Bremen

CG VR

Virtual Reality

Interaktionsmetaphern

G. Zachmann

University of Bremen, Germany

cgvr.cs.uni-bremen.de

Historie

- Das erste Computer-Spiel (vermutlich):
  - Spacewars, 1961, MIT
  - Damit auch die ersten Interaktionsgeräte und -metaphern
  - Zwei Spieler, zwei Spaceships ("wedge" und "needle"), feuern Tropedos

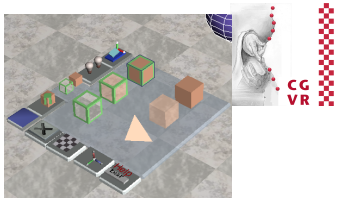
Wie interagiert man mit VEs?

- Grundlegende Aufgaben (= *Universal Interaction Tasks* [Bowman]):
  1. Navigation: Viewpoint ändern
  2. Selektion: Objekt oder Ort für den nächsten Task definieren
  3. Manipulation: Objekte greifen, bewegen, manipulieren
  4. *System control*: Menüs, *Widgets*, *Slider*, Zahlen eingeben, etc.
  - Geometrie modellieren und modifizieren (selten; nicht in Bowman's UITs)
- Elementare Interaktionsbausteine (*BITs = basic interaction tasks* [Foley / vanDam]):
  - Selektion (Objekte, Menüs, ..)
  - Positionierung (inkl. Orientierung) oder Manipulation
  - Quantifizierung
  - Texteingabe, Spracheingabe

Das Design von User-Interfaces

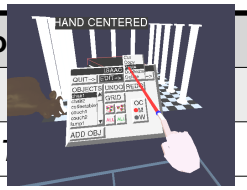
- Zwei grobe Richtungen:
  - Natürliche Interaktion
    - Versuche, die Realität und die Interaktion damit möglichst genau abzubilden
  - "Magische" Interaktion
    - Gib den Usern neue Möglichkeiten
    - Herausforderung dabei: den kognitiven Overhead dabei so klein wie möglich zu halten, so dass der User nicht von seiner Aufgabe abgelenkt wird!
- Hilfsmittel:
  - Direkte *User-Aktion* (Körperbewegung, Geste, ..)
    - Gut wenn intuitiv, Möglichkeiten beschränkt
  - Physikalische Geräte (z.B. Taste, Lenkrad)
    - Haptisches Feedback für präzisere Kontrolle
    - Evtl. schwer zu (er-)finden
  - Virtuelle Geräte (z.B. Menü, "*anything goes*")
    - Flexibel, rekonfigurierbar
    - Nicht leicht/präzise zu bedienen

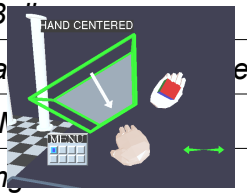
Classification of 3D-Widgets

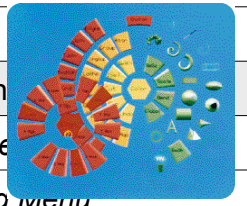


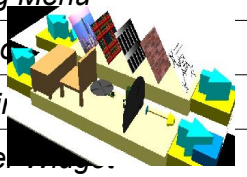
- **Direct 3D Object Interaction**
  - Object Selection
  - Geometric Manipulation
- **3D-Scene Manipulation**
  - Orientation and Navigation
  - Scene Presentation Control
- **Exploration and Visualization**
  - Geometric Exploration
  - Hierarchy Visualization
  - 3D Graph Visualization
  - 2D-Data and Document Visualization
  - Scientific Visualization
- **System / Application Control**
  - State Control / Discrete Valuators
  - Continuous Valuators
  - Special Value Input
  - Menu Selection
  - Containers

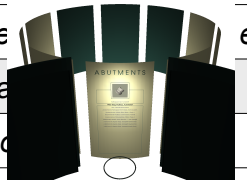
- **Menu Selectio**
  - Temporary
    - *Rotary*
    - *Menu B*
    - *Comma* *e*
    - *Popup M*
    - *Tool Fin*
    - *TULIP*
  - Single Men
    - *Ring me*
    - *Floating Menu*
    - *Drop-Do*
    - *Revolvi*
    - *Choose*
    - *3D-Pale* *etc.*
  - Menu Hiera
    - *Hands-*
    - *Hierarch* *us*
    - *Tool Ra*
    - *3D Pie*
    - → Hiera *ns*

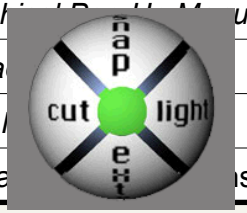












- Ziele (insbesondere in VR):
  - Intuitive / natürliche Interaktion (usability)
    - Leicht zu erlernen
    - Passt sich dem Benutzer an (expert vs. novice)
  - Effiziente Interaktion (user performance)
    - Genauigkeit, Geschwindigkeit, Produktivität des Users
- Probleme (vor allem in VR):
  - Keine Constraints
  - Insbesondere: fehlendes haptisches Feedback
  - Effiziente Interaktion mit Objekten außerhalb der Reichweite
  - Tracker-Rauschen / -Ungenauigkeit
  - Ermüdung
  - Fehlende Standards

There has never been a high performance task done in the history of this planet, to the best of my knowledge, that has ever been done well with an intuitive interface.
[Brian Ferran]

- "Effective Manipulation of Virtual Objects Within Arm's Reach" (VR 2011, Moehring & Froehlich) -- Beispiel dafuer, dass "more realistic" != "more efficient"

Gestenerkennung

- Klassifikationsproblem:
  - Gegeben: Flex-Vektor $x \in \mathbb{R}^n, \quad n \approx 20$
  - Gesucht: Geste $G(x) \in \{\text{“Faust“, “Hitch-hike“ ,...}\}$
- Gesucht: ein Algorithmus, der ..
  - .. benutzer**un**abhängig ist
  - .. robust ist (> 99%)
  - .. schnell ist
  - .. möglichst nur 1x trainiert werden muss (besser: 0x)

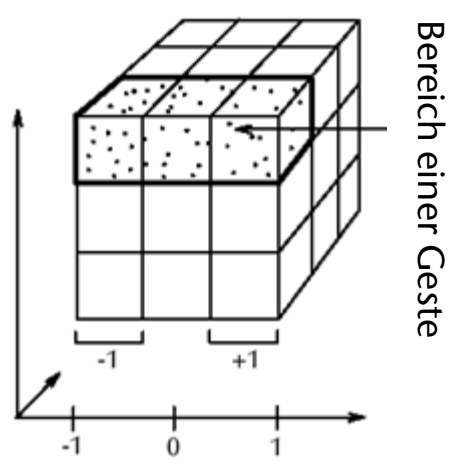
Einfache Gestenerkennung

- NN gut, falls viele Gesten, oder Flex-Werte im "Inneren"
- Falls wenige Gesten und alle "am Rand":
  - Diskretisiere Flex-Vektor $f \in [0,1]^d \rightarrow f' \in \{-1,0,+1\}^d$

    0 = Flex-Wert ist weder nah bei 0, noch nah bei 1
  - Bilde Randregionen im d-dimensionalen diskreten Würfel $\{-1,0,+1\}^d$
  - Wähle für jede Randregion (= Geste) einen Repräsentanten $g \in \{-1,0,+1\}^d$

    0 = *don't care*
  - Geste *i* ist erkannt, wenn

$$f' \cdot g_i = |g_i|$$



  - Bedingung: die Regionen der verschiedenen Gesten dürfen nicht überlappen

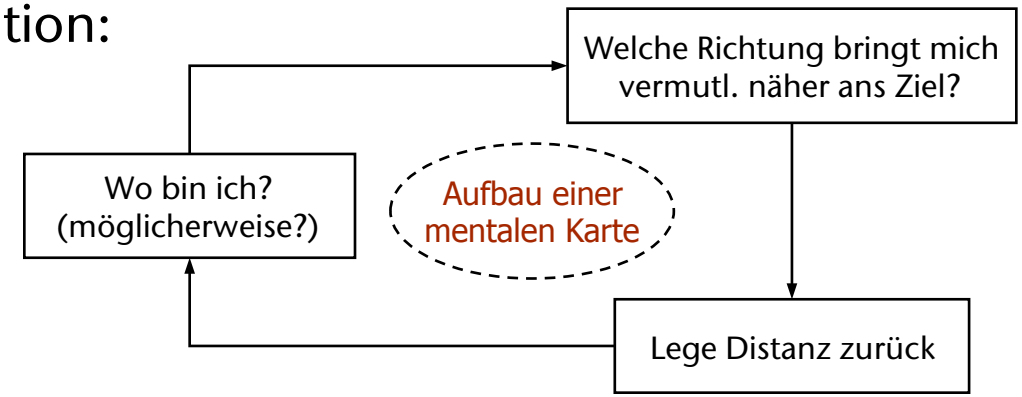
- Implementierungsdetails:
  - Automatische Nachkalibrierung auf [0,1]:
    - Min/Max mitführen und auf [0,1] mappen
    - Min/Max langsam schrumpfen
  - Transitorische Gesten ignorieren
- Dynamische Gesten:
  - Folgen von statischen Gesten (Zeichensprache)
  - Pfad eines Fingers / des Handrückens
  - Nutzen?

Navigation

- *Wayfinding* & *Locomotion*
- *Locomotion / Travel*:
  - Distanz überwinden
  - Maneuvrieren (= Viewpoint setzen, inkl. Orientierung)
  - Technik
- *Wayfinding*:
  - Strategie
  - Wissen

Wayfinding als Aufgabe

- Wie muss die virtuelle Umgebung aussehen, damit *Wayfinding* effektiv trainiert werden kann?
- Hinweise in der Umgebung für *Wayfinding*:
  - Natürliche Hinweise
  - Wegweiser
- User-Modell für Navigation:



- Navigationshilfsmittel:
  - Steigerung der Performance des Users in der virtuellen Umgebung
  - Steigerung in der realen Welt (= Steigerung des Trainingseffekts)

- Erzeugt der Mensch eine mentale Karte seiner Umgebung, um das Wayfinding-Problem zu lösen?
- Antwort: vermutlich ja, aber nicht wie eine gedruckte Straßenkarte, sondern eher wie ein nicht-planarer Graph mit Kantenlängen

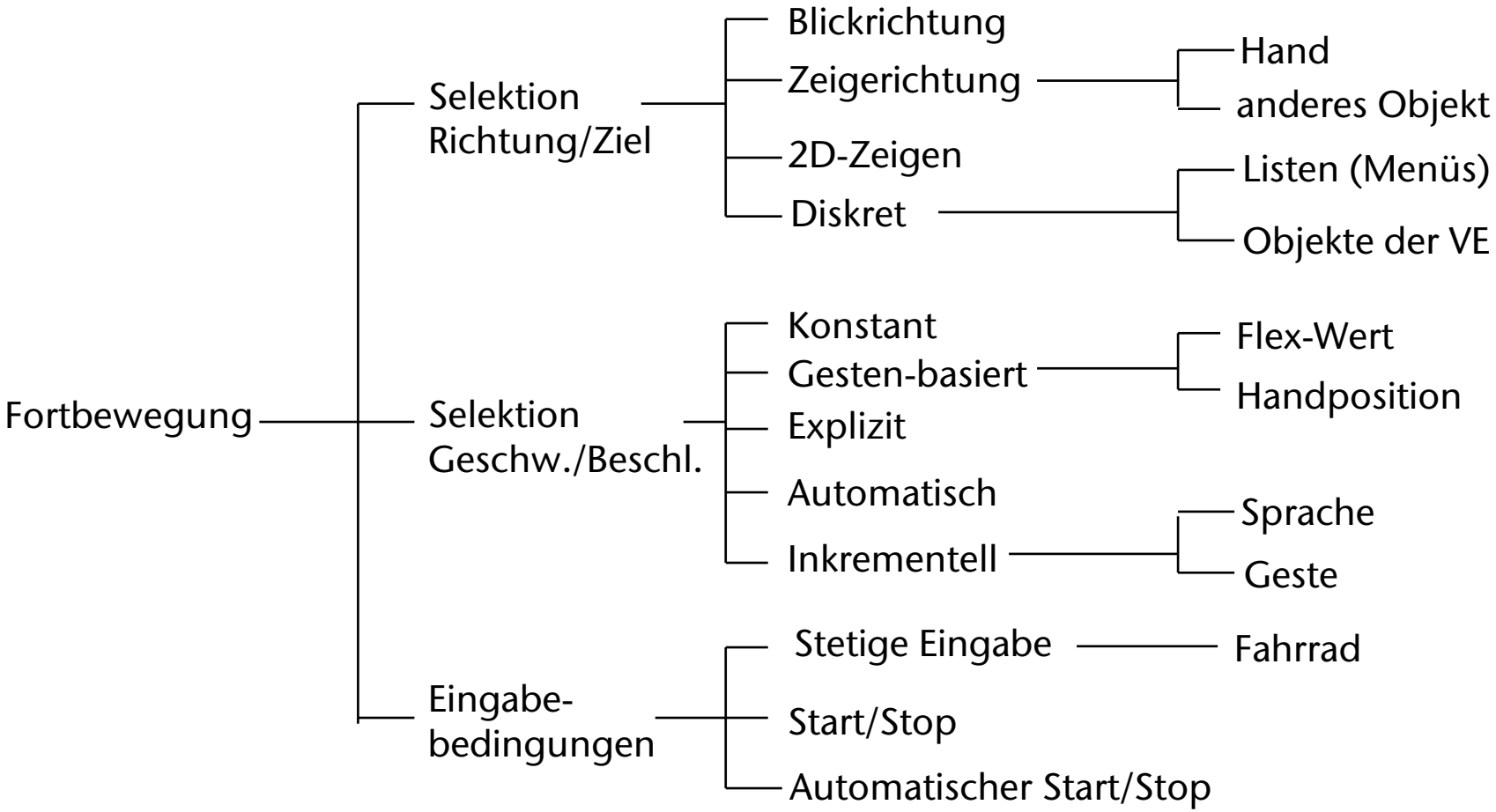
http://www.spiegel.de/wissenschaft/technik/0,1518,739416,00.html

Kerstin Schill, Uni Bremen

Navigationstechniken

- Real: Laufen, Kopf bewegen
- *Point-and-fly* (*Cave*, HMD)
- In Blickrichtung (Boom)
- Fadenkreuz
- *Scene-in-hand*
- *World-in-Miniature*
- Orbital mode
- Richtige Art hängt stark von der Applikation ab!

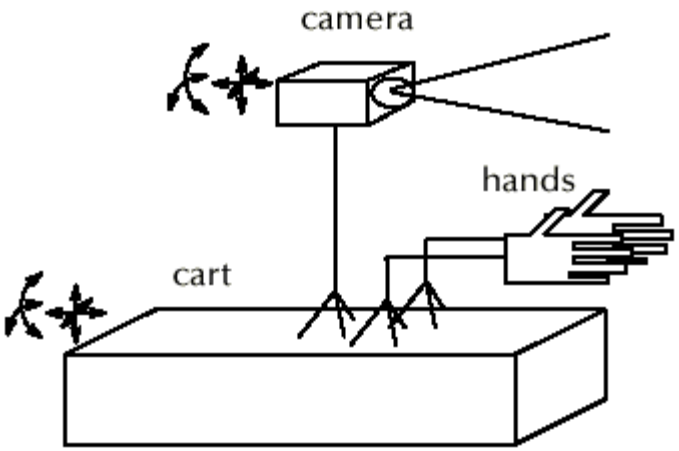
Eine mögliche Taxonomie für diese Interaktionsaufgabe

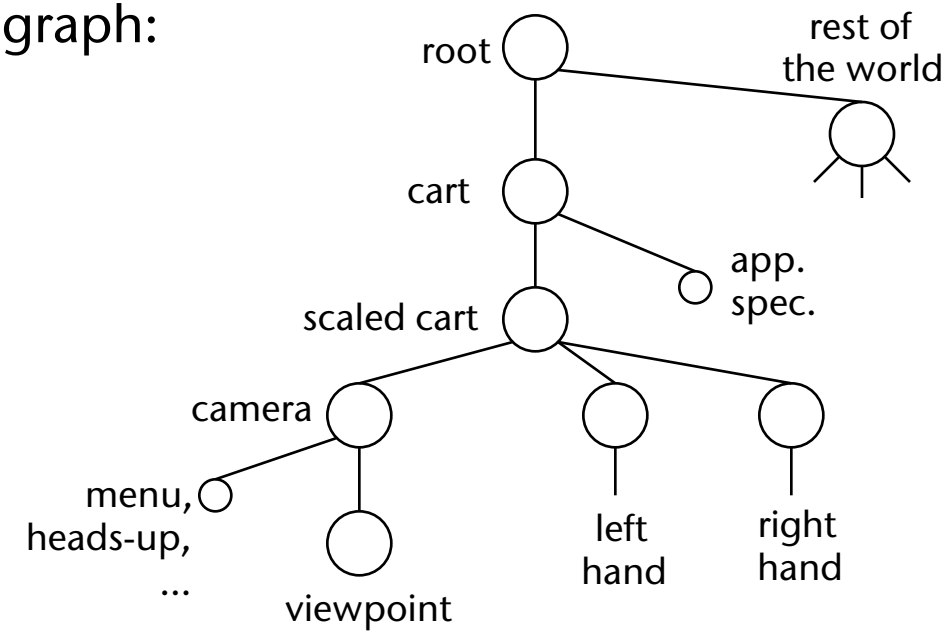


- **Taxonomien sind ein Hilfsmittel, um den *Design-Space* (möglichst vollständig) zu explorieren!**

Repräsentation des Users

- User = Kopf, Hand,
  evtl. ganzer Körper (Avatar)
- Metapher "fliegender Teppich":
  - User → Kamera
  - Kamera steht auf Wagen (Teppich)
- Abbildung in (Teil-)Szenengraph:

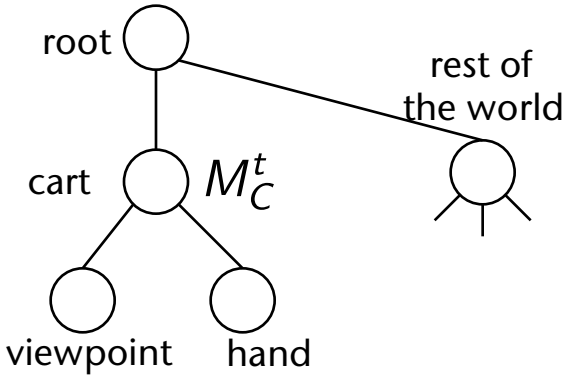




Point-and-Fly

- Kontrollierende Sensoren:
  - Kopfsensor → *Camera*
  - Handsensor → *Cart*

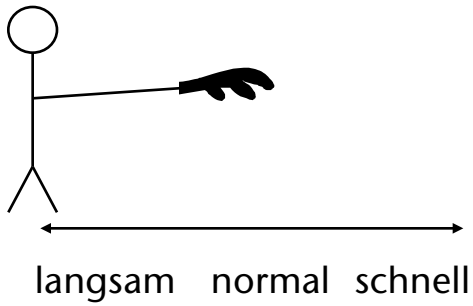
$$M_C^t = M_C^{t-1} \cdot v \cdot \underbrace{T(M_H^z)}$$



Translation, die aus der 3. Spalte der Rot.matrix des Handsensors erzeugt wird

- Verallgemeinerung:
  Graphische Objekte statt Sensoren

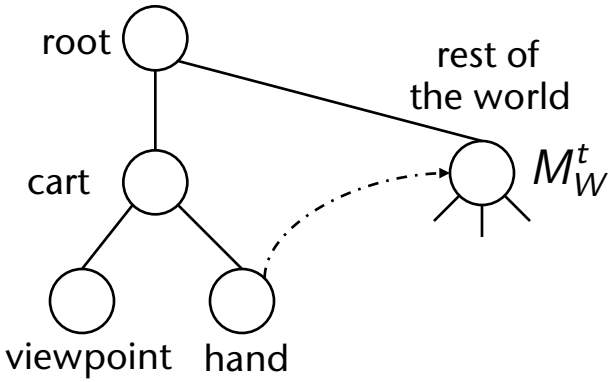
- Spezifikation der Geschwindigkeit:
  - Konstant (z.B. Boom)
  - Daumenkrümmung
  - Abhängig von Entfernung Hand – Brust
  - Manchmal unabhängig von *Framerate*



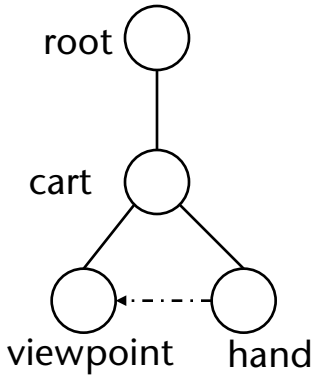
Scene-in-hand, Eyeball-in-hand

- *Scene-in-hand*:
  - *"grabbing the air" technique*
  - *Cart* bleiben stehen, Szene wird rotiert durch Handsensor
  - Die Transformation:

$$M_W^t = M_H^{t_0^{-1}} \cdot M_H^t \cdot M_W^{t_0}$$

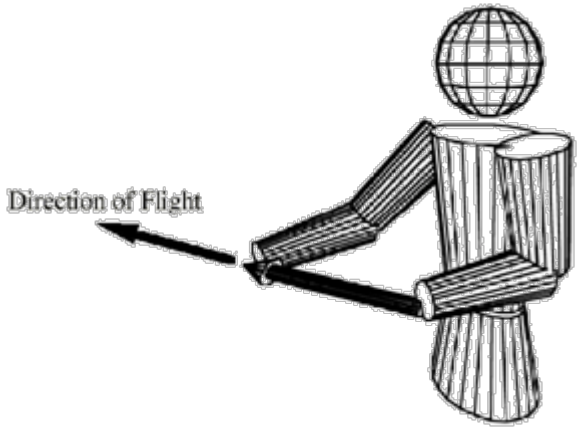


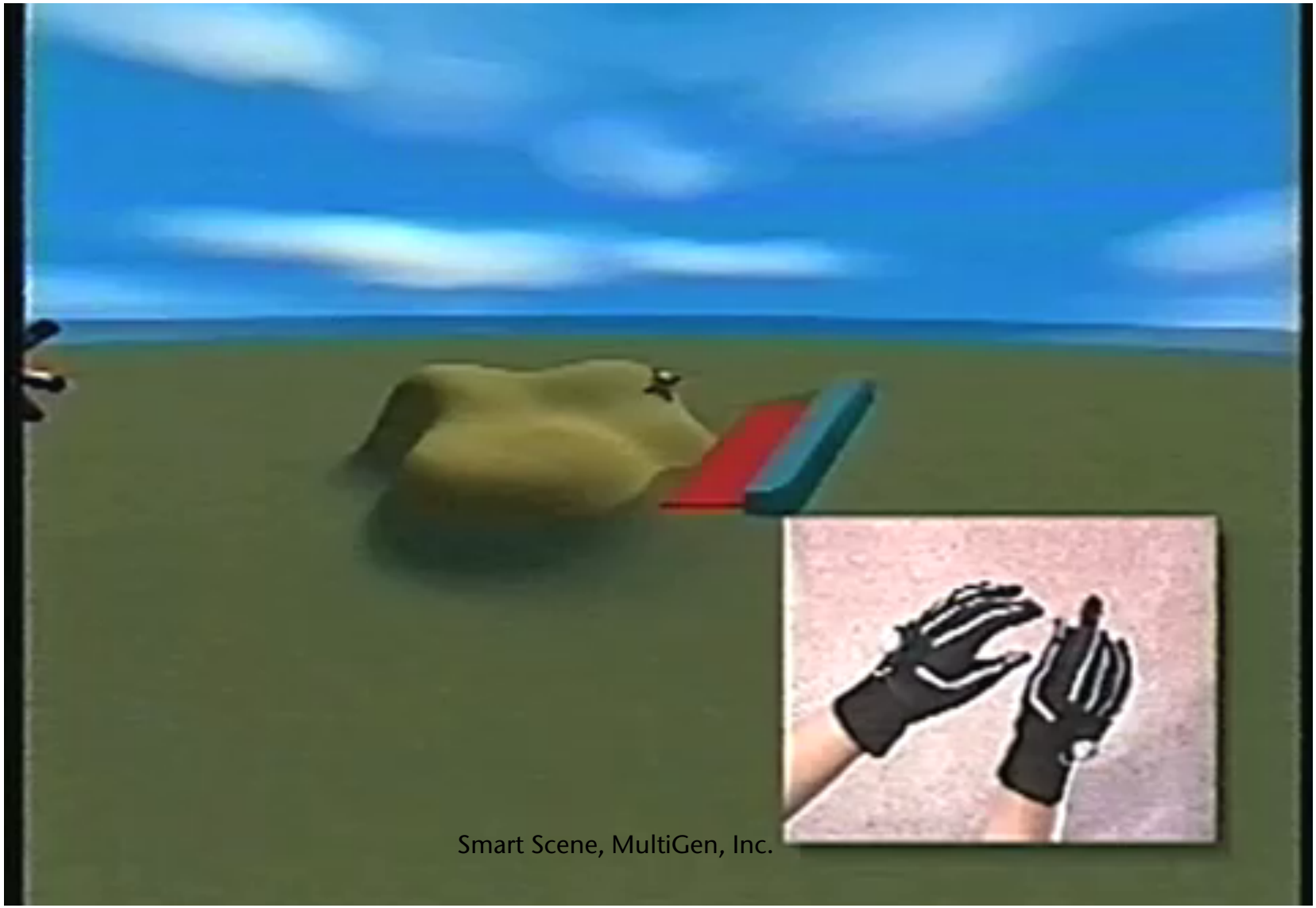
- *Eyeball-in-hand*:
  - Viewpoint wird direkt durch Hand gesteuert
  - Absolut oder relativ (akkumulierend)



Beidhändige Navigation (mit *Pinch Gloves*)

- Fragestellung: wie kann man mit 2 Punkten + 1–2 Trigger navigieren?
- Idee: "*scene-in-hand*"
  - 1 Trigger, 1 Punkt bewegt → Translation der Szene
  - 2 Trigger, 1 Punkt fest, 1 Punkt bewegt → Rotation der Szene
  - 2 Trigger, 2 Punkte bewegt → Skalierung der Szene
- Hat sich trotzdem nicht durchgesetzt
  (vermutlich, weil *Pinch Gloves* sich nicht durchgesetzt haben)
- Zwei-händige Navigation (Variation):
  - Vektor zwischen Händen = Richtung
  - Länge des Vektors = Geschwindigkeit




Smart Scene, MultiGen, Inc.

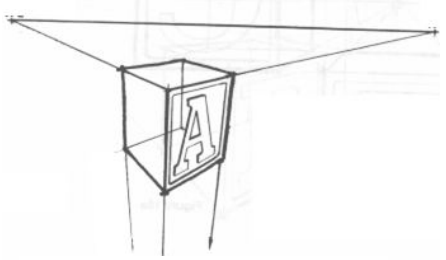
Navigation ohne Hände

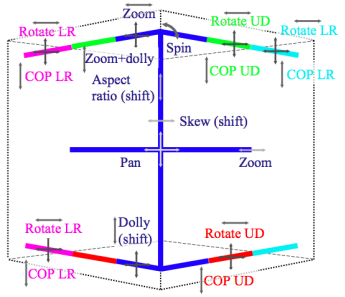
- Idee: projiziere VE verkleinert auf Boden und verwende Füße
- Grobe Navigation:
  Teleportation → User läuft zu Punkt auf Karte und triggert
- Systemkommandos:
  1. Karte einschalten = Blick zu Boden + Trigger
  2. Teleportation = Blick zu Boden + Trigger
  3. Karte ausschalten = Blick nach oben + Trigger
  - Trigger = Sprache oder "Fußgeste"

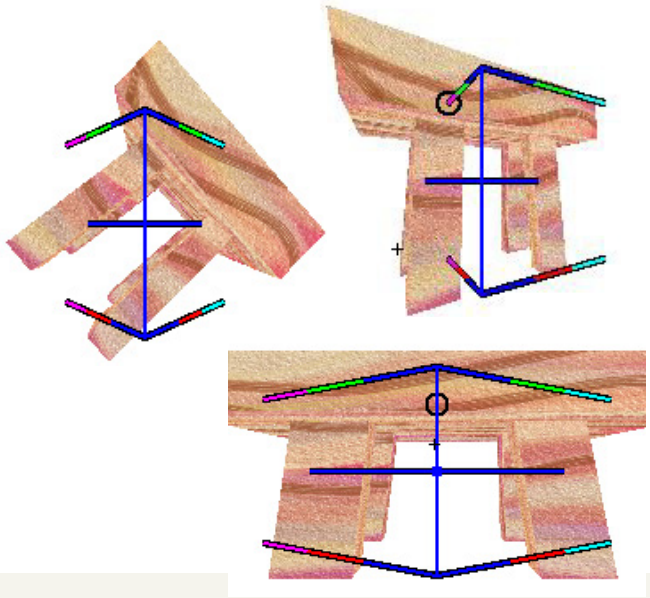
- Feine Navigation: in gewünschte Richtung "lehnen";
  Geschwindigkeit häbgt ab von
  - Neigungswinkel
  - Abstand vom Rand der Cave

Exkurs: das IBar — eine intuitive 2D-Metapher

- Aufgabe: intuitive Metapher zur Manipulation der perspektivischen Projektion
- Beobachtung: Zeichner konstruieren die Projektion durch Fluchtpunkte
- Idee:
  - Manipuliere diese Fluchtpunkte
  - Verwende dazu das Bild der Kanten eines Würfels
- Durch Manipulieren der "Handles" kann man verschiedene Parameter modifizieren:
  - Orientierung, Zoom, Pan, Proj.zentrum

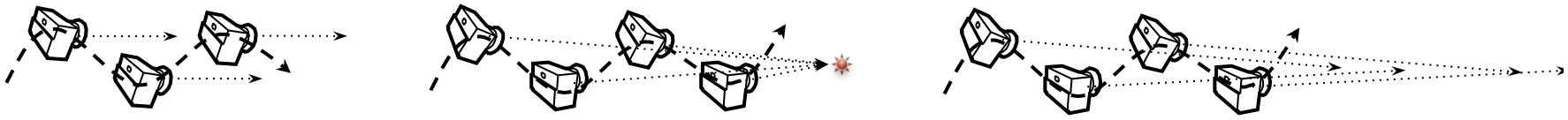






Wahrnehmung der zurückgelegten Distanz in VR [2009]

- Hängt zusammen mit der Frage: wie gut ist die Präsenz bei Navigation in VR?
- Idee:
  - Oszillation des Viewpoints wie in der Realität nachbilden
  - (Haben die first-person-shooter schon viel früher entdeckt ;-))

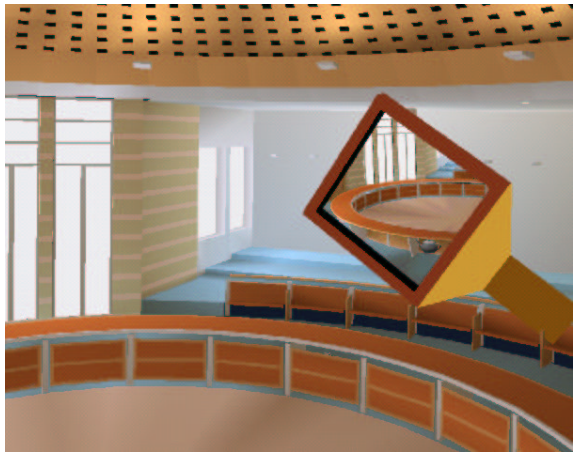


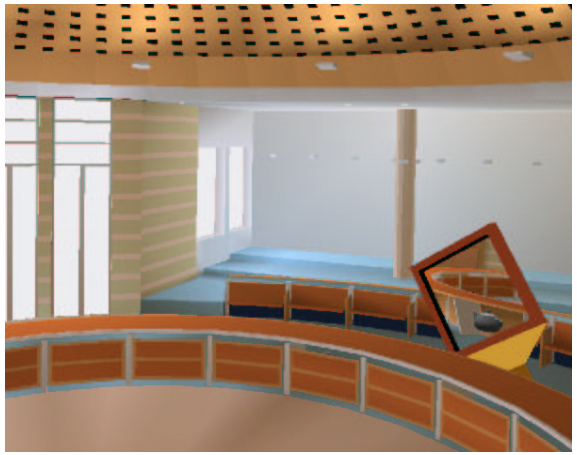
- Resultate:
  - Nur eine Oszillation entlang der Hochachse bringt etwas
  - User ziehen leichte Oszillation der Navigation ohne Oszillation vor
  - Kurze "Reisedistanzen" werden genauer eingeschätzt (ca. Faktor 2)

Exploration von Szenen: der Magic Mirror

- Aufgabenstellung: zweiten Viewpoint (Bild im Bild) intuitiv verständlich in eine Szene integrieren und manipulierbar machen
- Idee: der Spiegel → "magic mirror"
  - Ein Objekt in der VE dient als Handspiegel
  - Wird immer relativ zur Kamera positioniert (wandert mit)
  - Kann man manipulieren wie jedes andere Objekt auch
- Zusatz-Features (nicht bei realen Spiegeln):
  - Zoomen
  - Vergrößern / verkleinern des Spiegels
  - Clippen von Objekten vor dem Spiegel, die die Sicht versperren
  - "Richtig-herum-Drehen" der Szene im Spiegel
  - Positionieren des Haupt-Viewpoints an der Stelle des gespiegelten VP

- Beispiele:



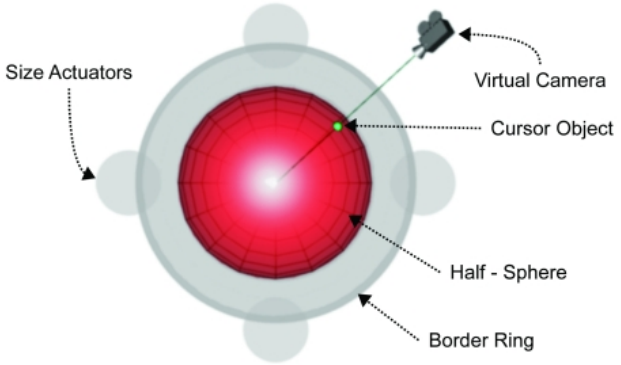


- Implementierung:
  - 2x rendern
  - Erstes Mal nur einen kleinen Viewport mit dem gespiegelten Viewpoint
  - Abspeichern als Textur
  - Zweites Mal main view rendern, ohne Spiegel
  - Dann Spiegel-Objekt drüber rendern ohne Z-Test

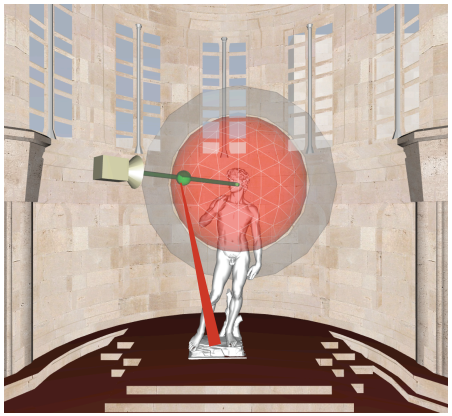
Immersive Navidget

[2008]

- Metapher zur Definition eines Viewpoints
- Eingabegerät: *Wand* mit Rädchen und Buttons
- Dekomposition:



1. Mittelpunkt einer Kugel festlegen
   - Wird der neue Center of Interest (COI)
   - Z.B. durch Ray-Casting: Schnittpunkt mit Szene = Mittelpunkt
2. Radius der Kugel festlegen = Abstand zum COI
   - Hier mit dem Rädchen am Wand
3. Viewpoint auf der Kugel bestimmen
4. Animation des Viewpoints auf einem Pfad zum neuen Viewpoint
5. Wechsel zwischen den einzelnen Phasen mit dem Button




Navidget for Immersive
Virtual Environments
Sebastian Knödel, Martin Hachet
iparla.labri.fr

Exkurs: User Models

- Idee: wenn man ein Modell davon hat, wie ein User funktioniert, dann kann man vorhersagen, wie er/sie mit einem bestimmten UI interagieren wird, insbesondere seine sog. "*user performance*"
- Vorteil (theoretisch): keine *user studies* und keine *UI mock-ups* mehr nötig
- Verwandte Gebiete: *Psychophysics, user interface design, usability*

Power law of practice

- Beschreibt, in welcher Zeit eine Tätigkeit nach der *n*-ten Wiederholung ausgeführt werden kann:

$$T_n = \frac{T_1}{n^a}$$

T_1 = Zeit für die erste Ausführung der Tätigkeit,

T_n = Zeit für die *n*-te Wiederholung,

$a \approx 0.2 \ldots 0.6$

  - Gilt nur für mechanische Tätigkeiten, z.B.:
    - Erlernen der Benutzung der Maus, oder Tippen auf der Tastatur
  - ... nicht für das Erlernen von Wissen! ;-)
- Dieser Effekt hat auch Auswirkungen auf Experimente mit Usern!

Hick's law

- Beschreibt die Zeit, die man benötigt, um eine 1-aus-*n* Auswahl zu treffen, bei der keine kognitive Leistung nötig sein darf:

$$T = I_c \log_2(n+1)$$

  $I_c \approx 150$ msec

  - Annahme: alle Möglichkeiten kommen gleich häufig vor!
  - Hat etwas mit der informationstheoretischen Entropie zu tun
  - Beispiel: *n* Tasten, *n* Lampen, eine wird zufällig angeschaltet, User muss zugehörige Taste drücken
- (Folge für UI Design: die sog. "*Rule of Large Menus*": one large menu is more time-efficient than several small submenus supporting the same choices, even if we ignore the time overhead of moving among submenus.)
  - Achtung: andere Effekte spielen evtl. eine größere Rolle (z.B. Fitts' Law)

Fitts' Law

- Beschreibt die Zeit benötigt zur sog. "target acquisition"
  - Aufgabe: mit der Hand aus der Ruhelage ein bestimmtes Ziel möglichst schnell erreichen und möglichst exakt treffen
  - Das Gesetz:

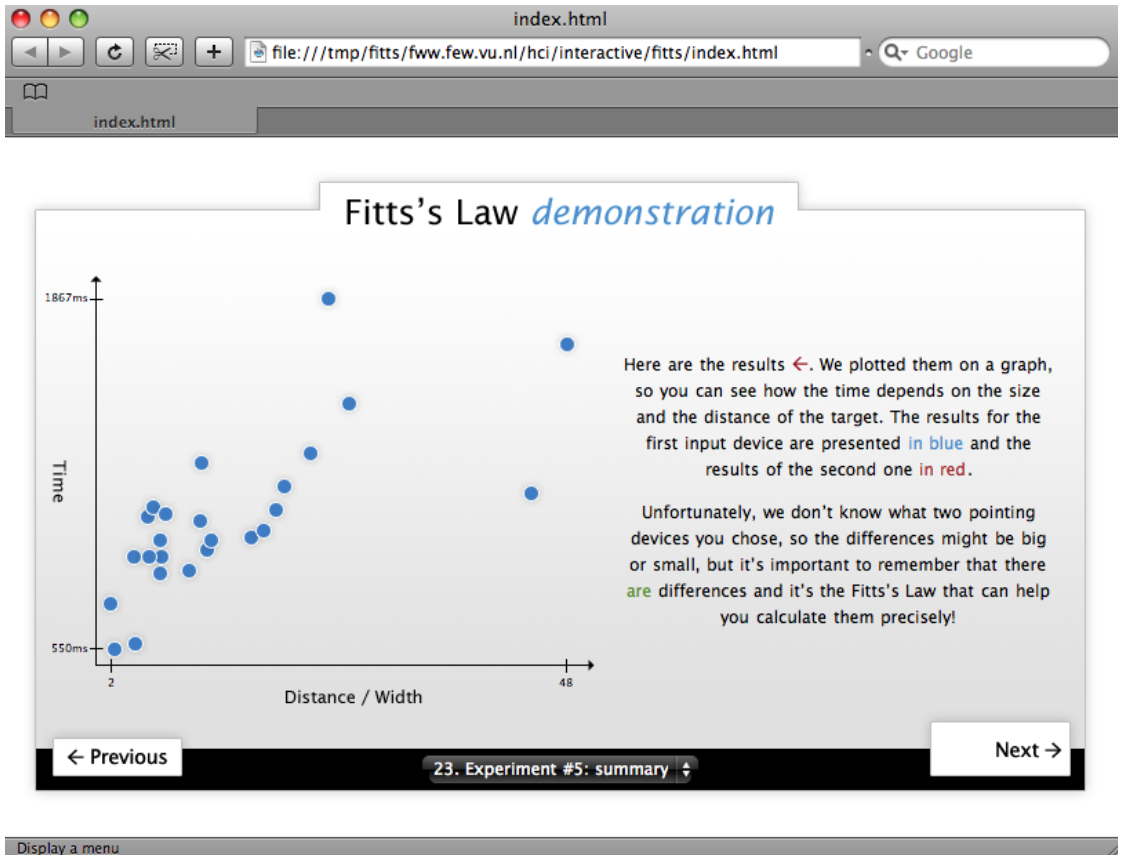
$$T = b \log_2\left(\frac{D}{W} + 1\right) + a$$

    wobei D = Target-Distanz, W = Target-Durchmesser
  - Der "index of difficulty" (ID):

$$\log_2\left(\frac{D}{W} + 1\right)$$

Demo / Experiment

- Fitt's Law läßt sich 1:1 auf Mausbewegungen und das "Anfahren" von Icons übertragen



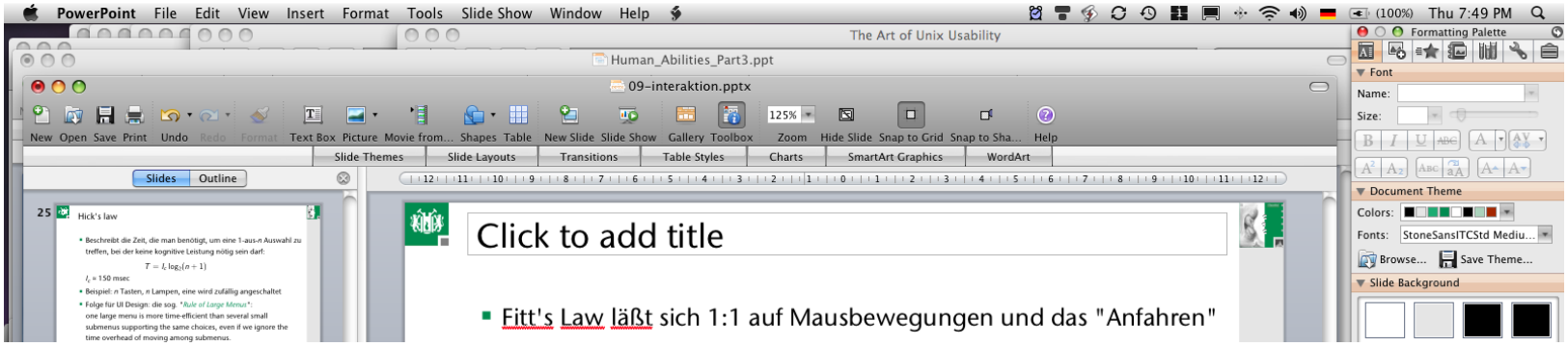
Marcin Wichary , Vrije Universiteit: http://fww.few.vu.nl/hci/interactive/fitts/

Anwendungen von Fitts' Law

- "Rule of Target Size": The size of a button should be proportional to its expected frequency of use.
- Weitere Folge:

  "*Macintosh fans like to point out that Fitts's Law implies a very large advantage for Mac-style edge-of-screen menus with no borders, because they effectively extend the depth of the target area off-screen. This prediction is verified by experiment.*"

  [Raymond & Landley: "The Art of Unix Usability", 2004]



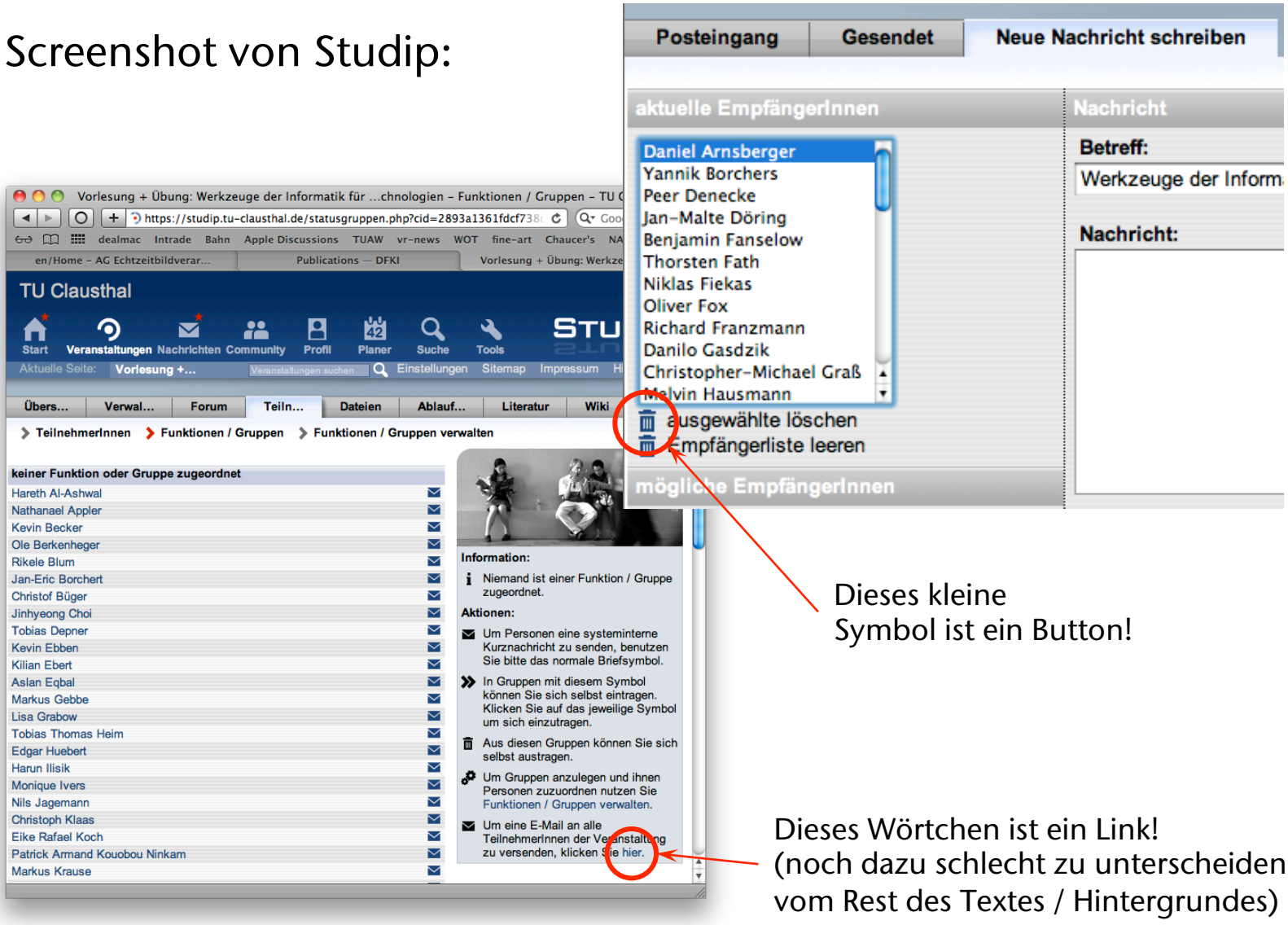
- *Tear-off menus* und *context menus*: damit wird die durchschnittliche Distanz verringert
- Apple's "Dock": die Größe der Buttons wird dynamisch angepasst

- Offensichtliche Grenzen von Fitts' Law:
  - Es gibt viele weitere Entscheidungen bzgl. eines Interface Design's, die einer konsequenten Umsetzung von Fitts' Law entgegenstehen

Unterhaltsames und lehrreiches Quiz:
http://zach.in.tu-clausthal.de/teaching/vr_literatur/A_Quiz_Designed_to_Give_You_Fitts.html

Schlechte Beispiele

- Screenshot von Studip:



Exkurs vom Exkurs: die 80/20-Regel

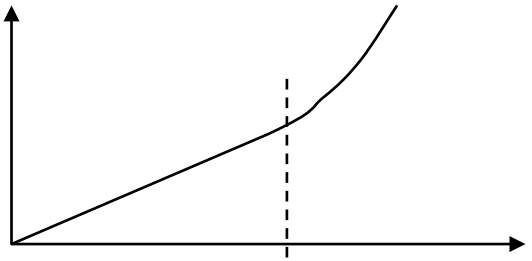
- 80% der Zeit benutzen wir nur 20% der Funktionen eines Produktes
  - Gilt für Menus, komplette Software, "consumer electronics", Auto, ...
- 80% aller Fehler eines Produktes entstehen in nur 20% seiner Komponenten
- 80% aller Fehler in einer Software werden von nur 20% seiner Programmierer und Designer verursacht
- 80% des Einkommens einer Firma werden von nur 20% ihrer Produkte generiert
- ...

Selektion

- *Task decomposition*:
  1. Selektionsmode an
  2. Objekt(e) auswählen
     - Währenddessen Feedback geben
  3. Bestätigen / abbrechen
  4. Feedback: welches Objekt ist selektiert (evtl. mehrere?)
- Definitionen:
  - Interaktionsraum (display / visual space) = Raum in der VE = Raum, in dem der virtuelle "Pointer" (z.B. virtuelle Hand) sich bewegt
  - Physikalischer Raum (control / motor space) = Raum außerhalb der VE = Raum, in dem der Tracker sich bewegt
  - Control-Display ratio (C-D ratio): Verhältnis zwischen Bewegung (Translation und/oder Rotation) im physikalischen Raum zu resultierender Bewegung im Interaktionsraum
    - Einfachstes Beispiel: 2D mouse acceleration

Direkte Selektion & Manipulation mit nicht-linearem Mapping (die "*go-go technique*")

- Direkte Selektion/Manipulation = direktes Berühren & Festhalten der Objekte mit der virtuellen Hand
- Ziel: Vergrößerung des Arbeitsbereiches
- Idee:
  - Tracker-Werte außerhalb des "Nahbereiches" nicht-linear skalieren
  - Im Nahbereich linear belassen wg. Präzision
- Geeignet für Kopf- und Hand-Tracking
- Nur bei absoluten Eingabegeräten
- Nachteile:
  - Propriozeption geht verloren
  - Geringere Präzision im Fernbereich

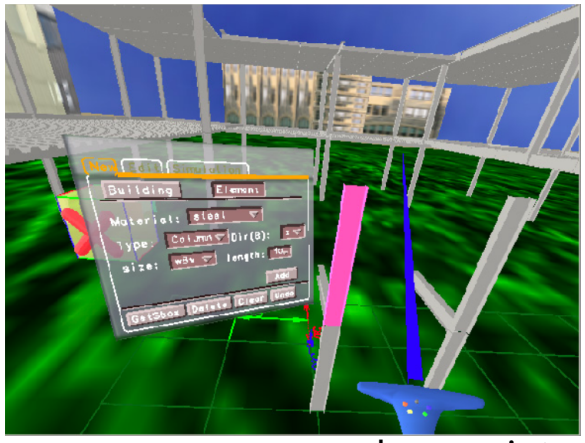


The Go-Go Interaction Technique:

... to reach farther in virtual environments

Einige Möglichkeiten für Schritt 2

- Strahl-basiert (*ray casting*)
  - Z.B. "Laserstrahl" aus virtueller Hand
  - Oder: gedachter Strahl vom Viewpoint durch Zeigefingerspitze (a.k.a. *occlusion technique* oder "sticky finger" technique)
- Volumen-basiert, z.B. Kegel
- Direkt = Berühren mit Hand
- Sprache
- Menü
- Mischformen:
  - *image plane interaction* (später)
  - World-in-Miniature (später)
  - Etc.



laser pointer

occlusion technique

Einige strahlbasierte Techniken im Überblick

- Verwendete Größen:

  H = Handposition

  E = Viewpoint

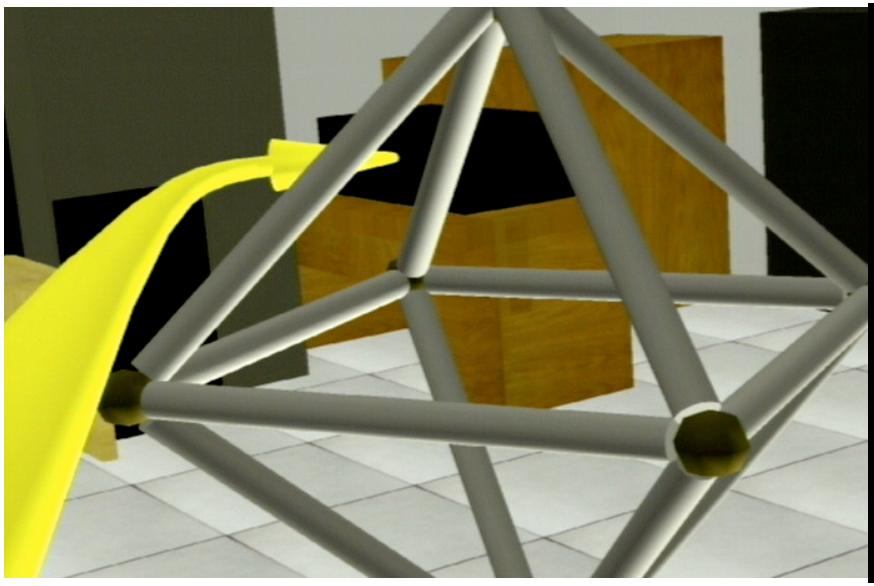
  h = "Zeigerichtung" der Hand

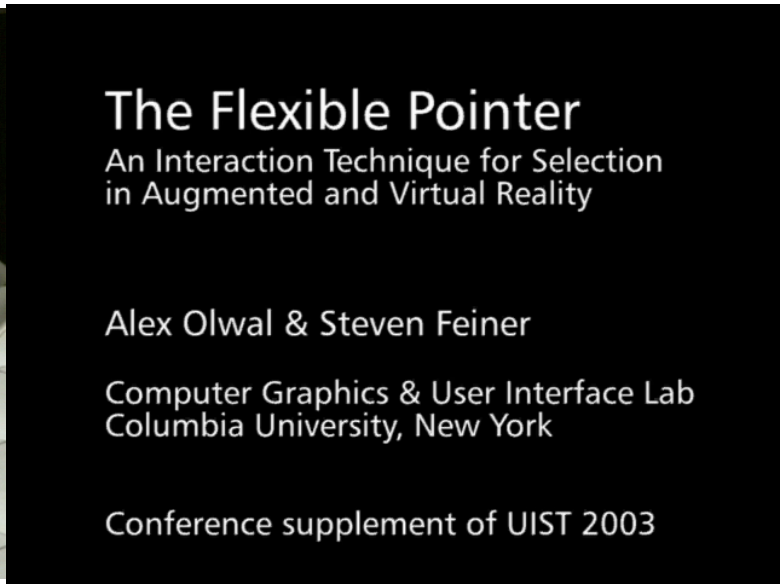
  H_2 = Position der linken Hand

Technik	Volumen	Ursprung	Richtung
Raycasting	ray	H	h
Flashlight	cone	H	h
Two-handed pointing	ray	H_2	$H - H_2$
Occlusion selection	ray	E	H - E
Aperture	cone	E	H - E

Flexible Pointer [2003]

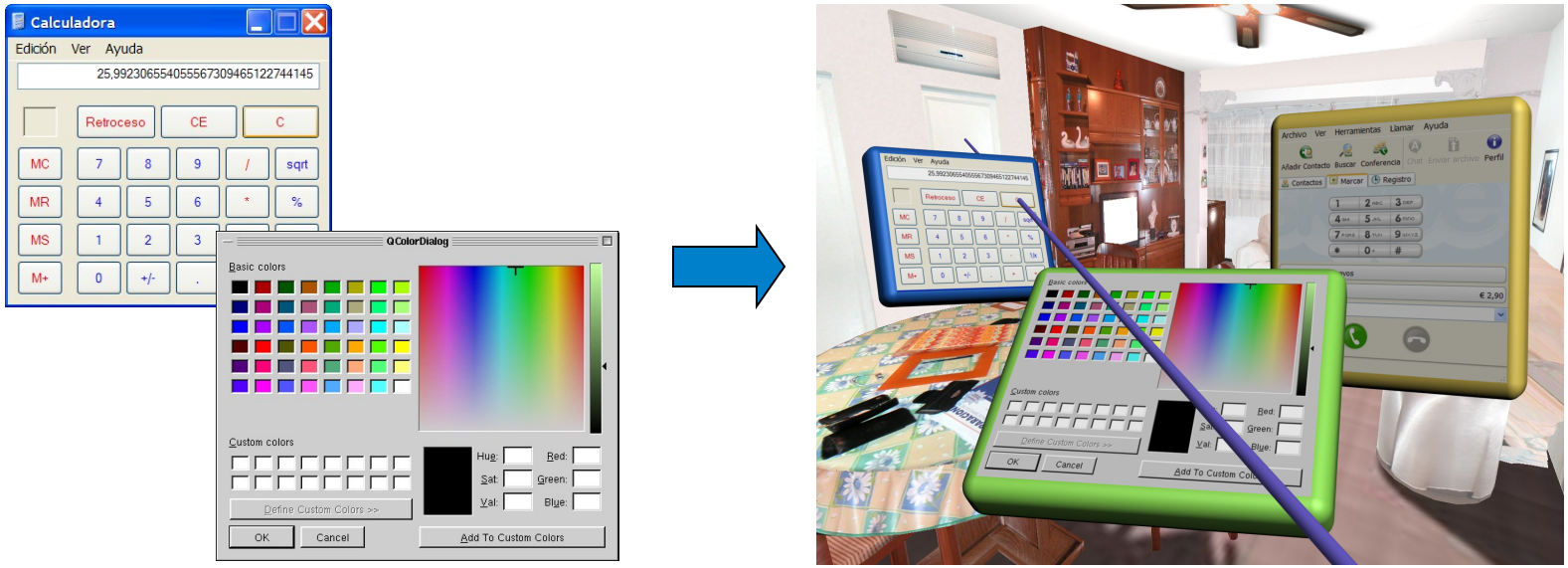
- Beobachtung: Menschen versuchen, mit der Zeigegeste eine "Kurve" zu beschreiben, wenn sie auf etwas zeigen, das nicht in der "*line of sight*" ist.
- Umsetzung in VR: gebogener Zeigestrahl
- Problem: intuitive und einfache Beschreibung der Krümmung mittels Eingabegeräten (Dataglove, Tracker, ...)





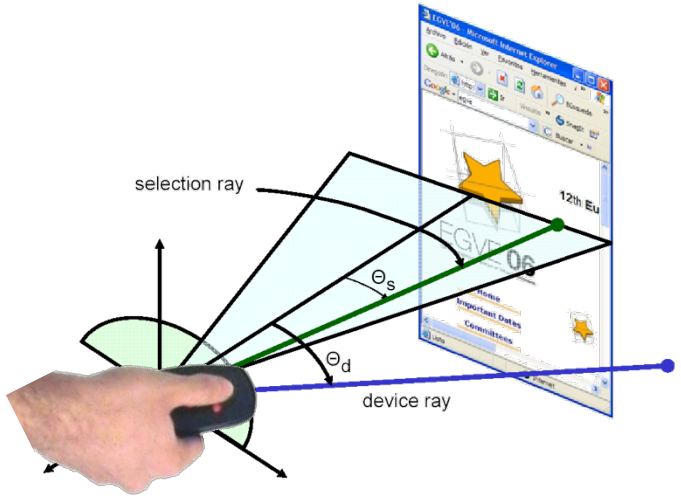
Friction Surfaces — Beispiel für die C-D Ratio [2006]

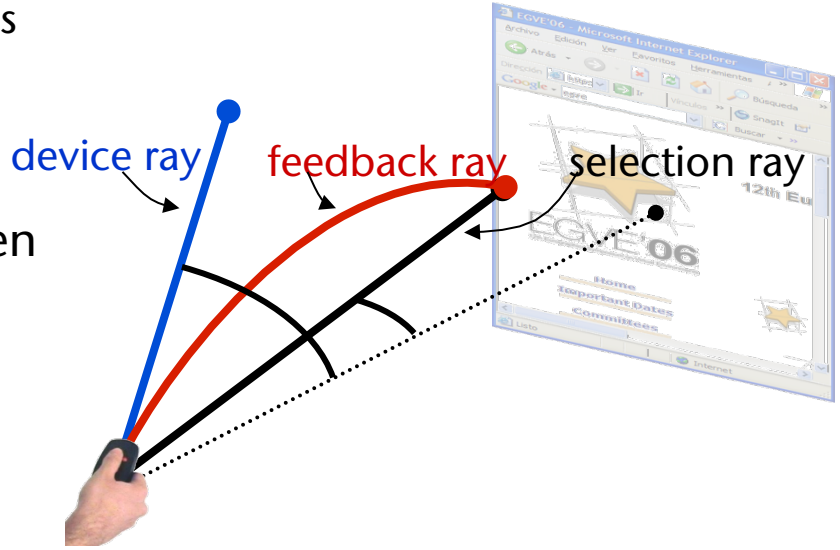
- Aufgabe hier: sog. hybride Interfaces bedienen
  - Ziel: 2D-GUIs von Desktop-Applikationen in VR bedienen
  - Implementierung: ein modifizierter VNC-Client



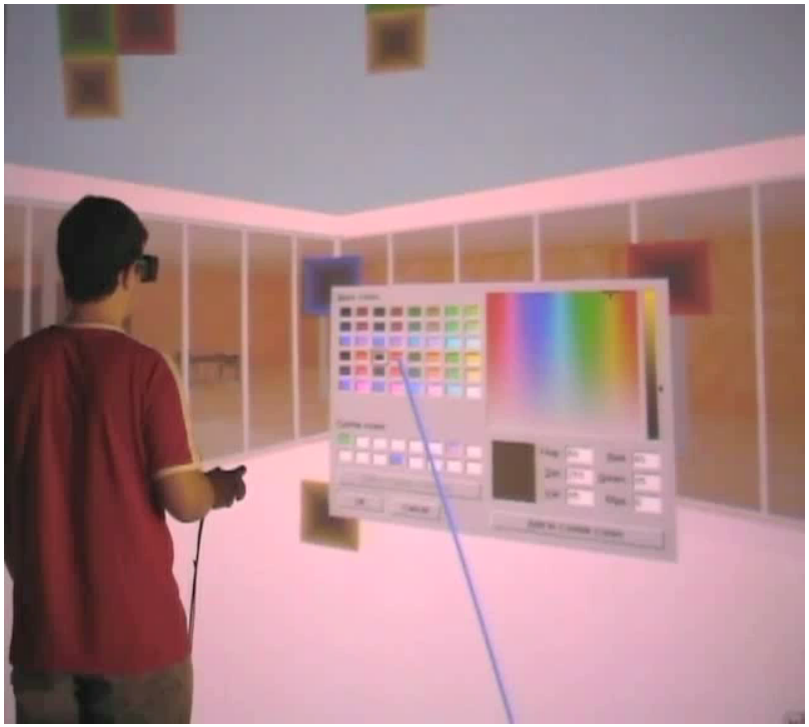
- Problem: die Target-Width (hier Raumwinkel!) ist extrem klein

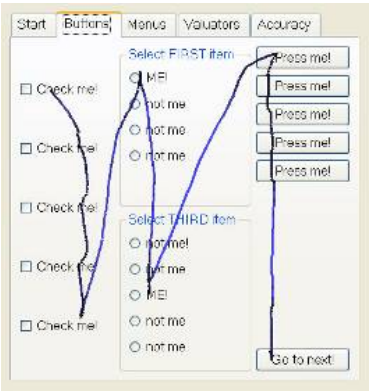
- Idee:
  - Skaliere die C-D Ratio, sobald der User mit einem 2D-Window in VR interagiert
  - Problem: wie überbrückt man die für den User sichtbare/spürbare Diskrepanz?
    - Zwei Strahlen anzeigen hat sich als störend erwiesen
  - Lösung:
    einen gebogenen Strahl anzeigen

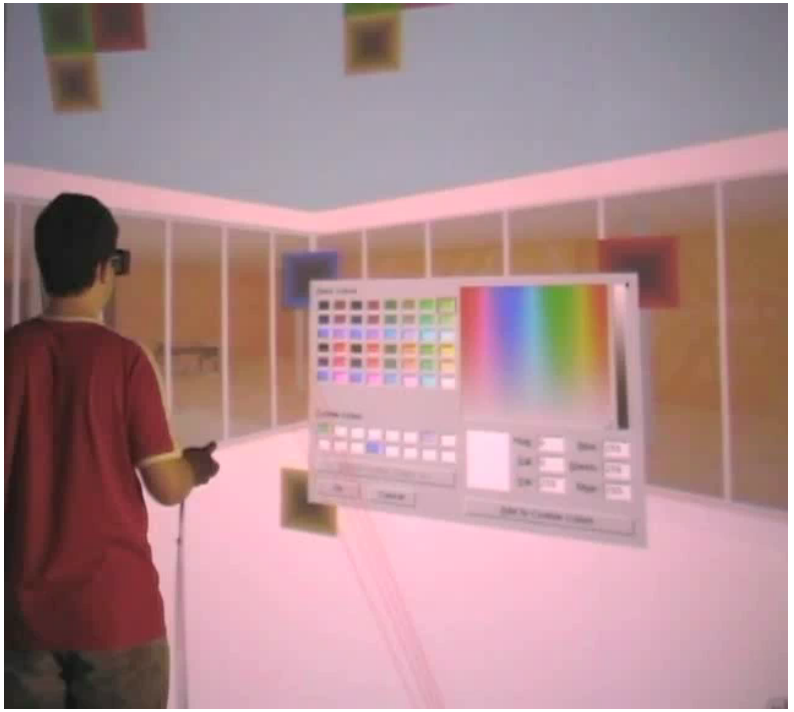


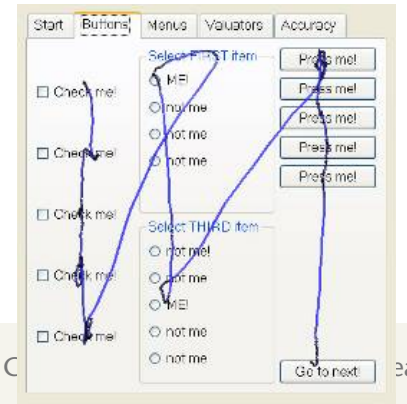


- Resultat: wesentlich höhere User-Effizienz:





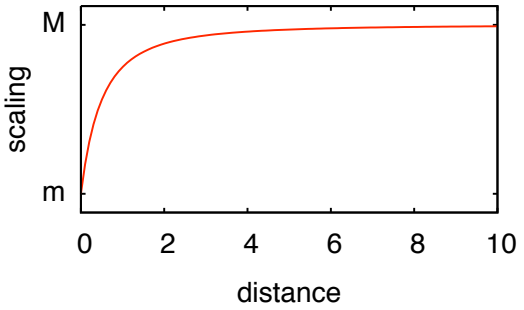




Nicht-konstante C-D Ratio: "Semantic" pointing [2008]

- Idee:
  - Modifiziere C-D-Ratio abhängig von der Entfernung des Cursors zum nächsten Target
  - Große Entfernung → große Skalierung der Bewegung in Motor Space
  - Kleine Entfernung → kleine Skalierung = hohe Präzision
  - Z.B. mit einer Funktion wie dieser:

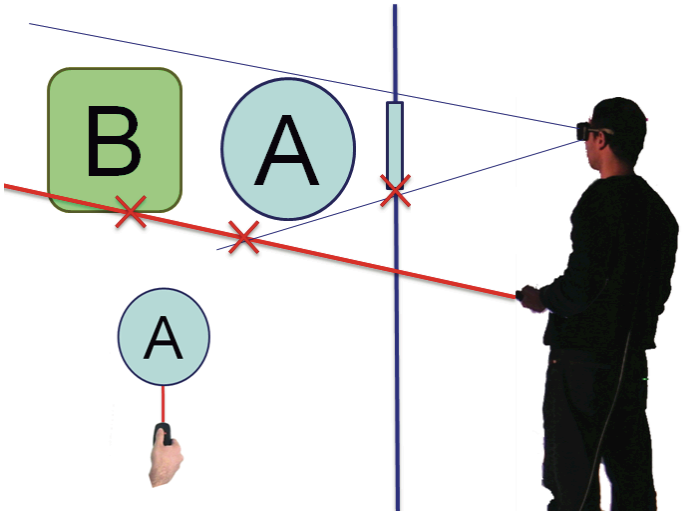
$$s(d) = M + \frac{m - M}{(1 + d)^{\alpha}}$$



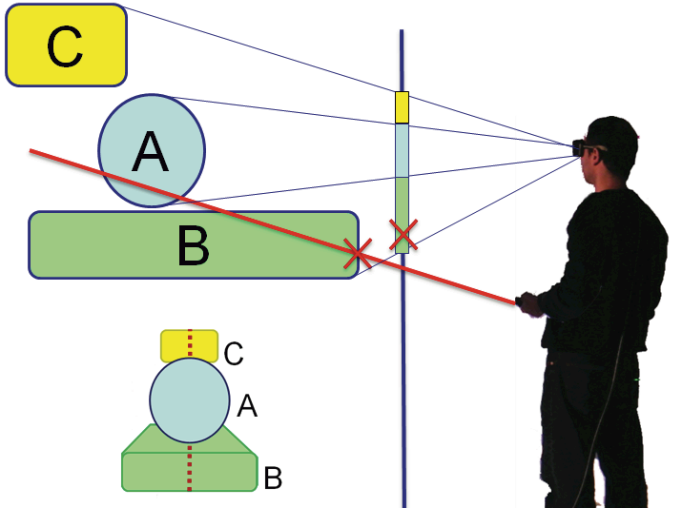
- Visuelles Feedback:
  - Cursor-Größe ~ C-D-Ratio
  - Farbe des Cursors signalisiert Nähe des Targets (z.B. "rot" = "hit")

The Eye-Hand Visibility Mismatch [2008]

- Offensichtliches Problem von Handstrahl-basierten Techniken:
  - Die Menge der von E aus sichtbaren Objekte ist nicht identisch mit der von H aus "sichtbaren" Menge

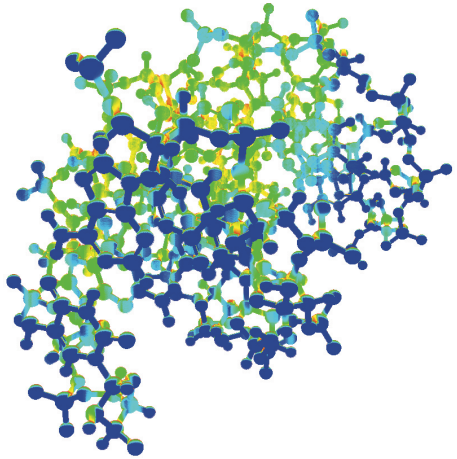


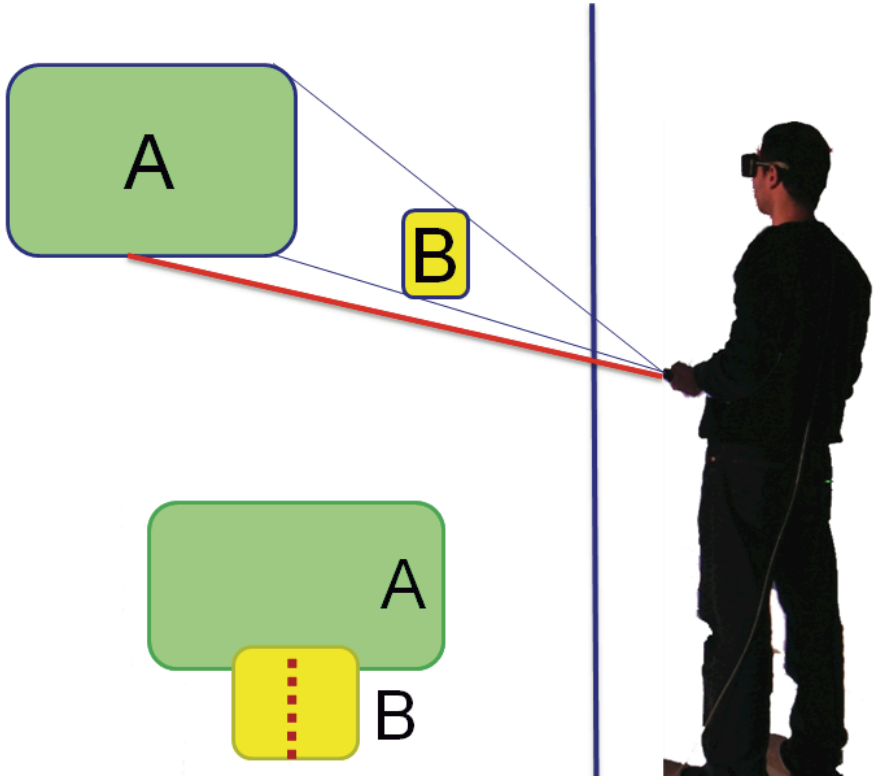
Objekt ist selektierbar,
aber nicht sichtbar



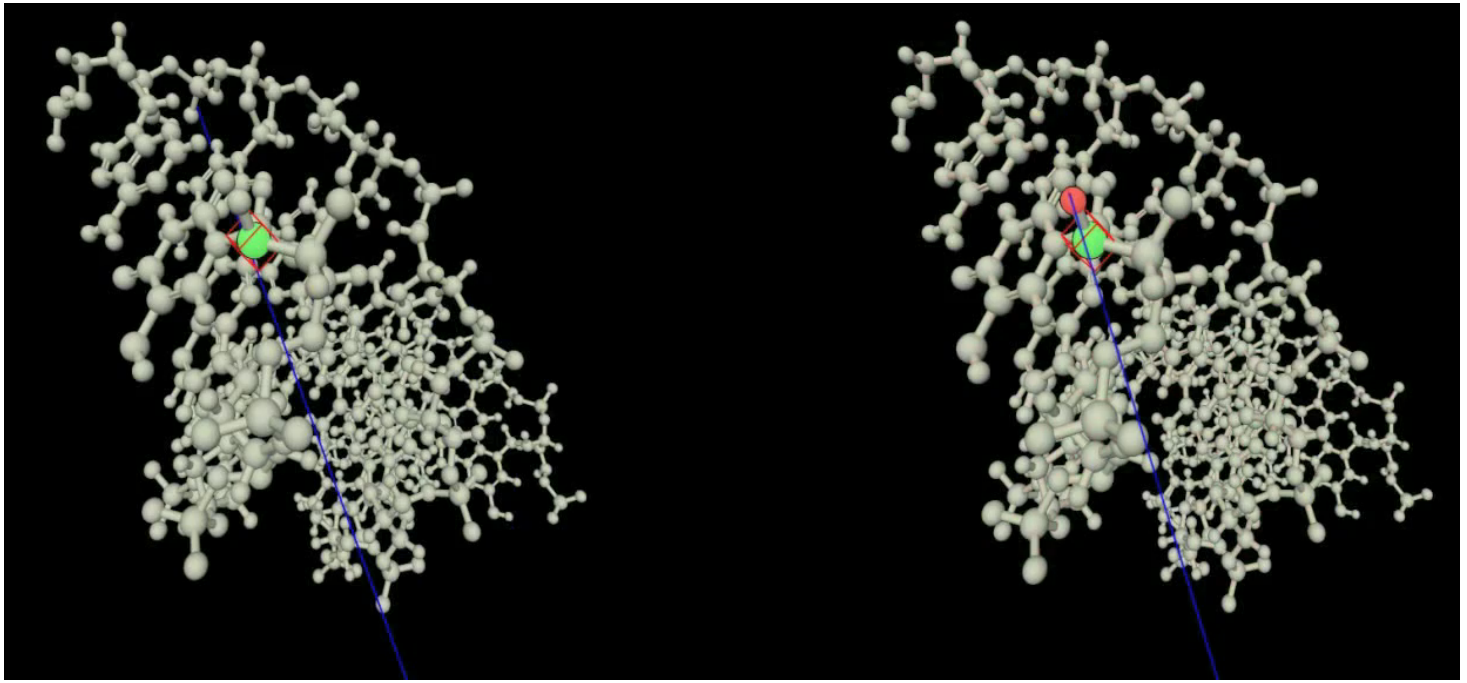
Objekt ist sichtbar,
aber nicht selektierbar

- Die von H aus "sichtbare" Oberfläche ist verschieden von der von E aus sichtbaren Oberfläche →
  - wahre Target-Width ist verschieden von der sichtbaren Target-Width
  - Evtl. kein / ungenügendes Feedback während der Selektion





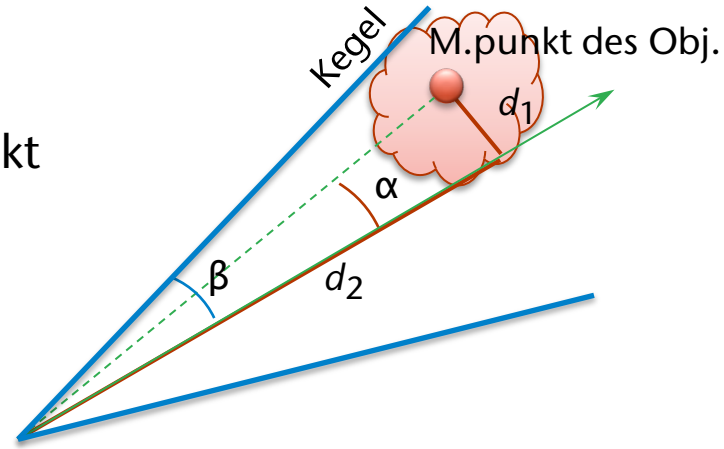
- Vorschlag:
  - Selektionsstrahl von E aus in Richtung h
  - Visuelles Feedback: Strahl von H zum ersten Schnittpunkt
- Experiment der Autoren zeigt: ca. 15% - 20% schneller als einfaches Raycasting



Argelaguet, Andujar, Trueba

IntenSelect: Ranking + Filtering [2005]

- Annahmen:
  - Kegel ist besser als Strahl
  - I.A. sind viele Objekte im Kegel (dense environment)
- Idee:
  - Definiere Skalarfeld im Kegel
  - Berechne daraus "Score" für jedes Objekt
  - Stelle Ranking der Objekte auf



- Eine einfache Score-Funktion:

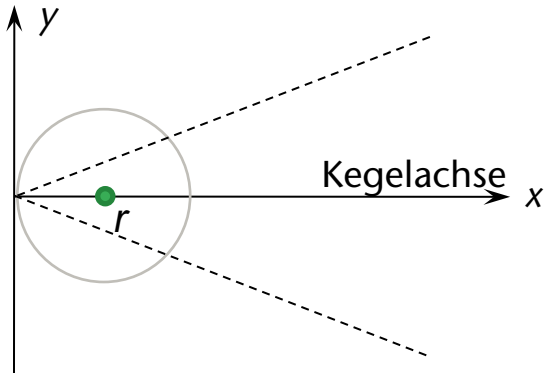
$$s = 1 - \frac{\alpha}{\beta}$$

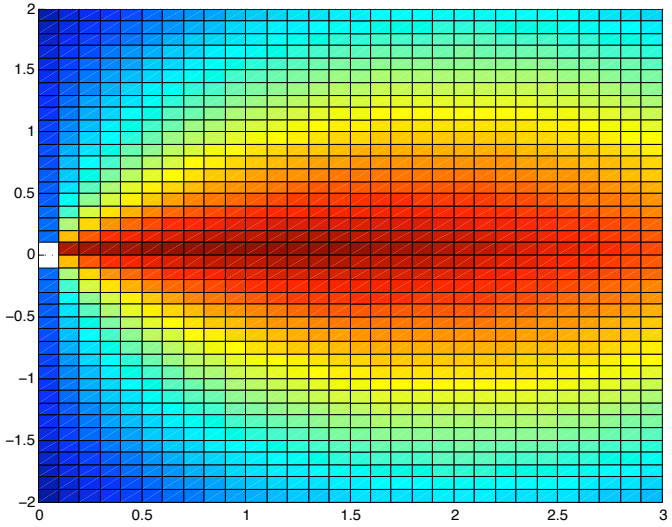
- Score-Funktion, die nahe Objekte (etwas) bevorzugt:

$$s = 1 - \frac{1}{\beta} \tan^{-1}\left(\frac{d_1}{(d_2)^k}\right), \quad k \in [\tfrac{1}{2}, 1]$$

- Noch stärkere Bevorzugung naher Objekte:

$$s = \left(1 - \frac{1}{\beta}\tan^{-1}\left(\frac{d_1}{(d_2)^k}\right)\right) + 0.1\left(1 - \frac{(x-r)^2 + y^2}{r^2}\right)$$





- Problem: Zittern der Hand führt zu häufigen Änderungen des Rankings
- Lösung: Filterung

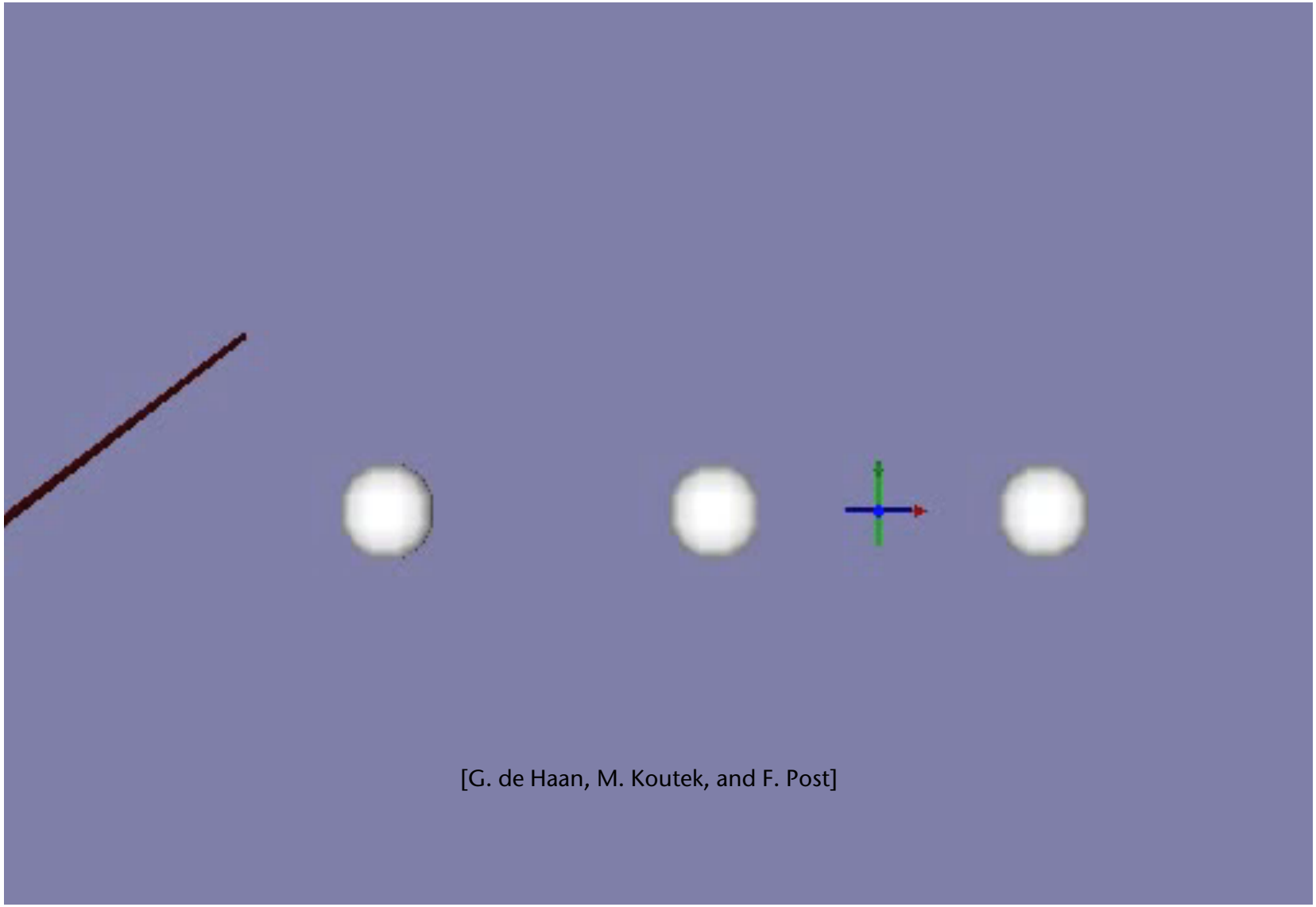
$$s = s(t)$$

$$\hat{s}(t) = \sigma \hat{s}(t-1) + \tau s(t)$$

  - σ = "stickiness", τ = "snappiness"

- Verallgemeinerung: FIR-Filter (siehe Kapitel 7)
- Feedback:
  - Gebogener Strahl zum Objekt mit dem höchsten Ranking
  - Gerader Strahl zur Anzeige der Kegelachse

Video

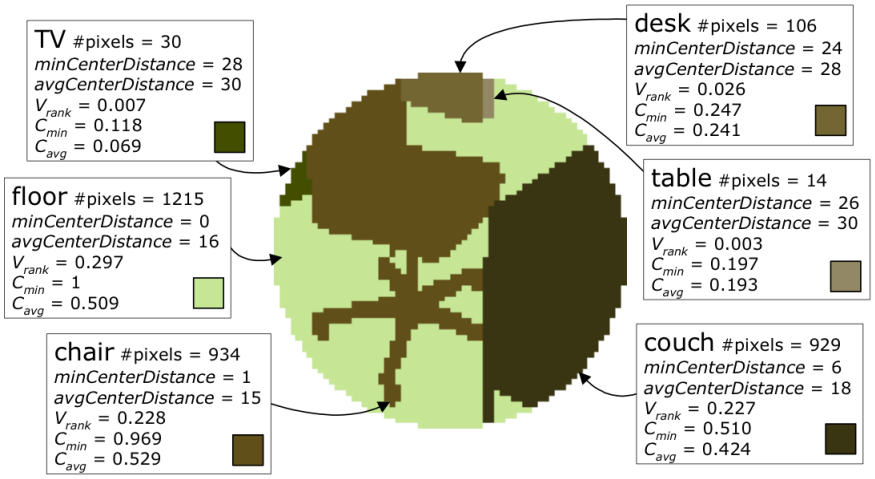


[G. de Haan, M. Koutek, and F. Post]

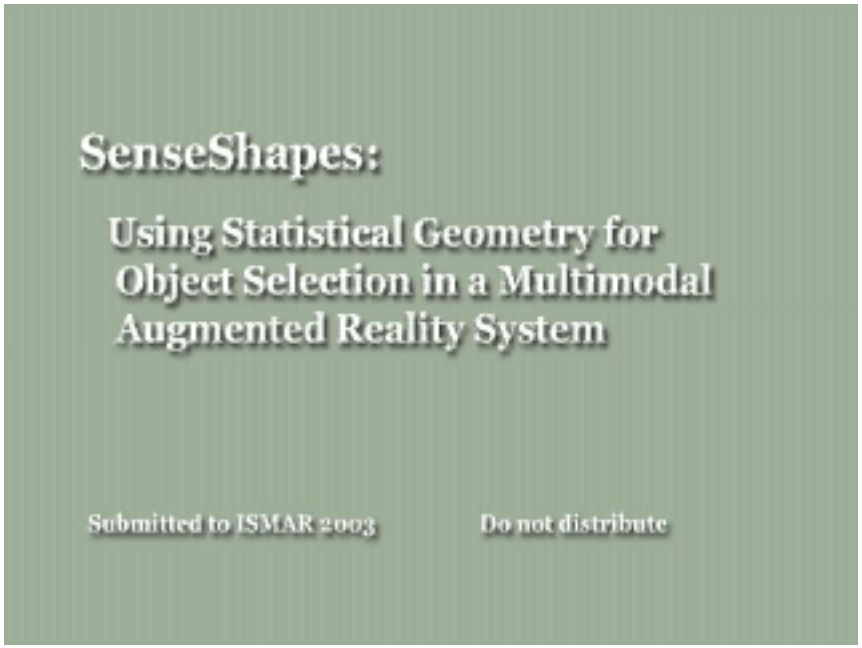
Weitere Ranking-Funktionen

- Umgekehrte Distanz-Funktion: ferne Objekte bevorzugen
- Bessere Berechnung des "Winkels" zur Kegelachse:
  - Rendere ein Objekt mit niedriger Auflösung in einen off-screen Buffer mit "Viewpoint" = Kegelapex, Blickrichtung = Kegelachse
  - Bestimme durchschnittlichen Abstand der Pixel vom Mittelpunkt:

$$s' = 1 - \frac{\frac{1}{n}\sum_{\text{pixel p}} d(p)}{\text{radius}}$$

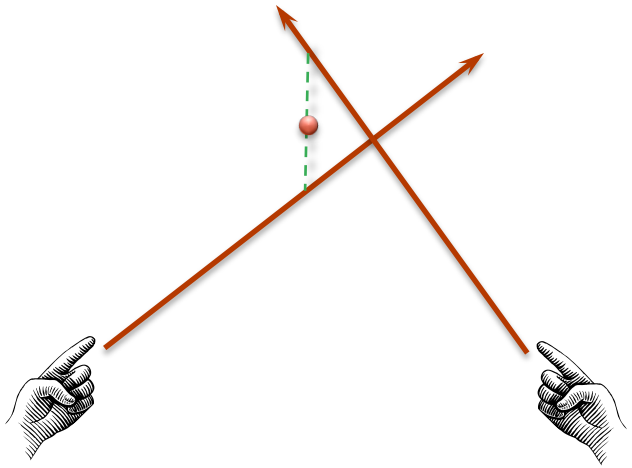


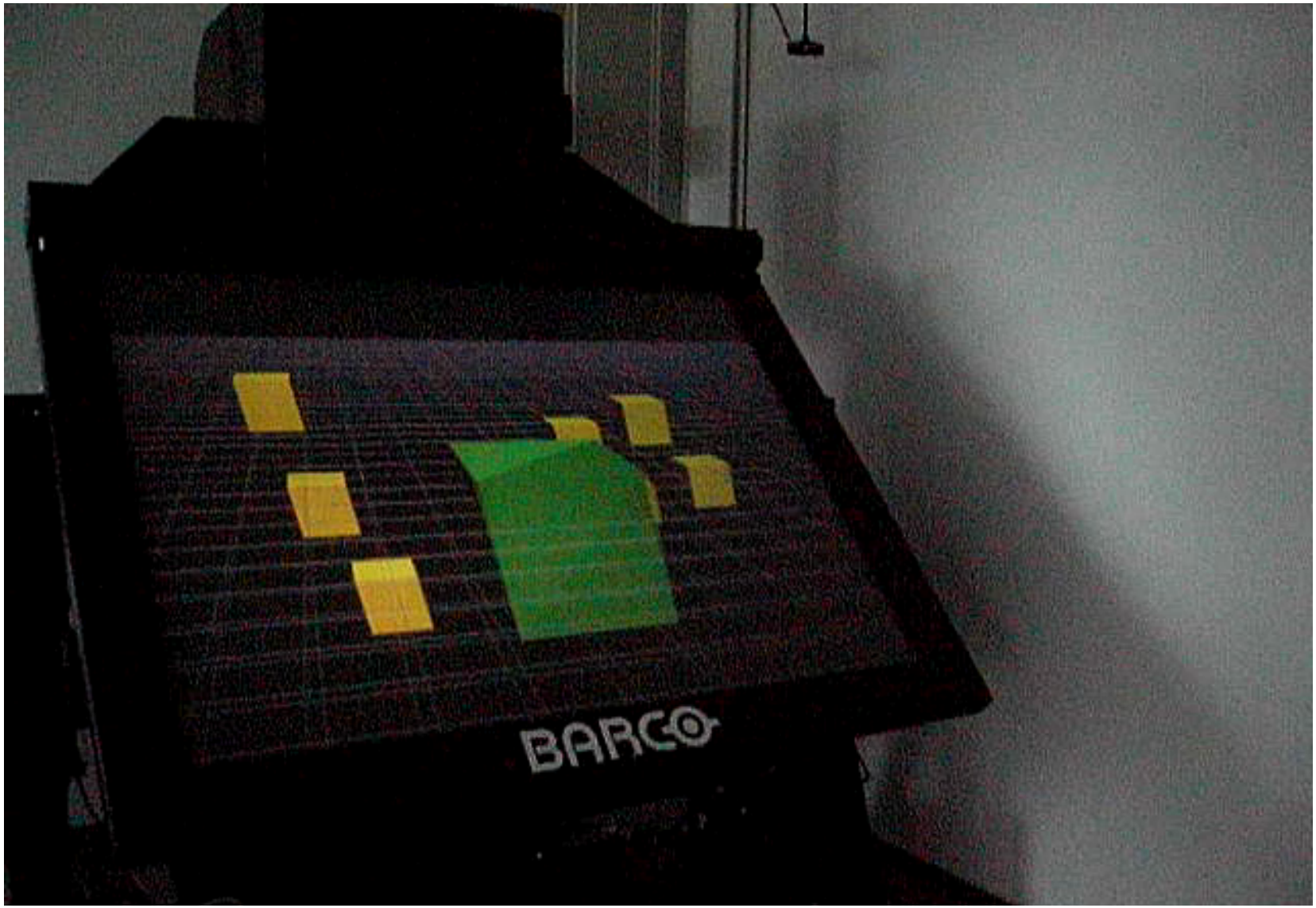
Anwendung in multi-modalem AR-Interface



iSith: beidhändige Selektion [2006]

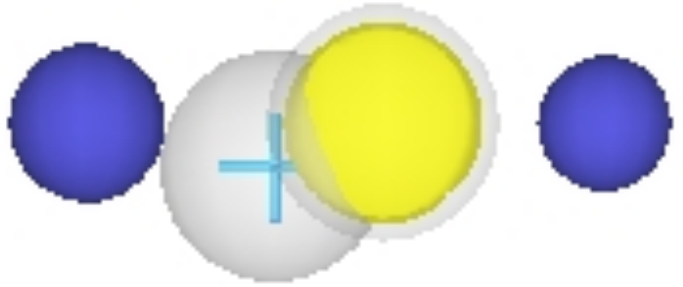
- Idee: Schnittpunkt zweier Strahlen definiert "Selektionszentrum"
- Praktische Umseztung:
  - Zwei Hände = zwei Strahlen
  - Selektionsmodus wird getriggert, wenn Abstand zwischen beiden Strahlen < Threshold
  - Mittelpunkt auf dem Lot zwischen beiden Strahlen berechnen
  - Falls Mittelpunkt "nahe genug" an einem Objekt → selektieren



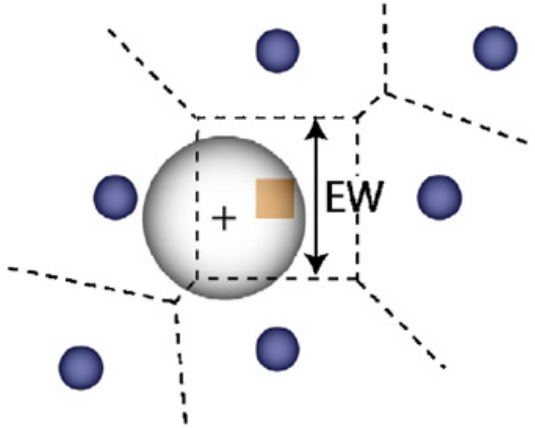

BARCO

Der Bubble Cursor und der Depth Ray [2008]

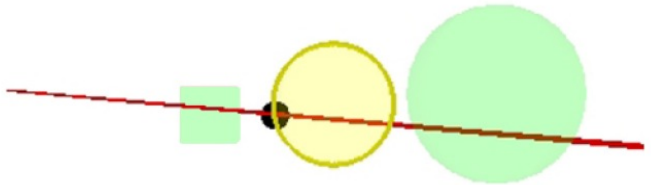
- Eine weitere Methode, die effektive Target-Größe zu erhöhen
- Bubble Cursor:
  - 3D Fadenkreuz
  - Selektiert wird immer das näheste Objekt

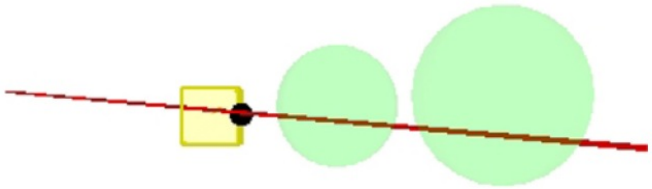


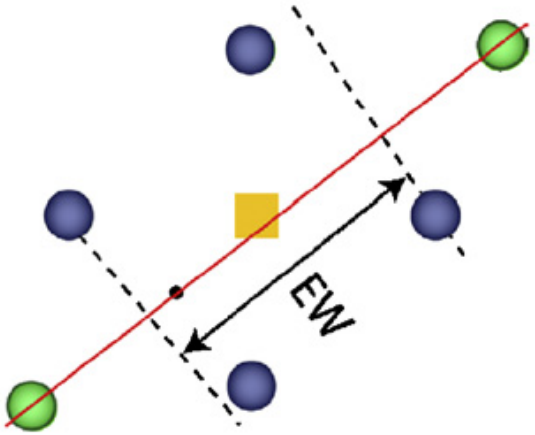
  - Transparente Kugel um Fadenkreuz ist immer so groß, wie Entfernung zum nähesten Objekt (Feedback für "density")
  - Feedback für aktives Objekt: transparente Kugel um selektiertes Obj
- Effektive Target-Größe = Voronoi-Region des Objektes:



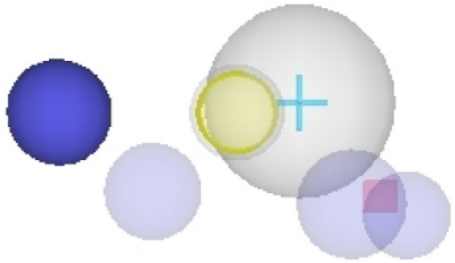
- Depth Ray:
  - Nur Objekte, die von Strahl getroffen werden, werden in Betracht gezogen
  - User kann einen "depth marker" auf dem Strahl verschieben
  - Von den getroffenen Obj.en ist das näheste selektiert / aktiv
- Effektive Target-Größe = Schnitt zwischen Strahl und Voronoi-Region = Segment auf dem Strahl

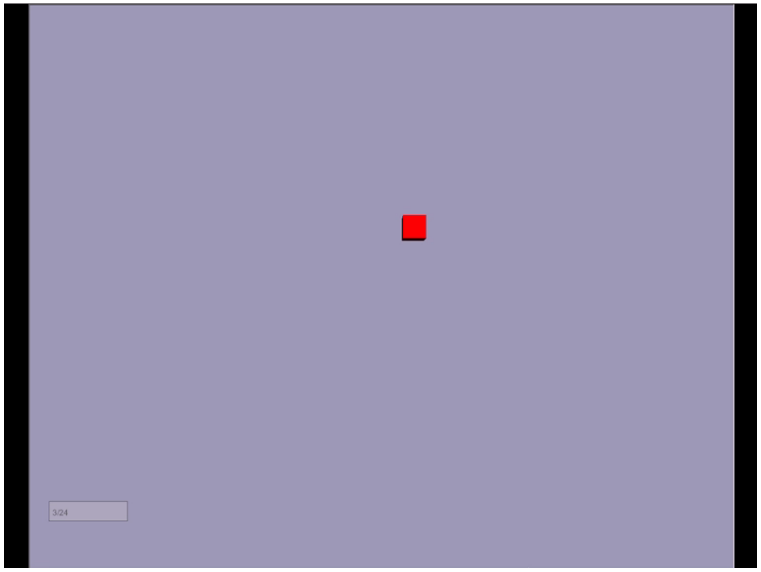




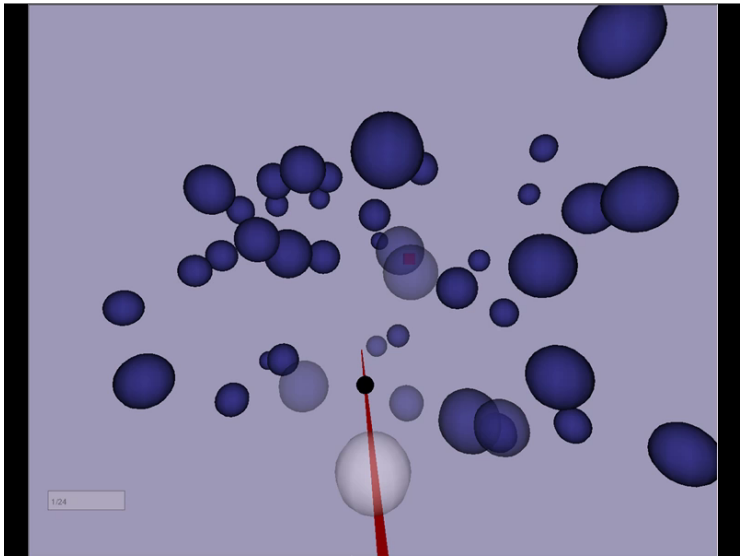


- Occlusion: verdeckende Objekte werden in der Nähe des 3D-Cursors / depth markers transparent (abhängig von der Entfernung zum Curor/Marker)





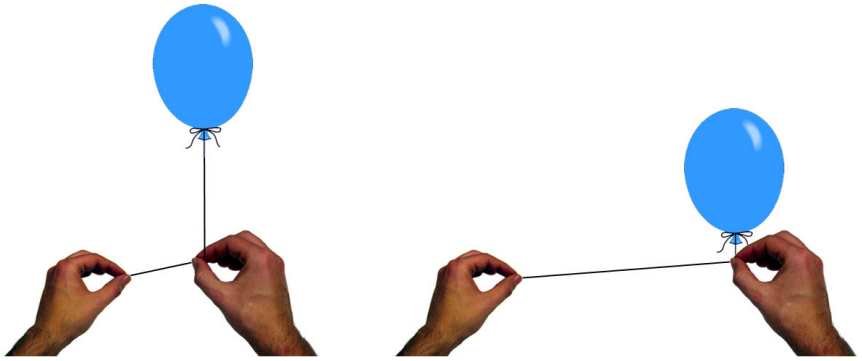
Bubble cursor [Lode van Acken]



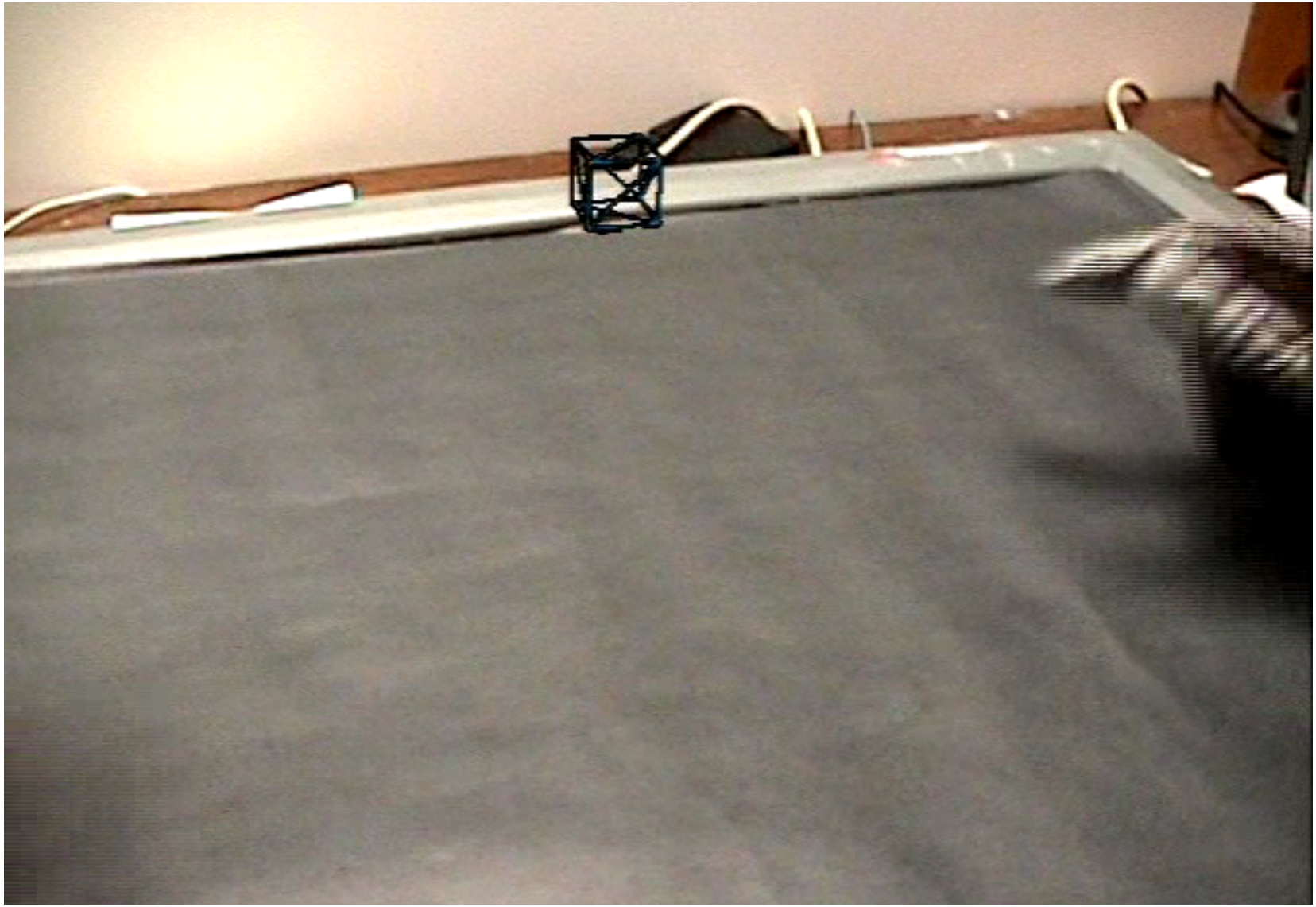
Depth Ray

Balloon Selection [2007]

- Idee: Helium-Ballon steuern
  - Dominante Hand steuert 2D-Position
  - Nicht-dominante Hand steuert 1D-Höhe

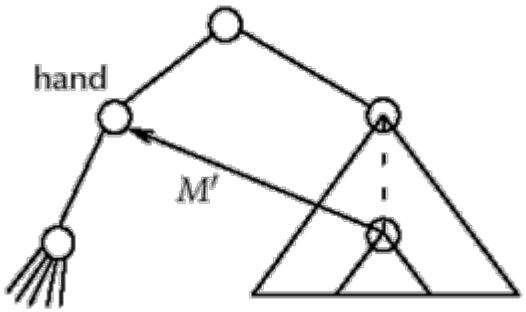


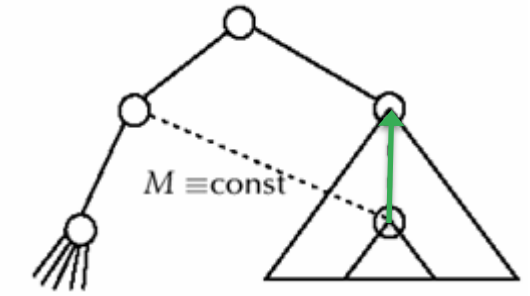
- Implementierung:
  - Zeigefinger geben Position / Höhe an
  - Beide Zeigefinger liegen auf Tisch auf
  - *System control* durch Kontakte im Datenhandschuh oder *Touch Table*
- Vorteil:
  - Dekomposition eines 3D-Tasks in zwei einzelne niedrig-dimensionale Tasks (2D und 1D)
  - Natürlicher Constraint (Tisch)



Objekt-Manipulation

- Zweithäufigste Interaktionsaufgabe
- Einfaches, direktes Greifen (nicht realistisch):
  1. Objekt selektieren
  2. Greifen triggern (Geste, Sprachkommando)
  3. Evtl. Kollision zwischen Hand und Objekt abwarten
  4. Objekt an die Hand "kleben"
  5. Trigger zum Loslassen abwarten
- Wie macht man "ankleben"?
  - Umhängen im Baum, oder
  - Transformationsinvariante erhalten
  - Meine Erfahrung: bei nicht-trivialen Anwendungen macht Umhängen Ärger!
- Natürliche Interaktion: Diplomarbeit! ☺





Taxonomie der natürlichen Greifposen

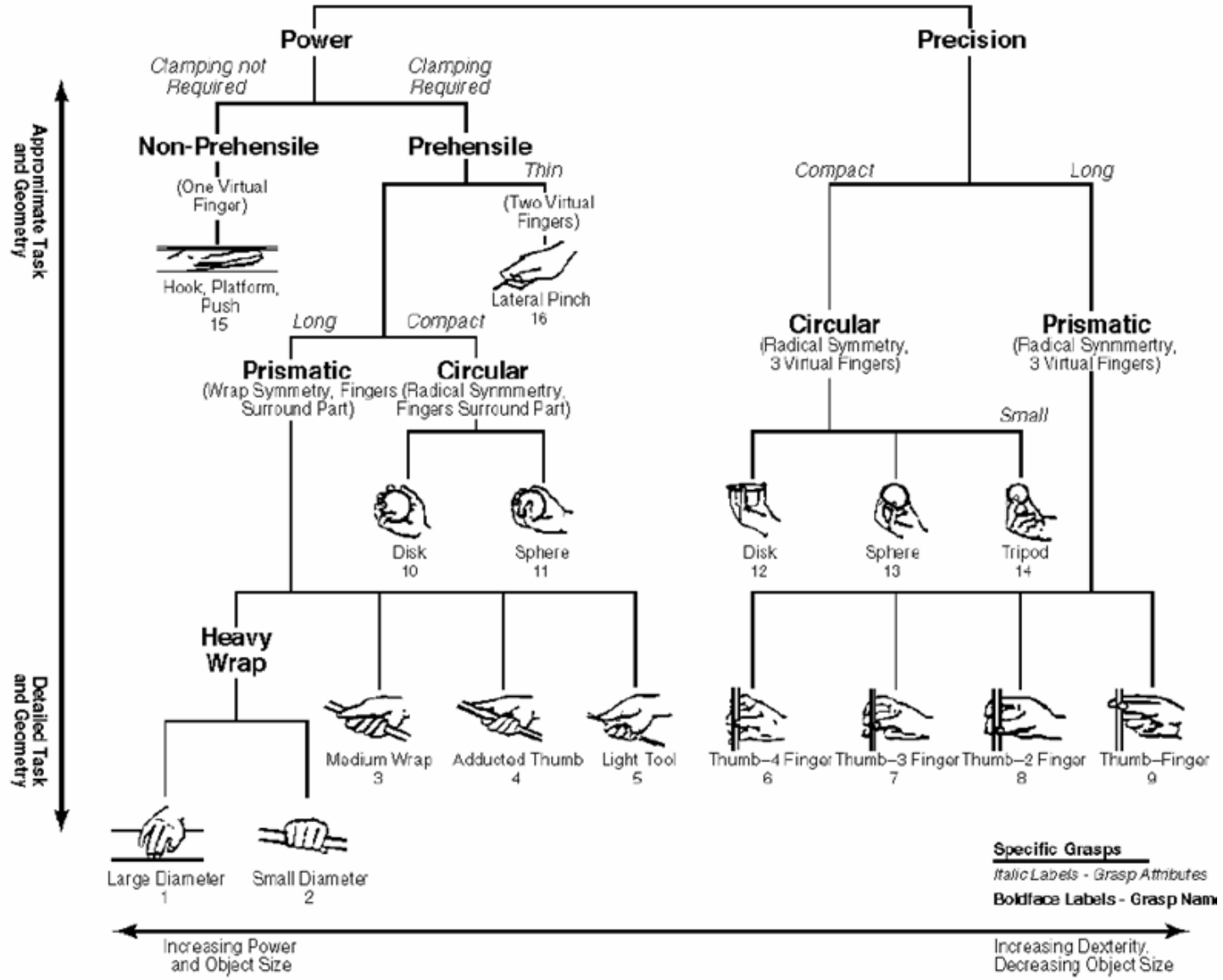


Figure 64. Taxonomy of Manufacturing Grasps

Action-at-a-Distance

- Allgemeines Interaktionsprinzip
- Beispiel: Verschieben eines Objektes aus der Entfernung:
  - Idee: skaliere Handbewegung so, daß sich die relative Position zwischen Hand und Objekt auf der Bildebene nicht ändert
  - Berechnung:
    1. d_O^t = Distanz des Obj. an alter Position
    2. P^t = Punkt auf Strahl S_t mit Distanz d_O^t
    3. d_H^t = Distanz zu Hand an alter Position
    4. d_H^{t+1} = Distanz zu Hand an neuer Position
    5. Bestimme d_O^{t+1} so, daß $\frac{d_O^t}{d_H^t} = \frac{d_O^{t+1}}{d_H^{t+1}}$
    6. P^{t+1} = Punkt auf Strahl S_{t+1} mit Distanz d_O^{t+1}
    7. Translation für das Objekt = $P^{t+1} - P^t$

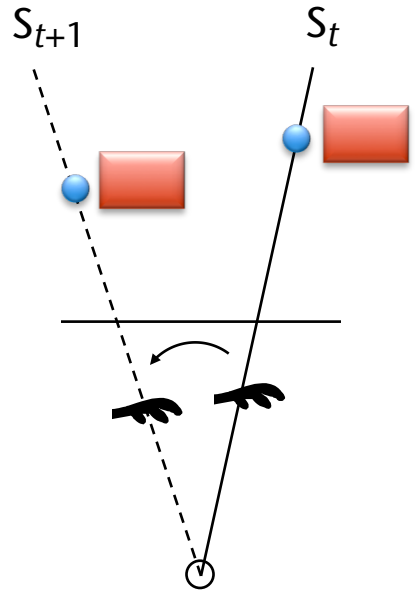
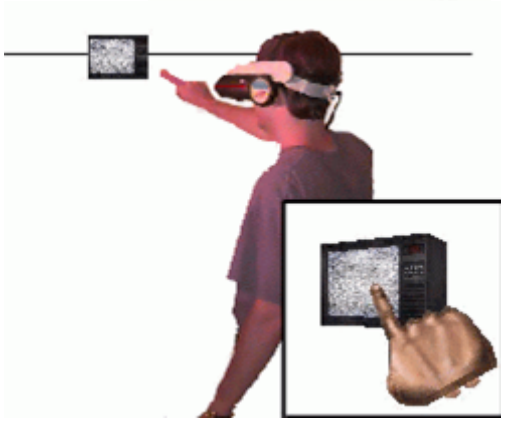
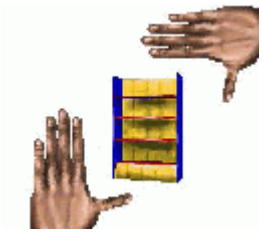


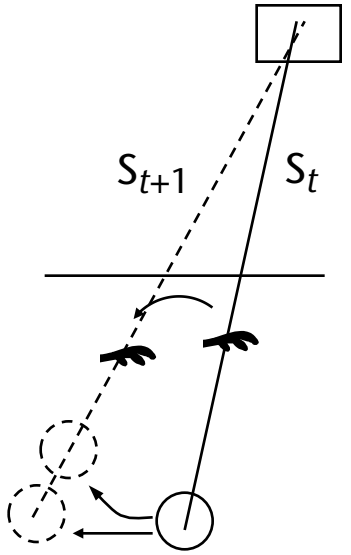
Image plane interaction

- User interagiert nicht mit 3D Objekten sondern deren 2D-Abbild
- Selektion:
  - schieße Strahl zwischen Daumen und Zeigefinger
  - Strahl von Auge durch Zeigefingerspitze
  - "*Lifting palm*"
  - Einrahmung mit den Händen
- Manipulation:
  - Bringe Objekt in Reichweite des Users
  - Transliere User zum Objekt
  - Evtl. Szene skalieren





- Navigation:
  - Projektionen von Finger und Objekt auf der Bildebene bleiben konstant zueinander
  - Translation oder Orbit
  - Ähnlich wie "Verschieben aus der Entfernung"
  - Distanz Hand–Auge → Zoom
- Problem: Stereo
  - Lösung: während der Navigation/Selektion/... Mono rendern.



PRISM

- PRISM = "precise and rapid interaction through scaled manipulation"
- Problemstellung: präzise Manipulation im Nahbereich
- Die Idee:
  - Skaliere die Handbewegung, d.h., C-D ratio > 1
  - Immer dann, wenn Handgeschwindigkeit < Schwellwert
  - Sei

    D_O = Distanz, um die das manipulierte Objekt transliert wird

    D_H = Distanz, die sich die Hand seit dem letzten Frame bewegt hat

    V_H = durchschnittl. Geschwindigkeit der Hand in der letzten ½ Sekunde

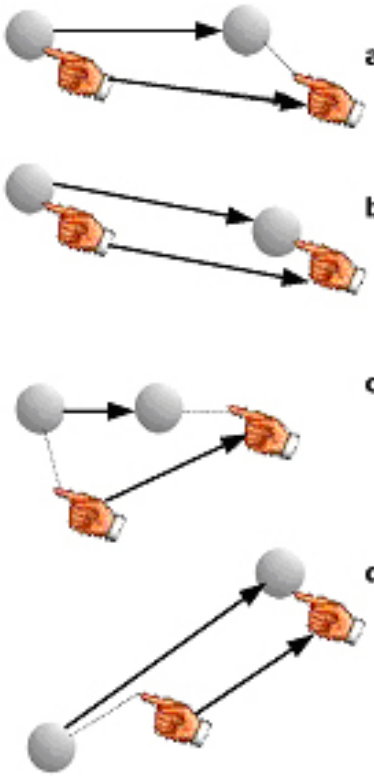
    S = Schwellwert;

    dann wählen wir

$$D_O = k \cdot D_H \qquad k = \begin{cases} 1 & , V_H > S \\ V_H / S & , \min < V_H < S \\ 0 & , V_H \leq \min \end{cases}$$

- Detail:
  - Diese Skalierung kann man unabhängig für jede Koordinate machen
  - Vorteil: man kann so Objekte leicht exakt entlang einer Koordinatenachse bewegen
- Offset-Recovery:
  - Problem: die Hand- und Obj-Position laufen mit der Zeit auseinander
  - "Lösung": wenn sich die Hand sehr schnell bewegt, wird dieser Offset vom System reduziert (das Objekt bewegt sich schneller als die Hand)
  - Merkt der User bei schneller Handbewegung nicht
- Diese Technik geht (fast) genauso analog für Rotationen:
  - Man muss nur die Handrotation in Achse + Winkel konvertieren, dann skalieren, dann wieder in Rotationsmatrix konvertieren

Beispiele

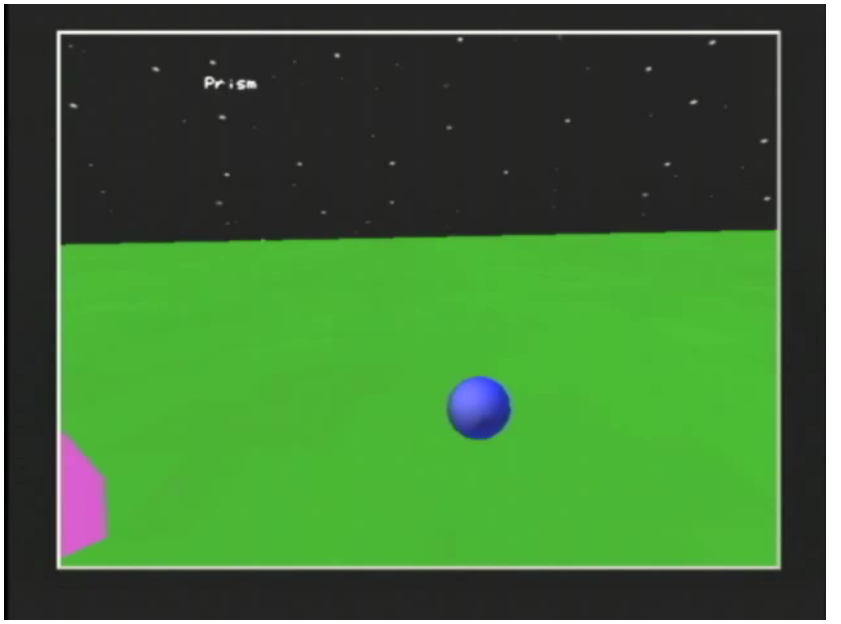


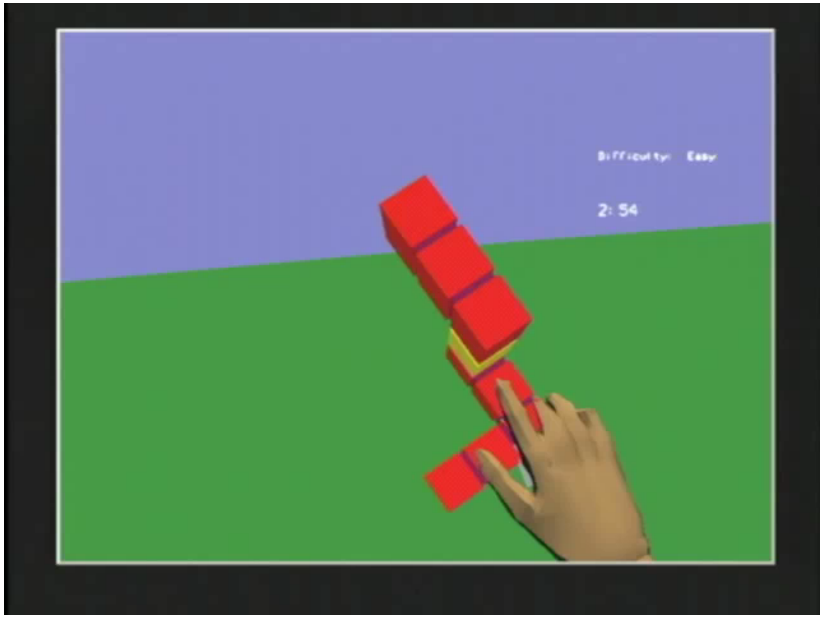
a. User moves hand slowly to the right and down. Some movement is scaled down in the horizontal direction, nearly all is scaled down in the vertical.

b. User moves in the same direction as in (a), this time quickly. Since interaction is in direct mode, object follows hand to its exact location.

c. User slowly moves to the right and up back towards object, vertical offset is recovered. Scaled motion performed in horizontal direction.

d. After accumulating offset, user moves hand quickly up and to the right. Offset is eliminated and mode has switched into direct.

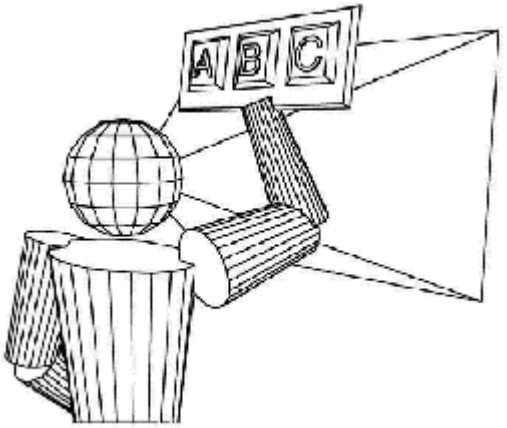


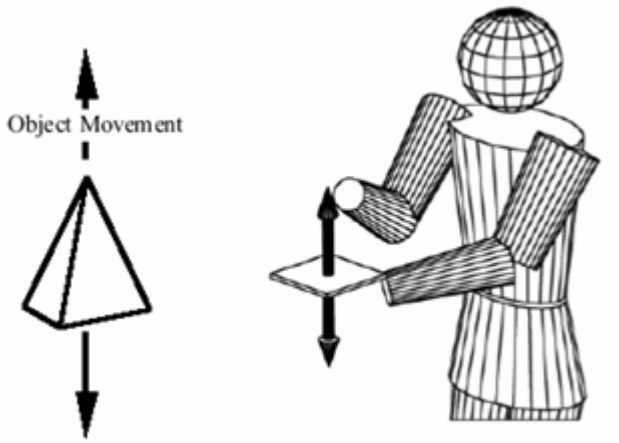


Frees, Kessler, Kay (http://give.ramapo.edu/prism/prism.html)

Proprioceptive Interaction

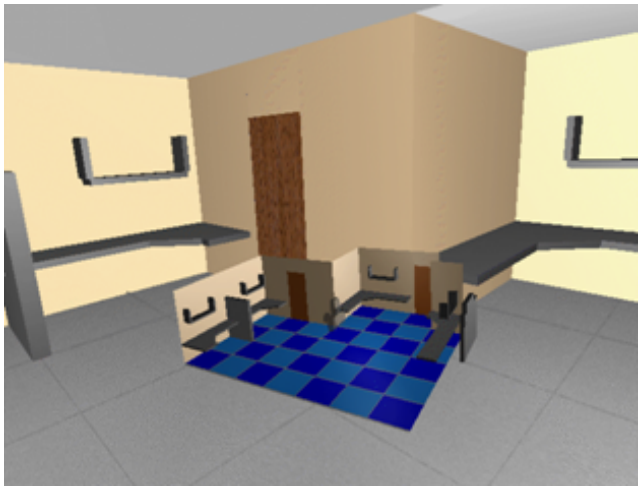
- *proprius* = (adj.) eigen
- Idee: nutze aus, dass Mensch auch mit geschlossenen Augen weiß, wo sich seine Hand befindet
- Echte "*pull-down*" Menüs:
  - User greift nach oben, Herunterziehen bewirkt Erscheinen von Menü
- Löschen = über die Schulter werfen
- Objekt werden aus der Ferne durch hand-held Widgets manipuliert



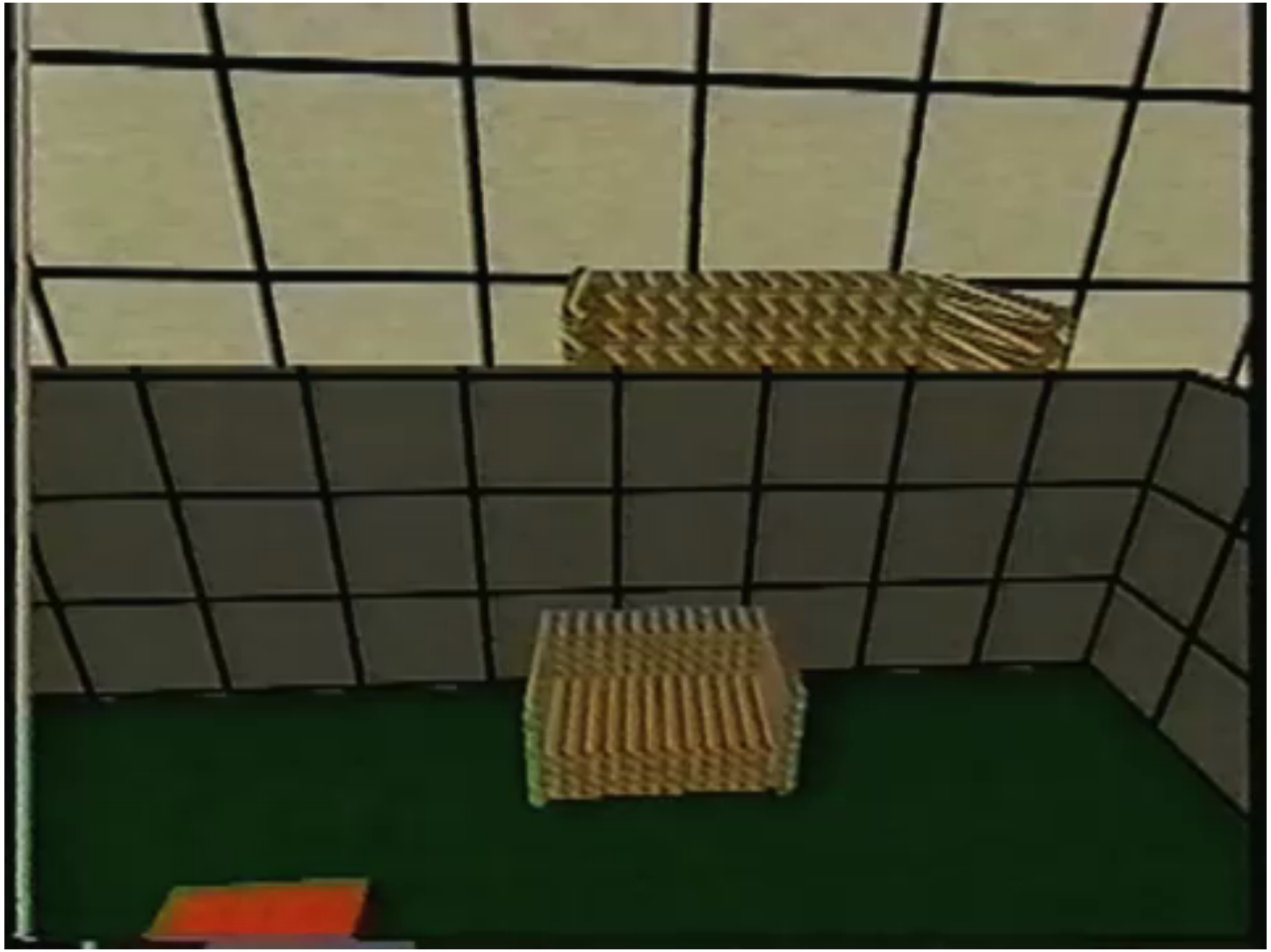


World-in-Miniature

- Idee: 3D-Miniatur-"Karte" (analog zu 2D-Karte)
- Interaktion in der 3D-Karte wird übertragen in Aktion in der virtuellen Umgebung
- Überblick gewinnen = WIM drehen
- Objekt bewegen = Objekt in WIM greifen und bewegen
- Navigation = Frustum in WIM verschieben, oder Punkt selektieren



Video



Doug Bowman

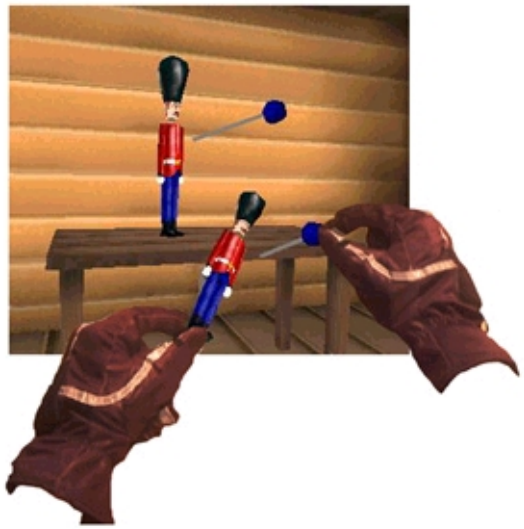
Zweihändige Interaktion

- Der Mensch hat 2 Hände:
  eine dominante (rechte) und eine nicht-dominante (linke)
- Funktion der nicht-dominanten Hand:
  Referenzkoordinatensystem, Kontext positionieren
- Funktion der dominanten Hand:
  Feinmotorik innerhalb des Kontexts

Voodoo dolls

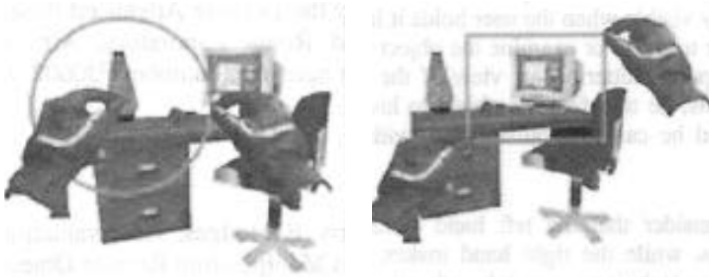
[1999]

- Technik zur "*remote*" Manipulation / Plazierung von Objekten
- Idee: Erzeuge eine temporäre Kopie (= voodoo doll) der entfernten Objekte
- Ablauf einer Manipulation:
  - Erzeuge Kopie eines Referenzobjektes, mache diese an linker Hand fest
    - Das Original der Kopie bleibt fest; Kopie liefert Referenzkoordinatensystem
  - Erzeuge Kopie des zu manipulierenden Objektes, attach to right hand
    - Das Original dazu wird bewegt
  - Bewegung der rechten Kopie — relativ zur Kopie des Referenzobjektes(!) — wird auf das Original übertragen



- Erzeugen einer Kopie (voodoo doll):
  - Image-plane-Technik: Pinch-Geste "vor" dem Objekt
  - Größe der Kopie = ½ Meter in der längsten Ausdehnung
- Kontext zur linken *voodoo doll*:
  - Erzeuge Kopien von Objekten in der Umgebung des Referenzobjektes
  - Wieder Image-plane-Techniken, hier "Framing" (mit Rechteck oder Kreis)

pinch copy attached to hand



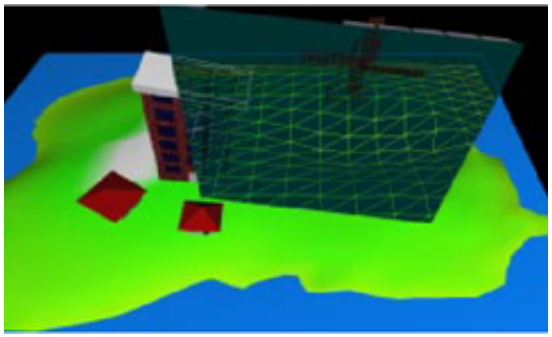
framing by circle framing by rectangle

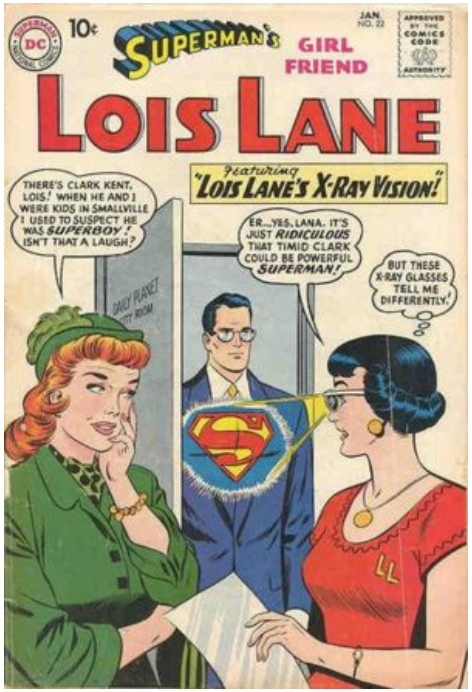
- Beispiel einer Manipulation:
  1. User "greift" Tisch mittels Pinch-Geste der linken Hand
     - Der Tisch kann sich in weiter Entfernung befinden
  2. System erzeugt Kopie, attacht an die linke Hand, plus Kontext-Objekte, hier z.B. Telefon und Monitor
  3. User "greift" Kopie des Telefons aus dem Kontext (wieder Pinch)
  4. System erzeugt Kopie des Telefons an der rechten Hand
  5. User plaziert Telefon an anderer Stelle auf der Kopie des Tisches
  6. System überträgt die Translation auf das originale Telefon

- Vorteile:
  - Linke Hand wird genau für den von Natur aus vorgesehenen Zweck eingesetzt
  - User kann auf beliebigen Skalen arbeiten, ohne explizit eine Skalierung vorgeben zu müssen
    - Skalierung geschieht implizit durch Selektion des Referenzobjektes
  - Schwer zugängliche / selektierbare Objekte werden leicht zugänglich

Magic Lenses

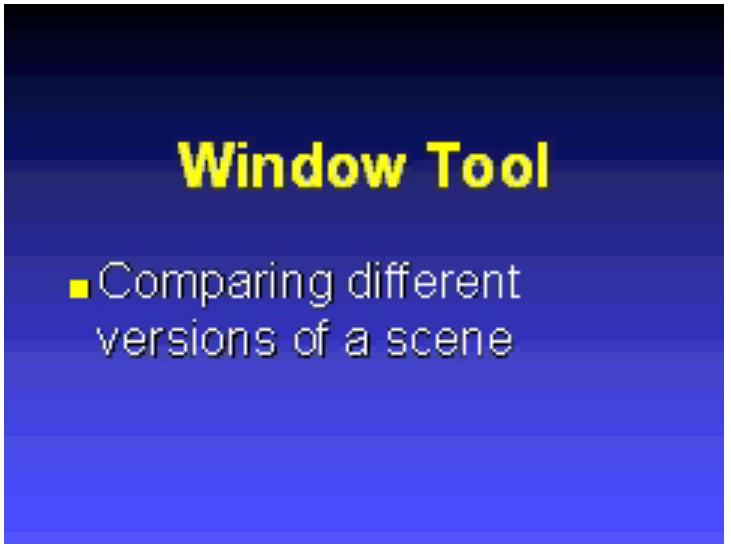
- Idee: durch eine Linse sieht man die virtuelle Umgebung "anders"
- Anders =
  - Andere Rendering-Parameter
  - Andere Geometrie
  - Anderer Viewpoint
  - Skalierung, ...
- Beispiele:
  - Wireframe
  - Vergrößerung
  - *Preview-Window* für *eyeball-in-hand* oder *scene-in-hand* Navigation
  - X-Ray
  - Zusätzliche, alternative Viewpoints
  - Alternative Geometrie
- Magic lenses können auch durch ein Volumen definiert werden





Videos

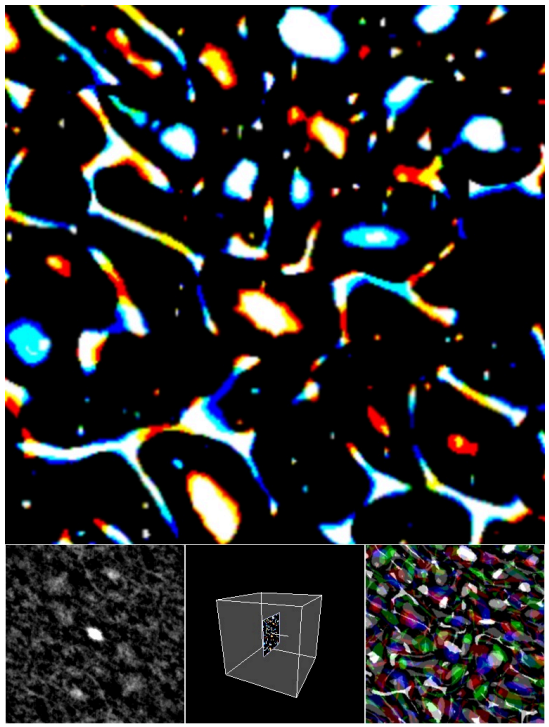

X-Ray Tool
Laying underground cables

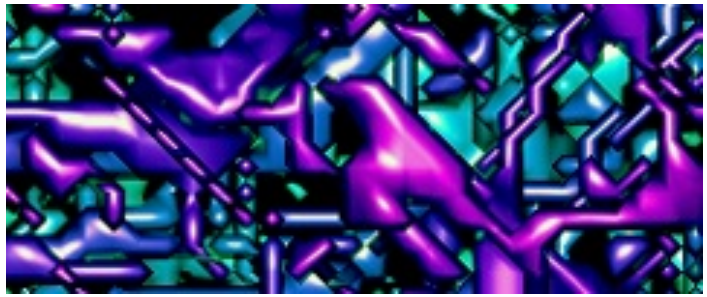

Window Tool
Comparing different versions of a scene


Window Tool
Multiple Views into scene

Eine Anwendung in Scientific Visualization

- Aufgabe:
  - Visualisierung von Volumen-Daten (hier: CT) auf iPad
  - Intuitive Navigation (= Bewegen des Viewpoints)
- Lösung: betrachte das iPad als "Magic Lens" in die virtuelle Umgebung



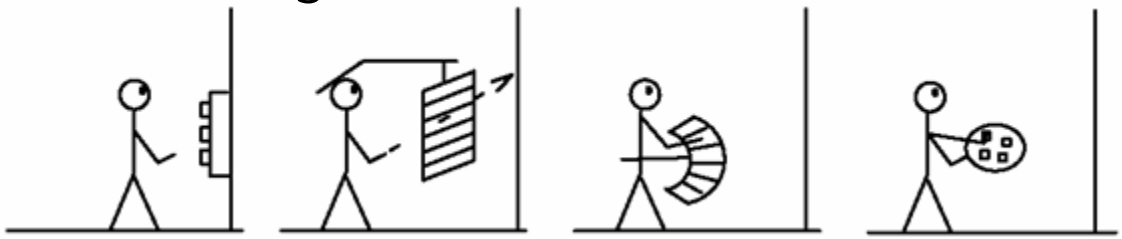


System Control

- Die "dritte" große Kategorie von Interakionsaufgaben in VR
  - Das ungeliebte Kind in der VR-Interaktions-Community ☺
  - Interaktionsaufgaben: Systemzustand ändern
    - Und alles andere, das man nicht recht einordnen kann ☺
- Eine Taxonomie ist hier kaum machbar
- Die typischen Techniken:
  - Menus
  - Spracherkennung
  - Gesten(-menus)
  - Physikalische Geräte

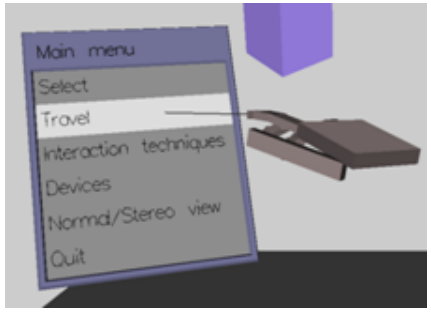
Menüs

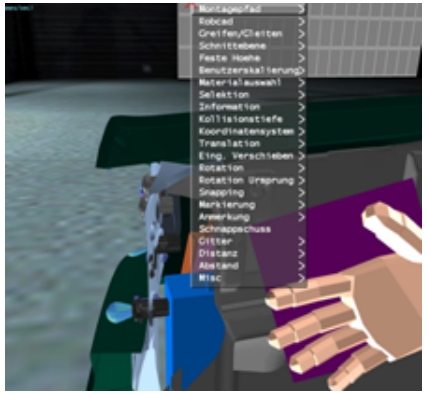
- 3 Stufen
  1. Menü aufklappen
  2. Navigieren durch Menü
  3. Item selektieren
- Taxonomie:
  - Eingabemöglichkeiten: Gesten, Sprache, Buttons, ...
  - Positionierung
  - Selektion
  - Dimension des Menus
- Beispiel Positionierung des Menus:



Beispiele

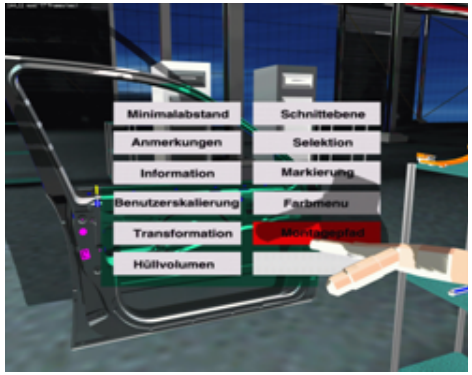
- 3D, Strahl aus Zeigefinger/Handrücken

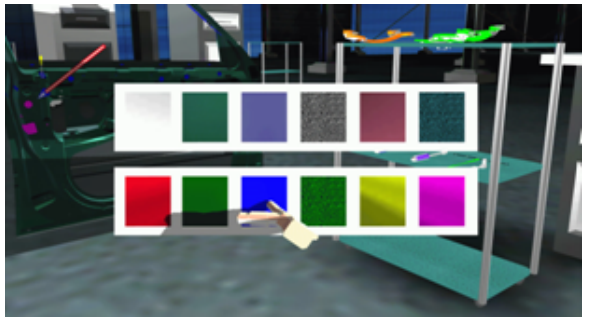




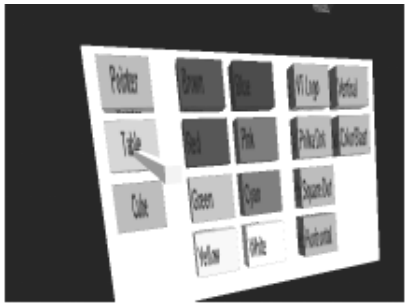
- 2D overlay, relative Handbewegung → Cursorbewegung

- 3D,
  fest oder
  Kopf-zentriert,
  Strahl von Auge
  durch Zeigefinger



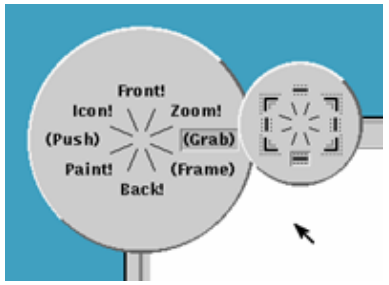


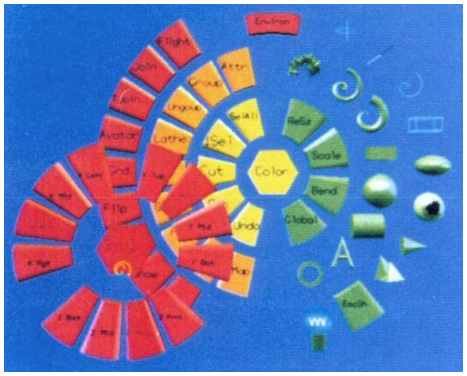
- Hand-held

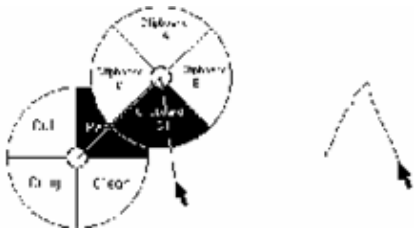


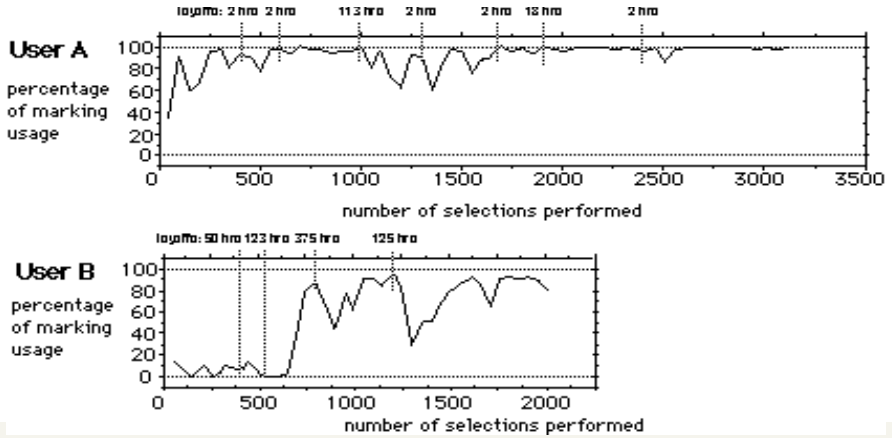
"Marking Menus" (a.k.a pie menu)

- Ordne Items um ein Zentrum herum an (Kreis, Würfel, ..)
- Beim Auslösen: ordne Menü um aktuelle Zeigerposition herum an
- Experten können Menü "blind" bedienen
- Übergang von "*novice mode*" zu "*expert mode*" ist kontinuierlich
- Marks (Mausgesten) sind wesentlich effizienter als Menüs:

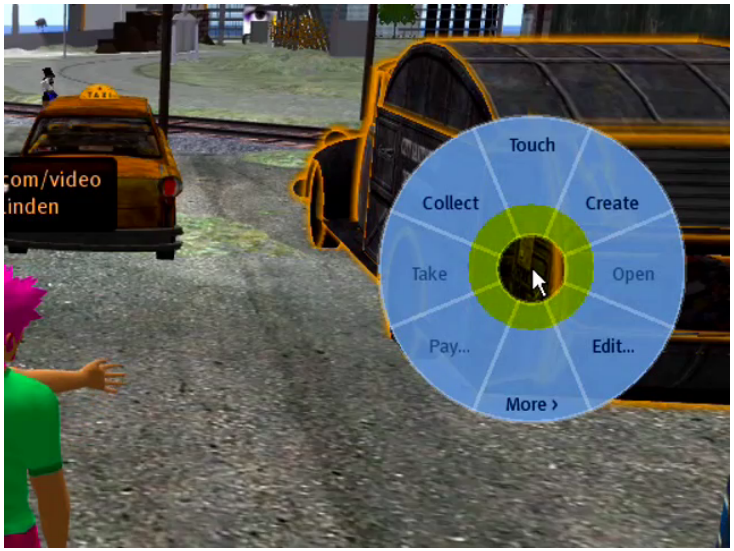






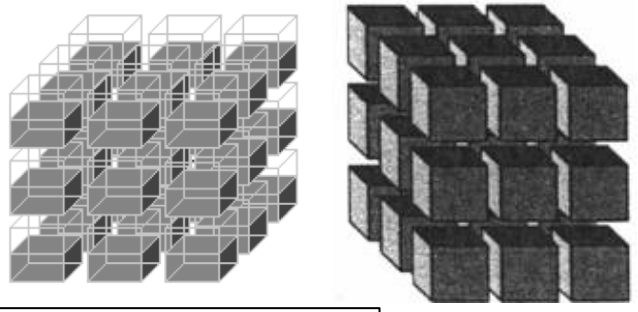


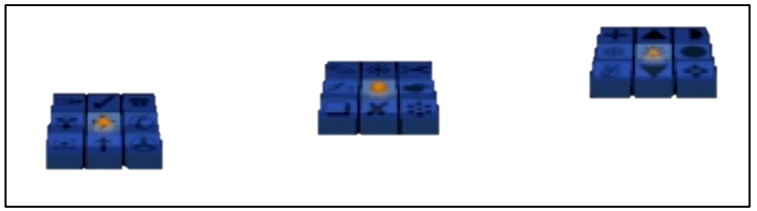
- Video (2D):



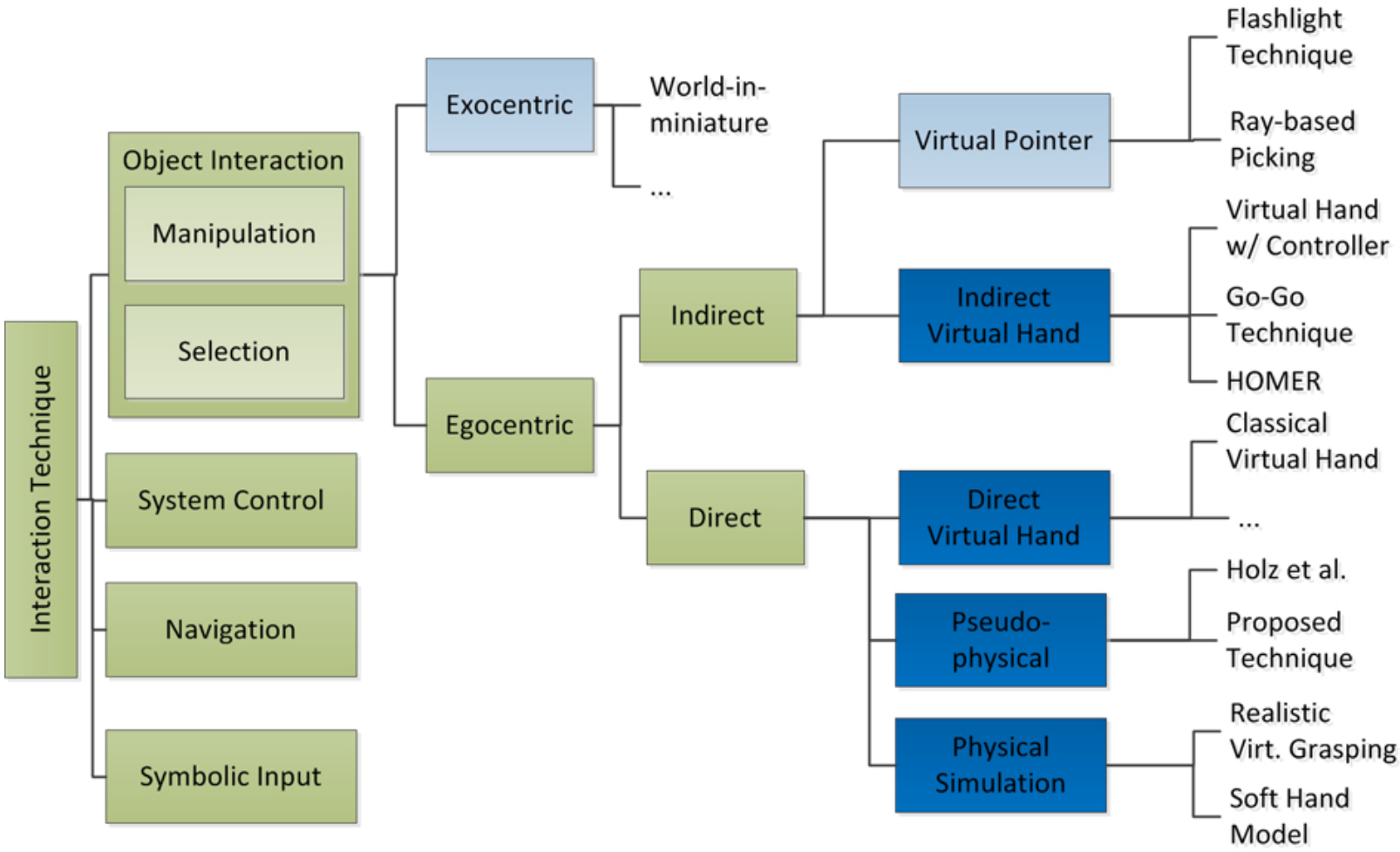
[SecondLife]

- In 3D?
  - Direkte Übertragung → "control cube"
  - Mäßig erfolgreich
  - Wie geht es besser ?



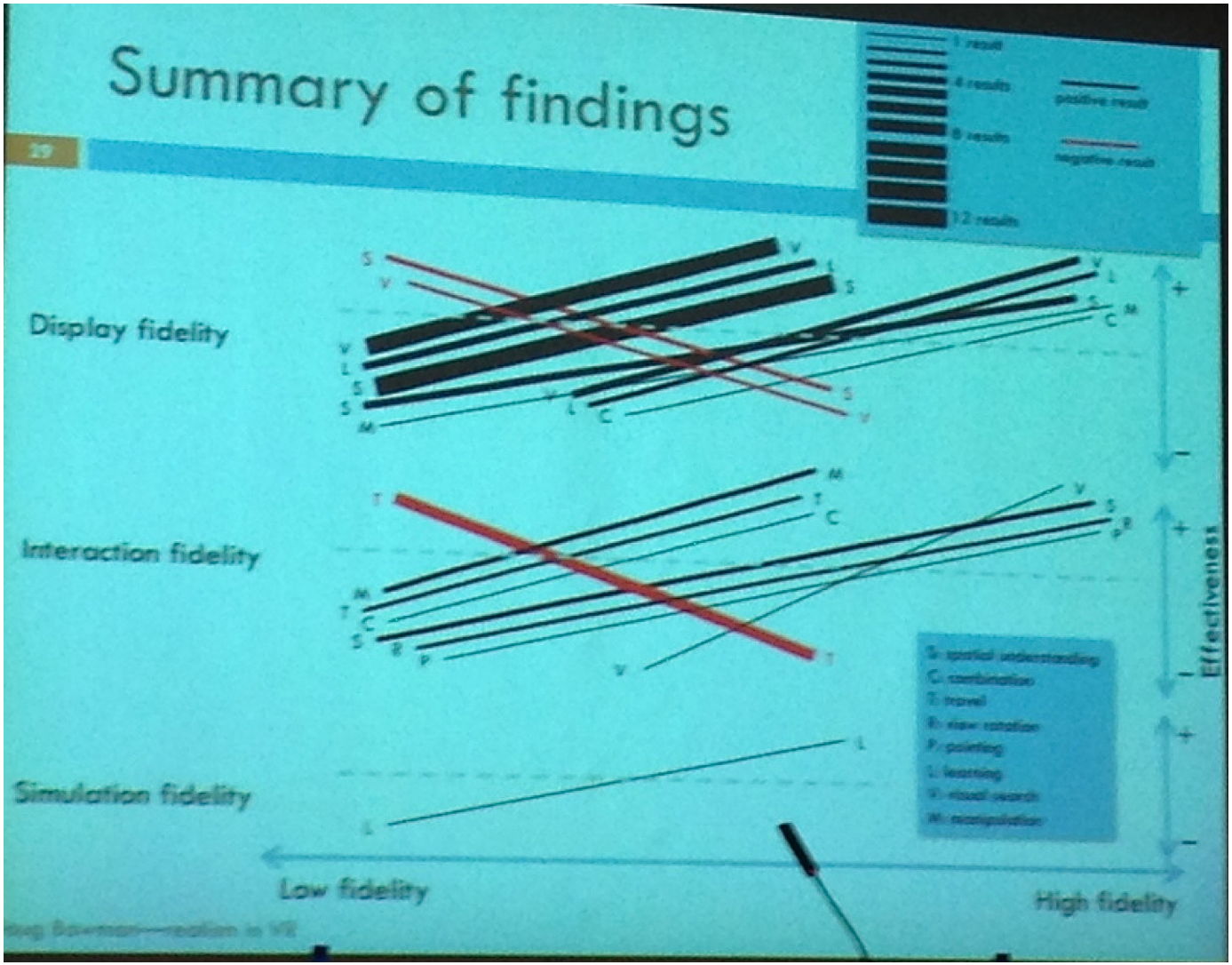


Eine Taxonomie



Mathias Moehring: Realistic Interaction with Virtual Objects Within Arm's Reach (Diss.; aufbauend auf Bowman's Taxonomie)

Is more fidelity always better?



Doug Bowman, JVRC'12

- Higher (interaction) fidelity often results in higher effectiveness
- Increasing fidelity does not always improve user effectiveness within a virtual environment (it does not decrease it either)
- Very few cases where higher fidelity is detrimental
  - Travel techniques are one strong case for *less* fidelity
- Best cases for high fidelity:
  - Difficult and complex visuospatial tasks
  - Learning / training
  - High-DOF interaction tasks

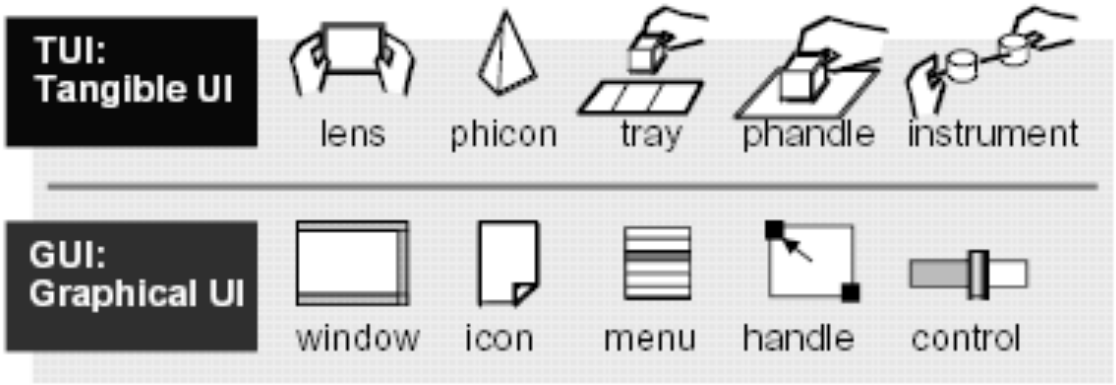
Tangible User-Interfaces

- Idee: Instanziiere virtuelle/abstrake Interaktionsmetaphern (Handles, Icons, Sliders, ...) wieder physikalisch

- **Defintion: Tangible User Interface (TUI):**
  An attempt to give physical form to digital information, making bits directly manipulable and perceptible by people.

  Tangible Interfaces will make bits accessible through
  - augmented physical surfaces (e.g. walls, desktops, ceilings, windows),
  - graspable objects (e.g. building blocks, models, instruments), and
  - ambient media (e.g. light, sound, airflow, water-flow, kinetic sculpture).

- Analogien zwischen GUIs und TUIs:



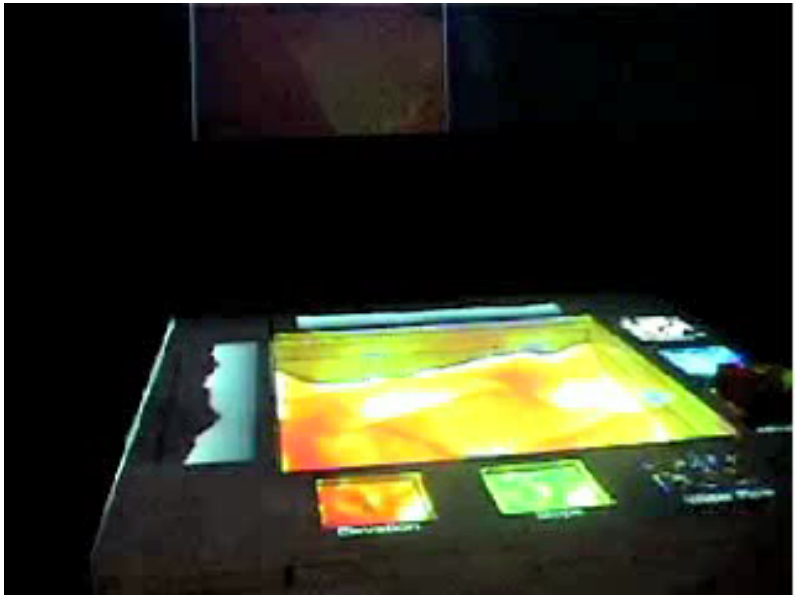
- Beispiele:

Tangible Magic Lens

Tangible Slider

Beispiele

Sandscape
(Sand als Terrain)



http://tangible.media.mit.edu

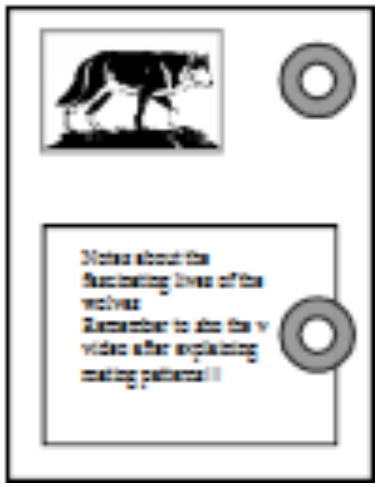
GranulatSynthese
(interaktive Installation)

http://imve.informatik.uni-hamburg.de/projects/GranulatSynthese

IP Network Design Workbench (Pucks zur Manipulation von Knoten und Kanten)

http://tangible.media.mit.edu/projects/ipnet_workbench

Palette & PaperButtons




CG
VR