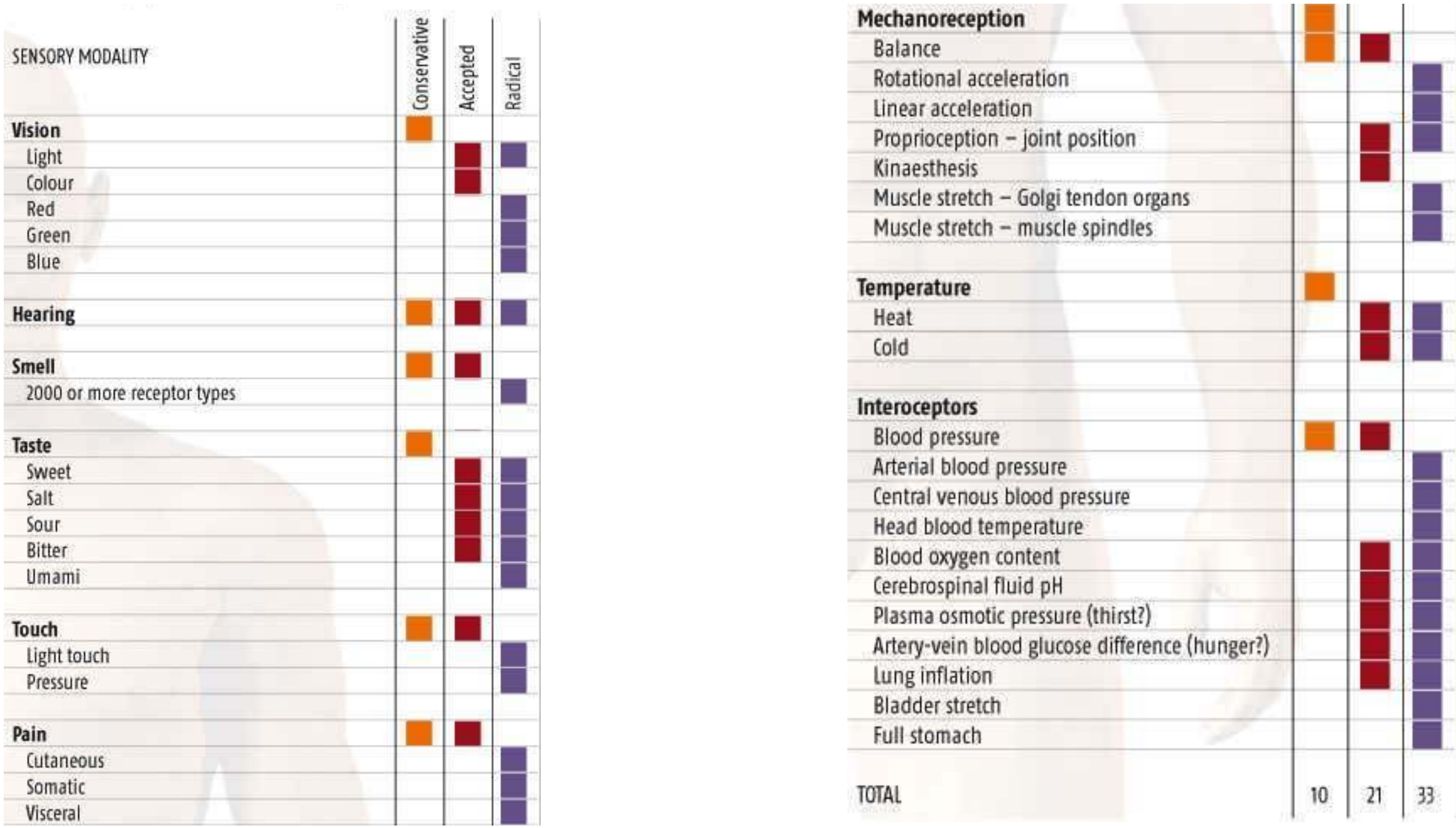
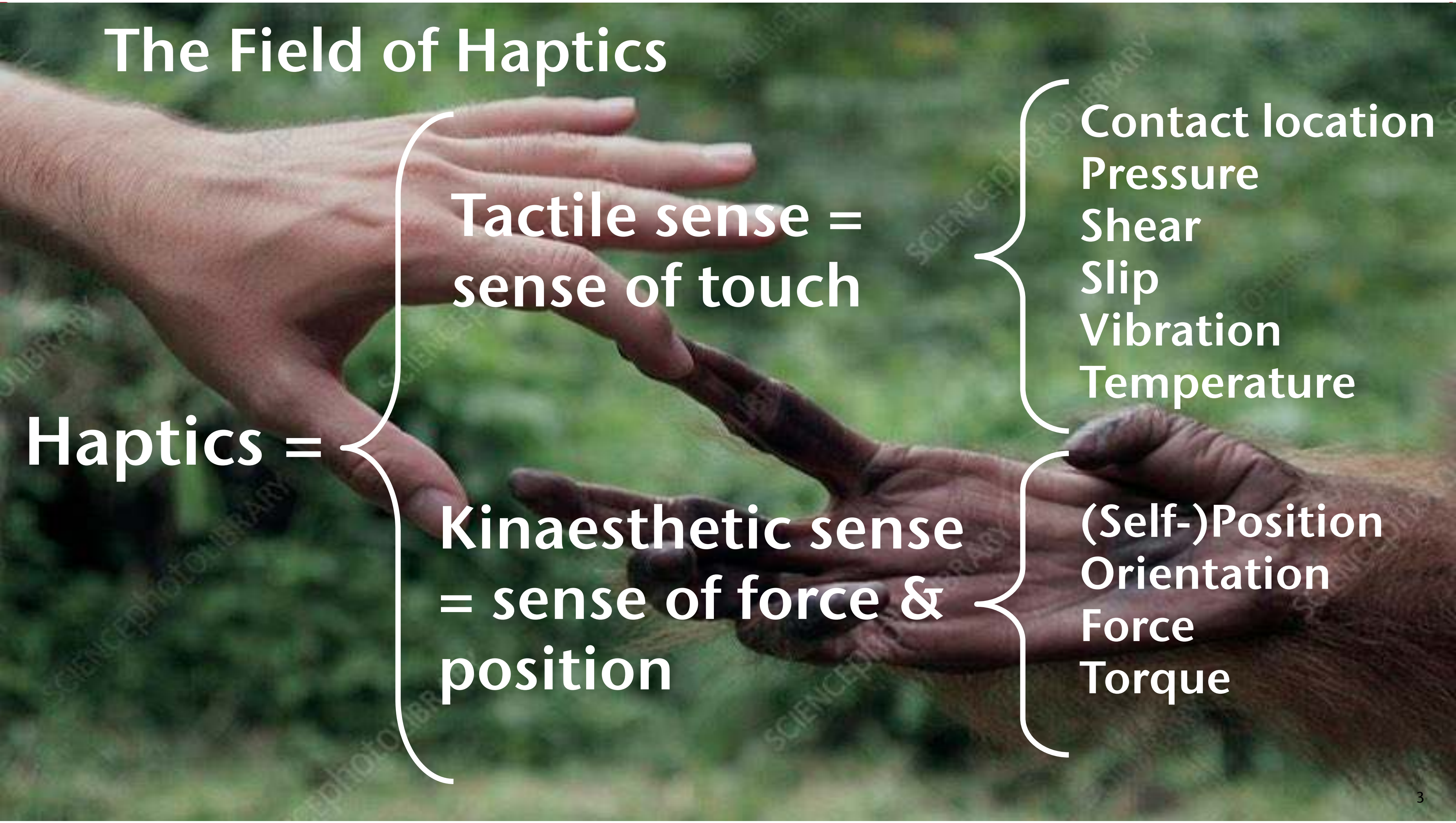
Virtual Reality & Physically-Based Simulation
Haptics

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

How Many Senses Do We Have? There Are Many Opinions …



[New Scientist, 2005]


The Field of Haptics
Haptics =
Tactile sense = sense of touch
Contact location
Pressure
Shear
Slip
Vibration
Temperature
Kinaesthetic sense = sense of force & position
(Self-)Position
Orientation
Force
Torque
3

What VR Systems Should Render

- Forces on the user's fingers / hand / arm (= haptic "image" of objects) → input to the user's muscles = force feedback / kinaesthetic feedback
- Haptic texture of surfaces (roughness, grain, friction, elasticity, ...) → input to the sensors under the user's skin = tactile feedback
  - Some people differentiate between tactile and vibrotactile feedback
- 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?"

Reminder: Rubber-Hand Illusion

- Shows how important visuo-tactile synchronicity is to create the illusion of body ownership, embodiment, and presence!

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

Exoskeletons

CyberForce

CyberForce

Sarcos

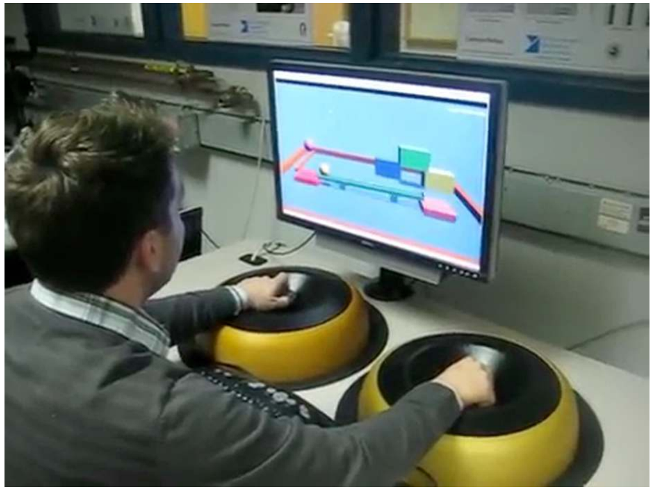
Stylus-like Point Probes

Phantom (3 DOF's in/out)

Virtuose (6 DOF's tracking, 3 DOF's force)

Hand-Held 6-DoF Probes (Tracking & Force) With/Without Props

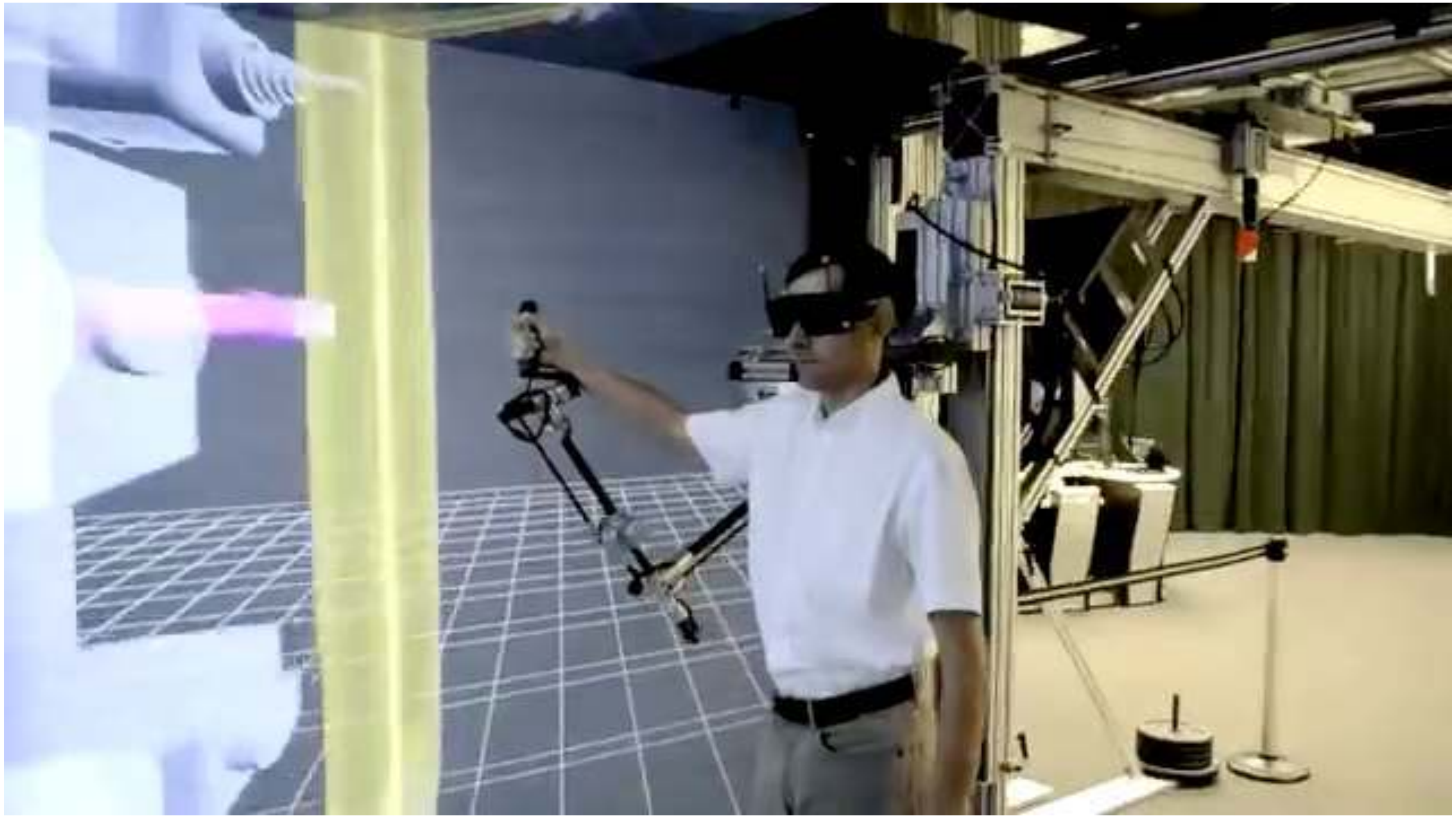
Force Dimension



Maglev (Butterfly Haptics)

KUKA light-weight robot

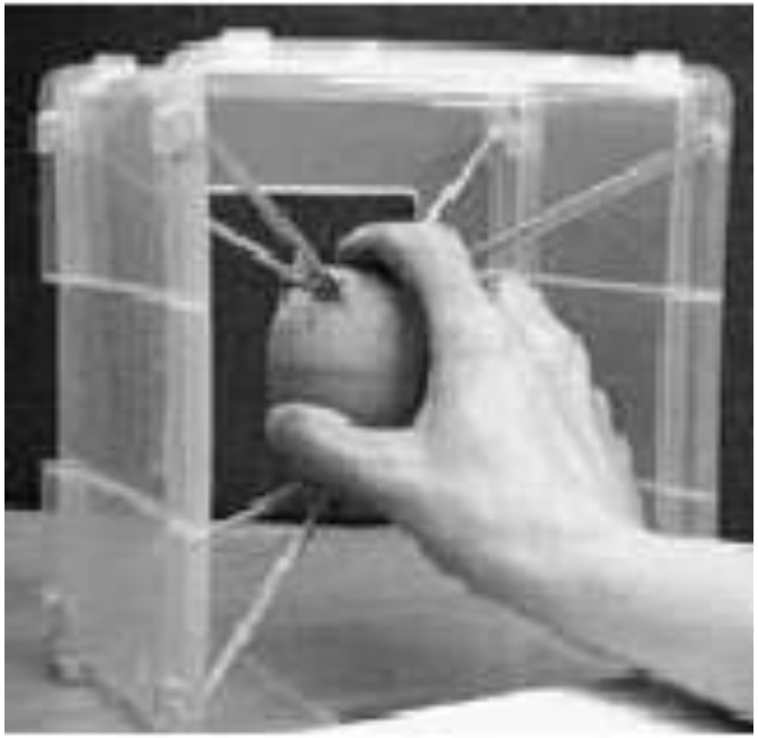
LapSim



Scale-1 by Haption

Tension-Based Force Feedback via Wires (Spidar Variants)

INCA 6D by Haption


Spidar

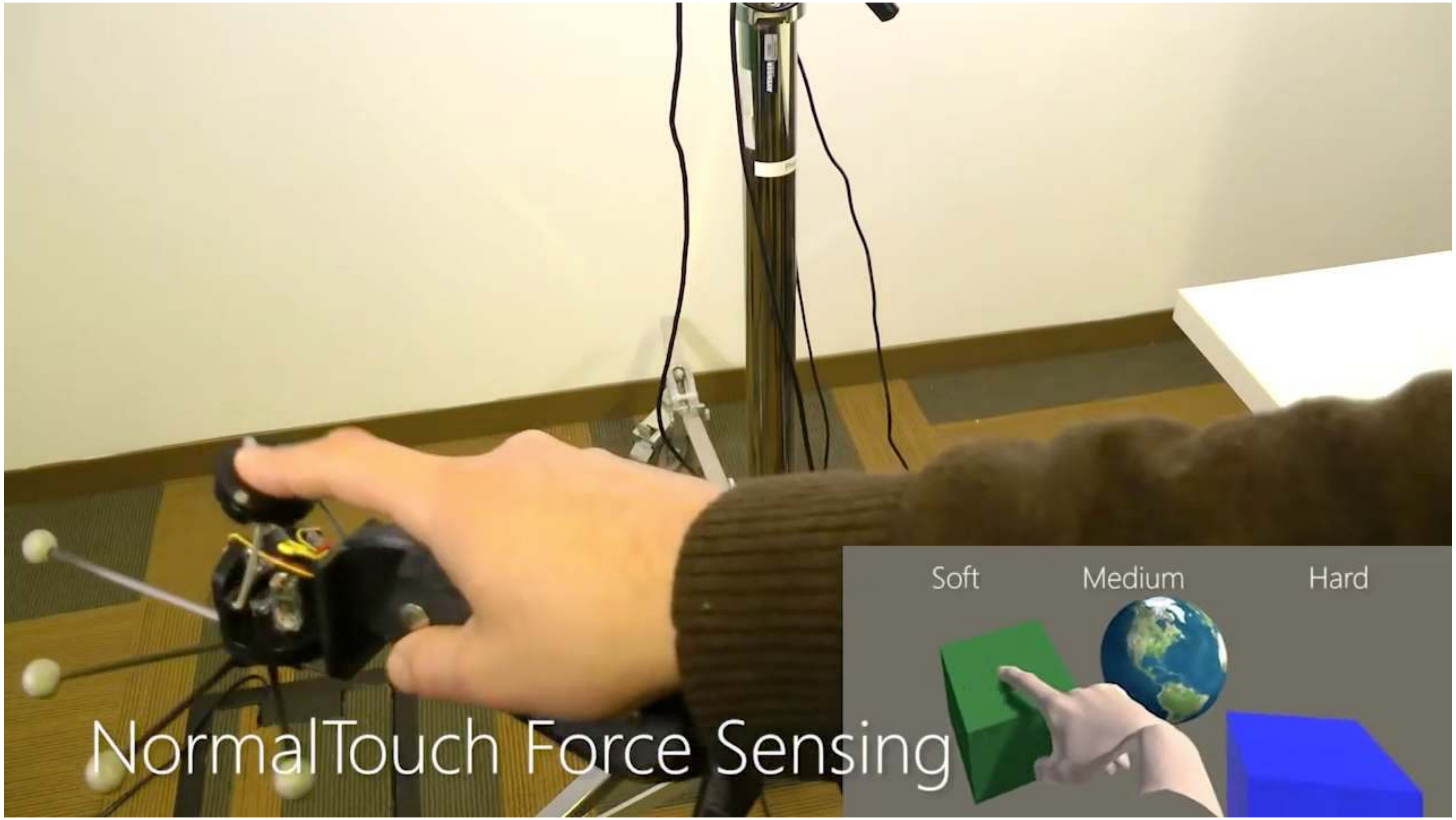
Finger- and Thimble-Devices

[Tsukuba]

(movies)

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

Wearable Haptics



NormalTouch & TextureTouch, 2016, Microsoft

Application: Reachability Checks in Manufacturing

More Distinctions / Categorizations

- Biomimetic / non-biomimetic devices: biomimetic devices move similar to the human body (example: exoskeleton)
- Passive vs. Resistive vs. active devices:
  - Passive: objects sitting at the correct positions providing feedback
  - Resistive ones use all kinds of brakes to restrict a user's motion
  - Active ones use motors to create motion/forces by themselves

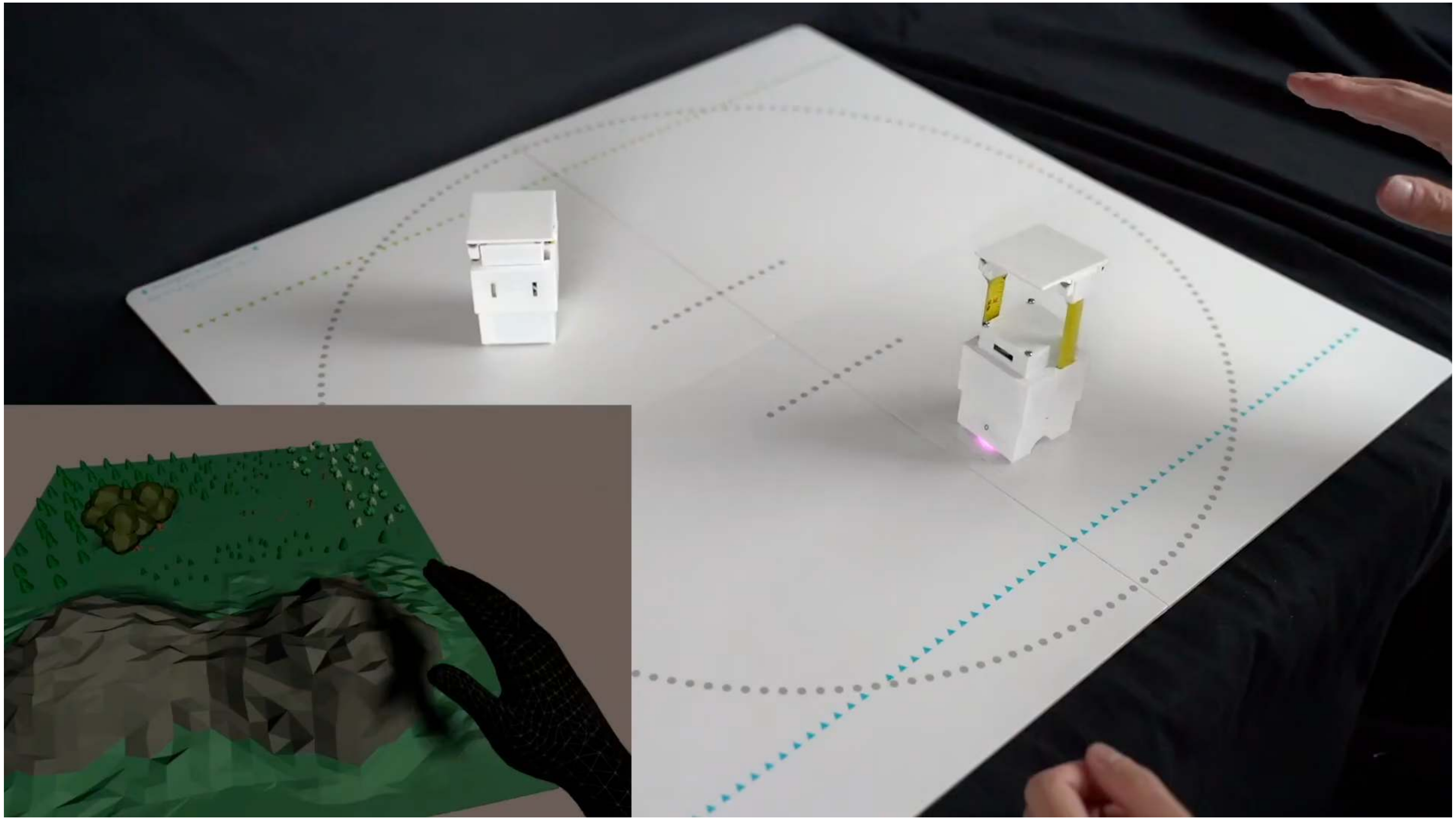
Passive Haptics (a.k.a. Encountered-Type Haptics)

Example: EncounteredLimbs

[Arata Horie et al., 2021]

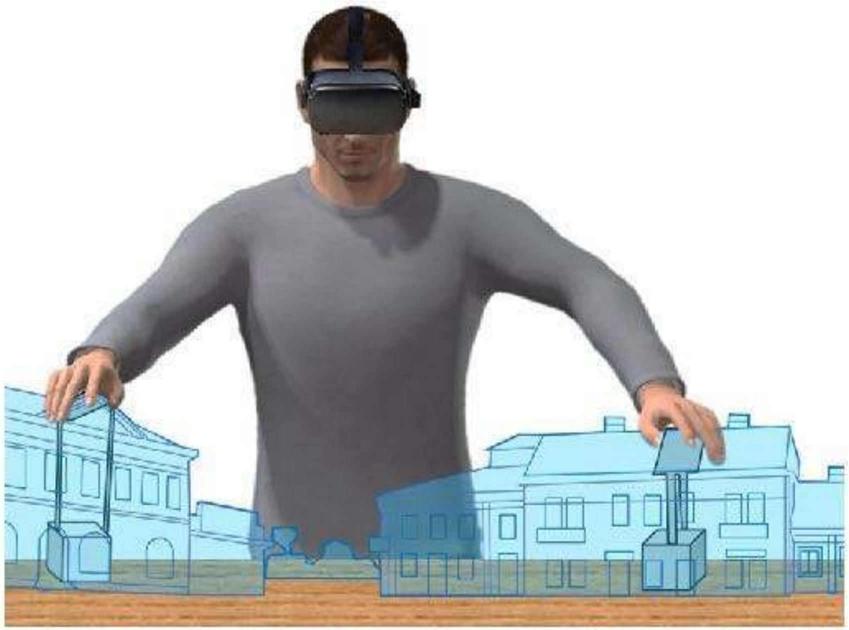
In Ergonomics Applications, Passive Haptics Would be Helpful

Example: HapticBots

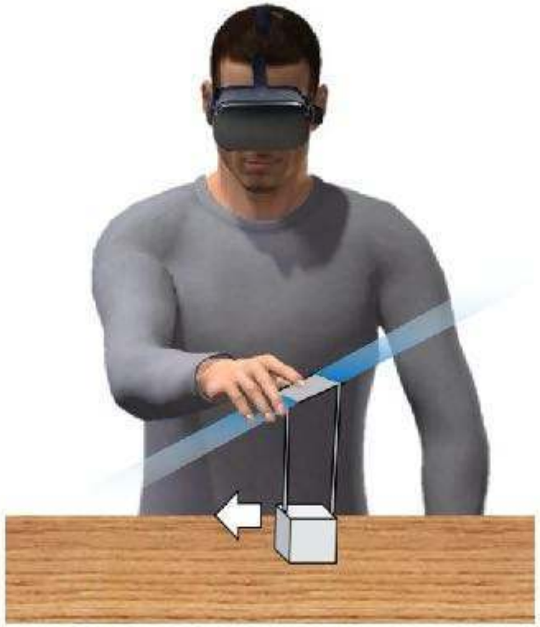


[Ryo Suzuki et al., 2021]

There is a multitude of similar technologies, such as moving furniture to the right place using a moving robot platform, or moving small room dividers around using a vacuum cleaning robot



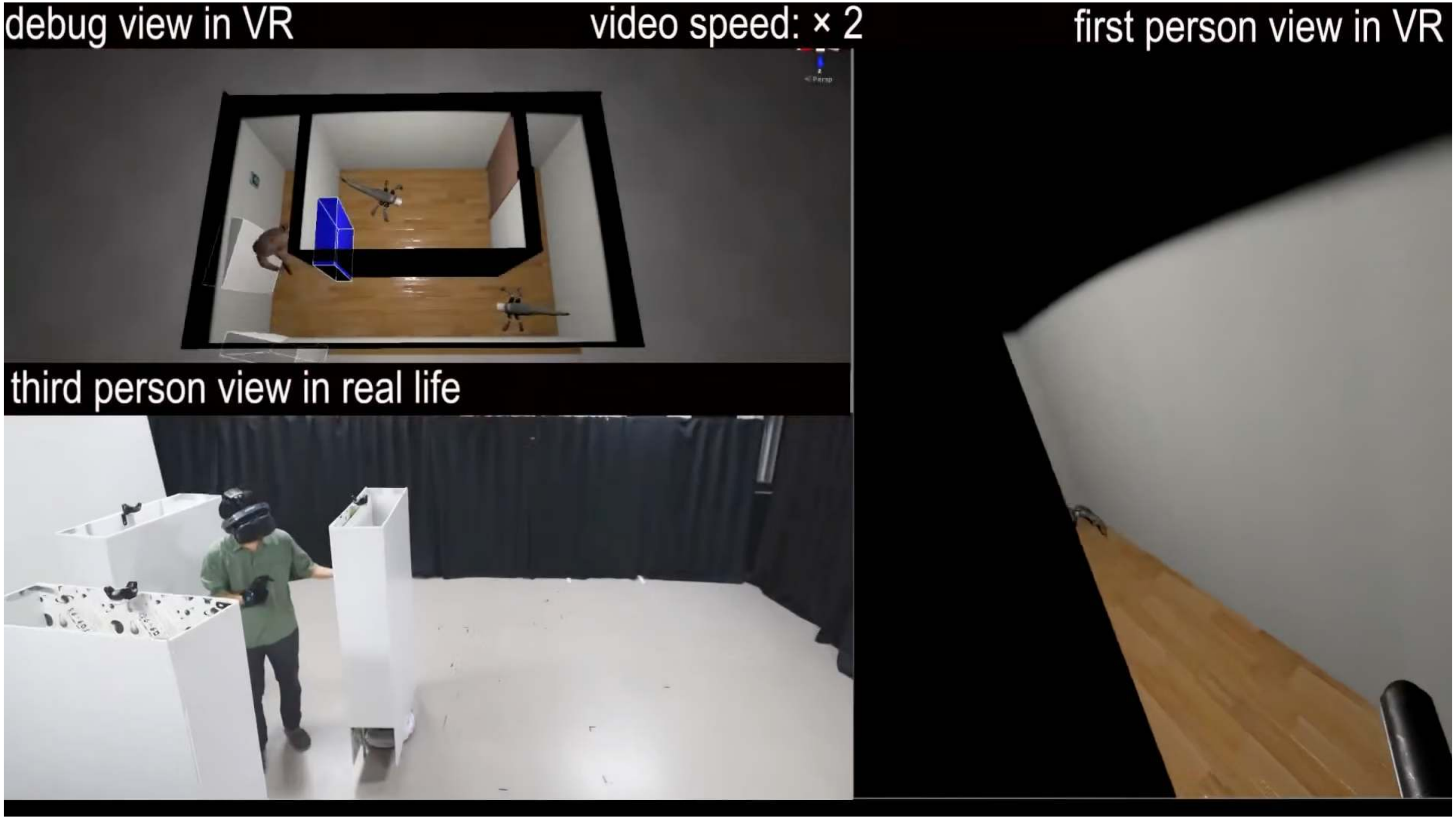
Large and flexible interaction area



Lateral and Continuous motion

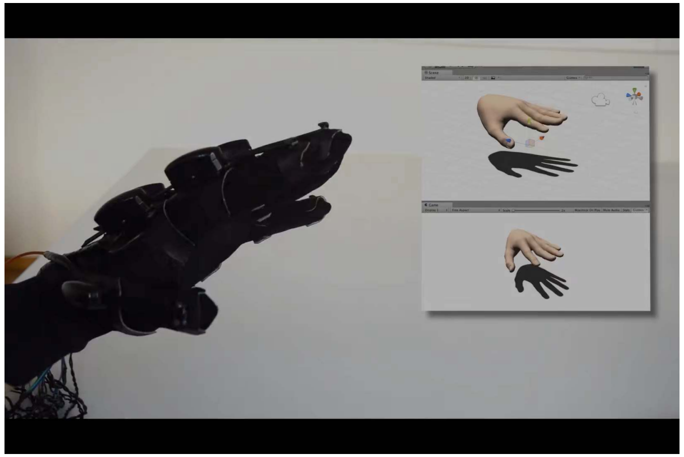
Swarm behavior

Passive Feedback for Virtual Walls and Doors

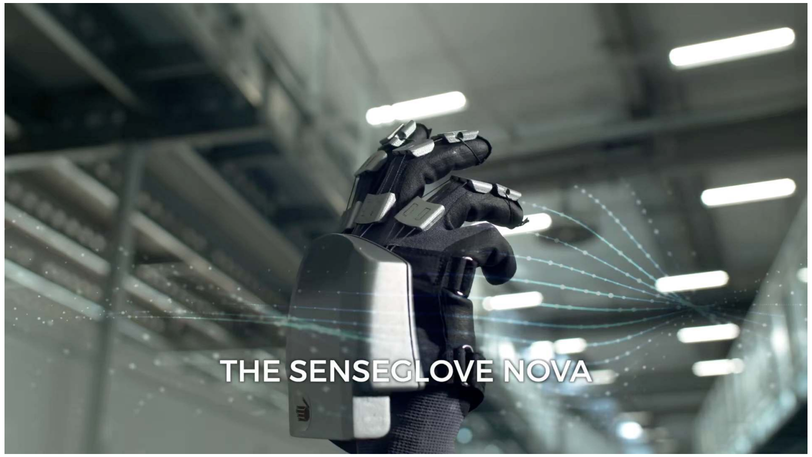


[ZoomWalls, 2020]

Haptic Gloves



HEAVE (uses tendons)

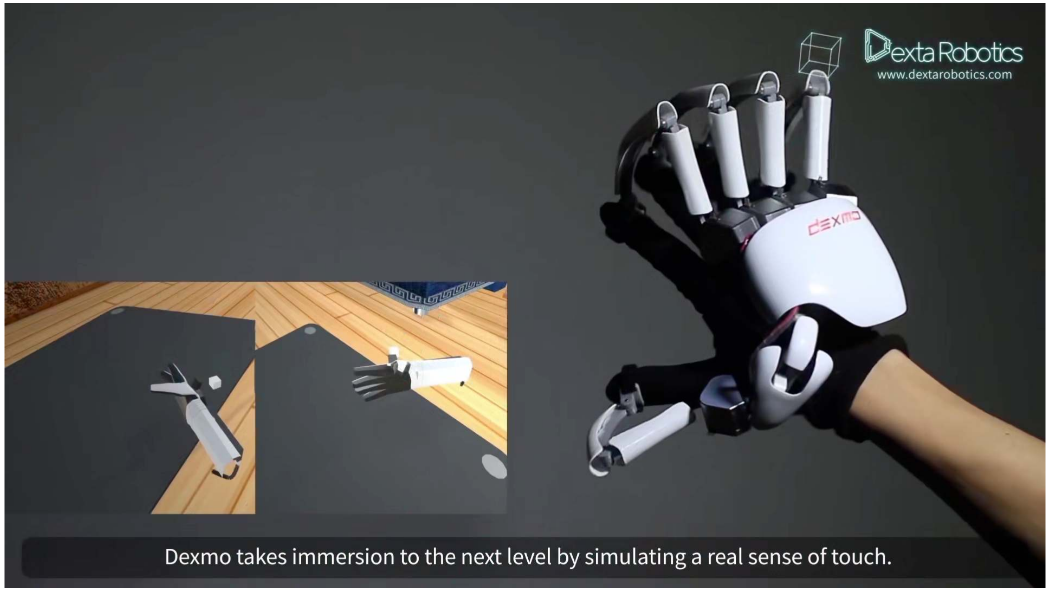


Sense Glove (tendons)

HaptX (armored exoskeleton, 100 actuators)

VRgluv (armored exoskeleton)

Cynteract (uses tendons for force-feedback)

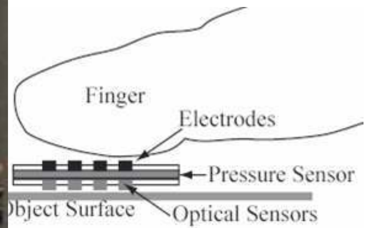


Dexmo by Dexta Robotics (exoskeleton)

Vibro-Tactile Displays in Consumer Electronics

Tactile Displays

SmartTouch



Vibrators for fingers

CyberTouch

GloveOne

"Haptic vest"

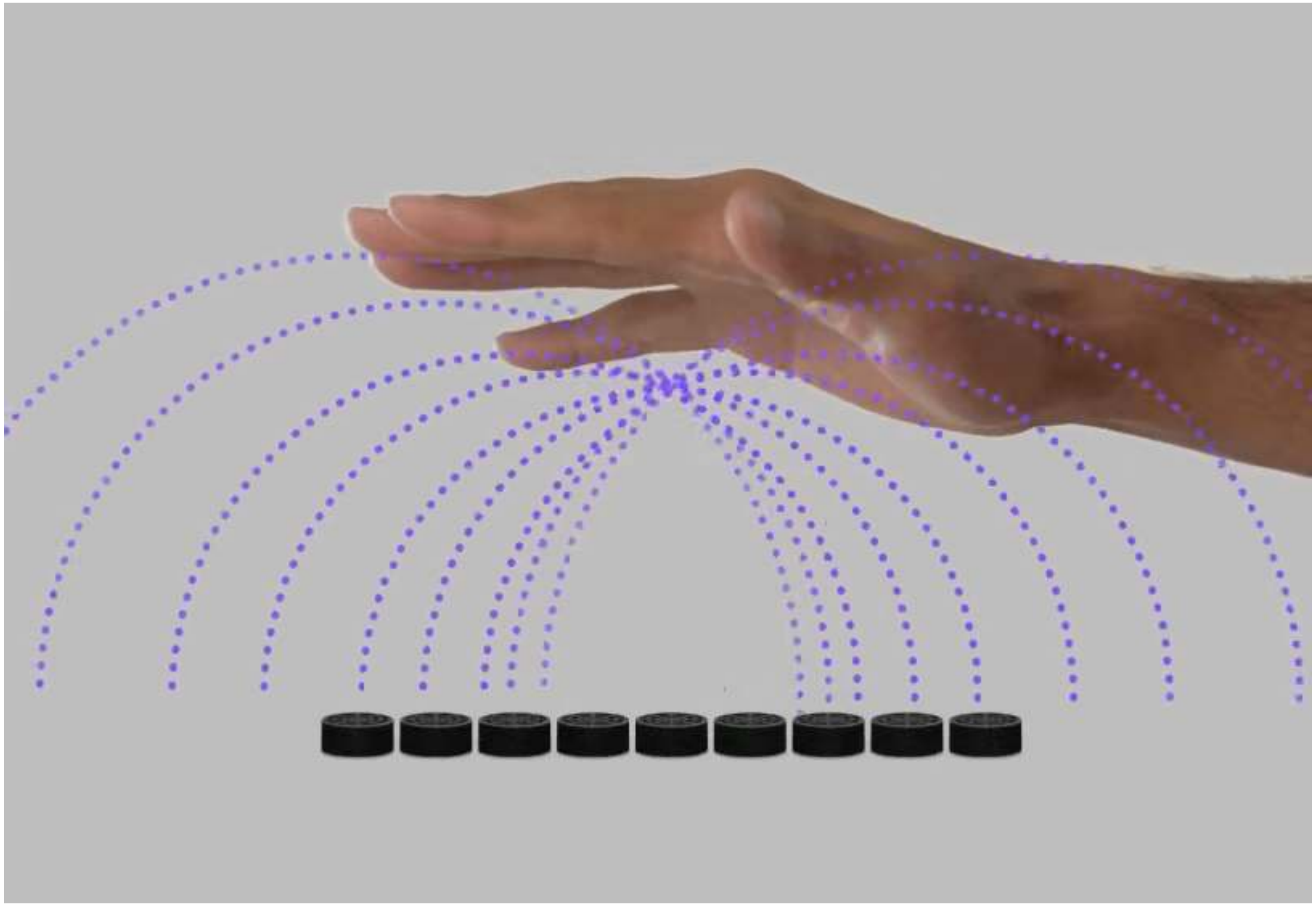
Aura Interactor (~ 1995)

Feelex (haptic surface)

Mid-Air Tactile Display using Ultrasound

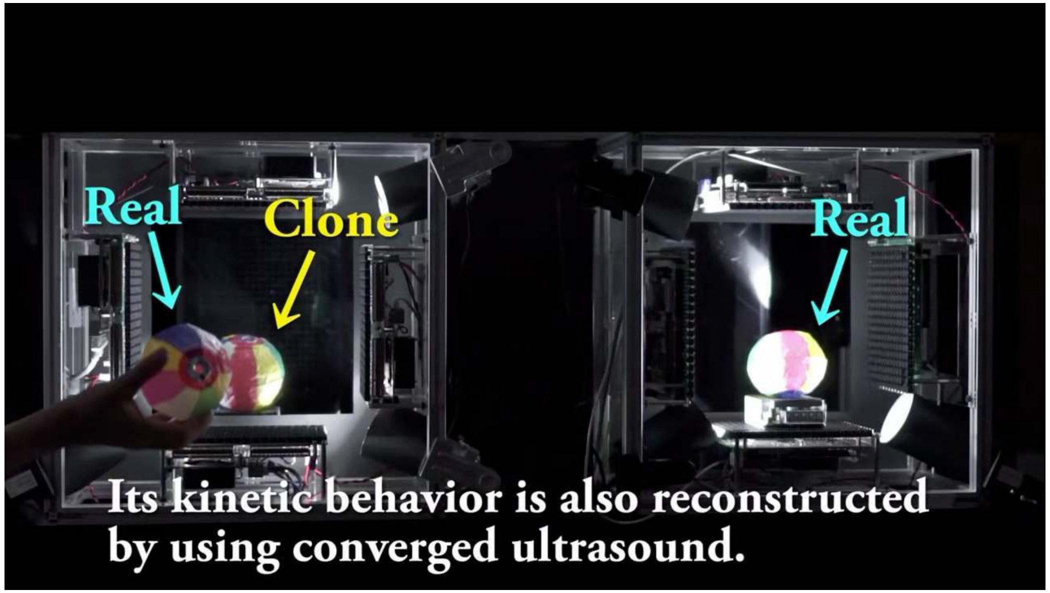
Tactile Feedback via Interference of Ultrasound

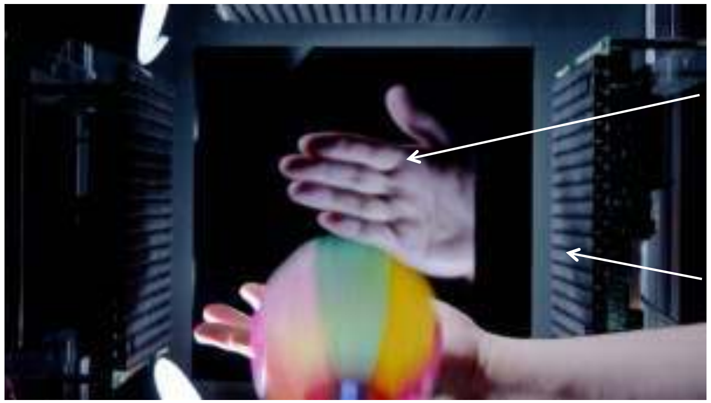
Basic Principle: Ultrasound Speakers Array Emitting with Different Phases



[ultraleap.com]

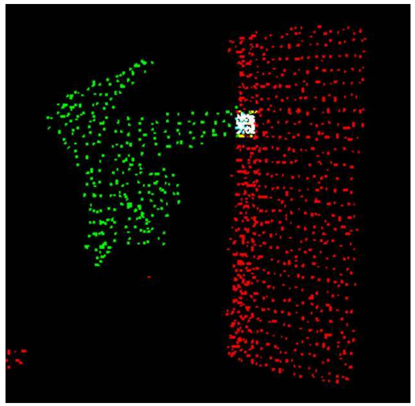
Tele-Haptics: Example *Haptoclone*





Mirrored objects from other box

1992 phase-controlled ultrasound array



Depth sensor for objs

Tactile and Force Feedback via Fans

Birdly

LevioPole, Inami et al.

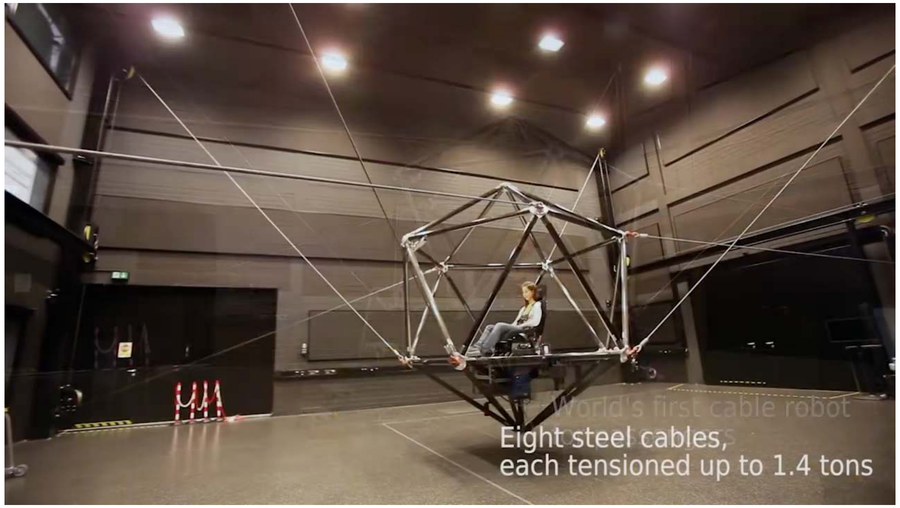
Whole Body Suits

Electro muscle stimulation
Transcutaneous electrical nerve stimulation
Motion capture (using IMU's)
Heart beat measurements

Teslasuit

Motion Platforms (Not Really Force-Feedback!)

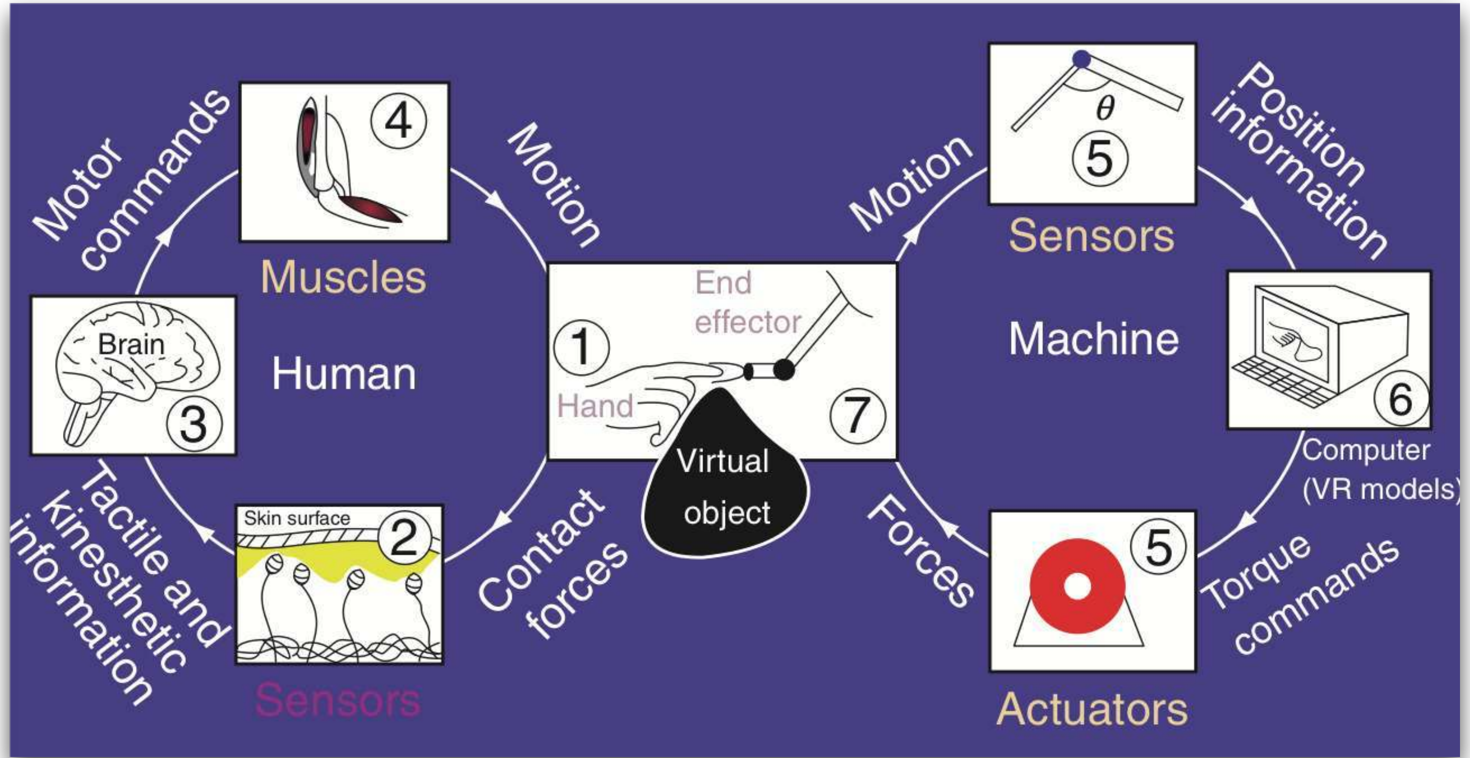
technicolor



MPI Tübingen

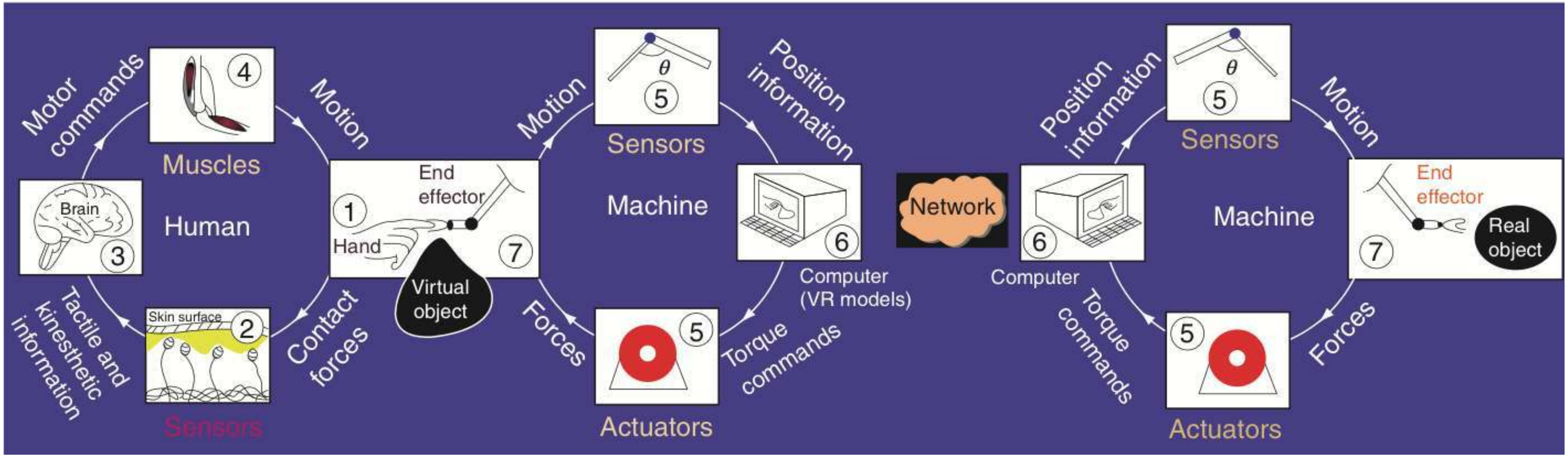
The Special Problem of Force-Feedback Rendering

The tight human-in-the-loop



[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

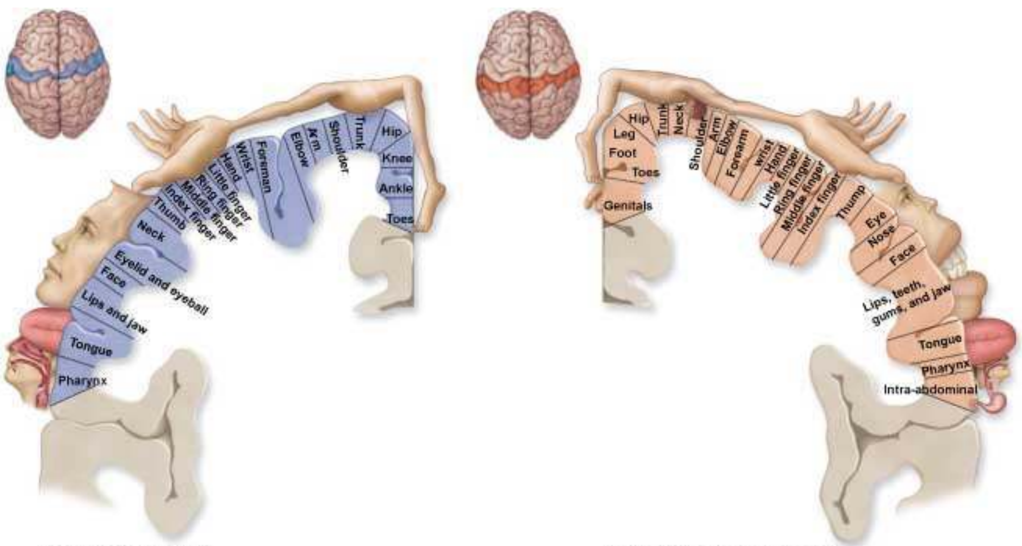


[M A Srinivasan & R Zimmer: *Machine Haptics*. New Encyclopedia of Neuroscience, Ed: Larry R. Squire, Vol. 5, pp. 589-595, Oxford: Academic Press, 2009]

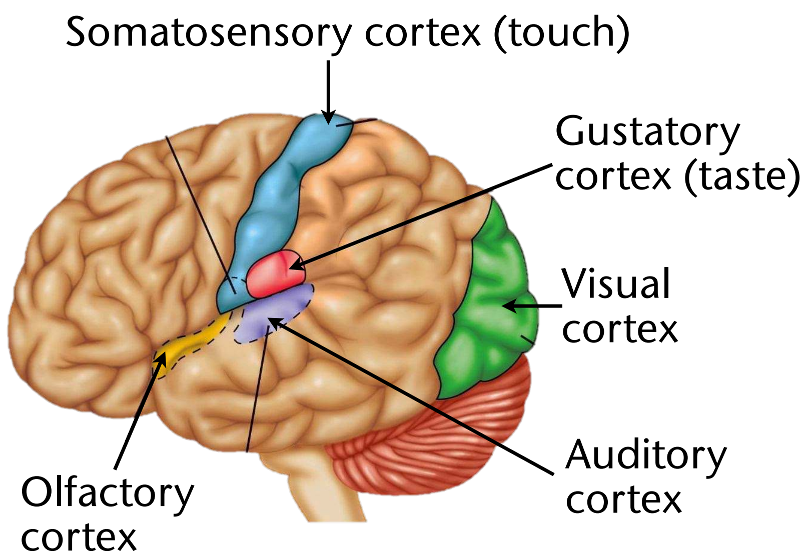
Putting the Human Haptic Sense Into Perspective

- 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

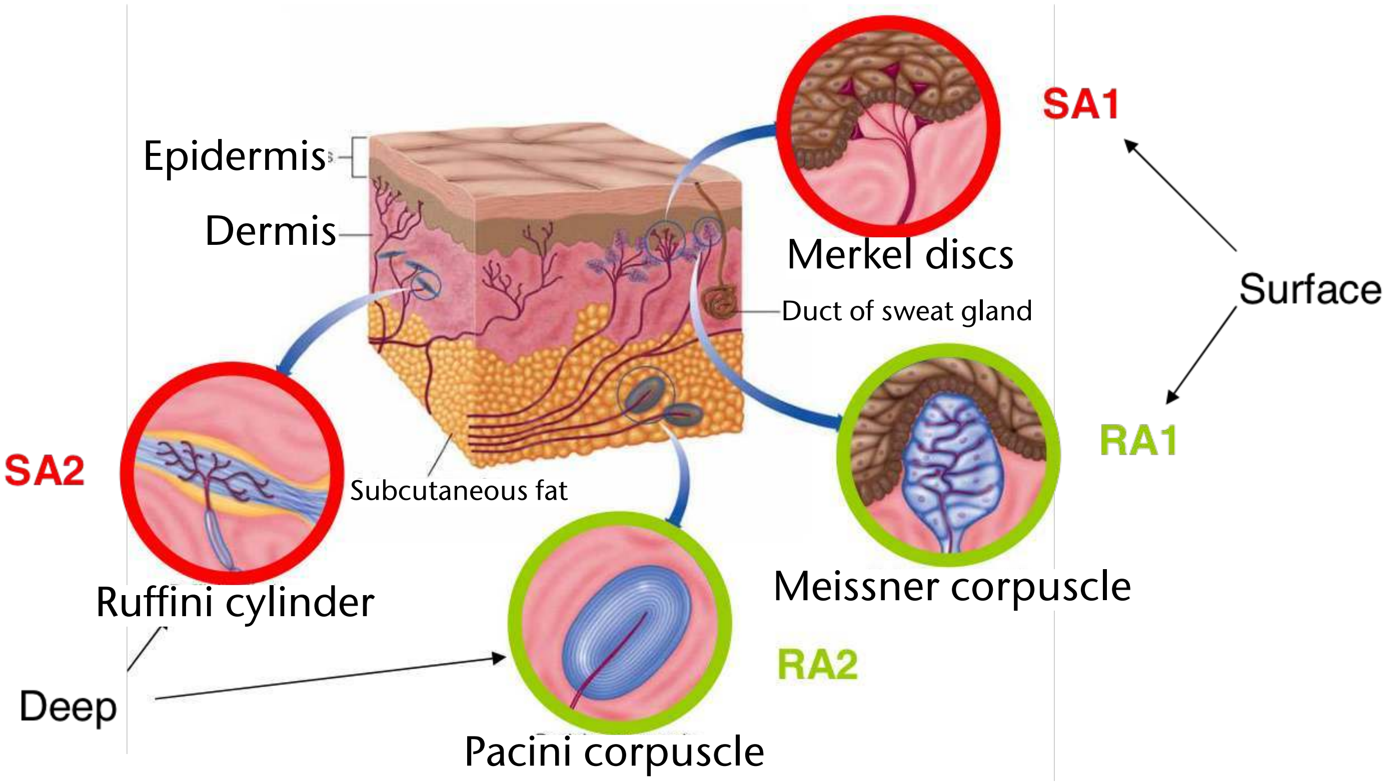


a. Primary motor area
b . Primary somatosensory area



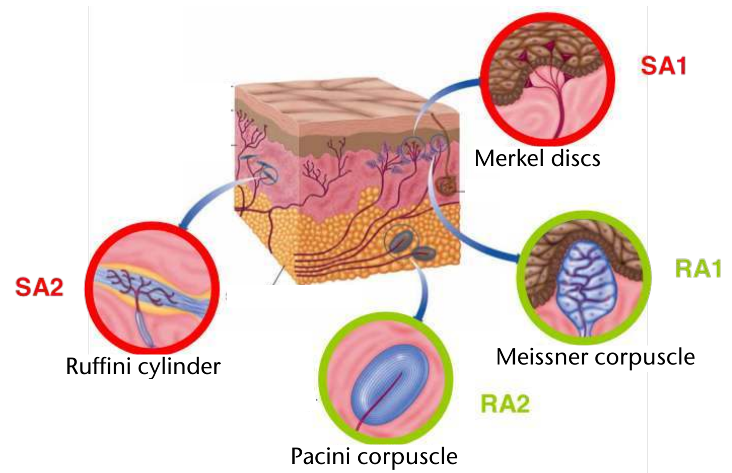
The Human Tactile Sensors

- 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 when stimulus changes



Adapting Rate

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

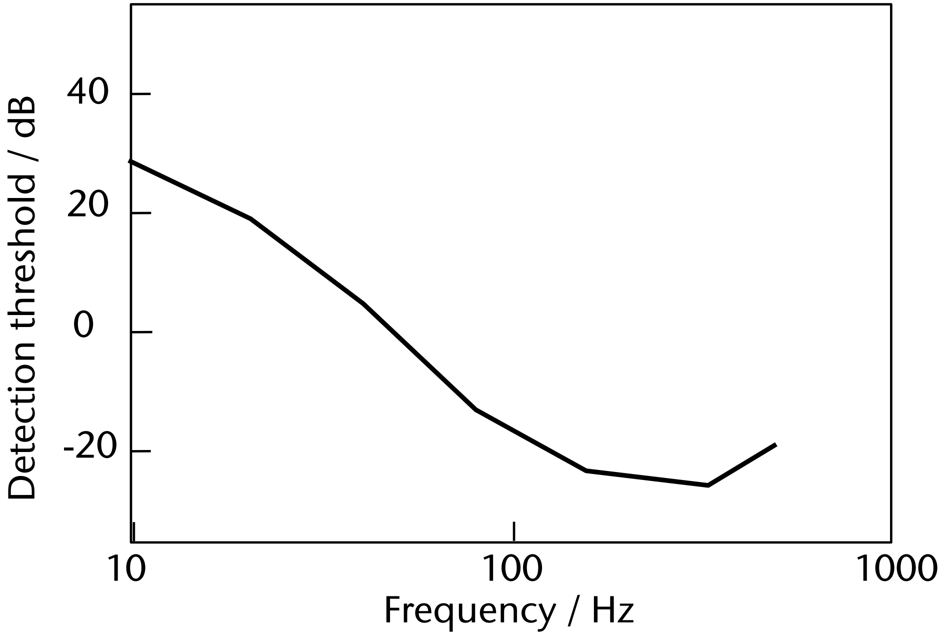
Some Human Factors Regarding Haptics

- 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 (i.e., 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)

- Forces of hand/arm:
  - Max. 50-100 N
  - Typ. 5-15 N (manipulation and exploration)
  - Just noticeable difference: $\text{JND} = \left| \frac{F_{ref} - F_{comp}}{F_{ref}} \right| = 0.1$ (10%)
- Lag 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

Factors Affecting Simulations (Hardware & Software)

- Detection threshold for vibrations:

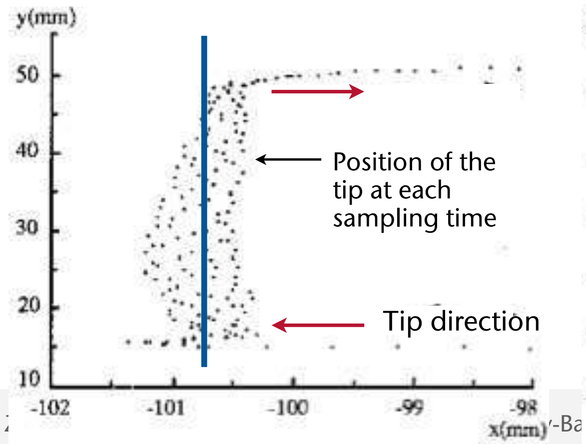


- Simulation must run at Nyquist frequency ⟶ in order to generate haptic signals with 500 Hz, the simulation loop must run at 1000 Hz

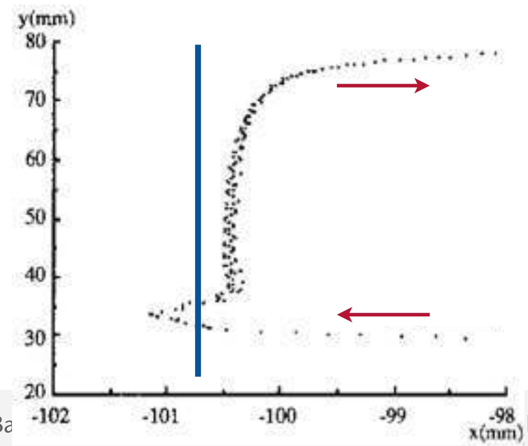
Rule of Thumb: 1000 Hz Update Rate Needed for Haptic Rendering

- 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

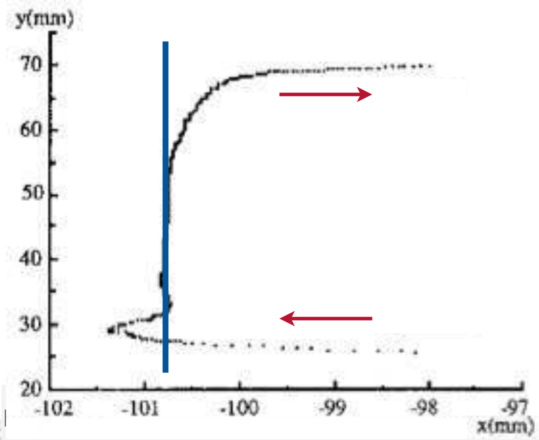
167 Hz

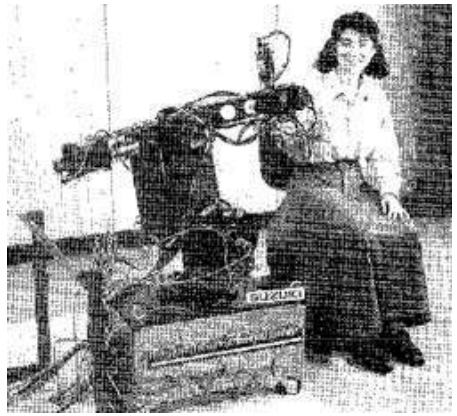


250 Hz



500 Hz



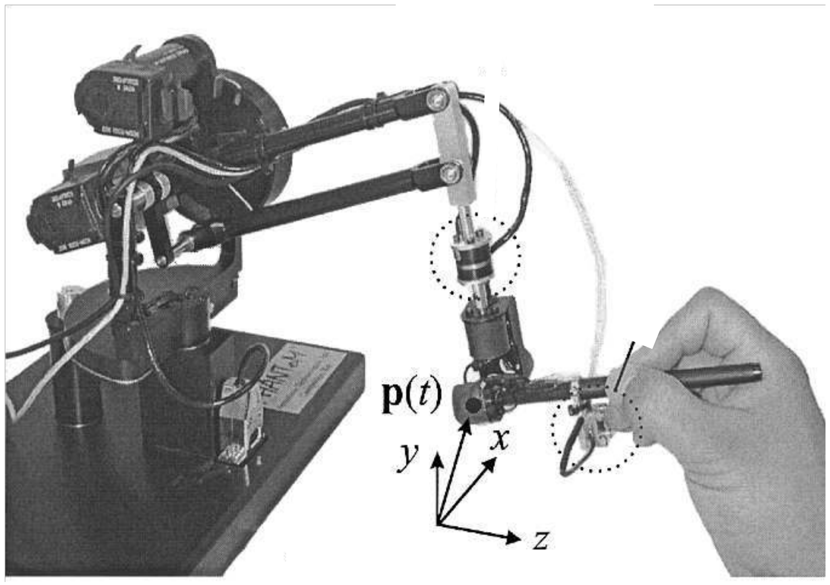


Stiffness

- Rule of thumb to generate the sensation of stiffness/rigidity: in order to render hard surfaces, you need >1 N/mm (better yet 10 N/mm)

Haptic Textures

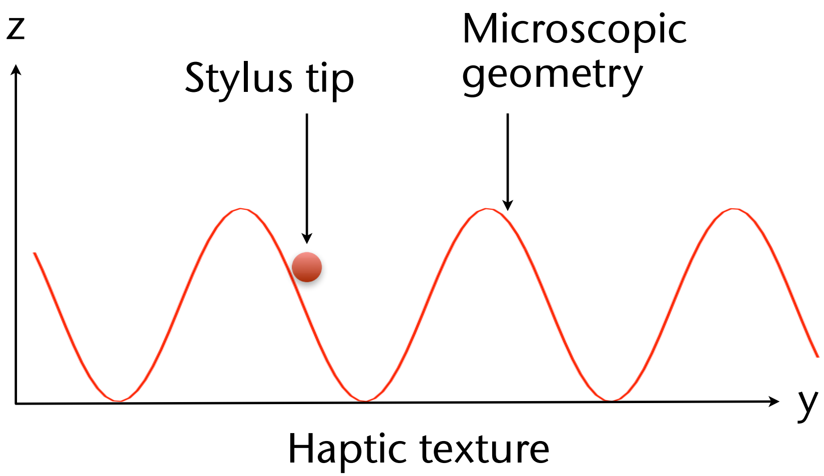
- Texture = fine structure of the surface of objects (= micro-geometry); independent of the shape of an object (= macro-geometry)
- Haptic textures can be sensed in two ways by touching:
  - Spatially
  - Temporally (when moving your finger across the surface)

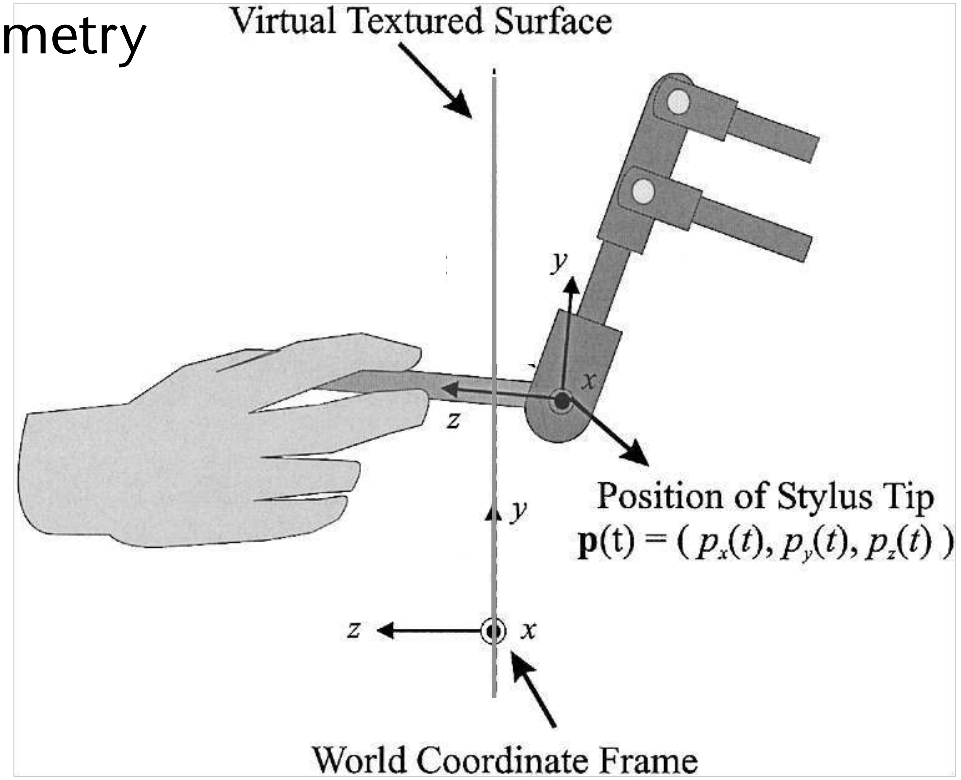


- Sensing haptic textures via force-feedback device: as you slide the tip of the stylus along the surface, texture is "transcoded" into a temporal signal, which is then output on the device (e.g., use IFFT to create the signal, or sample surface dircetly)

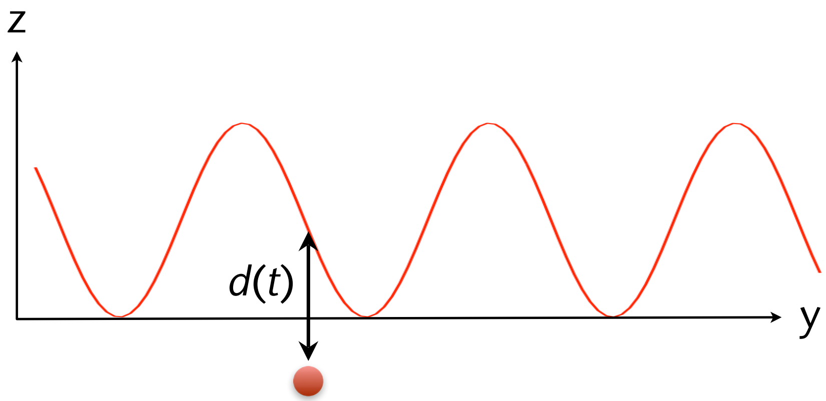
A Frequent Problem: "Buzzing"

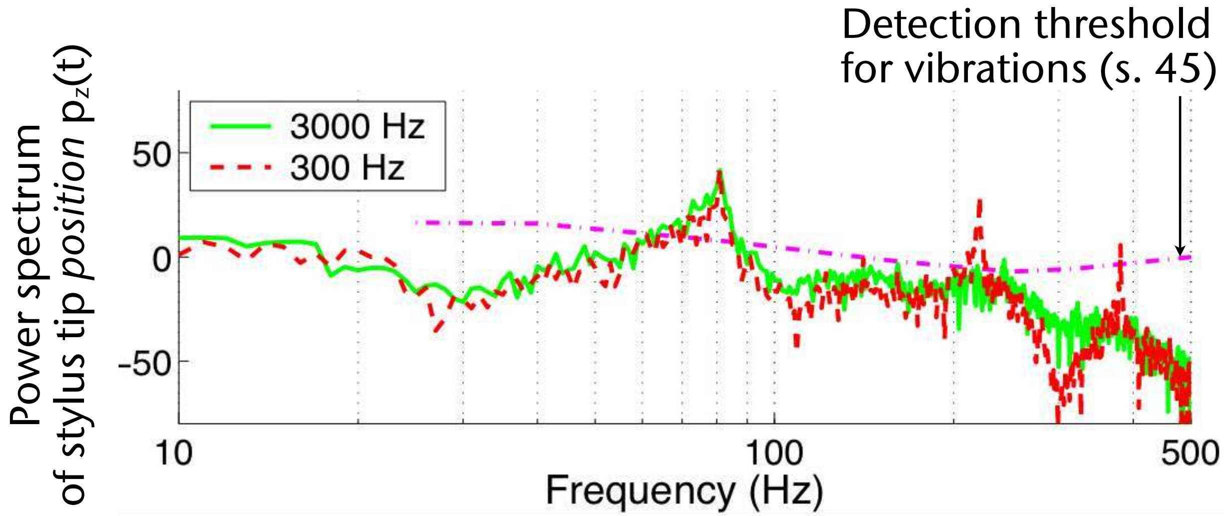
- Consider this experiment: a simple point probe device (e.g. Phantom) and a surface geometry in the shape of a microscopic sine-wave





- The force that should be rendered in case of constant tangential movement (= output on the actuators): $F(t) = k_s d(t)$
- Result with different rendering frequencies (user moves stylus across surface with a specific speed that yields ca. 80 Hz)
- Render forces with 1000 Hz!

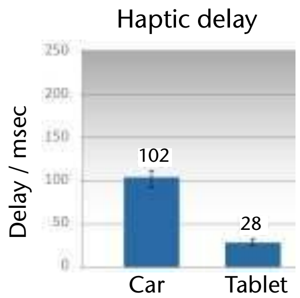


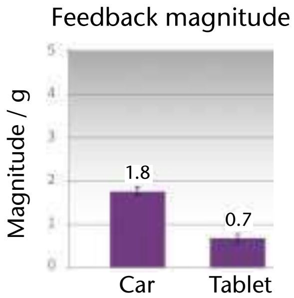


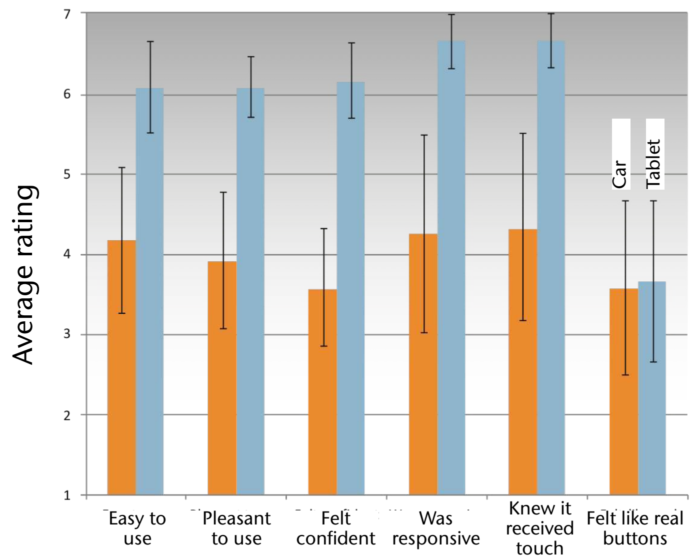
Latency in Haptic Feedback

- 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 infotainment system of a Cadillac model failed in 2012
- Replication study: infotainment and tablet, both with touch screen and haptic feedback, but different delays

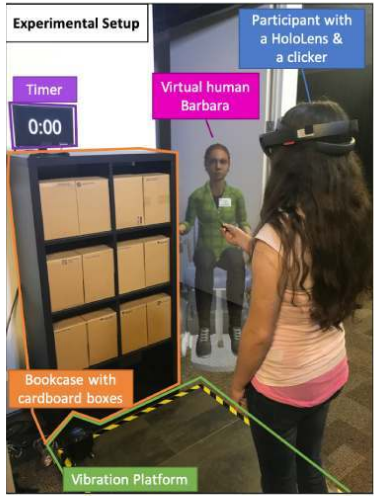
Infotainment system in car with haptic touch screen



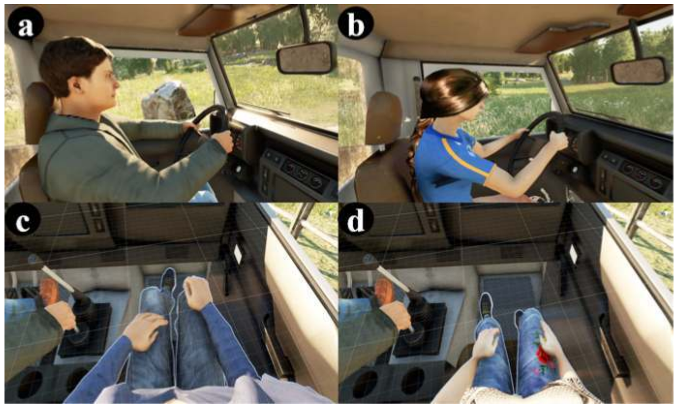




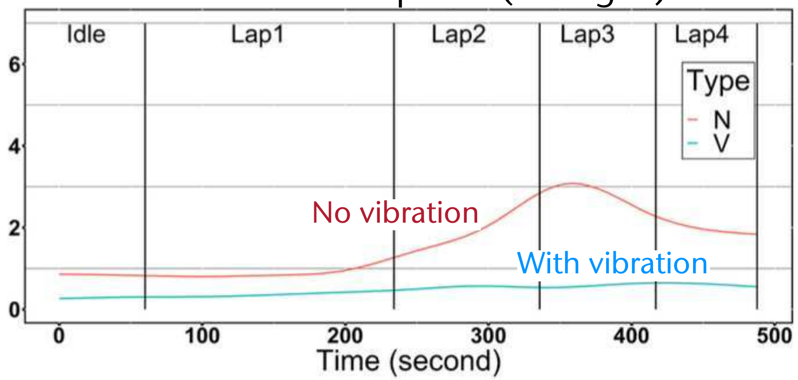
Effects of Haptic Feedback on Presence/Cybersickness



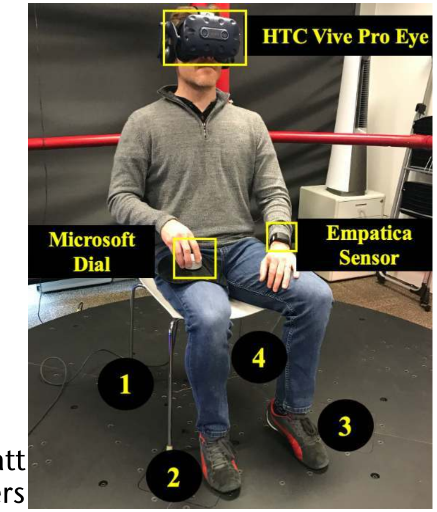
- Vibrotactile feedback increases spatial presence, social presence, and engagement
  - Haptic feedback = vibrotactile stimulation of feet through platform
- Floor vibrations also reduces cybersickness
  - Even if only a somewhat matching "rumble" is produced
  - Precisely matching motion cues reduce cybersickness, too (obviously)



Galvanic skin response (averaged)

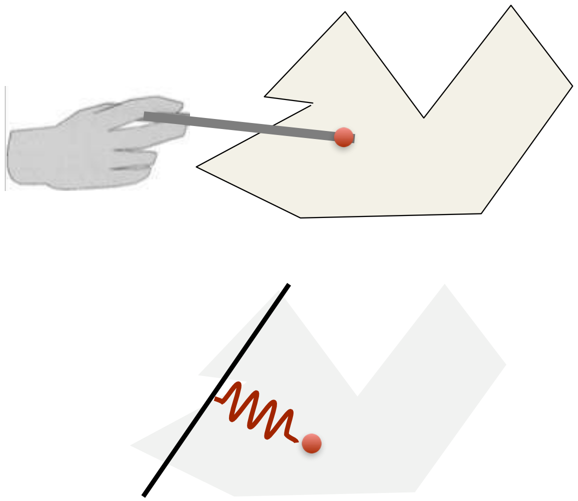


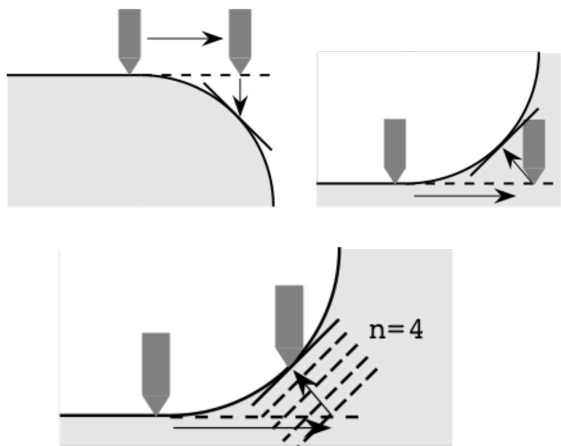
Vibrating platform using 4x 250 Watt audio transducers



Force-Feedback: Intermediate Representations (Proxy Geometries)

- 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



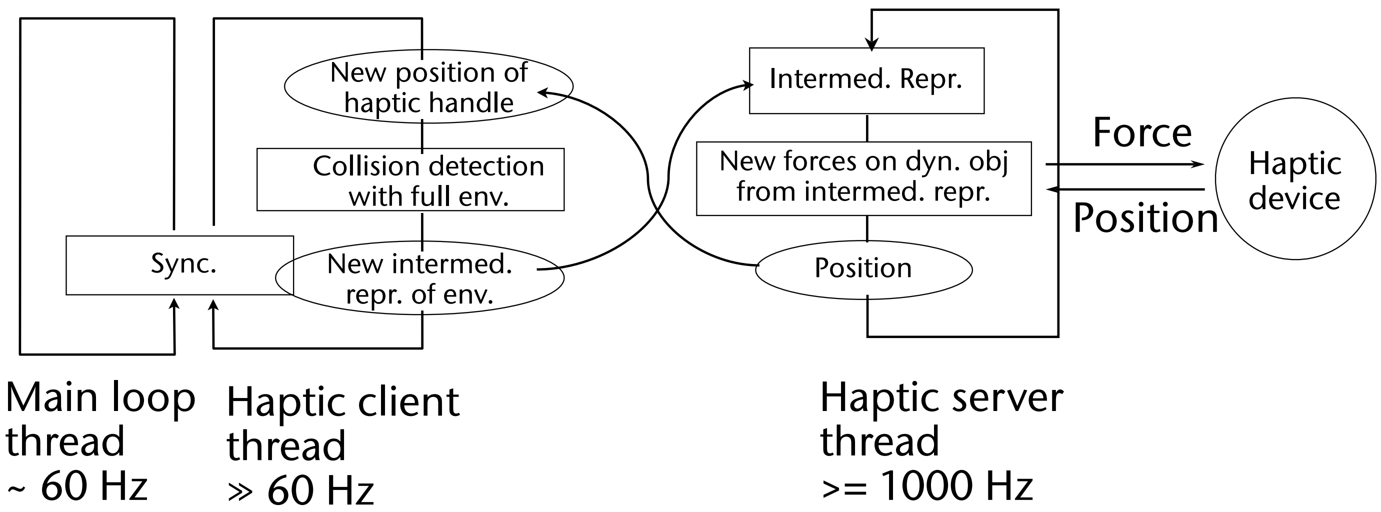


Two Principles for Haptic Rendering

- 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 (depending on the capabilities of the device):
  1. Impedance approach: haptic device returns current position, simulation checks collisions, calculates penalty forces, and sends these to device (to be exerted on human)
  2. Admittance approach: haptic device returns current forces (exerted by human), simulation moves virtual object (e.g. by Euler integration, then applying constraints), and sends new (desired) positions to device
- Penalty forces: the output force depends on the penetration depth of the dynamic object

Admittance-Based / Impedance-Based Haptics

- A haptic device works in one of two ways:
  - Sensors measuring forces (admittance-based) or positions (impedance-based)
  - Actuators move handle to a specific position (admittance-based) or produce a force/acceleration (impedance-based)
- Software architecture for impedance-based devices:

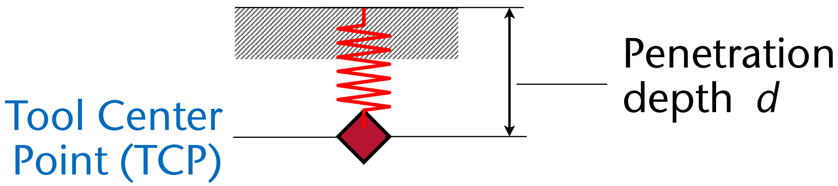


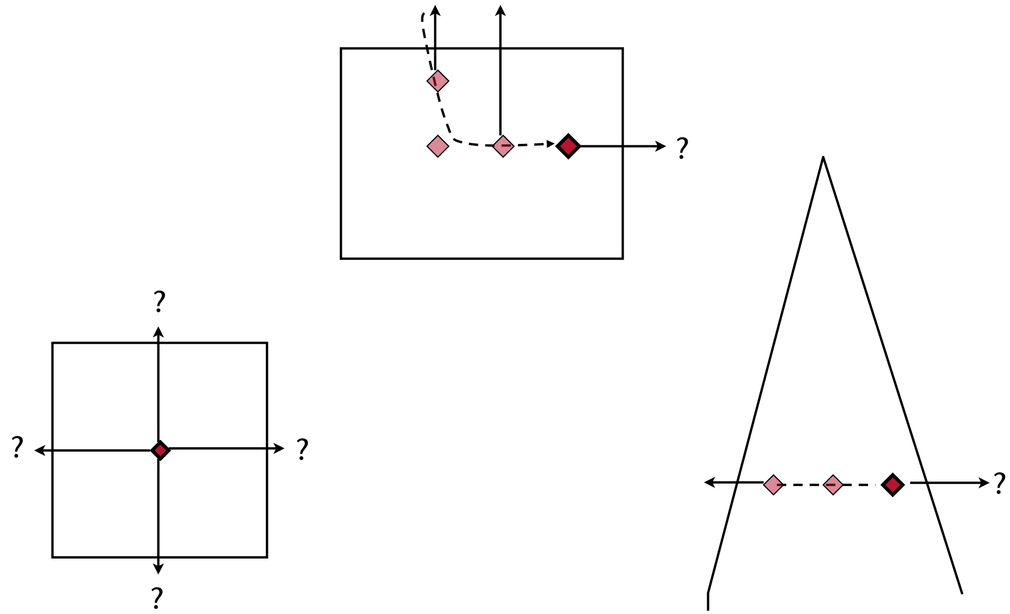
The "Surface Contact Point" Approach

- Often, penalty force is calculated using *Hooke's law*:

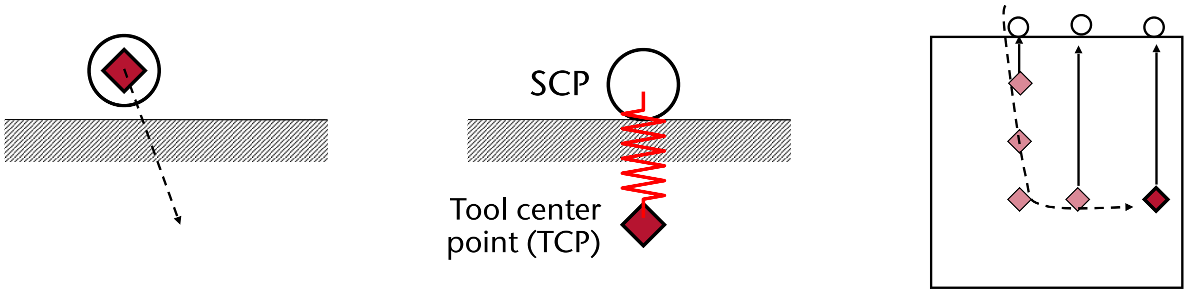
$$F = k \cdot d$$

- Question: what exactly is the penetration depth?
  - Naïve method: calculate closest point on surface and repulsion direction and magnitude towards that point
  - Problem: the history of the TCP is ignored



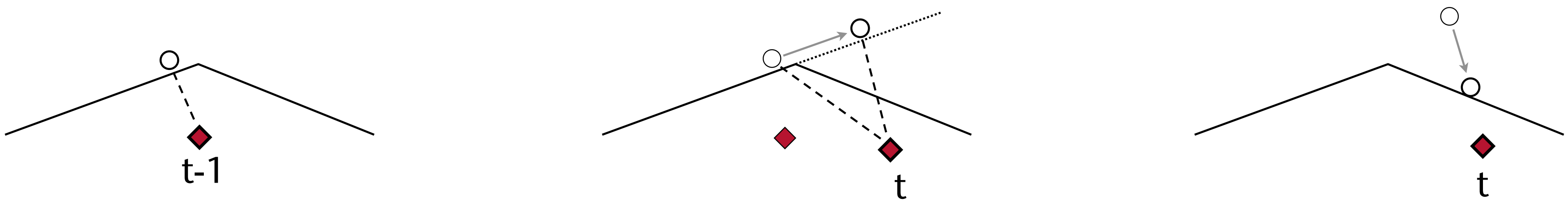


- Conclusion: with haptic rendering (at least) you need the history in some way
- Idea: represent the history as surface contact point (SCP)

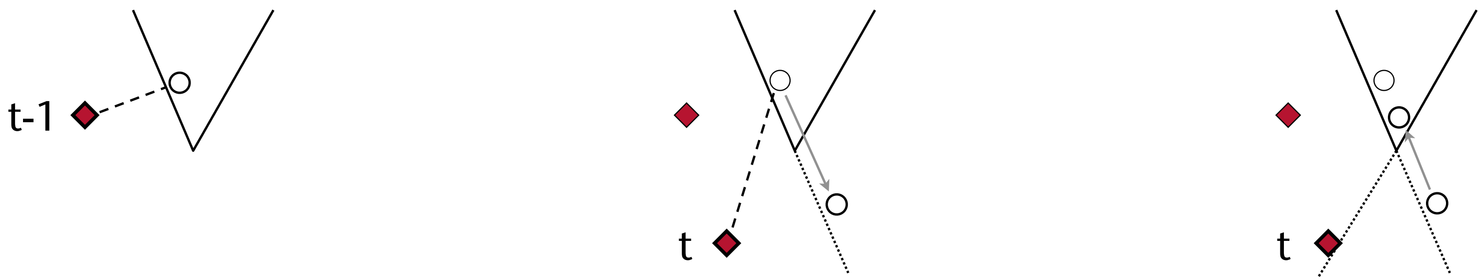


- Utilize temporal coherence: consider only polygons in the neighborhood of the current SCP
- In order to achieve numerical robustness: lift SCP slightly above the polygons

- Example for a convex edge:



- Example for a concave edge:



The Overall Algorithm

```
C = {pi1, pi2, pi3} // set of constraint polygons, at most 3, could be less
loop
  calc SCP'(t) = closest point to TCP(t) under constraint set C
  if any of the p in C is no longer a constraint:
    remove p from C
  if line SCP(t-1)SCP'(t) intersects any other polygon p in environment:
    add p to C
until constraints C do not change any more
```

- How to compute the SCP $\mathbf{x}$ under the constraints:
  - minimize $\|\mathbf{x} - \mathbf{x}_{TCP}\|^2$
    under the constraint $\mathbf{n}_i \mathbf{x} - d_i = 0 \; , \quad i = 1, 2, 3$
  - Approach: use method of Lagrange Multipliers (Lagrange'sche Multiplikatorenregel)

The Case for Constant Haptic Update Rates

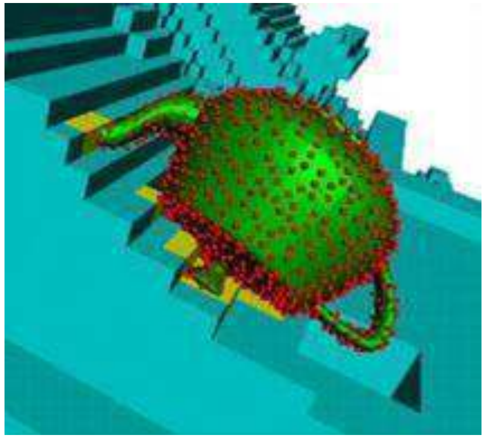
- Question: why is a **constant** update rate so important?
- Answer: because otherwise we get "jitter" (Rütteln, Ruckeln)
- Another reason will be given in the Voxmap-Pointshell method

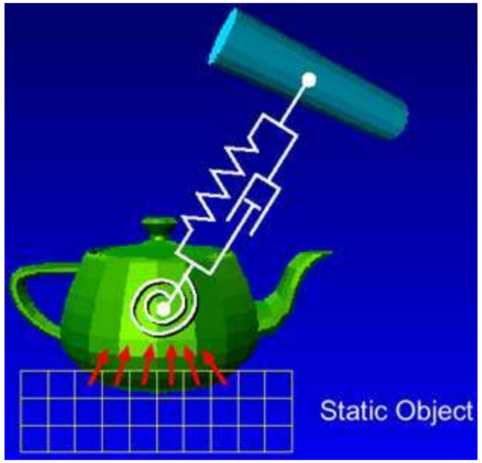
The Cause of Device Jitter

- 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)
- 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 longer than "usual":
  - The TCP penetrates the obstacle 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

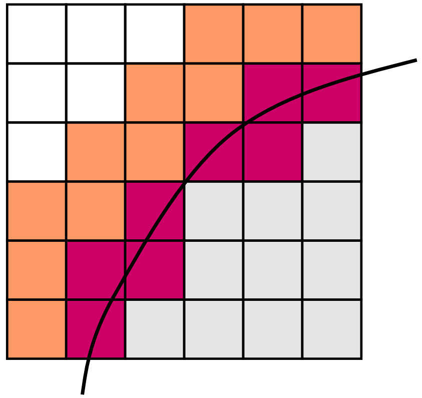
- Alternative representation of objects (no polygons):
  - Dynamic object: sample surface by lots of points = point shell
  - Rest of the scene: embed it in a 3D grid; voxmap = all voxels inside an obstacle
- Overview of the method:
  1. Compute forces for all penetrating points
  2. Compute total force on dynamic object
  3. Compute force on haptic handle

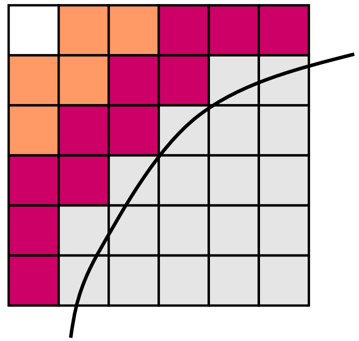




The VoxMap

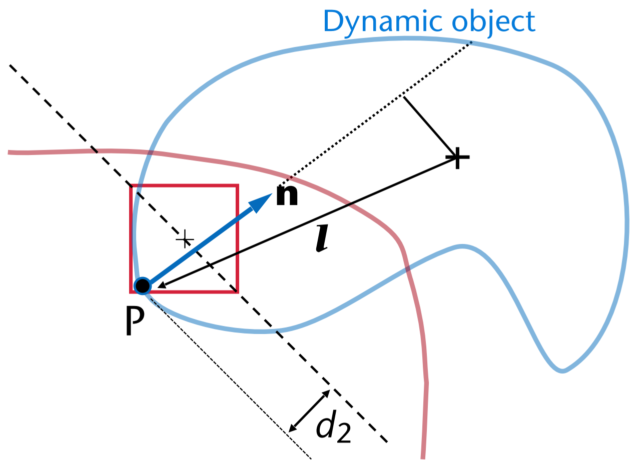
- Voxmap = 3D distance field
- Generation of a voxmap:
  - Scan-convert the surface (in 3D!) → gives all voxels that are intersected by the surface
  - Flood-fill from outside: do a breadth-first search starting from the border of the "universe" → all voxels outside any obstacles
  - All other voxels 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 (Chamfer method)
  - Usually: a "safety margin" is introduced





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 d = voxel depth (d_1) + distance, d_2, from P to the plane given by voxel center and normal **n**
  - Force: $\mathbf{F} = k_v \cdot d \cdot \mathbf{n}$
- Torque (Drehmoment): $\mathbf{M} = \mathbf{F} \times \boldsymbol{l}^0$



- Why use **n** instead of 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

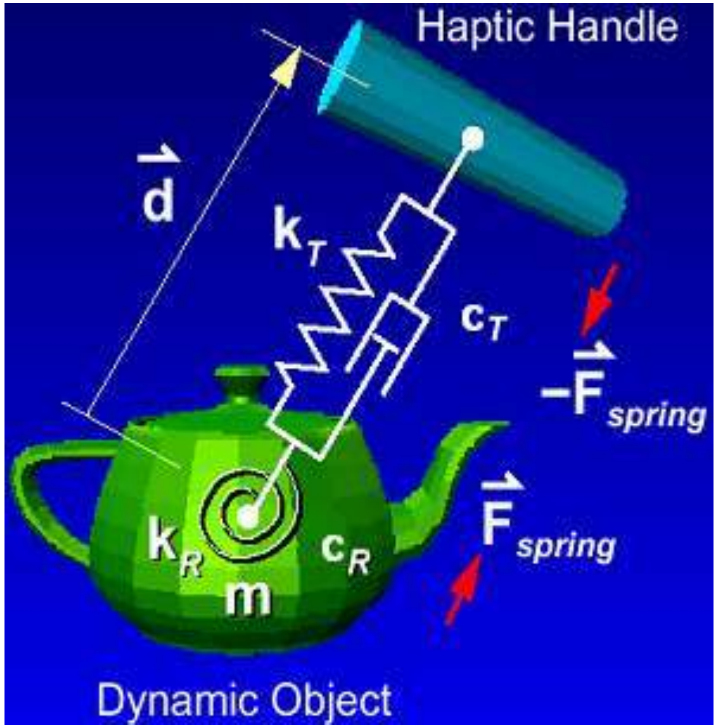
Virtual Coupling

- A virtual coupling = 6 DoF spring-damper
- Forces between the dynamic object and the 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 **v** that is in the direction of **d**
  - Set viscosity to 0, if **v** points away from the handle (for hard contacts)

Simulation of the Motion of the Dynamic Object

- Total force acting on the dynamic object:

$$F = F_{\text{handle}} + \frac{1}{N} \sum_{i=1\ldots N} F_i \quad , N = \#\text{ pointshell pts penetrating static objects}$$

(Analog for the torques)

- Integrate the following equations of motion:

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

where

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

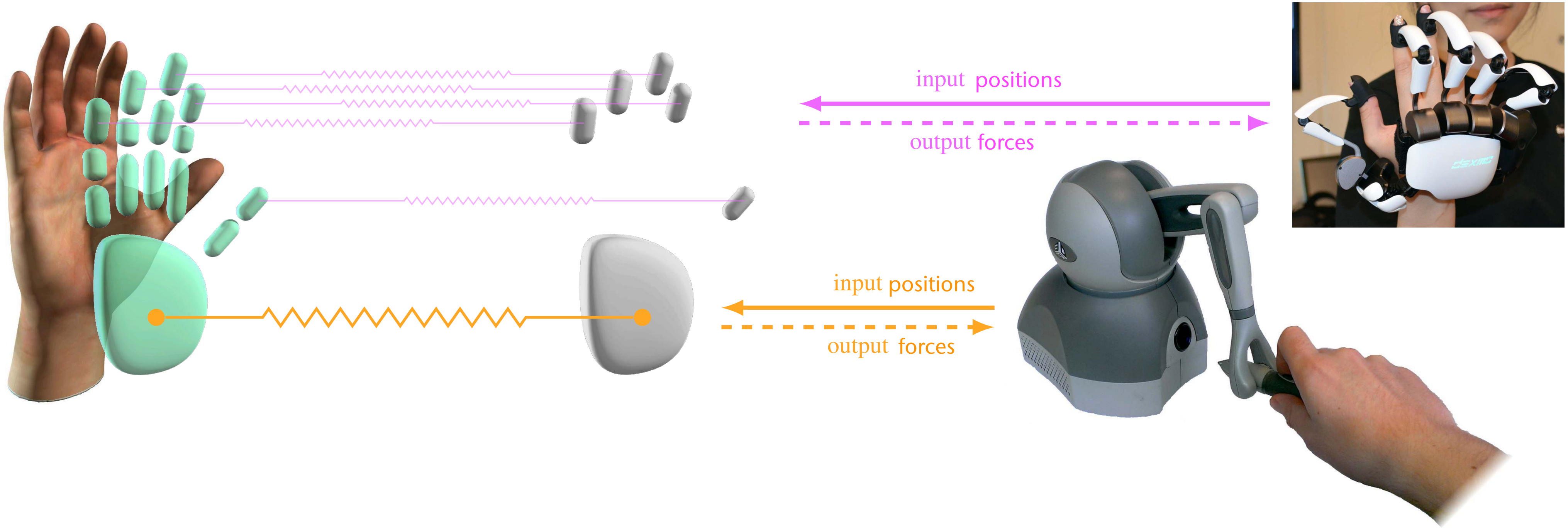
- Prerequisite: Δt is known in advance (e.g., because it is constant)

Overall Algorithm

1. Check collision between dynamic object and static universe
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. Compute new position of dynamic object (e.g., Euler integration)
6. Compute forces on haptic handle mediated by virtual coupling

- Effectively, virtual coupling = low-pass filter

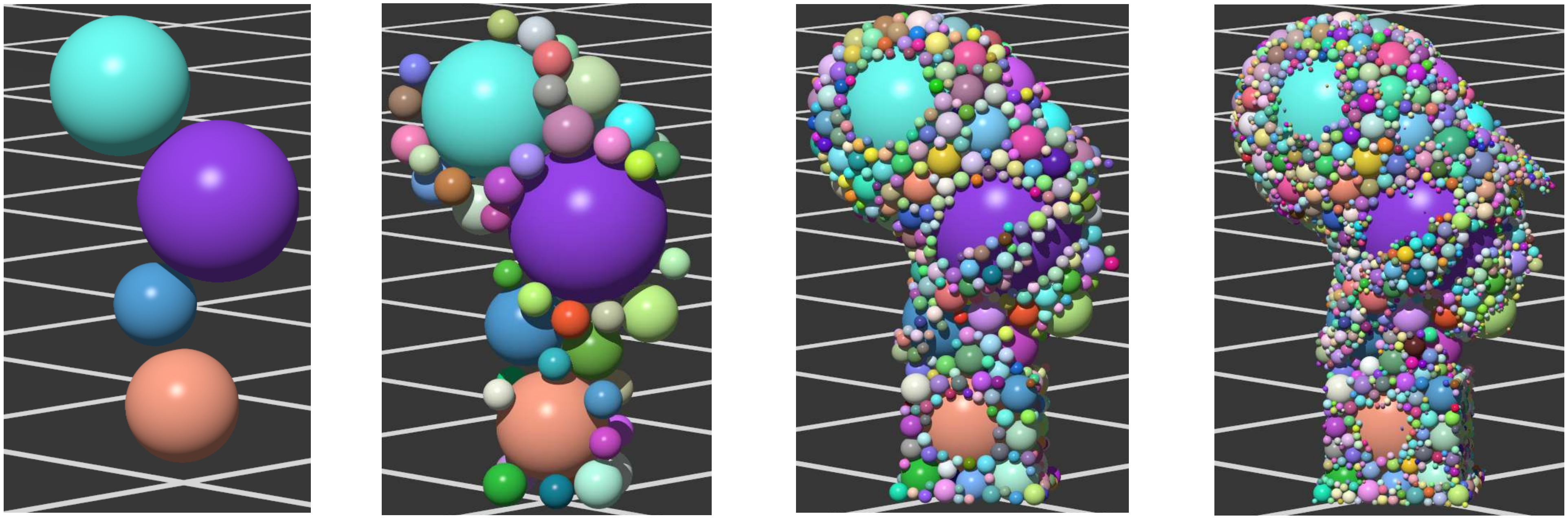
Illustration of the Concept of Virtual Coupling



[Achibet et al.]

Outlook: Use Sphere Packings

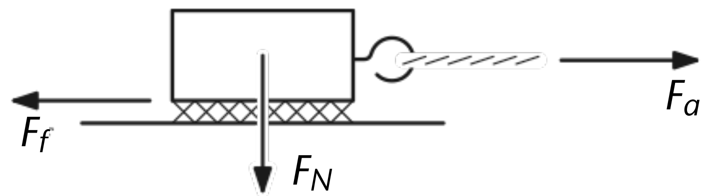
- See Chapter on *Collision Detection*

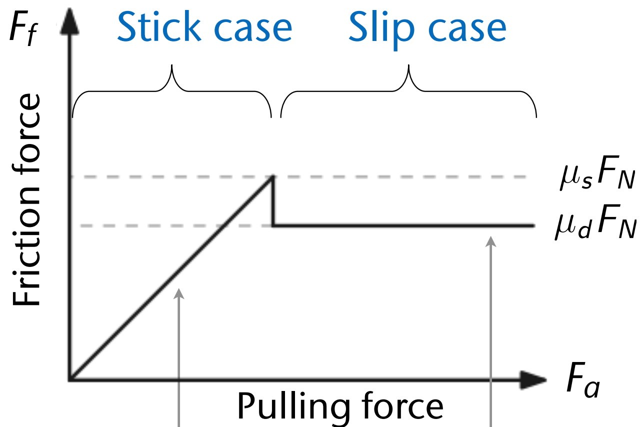


Friction

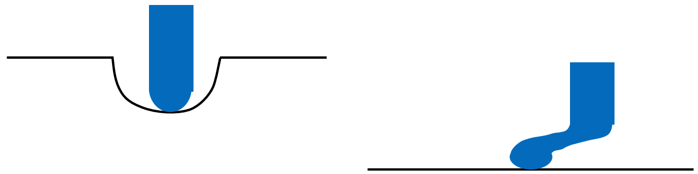
- Consider this situation:
  F_a = pulling force,
  F_N = force normal to surface,
  F_f = friction force
- Coulomb's Law of Friction:
  So long as
  $$|F_a| \leq \mu_s |F_N|$$
  the object will not move, i.e.,
  $$F_f = -F_a$$
  (stick case, Haftreibung).
  μ_s = static friction coeff.

  μ_d = sliding friction coeff.

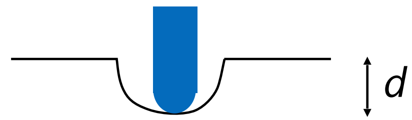




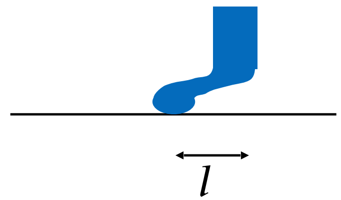
Friction in One Contact Point for Force Feedback



- 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: $F_N = k_N \cdot d$



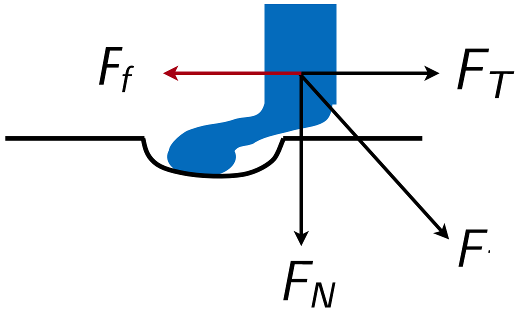
  - Force tangential to surface: $F_T = k_T \cdot l$



The Friction Cone

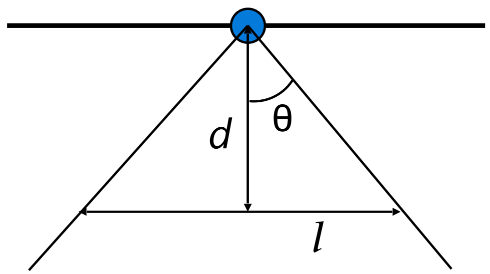
- The Coulomb friction model says:

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



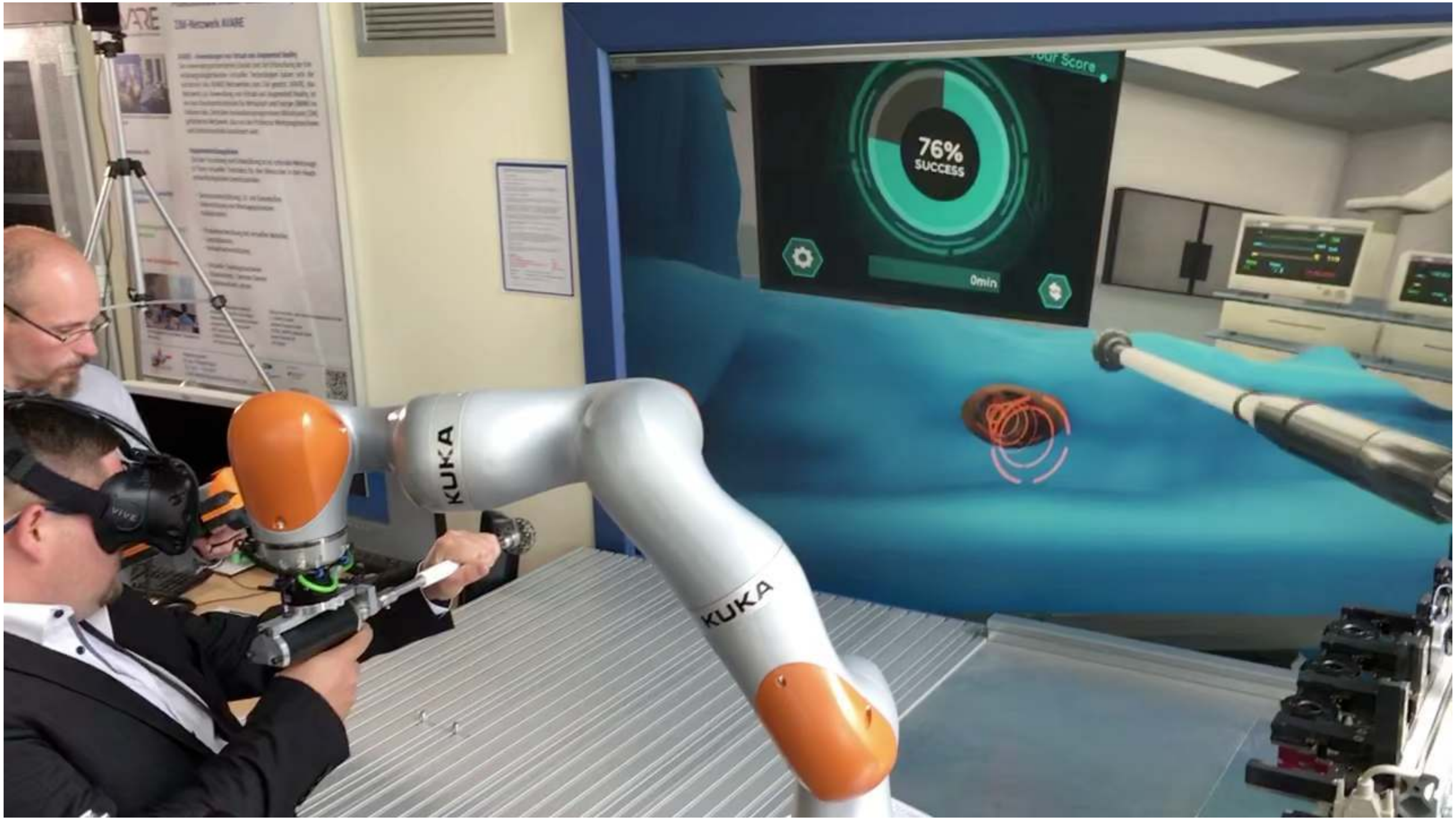
- The "cone of friction":
  describes the boundary between static friction and sliding friction (Gleitreibung; aka. kinetic friction)

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



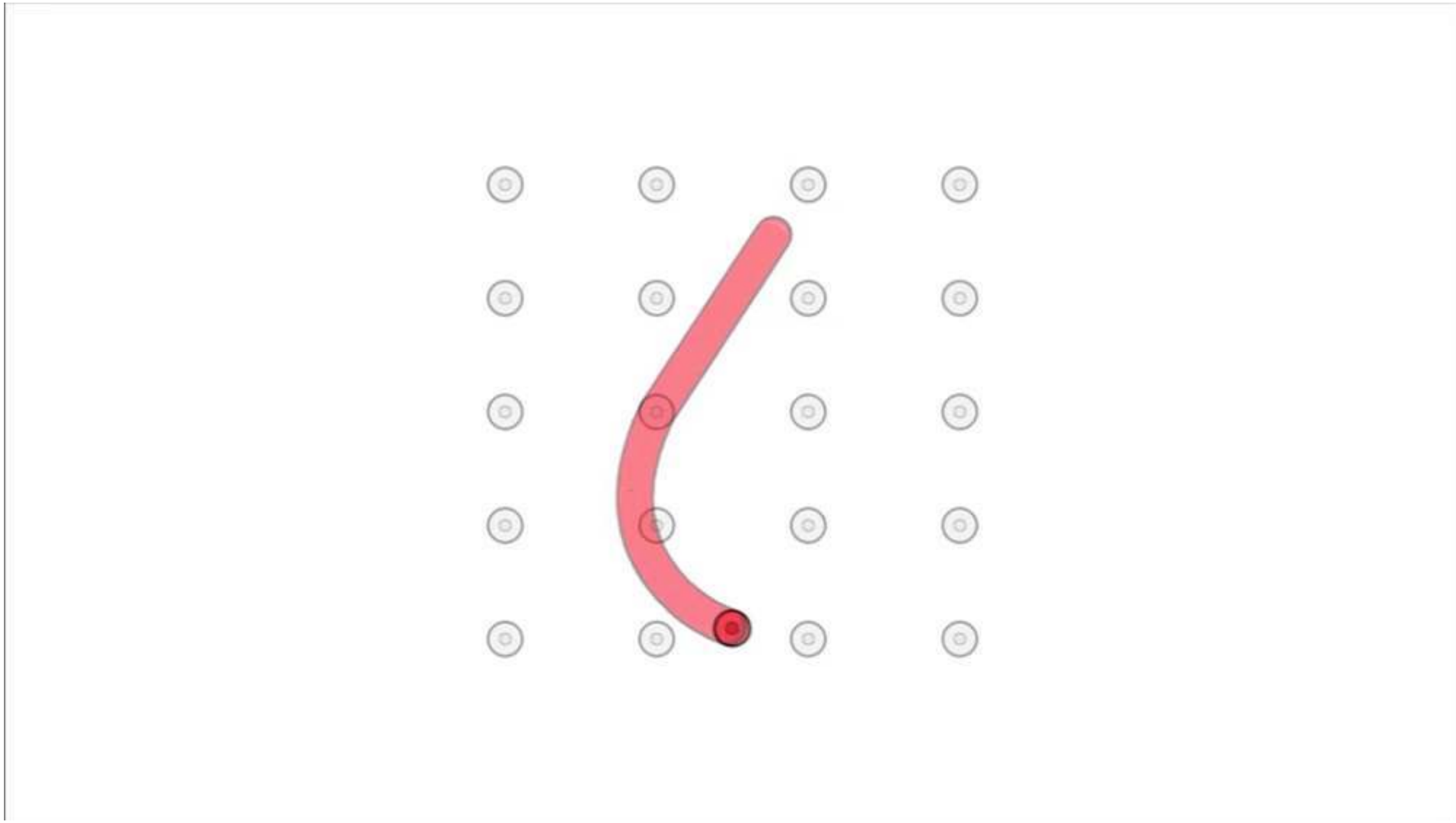
$$\theta = \tan^{-1}\left(\mu \frac{k_N}{k_T}\right)$$

Application: Virtual Hip Surgery Simulator



University of Bremen (CGVR), TU Chemnitz, FAKT Software, CAT Solutions

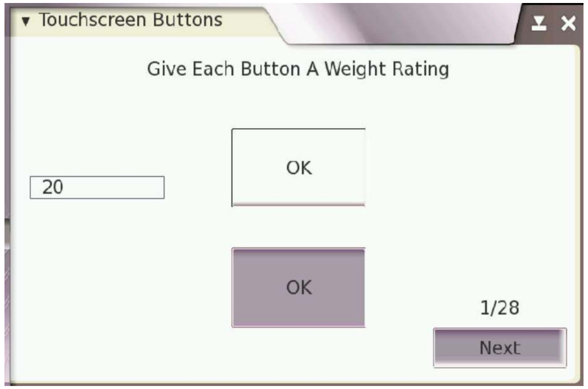
Haptic Illusions

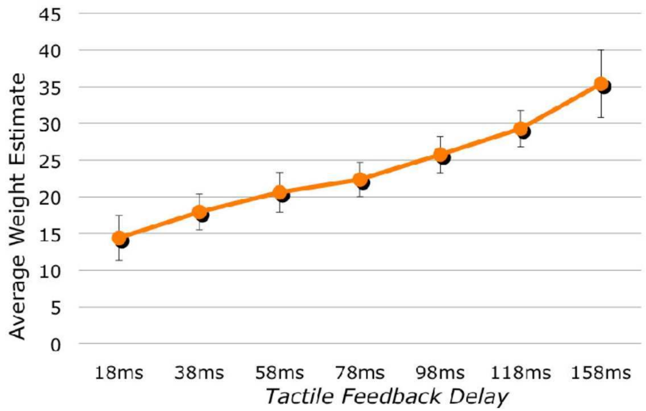


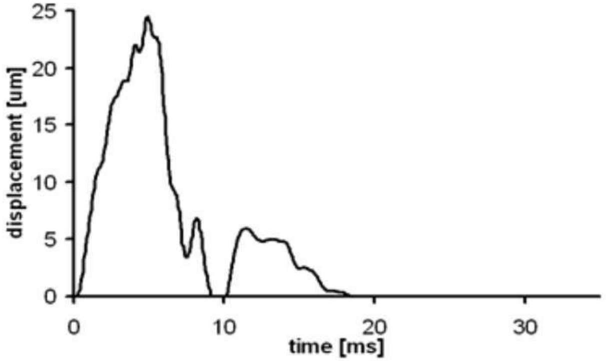
Surround Haptics Display / Haptic Chair by Disney Research, Pittsburgh

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:

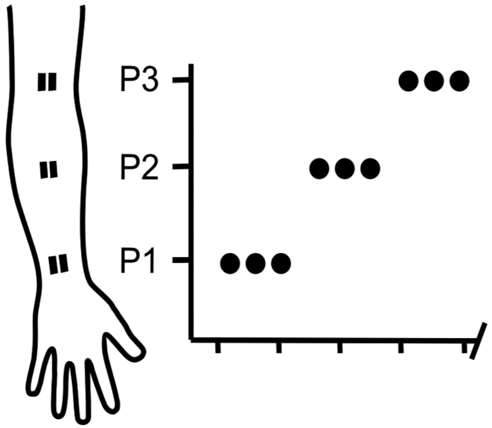


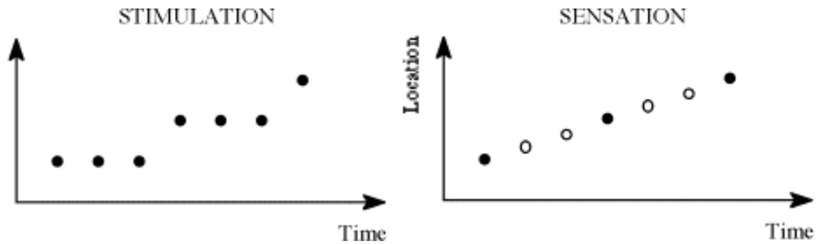


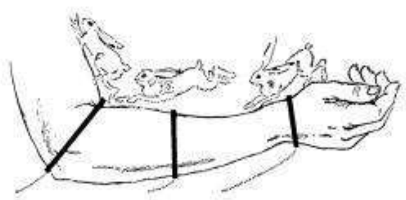


Cutaneous Rabbit Illusion

- 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








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