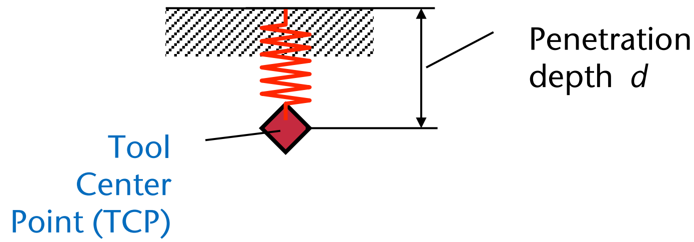
The Principle of Haptic Rendering

- 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

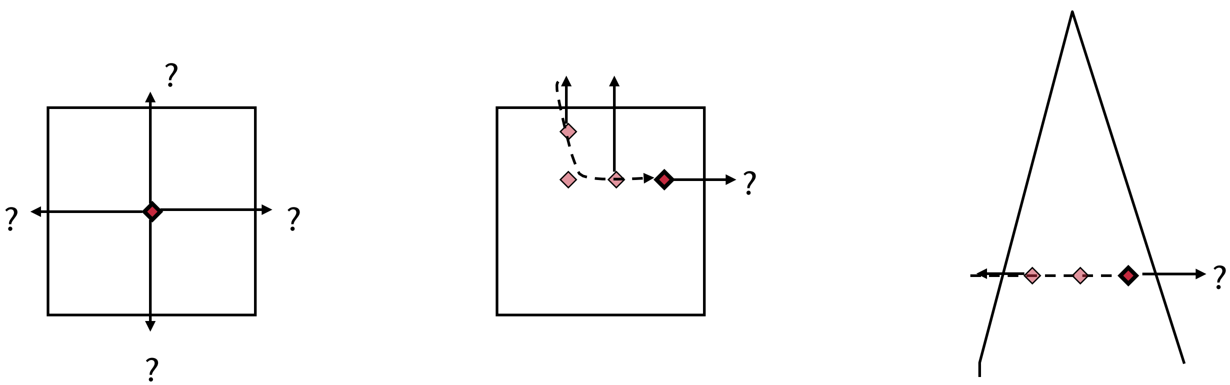
The "Surface Contact Point" Approach

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

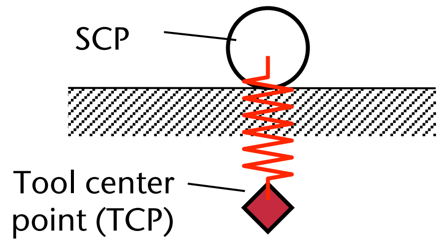
$$F = k \cdot d$$

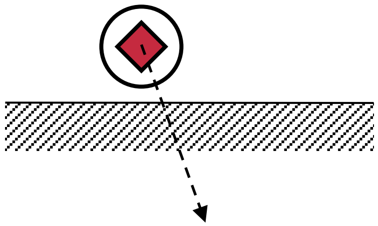


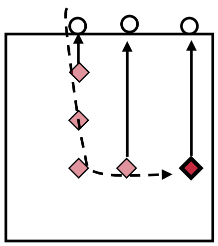
- Question: what exactly is the penetration depth?
  - Naïve method: assign a depth and restorationdirection to each inner point
  - Problem: the history of the TCP is ignored



- 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

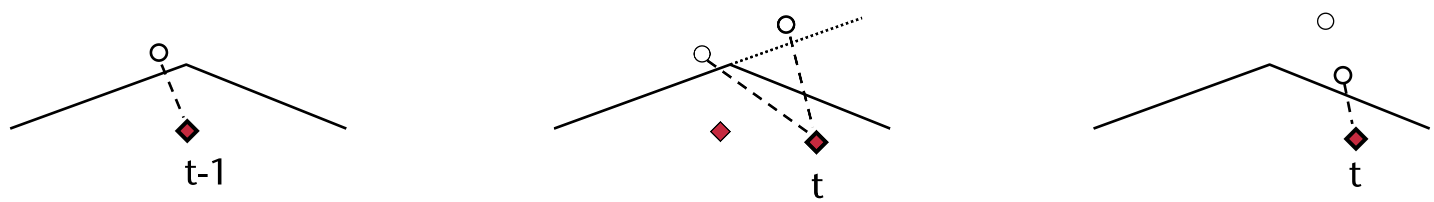
- In order to achieve numerical robustness: lift SCP slightly above the polygons
- Utilize temporal coherence: consider only polygons in the neighborhood of the current SCP

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

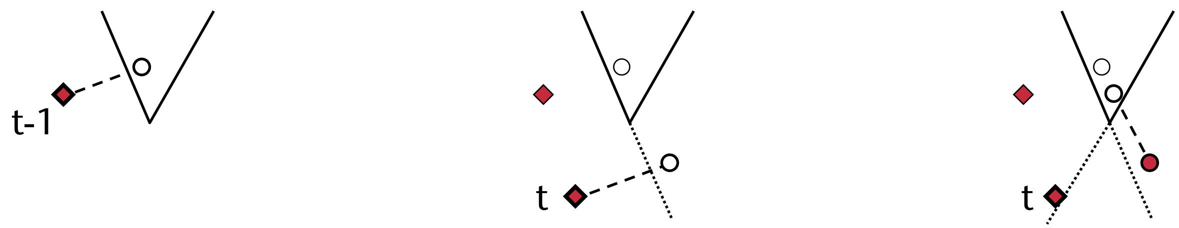
$$\text{minimize} \quad \|\mathbf{x} - \mathbf{x}_{\text{TCP}}\|^2$$

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

- 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?

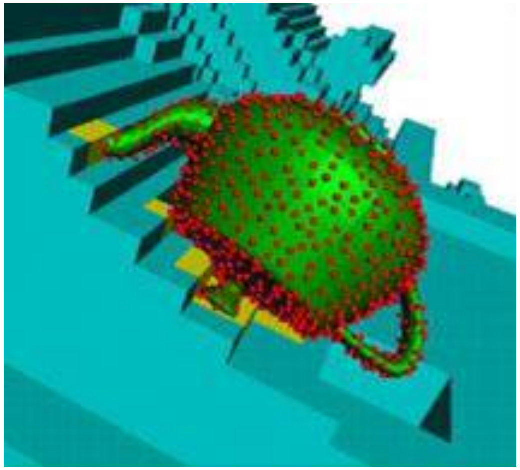
- Answer: because otherwise we get "jitter" (Rütteln, Ruckeln)

The Reason for Device Jitter

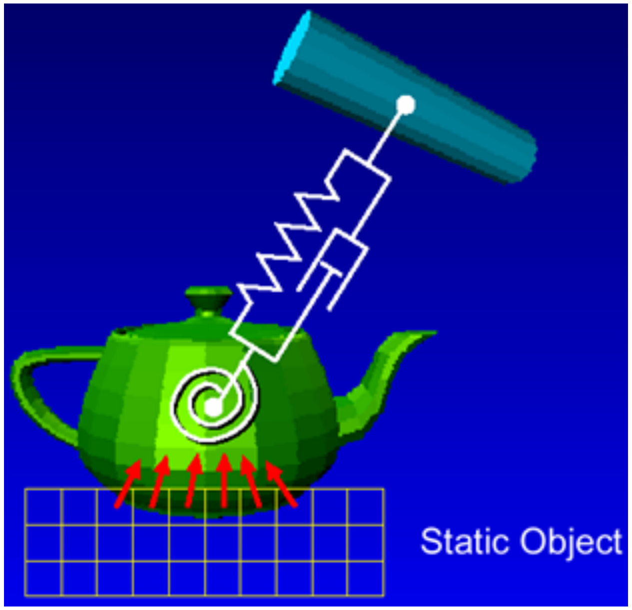
- Assumption:
  - The user is just starting to penetrate an obstacle with the TCP
  - The force generated by the device is still insignificantly small compared to the inertia of the complete system (= user + device)
  - The obstacle has a bit of elasticity (like a spring, possibly a stiff one)
- Consequence: 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

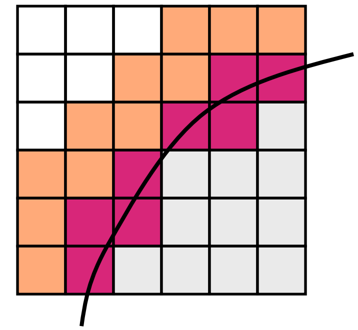


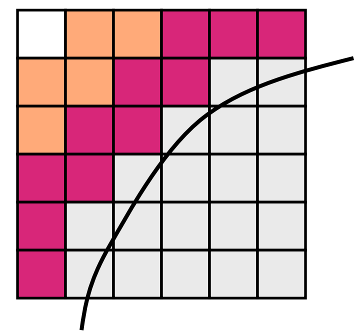
- Overall idea:
  1. Compute forces for all penetrating points
  2. Compute total force on dynamic object
  3. Compute force on haptic handle



The VoxMap

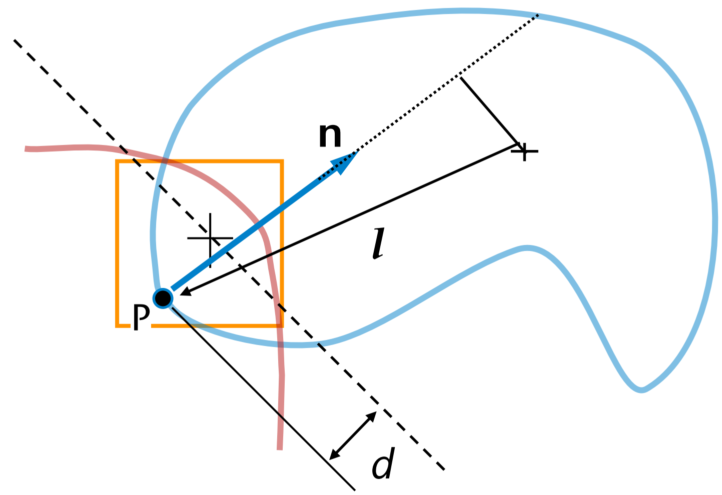
- Voxmap = 3D distance field
- Generation:
  - Scan-convert the surface (in 3D) → voxels that are intersected by the surface
  - Do a breadth-first search starting from the border of the "universe" → all voxels outside any obstacles
  - All other voxels must be inside
    - For each inner voxel, compute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

$$\mathbf{F} = k_T \mathbf{d} - c_T \mathbf{v}$$

$$\mathbf{M} = k_R \theta - c_R \omega$$

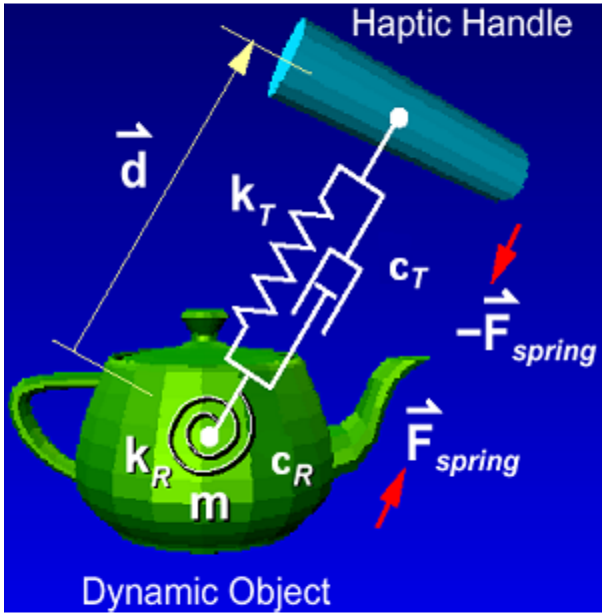
where

$$k_T, c_T = \text{transl. stiffness / viscosity}$$

$$k_R, c_R = \text{rot. stiffness / viscosity}$$

$$\mathbf{d}, \theta = \text{transl./rot. diplacement}$$

$$\mathbf{v}, \omega = \text{transl./rot. velocity}$$



- 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:

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

  (Analogously for the torques)

- Integrate the following equations of motion:

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

  where

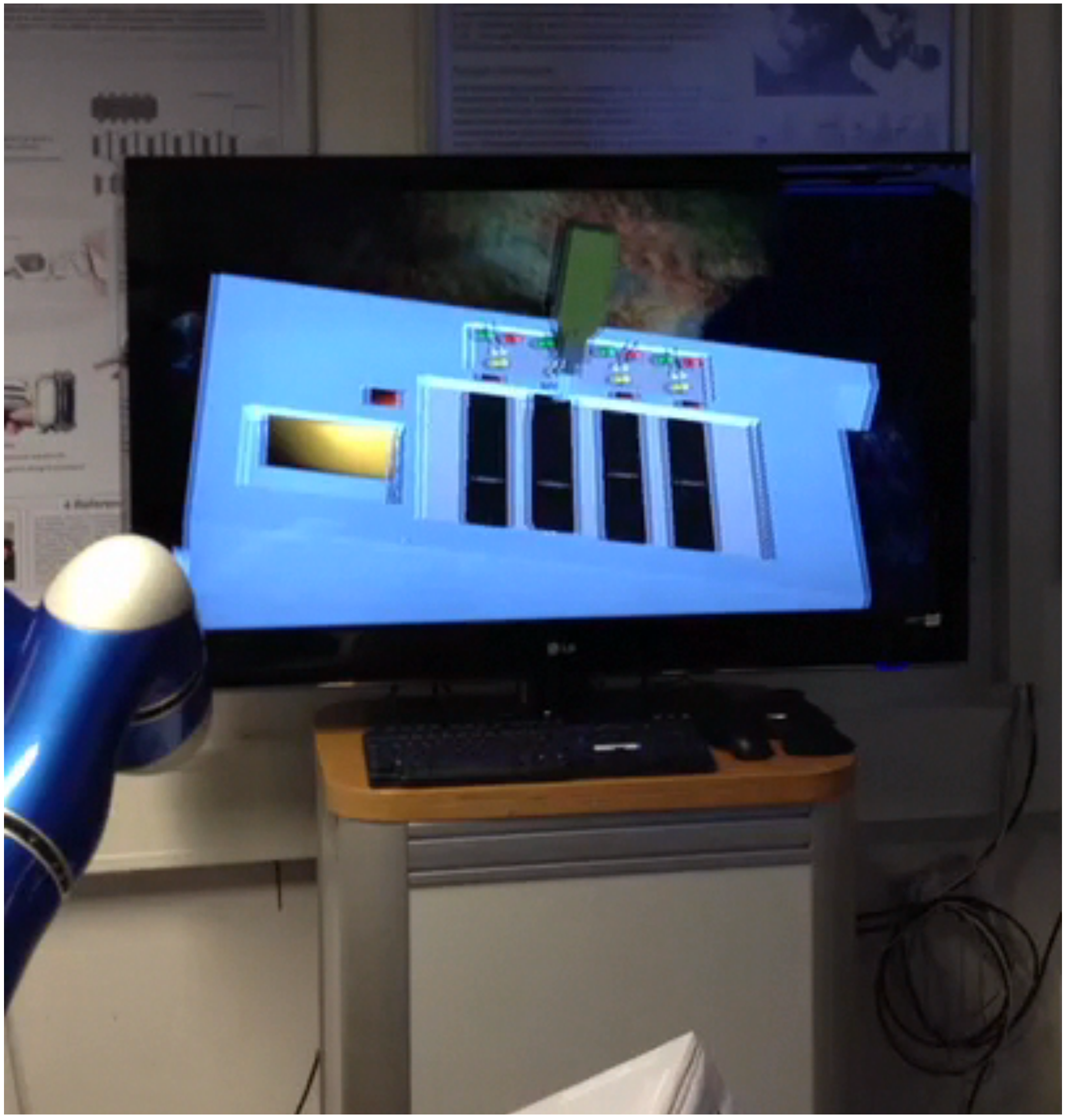
$$F, M = \text{force/torque acting on the center of mass}$$
$$a, \alpha = \text{translational/rotational acceleration}$$
$$m, J = \text{mass/inertia tensor}$$
$$\omega = \text{rotational velocity}$$

- 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

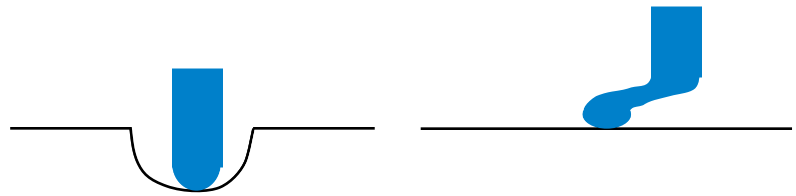
- Virtual coupling = low-pass filter



On-Orbit Servicing, DLR

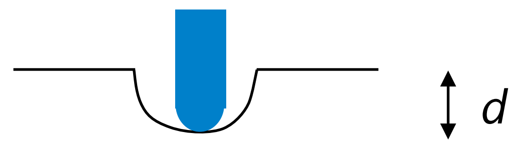
Friction in One Contact Point

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



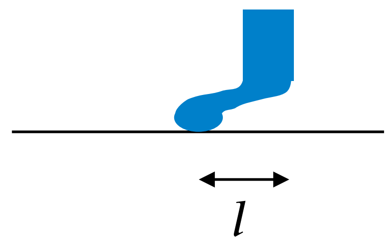
- Forces:
  - Force in direction of the surface normal:

$$F_N = k_N \cdot d$$



  - Force tangential to surface:

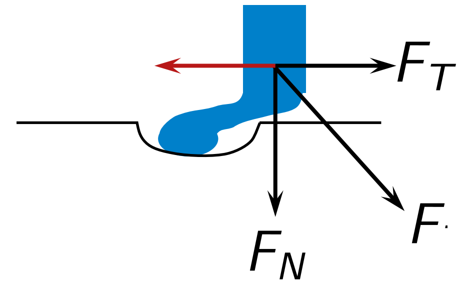
$$F_T = k_T \cdot l$$



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

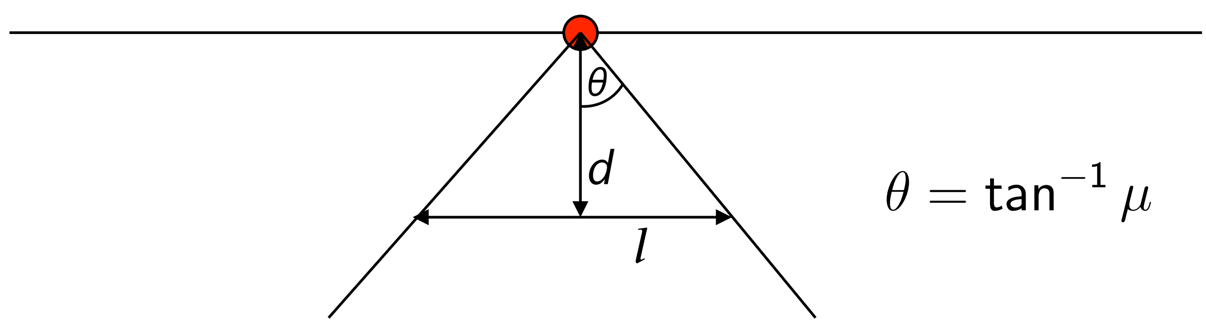
- The Coulomb friction model says:

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



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

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



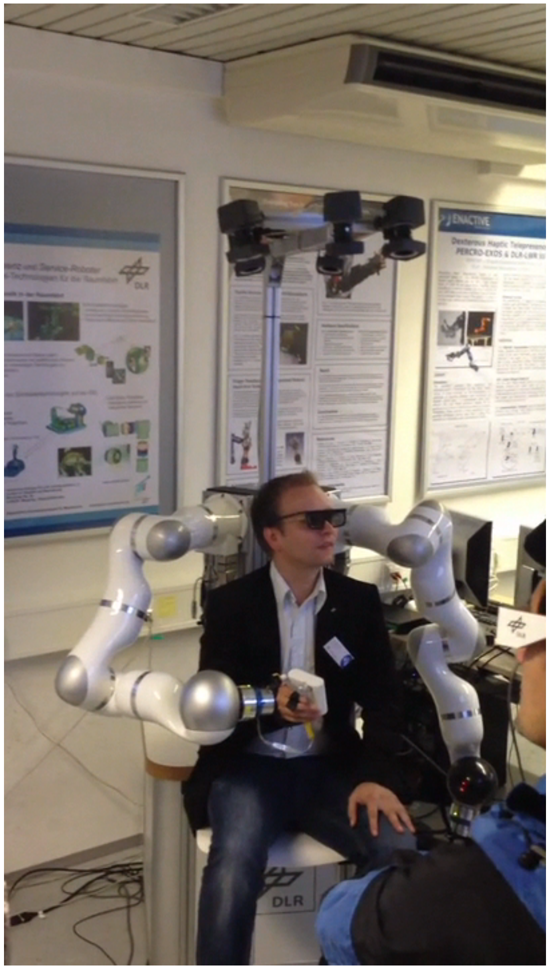
$$\theta = \tan^{-1} \mu$$

Future Applications of Force-Feedback Devices

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

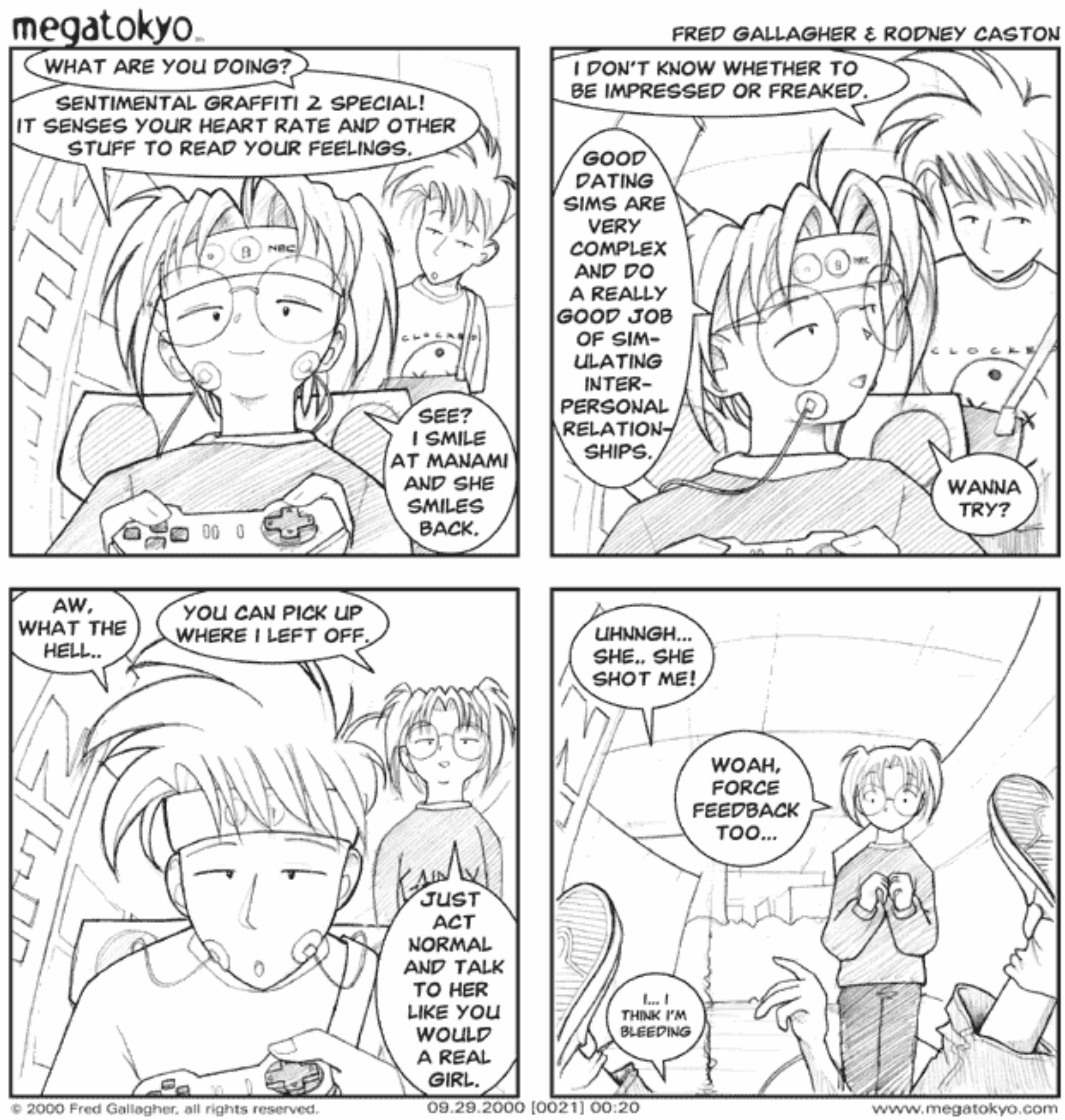
DLR, institute or robotics and mechatronics, Germany

- 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 the only optical illusions …

Surround Haptics Display / Haptic Chair by Disney Research, Pittsburgh

The Rubber-Hand Illusion