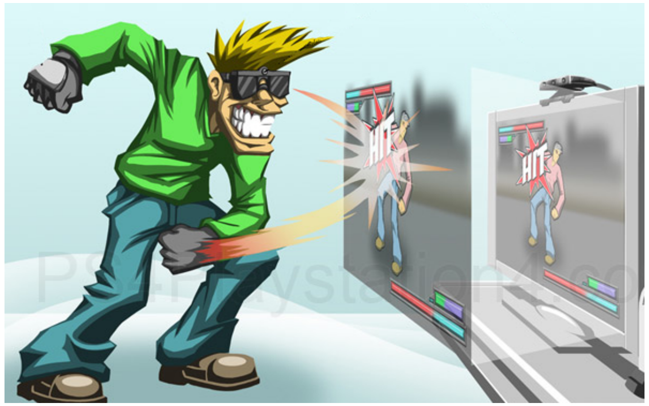
Virtual Reality & Physically-Based Simulation

Haptics



G. Zachmann

University of Bremen, Germany

cgvr.cs.uni-bremen.de

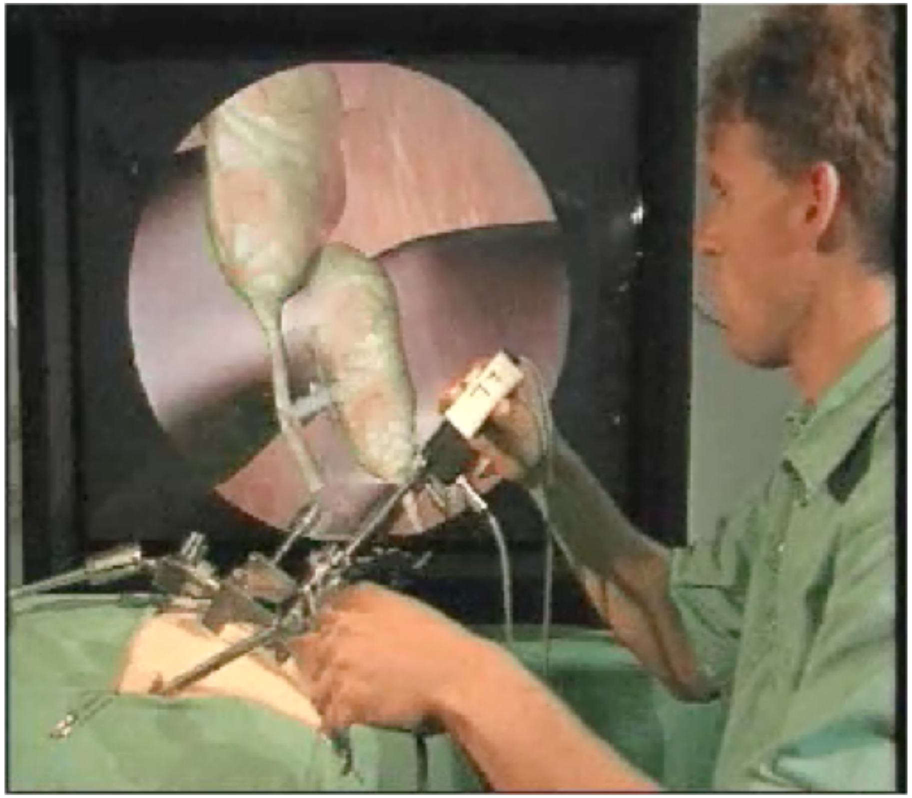
Some Technical Terms

- Haptics = sense of touch and force (greek *haptesthai* = berühren)
- Special case: force feedback
- What is to be rendered:
  - Forces on the user's hand / arm (= haptic "image" of objects)
  - Haptic texture of surfaces (roughness, grain, friction, elasticity, ...)
  - Shape of objects by way of touching/feeling

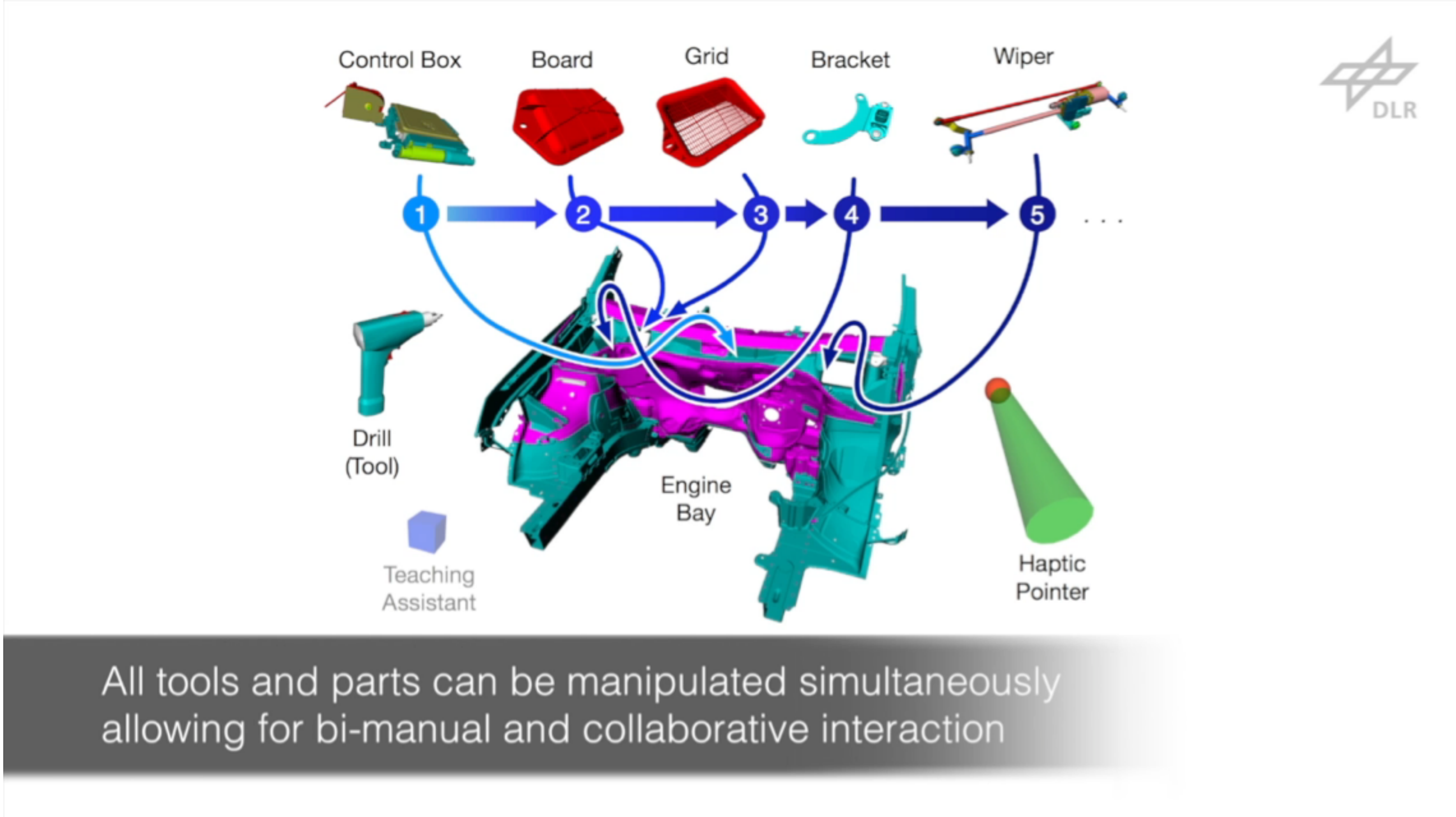
Applications

- Training of minimally invasive surgery (surgeons rather work by feeling, not seeing)
- Games? Can increase presence significantly (self-presence, social presence, virtual object presence)
- Industry:
  - Virtual assembly simulation (e.g., to improve worker's performance / comfort when assembling parts)
  - Styling (look & feel of a new product)
    - Ideally, one would like to answer questions like "how does the new design of the product feel when grasped?"

Example Application: Minimally Invasive Surgery



Another Application: Assembly Simulation



DLR: A Platform for Bimanual Virtual Assembly Training with Haptic Feedback in Large Multi-Object Environments

A Collection of Force Feedback Devices

CyberForce

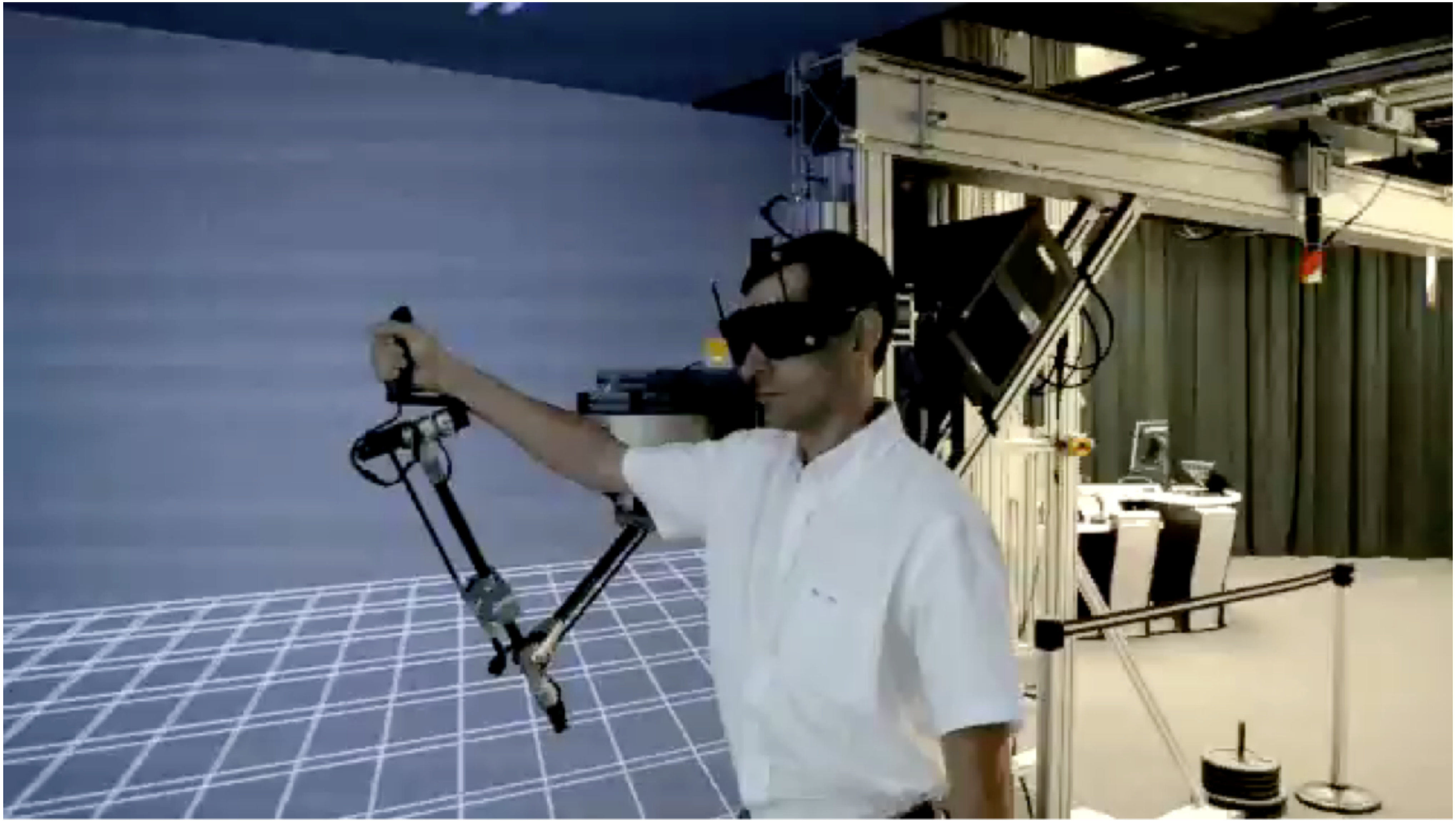
CyberForce

Phantom

Sarcos (movie)

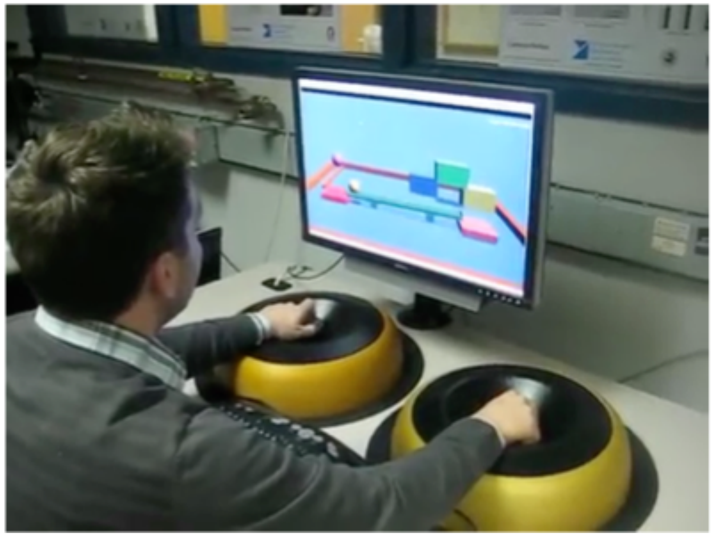
(movie)

Force Dimension

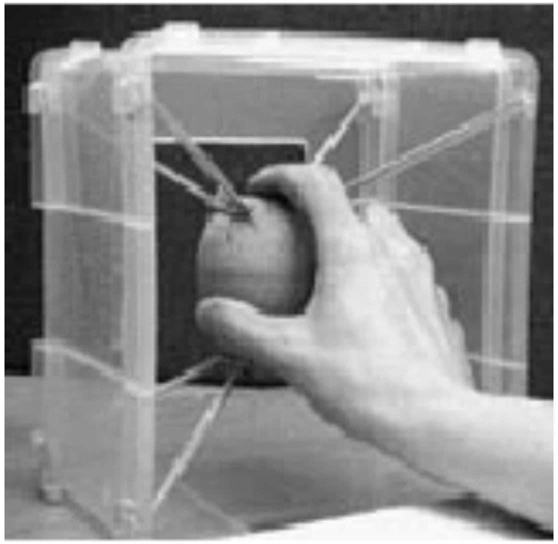


Scale-1 by Haption

(movies)



Maglev (Bytterfly Haptics)



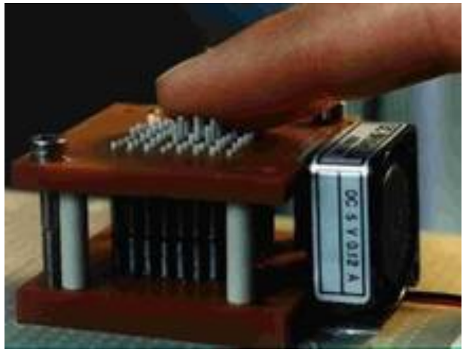
Spidar

Devices with Force Feedback via Wires (Spidar Variants)

Two-Handed Multi-Fingers Haptic Interface Device: SPIDAR-8

INCA 6D von Haption

Tactile Displays

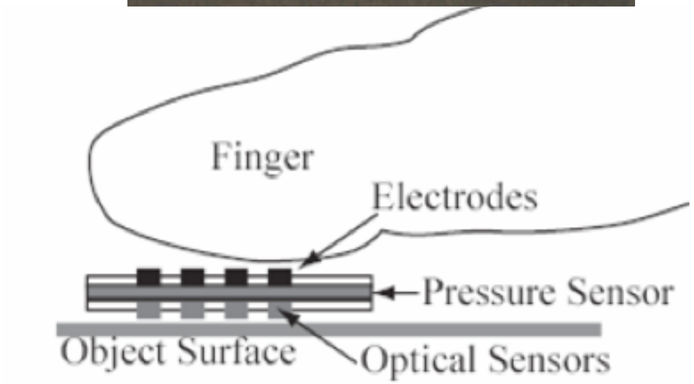


CyberTouch

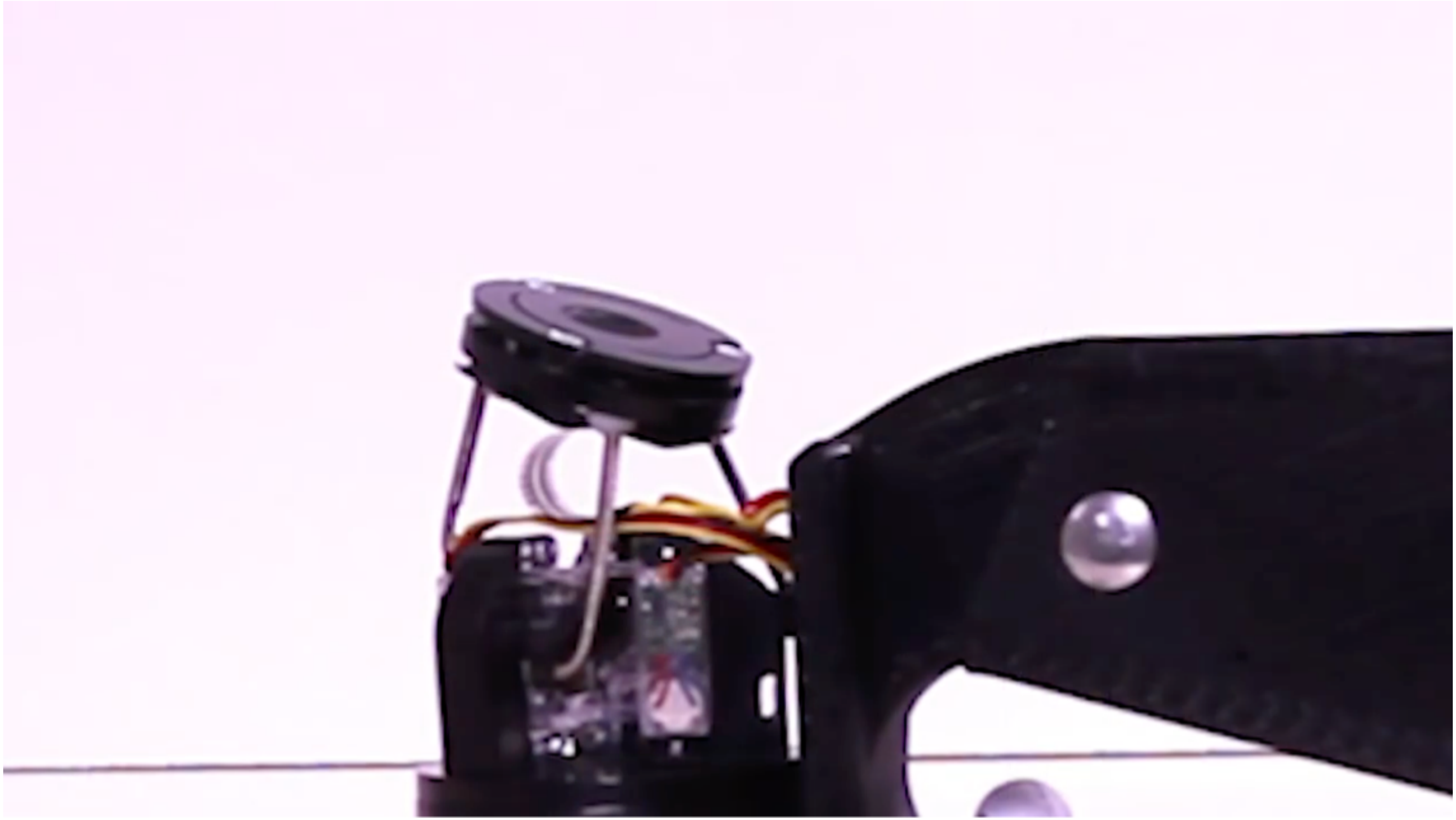
GloveOne

AURA INTERACTOR

Feelex

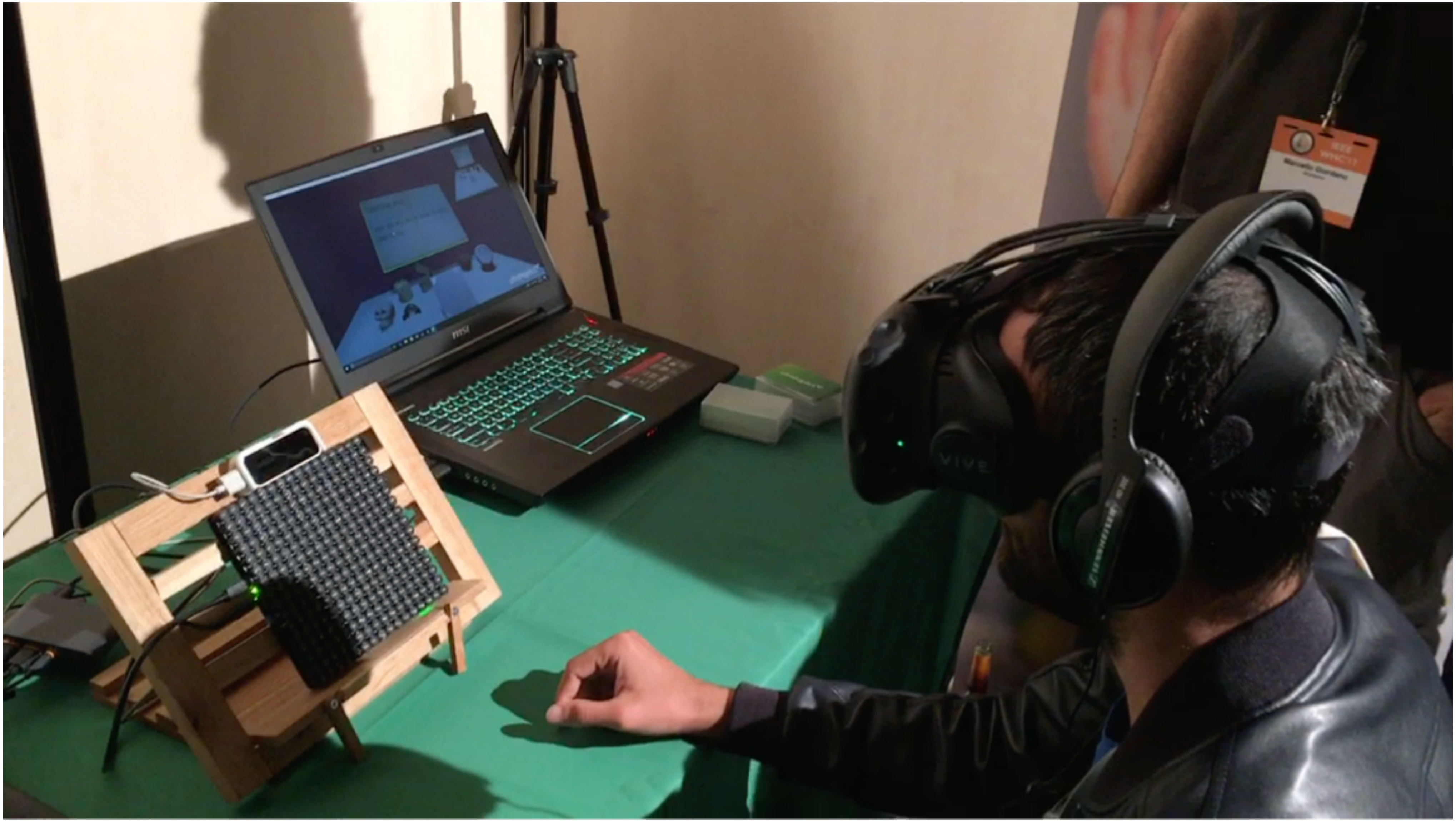


SmartTouch

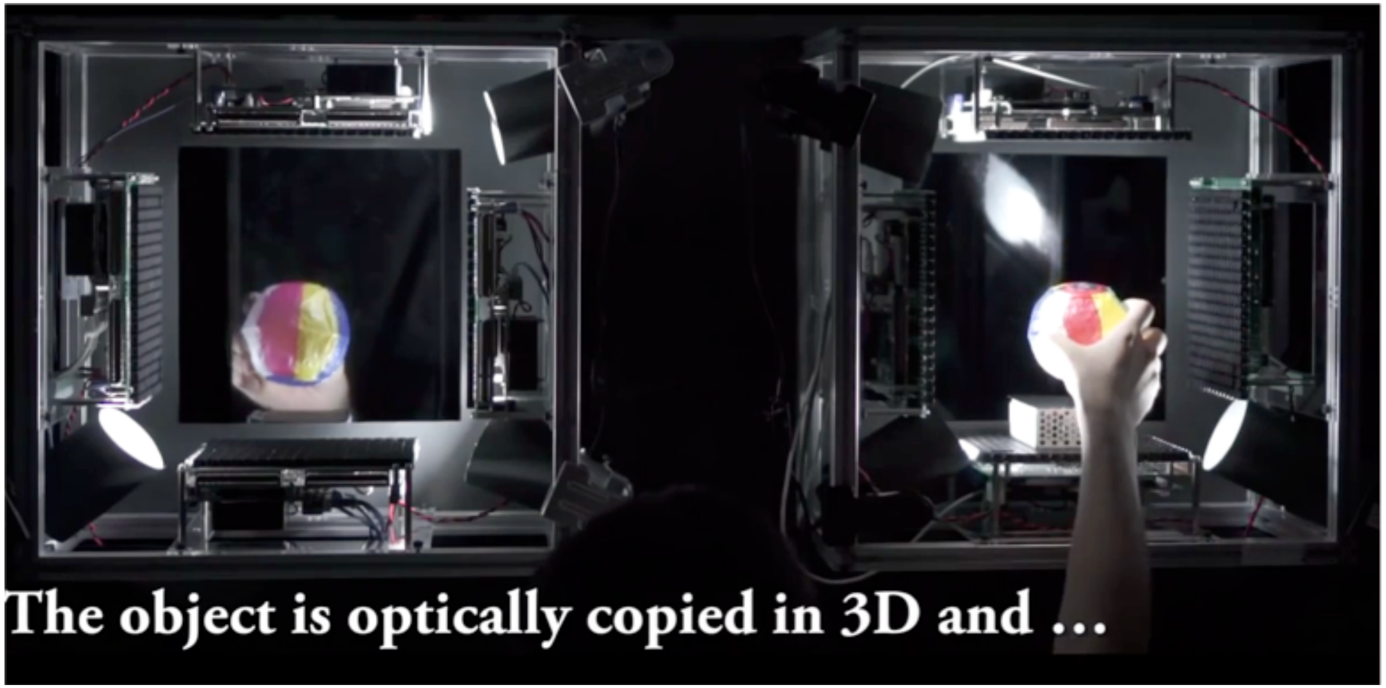


NormalTouch & TextureTouch, 2016, Microsoft

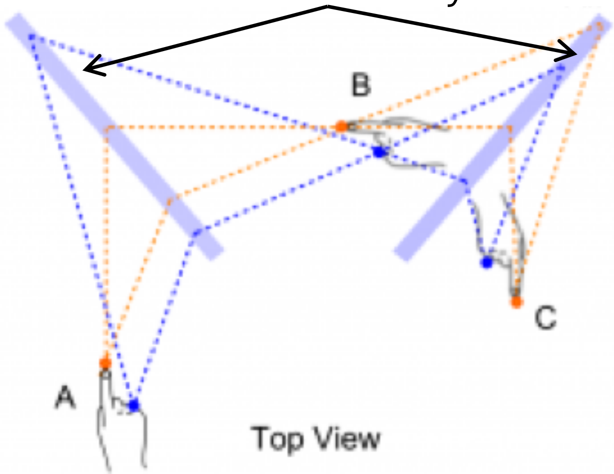
Haptic Feedback via Interference of Ultrasound



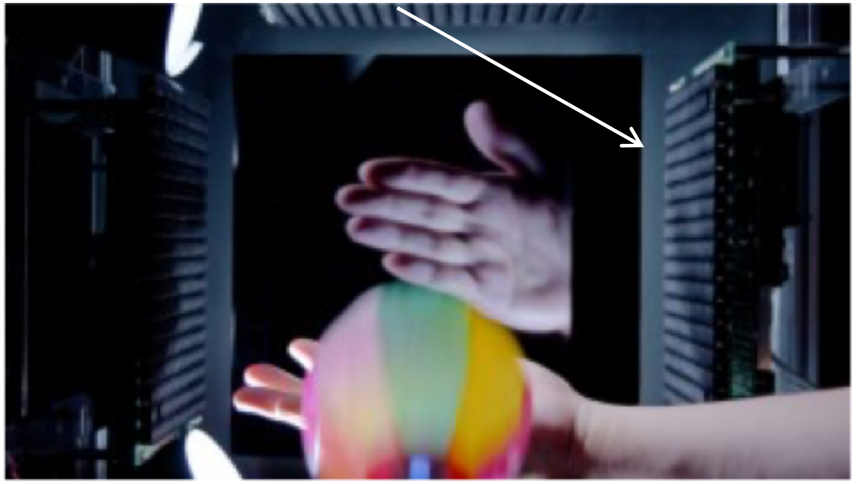
Haptoclone



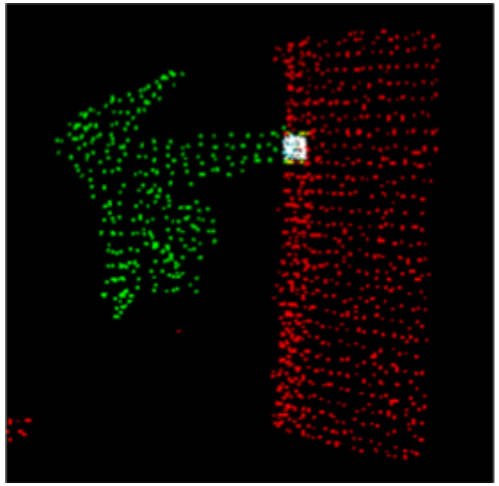
micro mirror array



1992 phase-controlled ultrasound array

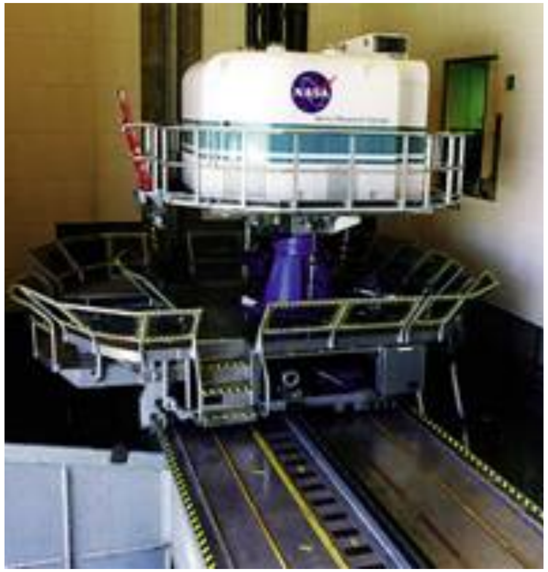


Depth sensor for objs

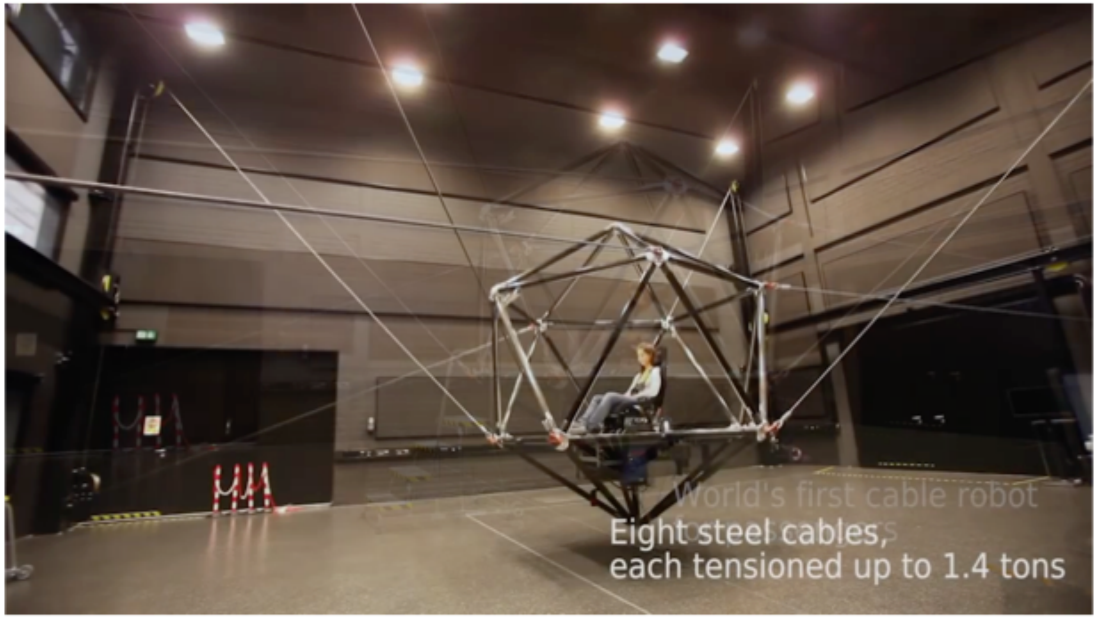


Motion Platforms (Not Really Force-Feedback)

BlueTiger

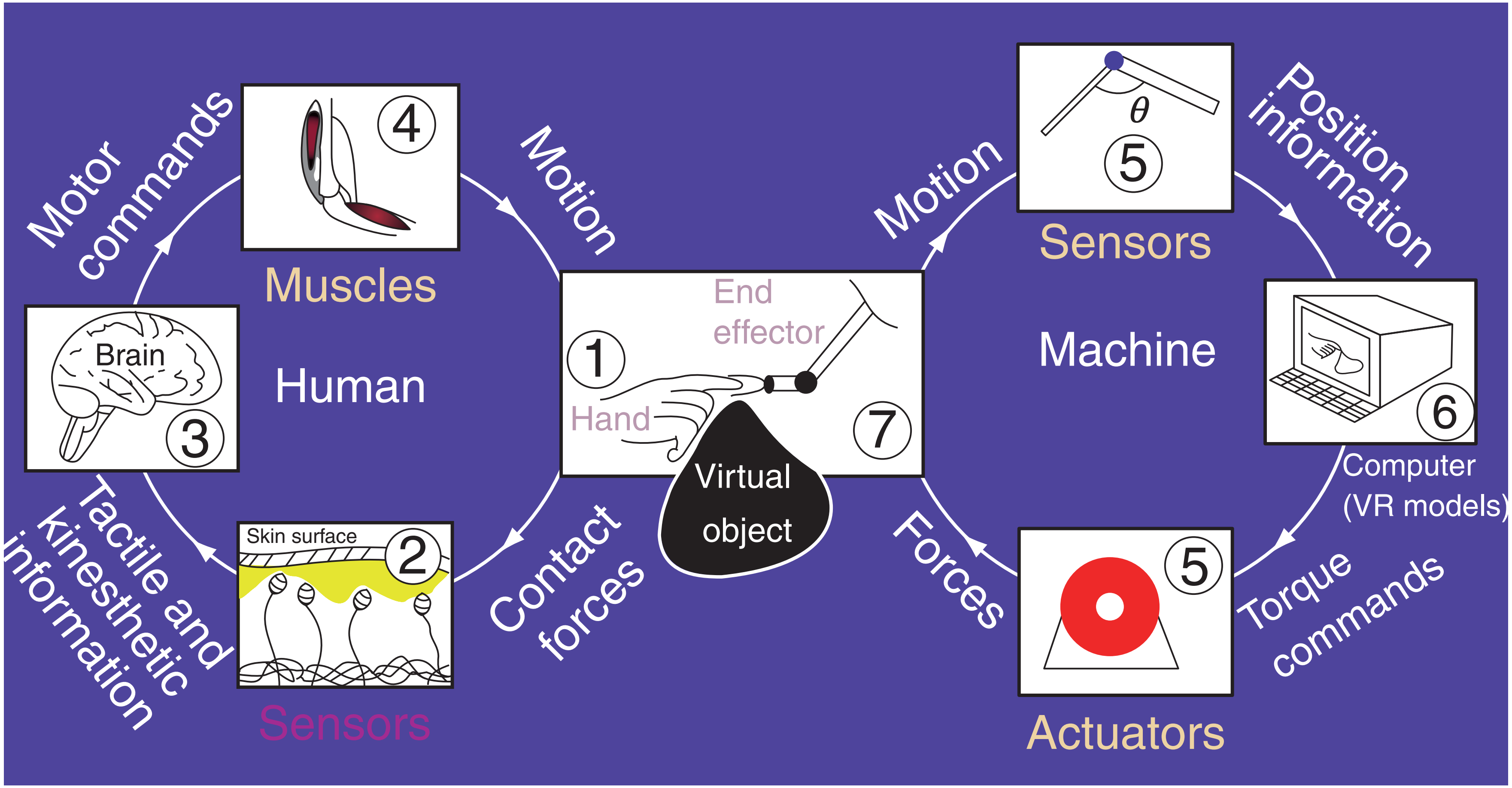


Flogiston



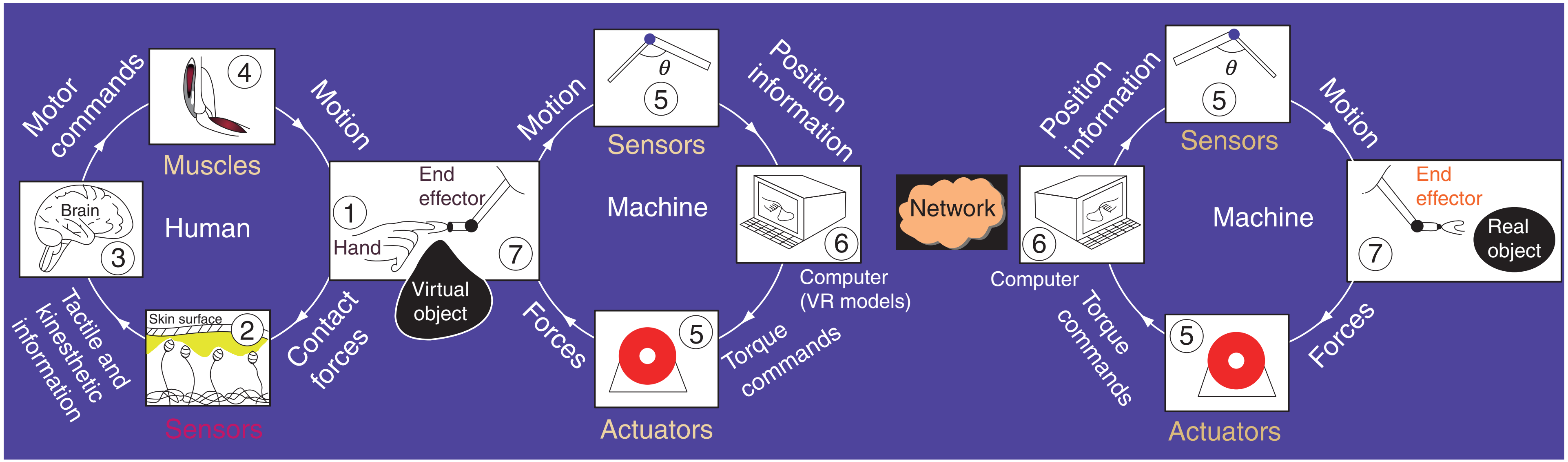
MPI Tübingen

The Special Problem of Force-Feedback Rendering



M A Srinivasan & R Zimmer: *Machine Haptics*.
New Encyclopedia of Neuroscience, Ed: Larry R. Squire, Vol. 5, pp. 589-595, Oxford: Academic Press, 2009

… and that of Telepresence

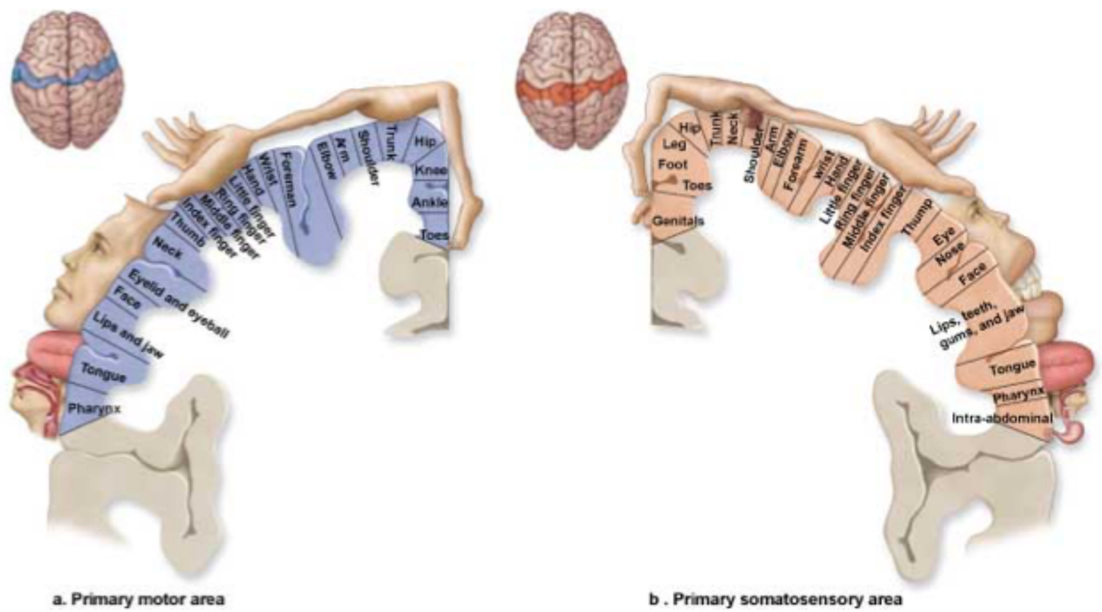


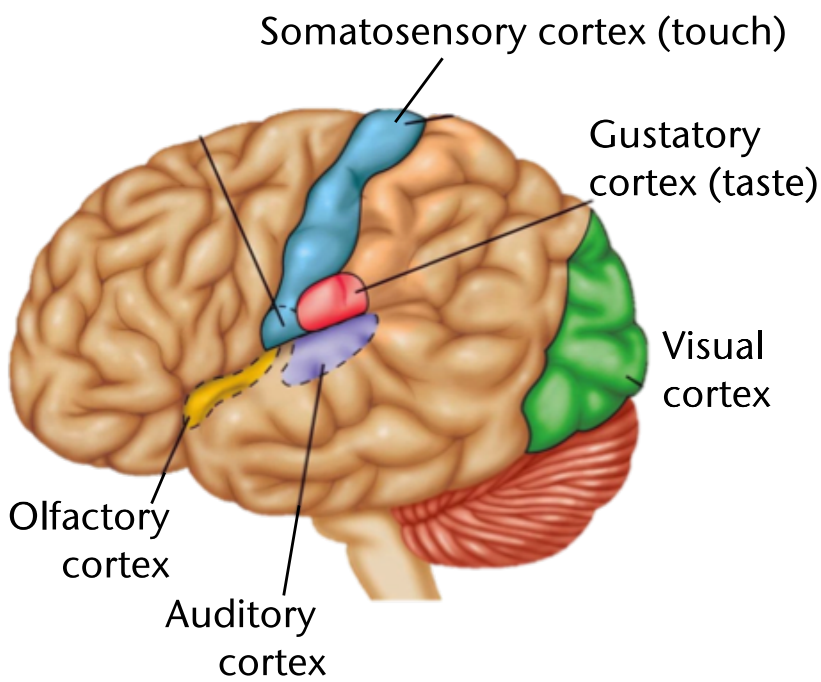
M A Srinivasan & R Zimmer: *Machine Haptics*.
New Encyclopedia of Neuroscience, Ed: Larry R. Squire, Vol. 5, pp. 589-595, Oxford: Academic Press, 2009

Putting the Human Haptic Sense Into Perspective

- Amount of the cortex devoted to processing sensory input:
  - Haptic sense is our second-most important sense

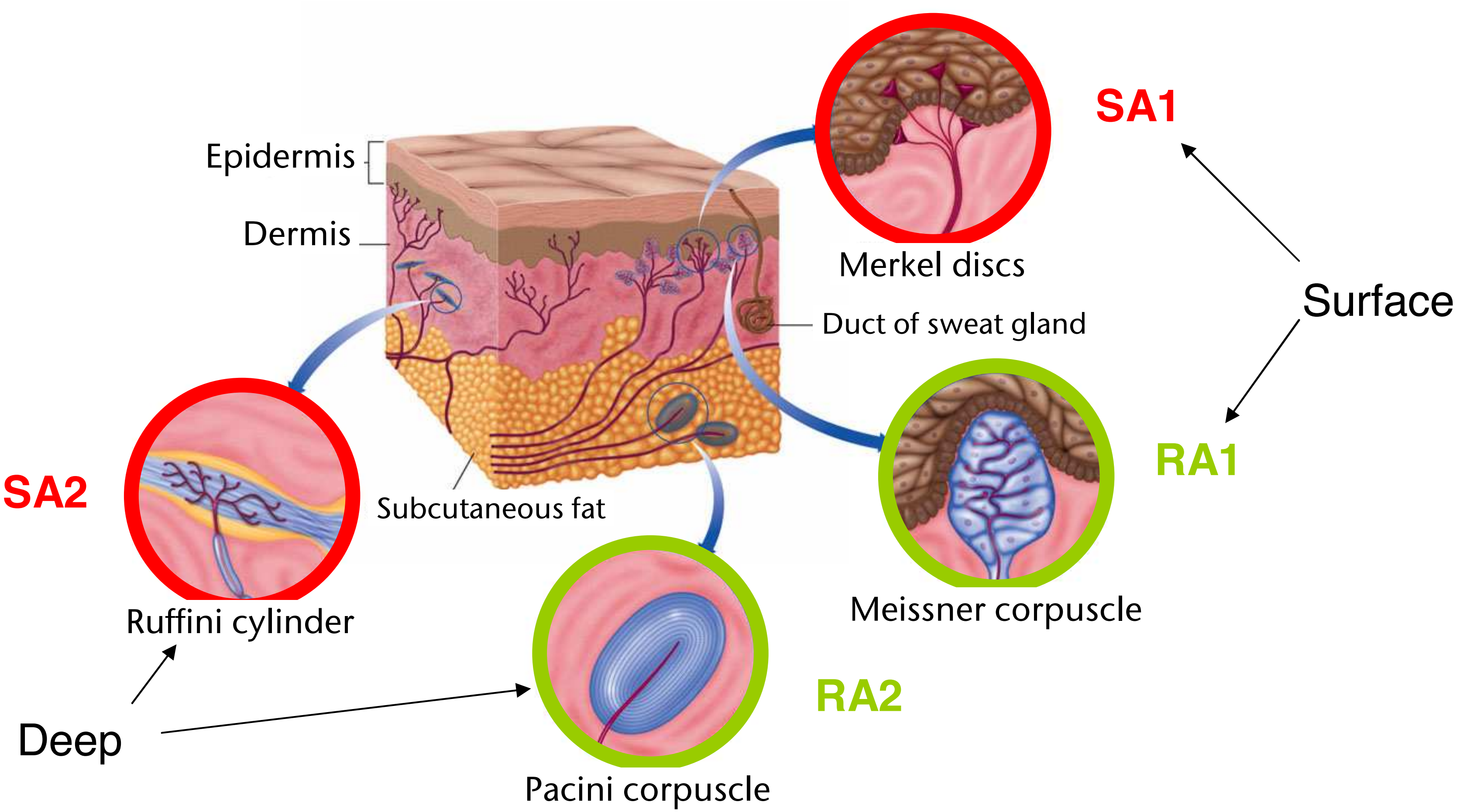
Sensory Input	Amount of cortex / %
Visual	30
Haptic	8
Auditory	3



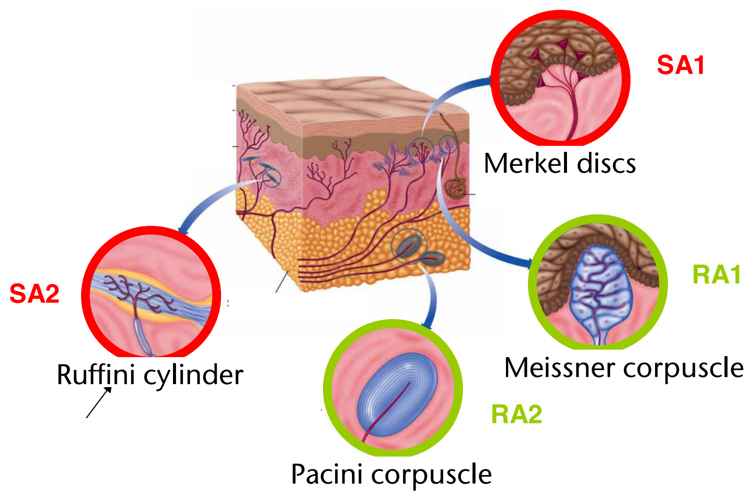


The Human Tactile Sensors

- There are 4 different kinds of sensors in our skin:



- Their characteristics:
  - Ruffini & Merkel: slowly adapting (SA)
    ⟶ fire as long as the stimulus persists
  - Meissner & Pacini: rapidly adapting (RA)
    ⟶ fire only at onset and offset of stimulus



Adapting Rate

Response to vibration frequency	**slow**	**fast**	Location in Skin
low	Merkel	Meissner	**surface**
high	Ruffini	Pacini	**deep**

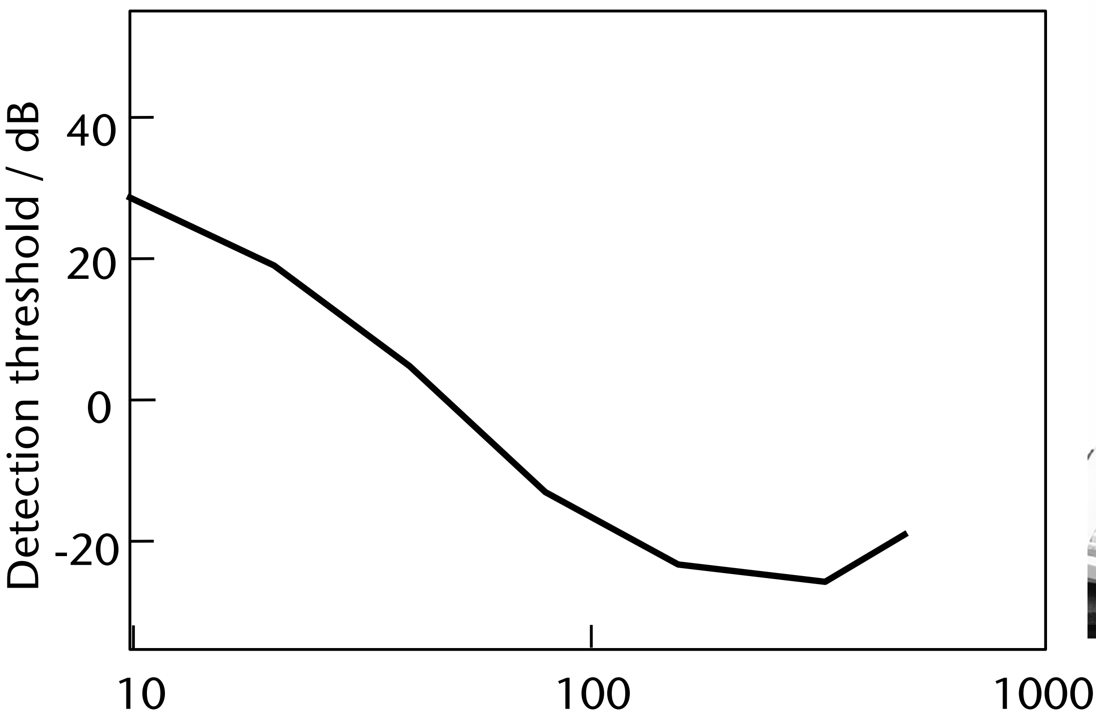
Some Human Factors Regarding Haptics

- Human factors of the tip of a finger:
  - Precision = 0.15 mm regarding the position of a point
  - Spatial acuity = 1 mm   (i.e., discrimination of 2 points)
  - Detection thresholds ("there is something"):
    0.2 micrometers for ridges; 1-6 micrometers for single points
  - Temporal resolution: 1 kHz   (compare that to the eye!)
- Kinaesthetic (proprioceptive) information:
  - Obtained by sensors in the human muscles
  - Can sense large-scale shapes, spring stiffness, ...
  - Human factors:
    - Acuity: 2 degrees for finger, 1 degree for shoulder
    - 0.5-2.5 mm (finger)

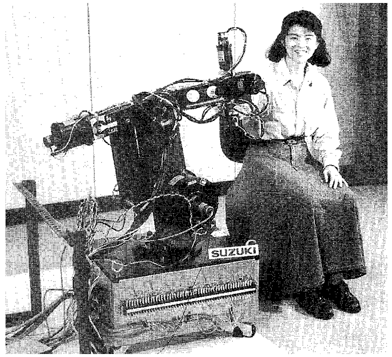
- Time until a reflex occurs:
  - Reflex by muscle: 30 millisec
  - Reflex through spinal cord: 70 millisec
  - Voluntary action: ?
- The bandwidth of forces generated by humans:
  - 1-2 Hz for irregular force signals
  - 2-5 Hz when generating periodic force signals
  - 5 Hz for trained trajectories
  - 10 Hz with involuntary reflexes
- Forces of hand/arm:
  - Max. 50-100 N
  - Typ. 5-15 N (manipulation and exploration)
  - Just noticeable difference (JND) = $\left| \frac{F_{ref} - F_{comp}}{F_{ref}} \right| = 0.1$ (10%)

Simulation Factors

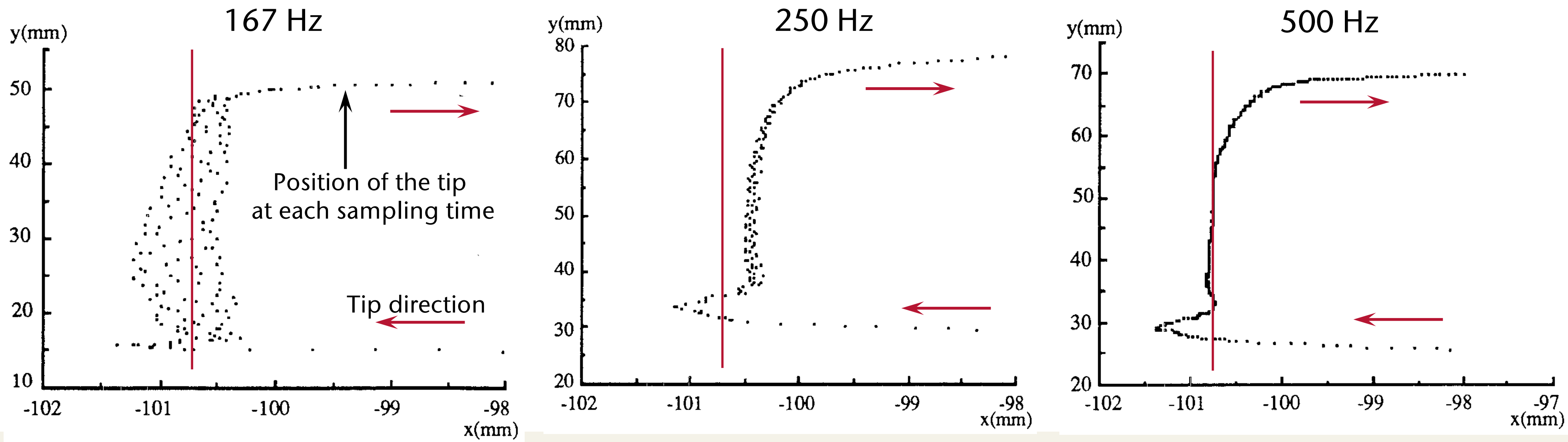
- Sensation of stiffness/rigidity: in order to render hard surfaces, you need >1 N/mm (better yet 10 N/mm)
- Detection threshold for vibrations:
  - Simulation must run at Nyquist frequency ⟶ in order to generate haptic signals with 500 Hz, the simulation loop must run at 1000 Hz



Rule of Thumb: 1000 Hz Update Rate Needed for Haptic Rendering

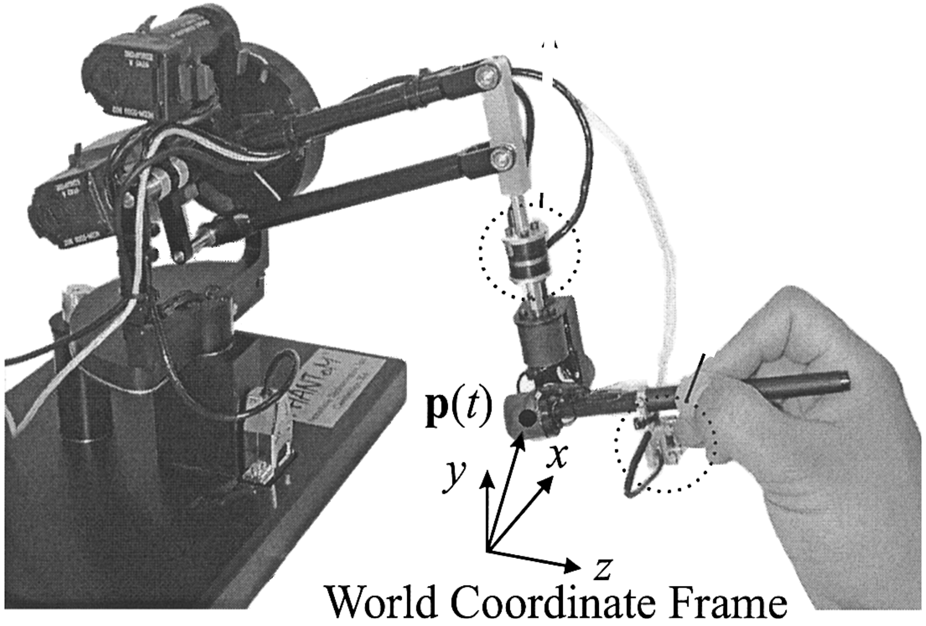


- An Experiment as "proof":
  - Haptic device with a pen-like handle and 3 DOFs
  - The virtual obstacle = a flat, infinite plane
  - Task: move the tip of the pen along the surface of the plane (*tracing task*)
  - Impedance-based rendering (later)
  - Stiffness = 10000 N/m, coefficient of friction = 1000 N/(m/sec)
  - Haptic sampling/rendering frequencies: 500 Hz, 250 Hz, 167 Hz



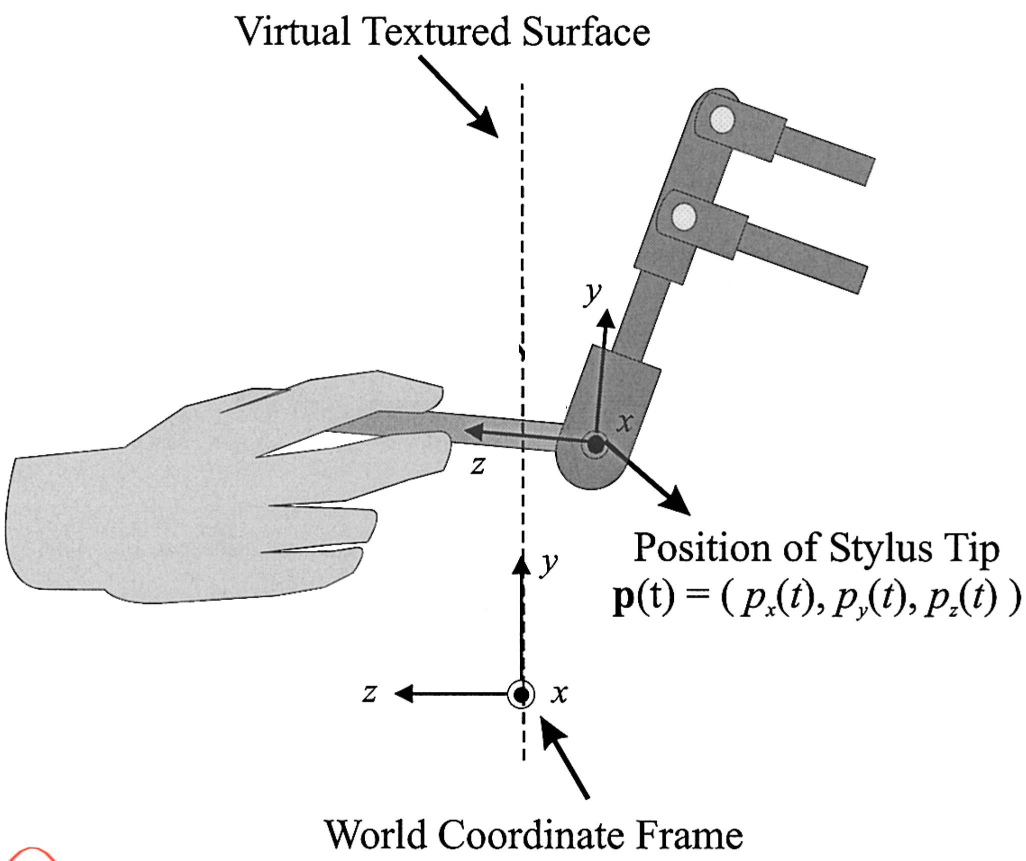
Haptic Textures

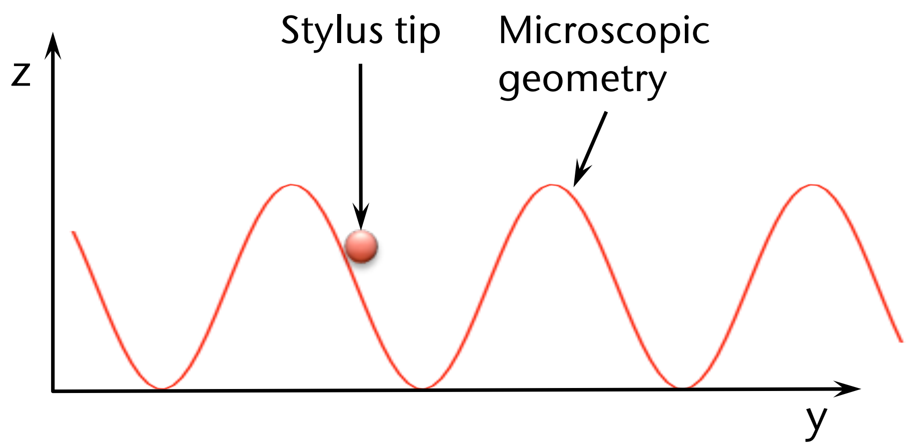
- Texture = fine structure of the surface of objects (= micro-geometry); independent of the shape of an object (= macro-geometry)
- Haptic textures can be sensed in two ways by touching:
  - Spatially
  - Temporally (when moving your finger across the surface)
- Sensing haptic textures via force-feedback device: as you slide the tip of the stylus along the surface, texture is "transcoded" into a temporal signal, which is then output on the device (e.g., use IFFT to create the signal)



A Frequent Problem: "Buzzing"

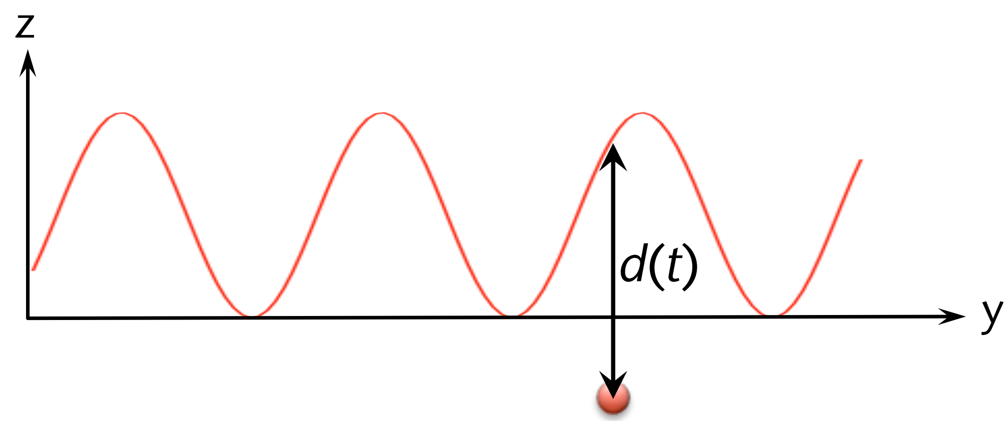
- Consider this experiment: a simple Phantom-like device and a surface geometry in the shape of a microscopic sine-wave



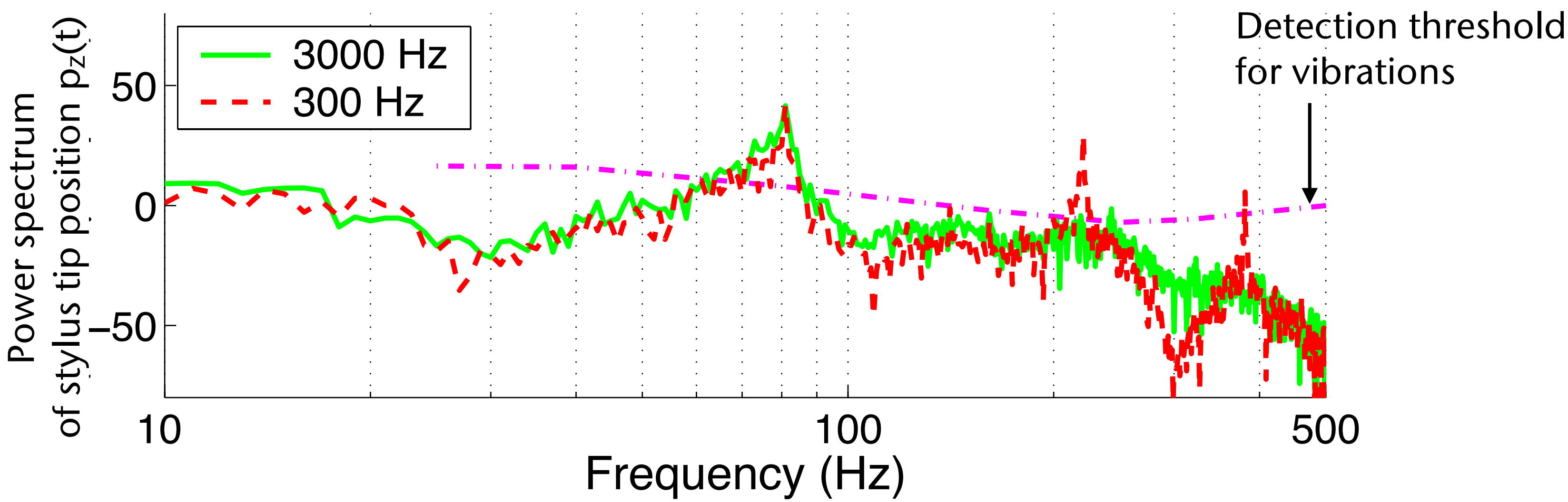


- The force that is rendered (= output on the actuators):

$$F(t) = k_s d(t)$$



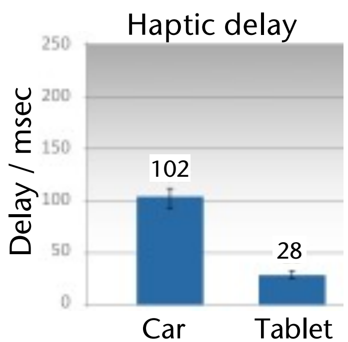
- Result with different rendering frequencies:

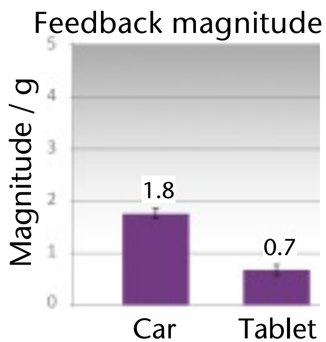


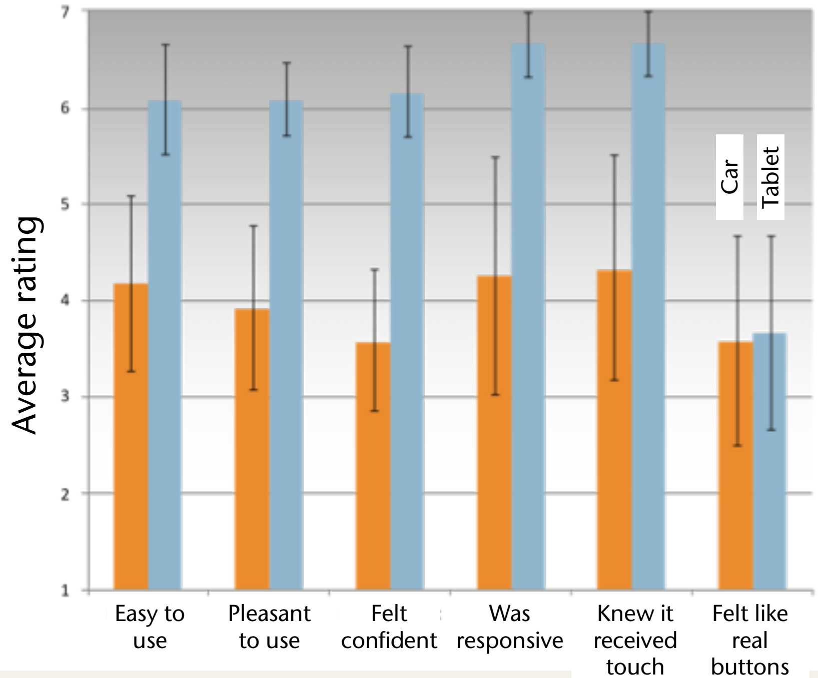
Latency in Haptic Feedback

- General results [2009]:
  - Latency for haptic feedback < 30 msec ⟶ perceived as instantaneous
  - Latency > 30 msec ⟶ subjective user satisfaction drops
  - Latency > 100 msec ⟶ task performance drops
- Real-life story: touch panel of the info-tainment system of a Cadillac, 2012
  - Conditions: infotainment and tablet, both with touch screen and haptic feedback, but different delay

Infotainment system in car with haptic touch screen

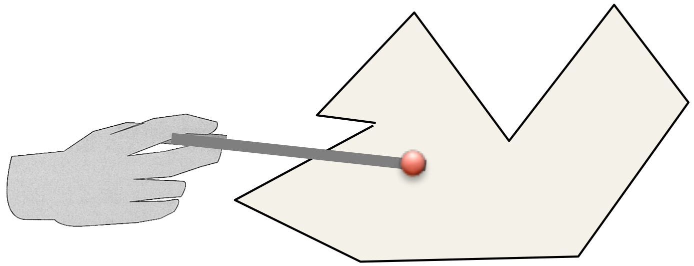


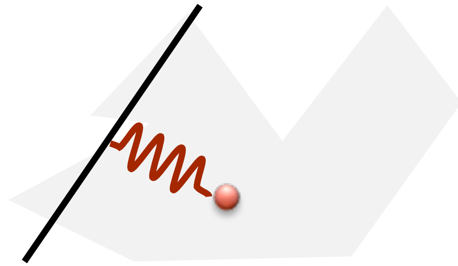


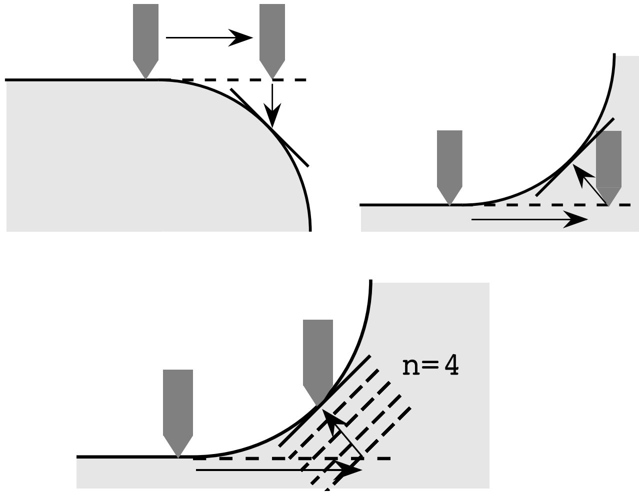


Intermediate Representations

- Problem:
  - Update rate should be 1000 Hz!
  - Collision detection between tip of stylus und virtual environment takes (often) longer than1 msec
  - The VR system needs even more time for other tasks (e.g., rendering, etc.)
- Solution:
  - Use "intermediate representation" for the current obstacle (typically planes or spheres)
  - Put haptic rendering in a separate thread
  - Occasionally, send an update of the intermediate representation from the main loop to the haptic thread



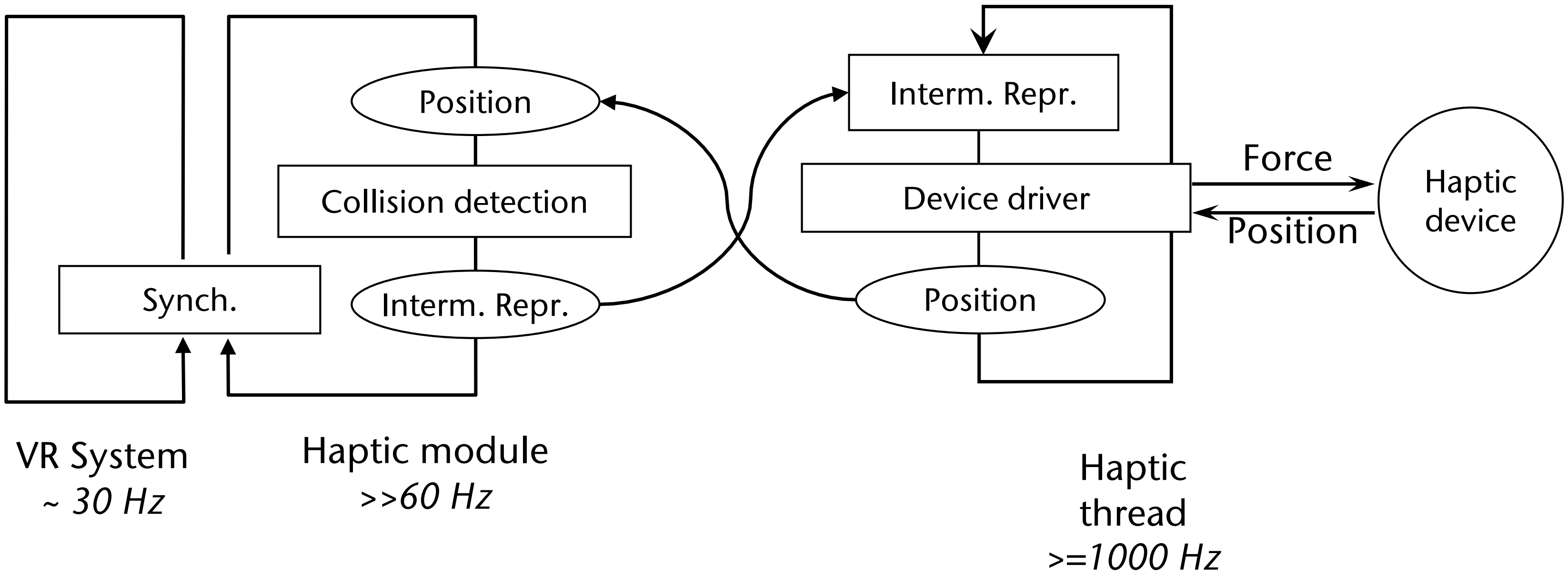




Software Archtecture

- A haptic device consists of:
  - Sensor measures force (admittance-based) or position (impedance-based)
  - Actuator moves to a specific position (admittance-based) or produces a force/acceleration (impedance-based)

- Archtiecture:



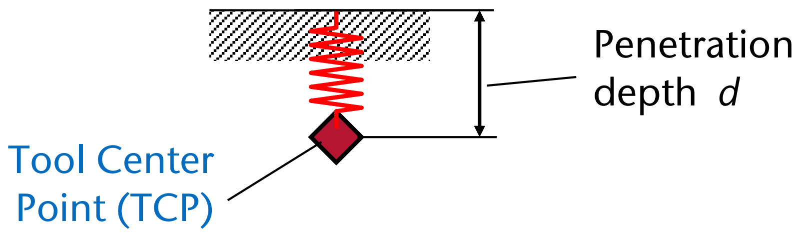
Two Principles for Haptic Rendering

- Dynamic object = object that is being grasped/moved by user;
  the end-effector of the haptic device is coupled with the dynamic object
- Dynamic models:
  - *Impedance approach*:
    haptic device returns current position,
    simulation sends new forces to device (to be exerted on human)
  - *Admittance approach*:
    haptic device returns current forces (created by human),
    simulation accumulates them (e.g. by Euler integration),
    and sends new positions to device that it assumes directly
  - In both cases, simulation checks collisions between dynamic object and rest of the VE
- Penalty-based approach: the output force depends on the penetration depth of the dynamic object
- Requirements:
  - 1000 Hz
  - Constant update rate

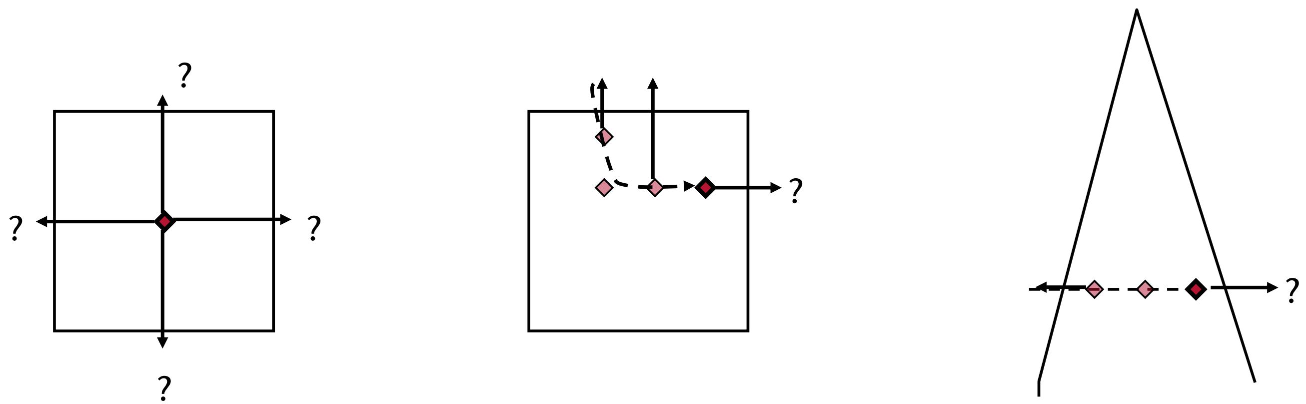
The "Surface Contact Point" Approach

- The penalty force given by *Hooke's law*:

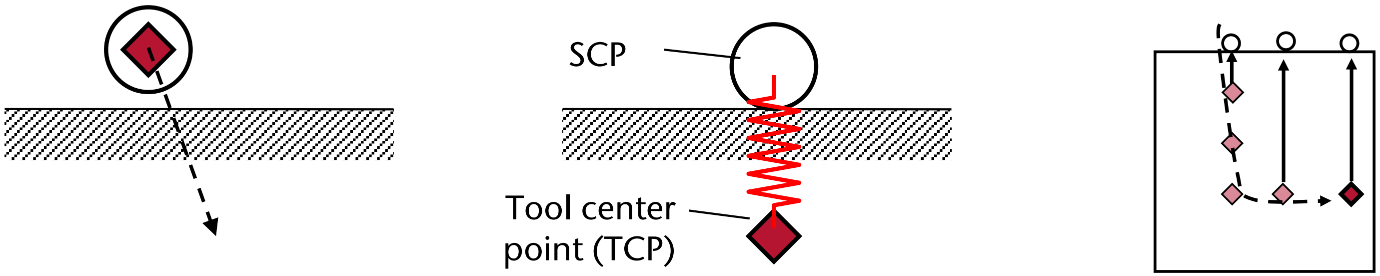
$$F = k \cdot d$$



- Question: what exactly is the penetration depth?
  - Naïve method: calculate a depth and repulsion direction for each inner point
  - Problem: the history of the TCP is ignored



- Conclusion: with haptic rendering (at least) you need the history in some way
- Idea: represent the history as surface contact point (SCP)



- Determining the constraints:

  Iterate at most 3 times:

  determine polygon p, that is intersected by $\overline{\mathrm{SCP}^{t-1}\mathrm{TCP}^{t}}$;

  determine point P that is on plane defined by p and has minimal distance to TCP

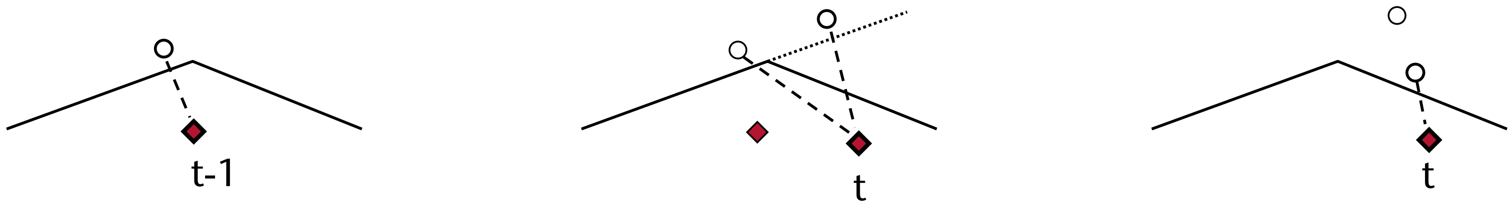
- In order to achieve numerical robustness: lift SCP slightly above the polygons
- Utilize temporal coherence: consider only polygons in the neighborhood of the current SCP

- How to compute the SCP $\mathbf{x}$:

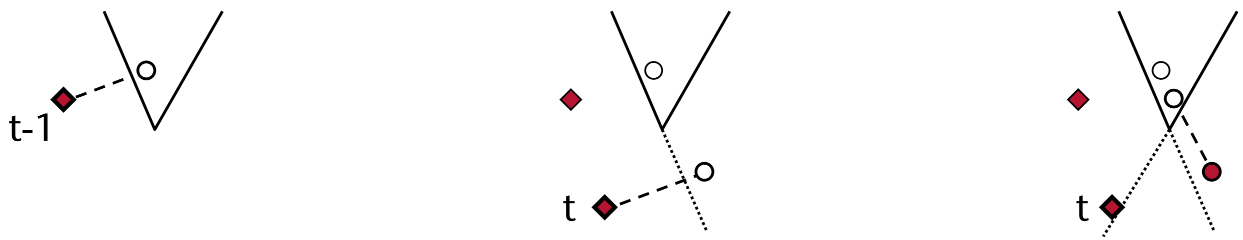
  minimize $\|\mathbf{x} - \mathbf{x}_{\mathrm{TCP}}\|^2$

  under the constraint $\mathbf{n}_i \mathbf{x} - d_i = 0 \ , \quad i = 1, 2, 3$

- With Lagrange's multiplication rule (Lagrange'sche Multiplikatorenregel), we obtain a simple system of linear equations
- Example of the algorithm for a convex edge:



- Example for a concave edge:



The Case for Constant Haptic Update Rates

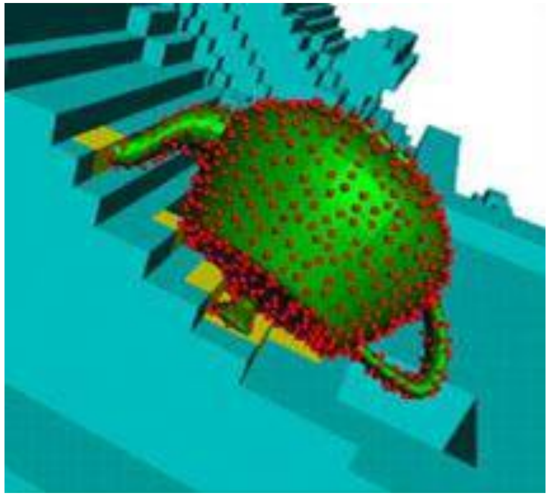
- Question: why is a **constant** update rate so important?
- Answer: because otherwise we get "jitter" (Rütteln, Ruckeln)
- Another reason will be given in the Voxmap-Pointshell method

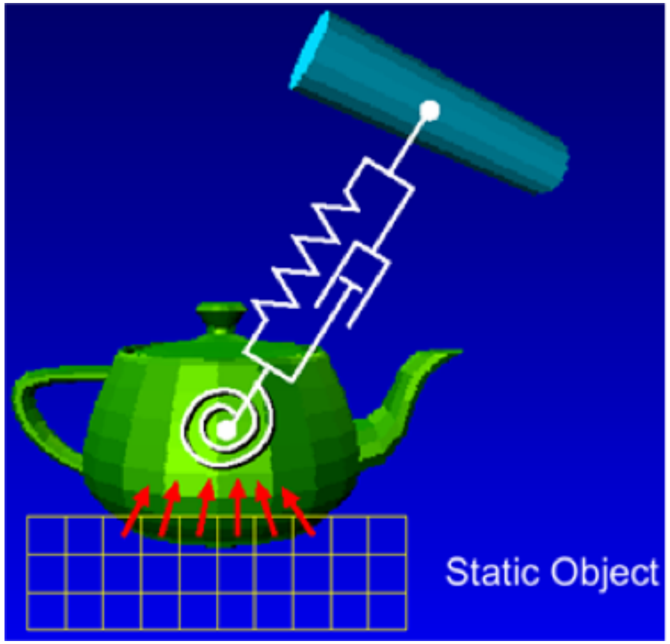
The Reason for Device Jitter

- Assumption:
  - The user is just starting to penetrate an obstacle with the TCP
  - The force generated by the device is still insignificantly small compared to the inertia of the complete system (= user + device)
  - The obstacle has a bit of elasticity (like a spring, possibly a stiff one)
- Consequence: the penetration depth of the TCP increases linearly
- We expect: the force generated by the device increases linearly, too (stepwise)
- Now, consider the case where the computations take somewhat longer time than usual:
  - The TCP moves by a larger distance (since the last update)
  - The force by the device exerted on the user remains the same!
  - Then, the device sends its current position to the haptic loop → the penetration depth in the simulation increases a lot from one iteration to the next
  - The force increases much more between two successive iterations!

The Voxmap-Pointshell Approach

- Representation of objects (no polygons):
  - Dynamic object → sample surface by lots of points = point shell
  - Rest of the scene → embed in 3D grid; voxmap = all voxels inside an obstacle

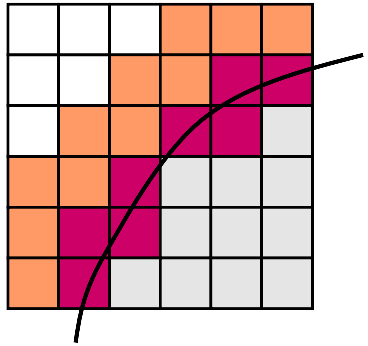


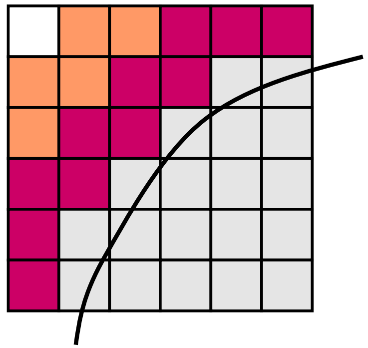


- Overall idea:
  1. Compute forces for all penetrating points
  2. Compute total force on dynamic object
  3. Compute force on haptic handle

The VoxMap

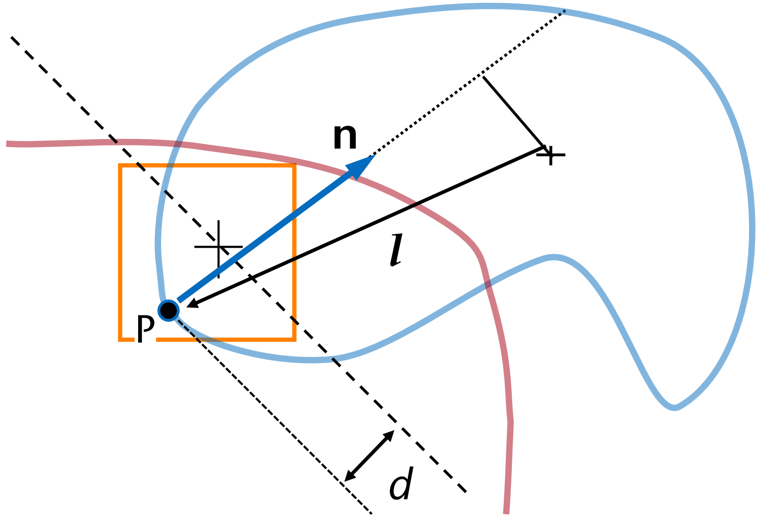
- Voxmap = 3D distance field
- Generation:
  - Scan-convert the surface (in 3D) → voxels that are intersected by the surface
  - Do a breadth-first search starting from the border of the "universe" → all voxels outside any obstacles
  - All other voxels must be inside
    - For each inner voxel, compute the minimum distance to the surface
    - Alternative: propagate the distance from the surface to the inner regions (by way of the Chamfer method)





The Force Acting on one Point

- Force acting on a point P on the surface of the dynamic object:
  - Direction = surface normal **n**
  - Penetration depth = voxel depth
    + distance from P to the plane
    given by voxel center and normal **n**
  - Force: $\mathbf{F} = k_v \cdot d \cdot \mathbf{n}$
- Torque (Drehmoment): $\mathbf{M} = l \times \mathbf{F}$
- Why use **n** and not the vector from the voxel to the closest point on the surface of the obstacle?
  - Then, the direction of F would not depend on the orientation of the dynamic object
  - Also, there would be discontinuities in the force F, when the object translates such that some points of the pointshell cross into other voxels



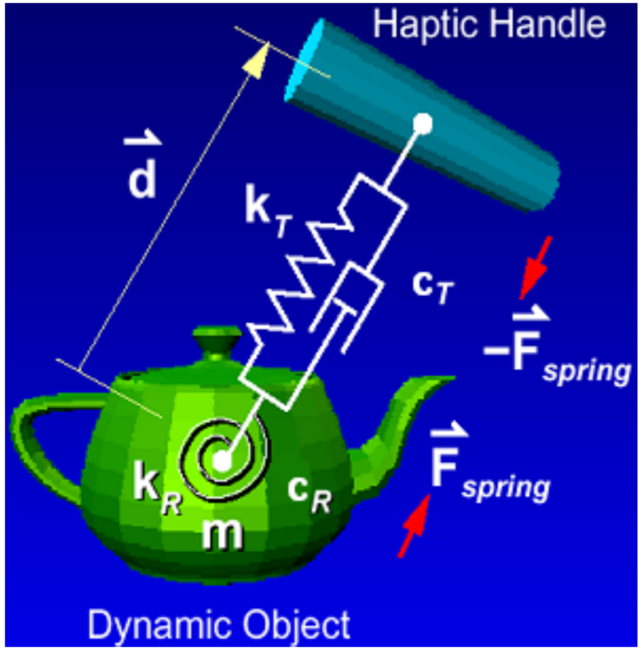
The Virtual Coupling

- A virtual coupling = 6 DoF spring-damper
- Forces between dynamic object and haptic handle:

$$\mathbf{F} = k_T \mathbf{d} - c_T \mathbf{v}$$
$$\mathbf{M} = k_R \theta - c_R \omega$$

where

$$k_T, c_T = \text{transl. stiffness / viscosity}$$
$$k_R, c_R = \text{rot. stiffness / viscosity}$$
$$\mathbf{d}, \theta = \text{transl./rot. diplacement}$$
$$\mathbf{v}, \omega = \text{transl./rot. velocity}$$



- Details:
  - Represent all vectors in the handle's coordinate frame
  - Consider only that component of $\mathbf{v}$ that is in the direction of $\mathbf{d}$
  - Set viscosity to 0, if $\mathbf{v}$ points away from the handle

Simulation of the Motion of the Dynamic Object

- Total force acting on the dynamic object:

$$F = F_{spring} + \frac{1}{N} \sum_{i=1 \dots N} F_i$$

(Analogous for the torques)

- Integrate the following equations of motion:

$$F = ma$$
$$M = J\alpha + \omega \cdot J\omega$$

where

$$F, M = \text{force/torque acting on the center of mass}$$
$$a, \alpha = \text{translational/rotational acceleration}$$
$$m, J = \text{mass/inertia tensor}$$
$$\omega = \text{rotational velocity}$$

- Prerequisite: Δt is known in advance (e.g., because it is constant)

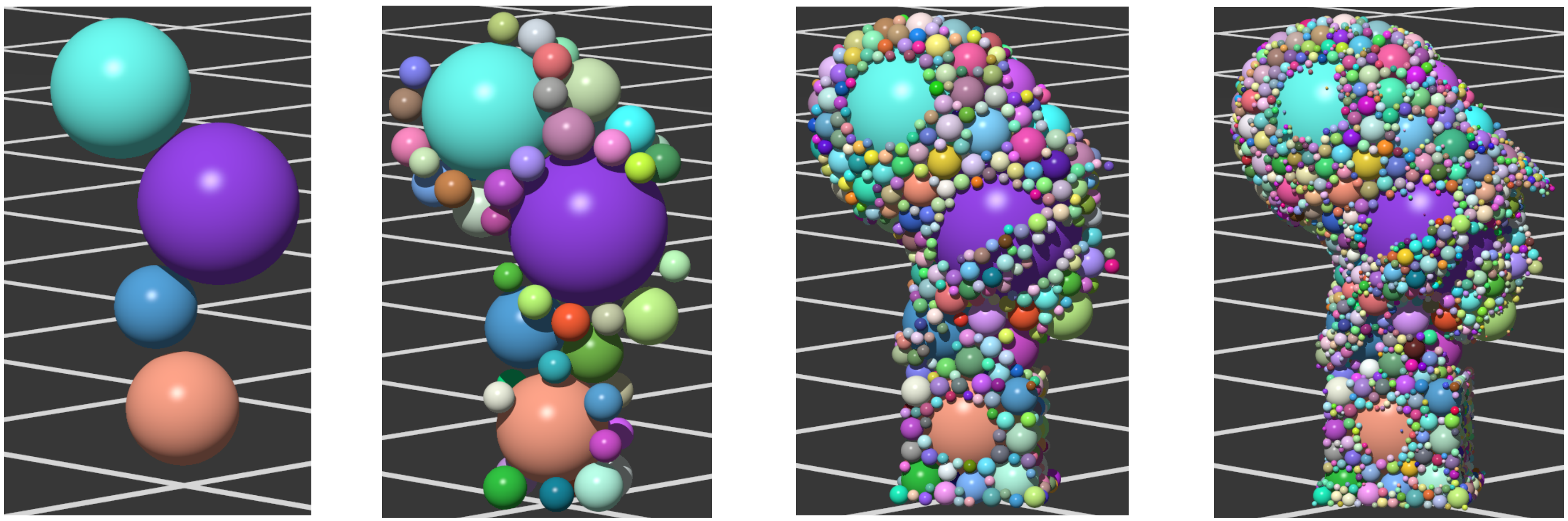
Overall Algorithm

1. Check collisions
2. Compute forces and torques of every point of the point shell
3. Compute total force on dynamic object
4. Compute the new acceleration on dynamic object
5. Computer new position of dynamic object
6. Compute forces on haptic handle mediated by virtual coupling

- Virtual coupling = low-pass filter

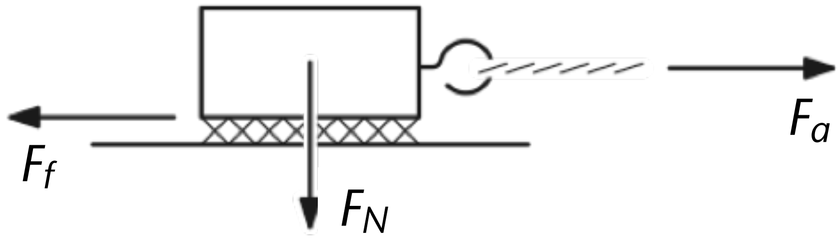
Another Method using Sphere Packings

- See Chapter on *Collision Detection*



Friction

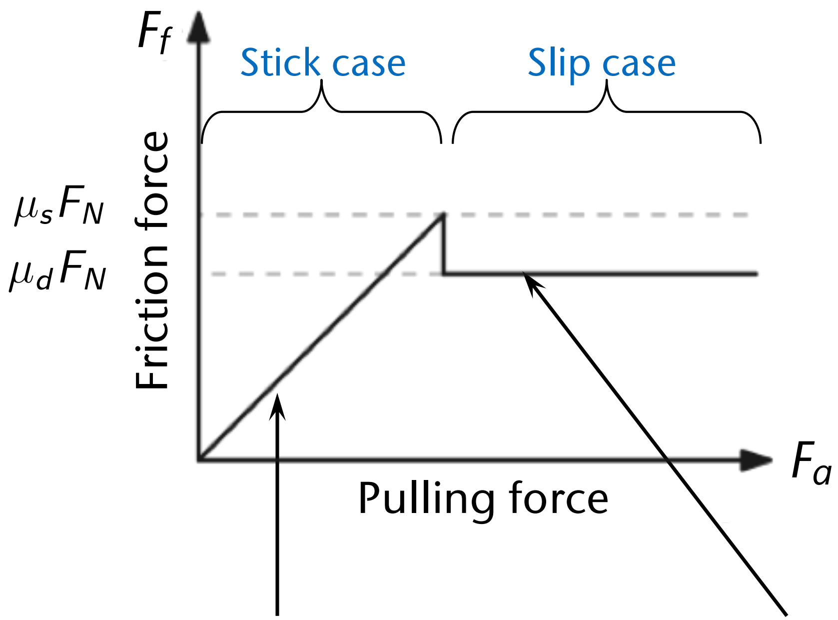
- Consider this situation:
  F_a = pulling force,
  F_N = force normal to surface,
  F_f = friction force



- Coulomb's Law of Friction:
  So long as

$$F_f = -F_a \leq \mu_s F_N$$

  the object will not move
  (stick case, Haftreibung).
  μ_s = static friction coeff.
  μ_d = sliding friction coeff.

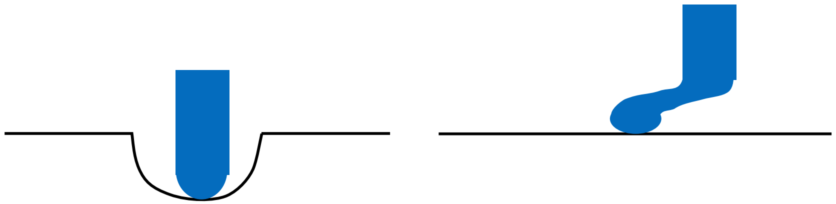


Static friction force balances pulling force, up to maximum specified by static friction coefficient

Once object begins moving, frictional force drops to constant value, called sliding friction or kinetic friction

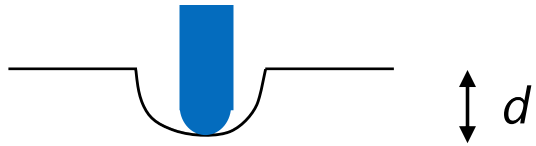
Friction in One Contact Point for Force Feedback

- The model:
  - Surface = membrane
  - Tool = laterally flexible stylus



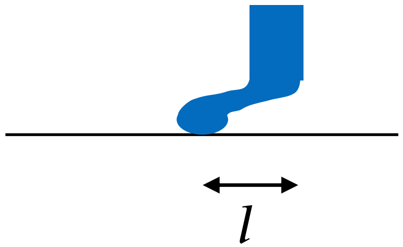
- *Point of Attachment*:
  - Point on the surface where first contact occurred
  - Alternatively, determined by the simulation
- Forces:
  - Force in direction of the surface normal:

$$F_N = k_N \cdot d$$



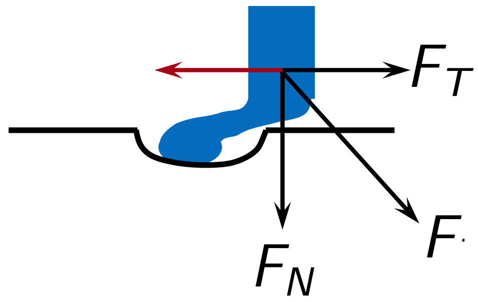
  - Force tangential to surface:

$$F_T = k_T \cdot l$$



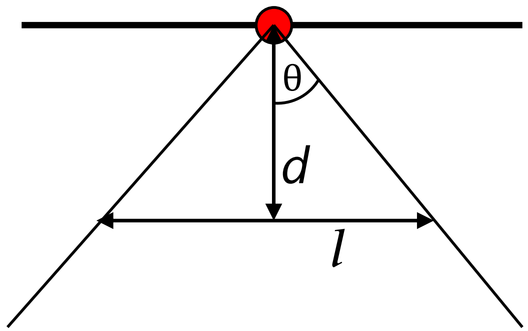
- The Coulomb friction model says:

$$F_f \overset{!}{\leq} \mu \cdot F_N = \mu \cdot k_N \cdot d$$



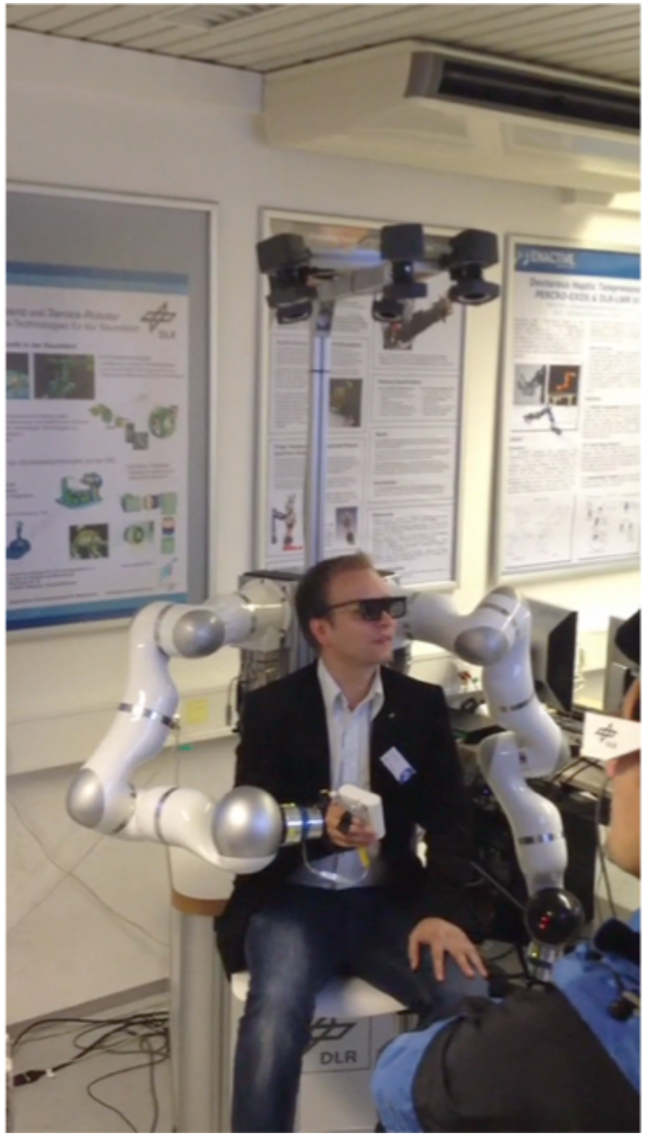
- The "cone of friction":
  describes the boundary between static friction and sliding friction (Gleitreibung; a.k.a. dynamic friction)

$$\text{obj slides} \;\Leftrightarrow\; F_T > F_f \;\Leftrightarrow\; k_T \cdot l > \mu \cdot k_N \cdot d \;\Leftrightarrow\; \frac{l}{d} > \mu \frac{k_N}{k_T}$$



$$\theta = \tan^{-1}\left(\mu \frac{k_N}{k_T}\right)$$

Application: On-orbit servicing of satellites



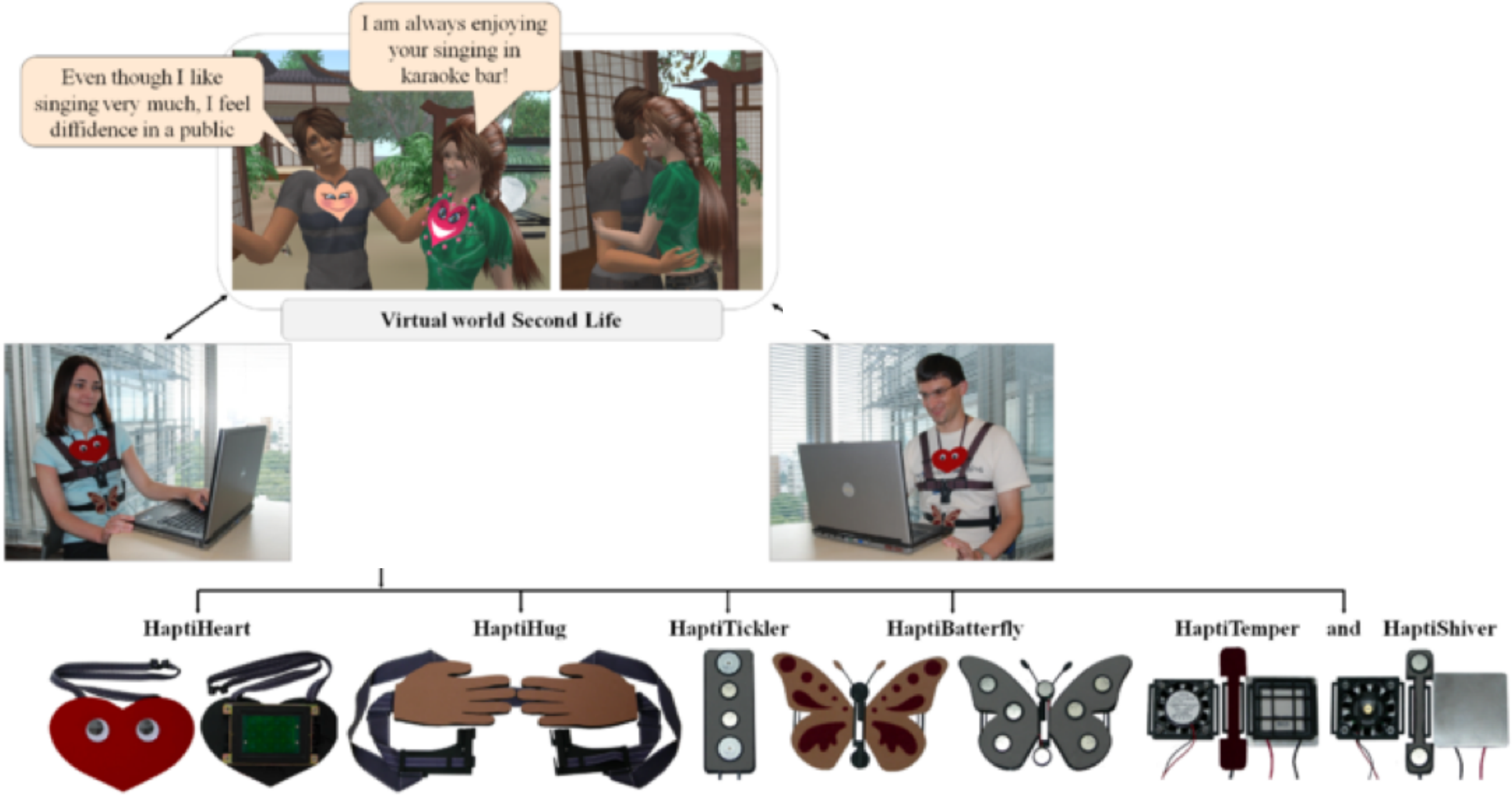
DLR, institute or robotics and mechatronics, Germany

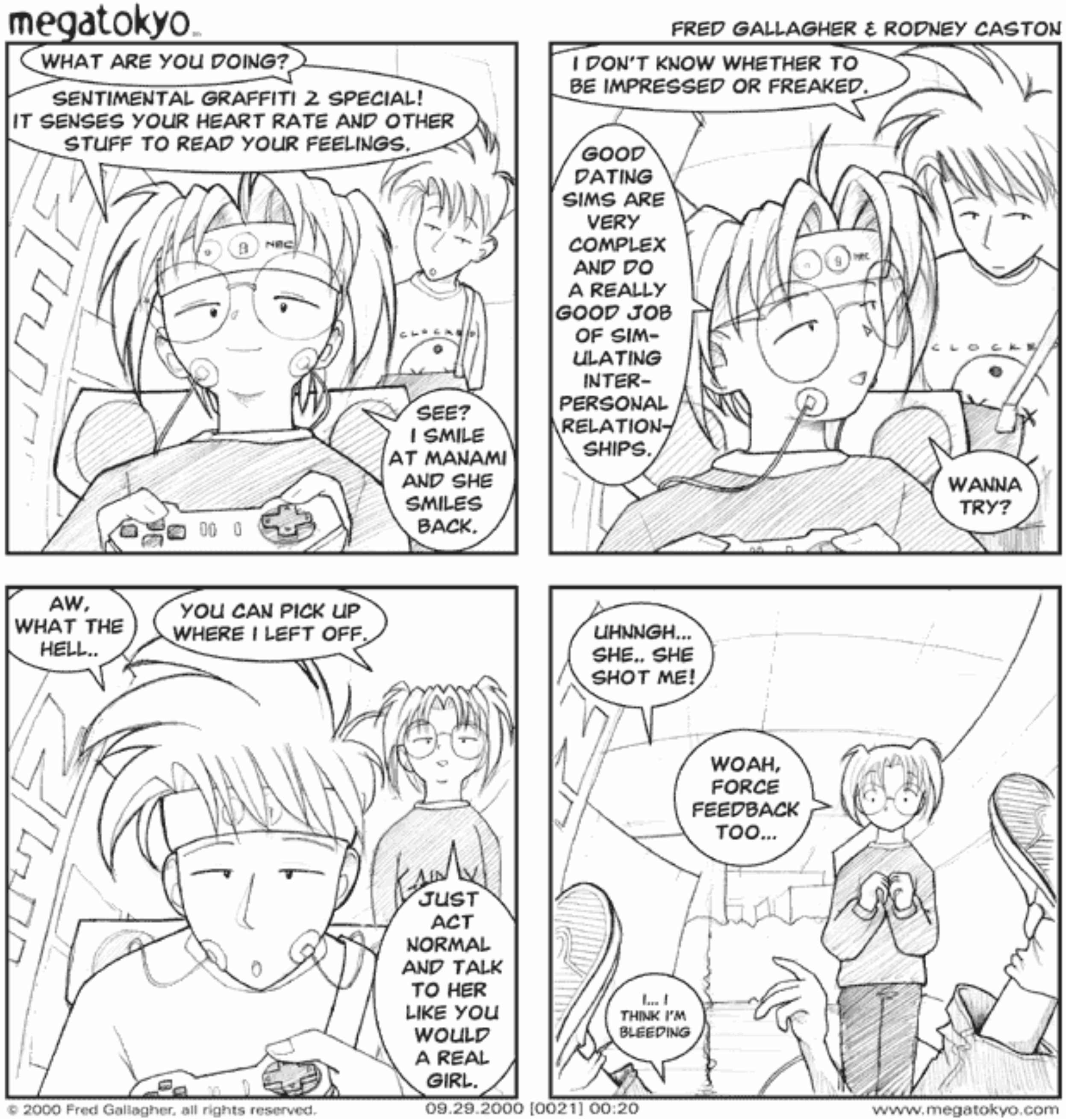
Future Applications of Force-Feedback Devices

- Micro-surgery (minimally invasive surgery) using remotely controlled robots:

DLR, institute or robotics and mechatronics, Germany

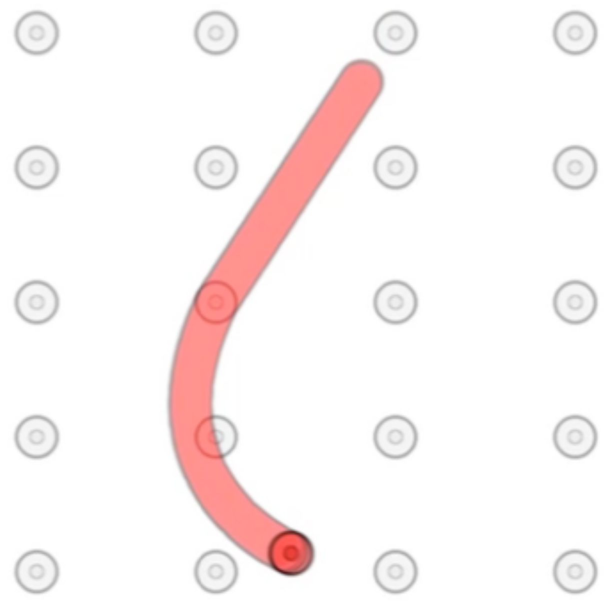
Affective Haptics




megatokyo
FRED GALLAGHER & RODNEY CASTON
WHAT ARE YOU DOING?
SENTIMENTAL GRAFFITI 2 SPECIAL! IT SENSES YOUR HEART RATE AND OTHER STUFF TO READ YOUR FEELINGS.
SEE? I SMILE AT MANAMI AND SHE SMILES BACK.
I DON'T KNOW WHETHER TO BE IMPRESSED OR FREAKED.
GOOD DATING SIMS ARE VERY COMPLEX AND DO A REALLY GOOD JOB OF SIM-ULATING INTER-PERSONAL RELATION-SHIPS.
WANNA TRY?
AW, WHAT THE HELL..
YOU CAN PICK UP WHERE I LEFT OFF.
JUST ACT NORMAL AND TALK TO HER LIKE YOU WOULD A REAL GIRL.
UHNNGH... SHE.. SHE SHOT ME!
WOAH, FORCE FEEDBACK TOO...
I... I THINK I'M BLEEDING
© 2000 Fred Gallagher, all rights reserved.
09.29.2000 [0021] 00:20
www.megatokyo.com

Haptic Illusions

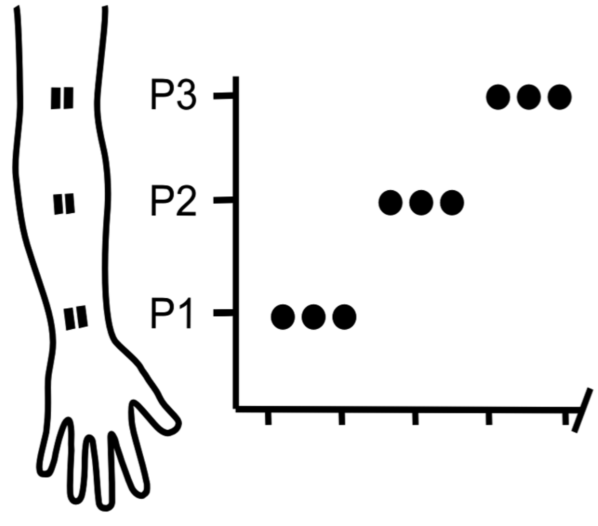
- There are not only optical illusions …

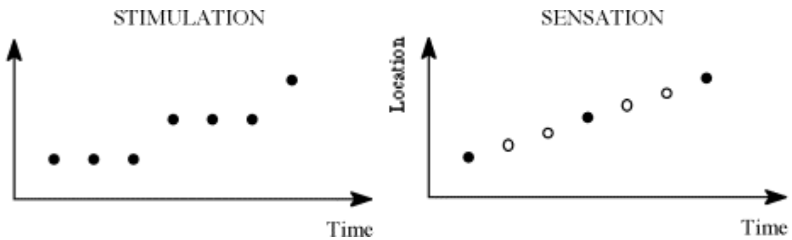


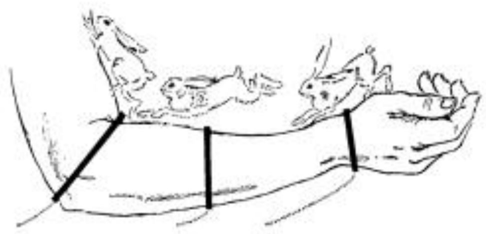
Surround Haptics Display / Haptic Chair by Disney Research, Pittsburgh

Cutaneous Rabbit Illusion

- Tap arm at 3 different positions, about 10 cm apart, 3 times at each position
  - Works also with electric pulses
  - Stimulus duration ≈ 5 ms , inter-stimulus interval = 50 ms
  - Subject has to close eyes and not get any other sensory input besides the taps
- Effect: subject perceives taps in between, like a (tiny) rabbit hopping up the arm

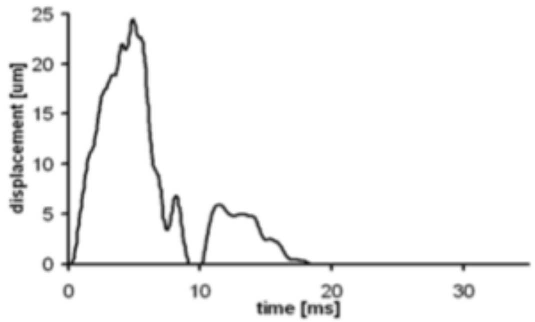


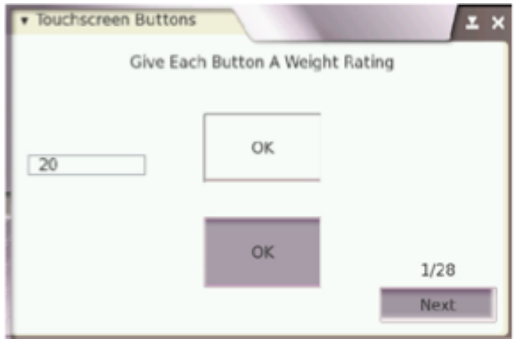


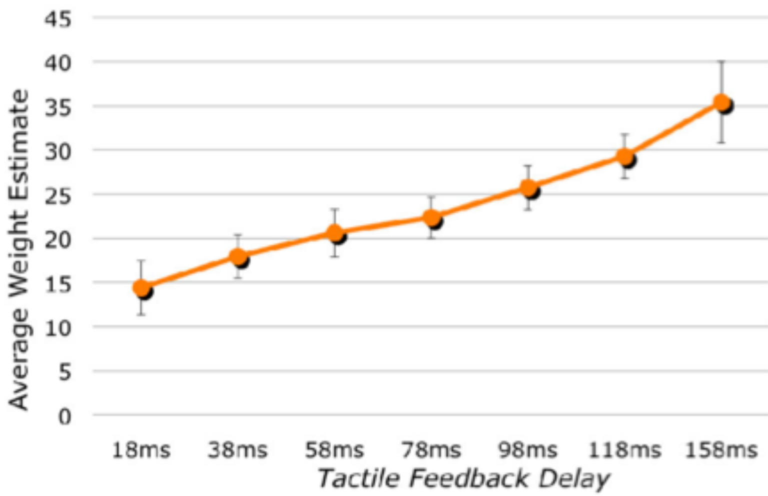


The Illusion of Heavy Buttons

- Experiment:
  - Tactile pulse when user pressed button on touchscreen
  - Delays for pulse: 18, …, 158 msec after click
  - Subjects were asked to assign a weight each time, relative to a baseline they defined themselves with the first click
- Results:







The Rubber-Hand Illusion

- Shows how important haptics is to create the illusion of body ownership, embodiment, and presence