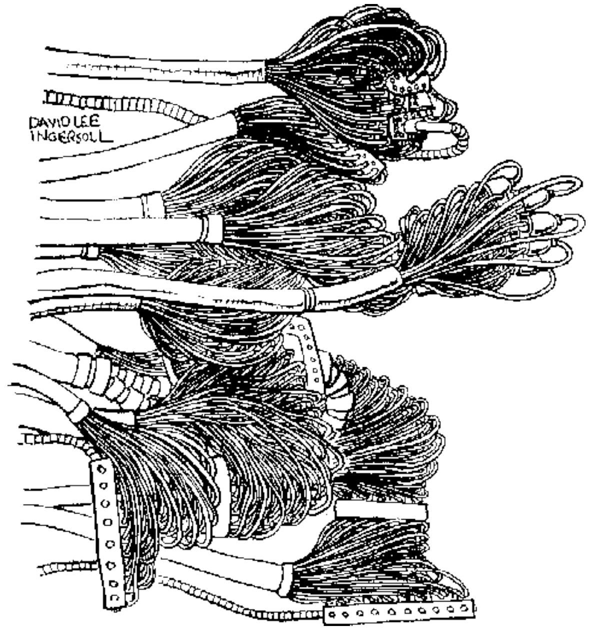
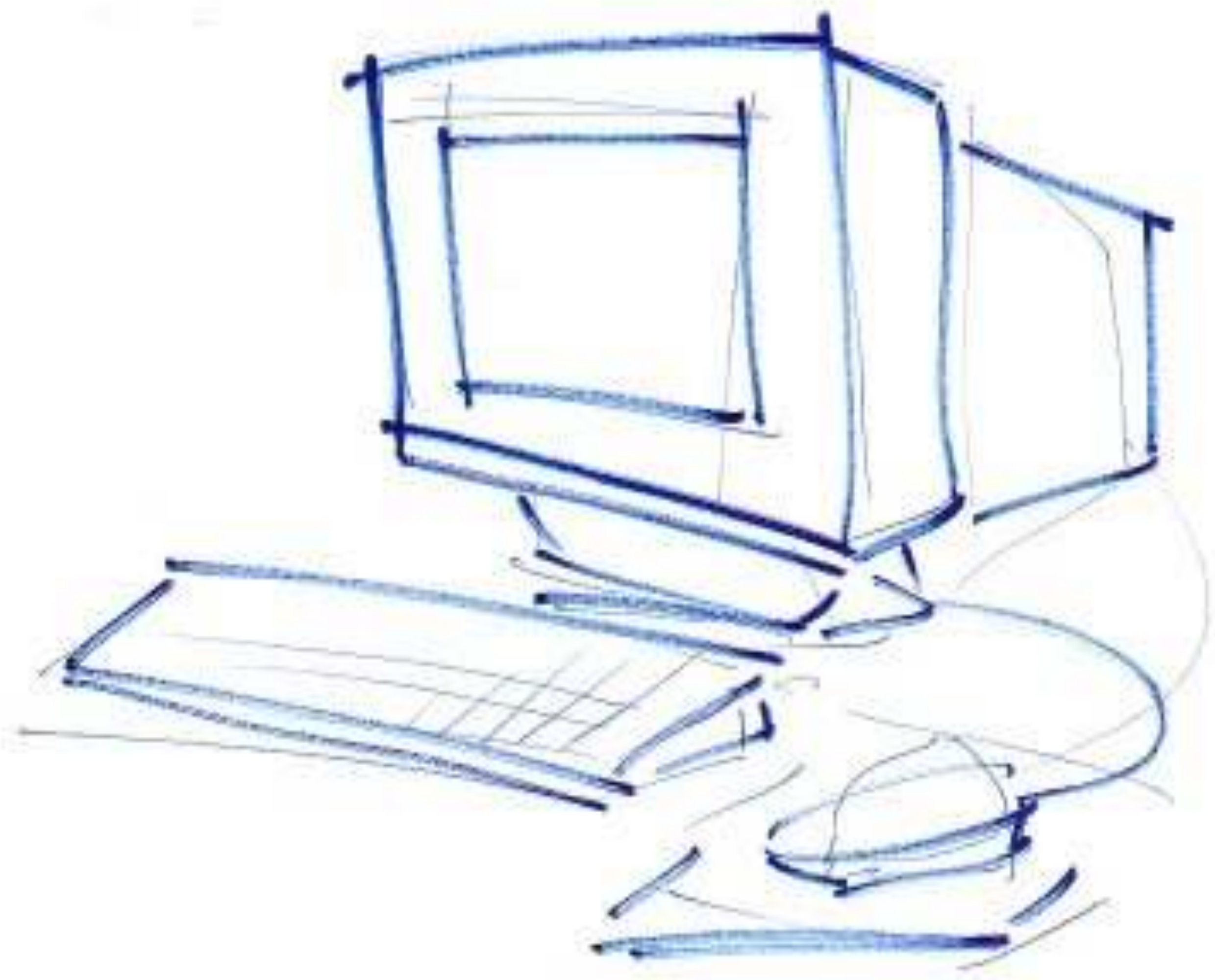
Virtual Reality & Physically-Based Simulation
Principles of Input Devices



G. Zachmann
University of Bremen, Germany
cgvr.cs.uni-bremen.de

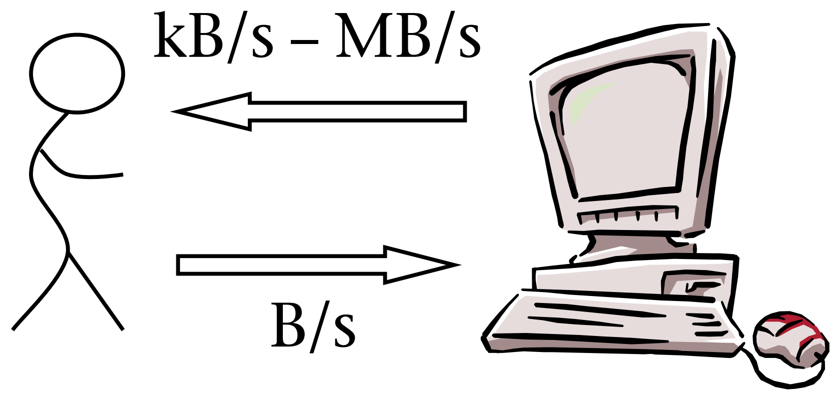
The "Bill Buxton Test"

- Draw a computer within 15(!) seconds
- Ca. 80% of all people draw something like this
  - Monitor
  - Keyboard
  - Mouse
- Remarkable:
  - No "computer" in the drawing!
  - Take-away message: users don't perceive the system as a computer, they just see a device on its surface, and they just perceive some kind of I/O behavior
    $\longrightarrow$ the interaction with the device is critical for success/failure

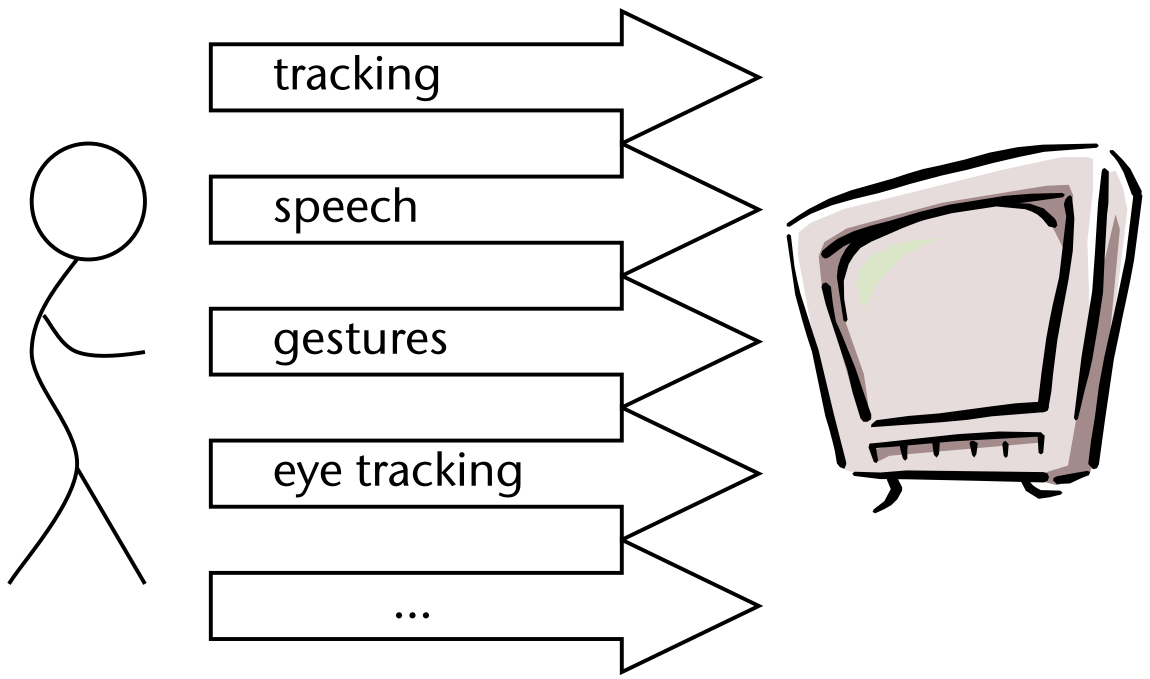


The Promise of Virtual Reality

- Problem of conventional input devices: bandwidth



- Multimodal input = input using different modalities, e.g., tracking and voice
  - Post-WIMP interfaces ("WIMP" = windows, icons, menus, pointers)
  - Challenge: make the devices non-intrusive

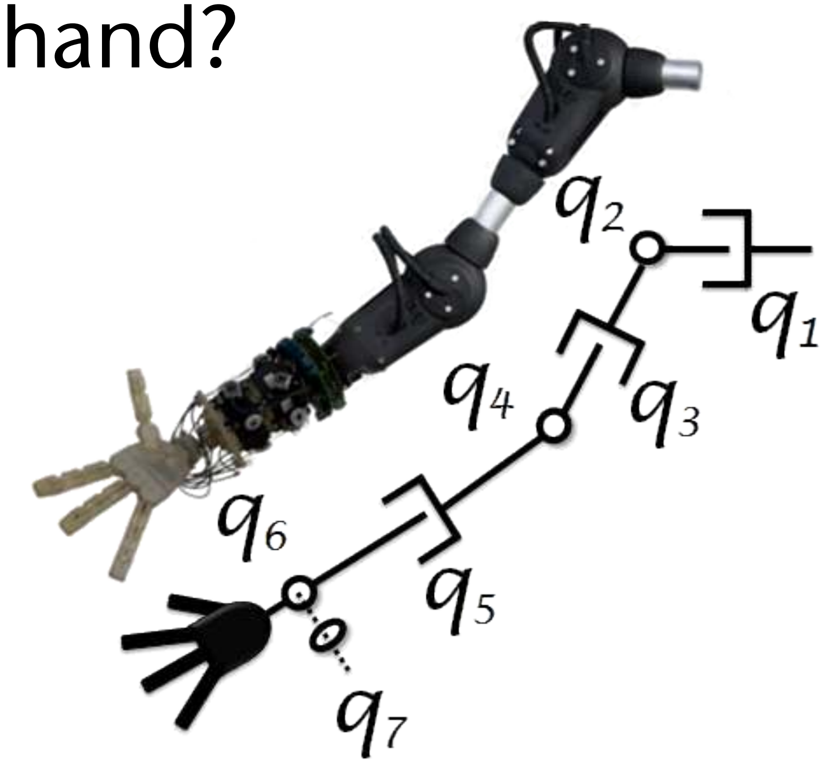


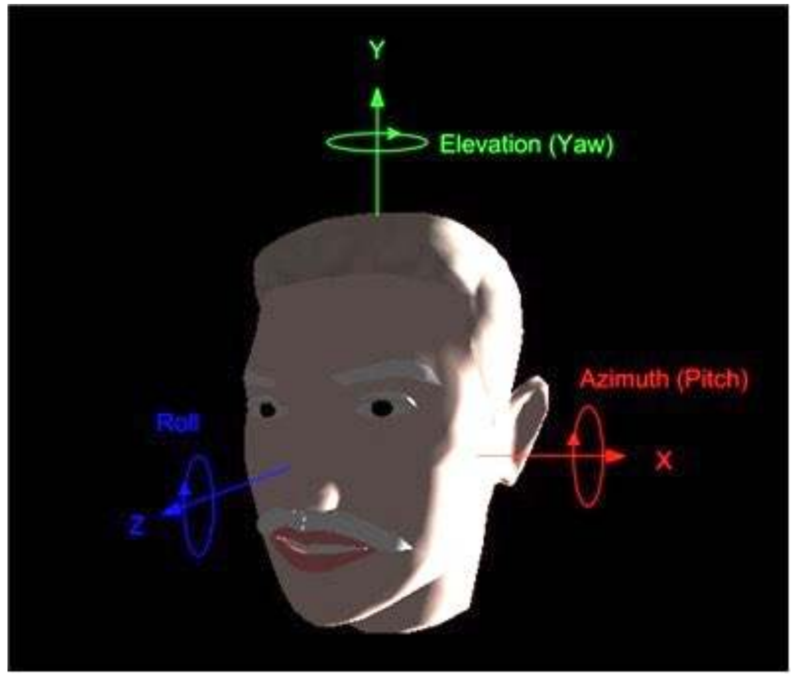
- Ultimate goal: "natural" user interaction (like in real life (?))

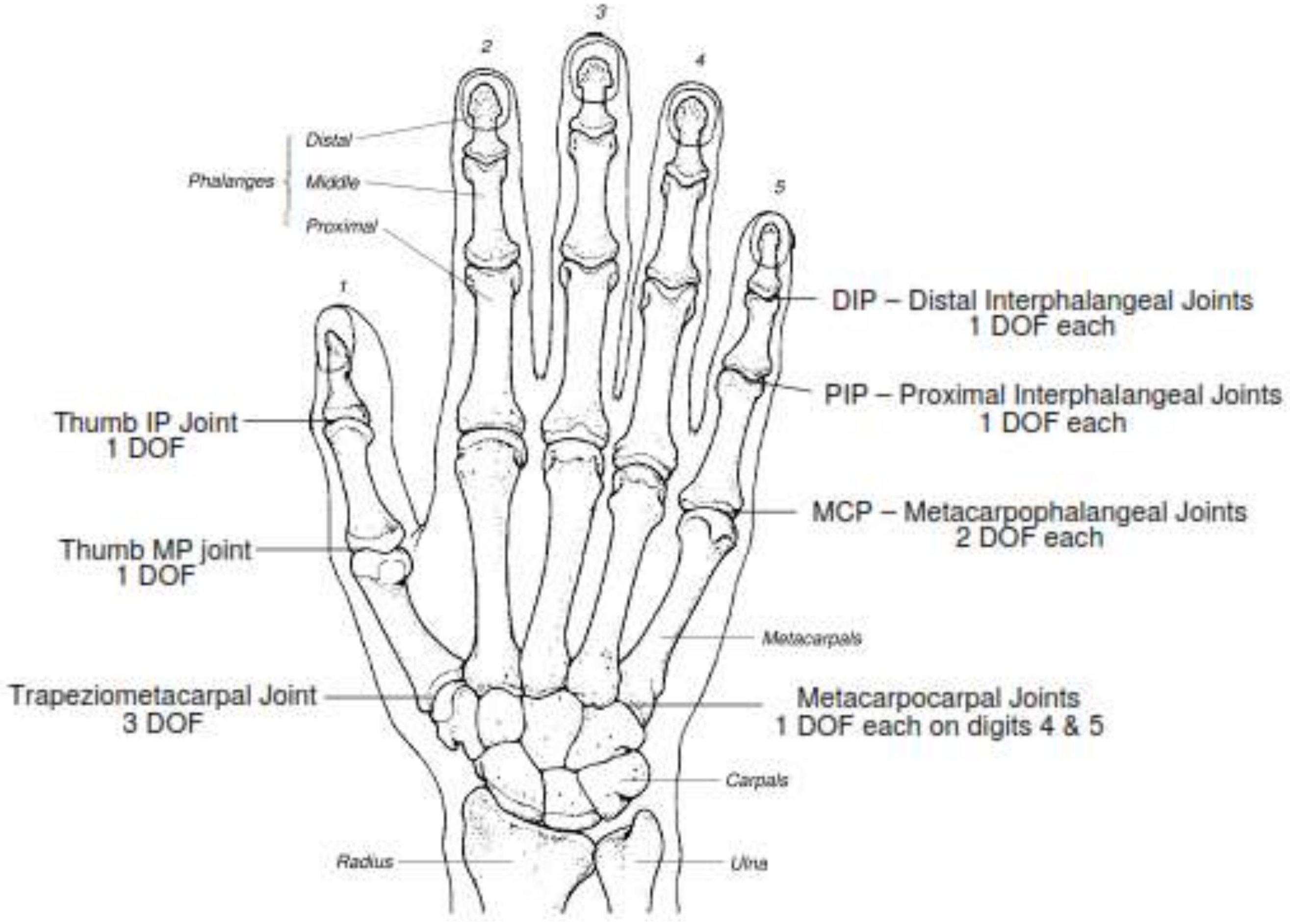
Extreme Examples of "Intrusive" I/O Devices

Degrees of Freedom

- Definition Degrees of Freedom (DOFs) :=
  number of free variables describing the state of a system
- Quiz about DOFs:
  - How many DOFs does our wrist joint have?
  - The head?
  - One human arm?
  - Our hand?





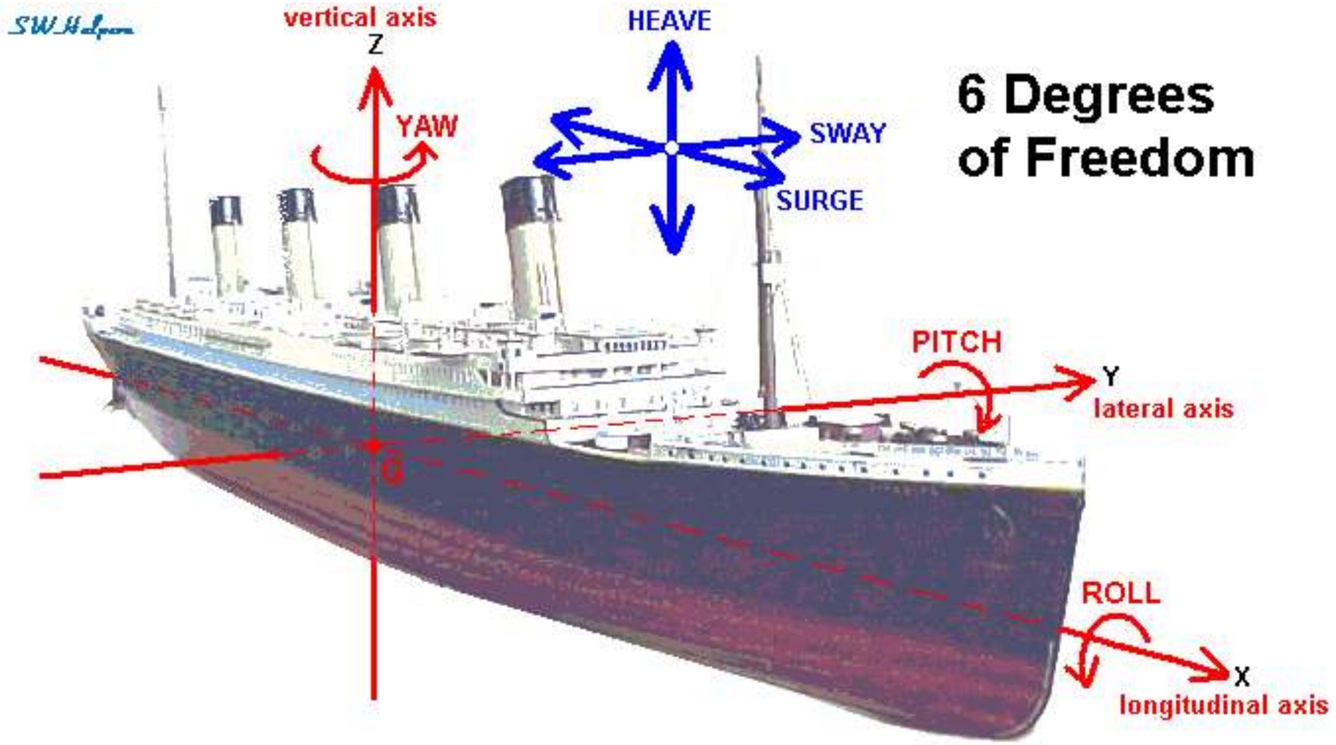


How Many DoF's Does Each Thing Have?

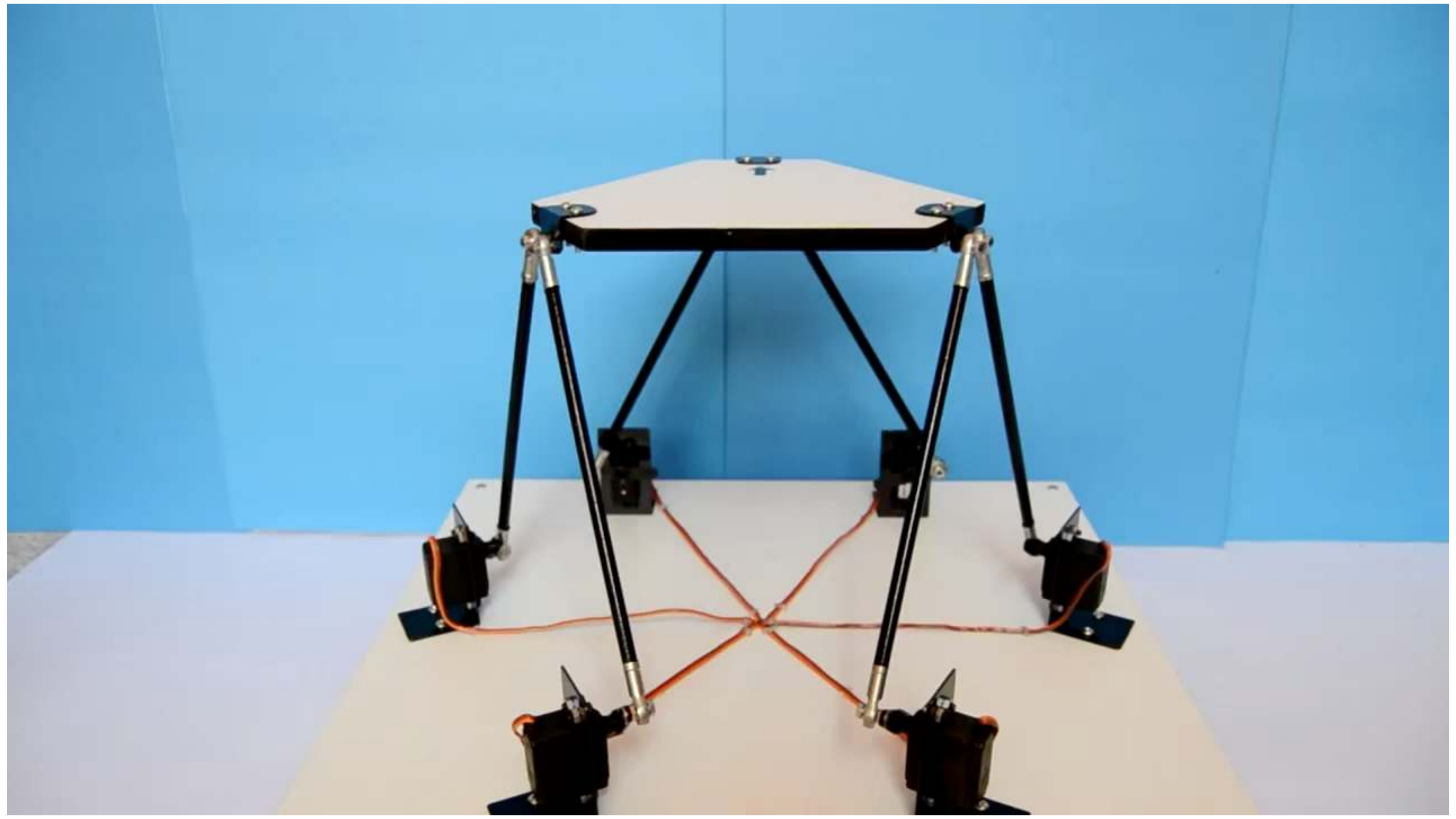
Please don't spoil by "look-ahead"!

https://www.menti.com/zkg58wx3dc

- A ship's pose

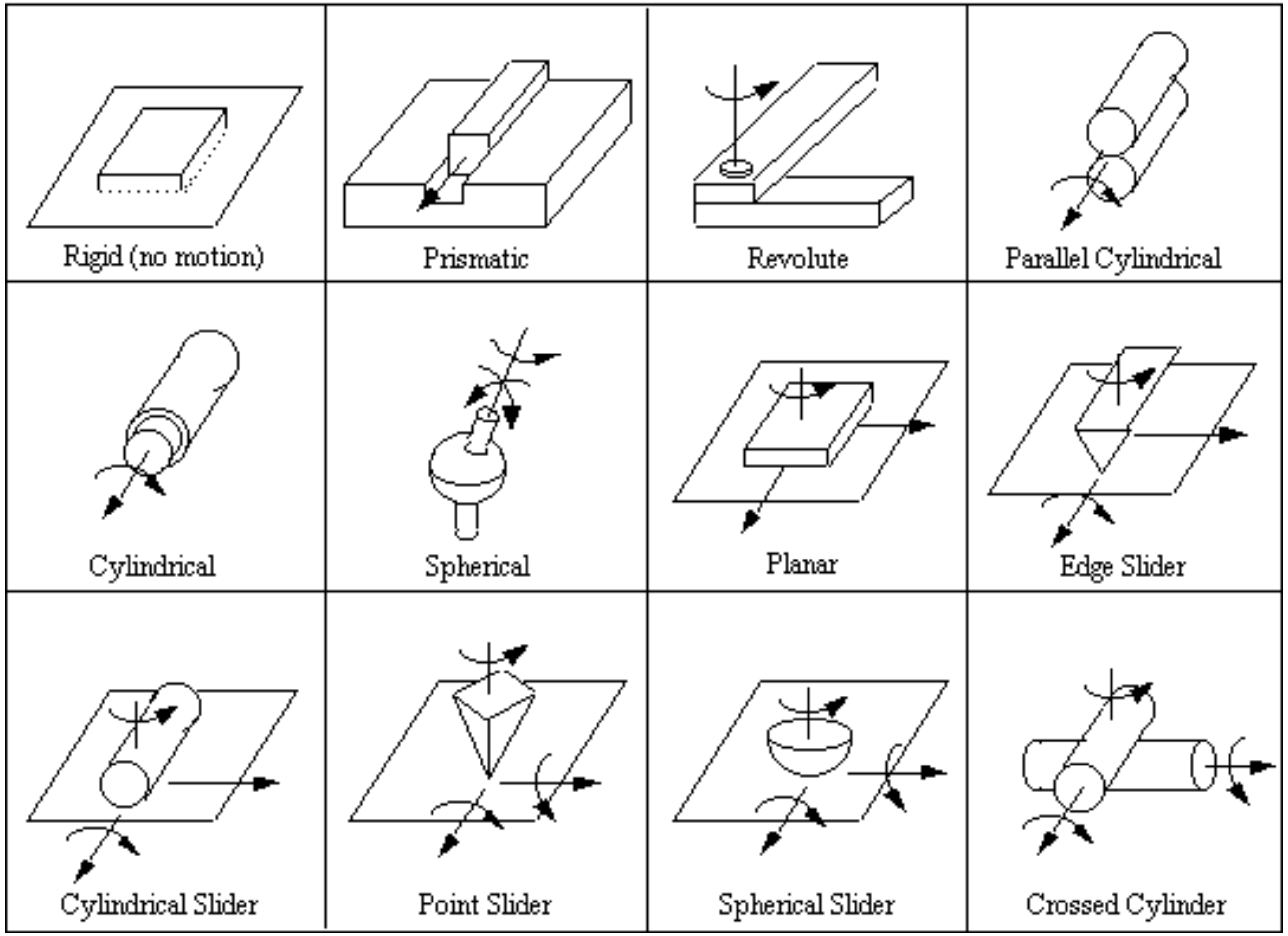


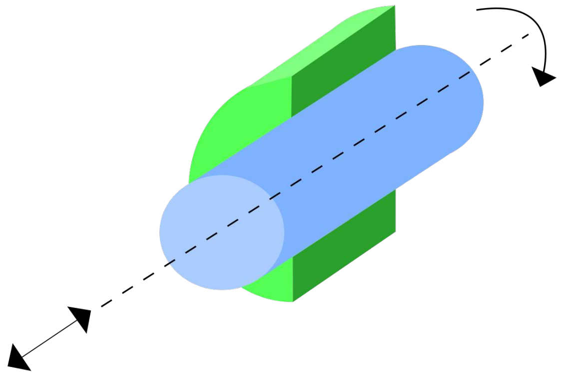
- The Stewart motion platform
  - How many independent DOFs?
  - How many dependent DOFs?



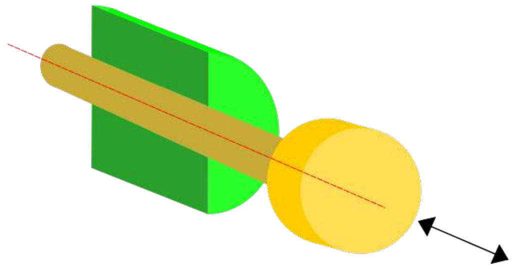
Video

DOF's in the main kinematic joints/pairs

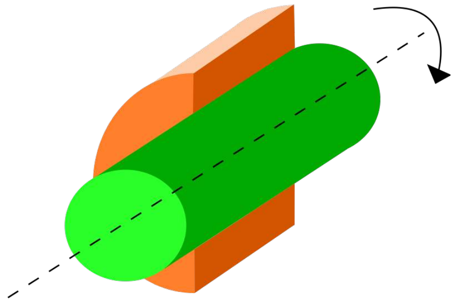




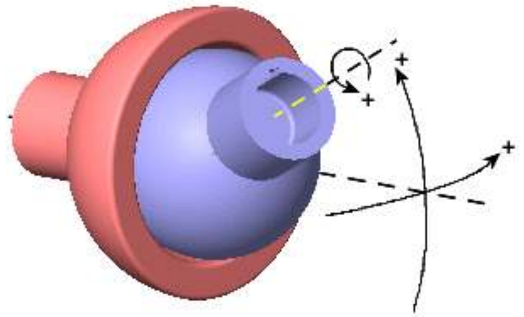
Cylindrical joint



Prismatic joint (slider)

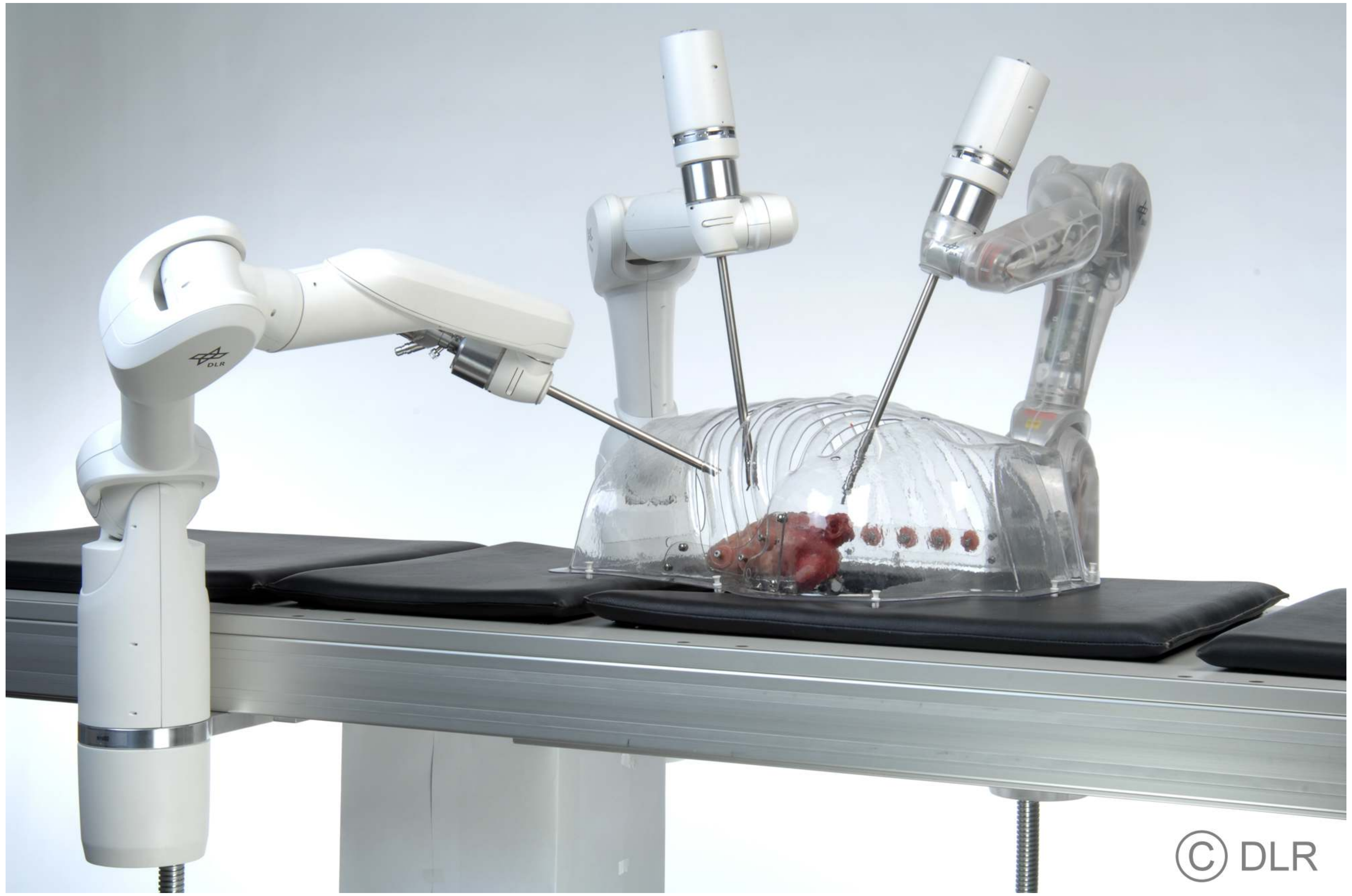


Revolute joint

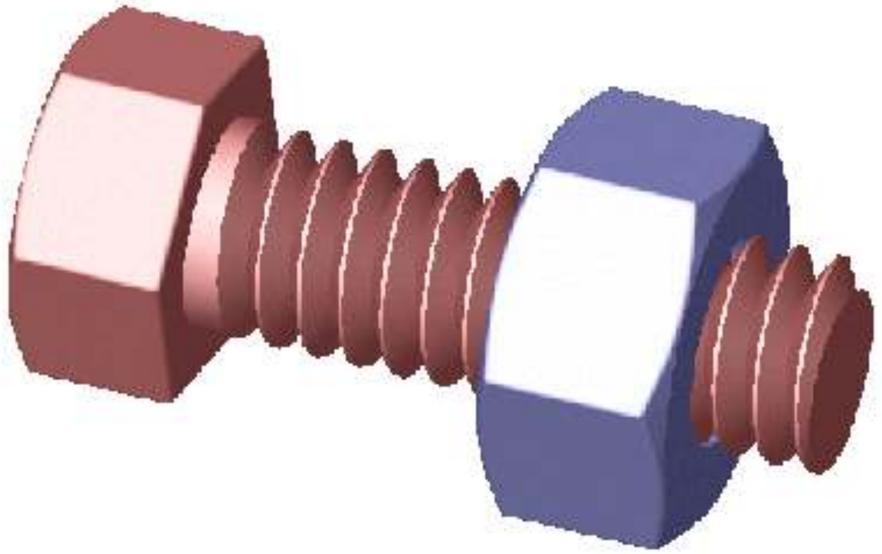


Spherical joint

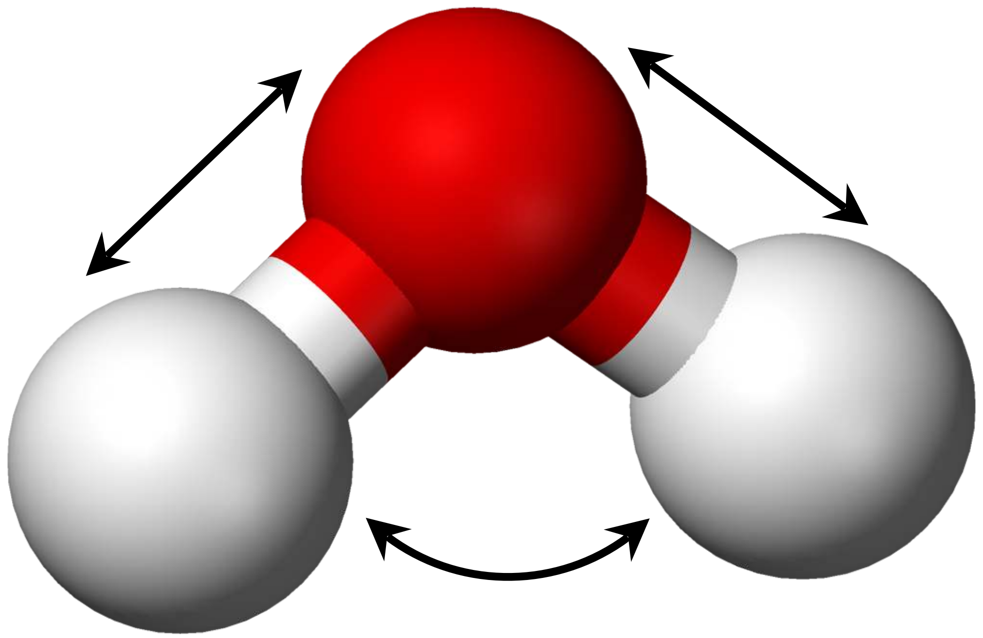
- How many independent DOFs in one robot arm of this surgery robot?



- The screw joint:
  - Joint with coupled rotational and translational degrees of freedom
  - One independent DOF, and one dependent DOF



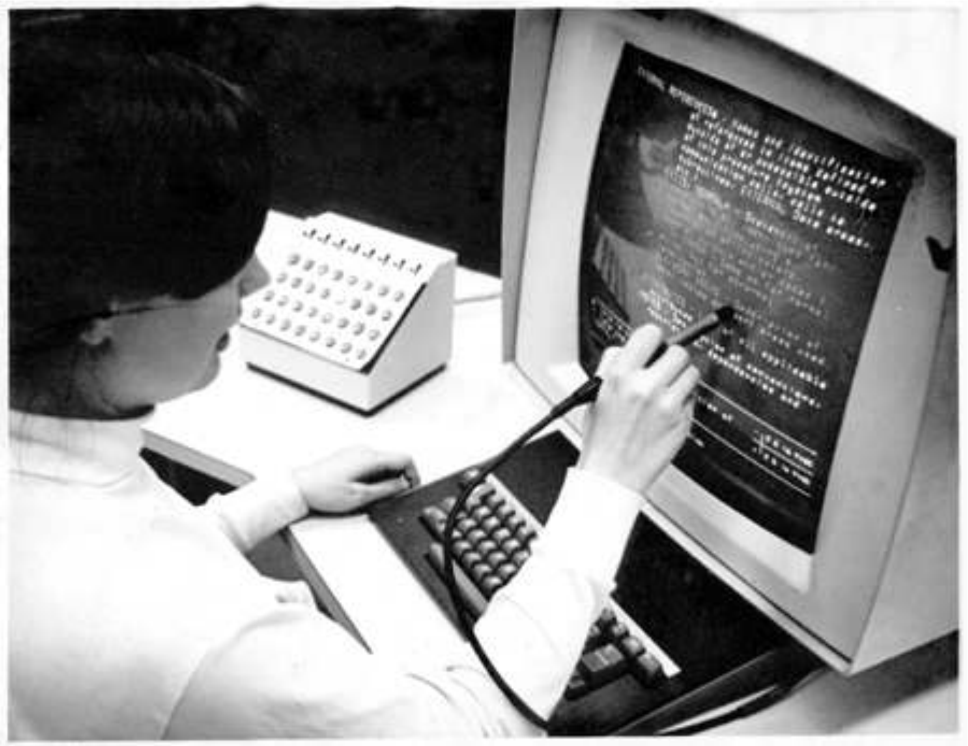
- The internal DOF's of a water molecule:




A fish about
to have
no degrees
of freedom
!!

Classical Input Devices

- Mouse:
  - Precise, inexpensive
  - Only 2D, input of orientations is cumbersome
- Drawing tablet:
  - Precise, very well suited for … drawing
  - 2D, input of orientations is virtually impossible
- Light pen (early version of touch/tablet screen)



The Virtual Trackball

- Interaction task: rotate an object around an arbitrary axis
- Real trackballs can provide 3 DOF rotations
- Interaction device: classic 2D mouse
- Problem: how to enter orientations with a mouse?

The Algorithm

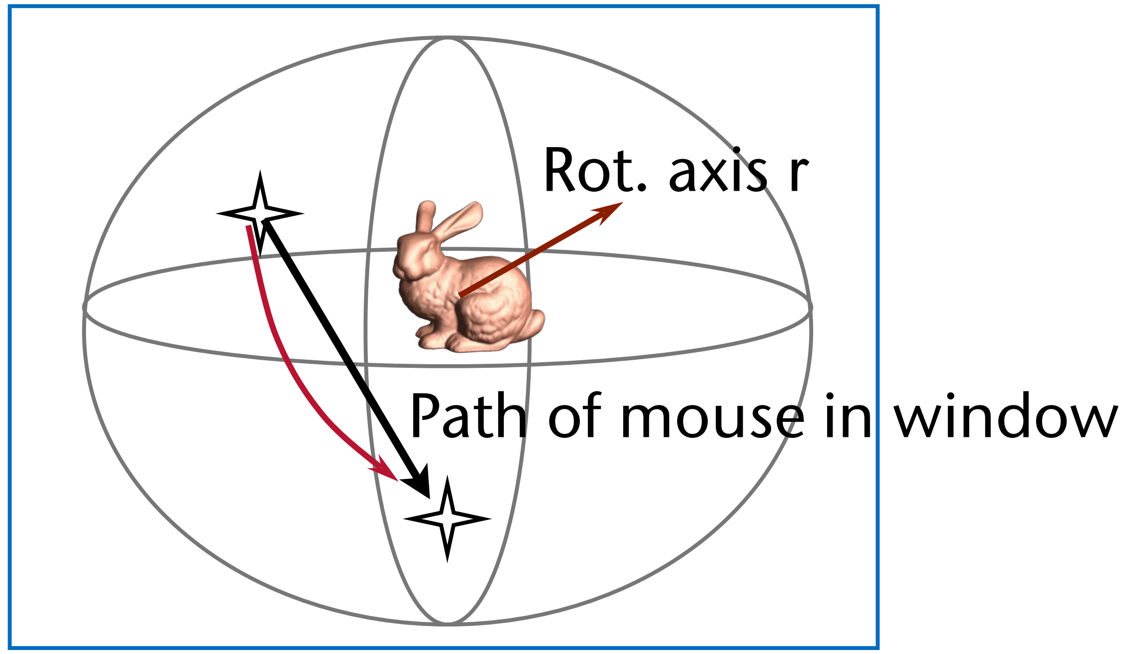
- Approach:
  - Conceptually, put a (virtual) sphere around the object
  - The sphere can rotate only abouts ist center
  - With the mouse, you drag points on the surface of the sphere
- Given: 2D points start = (x_1,y_1), end = (x_2,y_2)
- Wanted: rotation axis **r**
- Computation:

1. Derive 3D points

$$\mathbf{p}_i = (x_i, y_i, z_i) \qquad z_i = \sqrt{1-(x_i^2 + y_i^2)}$$

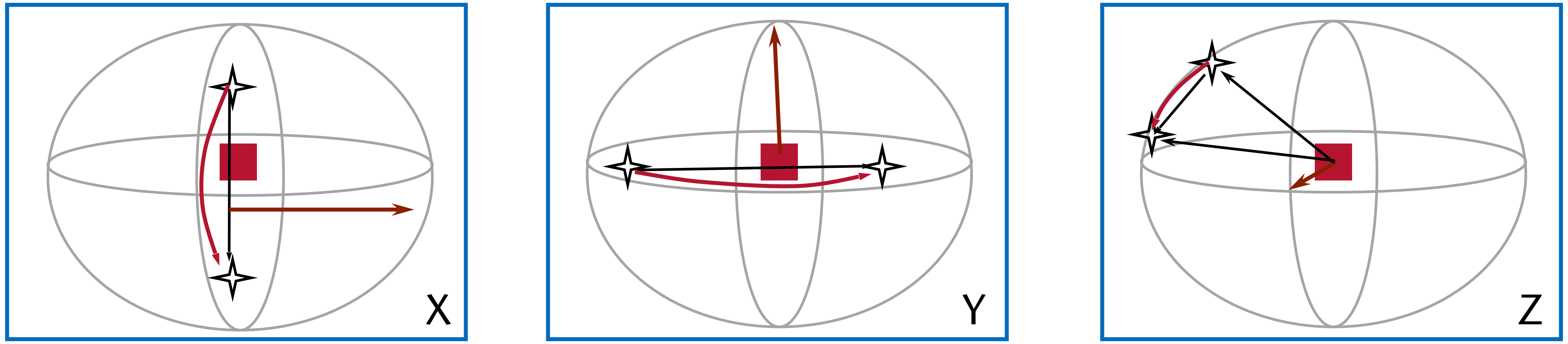
2. Rotation axis

$$\mathbf{r} = \mathbf{p}_1 \times \mathbf{p}_2$$



Conceptual path of the dragged point on the sphere = Segment of a great circle

- If $\mathbf{p}_1$ = first mouse click, $\mathbf{p}_2$ = current mouse pos. → not intuitive
- If $\mathbf{p}_1$ = mouse pos. as of last frame, $\mathbf{p}_2$ = current mouse pos. → intuitive, but rotation exactly about z-axis impossible



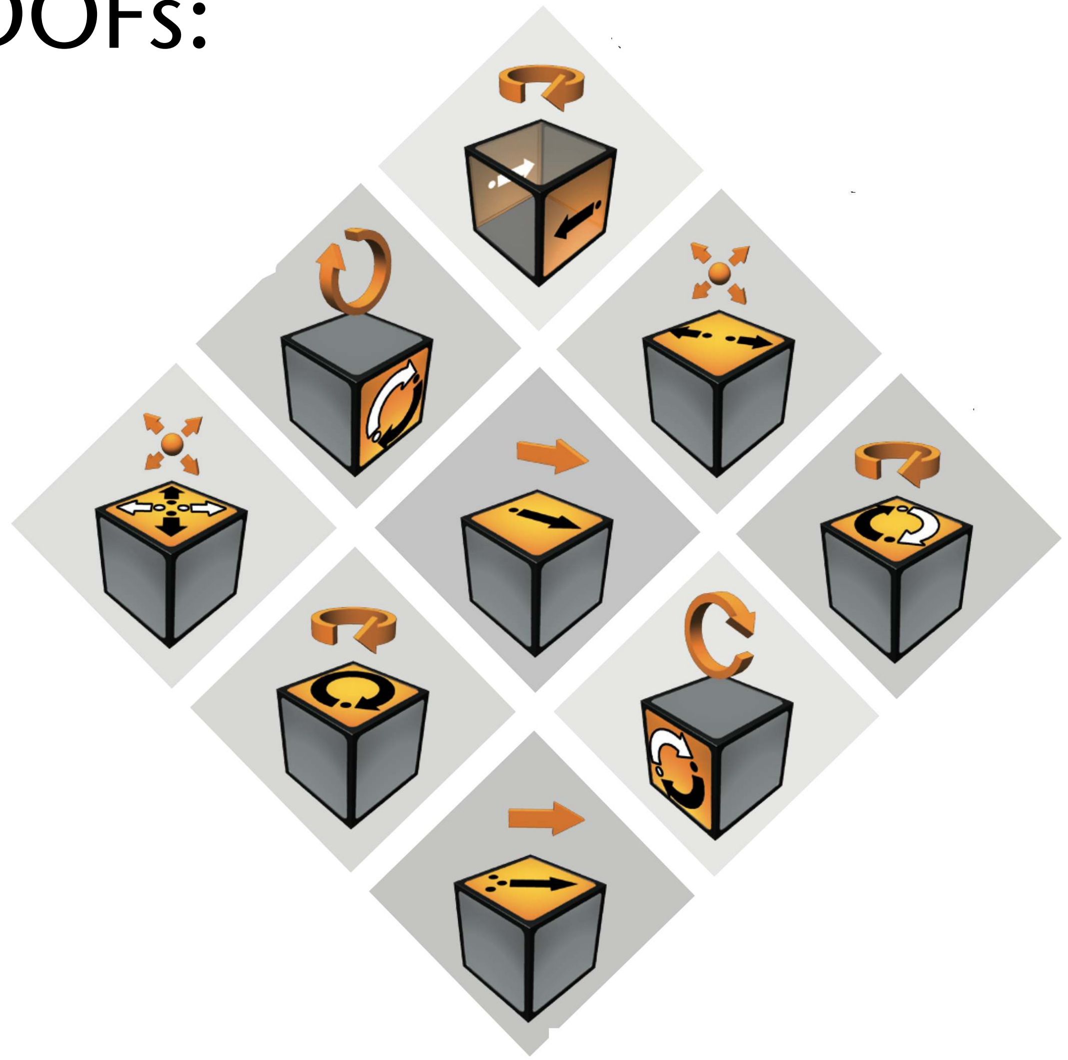
- Improvements / variants:
  - "Spinning trackball": "re-grabbing" the sphere is less often necessary
  - "Snapping": allows precise rotation around world/object coord. axes
  - In case $\mathbf{p}_2$ leaves the ellipse → could use different 3D surface that can be attached continuously to sphere (e.g., hyperboloid)

Remarks

- Rotation axis **r** is given in the camera coordinate frame!
  - You need it in the world frame or object frame
    - Depending on whether the rotation is to be applied to the object before or after all other transformations
- Warning: with variant 2 ("incremental trackball"), a lot of small rotations need to be accumulated! (one per frame) $\longrightarrow$ consider numerical robustness and drift

Cubtile

- 5 multi-touch surfaces arranged in a cube
- Cube acts as proxy for obj
- Bonus: very neat illumination ☺
- Number of DOFs:



Isotonic vs Isometric Sensing

- Definition isotonic sensing device:
  The user can move the device (or just that DoF) all the way without changing muscle tone.
  - Isotonic = "same muscle tone (tension) during contraction"
  - In practice: input value is proportional to distance from origin, which, ideally, does not require force

- Definition isometric sensing device:
  The device (or just that DoF) does not move when the user pushes/pulls the device
  - This is only true for an ideal device
  - In practice: input value is proportional to force
  - Isometric = "same muscle metric (length) during contraction"

Example for Isometric Device: Spacemouse

Example for Isotonic Device: Control Action Table

Rotations: controlled by an isotonic sensing mode (cyclic)

Translations: controlled by an isometric sensing mode (infinite)

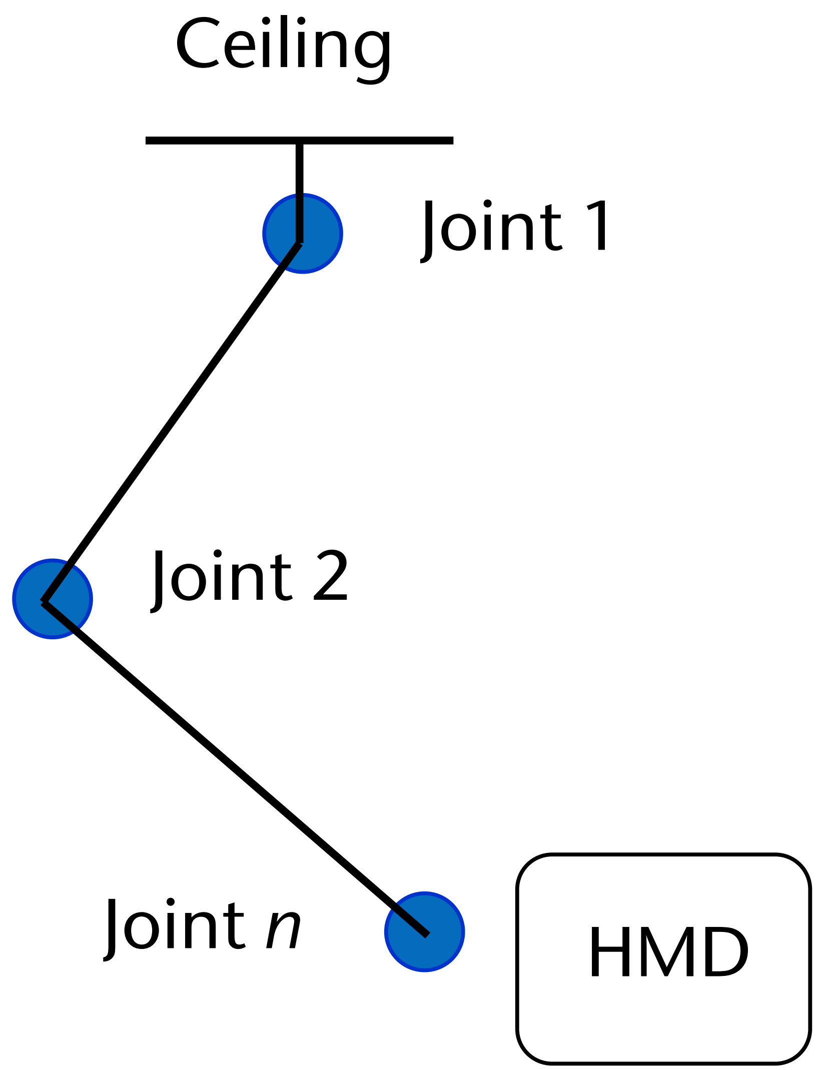
Tracking the User

- Task: determine "where is X of the users?"
    - X = head, hand, eyes, feet, whole body, ...
- Requirements:
    - *Non-intrusive*
    - High precision (1 mm)
    - Low latency (1 msec)
    - High *update rate* (100 Hz)
    - Works in all environments and conditions
    - Large working volume
- Doesn't exist (yet?)!

- Technologies for tracking:
  - Mechanical
  - Electro-magnetic
  - Acustic (ultra sound)
  - Optical
  - Computer vision-based
  - Inertia sensors
  - Laser
  - GPS
  - Hybrids

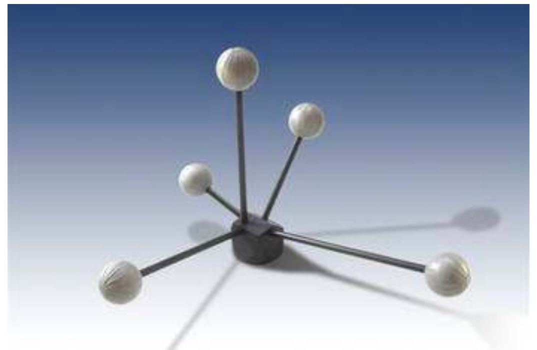
Mechanical

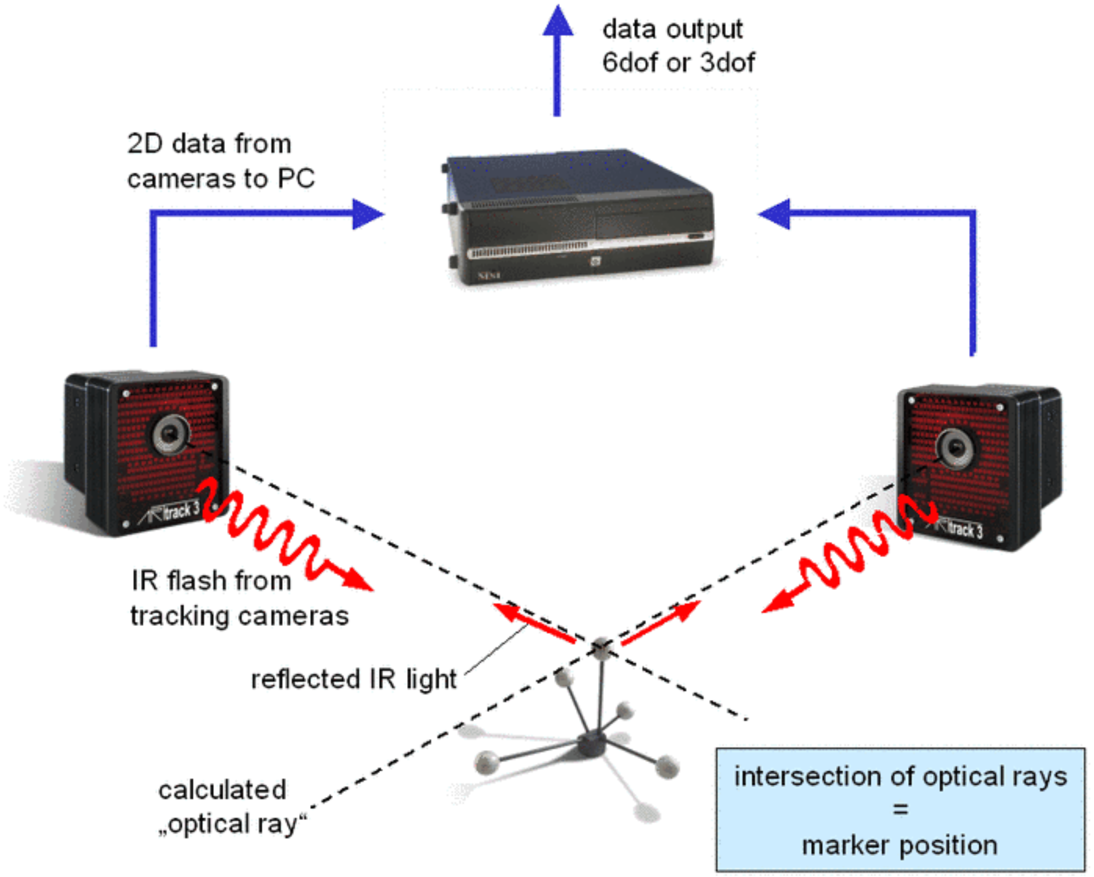
- Advantages:
  - Precision
  - Low Latency
  - No distortion by metal in environment
- Disadvantages:
  - Uncomfortable
  - Working volume
  - "Dead" zones
  - Intrusion
  - Calibration
  - Inertia b/c of mass

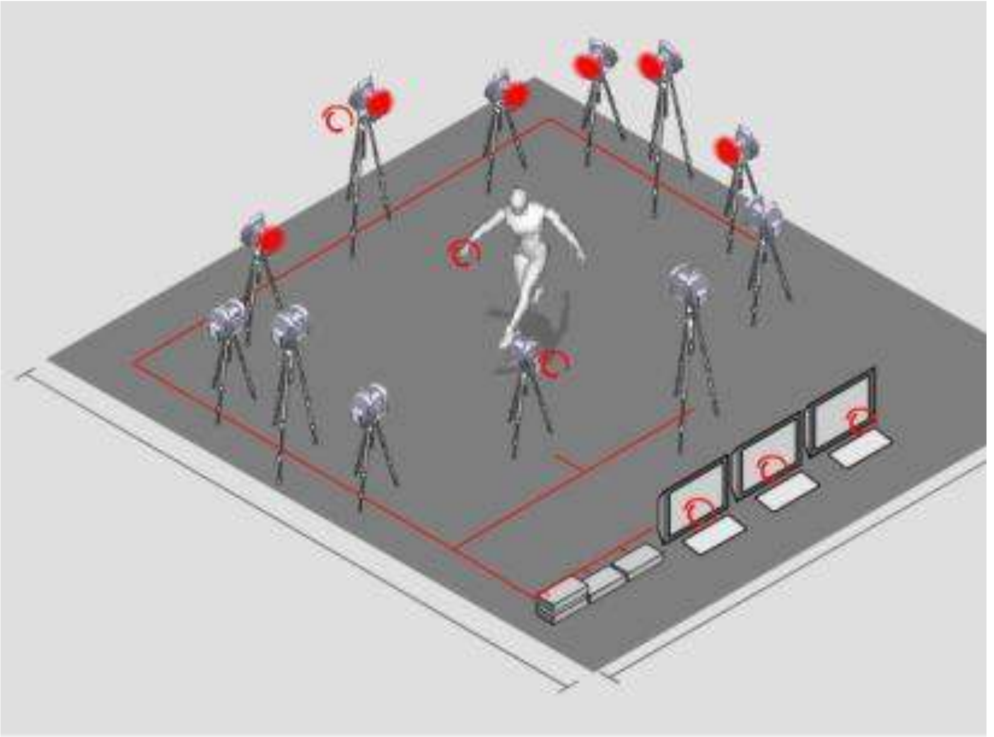


Optical Tracking

- Idea: track highly reflective markers using IR cameras
- 1 marker ⟶ 3D position
  - By way of triangulation
- ≥3 markers (a "rigid body")
  ⟶ position and orientation
- Standard technology for body tracking in animation studios and for game development
  - Motion capturing (MoCap)



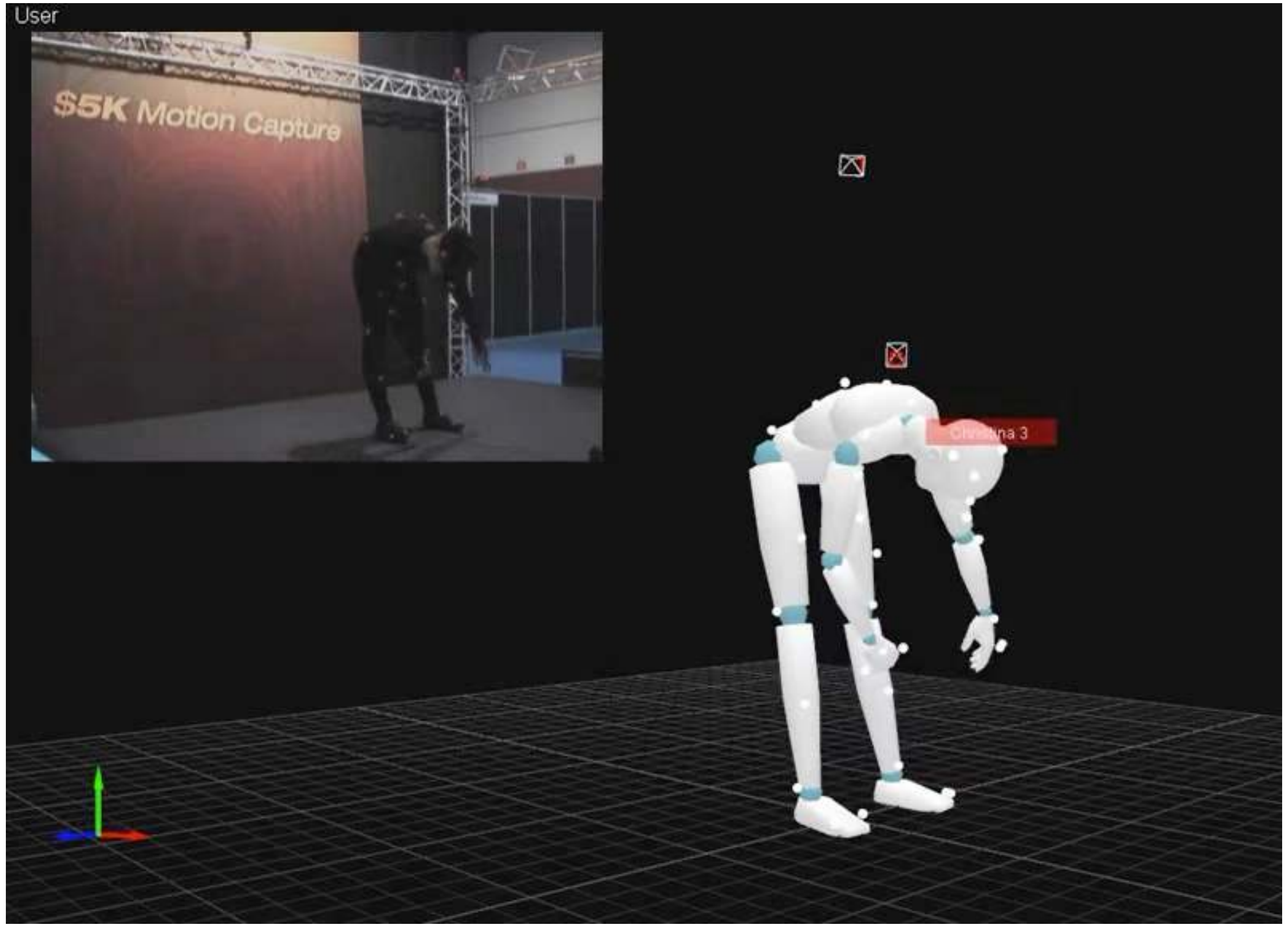




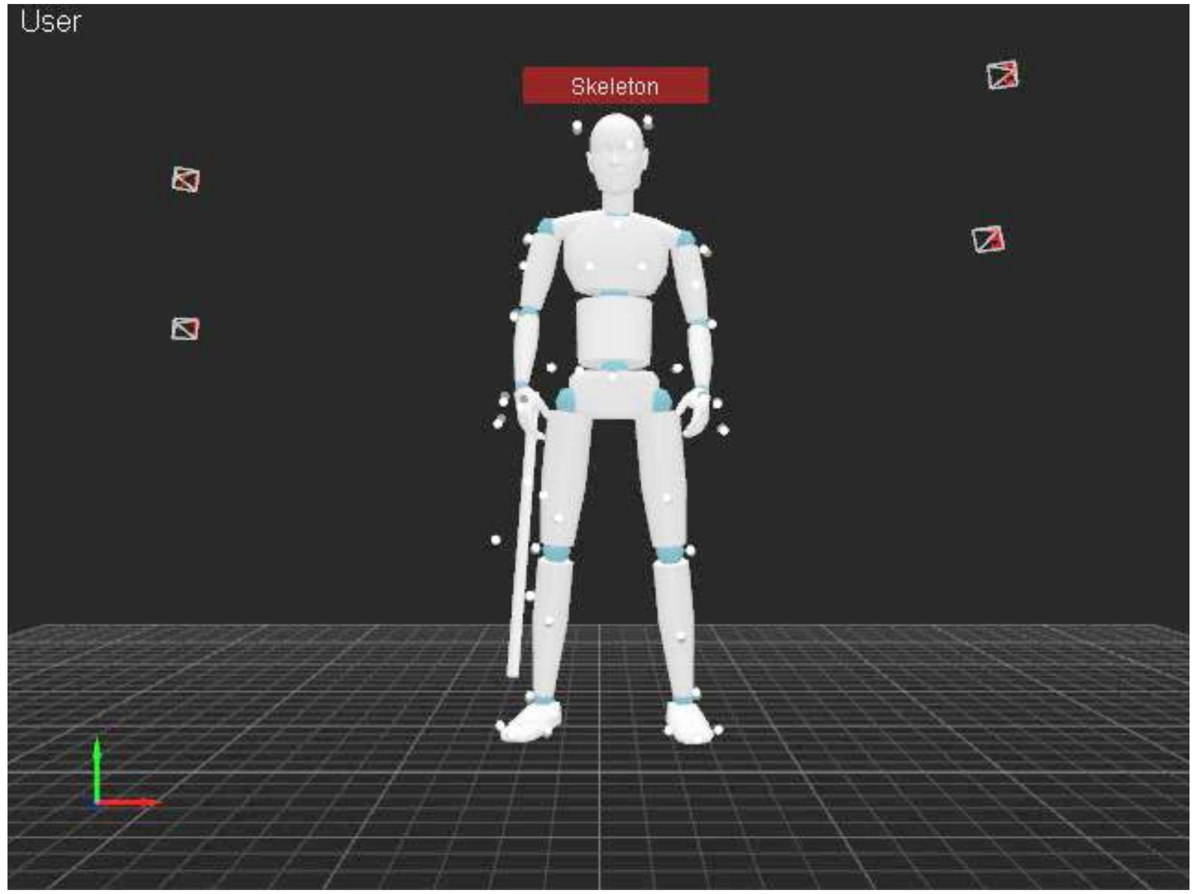
Some Use Cases

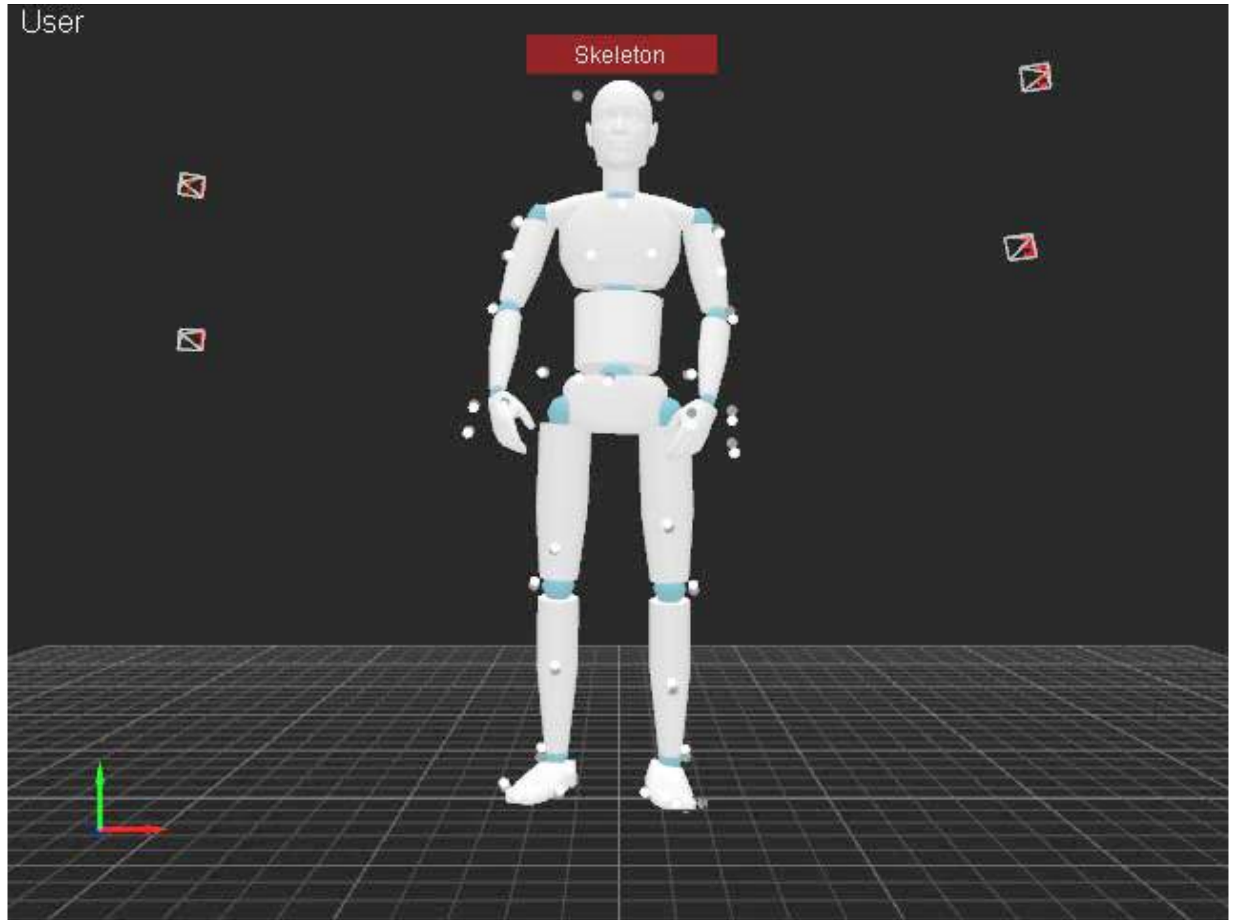
- Advantages:
  - Free movement for users / actors
  - Large working volume
  - High *sampling rate* (typically 120-250 Hz)
  - *Facial animation* is possible, too
- Disadvantages:
  - *Line-of-sight* needed (mitigation: lots of cameras)
  - Price ($6,000 - $100,000)

Fluid Images



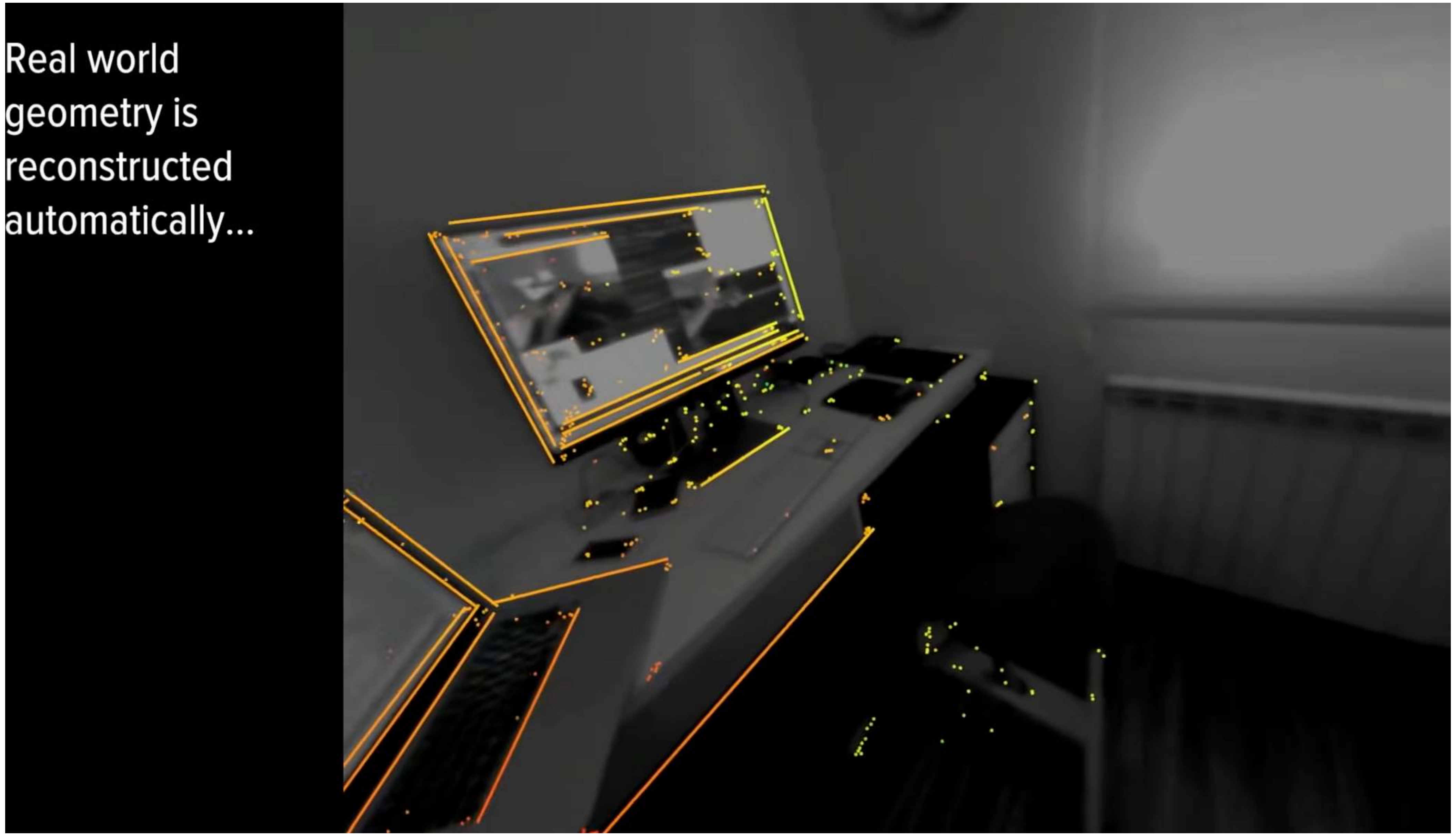
NaturalPoint (OptiTrack)


User
Skeleton

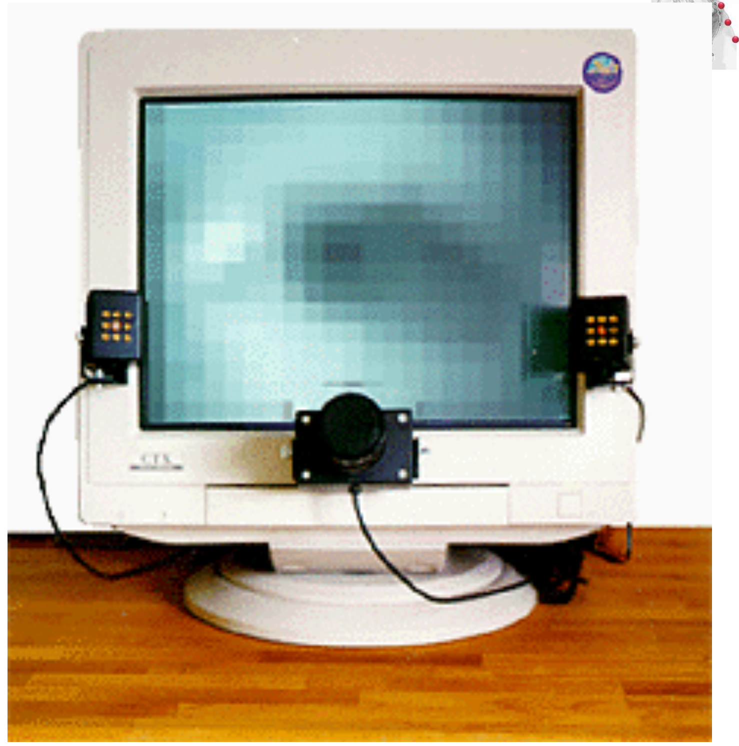

User
Skeleton

"Inside Out" Tracking

- Approach:
  - Camera(s) integrated in HMD look "out" into physical environment
  - Perform SLAM (Simultaneous Localization and Mapping)
  - Needs "reset" (registration) with "zero" position/orientation at beginning
- Advantages:
  - No additional hardware (e.g. "lighthouses")
  - Potentially unlimited working space
- Examples: HoloLens, Oculus Quest, HP Reverb G2



Eye Tracking



- Differentiation:
  - Where are the user's eyes? → eye tracking
  - In which direction does the user look? → eye gaze tracking
- Applications:
  - Head tracking
  - Controlling LODs, foveated rendering
  - Autostereo monitors
- Problems:
  - Precision
  - Sometimes additional hardware is needed

Acoustic Tracking

- Similar to sonar:
  - 1 ultra sound source
  - 3 receivers (for 3 DOFs)
  - Travel time → position
- Advantages:
  - Very inexpensive
- Disadvantages:
  - Echos
  - *Line-of-sight* prerequisite
  - 3 transmitters needed for 6 DOFs
  - Small range

Inertia Sensors

- Measures acceleration in one direction
- Advantages:
  - No transmitter necessary
  - Very small sensors
- Disavantages:
  - Drift
- Sometimes combined with other tracking technologies to compensate for drift (e.g., GPS)

Laser Tracker

- Measures just distance / position
- So far being used only in manufacturing industries (CNC machines)

Electromagnetic Tracking

- Transmitter =
  - 3 orthogonal coils (using 3 different frequencies)
  - Emit 3 orthogonal electromagnetic fields
- Sensor = receiver =
  - 3 orthogonal coils, too
  - Receive 9 signals in total
- Phase shifts between transmitted and receive signal → distance
- Strength of the 9 different signals → orientation
- Advantages:
  - Small sensors; Working volume = 3 m (or more)
- Disadvantages:
  - Tethering (cables)
  - Metal in environment has severe impact in field distortions
  - Noise

Characteristics of Tracking Systems in Gerenal

1. # DOFs
2. Precision, drift, replicability
3. Update rate, latency
4. Noise
5. Additional buttons
6. *Ease-of-use*, *tethering* (=cables) – *unintrusiveness*!
7. Working volume
8. Price

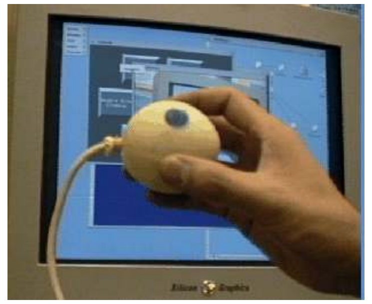
3D Pointers / Stylus / "Controllers"

- Analogue to 2D mouse
- Hardware = tracker with buttons
  - Sometimes with additional joystick, etc.
- Names: *flying mouse*, *flying joystick*, *wand* (= Stab), *bone*, *fly-stick*, etc...
- Advantage: physical object induces a strong feeling of presence while grasping a virtual object

Zhai and Milgram's Directness Continuum for Input Devices

Isomorphic input ⟷ **"Magic tools"**

	Position control			Velocity control	Higher order control (e.g., accel.)
	Absolute		Clutching ("Nachfassen")		
	1:1	scaled			
2-DOF example	*Touch screen*	*Drawing tablet*	*Mouse*	*Joystick*	?
6-DOF example	*Tracking the user*	*Finger-ball*	*Cyber-glove*	*Spacemouse*	?

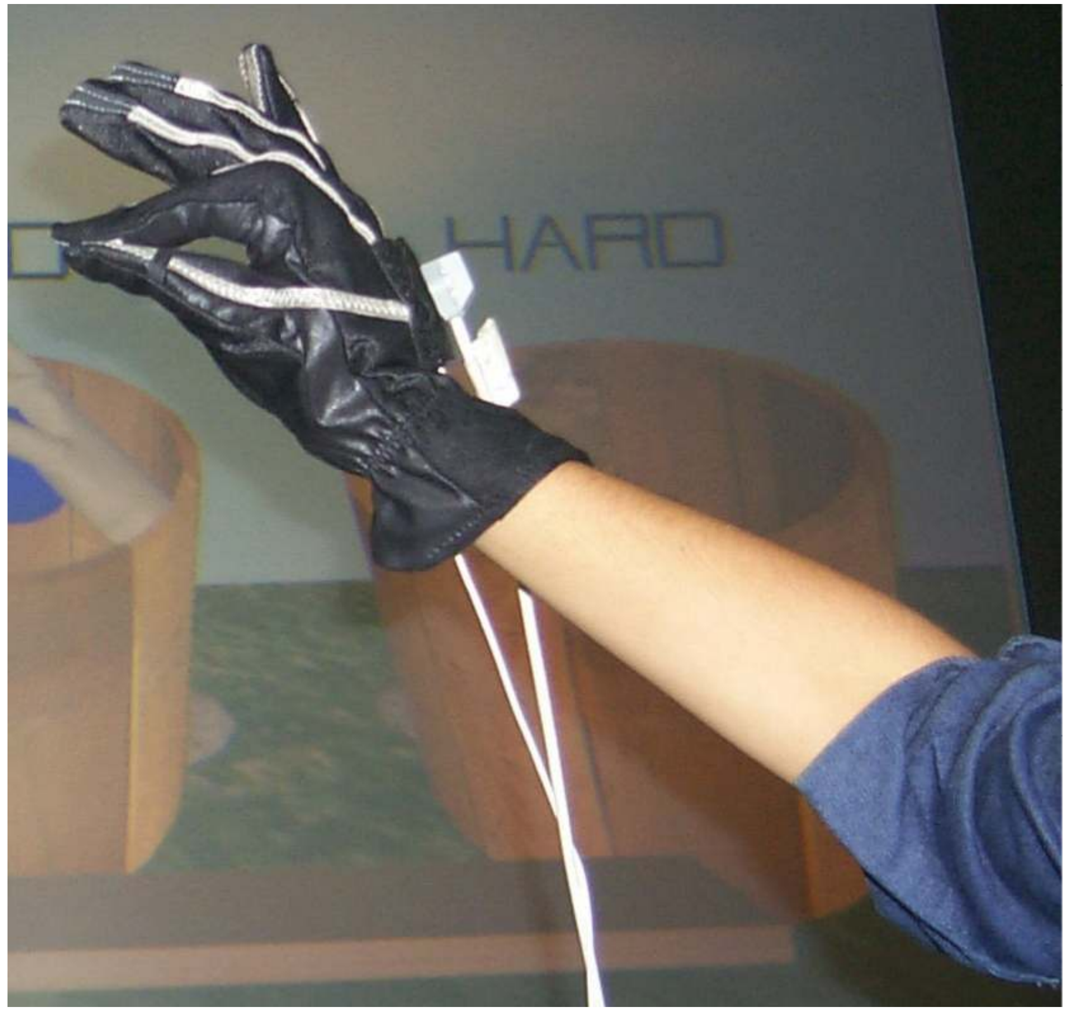


The Dataglove

- "Tracks" fingers of human hand = measures angles of joints
- One of the very early VR devices
- Different numbers of sensors:
  - 12 = 4 (thumb) + 4x2 (2 sensors per finger)
  - 22 = 4 (thumb) + 4x3 (3 sensors per finger) + 3 sensors between fingers + 1 sensor on back of hand (Handrücken)
- Sensor technologies:
  - Glass fibers (not very robust)
  - Bimetallic strips
- Disadvantages:
  - Low precision
  - Glove by and itself (not really accepted)

Variants

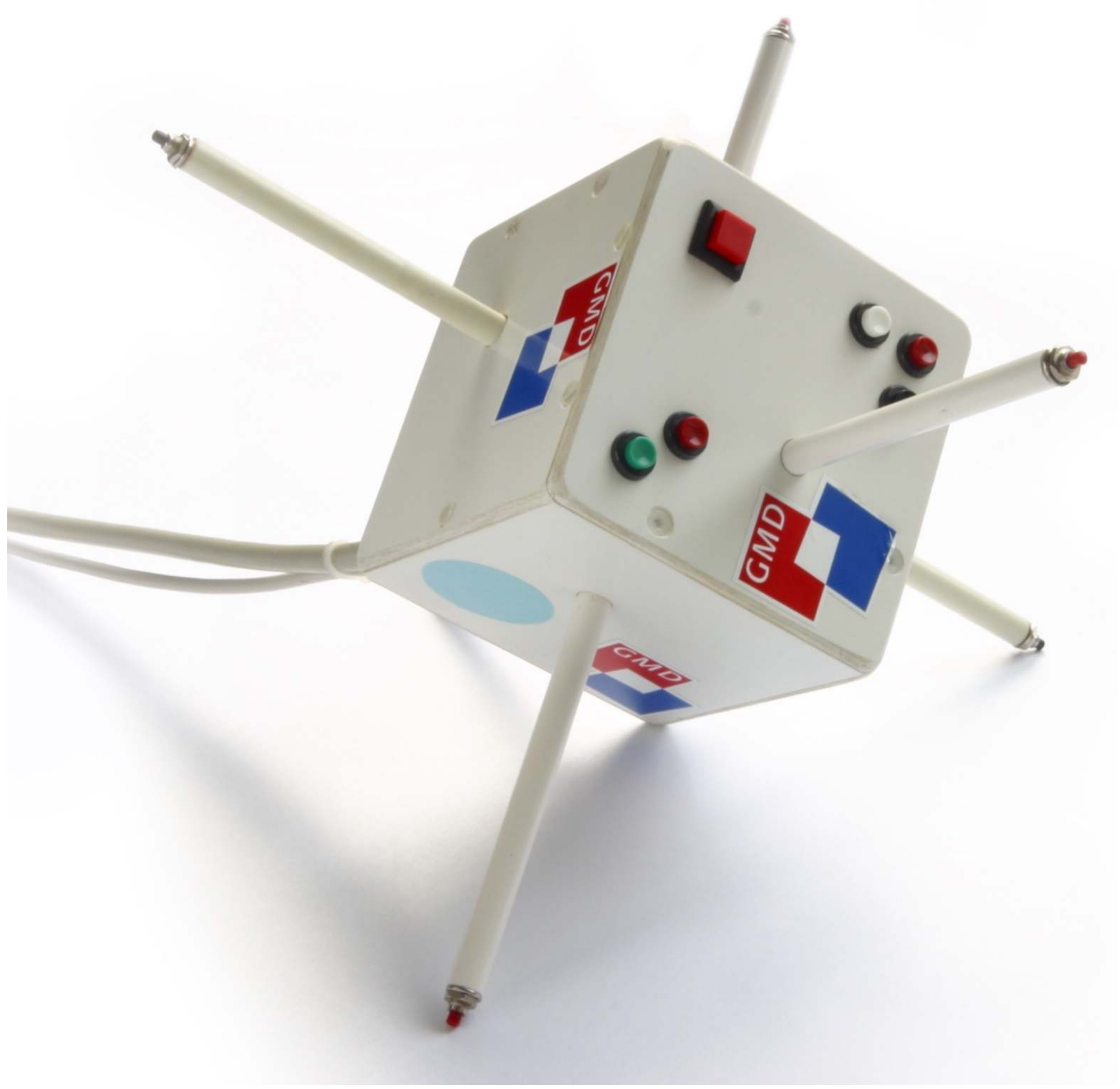
- Pinch glove:
  - No tracking, just detects contact between finger tips $\longrightarrow$ each finger acts like a button
- Usefuly only using 2 tracked pinch gloves; with, though, pretty clever navigation and manipulations can be performed:
  - Grasping and moving
  - Scaling (using *handles* à la Inventor)
  - Will be presented later …
- Disadvantage: a virtual hand cannot be rendered

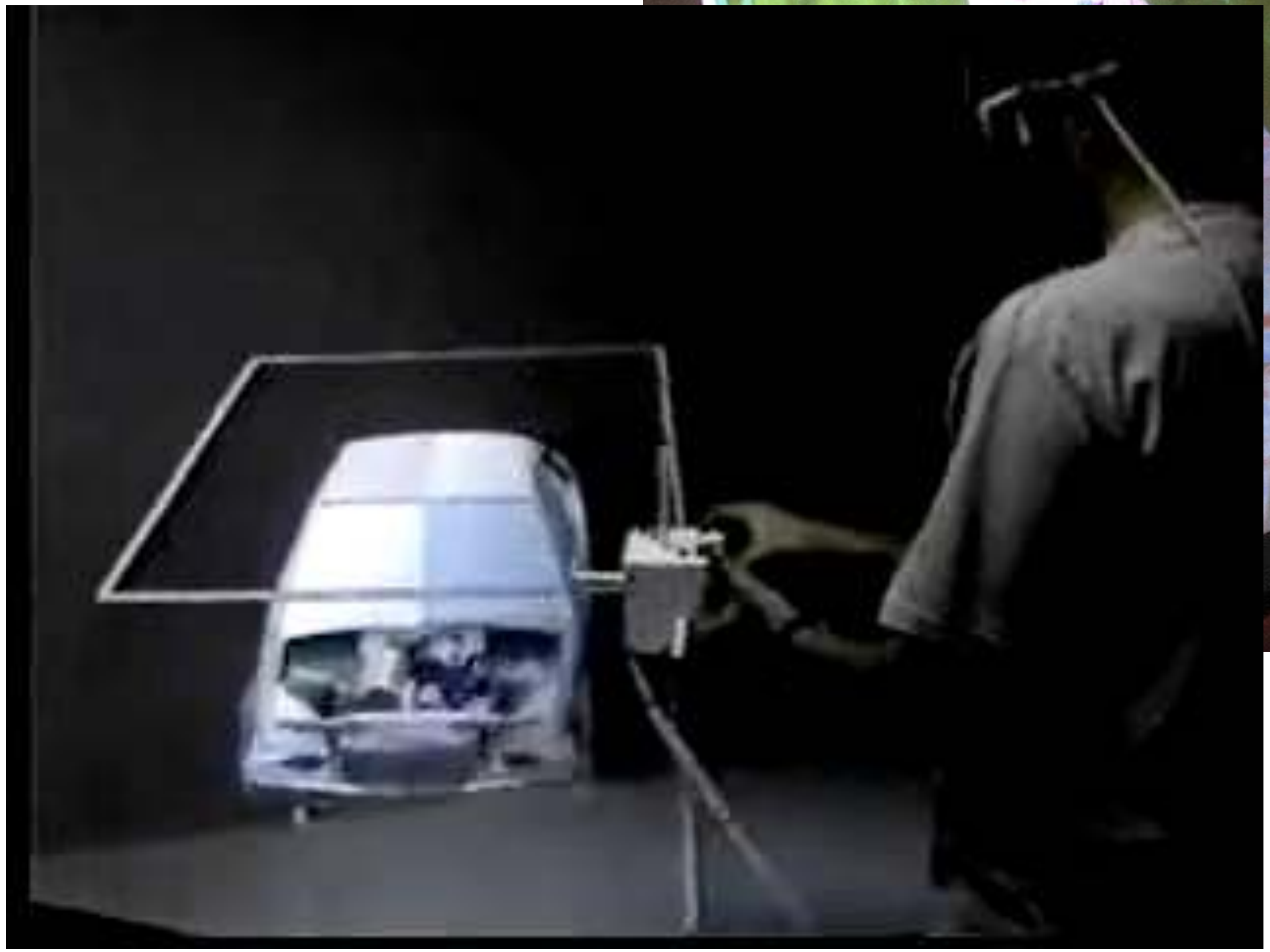


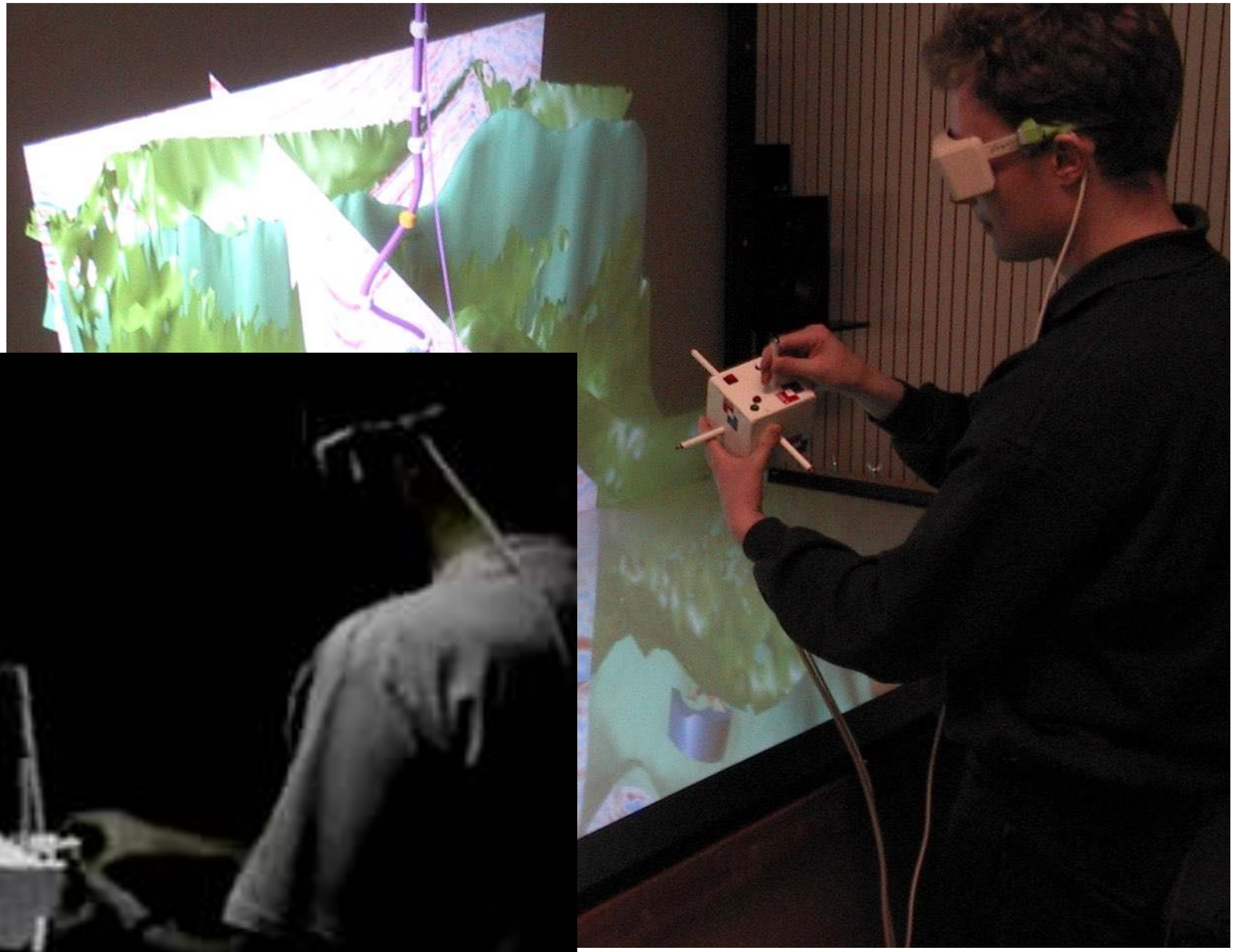
Myo

Other High-Dimensional Input Devices

- Cubic Mouse:
  - Number of DOFs = 9

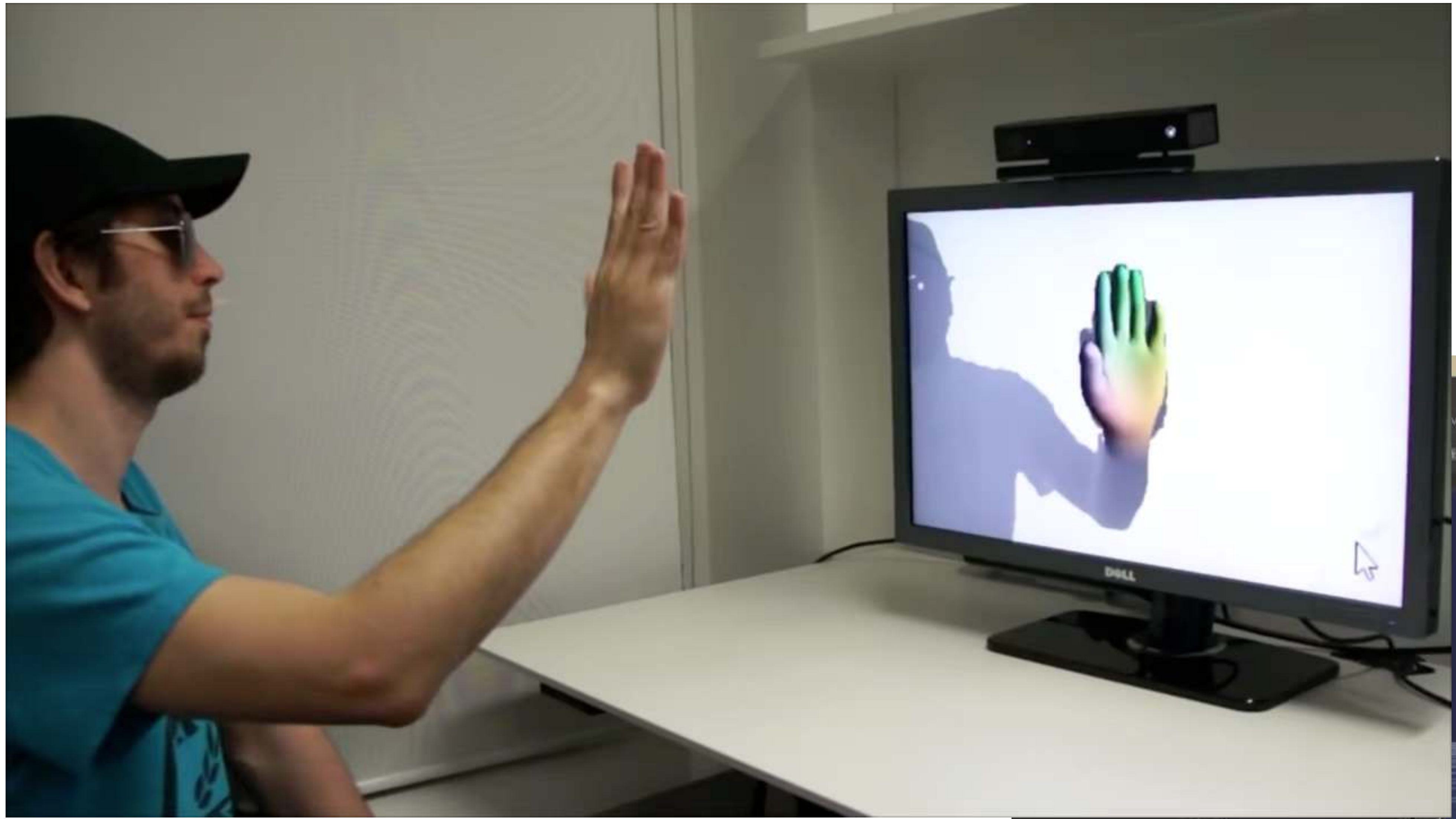


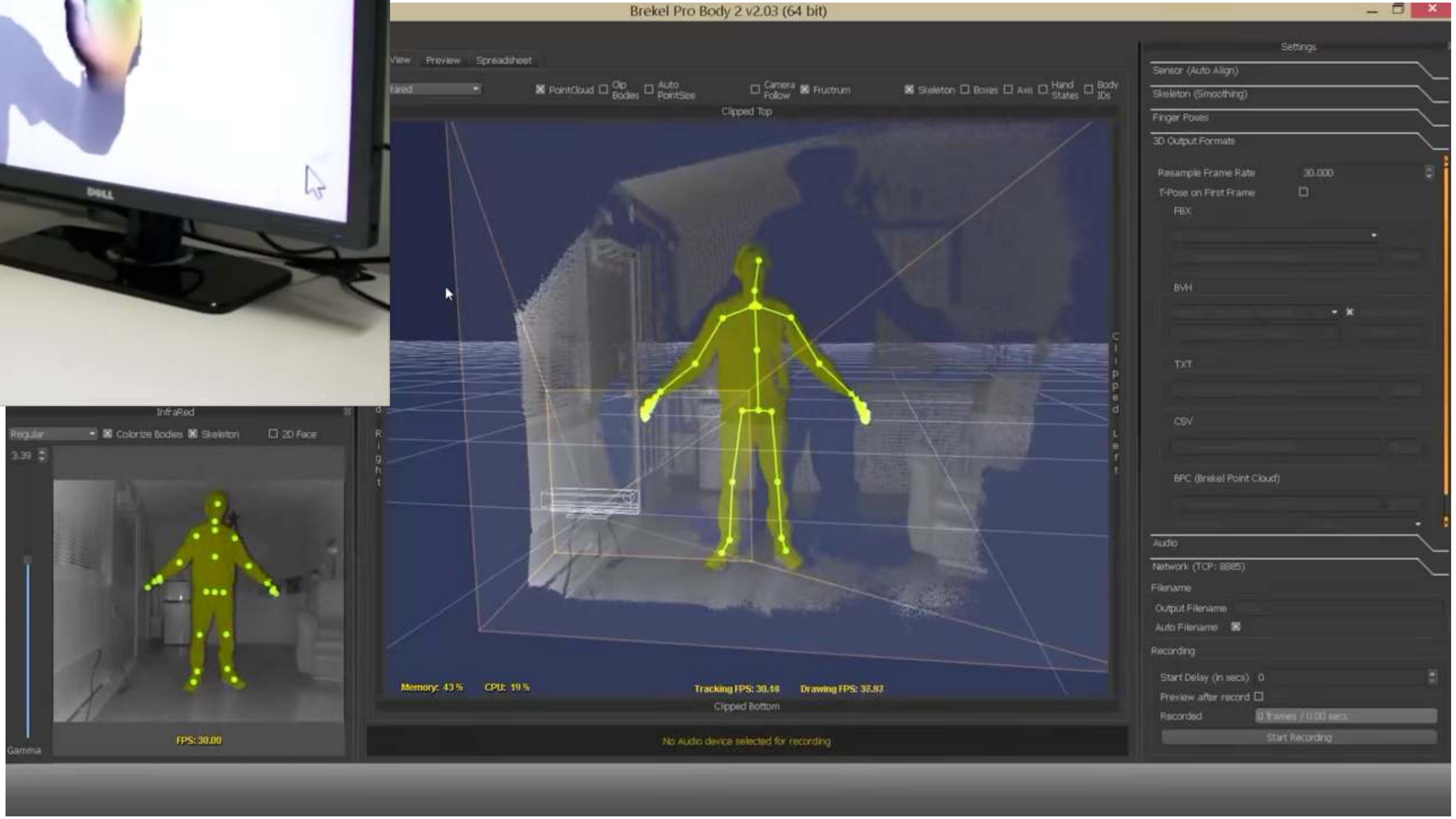




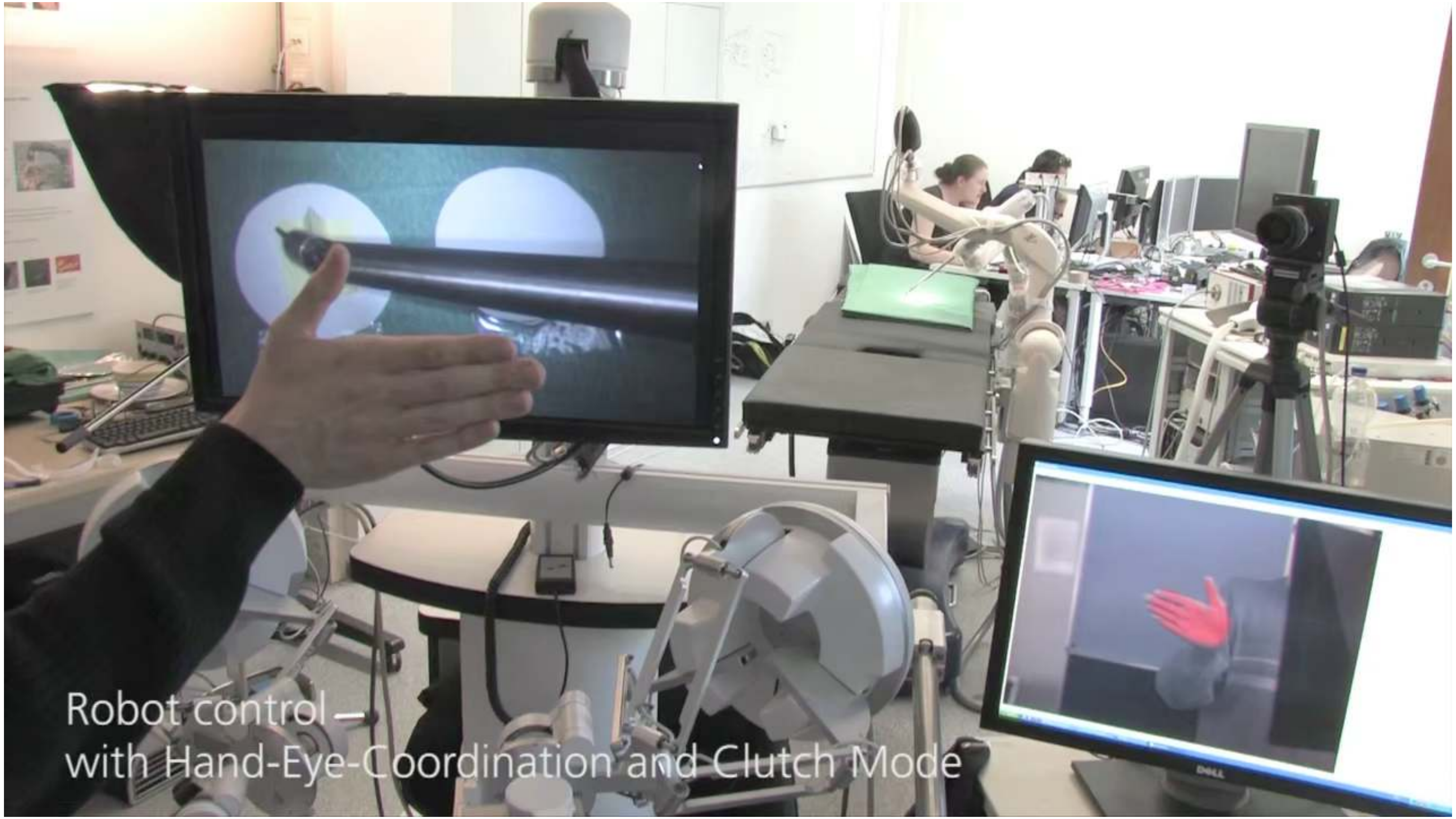
3D Range Sensors

- First consumer device: Microsoft Kinect
- Deliver depth image (range image)
- Lead to so-called natural user interaction (NUI)
  - This vision existed from the beginning of VR




Brekel Pro Body 2 v2.03 (64 bit)
View
Preview
Spreadsheet
Settings
Sensor (Auto Align)
Skeleton (Smoothing)
Finger Poses
3D Output Formats
Resample Frame Rate
T-Pose on First Frame
FBX
BVH
TXT
CSV
BPC (Brekel Point Cloud)
Audio
Network (TCP: 8885)
Filename
Output Filename
Auto Filename
Recording
Start Delay (in secs)
Preview after record
Recorded
Start Recording
Clipped Top
Clipped Bottom
No Audio device selected for recording
InfraRed
Colorize Bodies
Skeleton
2D Face
FPS: 30.00
Gamma

A Possible Application: Control of Micro-Surgery Robots



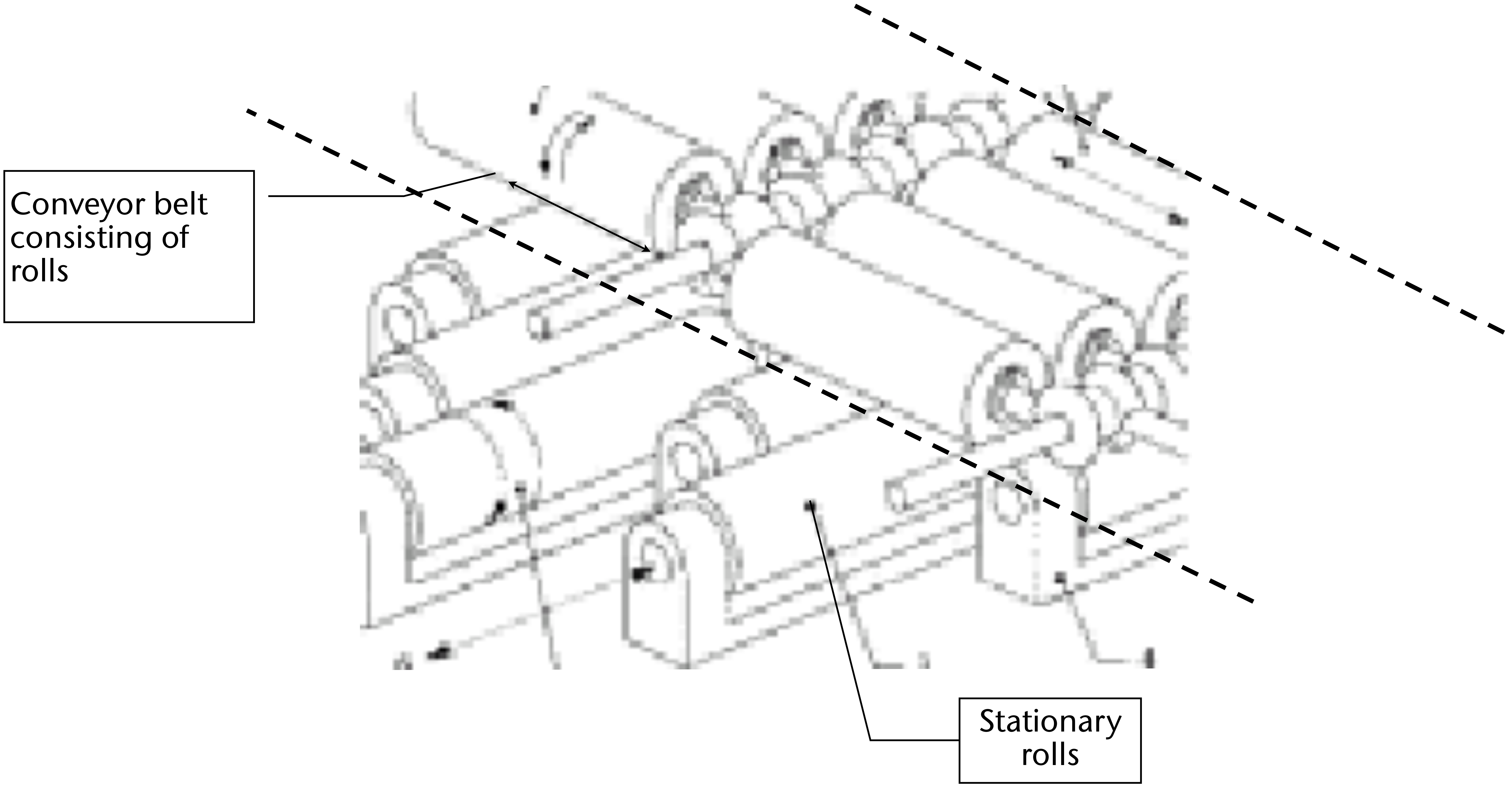
Collaboration with DLR, Institute for Mechatronics, Oberpfaffenhofen

Locomotion Devices

Sarcos, Utah

Omni-Directional Treadmill

Cyberwalk omni-directional treadmill, 2005-2008, TU München


Conveyor belt consisting of rolls
Stationary rolls

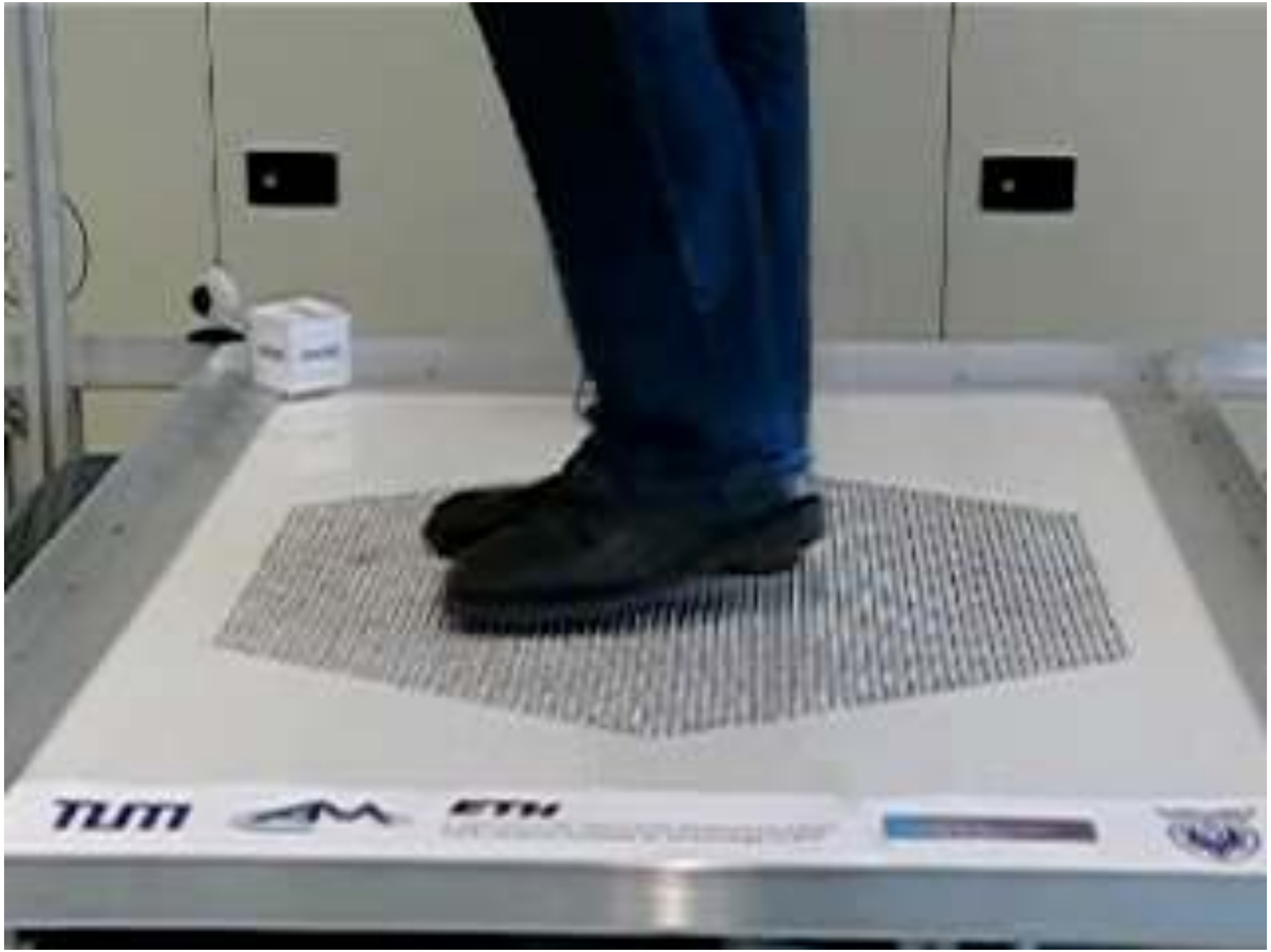
How it Works

Consumer Treadsmill

Virtuix: Omnidirectional treadmill for the home [2013]

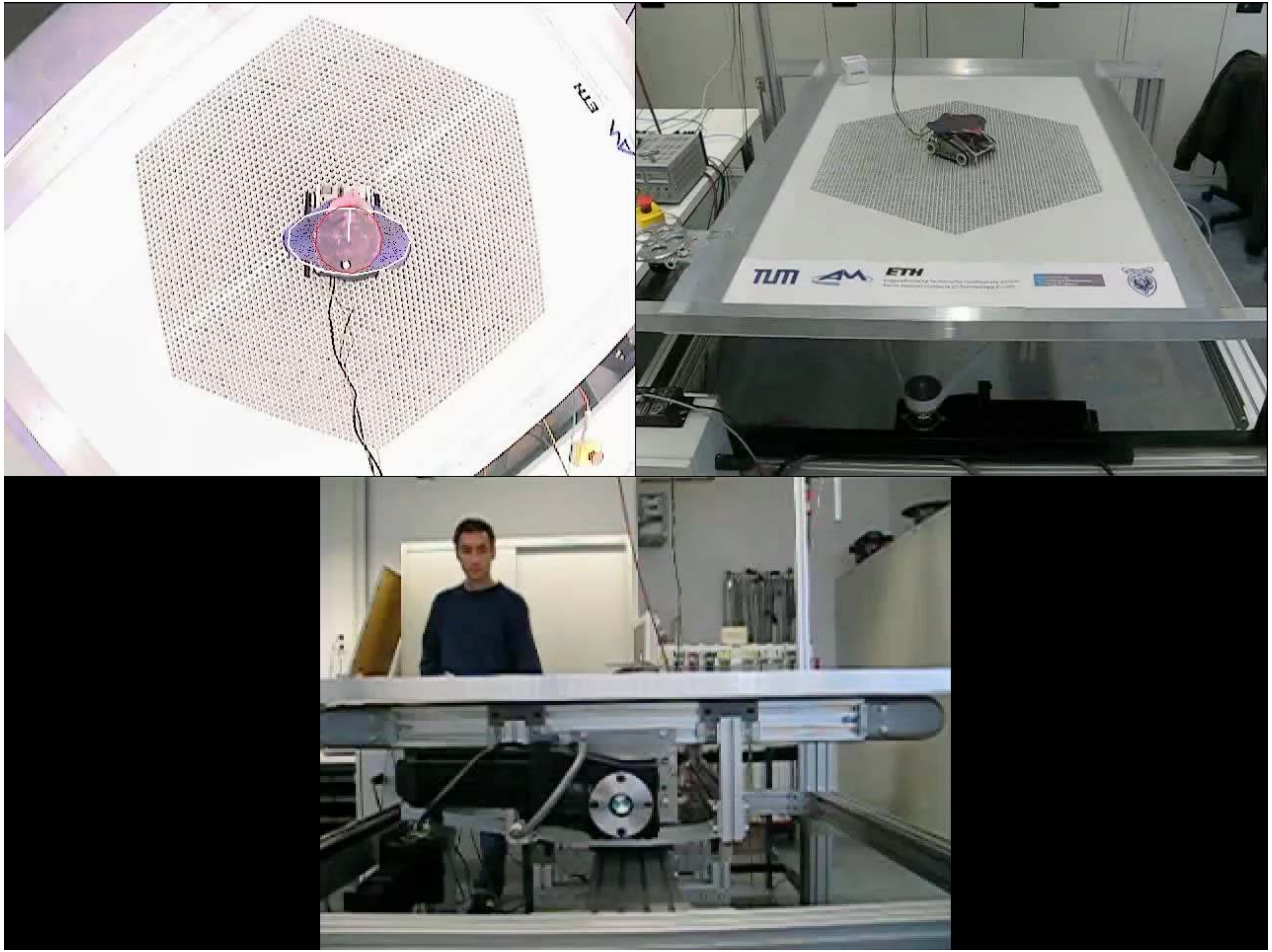
Cyberith Virtualizer

Other Locomotion Devices



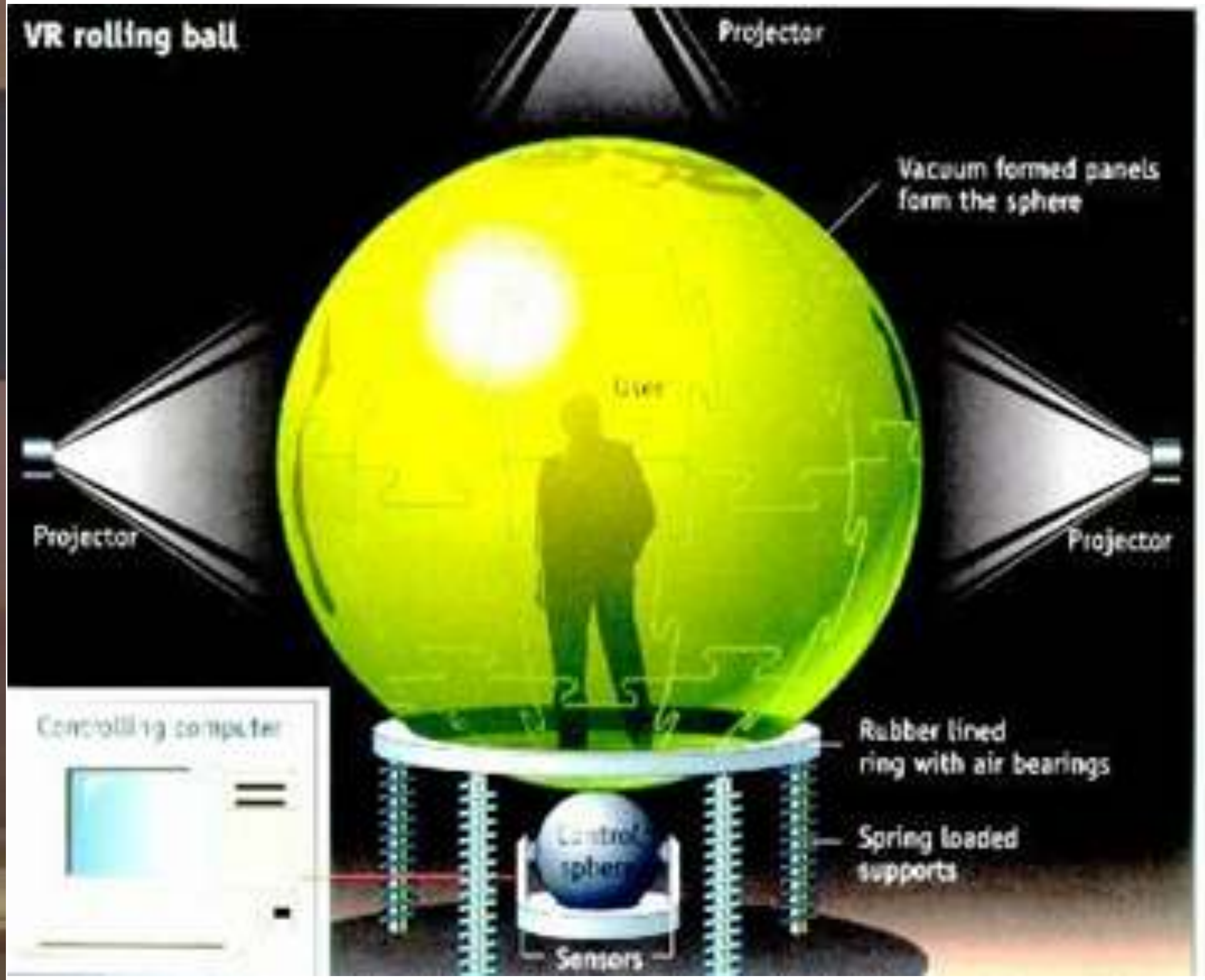
CyberCarpet
Martin Schwaiger,
Dr. Thomas Thümmel,
TU München

How it Works



VirtuSphere

www.virtusphere.com



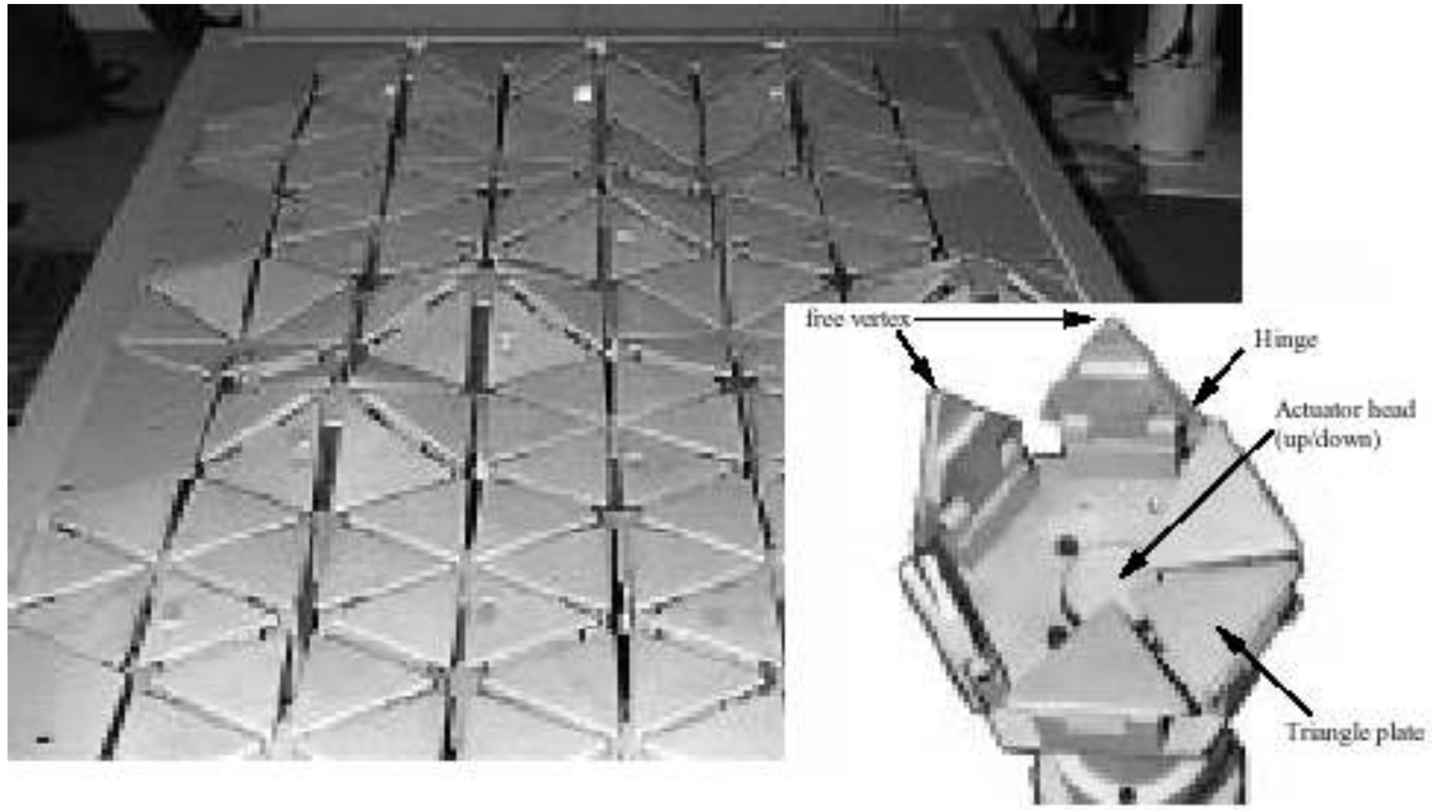
Very Early Concept

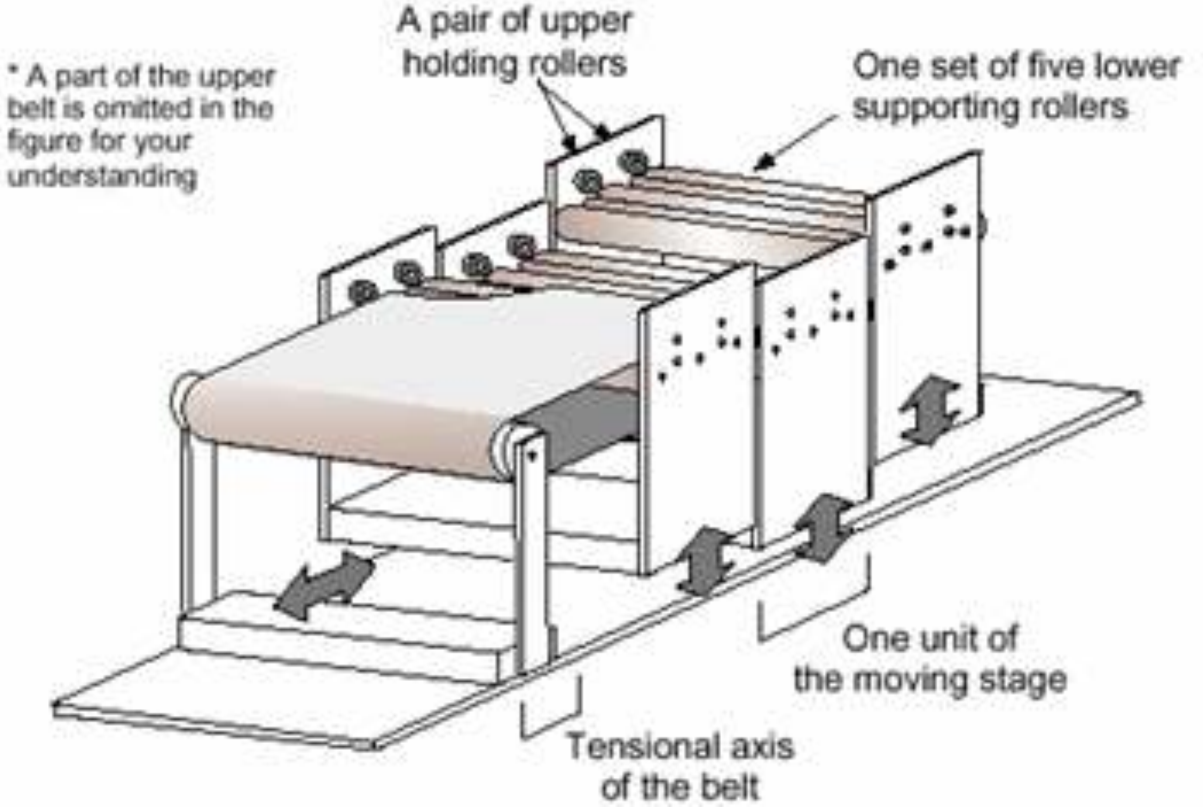
CirculaFloor, 2006

Potential Applications

- Research on behavior & cognition, brain research
- Sports medicine
- Training of soldiers and security staff
- Fun parks (?)
- Architecture:
  - Visualization and realistic exploration of architectural designs
  - Test of escape routes
- Tests on humanoid robots

Simulation of Ground for Real Walking





Other Locomotion Devices

Unconventional Input Devices

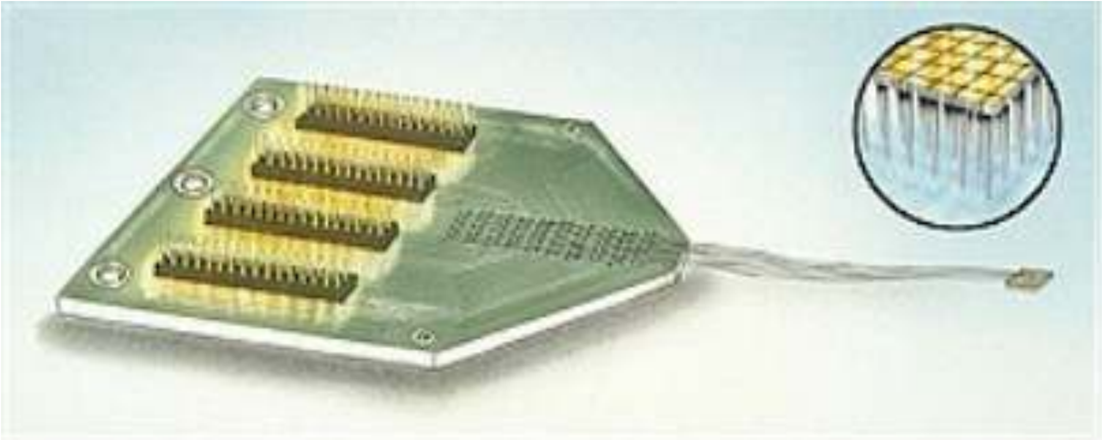
The Shape tape

Are You Aware of Any Other Cool VR Input Device?

https://www.menti.com/zkg58wx3dc

Brain Computer Interfaces

- Idea: control the machine by your brain only (no intermediary devices)
  - So far: EEG
  - SciFi: implant



Another Type of Classification of Input Devices

	Proprioception	Consistent	Useable in lap or the side	Haptics capable	Unencumbered	Physical buttons	Hands free to interact with real world	General Purpose
Hand								
World-Grounded Devices	✓	✓		✓	✓	✓	✓	
Non-Tracked Hand-Held Controllers		✓	✓	✓		✓		
Bare Hands	✓				✓		✓	✓
Tracked Hand-Held Controllers	✓	✓	✓	✓		✓		✓
Hand Worn	✓	✓	✓	✓		✓	✓	✓
Non Hand								
Head Tracking	✓	✓					✓	✓
Eye Tracking							✓	
Microphone			✓		✓		✓	✓
Full-Body Tracking	✓	✓	✓	✓			✓	✓
Treadmills	✓	✓			✓		✓	

Logical Devices

- Problem:
  - Relative / absolute devices (e.g., spacemouse vs. optical tracker)
  - Different dimensionality
  - Different interfaces / APIs to devices
- Solution:
  - Abstract from physical devices → logical devices
  - Classify according to dimensinality of device input
  - Make all logical devices *absolute* devices (integrate relative ones)
- Logical devices [inspired by Wallace 1976]:
  - 0D = "Button" (boolean)
  - 1D = "Value" (float)

- Mapping matrix:

	Mouse	Space-mouse	Tracker	Speech	Glove	Dials
Button (0D)	x	x	(x)	x	x	
Value (1D)	(x)	(x)	(x)	(x)	x	x
Space (6D)	(x)	x	x			
Choice (discr.)	x	x			x	

Software-Side Considerations

- Requirements on architecture:
  - Device could be at arbitrary host → client-server architecture
  - Lots of clients per server
  - Fault tolerant, in case of wrong parameters (e.g., wrong port), device switched off at init time, etc.
  - Ideal: substitute other physical device for logical device by config file (e.g., for driving the navigation)
- 2 kinds of quality of service (QoS): fast or reliable

Kind of data	Treatment of latency	Kind of transport	Data structure
continuously	"better never than late"	UDP	Shared memory
discrete	"better late than never"	TCP	Queue