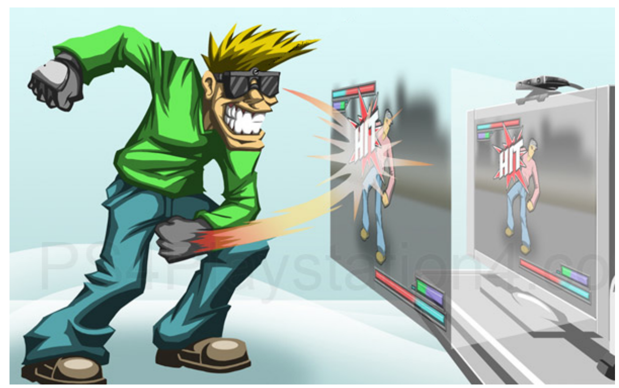
Virtual Reality & Physically-Based Simulation
Haptics



G. Zachmann

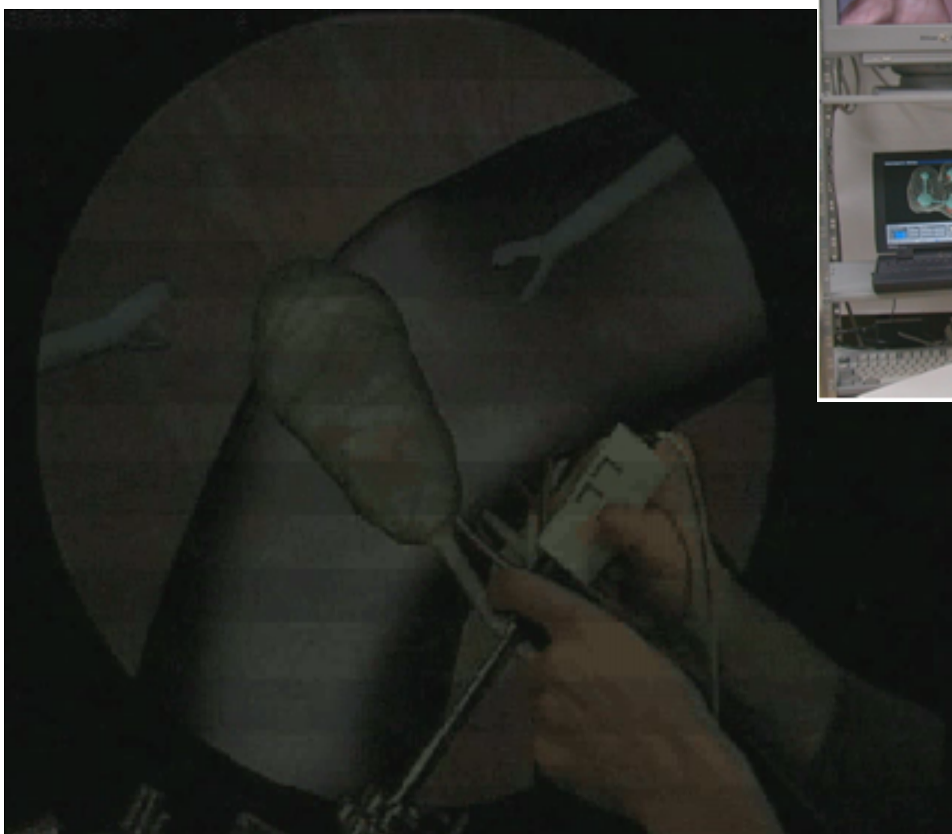
University of Bremen, Germany

cgvr.cs.uni-bremen.de

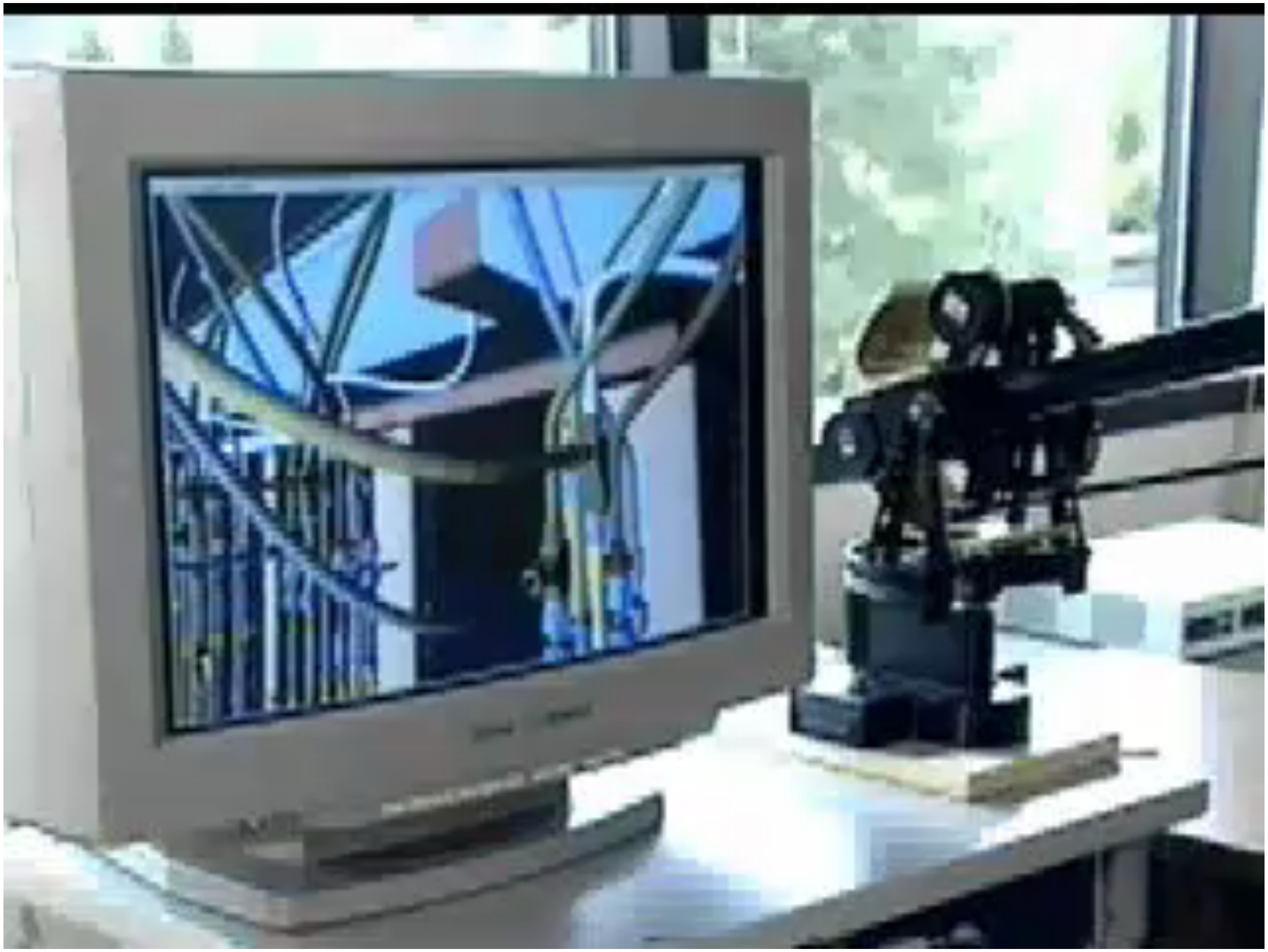
In the Following, We'll Consider a Class of Non-Visual Displays

- Haptics = sense of touch and force (griech. *haptesthai* = berühren)
- Special case: force feedback
- What is to be rendered:
  - Forces on the user's hand / arm (= haptic "image" of objects)
  - Haptic surface texture (roughness, grain, friction, elasticity, ...)
  - Shape of objects by way of touching/feeling (think pin board)
- Applications:
  - Training of minimally invasive surgery (surgeons mostly work by feeling, instead of seeing)
  - Games? (probably would increase presence)
  - Industry: virtual assembly simulation (e.g., to improve worker's performance / comfortwhen assembling parts), styling (look & feel of a new surface)
  - Ideally, one would like to answer question like "how does the new design of the razor feel when grasped?"

Example Application: Minimally Invasive Surgery



Another Application: Assembly Simulation



Some Info on Human Haptics

- Tactile information:
  - Obtained by sensors in the human skin
  - Can sense details of a shape, texture, friction, ...
  - Human factors of the tip of a finger:
    - Precision = 0.15 mm with position of a point
    - Spatial acuity = 1 mm (= discrimination of 2 points)
    - Detection thresholds: 0.06 micrometers for ridges; 2 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 stiffnesses, ...
  - Human factors:
    - Acuity: 2 and 1 degrees for finger and shoulder, resp.
    - 0.5-2.5 mm (finger)

A Collection of Force Feedback Devices

CyberForce

(movie)

Phantom

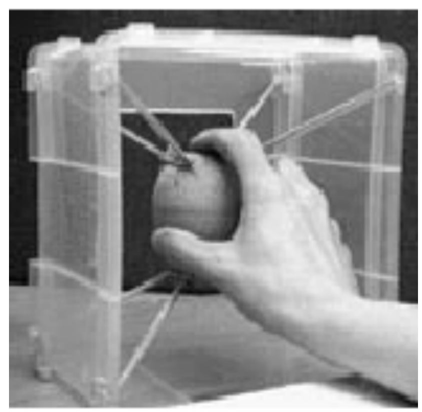
Sarcos (movie)

CyberForce

"Maglev" (magnetic levitation device)

(movies)

Tsukuba



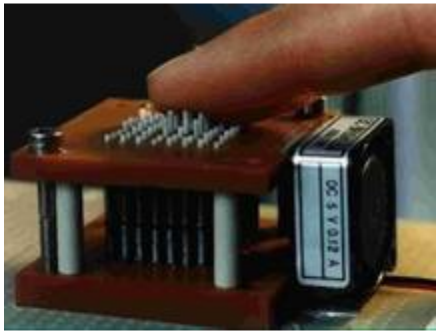
Spidar

Devices with Force Feedback via Wires (Spidar Variants)

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

INCA 6D von Haption

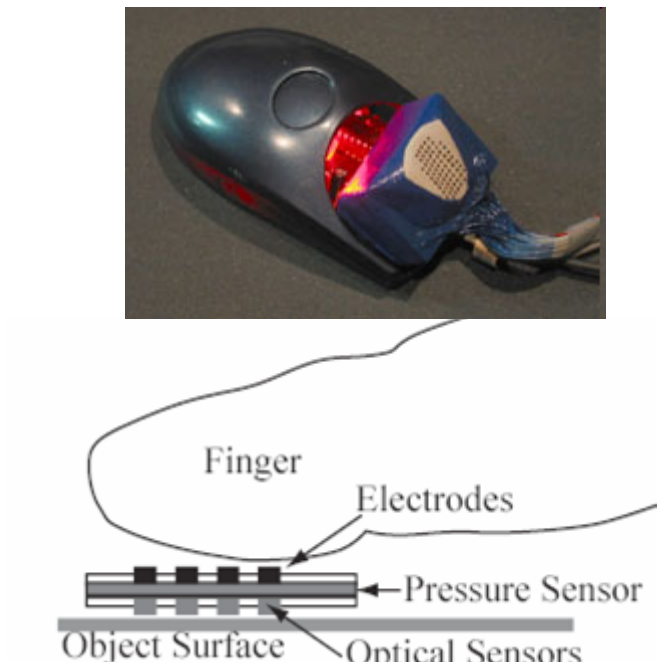
Tactile Displays



CyberTouch

AURA INTERACTOR

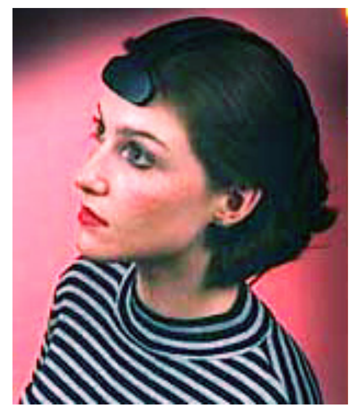
Feelex



SmartTouch (movie)

Motion Platforms (not really Force-Feedback)

Flogiston

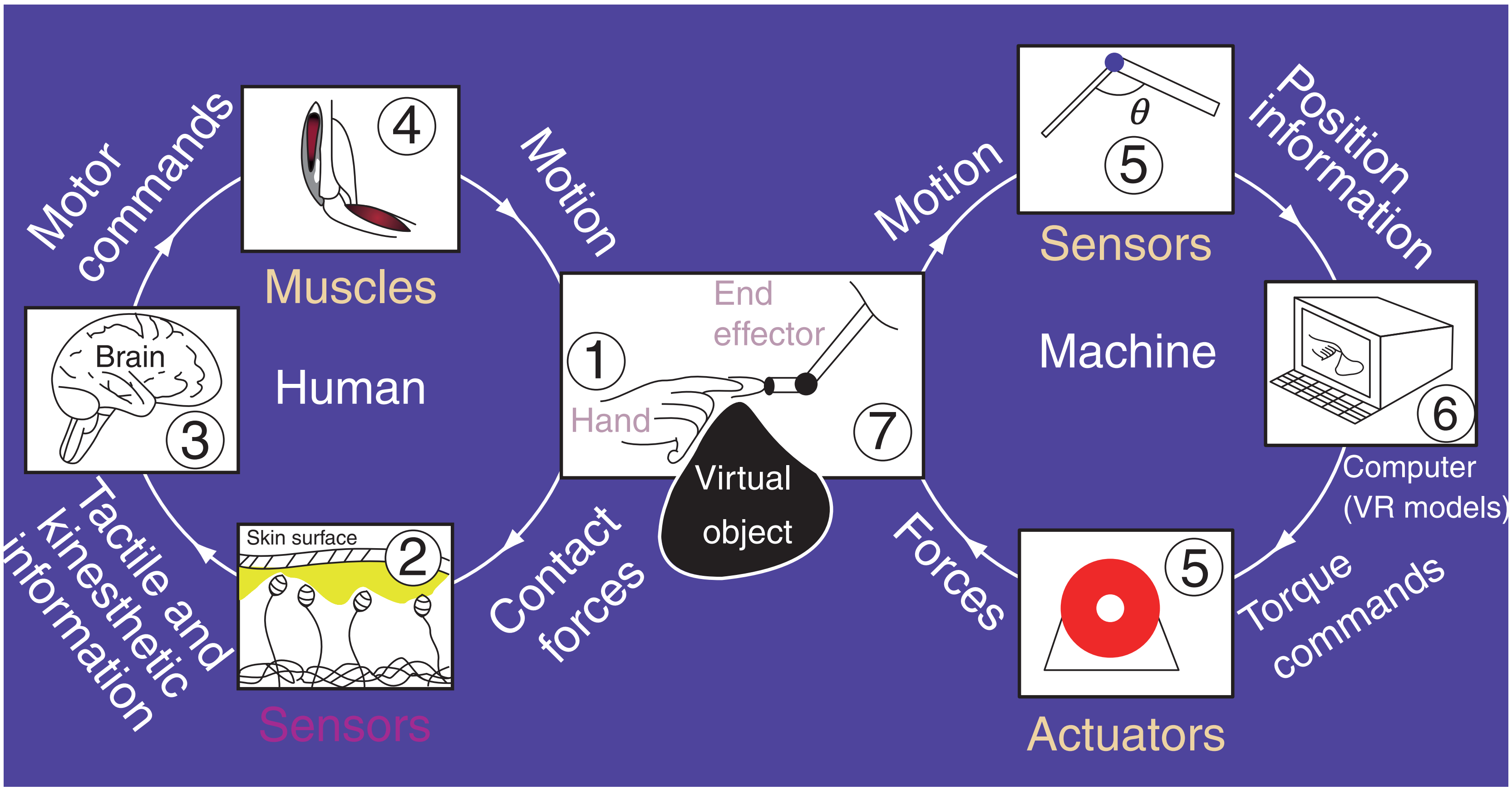


The Number of Degrees-of-Freedom with Force-Feedback Devices

- Number of DoF's a device can display: typ. ≤ 6 with *Force-Feedback*
- Number of inner DoF's = sum of DoF's of all joints

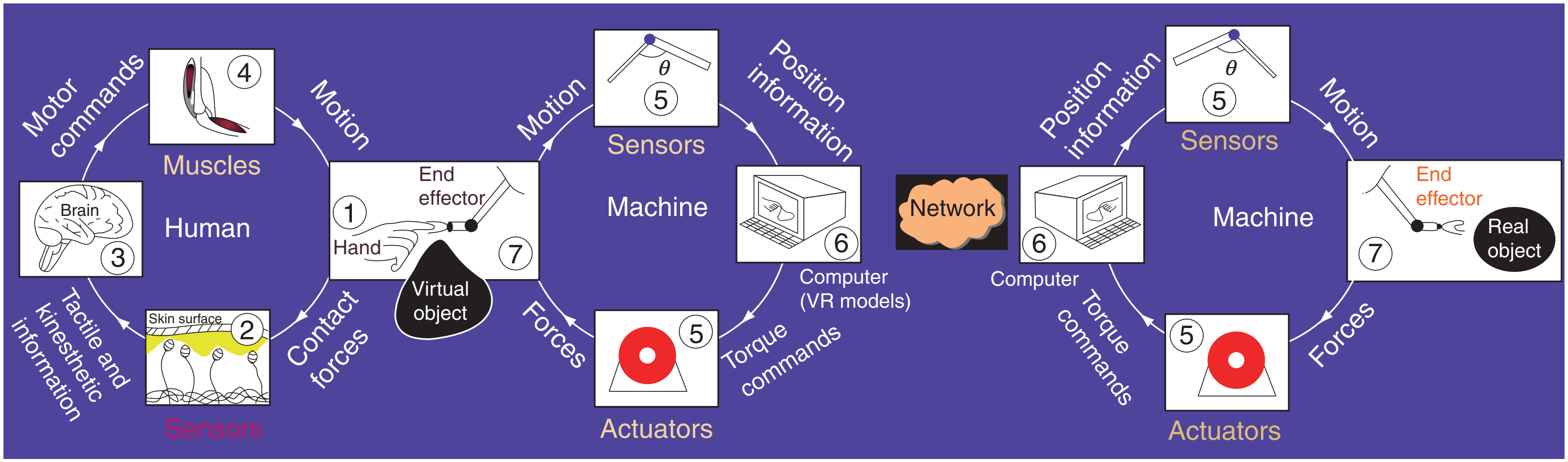
Cybernet

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

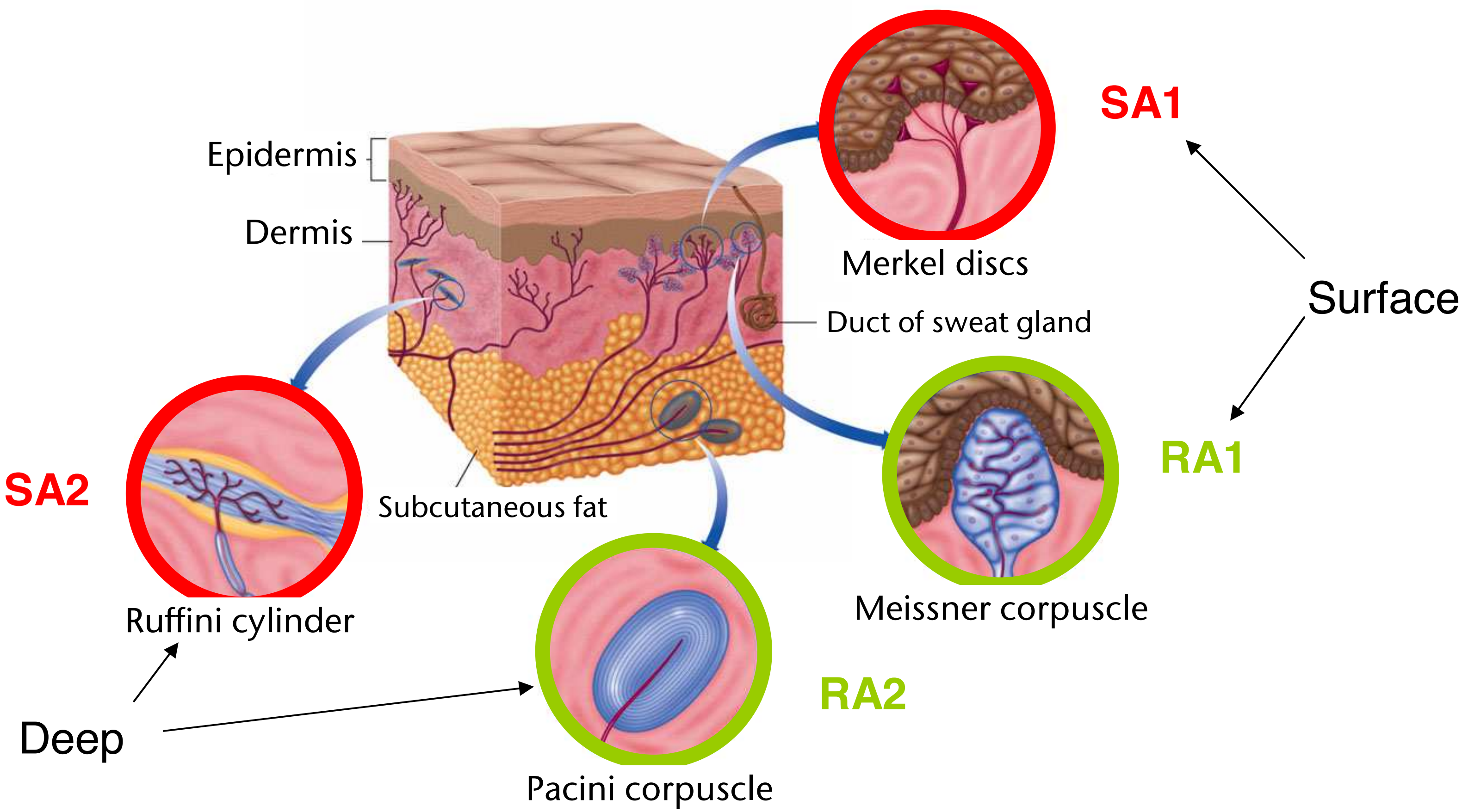
… and that of Telepresence



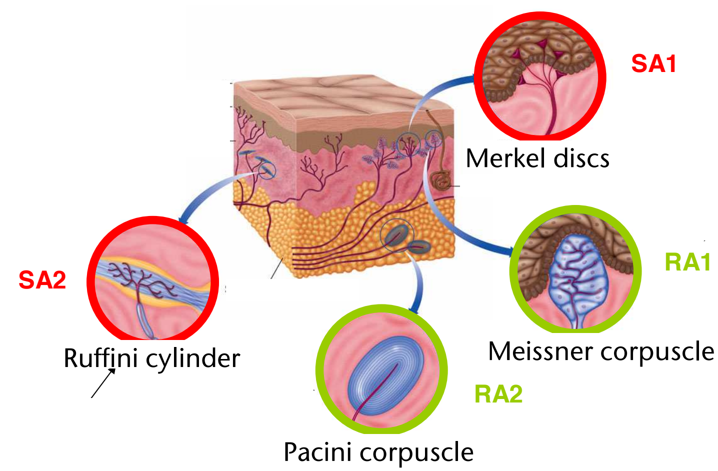
M A Srinivasan & R Zimmer: *Machine Haptics*.
New Encyclopedia of Neuroscience, Ed: Larry R. Squire, Vol. 5, pp. 589-595, Oxford: Academic Press, 2009

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

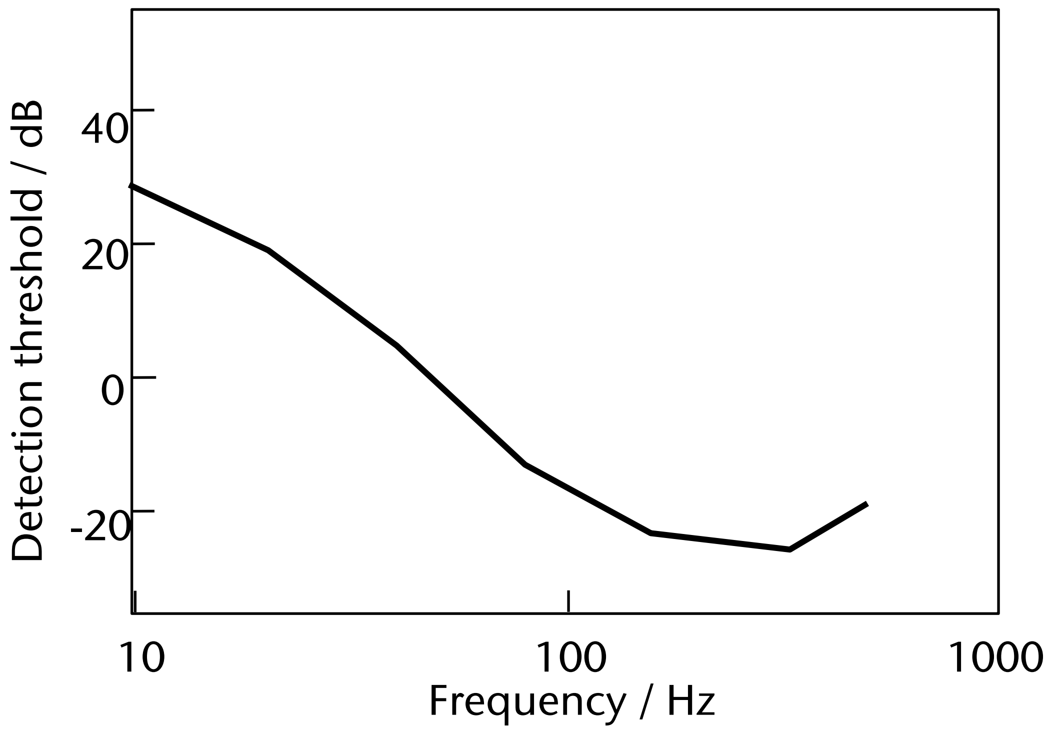
Response to vibration frequency	slow	fast	Location in Skin
low	Merkel	Meissner	surface
high	Ruffini	Pacini	deep

Human Factors with Respect to Haptics

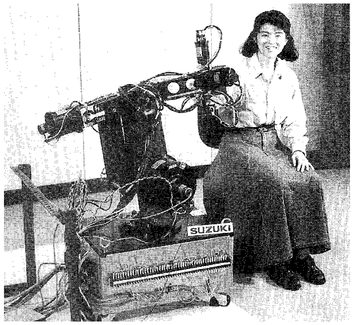
- Time until a reflex occurs:
  - Reflex through muscle: 30 millisec
  - Reflex through spinal cord: 70 millisec
  - Voluntary action: ?
- The frequency of human forces generation:
  - 1-2 Hz for irregular force signals
  - 2-5 Hz when generating periodic force signals
  - 5 Hz for trained trajectories
  - 10 Hz with involuntary reflexes
- Forces:
  - Max. 50-100 N
  - Typ. 5-15 N (manipulation and exploration)

- Sensation of stiffness/rigidity: in order to render hard surfaces, you need >1 N/mm (better yet 14 N/mm)

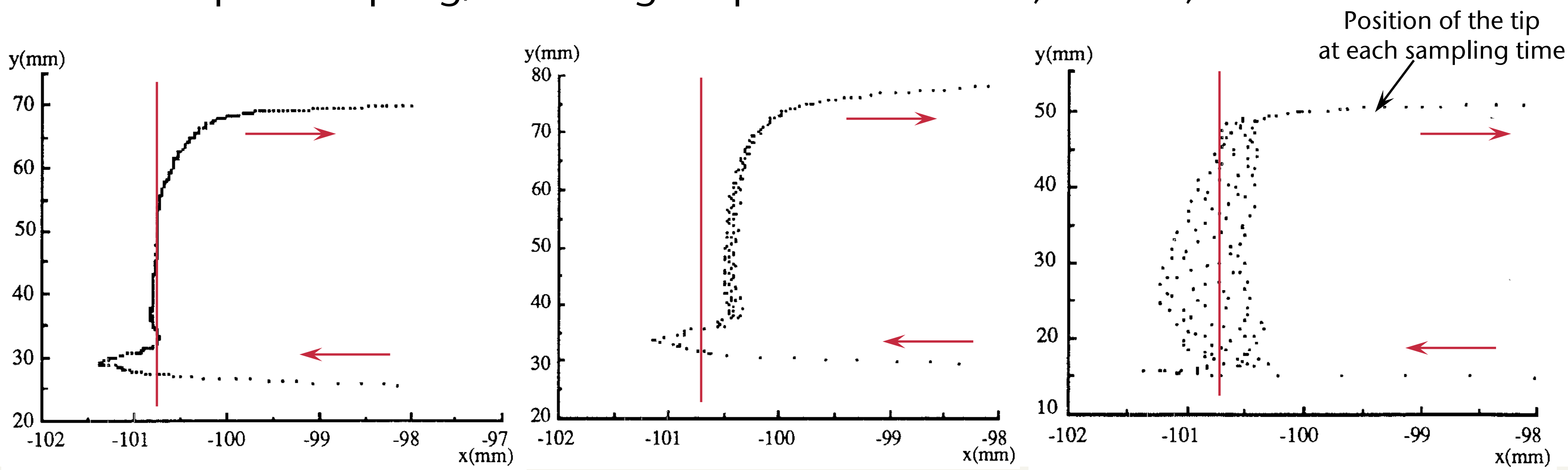
- Detection threshold for vibrations:



Rule of Thumb: 1000 Hz Update Needed for Haptic Rendering

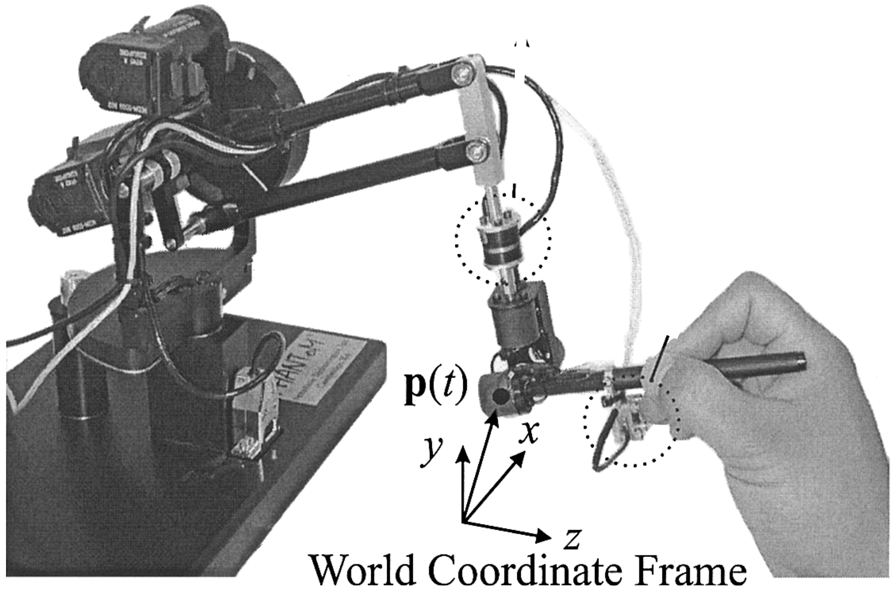


- An Experiment as "proof":
  - That haptic device with a pen-like handle and 3 DOFs
  - The haptic obstacle = a flat polygon
  - Task: move the tip of the pen along the surface of the polygon (*tracing task*)
  - Impedance-based rendering (later)
  - Stiffness = 10000 N/m, coefficient of friction =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



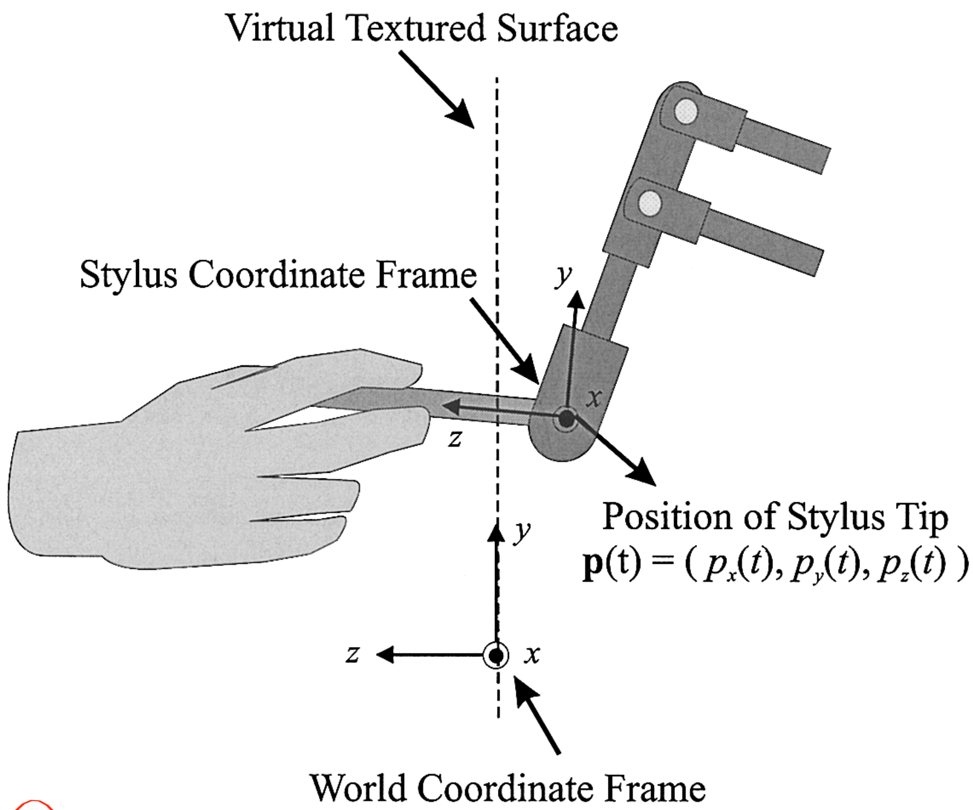
Haptic Textures

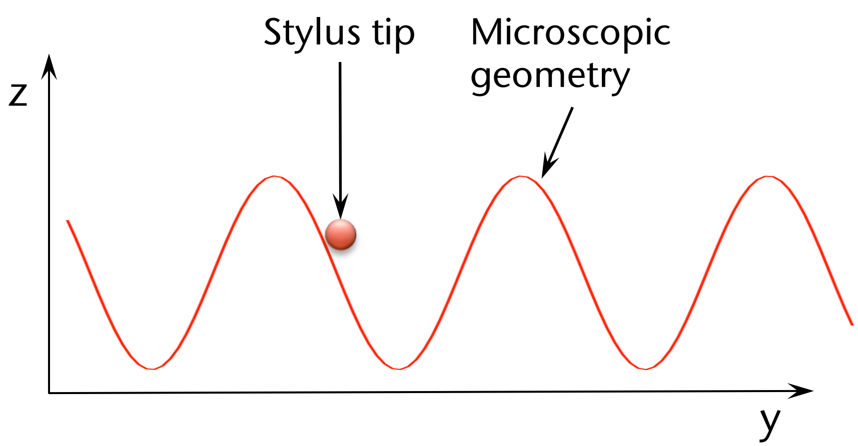
- Texture = fine structure of the surface of objects (= micro-geometry); independent of the shape of an object (= macro-geometry)
- Haptic textures can be sensed in two ways by touching:
  - Spatially
  - Temporally (when moving your finger across the surface)
- Sensing haptic textures via force-feedback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



A Frequent Problem: "Buzzing"

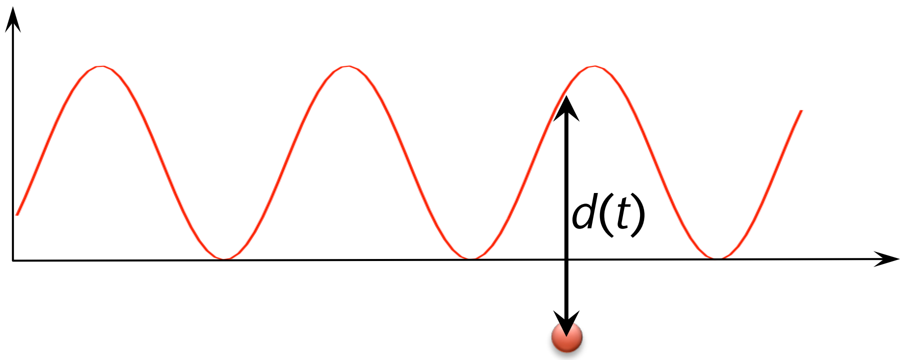
- Consider this experiment: a simple Phantom-like device and a sinus-wave haptic texture



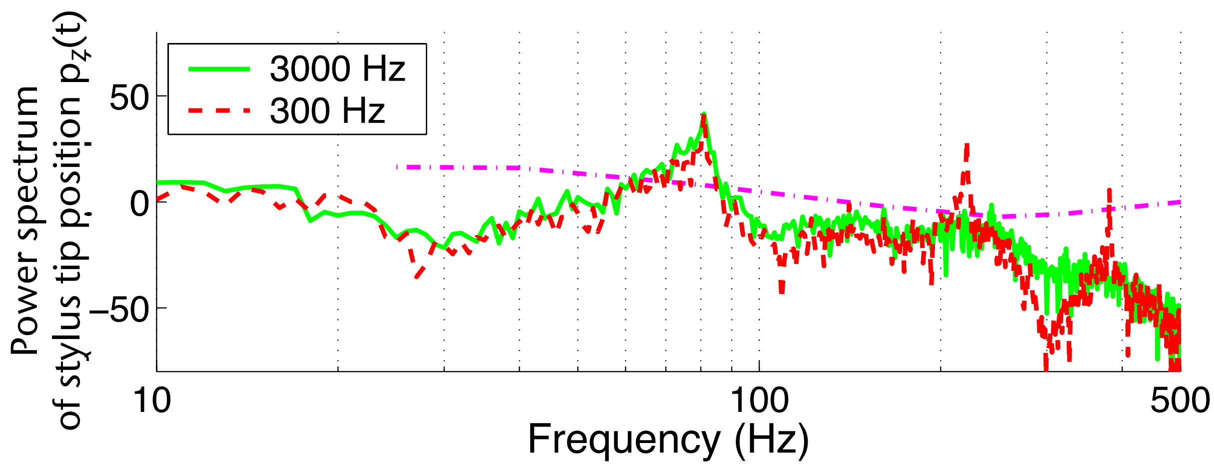


- The force that is rendered (= output on the actuators):

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

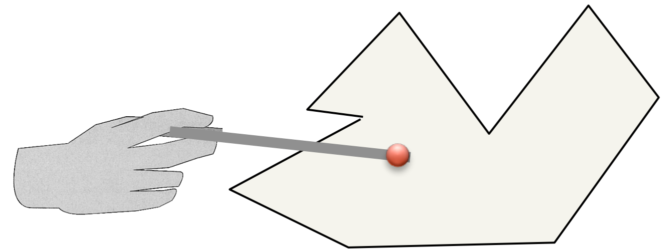


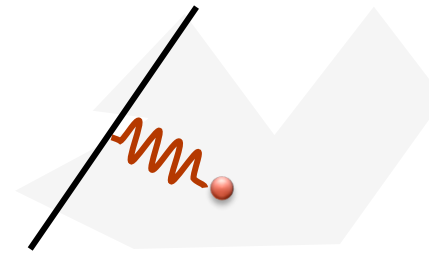
- Result with different rendering frequencies:

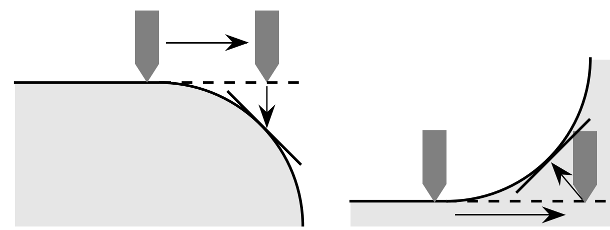


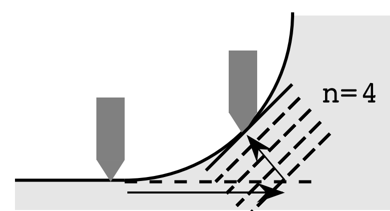
Intermediate Representations

- 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





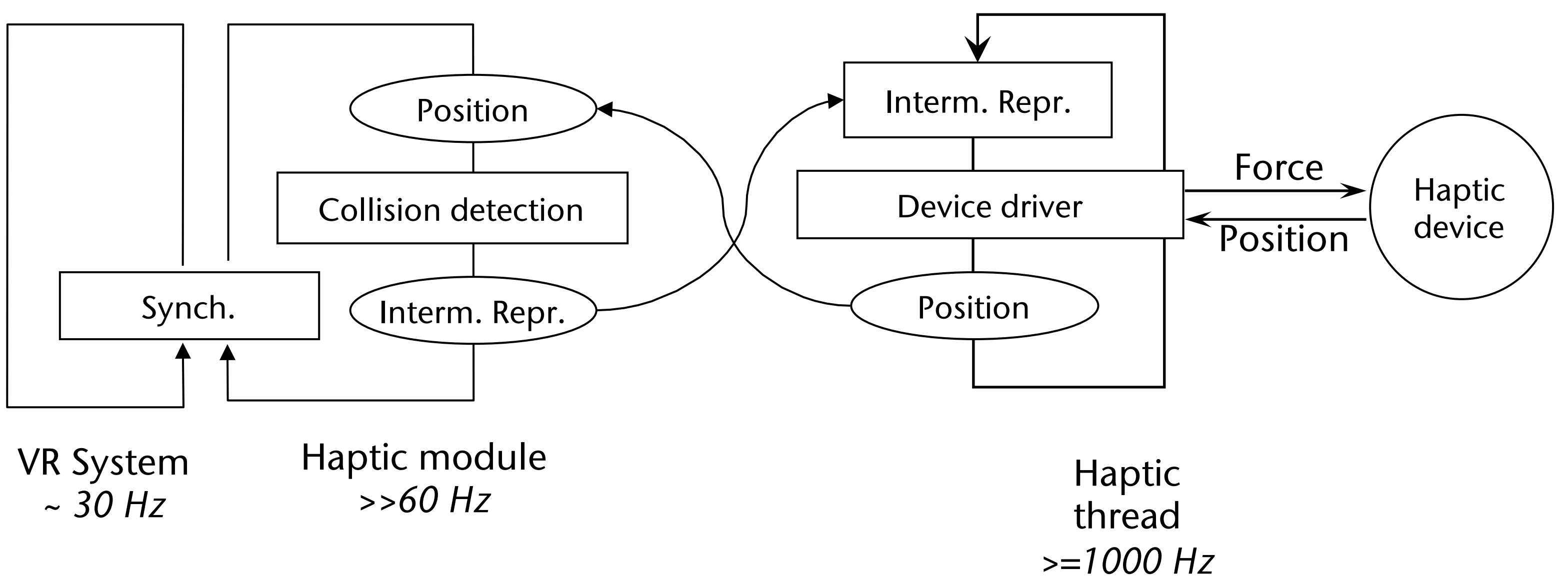




Software Archtecture

- 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:



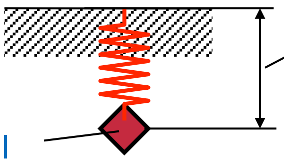
The Principle of Haptic Rendering

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

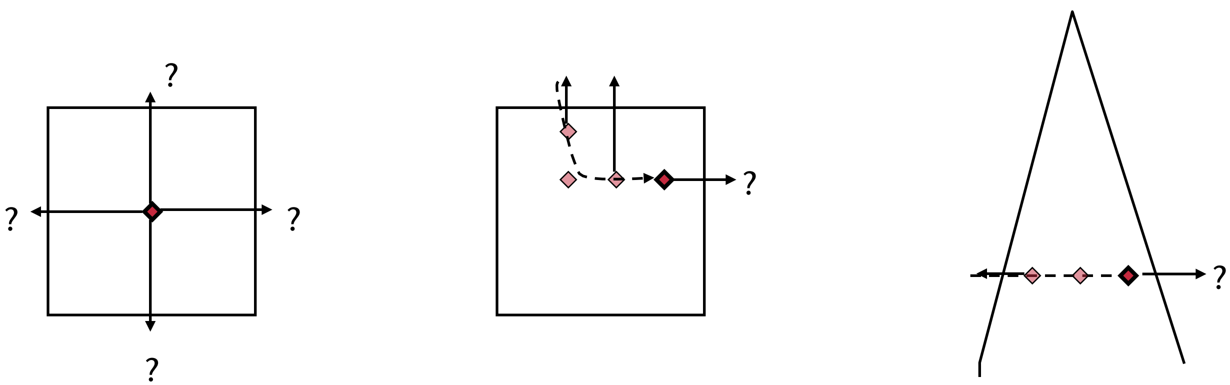
The "Surface Contact Point" Approach

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

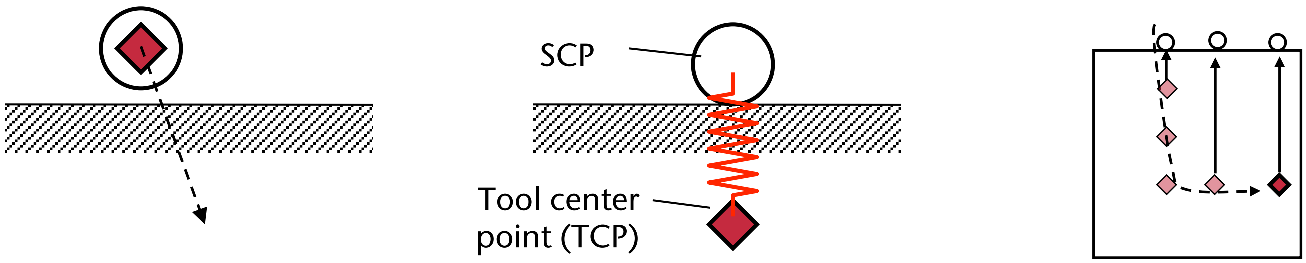
$$F = k \cdot d$$



- Question: what exactly is the penetration depth?
  - Naïve method: assign a depth and restorationdirection to 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 $\overline{\mathrm{SCP}^{t-1}\mathrm{TCP}^{t}}$ schneidet;

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

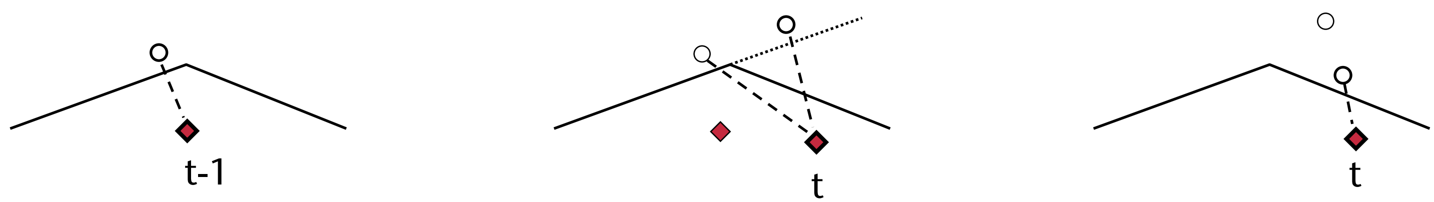
- In order to achieve numerical robustness: lift SCP slightly above the polygons
- Utilize temporal coherence: consider only polygons in the neighborhood of the current SCP

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

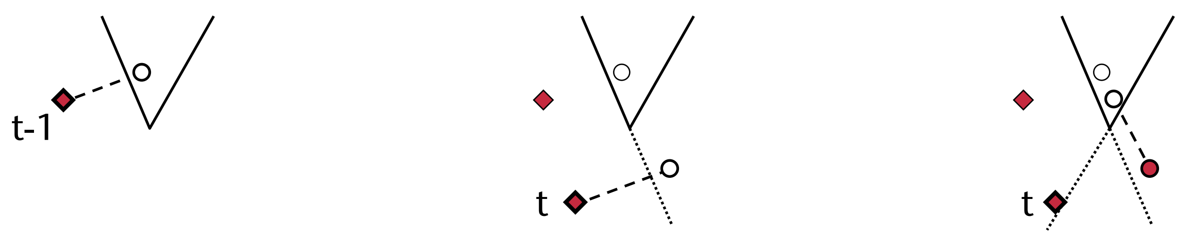
$$\text{minimize} \qquad \left\| \mathbf{x} - \mathbf{x}_{\text{TCP}} \right\|^2$$

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

- 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:



- Question: why is a **constant** update rate so important?

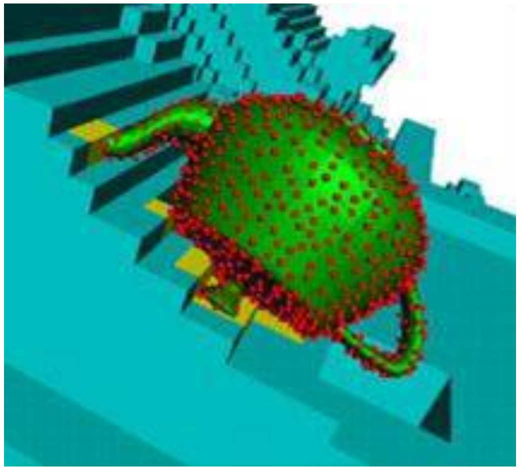
- Answer: because otherwise we get "jitter" (Rütteln, Ruckeln)

The Reason for Device Jitter

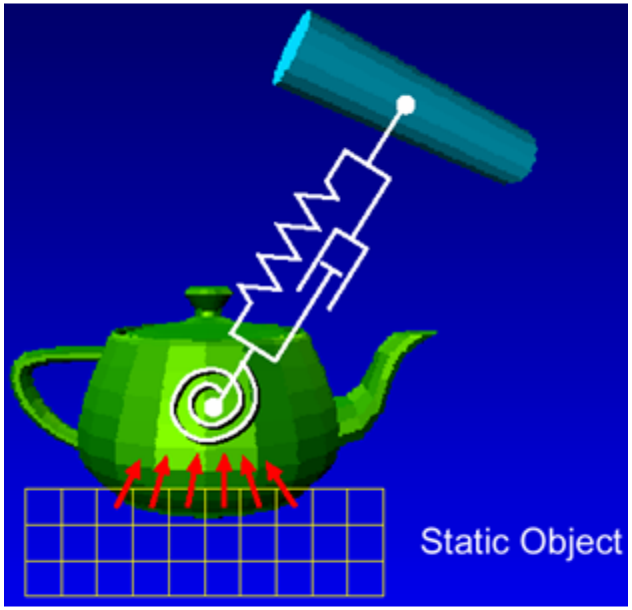
- Assumption:
  - The user is just starting to penetrate an obstacle with the TCP
  - The force generated by the device is still insignificantly small compared to the inertia of the complete system (= user + device)
  - The obstacle has a bit of elasticity (like a spring, possibly a stiff one)
- Consequence: 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 on the user exerted by the device remains the same
  - Then, the device sends its current position to the haptic loop → the penetration depth has increased a lot
  - The force increases much more between two successive frames!

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

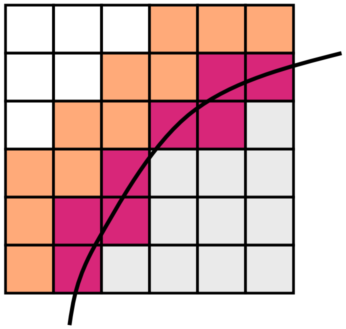


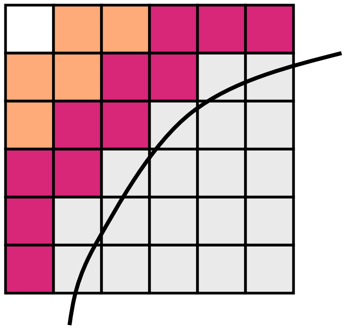
- Overall idea:
  1. Compute forces for all penetrating points
  2. Compute total force on dynamic object
  3. Compute force on haptic handle



The VoxMap

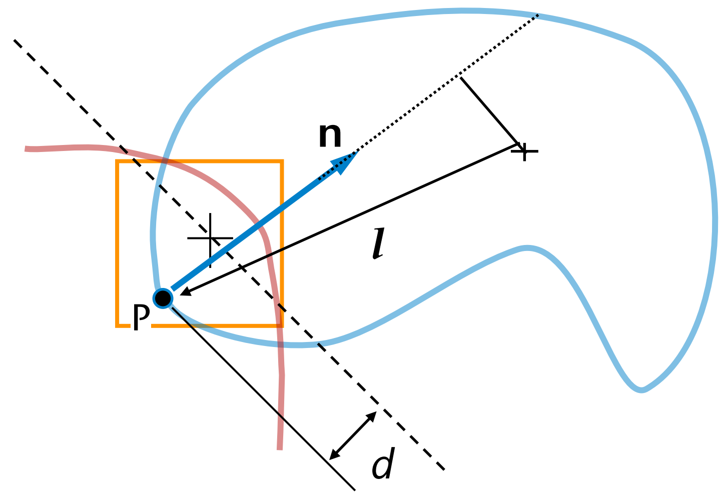
- Voxmap = 3D distance field
- Generation:
  - Scan-convert the surface (in 3D) → voxels that are intersected by the surface
  - Do a breadth-first search starting from the border of the "universe" → all voxels outside any obstacles
  - All other voxels must be inside
    - For each inner voxel, compute 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): $\mathbf{M} = \boldsymbol{l} \times \mathbf{F}$
- 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

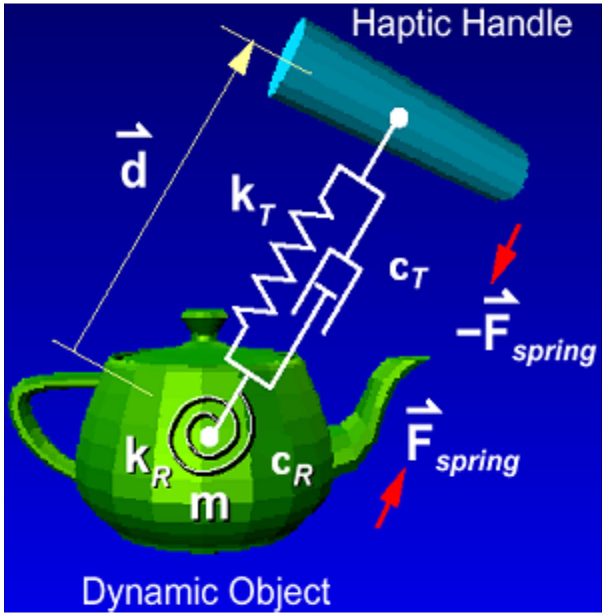
The Virtual Coupling

- 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 **v** that is in the direction of **d**
  - Set viscosity to 0, if **v** points away from the handle

Simulation of the Motion of the Dynamic Object

- Total force acting on the dynamic object:

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

(Analogously for the torques)

- Integrate the following equations of motion:

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

where

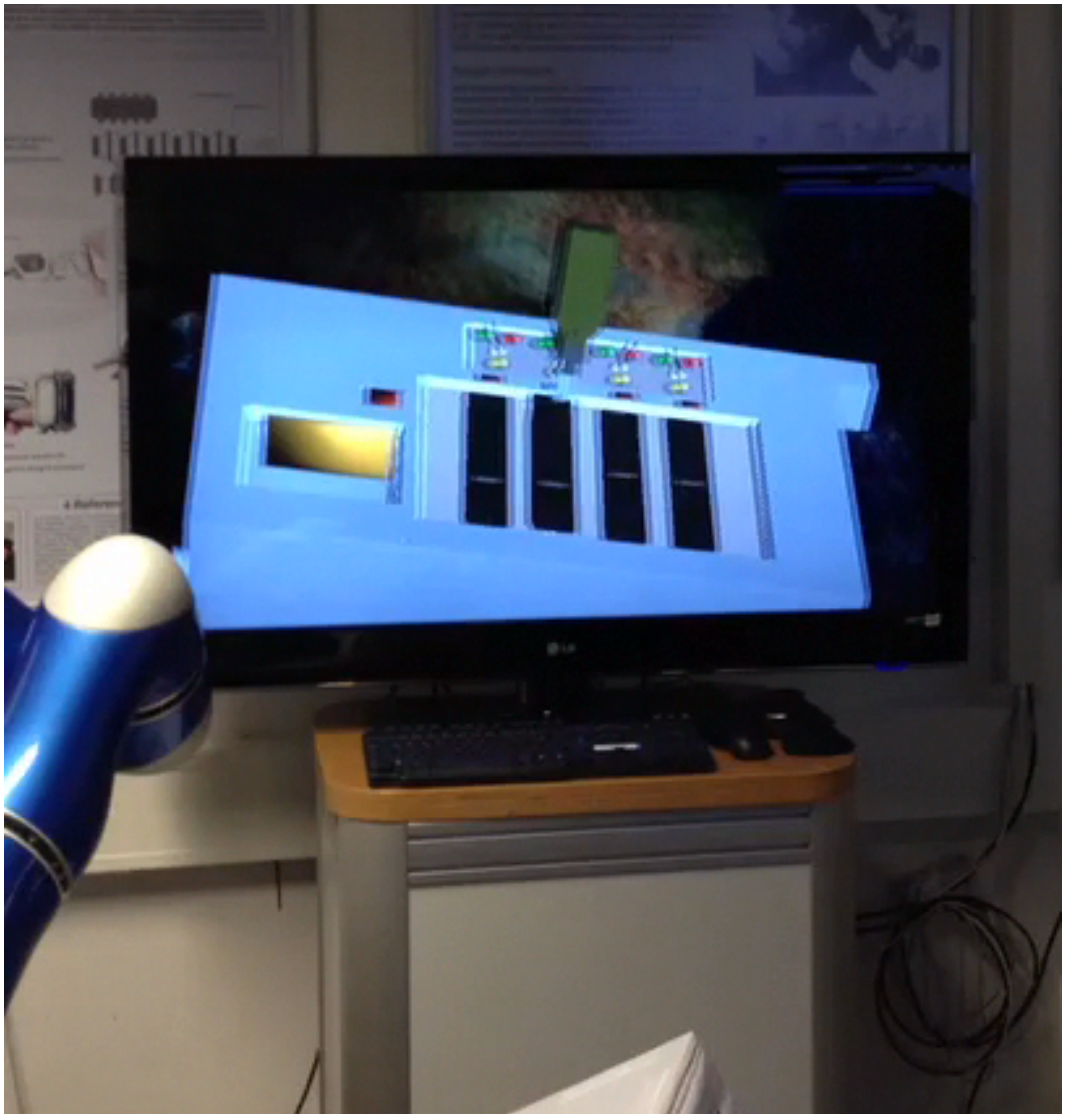
$$F, M = \text{force/torque acting on the center of mass}$$
$$a, \alpha = \text{translational/rotational acceleration}$$
$$m, J = \text{mass/inertia tensor}$$
$$\omega = \text{rotational velocity}$$

- 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

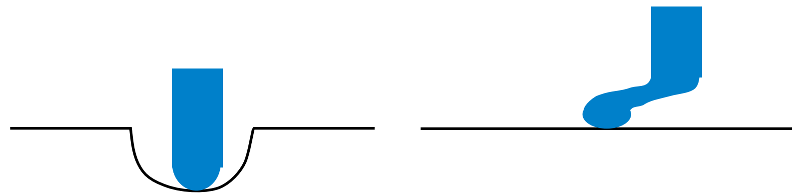
- Virtual coupling = low-pass filter



On-Orbit Servicing, DLR

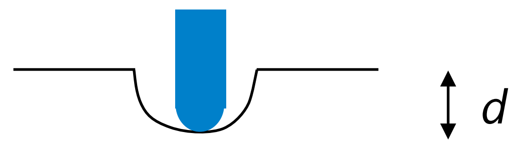
Friction in One Contact Point

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



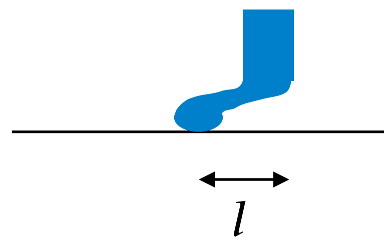
- Forces:
  - Force in direction of the surface normal:

$$F_N = k_N \cdot d$$



  - Force tangential to surface:

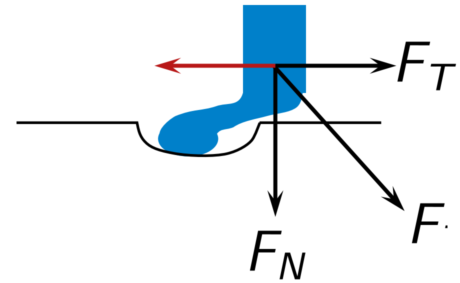
$$F_T = k_T \cdot l$$



- *Point of Attachment*:
  - Point on the surface where first contact occurred
  - Alternatively, determined by the simulation

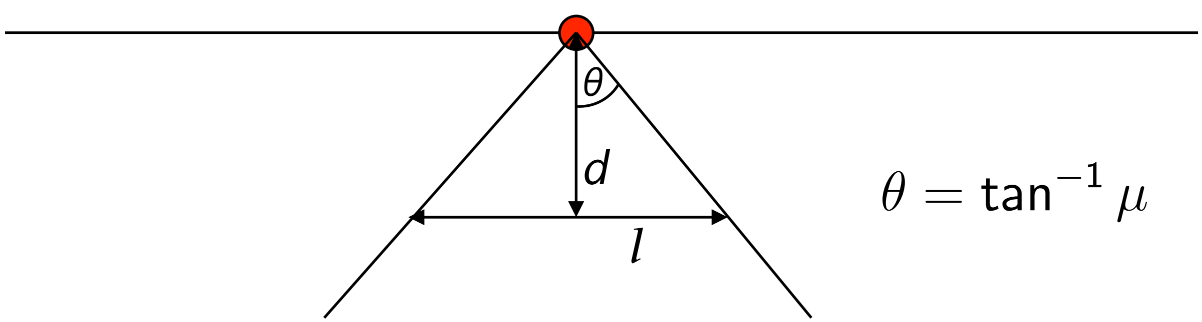
- The Coulomb friction model says:

$$F_f \leq \mu \cdot F_N = \mu \cdot k_N \cdot d$$



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

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



$$\theta = \tan^{-1} \mu$$

Future Applications of Force-Feedback Devices

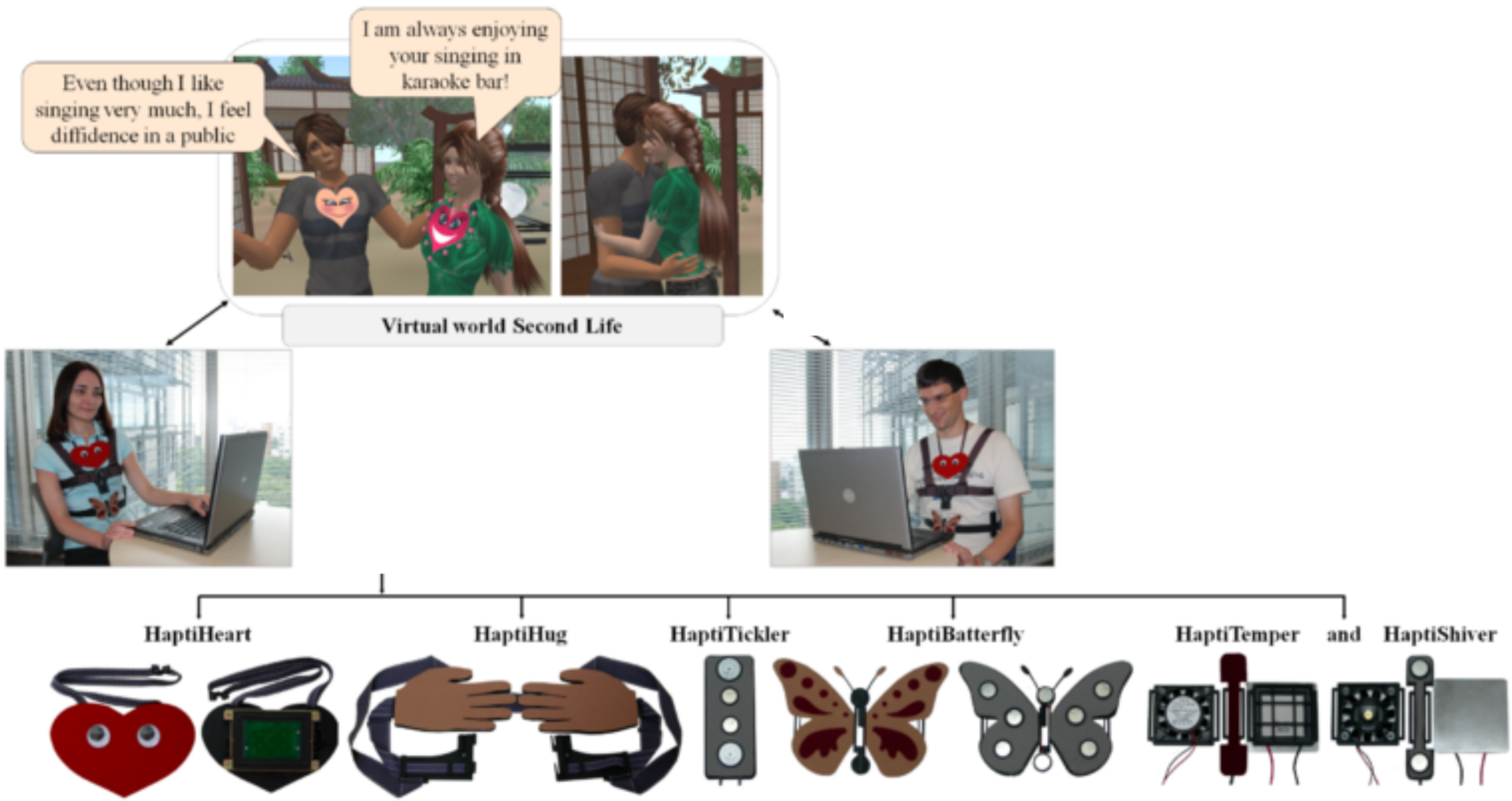
- Micro-surgery (minimally invasive surgery) using remotely controlled robots:

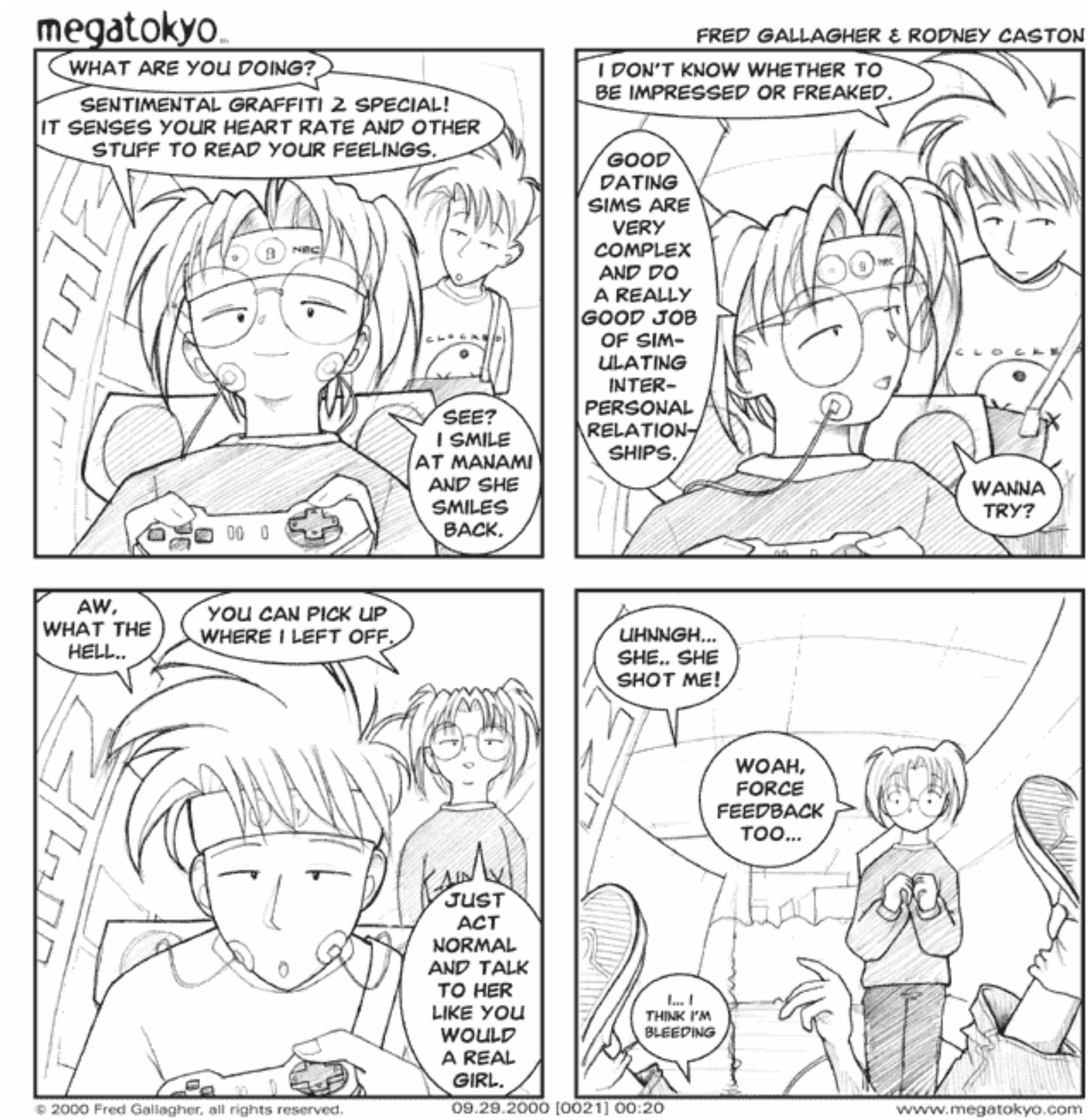
DLR, institute or robotics and mechatronics, Germany

On-orbit servicing of satellites

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

The Rubber-Hand Illusion

Embodiment

M. Slater, 2009 ; body ownership / body distortion

Synchronous multisensory stimulation

Embodiment can change behavior