

Geräteübergreifende effiziente 3D-Modelliertechnik für 3D-Interaktionsräume, Touch-Interfaces und WIMP-Systeme basierend auf implizit bestimmten Modellierebenen oder -geraden

Thomas Jung

HTW Berlin,
Wilhelminenhofstraße 75A, D-12459 Berlin
eMail: t.jung@htw-berlin.de
URL: <http://www.f4.htw-berlin.de/~tj>

Zusammenfassung Heutige 3D-Modellierwerkzeuge basieren in der Regel auf 2D-Eingabegeräten, da diese weit verbreitet und einfach bedienbar sind. 3D-Interaktion gilt als schwierig, so dass die Verwendung von Constraints empfohlen wird. Im Folgenden wird eine 3D-Modelliertechnik beschrieben, die auf der impliziten Bestimmung von Modellierebenen oder -geraden basiert. Diese Modelliertechnik kann mit Mouse oder Multitouch-Interface genauso wie mit 3D-Eingabegeräten verwendet werden. Dadurch wird die Integration von 3D-Eingabegeräten in kommerzielle Modellierwerkzeuge ermöglicht. Ein Prototyp basierend auf einem 3D-Eingabegerät und einem Head-Mounted Display zeigt die Funktionsweise des beschriebenen Ansatzes.

1 Einleitung

3D-Modelliertechniken in kommerziellen Modellierwerkzeugen basieren in der Regel auf 2D-Eingabegeräten wie Mouse und Tastatur oder Grafiktablett, da WIMP-Interfaces an üblichen PC-Arbeitsplätzen verfügbar und Anwender in deren Bedienung geübt sind. 3D-Eingabegeräte sind nur selten verfügbar, so dass zurzeit zentrale Interaktionskonzepte in einem kommerziellen Werkzeug nicht auf 3D-Eingaben beruhen können.

Spezialisierte Geräte mit mehr als zwei Freiheitsgraden wie z. B. die Spacemouse der Firma 3DConnexion [1] werden zurzeit nur als Ergänzung, z. B. zur Kamerasteuerung eingesetzt.

Modellieroperationen erfolgen in der Regel in mehreren aufeinander folgenden Phasen (vgl. 3dsmax [2] oder Maya [3]). Zunächst werden zum Beispiel mit Hilfe des 2D-Eingabegeräts zu verändernde Elemente selektiert, anschließend wird die Modellieroperation gewählt, schließlich werden die Elemente interaktiv durch das Eingabegerät verändert, wobei oft ein einzelner Parameter separat eingestellt wird.

Grundsätzlich werden drei unterschiedliche Arten von *3D-Interaktionen* unterschieden: Systemsteuerung, Travel und Manipulation. Manipulationsaufgaben, können wiederum in die Teilaufgaben Selektion, Positionierung und Drehung zerlegt werden.

Selektionstechniken erfordern dabei eine Möglichkeit zur Identifikation von Objekten, zur Bestätigung der Selektion und zum Darstellen des Feedbacks für den Benutzer [4].

Während bei 2D-Interaktionen die Einzelaufgaben in der Regel nahtlos integriert werden (vgl. Direkte Manipulation [5] oder Drag-und-Drop-Operationen bei Verwendung der WIMP-Metapher), werden in 3D-Umgebungen die Teilaufgaben häufig nacheinander ausgeführt. Im Werkzeug Holostudio [6] von Microsoft für das angekündigte Head-Mounted-Display Hololens werden Modellieroperationen z. B. durch spezielle Fingergesten („Simple-to-use tools“) eingeleitet [7], bevor die eigentliche Veränderung des 3D-Modells stattfindet.

Manche Modelliersysteme vereinfachen diesen Prozess, indem die Modellieroperation direkt beim Verändern des Parameters implizit gewählt wird (z. B. bei Verwendung so genannter 3D-Widgets [8]). Eine bessere Integration der Teilaufgaben verspricht hier für 3D-Interaktionen eine Effizienzsteigerung. Dieses Prinzip lässt sich nicht nur bei Manipulationsaufgaben sondern auch bei Modelliertechniken anwenden.

Interaktionen im 3D-Raum, bei denen der Benutzer frei die 3D-Positionen steuern kann, gelten darüber hinaus als schwierig [9], so dass die Gebrauchstauglichkeit solcher Systeme eingeschränkt sein kann. In diesem Zusammenhang wird oft die Verwendung von Constraints empfohlen, die die Freiheit der Bewegung einschränken, um die Bedienung zu vereinfachen [10][11, Seite 180f].

Hersteller von 3D-Eingabegeräten propagieren deshalb vorrangig deren Einsatz für spezifische Modellieroperationen wie z. B. Sculpting, da 3D-Eingaben zumindest dort Vorteile gegenüber 2D-Eingaben haben sollen [12]. Als ein Beispiel sei hier die 3D-Modellierung nach der Töpferscheibenmetapher genannt [13].

Ein interessanter Ansatz ist auch die Verwendung von Multitouch-Interfaces für die 3D-Modellierung, da die Verwendung mehrerer Finger besondere Modellieroperationen wie z. B. Deformationen ermöglicht, die mit der Mouse allein nicht durchgeführt werden könnten. Multitouch-Interfaces stehen jedoch in der Regel nur auf mobilen Geräten zur Verfügung, die oft nicht die erforderliche Rechenleistung für kommerzielle 3D-Modellierwerkzeuge besitzen. Da Touch-Input in der Regel auch weniger genau ist als die Eingabe per Mouse, und Tastatureingaben auf mobilen Geräten nicht komfortabel verfügbar sind, setzen Anbieter kommerzieller Modellierwerkzeuge in der Regel nicht auf Touch-Interfaces.

2 Implizite Modellierebenen in Artist3D

Mit dem Modellierwerkzeug Artist3D konnte gezeigt werden, dass die Verwendung eines Multitouch-Interfaces Vorteile bei der 3D-Modellierung bietet [14][15]. In Artist3D werden Modellierebenen verwendet, die implizit durch den aktuellen Interaktionsmodus und den selektierten Eckpunkt festgelegt werden. Die implizite Bestimmung von Modellierebenen erfolgt z. B. bei der Veränderung eines Quaders in Artist3D wie folgt:

- a) Durch Selektieren eines Eckpunkts eines sichtbaren Quaders in perspektivischer Projektion legt der Benutzer implizit fest, dass Operationen zum Verändern von Quaderparametern stattfinden sollen (Veränderung der Höhe, Tiefe, Breite, Position oder Ausrichtung).
- b) Der selektierte Eckpunkt legt dann implizit fest, welche Parameter verändert werden sollen und welche Modellierebene dafür gewählt wird.
- c) Der linke obere Punkt ermöglicht dabei z. B. die Veränderung der Höhe und Tiefe des Quaders per Drag-and-drop, wobei die rechte seitliche Fläche des Quaders der Modellierebene entspricht.

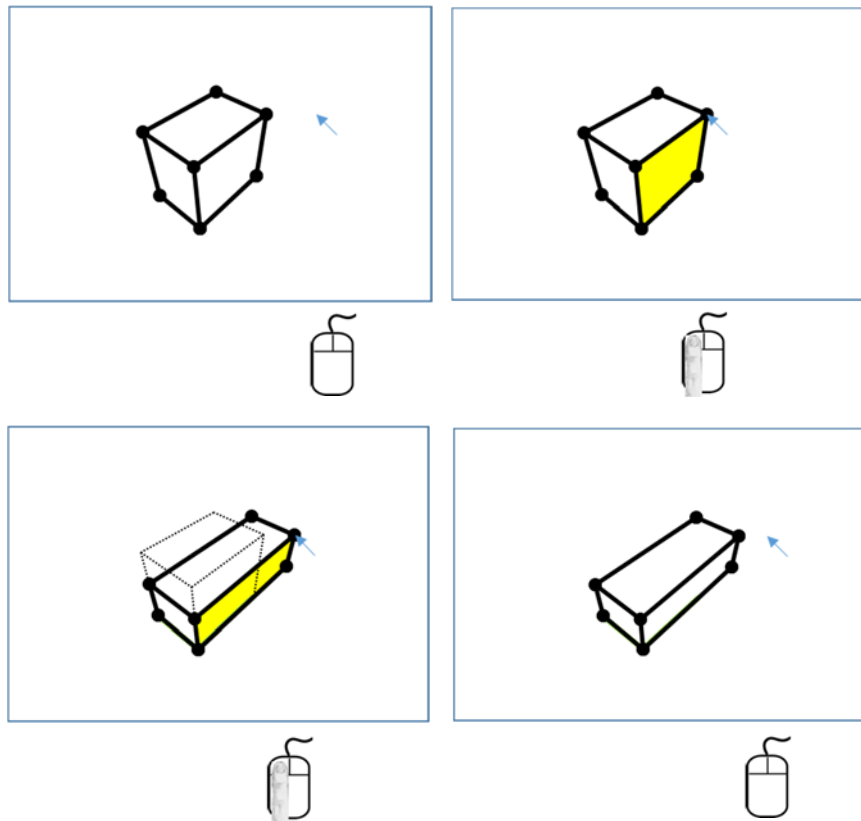


Abb. 1: *oben rechts:* Selektion des Punktes legt Operation und Modellierebene (gelb) fest, *unten links:* Verschieben bei gedrückter Taste, *unten rechts:* Beenden der Operation durch Loslassen der Taste

Die beschriebenen Schritte werden in den Abbildungen 1 für die Interaktion über ein WIMP-Interface und in Abbildung 2 für die Interaktion mittels eines Multitouch-Interfaces veranschaulicht.

Im Modellierwerkzeug Artist3D wird dieses Prinzip u. a. bei unterschiedliche Geometrietypen wie Rotationskörper, Prisma, Dach und verschiedene Modellieroperationen angewendet.

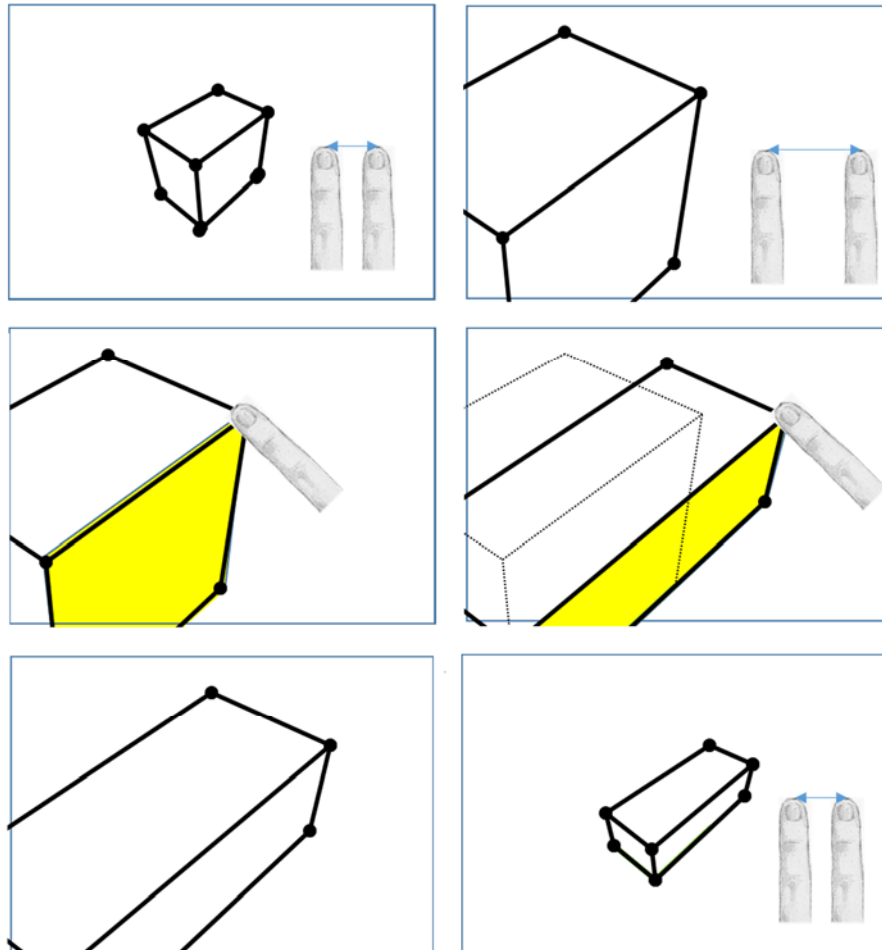


Abb. 2: *oben:* Optionales hineinzoomen, um geringere Genauigkeit auszugleichen, *Mitte links:* Selektion des Punktes legt Operation und Modellierebene (gelb) fest, *Mitte rechts:* Verschieben mit Fingerkontakt, *unten links:* Beenden der Operation durch Lösen des Fingers, *unten rechts:* Optionales herauszoomen

Die Veränderung der Modellparameter erfolgt per Touch-and-drag, so dass die Modellieroperationen in der Regel effizienter als in kommerziellen Modellierwerkzeugen durchgeführt werden können.

Die Kamerasteuerung erfolgt bei vorhandenem Multitouch-Interface über Fingergersten, bei WIMP-Interfaces über die rechte Mouse-Taste. Der Modelliermodus muss dazu jeweils nicht verlassen werden.

3 Übertragung auf 3D-Interaktionsräume

Die beschriebene Modelliertechnik kann auf Interaktionsräume, in denen Fingerpositionen berührungslos getrackt werden, abgebildet werden. Besonderheit dieser Interaktionsräume ist, dass z. B. im Gegensatz zur Mouse, die ja Tasten besitzt, kein expliziter Interaktionskanal für das Starten der Interaktion vorhanden ist.

Im Gegensatz z.B. zu Holostudio, wo der Manipulation eine Geste vorausgehen muss, wirkt die hier beschriebene Modellieroperation direkt auf das 3D-Modell. Dazu werden folgende algorithmische Schritte durchlaufen:

- 1.) Die Position des/eines Fingers wird getrackt und der 3D-Abstand zum nächsten Polygon eines Objektes oder zur Bodenfläche wird bestimmt.
- 2.) Sollte ein Mindestabstand unterschritten werden, wird das Polygon mit dem geringsten Abstand zur Fingerspitze oder die Bodenfläche selektiert.
- 3.) Das Modelliersystem muss anhand des selektierten Objekts eine Modellieroperation und eine passende Modellierebene oder –gerade bestimmen
- 4.) Der Benutzer kann dann unmittelbar seinen Finger innerhalb der Modellierebene oder –gerade im 3D-Raum bewegen.
- 5.) Solange der Abstand des Fingers einen maximalen Abstand zur Ebene bzw. Geraden nicht überschreitet, wird die Fingerposition senkrecht auf die 3D-Modellierebene oder –gerade projiziert. Der so bestimmte 3D-Punkt wird dem Modelliersystem übergeben, so dass dieses simultan die Modellieroperation steuern kann. (In der Regel sollte der vorgegebene Maximalabstand hier größer sein als der Mindestabstand unter Punkt 2.)
- 6.) Zum Beenden der Modellieroperation entfernt der Benutzer seinen Finger von der Modellierebene oder -geraden, so dass der Maximalabstand überschritten wird. (Die Operation endet am Projektionspunkt beim Überschreiten des Maximalabstands.)

Der Mindestabstand (Punkt 2) und der Maximalabstand (Punkt 5 und 6) sollten anhand der Genauigkeit und der Größe des Interaktionsbereichs festgelegt werden. Der Ablauf der Modellieroperation ist in Abbildungen 3 veranschaulicht. Ein Ausgabegerät, das mit dem Eingaberaum registriert ist (z. B. Microsoft HoloLens) ist dabei vorteilhaft aber nicht zwingend erforderlich.

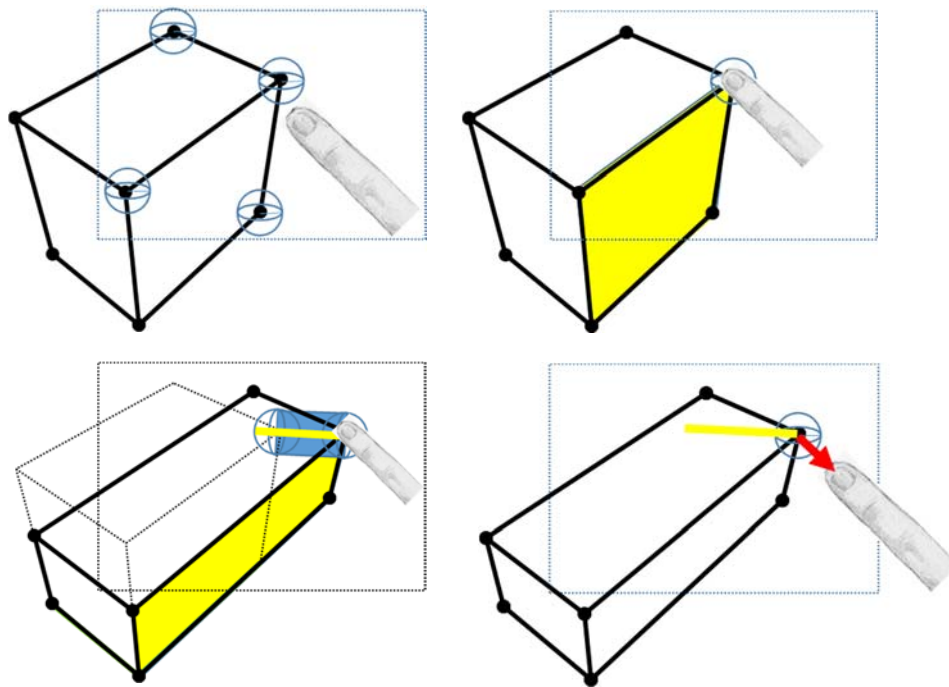


Abb. 3: *oben links:* Annäherung an Objekt, *oben rechts:* Selektion durch Annäherung an einen Punkt, dabei wird Operation und Modellierebene (gelb) festgelegt, *unten links:* Verschieben indem 3D-Abstand zur Modellierebene Schwellwert unterschreitet, *unten rechts:* Beenden der Operation durch Überschreiten des Schwellwerts

Schritt 2 entspricht bei der Verwendung einer Computermouse der Selektion mit Hilfe der Mouse-Taste (Abbildung 1, oben rechts) bzw. dem Berühren eines Touchscreens (Abbildung 2, Mitte links), Schritt 6 entspricht dem Loslassen dieser Taste (Abbildung 1, unten rechts) bzw. dem Lösen des Fingers vom Touchscreen (Abbildung 2, unten links).

In Abbildung 3 ist eine solche Modellieroperation nun anhand des gleichen Beispiels (Veränderung von Höhe und Tiefe eines Quaders) wie in den Abbildungen 1 und 2) dargestellt.

Mit der beschriebenen Technik können zukünftige 3D-Modellierwerkzeuge einerseits auf konventionellen WIMP-Plattformen in gewohnter Weise funktionieren,

andererseits können solche Modellierwerkzeuge direkt von 3D-Interaktionsräumen profitieren, wobei der Lernaufwand für die Umstellung beim Benutzer relativ gering sein wird

Voraussetzung dafür ist, dass die Modellierwerkzeuge ein Interface mit folgenden Funktionen realisieren:

```
bool getPlaneOrLine(in Vec3 pos, in float maxDistance, out PlaneOrLine  
planeOrLine);  
void drag(in Vec3 pos, inout PlaneOrLine planeOrLine);  
void drop(in Vec3 pos);
```

Der Datentyp „*PlaneOrLine*“ beschreibt entweder eine Modellierebene oder eine Modelliergerade sowie die Information darüber welcher der beiden Geometrietypen gespeichert ist. In der Funktion „**getPlaneOrLine**“ wird anhand der Fingerposition „pos“ im 3D-Raum erkannt, ob ein Objekt selektiert wurde. Falls dem so ist, wird die Modellieroperation und die Modellierebene oder -gerade bestimmt. Das Modellierwerkzeug muss dann in „**drag**“ die Veränderung der Szene realisieren.

Bei Verwendung eines WIMP- oder Multitouch-Interfaces wird die 3D-Position bei „**getPlaneOrLine**“ durch Raycasting vom 2D-Punkt in die 3D-Szene als Schnittpunkt mit einer Objektoberfläche oder der Bodenebene bestimmt. In „**drag**“ und „**drop**“ entspricht die 3D-Position der Projektion des 2D-Punkts auf die Modellierebene oder -gerade.

Bestehende kommerzielle Modellierwerkzeuge können um dieses Interface erweitert werden, sofern sie über eine geeignete Scripting-Schnittstelle verfügen.

4 Kamerasteuerung

Die Kamerasteuerung nimmt im 3D-Modellierprozess eine besondere Rolle ein. Mit ihr ist es unter anderem möglich, den zu bearbeitenden Bereich und die zu bearbeitende Teiloberfläche in den Fokus (bei 2D-Werkzeugen als Ansicht auf den Bildschirm) zu bekommen.

Da der Fokus während der Modellierung häufig geändert wird, muss der Wechsel zwischen Modellieroperation und Kamerasteuerung möglichst effizient erfolgen. In kommerziellen Werkzeugen werden für die Kamerasteuerung deshalb häufig Tastaturkürzel („Shortcuts“) oder in die Ansicht integrierte 3D-Widgets angeboten. Für die Kamerasteuerung können häufig auch spezialisierte Eingabegeräte wie die 3D-Mouse verwendet werden.

Die Multitouch-basierte Kamerasteuerung kann direkt auf den 3D-Raum übertragen werden. In Artist3D [14] erfolgt die Kamerasteuerung, sobald vom Gerät mindestens zwei Finger auf dem Touch-Display erkannt werden. Bei zwei Fingern wird die Veränderung des Abstands zwischen den Fingern zur Skalierung der Szene (bzw. Zoom) verwendet, die Veränderung der Ausrichtung der Verbindungslinie zwischen den beiden Fingern zur Rotation in der Bildebene und die Veränderung des Mittelpunkts der Verbindungslinie zur Translation in der Bildebene.

Dieses Prinzip wird auf 3D übertragen. Die Veränderung der Kameratransformation (View-Matrix) ist dabei äquivalent zur Transformation der gesamten Szene (Model-Matrix). Im nachfolgenden wird deshalb die Transformation der Szene beschrieben. Seien $P1$ und $P2$ die Positionen der Fingerspitzen zweier getrackter Finger zum letzten Abtastzeitpunkt, $P1'$ und $P2'$ die Positionen der gleichen Finger zum aktuellen Abtastzeitpunkt. Die Transformation der Szene lässt sich dann wie folgt berechnen:

$$\begin{aligned}
 Pdiff &= P2 - P1 \\
 Pdiff' &= P2' - P1' \\
 Trf &= Tran(P1') * Skal\left(\frac{\|Pdiff'\|}{\|Pdiff\|}\right) * Rot(Pdiff' \\
 &\quad \times Pdiff, angle(Pdiff', Pdiff)) * Tran(-P1)
 \end{aligned}$$

Die Szene wird zur Position des ersten Fingers beim letzten Abtastzeitpunkt verschoben, dann in der Ebene gedreht, die die beiden Verbindungslinien aufspannen und zwar mit dem Winkel zwischen diesen beiden Linien, dann wird die Szene mit dem Verhältnis der Länge der beiden Verbindungslinien skaliert und schließlich zur Position des ersten Fingers beim aktuellen Abtastzeitpunkt verschoben.

Für den Benutzer entsteht dadurch der Eindruck, dass zwei virtuelle Punkte des 3D-Raums an seinen Fingern haften. Dabei hat er die Möglichkeit die Szene zu verschieben, zu rotieren und zu skalieren. An den Fingerpositionen muss sich kein 3D-Objekt befinden.

Wird der zweite ausgestreckte Finger für eine gewisse Zeitspanne erkannt, startet die Kamerasteuerung. Findet gerade eine Modellieroperation statt (z. B. in „drag“), wird diese unmittelbar beendet („drop“) und rückgängig gemacht, bevor die Veränderung der Kameratransformation beginnt. Der Abbruch der Operation ist für den Benutzer unproblematisch, da er ja ohnehin die Veränderung der Kamera intendiert, wenn er mehr als einen Finger einsetzt. Dieses Prinzip hat sich bei Verwendung des Multitouch-Interfaces bereits bewährt. Bei der Übertragung auf die 3D-Interaktion muss der Benutzer darauf achten, dass er in der Regel nur einen Finger ausstreckt.

Das Beenden der Veränderung der Kameratransformation kann z. B. durch „Entfernen“ eines Fingers oder durch das Stillhalten der beiden Finger über eine gewisse Zeitspanne erfolgen.

5 Ergebnis und Ausblick

Es wurden insgesamt zwei Prototypen erstellt. Der erste Prototyp erweiterte die WIMP-basierte Version von Artist3D um eine Fingersteuerung mit Hilfe des Geräts Leap Motion[16]. Im Grundsatz könnten alle Operationen mit Hilfe des Fingertrackings erfolgen. Der Einfachheit halber wird in dieser Version der Modelliermodus jedoch noch mit Hilfe der Mouse ausgewählt. Die geeignete Anpassung des Interaktionsraums an den ungefähr halbkugelförmigen Trackingbereich von Leap Motion erwies sich als schwierig. Oft ist die Erkennung

der Fingerspitze zu unzuverlässig oder der Finger des Benutzers verlässt ungewollt den Trackingbereich des Geräts, der ja nur Entfernungen von ungefähr maximal einem Meter unterstützt. Da das Gerät eine Stereokamera verwendet, können Finger nur solange zuverlässig erkannt werden, wie der Benutzer Handteller oder Handrücken zur Kamera hin ausrichtet, da sich die Finger sonst gegenseitig verdecken. Die indirekte Steuerung eines 3D-Punkts auf dem Bildschirm erwies sich darüber hinaus als schwierig.

Aus diesen Gründen wurde ein zweiter Prototyp entwickelt. Als Zielplattform diente eine Kombination der Leap Motion mit dem Head-Mounted-Display (HMD) Oculus Rift DK2 [17]. Die Leap Motion wird dabei am HMD befestigt, so dass ein für Augmented-Reality-Anwendungen geeignetes Gerät entsteht.

Der Benutzer sieht durch Überlagerung des Infrarotbildes der Leap Motion seine eigenen Hände. Sofern die Leap Motion die Hände erkennt, wird die Fingerspitze mit der 3D Szene registriert (siehe Abbildung 4).

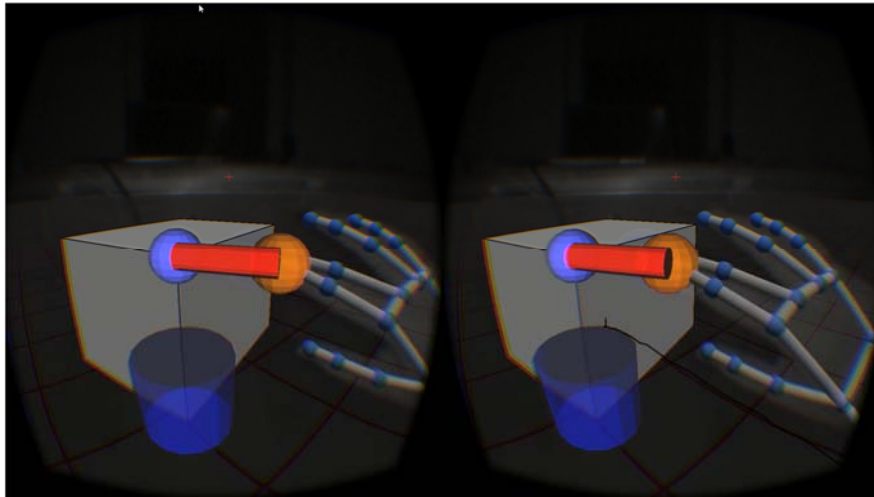


Abb. 4: Bilder für beide Augen des HMDs Oculus Rift, grau überlagert die reale Hand des Benutzer, blau und grau das erkannte Skelett der Hand, orange die Fingerspitze, blau der manipulierte Punkt des Quaders.

Die Steuerung ist im Vergleich zum ersten Prototyp viel einfacher, die Berührung eines virtuellen Objekts mit der realen Fingerspitze ist problemlos ausführbar, solange die Leap Motion die Finger erkennt. Der Benutzer kann dabei ein Gefühl dafür bekommen, wie weit er seine Finger bewegen kann ohne den getrackten Bereich zu verlassen. Die größte Schwierigkeit besteht aber immer noch in der relativ unzuverlässigen Erkennung der Finger durch die Leap Motion.

Erste Tests zeigen jedoch, dass die Modellieroperationen relativ einfach ausgeführt werden können, solange sich die Objekte im Zentrum des Trackingbereich des Geräts befinden. Die Kamerasteuerung mit mehreren Fingern wurde bisher nicht

realisiert, da die Fingererkennung generell noch zu instabil ist.

Im nächsten Schritt soll der beschriebene Ansatz im Rahmen von Usability-Tests überprüft werden. Dabei könnte auch untersucht werden, wie Erfahrungswissen aus der Interaktion mit WIMP- oder Touch-Interface auf die Steuerung mit Fingertracking übertragen werden kann. Bisher wurden noch keine systematischen Usability-Tests durchgeführt, da das beschriebene Verfahren zum Patent angemeldet wird. Es ist jedoch zu überlegen, ob Usability-Tests später nicht gleich mit Hilfe eines leistungsfähigeren Geräts wie z. B. Microsoft Hololens erfolgen sollten.

Literatur

1. <http://www.3dconnexion.de/products/spacemouse.html>, Abrufdatum 15.5.2015
2. <http://www.autodesk.de/products/3ds-max/overview>, Abrufdatum 15.5.2015
3. <http://www.autodesk.de/products/maya/overview>, Abrufdatum 15.5.2015
4. Jacek Jankowski and Martin Hachet: A Survey of Interaction Techniques for Interactive 3D Environments, Eurographics State of the Art Reports, 2013
5. Ben Shneiderman: Direct Manipulation. A Step Beyond Programming Languages, IEEE Transactions on Computers, Vol. 16, No. 8, August 1983, Seiten 57-69
6. <http://www.microsoft.com/microsoft-hololens/en-us>, Abrufdatum 15.5.2015
7. https://www.youtube.com/watch?v=b6sL_5Wgvrg&t=8m22s
8. Paul S. Strauss, Rikk Carey: An object-oriented 3D graphics toolkit, Proc. SIGGRAPH 1992, Seiten 341-349
9. VR Best Practices Guidelines, <https://developer.leapmotion.com/assets/Leap%20Motion%20VR%20Best%20Practices%20Guidelines.pdf>, Abrufdatum 15.5.2015
10. Wolfgang Stuerzlinger, Chadwick A. Wingrave: The Value of Constraints for 3D User Interfaces, Dastuhl-Seminar 2008, Springer 2011, s. a. <http://ws.iat.sfu.ca/papers/constraints3duis.pdf>, Abrufdatum 15.5.2015
11. Doug A. Bowman, Ernst Kruijff, Joseph L. LaViola, Ivan Poupyrev: 3D Userinterfaces Theory and Practice, Addison Wesley 2004
12. <https://apps.leapmotion.com/apps/sculpting/windows>, Abrufdatum 15.5.2015
13. Vinayak, Karthik Ramani: A Gesture-free Geometric Approach for Mid-Air Expression of Design Intent in 3D Virtual Pottery, Journal Computer-Aided Design, July 2015
14. Thomas Jung: Artist3D- Ein einfach bedienbares Werkzeug zur 3D-Modellierung von Oberflächen, In Proc Workshop 3D-NordOst 2006, Berlin, Germany, Dezember 2006, Seiten 13-20
15. <http://www.artist3d.de>, Abrufdatum 15.5.2015
16. <https://www.oculus.com/en-us/dk2/>, Abrufdatum 15.5.2015
17. <https://www.leapmotion.com/>, Abrufdatum 15.5.2015