griffe erscheinen, die mit der Hand bewegt werden können (Abbildung 5). Hauptnachteil der Interaktion mit einem Handcursor ist der erheblich größere Aufwand für die Darstellung der Hände im Vergleich zu den Handleboxes. Hier muß wiederum ein Kompromiß zwischen der Anschaulichkeit der Griffe und dem Aufwand ihrer Darstellung.

Transformation von 3D-Objekten durch Interaktive Schatten

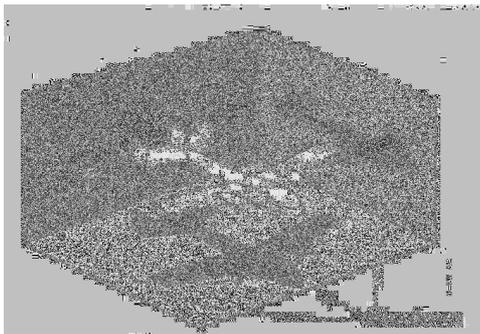


Abbildung 6 Interaktive Schatten zur Positionierung von Objekten im 3D-Raum, Quelle Herndon et al. [1992], S. 2

Die Möglichkeiten von 3D-Widgets zur Transformation sind systematisch von Houde (1991) untersucht worden. Dabei wurden die Handleboxes (Abbildung 3) mit einer Interaktion verglichen, bei der Hände als Widgets verwendet wurden. Diese Methode hat zwei Vorteile: Zum einen sind wir es gewohnt, Dinge mit der Hand zu bewegen; zum anderen kann die Hand als Cursor verschiedene Formen annehmen, die für die jeweiligen

Der Schattenwurf ist ein wichtiger Indikator für räumliche Verhältnisse, und die Projektion von Schatten auf orthogonale Wände erleichtert die Orientierung im 3D-Raum erheblich (siehe Wanger *et al.* [1992] für eine empirische Untersuchung). Davon ausgehend haben Herndon *et al.* [1992] „interaktive Schatten“ entwickelt, also Schatten, die selbst Interaktionsobjekte sind. Dabei werden die Schatten eines 3D-Modells auf die orthogonalen Wände projiziert. Die Transformation des 3D-Modells erfolgt dadurch, daß die Schatten verschoben werden. Die interaktiven Schatten können auch zur Rotation des 3D-Modells

verwendet werden. Dazu werden die Oberflächennormalen auf den „Schattenwänden“ visualisiert und können interaktiv verändert werden. Im Vergleich zu den zuvor beschriebenen Widgets ist festzustellen, daß kein zusätzliches Widget für die Interaktion angezeigt werden muß. Beide Ansätze erlauben die kontrollierte Benutzung der Freiheitsgrade. Bei den Manipulatoren geschieht dies durch die Handles, bei den interaktiven Schatten durch die Manipulation an einer der drei Wände für die X-, Y- und Z-Richtung.

Interaktive Schatten können auch zur Transformation einzelner Objekte eingesetzt werden. Dazu müssen die Schatten so dargestellt werden, daß einzelne Objekte erkennbar sind und in einer der orthogonalen Sichten selektiert werden können. Trotz der Projektion auf drei Wände ist die direkte Selektion von Objekten ein Problem bei großen Szenen, und so ist diese Form der Transformation nur begrenzt für einzelne Objekte anwendbar.

3.3 Parametrisierung von Lichtquellen und Kameras

Die Parametrisierung von Lichtquellen und Kameras hat eine entscheidende Gemeinsamkeit: In beiden Fällen erfolgt eine Positionierung relativ zu einer 3D-Szene. Bei gerichtetem Licht und einer Kamera wird zudem ein Richtungsvektor definiert.

In der Literatur existiert kein Ansatz, der diese Aufgaben in einer generischen Weise mit 3D-Widgets löst. Allerdings ist im Bereich der Medizin ein System entwickelt worden, bei dem die Kamerapositionierung auch auf die Lichtquellenpositionierung übertragen werden kann (Poston *et al.* [1996]). Diese Anwendung zielt auf die Behandlung des Gehirns; daher der Name BRAINBENCH.

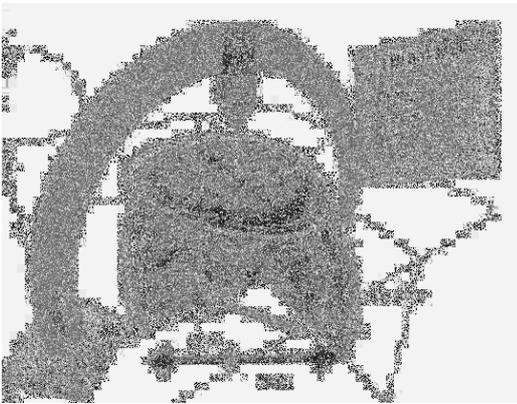


Abbildung 7 Die „Werkbank“ eines Neurochirurgen. Querschnitte des Gehirns in X-, Y- und Z-Richtung können selektiert werden. Auf dem Kreisbogen wird eine Kamera bewegt.

Dabei wird um ein 3D-Modell ein Kreisbogen konstruiert, auf dem eine Kamera entlangbewegt werden kann (Abbildung 7). Der Kreisbogen kann als Ganzes rotiert werden, in dem der Benutzer ihn mit einem Stift antippt und bewegt. Außerdem kann der Kreisbogen nach verschoben werden. Damit sind beliebige Kamerapositionen einstellbar. Bei einer Veränderung der Kameraposition werden die Objekte, die im Brennpunkt liegen, hervorgehoben. Dies ist sehr natürlich, da eine Kamera häufig genau auf gewisse Objekte gerichtet wird. In der BRAINBENCH wird diese Interaktion auch zur Simulation einer Bestrahlung genutzt. Dabei wird auf dem Kreisbogen die Bestrahlungsquelle positioniert und als Rückkopplung wird dargestellt, welche Gewebeteile beeinflusst werden.

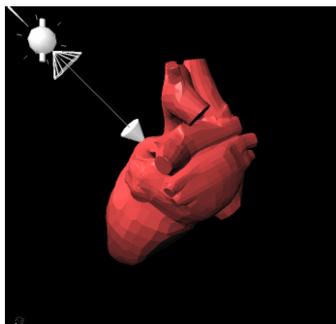


Abbildung 8 Virtuelle Lichtquelle zur Parametrisierung einer Spotlichtquelle

Bei der Parametrisierung von Spotlichtquellen ist zusätzlich zur Positionierung noch der Winkel des Spotlichtes einzustellen. Dafür ist die virtuelle Lichtquelle, wie sie von Open Inventor angeboten wird, sehr gut geeignet (siehe Abbildung 8).

Bei der virtuellen Lichtquelle kann der Benutzer die Kugel anfassen und bekommt dann Projektionsstrahlen zu sehen, auf denen er die Lichtquelle bewegen kann. Der Schirm kann aktiviert und in seinem Radius verändert werden, was den Winkel, in dem das Spotlicht strahlt, verändert. Außerdem kann durch Manipulation des Pfeils die Richtung verändert werden, in die die Lichtquelle strahlt.

Evaluierung

Die vorgestellten 3D-Widgets sind intuitiv anwendbar und lösen die beschriebenen Aufgaben recht gut. Ob sie für eine konkrete Anwendung günstig sind, hängt vor allem von der erforderlichen Genauigkeit ab. Geht es z.B. um den Entwurf einer Wohnungseinrichtung sind die Interaktionstechniken ausreichend, für konkrete Planungen im fortgeschrittenen Stadium einer Konstruktion dagegen müssen sie durch numerische Eingaben und Grid-Funktionen zur exakten Ausrichtung ergänzt werden. Die Erkennbarkeit der Widgets hängt stark von der konkreten Szene ab. In einer Szene mit vielen Linien (z.B. mit Wireframe-Objekten) sind der Trackball und die Handlebox unter Umständen schlecht als Interaktionselemente zu erkennen. Hier ist es nötig, einen Farbkontrast zu den Linien der manipulierten Szene herzustellen. Dies setzt voraus, daß nicht nur die Geometrie eines 3D-Widgets leicht modifizierbar ist, sondern auch die Farben anpaßbar sind. Die mit den Manipulatoren durchgeführten Veränderungen führen unmittelbar zum erneuten Rendern der Szene. Es ist offensichtlich, daß die Reaktionsgeschwindigkeit des Systems essentiell für die Nutzbarkeit der Interaktionstechnik ist.

Parametrisierung von Lichtquellen und Kameras

Dieser Abschnitt beschreibt Systeme, in denen Kameras und Lichtquellen nicht direkt spezifiziert werden, sondern in denen der Benutzer gewisse Effekte oder Bedingungen vorgibt, aus denen das System Lichtquellen und Kameras parametrisiert.

Poulin und Fournier [1992] stellen ein System zur Positionierung von Lichtquellen vor, bei dem die Beleuchtung dadurch gesteuert wird, daß gewünschte Glanzlichter gesetzt werden. Dies kommt dem Modellierer entgegen, da die Position des Glanzlichtes häufig der gewünschte Effekte ist. Der Entwurf eines Widgets wird dadurch – wie bei den interaktiven Schatten – umgangen. In gleicher Weise muß allerdings der Benutzer damit vertraut gemacht werden, daß etwas, was sonst nur ein visueller Effekt ist, manipulierbar ist.

Als vielversprechender Ansatz zur interaktiven Steuerung einer Kamera muß der Beitrag von Witkin und Gleicher (1992) erwähnt werden. Sie stellen verschiedene Möglichkeiten zur Kamerapositionierung bereit. Insbesondere erlauben sie es, Punkte in einer 3D-Szene zu spezifizieren und zu priorisieren und errechnen daraus eine passende Kameraposition.

3.4 3D-Widgets für geometrisches Modellieren

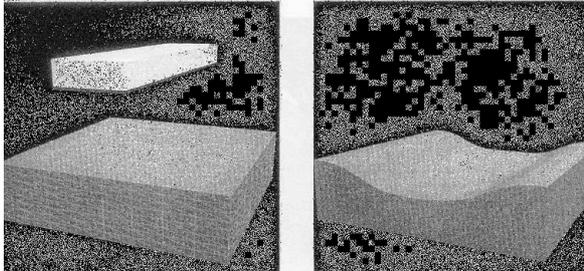


Abbildung 9: Indirekte Deformation durch Simulation einer Kraftwirkung

Flächen die Definition über Kontrollpunkte schwierig ist und zudem ein gewisses Verständnis der darunterliegenden Mathematik erfordert. Als intuitivere Methode schlagen sie eine indirekte Methode vor, bei der Widgets zur Deformation eingesetzt werden (Abbildung 9). Dies erspart zudem die verwirrende Anzeige der Kontrollpunkte.



Abbildung 10: Deformation eines 3D-Modells durch direkte Manipulation

Mausoperationen bedient werden. Aus den VR-Interaktionstechniken stammt dagegen eine andere Klasse von Interaktionstypen: Die pseudo-physikalische Manipulation. Dies sind Interaktionselemente, die bei räumlicher Nähe (Berührung) Veränderungen an 3D-Objekten bewirken, z.B. eine stilisierte Hand, mit der ein Objekt verformt wird, Feile, Töpferscheibe oder auch Pinsel als Handwerksmetaphern zur Objektbearbeitung (Deering [1996]). Sie werden entweder direkt durch Datenhandschuh oder indirekt über die Maus bedient.

Ihr Vorteil liegt in der starken Betonung des direkt-manipulativen Vorgehens. Das Objekt wird in Nachbildung eines mechanisch-physikalischen Vorgangs bearbeitet. Nachteilig ist die geringe Genauigkeit aus, die damit erreichbar ist.

Die Modellierung komplexer geometrischer Formen allein durch die Kombination geometrischer Primitive ist äußerst aufwendig. Daher werden häufig Freiformflächen zur Modellierung eingesetzt. Dabei kann der Modellierer die Kontrollpunkte einer Bezier- oder Spline-Approximation verändern und dadurch eine Form definieren. Hsu *et al.* [1992] zeigen, daß schon für einfache

Eine direkte Methode, mit 3D-Widgets geometrische Formen zu manipulieren, wird von Brookshire *et al.* [1992] vorgestellt. Dabei wird ein sogenanntes Rack-Widget genutzt (Abbildung 10), das drei Handles zur Deformation anbietet: Ein Handle zur Verjüngung eines Modells, ein Handle für eine Krümmung und ein Handle, mit dem ein Objekt in sich verdreht werden kann. Durch die Visualisierung des Schattenwurfs wird die Orientierung erleichtert.

Weitere 3D-Widgets

Den bisher vorgestellten Interaktionselementen ist gemeinsam, daß sie durch

4 3D-Widgets für nichträumliche Anwendungen

Ausgehend von den Anwendungsfeldern beschreibt dieser Abschnitt den Entwurf von 3D-Widgets für nichträumliche Anwendungen und für Navigationsaufgaben. Dieser neuere Typ von 3D-Widgets ist bisher nicht in einem Überblick zusammengefaßt worden.

Wie beim Entwurf von 2D-Widgets bilden Vorbilder der realen Welt den Ausgangspunkt für den Entwurf von 3D-Interaktionselementen. Für die 3D-Präsentation von Daten und Strukturen finden sich viele Bedienmetaphern wieder, die der Mensch aus seiner Anschauung her kennt: z.B. Länder- oder Stadtmetapher (siehe Volkmann [1994]). Dabei wird sowohl die äußere Form als auch die Reaktion realer Schalter, Regler oder anderer Gegenstände nachgebildet. Außerdem orientiert sich der Entwurf an verbreiteten 2D-Widgets, wie Slidern und Menüs. Diese müssen an spezielle 3D-Anforderungen angepaßt werden. Schließlich müssen, besonders für Navigationsaufgaben, neuartige Interaktionsmetaphern und -elemente entwickelt und evaluiert werden.

Als 3D-Widget betrachtet, bieten die Objekte dieser Metaphern meist ein einfaches Interaktionsverhalten an. Wie bei einem Button können Objekte angeklickt werden und eine Reaktion wie Verzweigung, Dienst- oder Dokumentzugang ausgelöst werden. Versucht man zur weitverbreiteten 2D-*Desktop-Metapher* (der Bildschirm als Schreibtisch mit Akten und Dokumenten) ein 3D-Analogon zu finden, so ist die *Raummetapher* das nächstliegende Pendant. Der Bildschirm wird zum Fenster in ein virtuelles Büro. Geht man davon aus, daß mindestens die gleichen Interaktionsmöglichkeiten bestehen sollen, wie in 2D-Oberflächen, müssen folgende Probleme gelöst werden:

- Umschalten von Buttons, die einen Zustand symbolisieren (Funktionalität von Radiobutton, Togglebutton)
- Selektion von Einträgen aus Listen (Funktionalität von Pull-Down-Menüs)
- Navigation und Editieren in hierarchischen Strukturen

Buttons

Als Entwurf eines 3D-Buttons erscheint es natürlich, einen Druck- oder Kippschalter nachzubilden. Während bei einer 2D-Oberfläche der Button immer auf den Benutzer hin ausgerichtet ist, kann er in 3D in einem ungünstigen Winkel oder sogar vom Benutzer weggewandt liegen. Aus Gründen der Erreichbarkeit ist es daher günstiger, wenn immer ein Teil des sensitiven Bereiches oberhalb der Grundfläche liegt, wie es bei einem Kippschalter der Fall ist (Ressel [1995]).

Beim Entwurf eines Buttons ist zu berücksichtigen, daß klar erkennbar sein muß, ob der Button gedrückt ist oder nicht. Während im Zweidimensionalen eine weitverbreitete Konvention den fiktiven Lichteinfall auf links oben festlegt und so Zustände eindeutig macht, ist dies in 3D-Interfaces nicht möglich. Um dennoch *an/aus* oder *gedrückt/nicht gedrückt* optisch unterscheiden zu können, ist eine Übersteigerung der vom Standard-Rendering gelieferten Ergebnisse nötig. Dies kann durch zusätzlichen Farbumschlag, größeren Schaltgang oder Nachschattierung geschehen. Nur so ergibt die Darstellung genügend Information für die räumliche Wahrnehmung des Benutzers.

Menüs

Zur Lösung der ersten Aufgabe wird (z.B. in Leiner [1996]) vorgeschlagen, die Einträge in Schränken, Regalen und Schubläden zu positionieren. Dies erlaubt die Unterbringung großer Datenbestände und ist z.B. zur Auswahl von Dateien angemessen. Allerdings ist diese Interaktionstechnik nicht intuitiv, wenn die Auswahl eines Interaktionsmodus gesteuert wird. Die Anschaulichkeit der Raummetapher führt zu einer hohen Diskrepanz zwischen den oft abstrakten Konzepten von Interaktionsmodi.

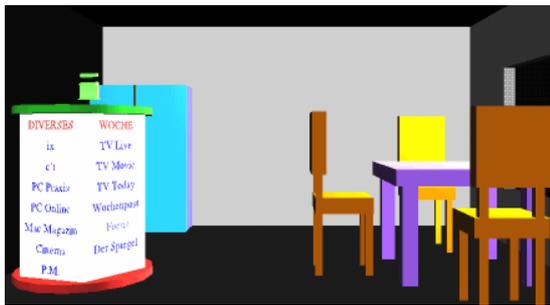


Abbildung 11: Integration eines Rondells als 3D Auswahlmöglichkeit in einer Büroszene

Eine Alternative dazu stellt eine rondellähnliche Auswahl (siehe Ressel [1995]) dar, bei der die Einträge an einem Ständer positioniert werden. Der Ständer kann in Rotation versetzt werden, so daß die einzelnen Seiten sichtbar und damit erreichbar werden. Abbildung 11 zeigt eine Anwendung dieses Interaktionselements in einer einfachen Büroszene. Günstig ist, daß das Rondell mit seiner Bodenfläche gut in einer Szene verankert werden kann.

Dieses Interaktionselement ist auch in einem interaktiven Programm zur Erklärung anatomischer Modelle eingesetzt worden (Preim *et al.* [1996]). Dabei werden die Beschriftungen der Objekte auf die Seiten des Rondells projiziert (Abbildung 12). Die Zuordnung der Beschriftungen zu den Seiten des Rondells erfolgt anhand der Organsysteme, die in der Anatomie eine sinnvolle Klassifikation darstellen. Da dieses Programm mit 3D-Modellen arbeitet, ist der Einsatz des Rondells in dieser Anwendung natürlich. Das Rondell dient zur Selektion von Objekten, die daraufhin beschriftet und grafisch hervorgehoben werden.

Erste Tests mit Benutzern zeigen, daß diese Interaktion als hilfreich empfunden wird. Außerdem wird sie – entgegen der Intention der Entwickler – nicht nur zur Selektion einzelner, nicht-beschrifteter Objekte benutzt, sondern zur systematischen Erkundung aller Teile eines Modells eingesetzt. Der große Platzbedarf des Rondells ist gerechtfertigt, wenn es häufig zur Interaktion benutzt wird und erspart in diesem Fall die häufige Aktivierung eines Pull-Down-Menüs. Problematisch ist, daß der 3D-Text einen erheblichen Berechnungsaufwand erfordert. Hier muß ein Kompromiß zwischen der Qualität der Textdarstellung und der Geschwindigkeit eingegangen werden.

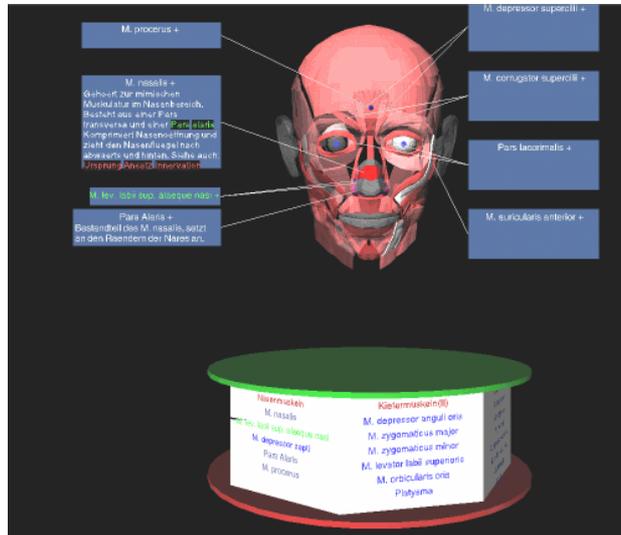


Abbildung 12: Anwendung des Rondells zur Selektion in einem Lehrsystem, ein selektierter Eintrag wird hervorgehoben und in der Illustration in der gleichen Farbe beschriftet

Sowohl ein 3D-Kippschalter als auch eine rondellähnliche Auswahlmöglichkeit sind wesentlich näher zu realen Vorbildern als die 2D-Widgets mit vergleichbarer Funktionalität. Ihr Vorteil liegt daher in einer natürlicheren Anwendung. Skalierbarkeit ist ein wichtiges Kriterium bei der Evaluierung des Rondells. Hier zeigt sich, daß ein Rondell – ähnlich wie ein Pull-Down-Menü – bei zu vielen Möglichkeiten unübersichtlich wird. Auch die permanente Verdeckung eines Teils der Einträge ist negativ zu bewerten.

Struktur-Browser

Zur Navigation in hierarchischen Strukturen wurden am XEROX PARC die dreidimensionalen *ConeTrees* (deutsch etwa: Kreiskegelbäume) (Robertson *et al.* [1991]) entwickelt. Bei ihnen hängt der Wurzelknoten des Baumes an der Decke eines Raumes. Die Kinder eines Knotens werden jeweils in einem Kreiskegel angeordnet, wobei die Spitze des Kegels mit dem Vaterknoten verbunden ist. Wird ein Knoten selektiert, dreht sich der Baum so, daß der Pfad der Wurzel bis zum Knoten auf den Betrachter ausgerichtet wird. Auf diese Weise werden einzelne Einträge aktiviert. Mit diesem System sind viele Knoten gleichzeitig darstellbar. Inzwischen entstanden mehrere Nachahmungen und Weiterentwicklungen dieser Idee, z.B.:

- *Lyber Trees* wurden bei der GMD entwickelt (Hemmje [1993]). Dabei wurde der Baum um 90 Grad gedreht, was den Vorteil hat, daß die Richtung der Rotationsachse des Baumes und die Schriftrichtung übereinstimmen.
- *Filesystem-Browser* von SGI (<http://www.sgi.com/Fun/free/>)
- *WebViz* von der Universität Minnesota (<http://www.geom.umn.edu:80/docs/research/webviz>) eine Darstellung beliebiger Netzstrukturen in einer speziellen 3D-Fisheye-Technik.

Diese 3D-Struktur-Browser sind nicht intuitiv; da sie kein Vorbild in der realen Welt besitzen. Ihr Vorteil liegt in der guten Skalierbarkeit, in der Anwendbarkeit auf große Datenbestände. Die semi-transparente Darstellung der vorderen Flächen ist dabei wesentlich.

5 Implementierung von 3D-Interaktionselementen

Die Entwicklung von 3D-Widgets setzt ein leistungsfähiges Toolkit voraus. Dieses Toolkit stellt idealerweise 3D-Widgets zur Verfügung und läßt sich leicht erweitern. Dies legt den Einsatz objektorientierter Techniken nahe, mit denen vorgefertigte Widgets spezialisiert (Vererbung) oder aggregiert werden können. Wichtig ist, daß die Form und das Verhalten (Interpretation der Benutzereingaben) unabhängig voneinander entworfen werden können.

Das bislang einzige kommerziell verfügbare System, daß diesen Anforderungen nahekommt, ist der Open Inventor von Silicon Graphics. Open Inventor ist eine Graphikbibliothek, die auf die interaktive Manipulation von 3D-Daten zugeschnitten ist. Es werden zusammengesetzte 3D-Widgets für lineare Transformationen und für die Lichtquellenparametrisierung zur Verfügung gestellt (Abbildung 3 und Abbildung 8). Die leichte Anpaßbarkeit der Form und Farbe der 3D-Widgets sowie eine Re-Interpretation ihres standardmäßigen Verhaltens sind dabei wichtige Entwurfsziele. Open Inventor bildet nicht nur die Grundlagen der eigenen Arbeiten, sondern war beispielsweise auch Basis für das Lyber-World-Project der GMD (Hemmje [1993]). Trotz der Vorteile, die Open Inventor für die Erstellung von 3D-Interfaces bietet, sind auch dort noch erhebliche Defizite im Vergleich zu komfortablen Ressourcen-Workshops aus dem 2D-Bereich festzustellen, wie den SX-Tools (Kühme, Schneider-Hufschmidt [1992]). Der mit Open Inventor arbeitende Entwickler muß programmieren, Klassen ableiten oder aggregieren. Die Parametrisierung von Form und Farbe der Manipulatoren ist sehr aufwendig.

Neben Open Inventor existiert ein Forschungsprototyp, mit dem 3D-Widgets erzeugt werden können – das UGA-System (Zeleznik *et al.* [1991] und Zeleznik *et al.* [1993]). Dieser Prototyp ist die Grundlage für die Systeme von Conner *et al.* [1992] und Snibbe *et al.* [1992]. Er weist viele Ähnlichkeiten zu Open Inventor auf. Die entscheidende Erweiterung liegt in der Integration von Beschreibungsmöglichkeiten von UIMS-Systemen.

6 Zusammenfassung und zukünftige Aufgaben

Dieser Beitrag macht deutlich, daß es eine Vielzahl von Anwendungen gibt, in denen 3D-Interaktionselemente wesentlich sind. Er zielt darauf, den Gestaltungsspielraum beim Entwurf von 3D-Widgets aufzuzeigen und anhand typischer Interaktionsaufgaben bisherige Entwürfe zu charakterisieren. 3D-Widgets haben mehr Freiheitsgrade als 2D-Widgets, was die Bedienung erschwert. Diese Freiheitsgrade beherrschbar zu machen, ist eine Herausforderung beim Entwurf von 3D-Schnittstellen.

Die Präzision von Eingaben ist ein wichtiges Kriterium für die Evaluation von 3D-Widgets für die Handhabung von 3D-Daten. Die direkt-manipulative Handhabung allein reicht für eine hohe Präzision nicht aus. Die Eignung von 3D-Widgets für nicht-räumliche Anwendungen hängt vor allem von der Skalierbarkeit auf die Interaktion in großen Datenbeständen ab. Daran muß auch gemessen werden, ob 3D-Widgets den in diesem Bereich dominierenden 2D-Widgets überlegen sind.

Mit Open Inventor steht ein portables und verbreitetes Werkzeug zur Verfügung, mit dem 3D-Widgets entworfen und implementiert werden können. Der Entwicklungsprozeß ist dadurch wesentlich erleichtert worden, erfordert aber immer noch erheblichen Programmieraufwand. Ausgehend von Untersuchungen zur Wirksamkeit von „depth cues“ in statischen Bildern (z.B. Wanger *et al.* [1992]) müssen empirische zum Umgang mit 3D-Widgets erfolgen.

Zudem ist es erforderlich, Toolkits zu entwickeln, in denen Interaktionselemente spezifiziert werden, so daß eine Entwicklung mit möglichst geringem Programmieraufwand möglich wird. Damit wird es möglich, interaktiv Bedienelemente in virtuellen Räumen anzuordnen.

Danksagung

Wir möchten uns bei unseren Kollegen Oliver Deussen, Ralf Helbing, Andreas Raab und Alf Ritter für fruchtbare Diskussionen und Kommentare zu initialen Versionen dieses Beitrags bedanken. Außerdem sind wir dankbar für die ausführlichen Kommentare und Literaturhinweise der Gutachter.

Literatur

- Conner, D. B., S. C. Snibbe, K. P. Herndon, D. C. Robbins, R. C. Zeleznik und A. van Dam (1992)
„Three-Dimensional Widgets“, *Proc. of Symposium on Interactive 3D Graphics*, März, S. 183-188
- Deering, M. (1996)
„The HoloSketch: VR Sketching System“, *Communications of the ACM*, Mai, S. 54-61
- Gleicher, M. und A. Witkin (1992)
„Through the Lens Camera Control“, *Proc. of SIGGRAPH'92*, Juli, S. 331-340
- Hemmje, M. (1993)
„Lyber-World – Eine 3D-basierte Benutzerschnittstelle für die computerunterstützte Informationssuche in Dokumentmengen“, *GMD-Spiegel 1/93*, S. 58-63
- Herndon, K. P., R. C. Zeleznik, D. C. Robbins, D. B. Conner, S. S. Snibbe und A. van Dam (1992)
„Interactive Shadows“, *Proc. of UIST'92*, November, S. 1-6
- Herndon K. P., A. van Dam und M. Gleicher (1994)
„The Challenge of 3D Interaction“, *SIGCHI-Bulletin October*, S. 36-43
- Hinckley, K., R. Pausch, J. C. Goble und N. F. Kassell (1994)
„Passive Real-World Interface Props for neurosurgical visualization“, *Proc. of CHI'94*, April, S. 452-458
- Houde, S. (1991)
„Design of an Interface for Easy 3D-Interaction“, *Proc. of CHI'91*, April, S. 135-142
- Hsu, W. M., J. F. Hughes und H. Kaufmann (1992)
„Direct manipulation of free-form deformations“, *Proc. of SIGGRAPH'92*, S. 177-184
- Isaacs, P., K. Mathews, M. Mine, R. Ormandy und K. Smit (1995)
„Practical 3D User Interface Design“, *Course Notes, 22nd International Conference on Computer Graphics and Interactive Techniques*, Los Angeles, 6-11. August
- Kühme, Th. und M. Schneider-Hufschmidt (1992)
„SX-Tools – An Open Design Environment for Adaptable Multimedia Interfaces“, *Proc. of Eurographics'92*, Blackwell, Cambridge, 7-11. September, UK, S. 93-105
- Leiner, U. (1996)

- An der Schnittstelle von Mensch und Maschine, Skript zur Vorlesung User Interface Design, Otto-von-Guericke-Universität, Sommersemester 1996
- Poston, T., W. L. Nowinski, L. Serra, B. C. Chua, H. Ng und K. Ridge (1996)
 „The Brain Bench; Virtual Stereotaxis for rapid Neurosurgery Planning and Training“, *Proc. of Visualization in Biomedical Computing*, Hamburg, September, S. 491-500
- Poulin, P. und A. Fournier (1992)
 „Lights from Highlights and Shadows“, *Proc. of Symposium on Interactive Three-Dimensional Graphics*, ACM SIGGRAPH, März, S. 31-38
- Preim, B., A. Ritter und T. Strothotte (1996)
 „Illustrating Anatomic Models – A Semi-Interactive Approach“, *Proc. of Visualization in Biomedical Computing*, Hamburg, September, S. 23-32
- Ressel, S. (1995)
Aufbau einer Klassenbibliothek von dreidimensionalen Interaktionselementen, Diplomarbeit, Otto-von-Guericke-Universität, Institut für Simulation und Graphik
- Robertson, G. G., J. D. Mackinlay und S. K. Card (1991)
 „Cone Trees: Animated 3D Visualization of Hierarchical Information“, *Proc. of CHI*, S. 189-202
- Snibbe, S. S., K. P. Herndon, D. C. Robbins, D. B. Conner und A. van Dam (1992)
 „Using deformations to explore 3D-widget design“, *Proc. of SIGGRAPH'92*, S.351-352
- Strauß, P. S. und R. Carey(1992)
 „An Object-Oriented 3D-Graphics Toolkit“, *Proc. of SIGGRAPH'92*, Juli, S. 341-349
- Volkman, H. (1994) (siehe auch: <http://www.is.in-berlin.de/~mfz/citymetaphor.html>)
 Impressionen zum Leitbild "Städte des Wissens als Stätten der Begegnung" mit ersten Berichten aus XENIA, der Wissenstadt am Wege zur Informationsgesellschaft. In: GROTE, v.C. U.A. (Hrsg) (1994) Kommunikationsnetze der Zukunft - Leitbilder und Praxis. Dokumentation einer Konferenz am 3.6.1994 im WZB
- Wanger, L., J. Ferwerda und D. Greenberg (1992)
 „Perceiving Spatial Relationships in Computer-Generated Images“, *IEEE Computer Graphics & Applications* S. 44-58
- Zelevnik, R. C., D. B. Conner, M. M. Wloka, D. G. Aliaga, N. T. Huang, P. M. Hubbard, B. Knep, H. Kaufmann, J. F. Hughes und A. van Dam (1991)
 „An Object-Oriented Framework for the Integration of Interactive and Animation Techniques“, *Proc. of SIGGRAPH'91*, Juli, S. 105-112
- Zelevnik, R. C., K. P. Herndon, D. C. Robbins, N. Huang, T. Meyer, N. Parker und J. F. Hughes (1993)
 „An interactive toolkit for constructing 3D interfaces“, *Proc. of SIGGRAPH'93*, Juli, S. 81-84
- Zhai, S., W. Buxton und P. Milgram (1996)
 „The Partial Occlusion Effect: Utilizing Semitransparency in 3D Human Computer Interaction“, *ACM Transactions on HCI*, Vol. 3, No. 3, S. 254-284