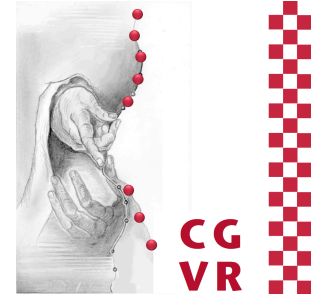
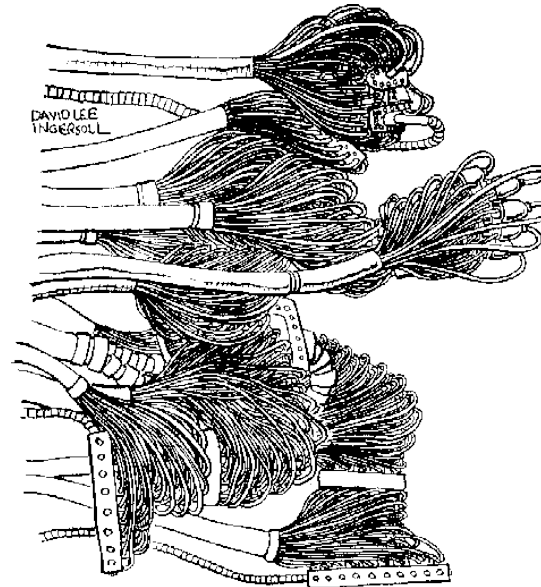


Bremen



Virtuelle Realität Eingabegeräte



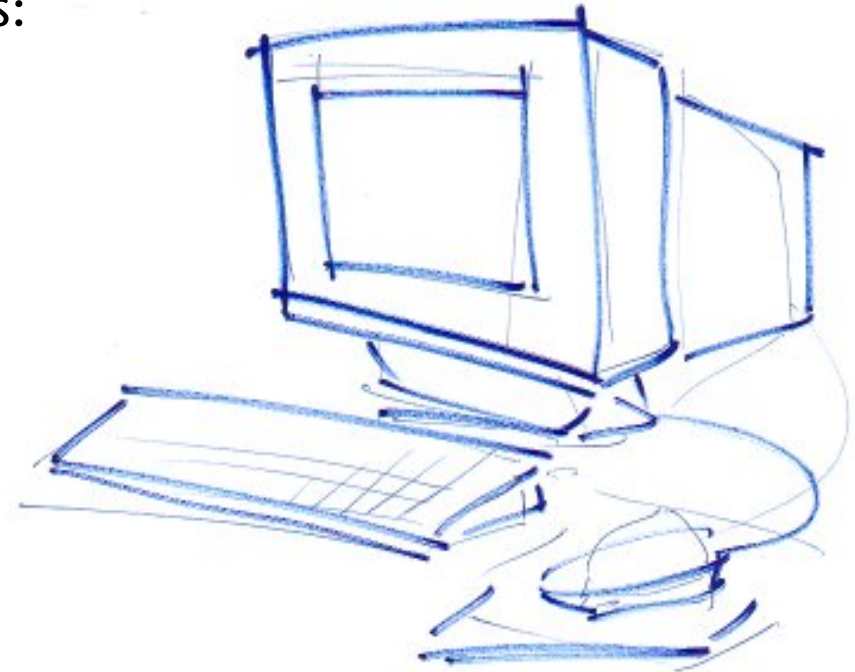
G. Zachmann

University of Bremen, Germany

cgvr.cs.uni-bremen.de

Der "Bill Buxton Test"

- Zeichnen Sie einen Computer in 15 Sek.
- Ca. 80% der Fälle sehen so aus:
 - Monitor
 - Tastatur
 - Maus
- Interessant dabei:
 - Kein "Computer" auf dem Bild
 - Message: Benutzer nehmen "Computer" hauptsächlich über Ein- und Ausgabegeräte wahr

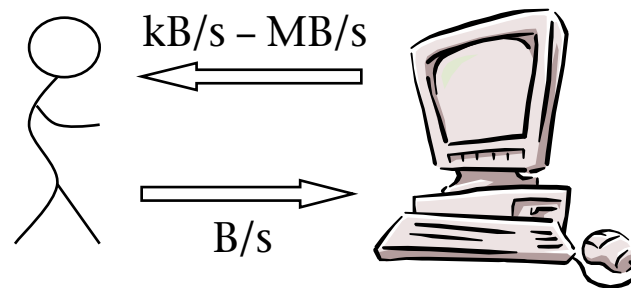


Extreme Beispiele eines "intrusive" I/O-Gerätes

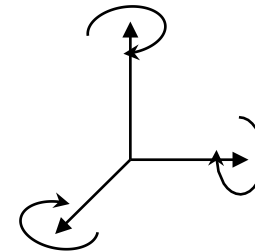
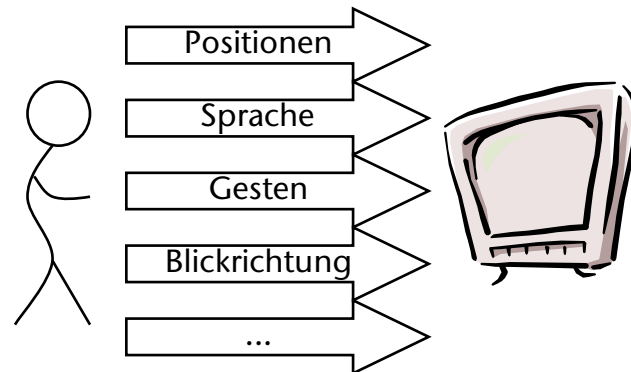




- Die "Spielwiese" der VR
- Vision: *keine* Eingabegeräte
- Bandbreite:



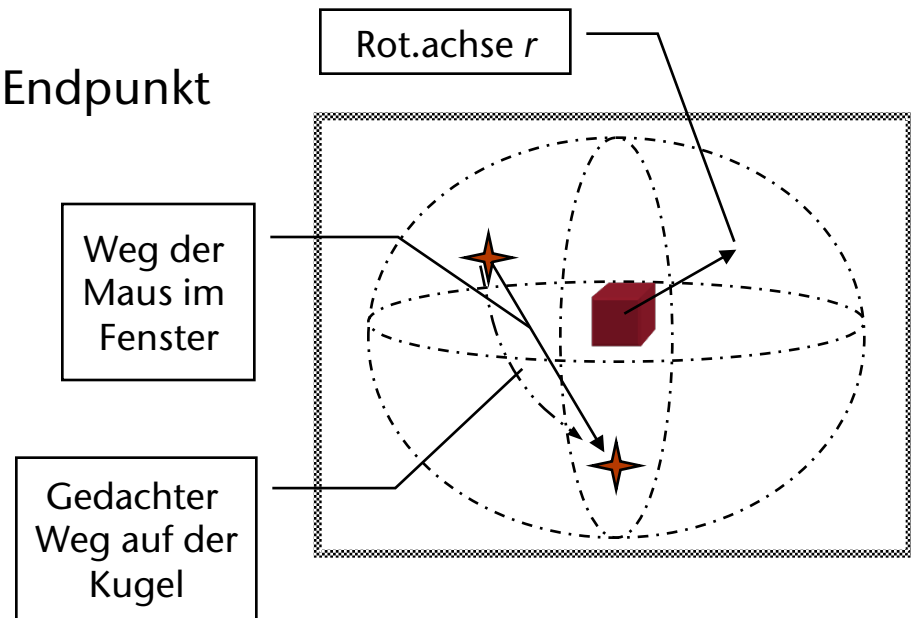
- Freiheitsgrade (= "*degrees of freedom*", DOF)
- Multimodal:



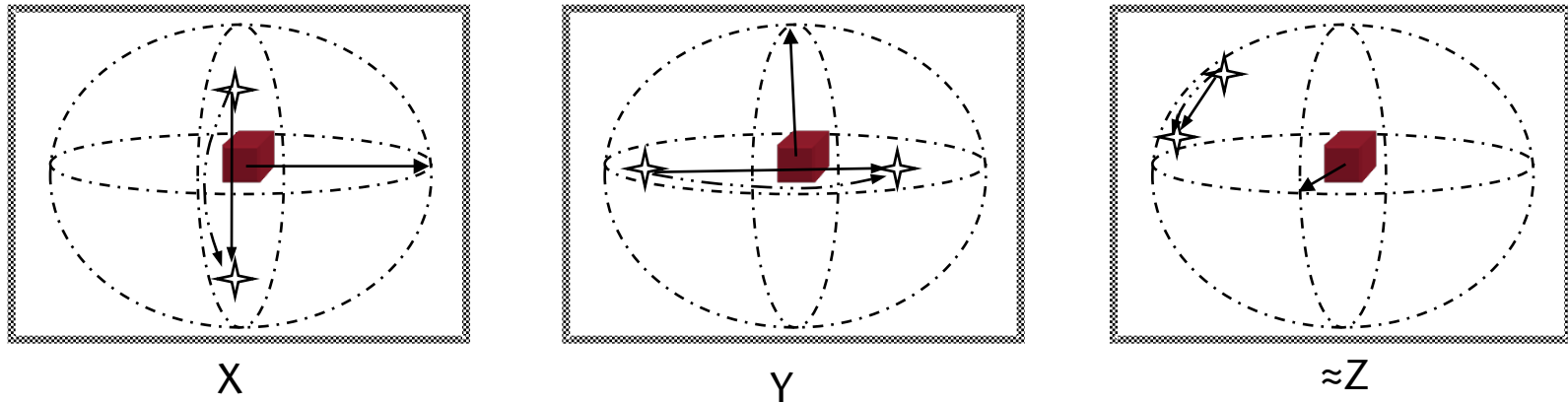
- Maus:
 - Präzise, billig
 - Nur 2D, Eingabe von Orientierungen mühsam
- Zeichentablett:
 - Präzise, gut fürs Zeichnen
 - 2D, Eingabe von Orientierungen fast unmöglich
- Lichtgriffel?



- Wie gibt man Orientierungen mit der Maus ein?
- Idee:
 - Lege Kugel um das Objekt / die Szene
 - Kugel kann um ihr Zentrum rotieren
 - Maus zieht Punkt auf Oberfläche der Kugel
- Berechnung:
 1. (x_1, y_1) Startpunkt, (x_2, y_2) Endpunkt
 2. $z = \sqrt{x^2 + y^2}$
 3. $\vec{r} = \vec{p}_1 \times \vec{p}_2$



- Man kann um alle Achsen (bis auf eine) direkt rotieren:



- Verbesserungen:
 - "*Spinning trackball*" (à la Inventor) vermeidet Nachfassen teilweise
 - "*Locking*" für exaktes Rotieren um eine Koord.achse

- Spacemouse:
 - 6 DOF
 - Gut für CAD, Viewpoint-Navigation, Szene rotieren

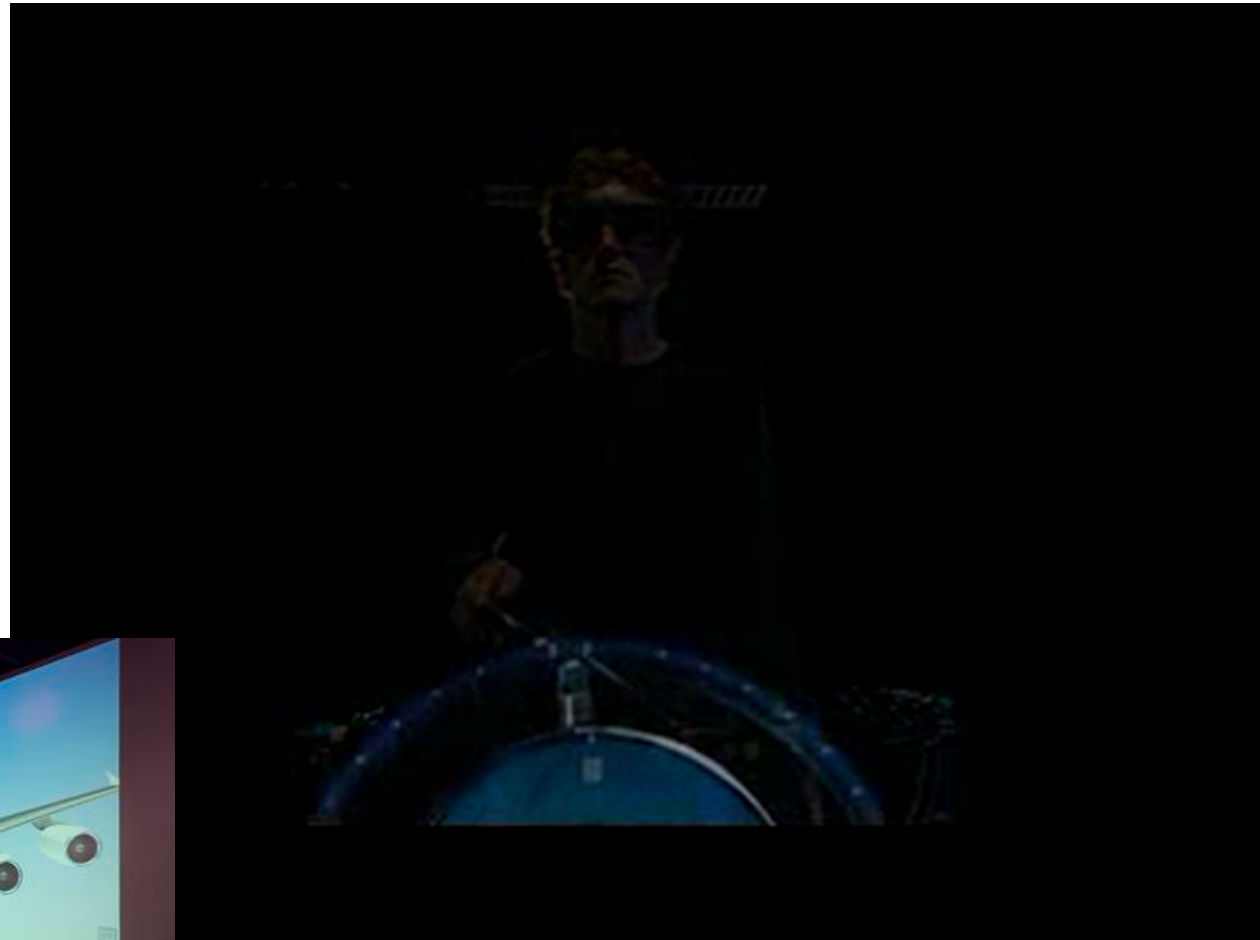
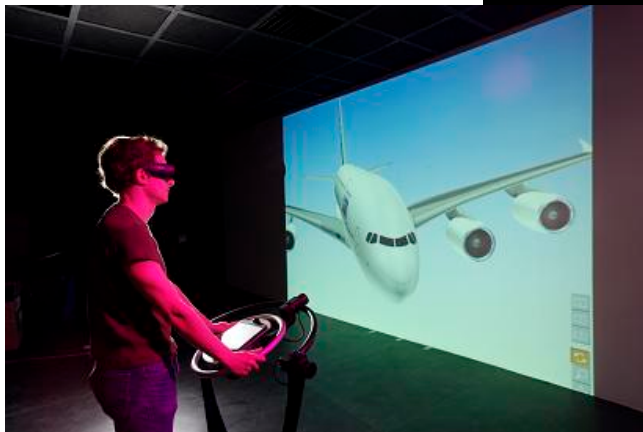


- Lenkrad
 - Mit Force-Feedback
- Weitere ? ...



Beyond Desktop: CAT – Control Action Table

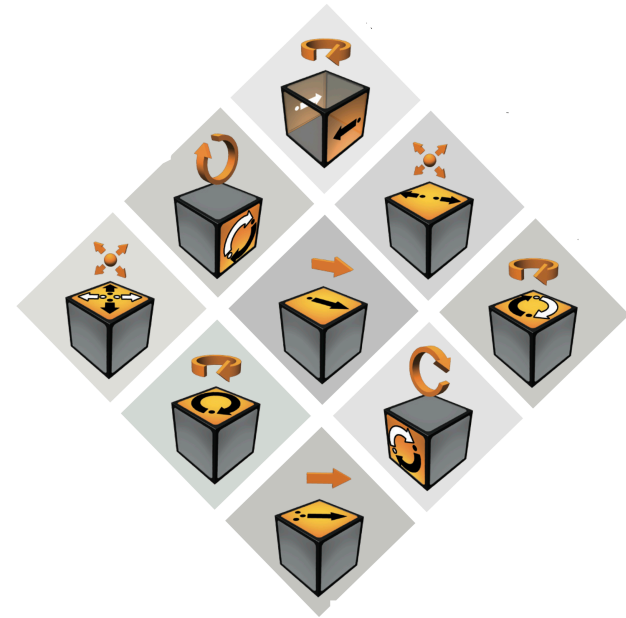
- 6 DOF free-standing, plus tablet



Project "IPARLA", INRIA, France

Cubtile

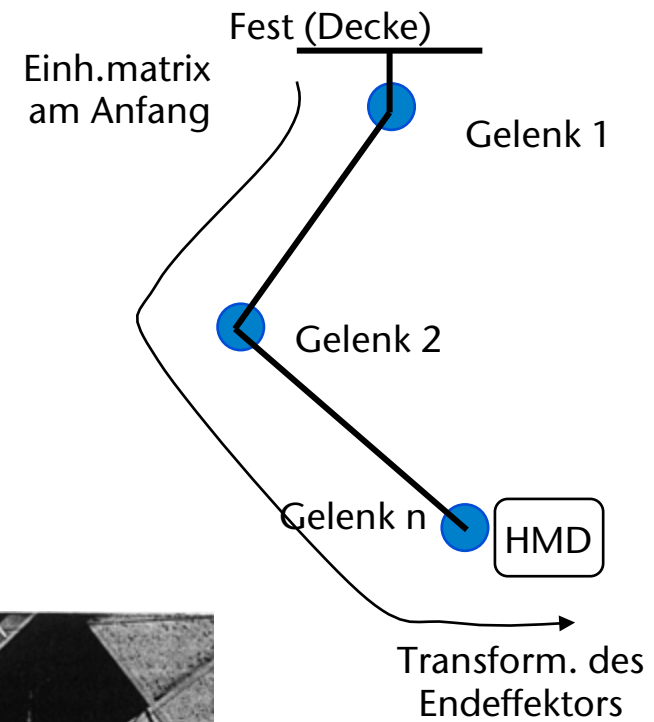
- 5 multi-touch-Flächen in einem Würfel
- Bonus: hübsche Beleuchtung ☺



- Aufgabe: "Wo befindet sich X des Users?"
 - X = Kopf, Hand, Augen, Füße, gesamter Körper, ...
- Anforderungen:
 - *Non-intrusive*
 - Hohe Genauigkeit (1 mm)
 - Geringe Latenz (1 msec)
 - Hohe *Update-Rate* (100 Hz)
 - In jeder Umgebung und Situation
 - Großer Bewegungsspielraum
- Existiert nicht!

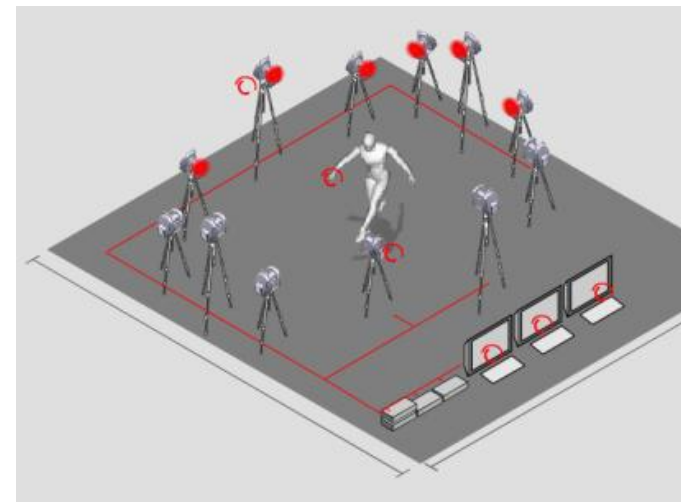
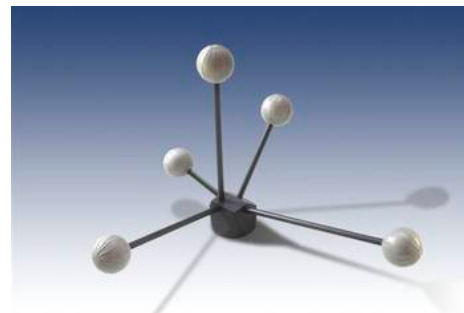
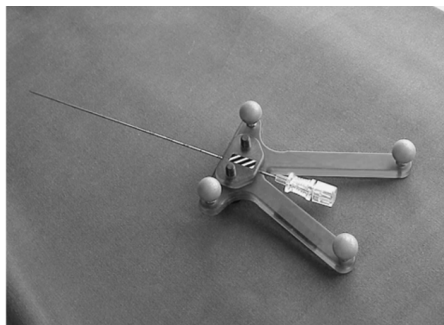
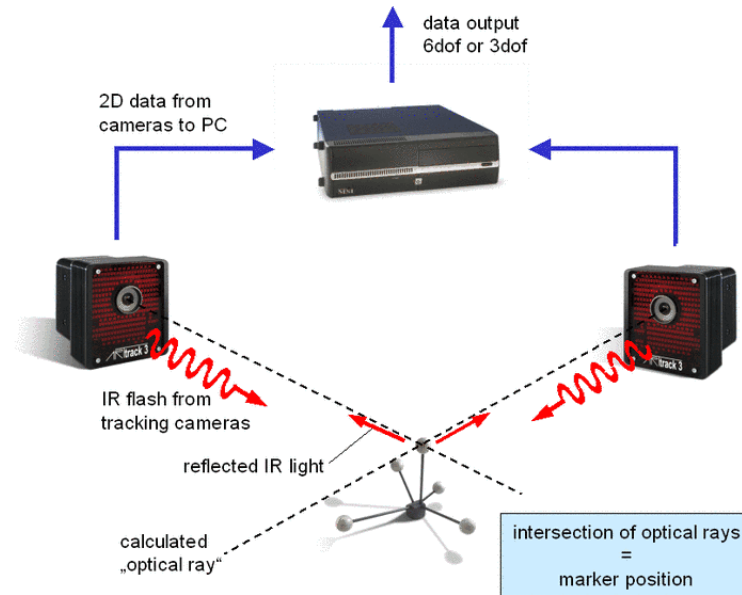
- Arten:
 - Mechanisch
 - Elektro-magnetisch
 - Akustisch (Ultraschall)
 - Optisch
 - Computer-Vision
 - Trägheitssensoren
 - Laser
 - GPS
 - Hybride

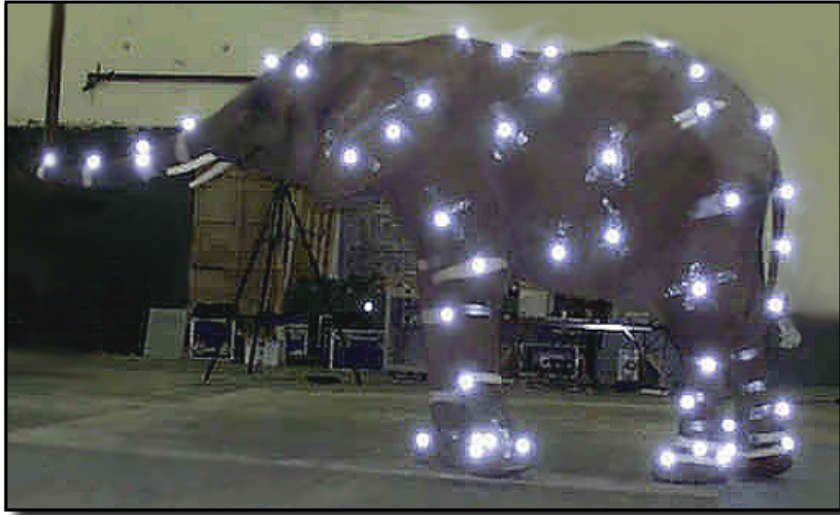
- Vorteile:
 - Präzision
 - Keine Latenz
 - Keine Störung durch Metall
- Nachteile:
 - Unbequem
 - Reichweite
 - "Tote" Winkel
 - Festmachen am Körper
 - Kalibrierung
 - Trägheit



Optisches Tracking

- Verfolgen von reflektierenden Markern mit IR-Kameras
- 1 Marker → Position
 - Mittels **Triangulation**
- ≥ 3 Marker ("*rigid body*") → Position und Orientierung
- Standard-Technik für *Body-Tracking* in Animationsstudios und für Spiele
 - **Motion Capturing (MoCap)**





- Vorteile:

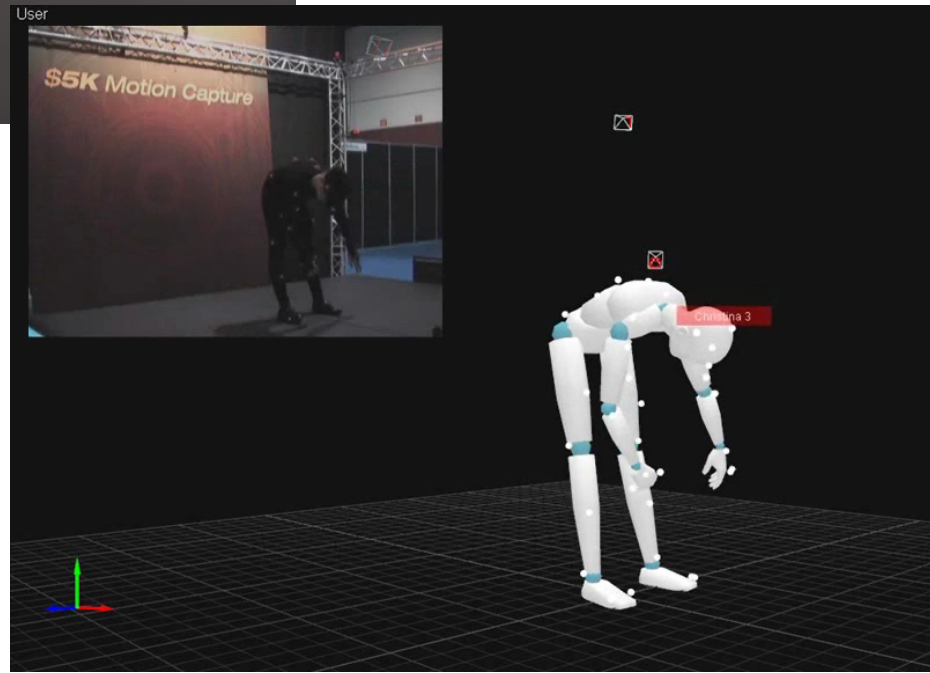
- Leicht
- Freie Bewegung
- großes Volumen
- hohe *Sampling-Rate* (typ. 120-250 Hz)
- *Facial animation* geht auch

- Nachteile:

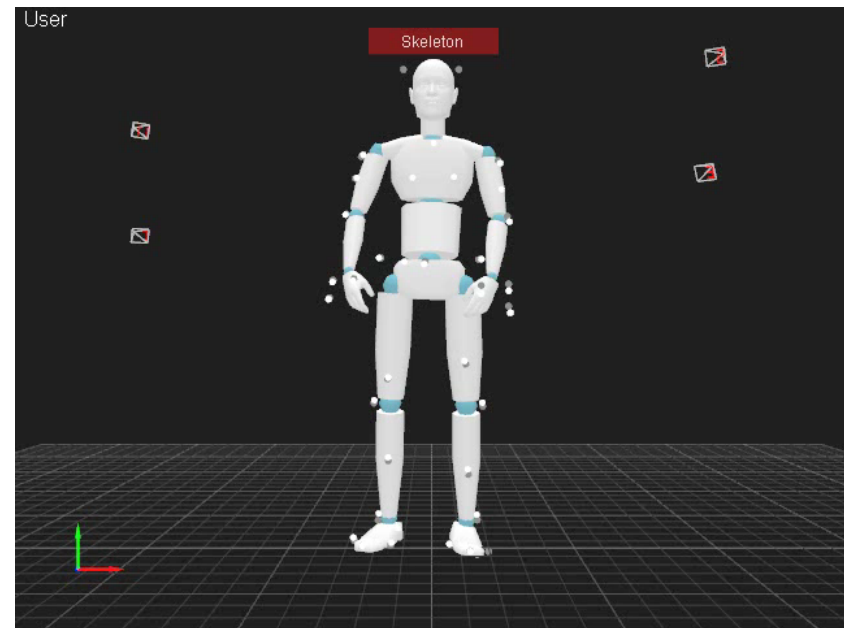
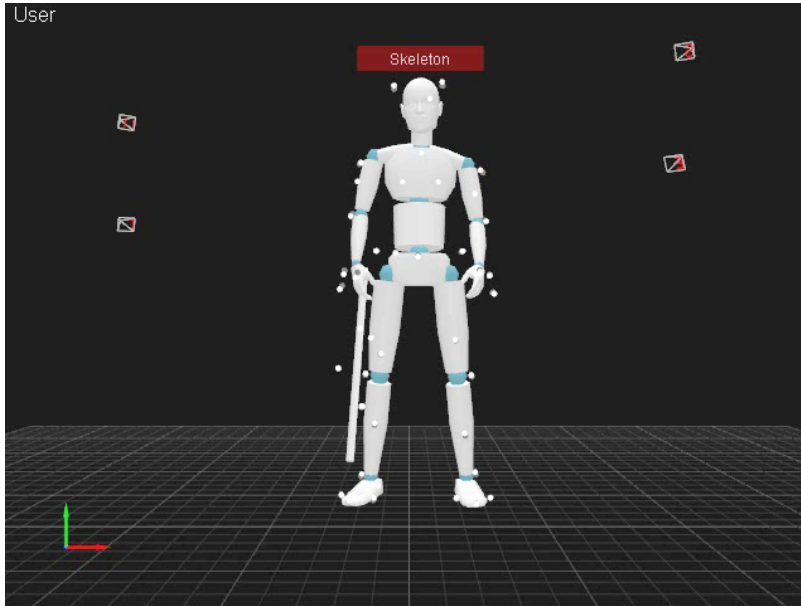
- *Line-of-sight* (viele Kameras)
- Preis (\$40,000 – \$140,000)
 - Seit kurzem nur noch \$6,000
- Inzwischen auch *real-time* (kein *post-processing*)



Fluid Images



NaturalPoint (OptiTrack)



Optisches Tracking "*inside out*"

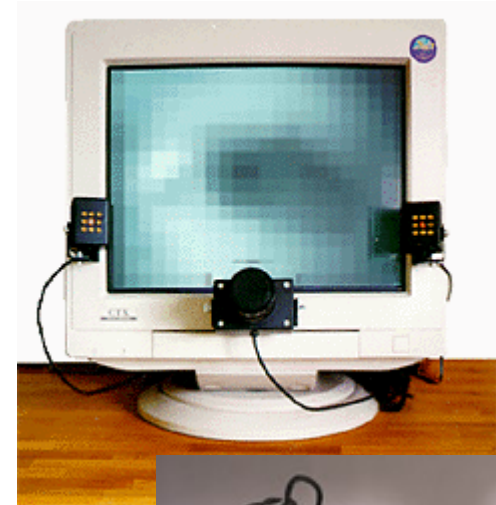
- Kamera auf dem Kopf,
"sieht" Array gepulster LEDs an der Decke
- Vorteile:
 - Nur 1 Kamera nötig
 - Schnell (1 msec, 1500 Hz)
 - Genau (1/10 mm)
- Nachteile:
 - Wie trackt man die Hand?
 - Aufwendige Installation
- Beispiel: UNC's "HiBall"

<http://www.cs.unc.edu/~tracker/>



Eye-Tracking

- Wo befinden sich die Augen des Users?
Wohin schaut der User gerade?
- Anwendungen:
 - Head-Tracking
 - LOD-Steuerung
 - Autostereo-Monitore
- Probleme:
 - Präzision, insbesondere bei Orientierung



- Ähnlich zu Echolot
 - 1 Ultraschall-Quelle
 - 3 Empfänger (für 3 DOF)
 - Laufzeit → Position
- Vorteile:
 - Billig
- Nachteile:
 - Echos
 - *Line-of-sight*
 - 3 Sender für 6DOF
 - Geringe Reichweite
 - Schallgeschwindigkeit hängt ab von Lufttemperatur



- Misst Beschleunigung in eine Richtung
- Vorteile:
 - Kein Sender nötig
 - Klein
- Nachteile:
 - Drift
- Oft in Kombination mit anderen Tracking-Verfahren, z.B. Ultraschall



- Mißt i.A. nur Position
- Bisher nur in der Fertigungsindustrie (CNC-Maschinen)



- Sender stationär, Empfänger = Sensor
- Langwelliges Feld, Phasenverschiebung empfangenem Signal → Entfernung
- 3 Spulen im Sender (3 versch. Frequenzen) pro Sensor 3 Empfängerspulen orthogonal → 9 empfangene Feldstärken
- Vorteile:
 - Kleine Sensoren
 - Reichweite 3m (auch mehr)
- Nachteile:
 - Kabel
 - Fremdmetall stört
 - Rauschen



Allg. Charakteristika

1. # DOFs
2. Präzision, Drift, Wiederholbarkeit
3. Update-Rate, Latenz
4. Rauschen
5. Zusätzliche Buttons
6. Bequemlichkeit ("*ease-of-use*"), kabellos (*tethered*) / verkabelt (*unethered*) – *unintrusiveness*!
7. Arbeitsvolumen
8. Preis

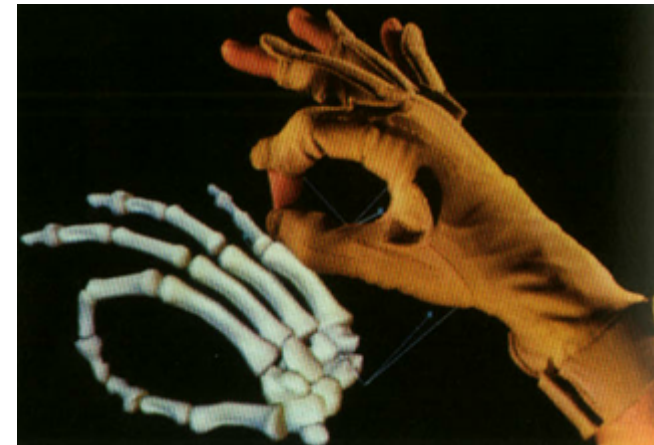
3D-Zeiger

- Analogon zur 2D-Maus.
- Hardware = Tracker mit Buttons dran
 - Evtl. zusätzlich mit Joystick oder Jog-Dial
- Namen: *flying mouse*, *flying joystick*, *wand* (= Stab),
bone, *fly-stick*, etc...
- Physisches Objekt ergibt starkes Präsenzgefühl, wenn man virtuelles Objekt gegriffen hat

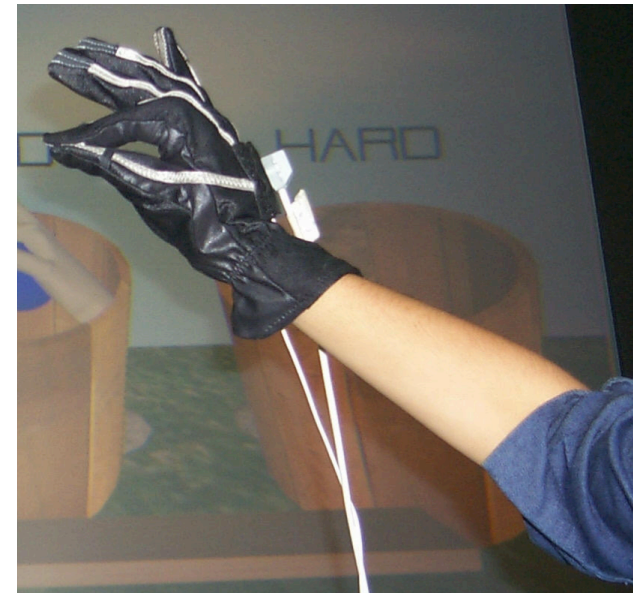


Datenhandschuh (data glove)

- "Trackt" Finger = mißt Winkel der Fingergelenke
- Das erste VR-Eingabegerät
- Verschieden viele Sensoren:
 - Min. 4x Daumen + 4x2 Finger = 12
 - Max. 4x Daumen + 4x3 Finger + 3x dazwischen + 2x Handgelenk + 1 Handrücken = 22
- Technik:
 - Glasfaser (nicht bewährt)
 - Bimetallstreifen
- Nachteile:
 - Niedrige Genauigkeit
 - Handschuh (umständlich, Akzeptanz)
 - (Mehr DOFs als gebraucht)



- Pinch Glove:
 - Kein Tracking, misst nur Kontakt zweier Finger → jeder Finger ein Button
 - Nur 2 getrackte Gloves sinnvoll, damit recht pfiffige Navigation und Objektmanipulation möglich:
 - Greifen und Bewegen
 - Skalieren (*Handles à la Inventor*)
 - Virtuelle Hand nicht darstellbar

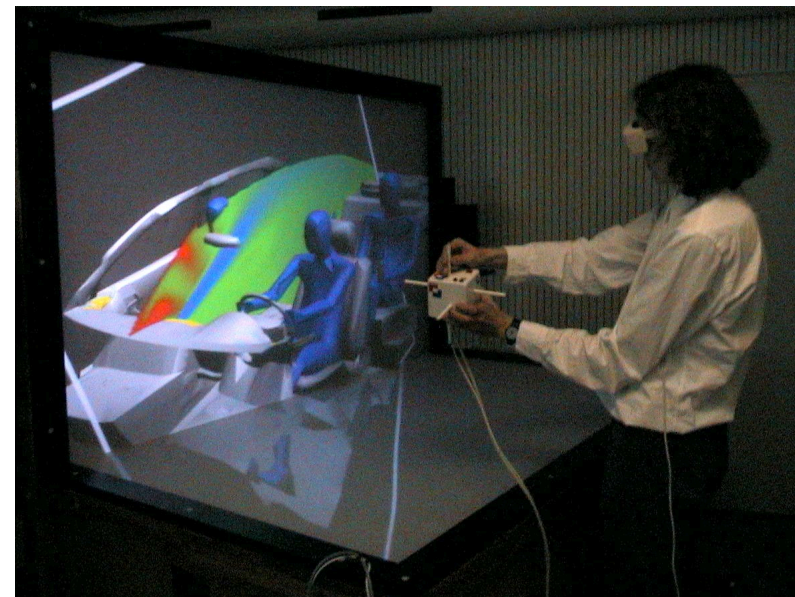
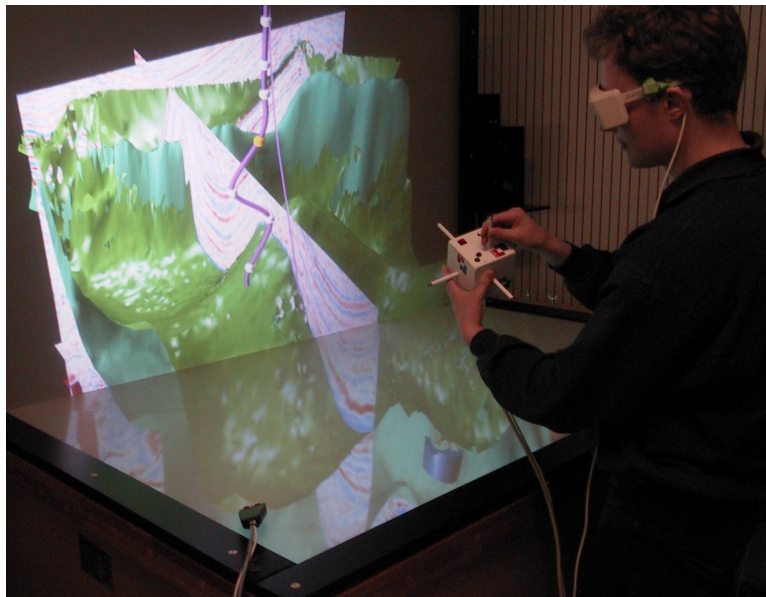
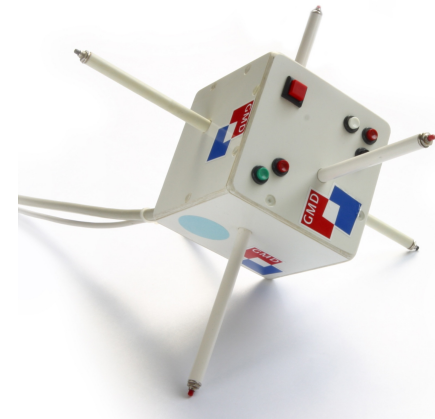


- Der P5 von Virtual Realities (www.vrealities.com):



Weitere hoch-dimensionale Input-Geräte

- Cubic Mouse:
 - Anzahl Freiheitsgrade = 9

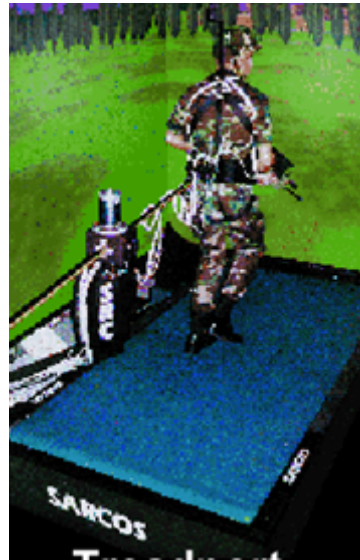




Fortbewegungseingabegeräte (*locomotion devices*)



Sarcos, Utah



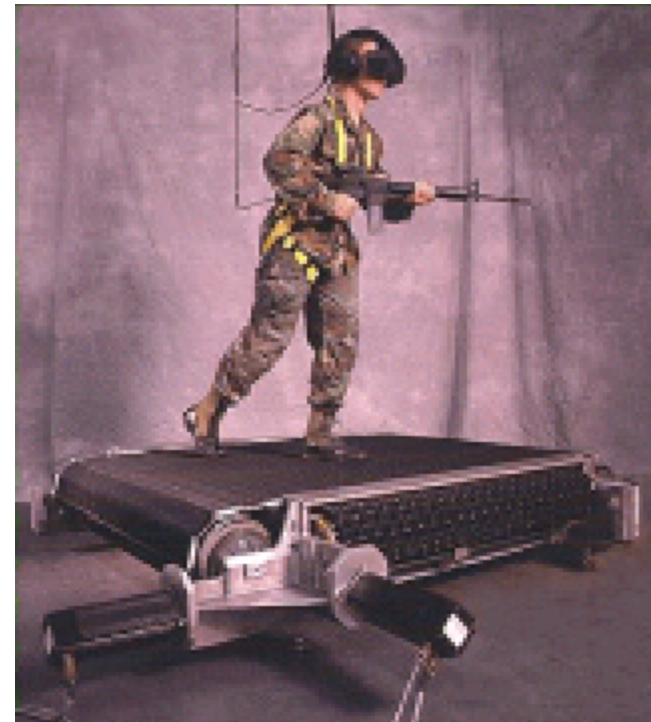
Sarcos



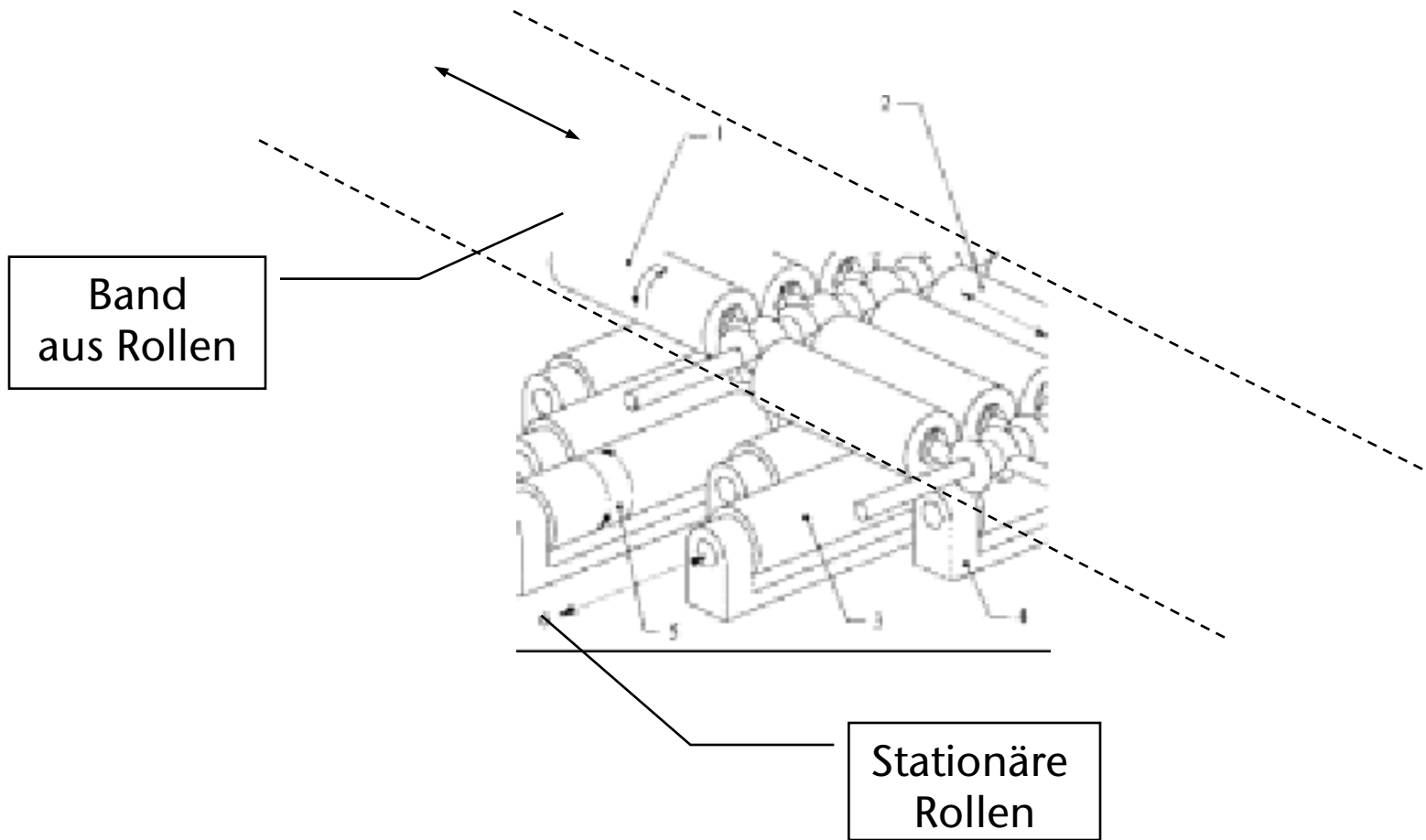
Uni Tsukuba, Japan

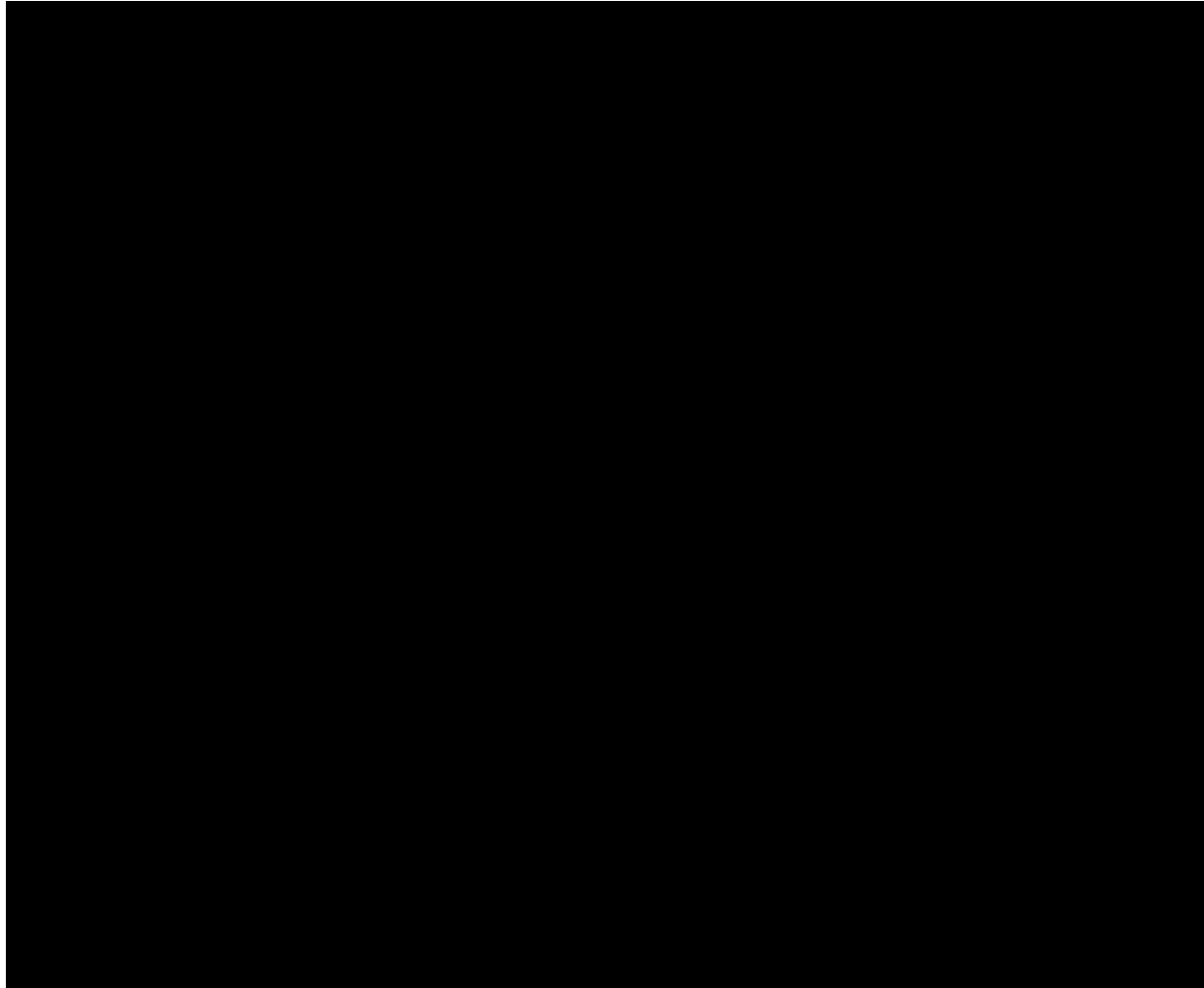


- *Omni-Directional Treadmill* (omni-direktionale Tretmühle)

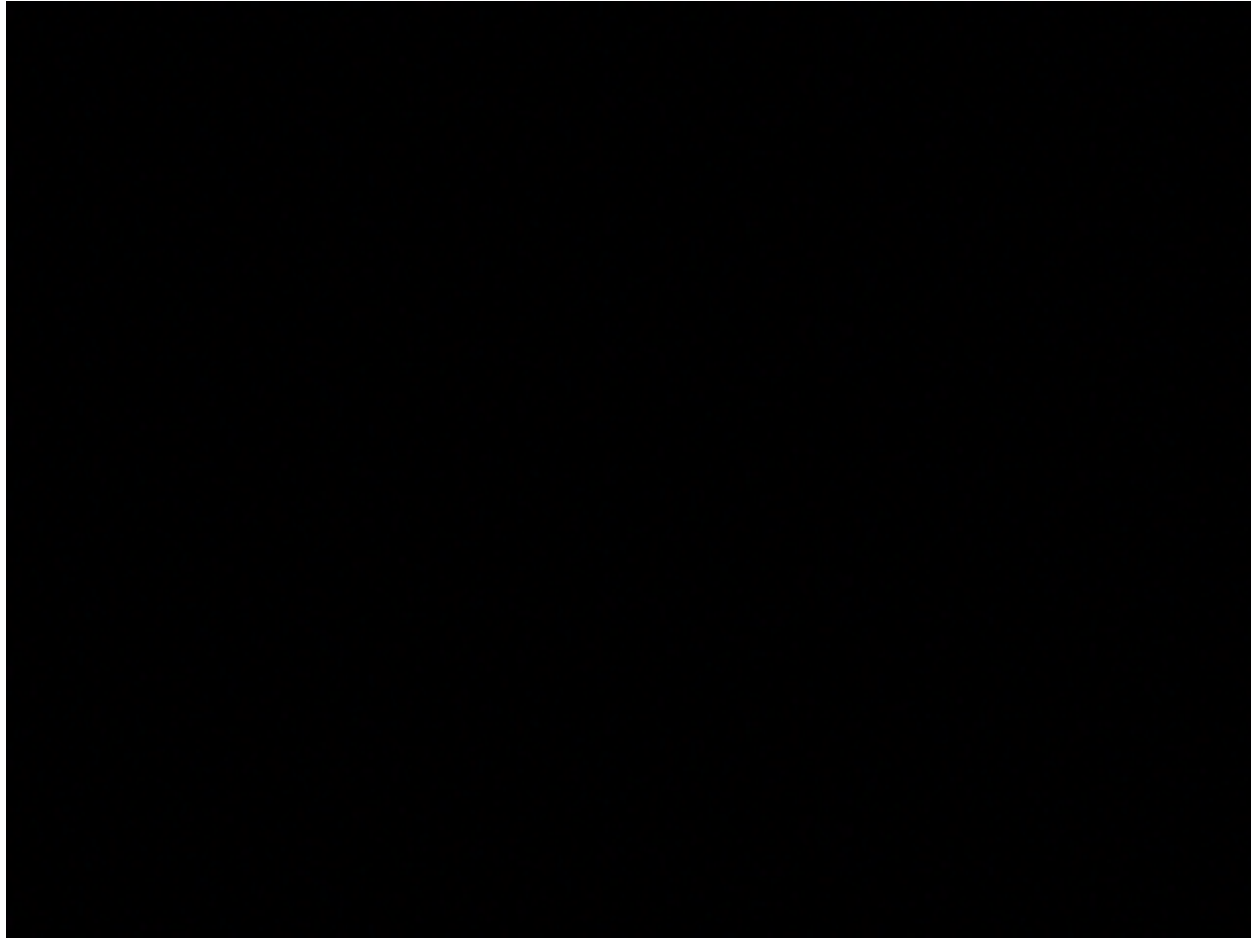


Virtual Space Devices, Inc.





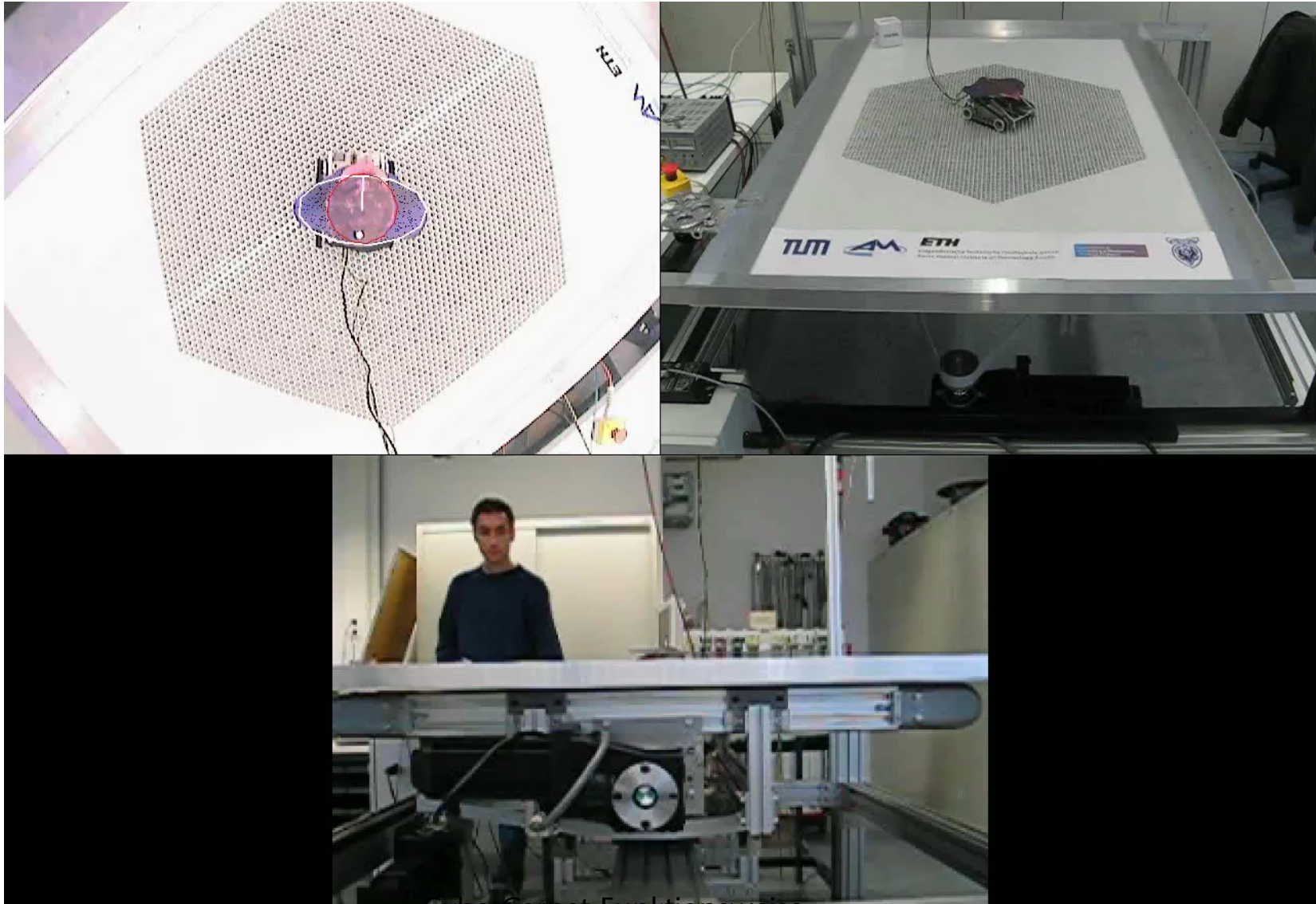
Cyberwalk omni-directional treadmill, 2005-2008
TU München



Funktionsweise der Cyberwalk omni-directional treadmill

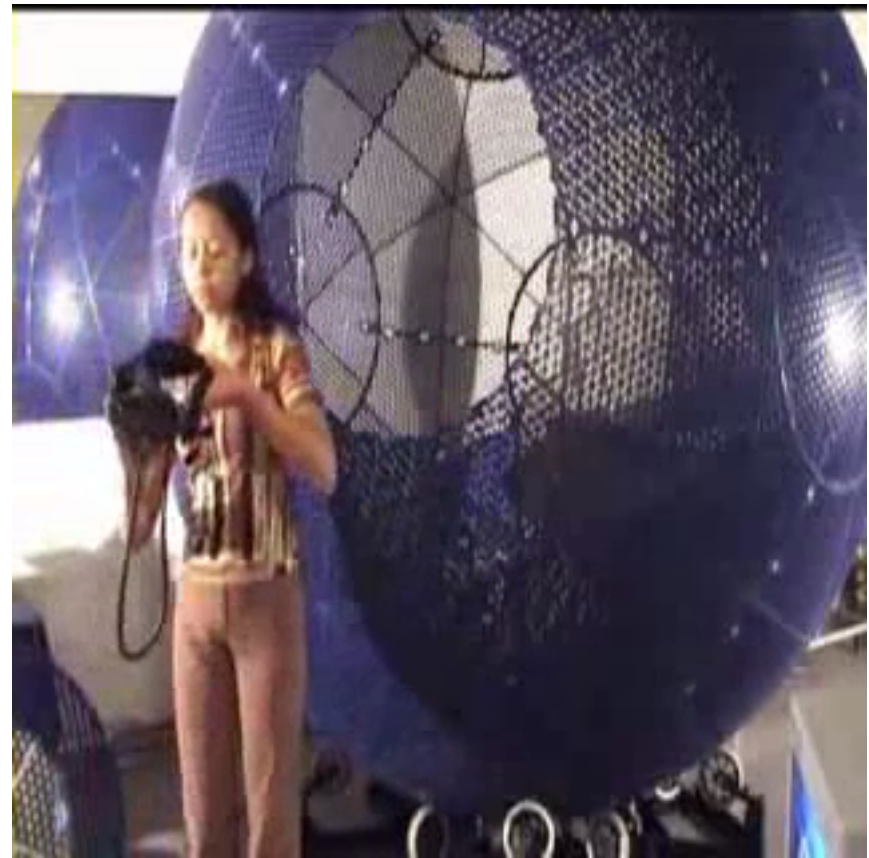


CyberCarpet
Martin Schwaiger, Dr. Thomas Thümmel, TU München

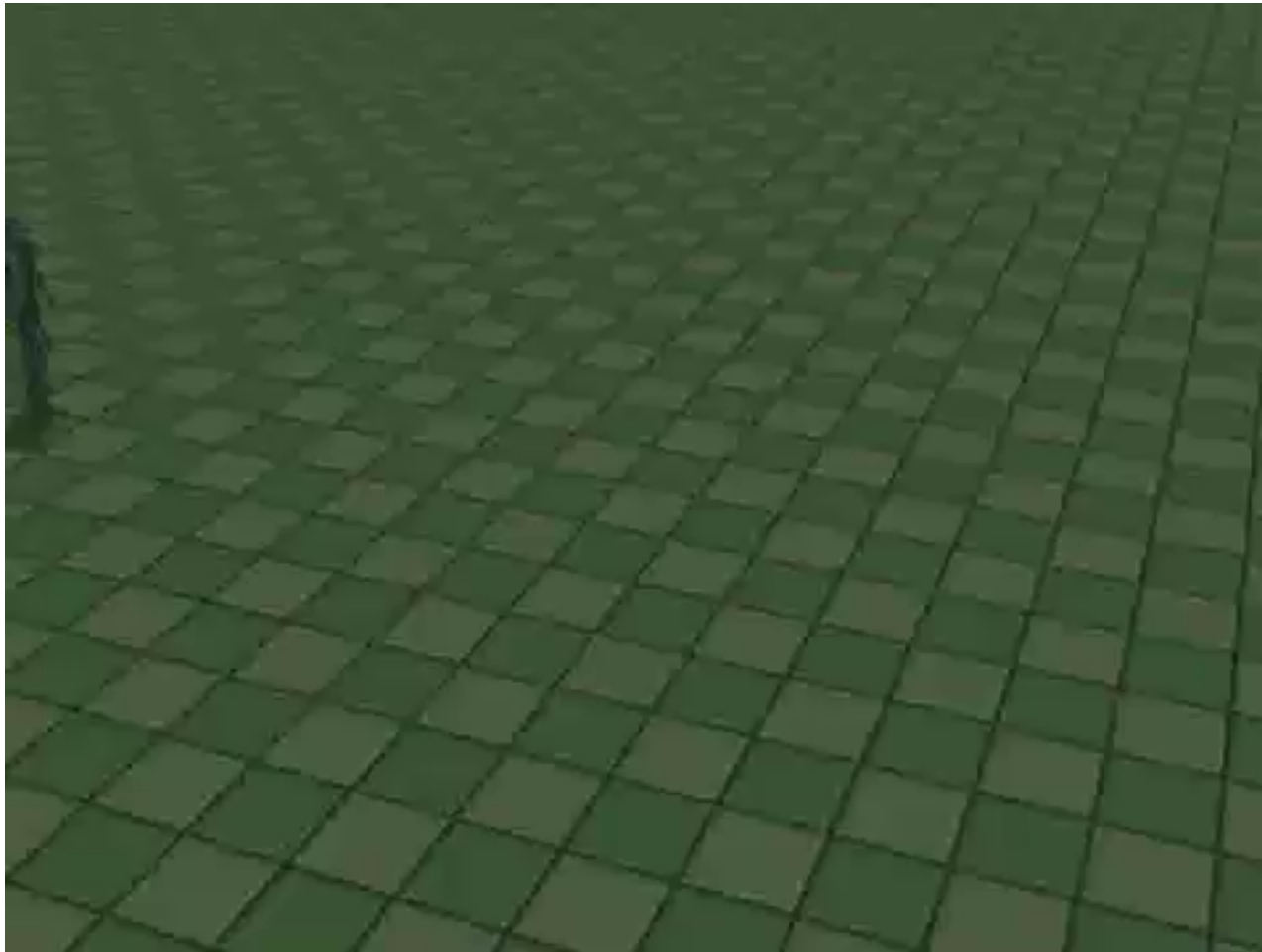


CyberCarpet-Funktionsweise

- Mögliche Anwendungen:
 - Verhaltensforschung, Gehirnforschung
 - Sportmedizin
 - Training von Soldaten und Sicherheitskräften
 - Fabrikplanung (?)
 - Erlebniswelten
 - Architektur:
 - Visualisierung und Begeharmachung von Entwürfen
 - Test von Fluchtwegen
 - Test von humanoiden Robotern



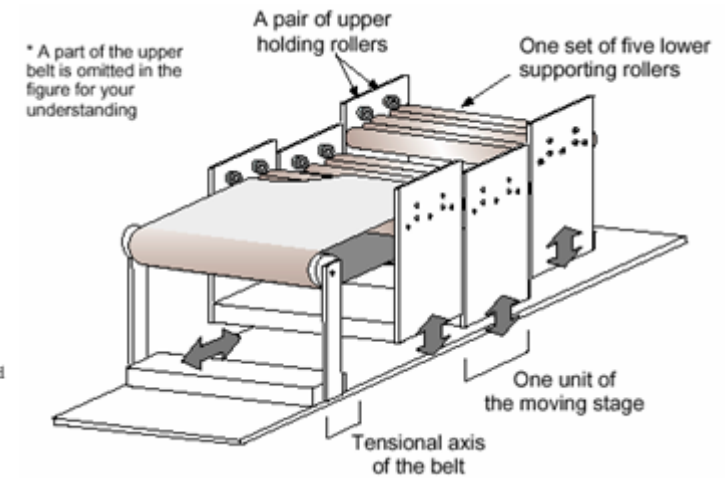
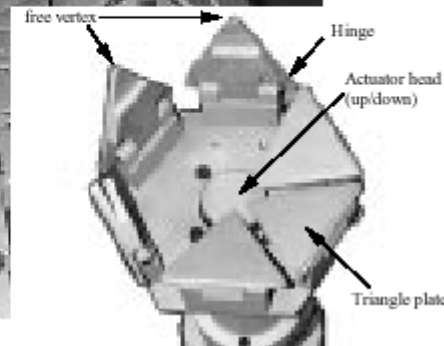
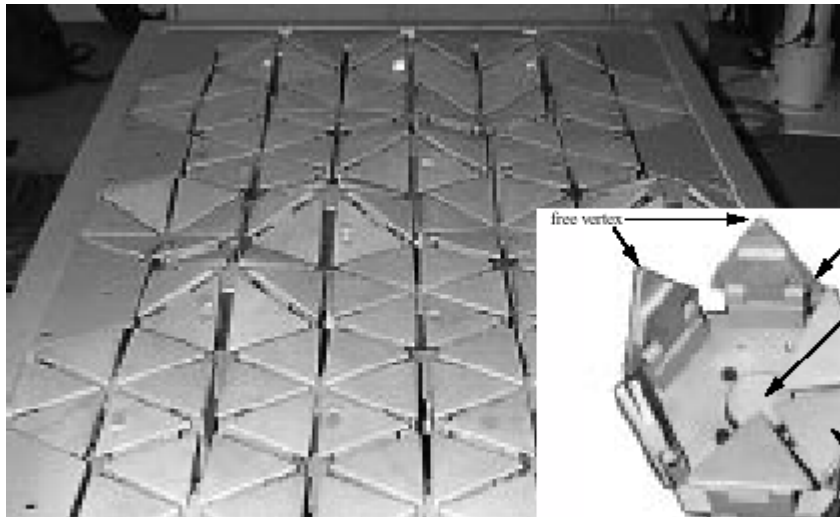
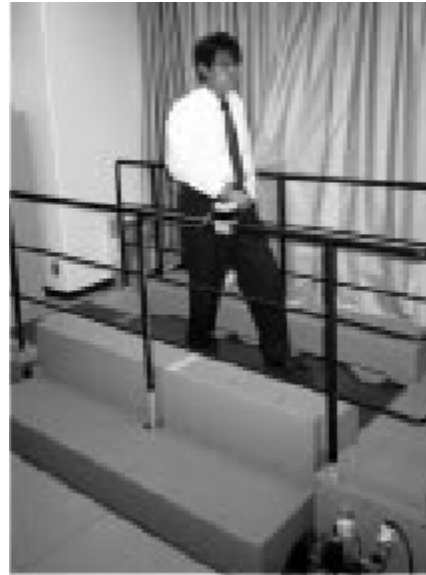
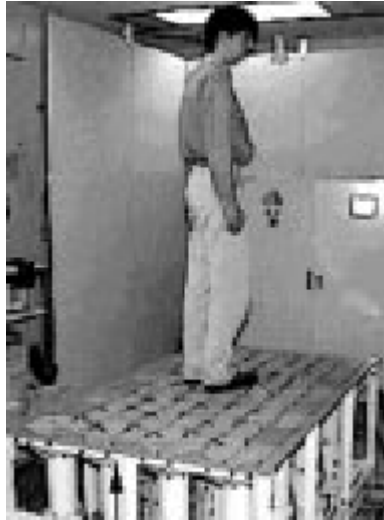
VirtuSphere



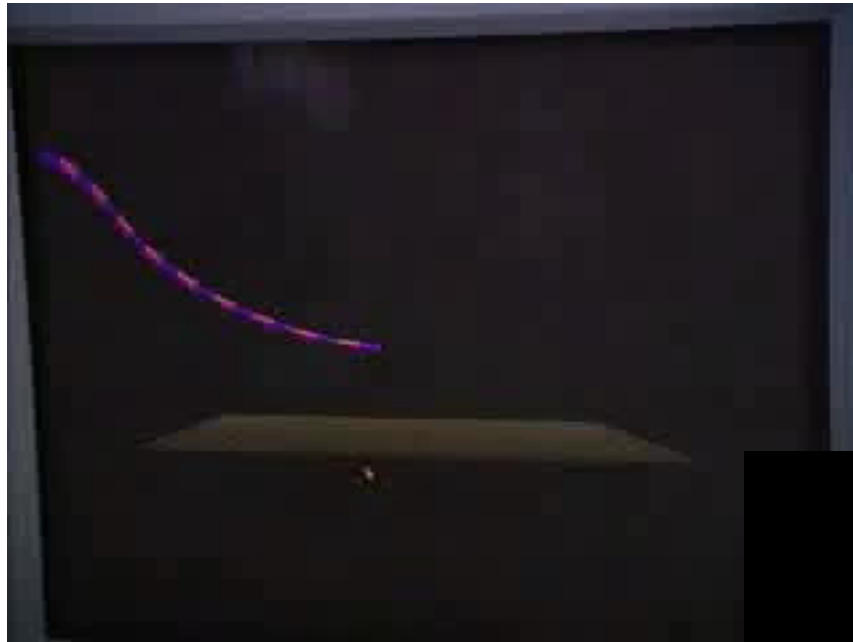
CirculaFloor, 2006



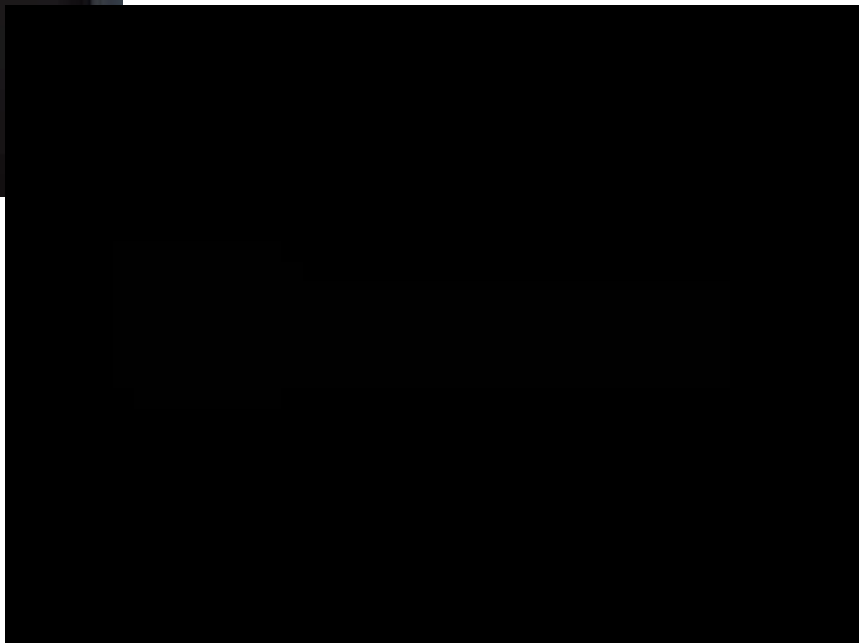
Bodenoberflächensimulator







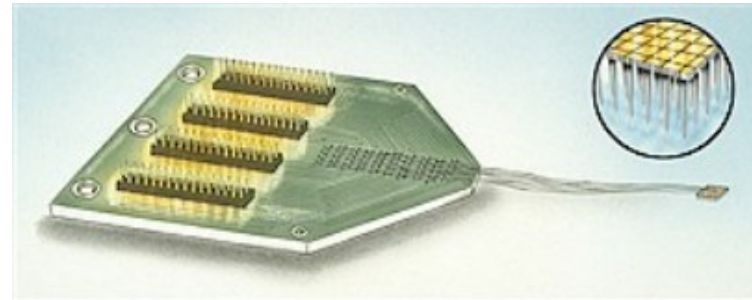
Shape tape



Virtual Keyboard

Brain-Computer-Interfaces

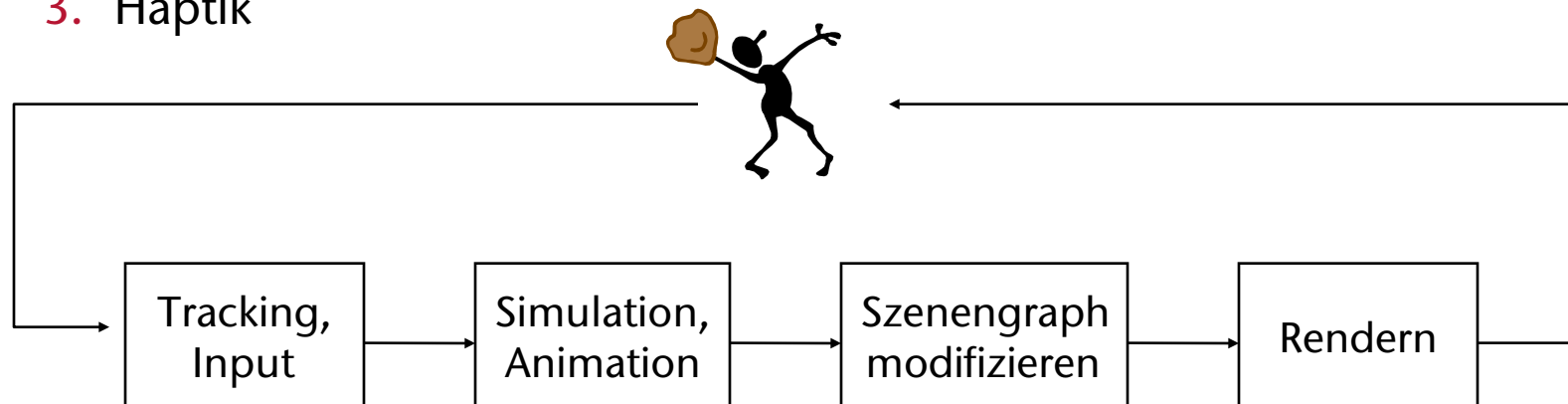
- Steuerung durch:
 - EEG, oder
 - Implantat



Exkurs: Affective Computing

- Sense user's attention and emotions, then alter system behavior accordingly.
- Parameters:
 - Gesture, posture
 - Voice
 - Eye gaze
 - Breathing
 - Pulse & blood pressure
 - Electrical activity of muscles
 - Skin conductance
- <http://www.media.mit.edu/affect/>
- Sense user's health: <http://www.bodymedia.com>,
Pilotversuch in NRW(?) mit Herzpatienten

1. Hole *Tracking-Daten*
2. Transformiere Geometrie und Viewpoint
3. Hole "binäre" Eingaben (Gesten, Sprache)
4. Simuliere und animiere Objekte
5. Rendere ...
 1. 2x Bild
 2. Sound
 3. Haptik



- Problem:
 - Relative / absolute Geräte
 - Verschiedene Dimensionalität
 - Verschiedene Interfaces
- Lösung:
 - Abstraktion "log. Gerät"
 - gemäß Dimension
 - Alle log. Geräte absolut (integrieren)
- Logische Geräte [angelehnt an Wallace 1976]:
 - 0D = "Button" (bool)
 - 1D = "Value" (float)
 - 6D = "Space" (matrix)
 - 1-aus-n = "Choice" (int)
 - Glove (float-array)

- **Abbildungsmatrix:**

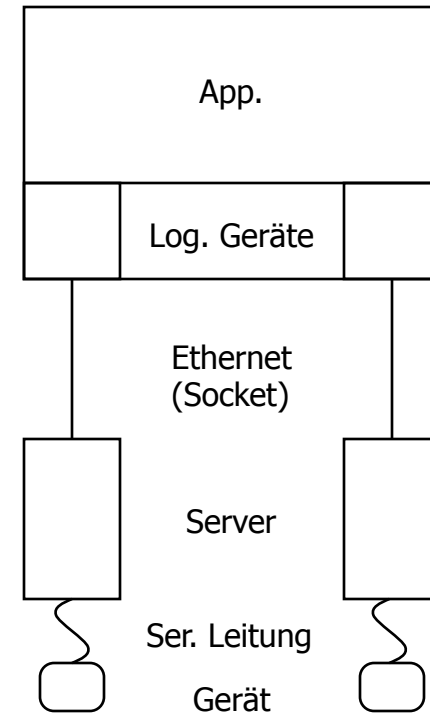
	Maus	Space- mouse	Trak- ker	Spra- che	Tasten	Lauf- band	Glove	Dial
Button	x	x	(x)	x	x	(x)	x	
Value	(x)	(x)	(x)	(x)		x	x	x
Space	(x)	x	x					
Choice	x	x					x	

- **Ablauf:**

- Initialisierung mit Parametern (phys. Gerät, Port, ..)
- Danach nur noch "logischen" Wert abholen
- Relative Geräte müssen über die Zeit integrieren

- Anforderungen an Architektur:
 - Gerät an beliebigem Rechner → Client-Server
 - Viele Clients pro Server möglich
 - Fehlertolerant, falls falsche Parameter, Gerät nicht angeschaltet, etc.
 - Austauschbarkeit der Geräte
 - Unabhängige Sampling-Rate

- 2 QoS: schnell oder zuverlässig:



Datenart	Behandlung der Latenz	Transportart	Datenstruktur
kontinuierlich	"besser nie als spät"	UDP	Shared mem
diskret	"besser spät als nie"	TCP	Queue

