Rainbow

Haptic rendering

Claudio Pacchierotti

CNRS

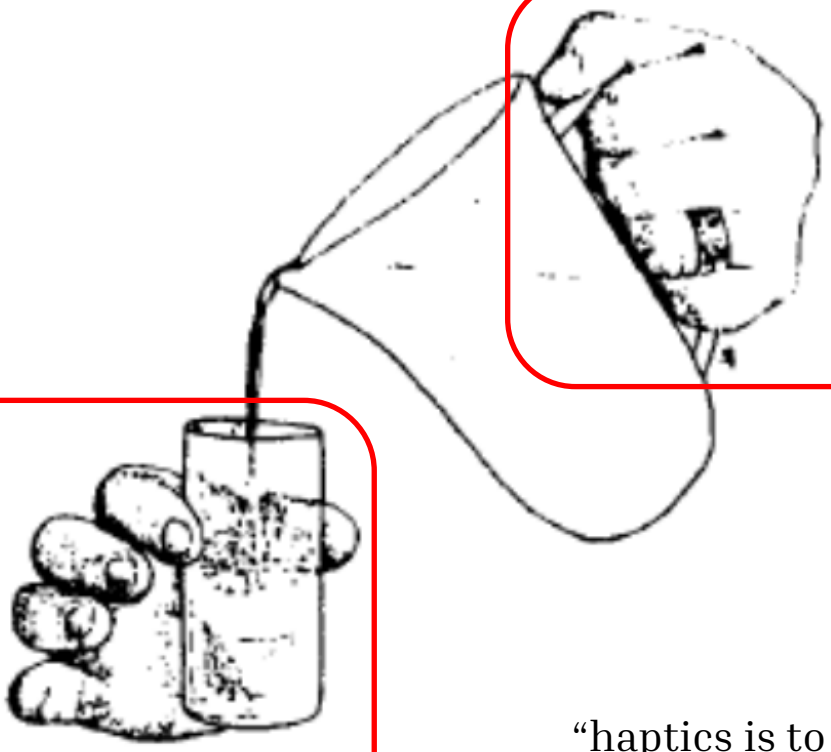
"Haptic rendering is the process of computing and generating forces in response to user interactions with virtual objects."

–J. Kenneth Salisbury

Today's Outline

- The haptic loop
- God-object/Proxy model
- Examples of rendering of a wall and a sphere
- Data-driven rendering
- Personalization of haptic rendering techniques

Haptics = Cutaneous and kinesthetic stimuli

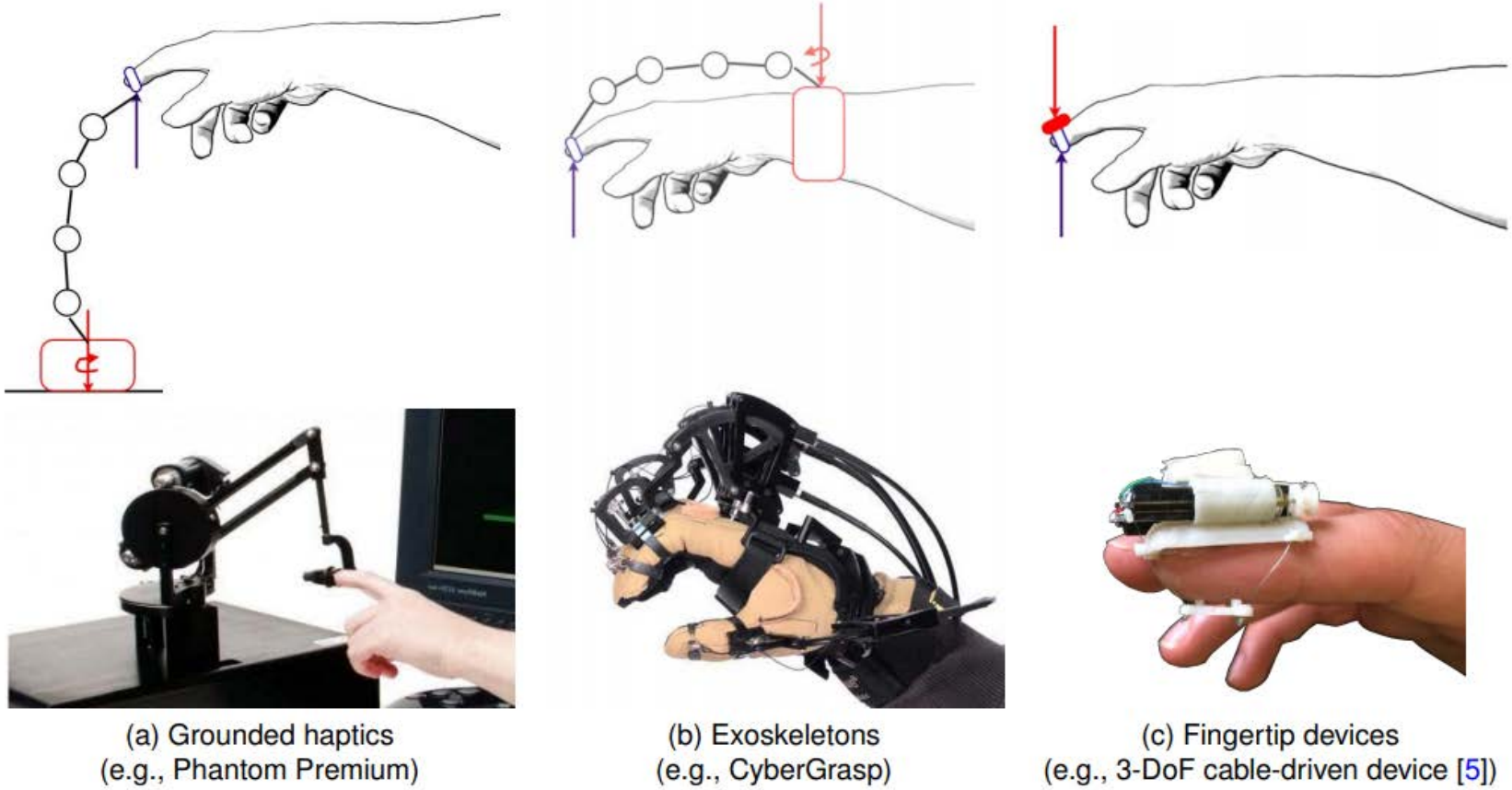


Kinesthesia
Location/configuration
Motion
Force
Compliance

Cutaneous
Temperature
Texture
Slip
Vibration
Force

"haptics is to touch as optics is to sight"
(A. M. Okamura), 2008.

From grounded to wearable haptics



(a) Grounded haptics (e.g., Phantom Premium)

(b) Exoskeletons (e.g., CyberGrasp)

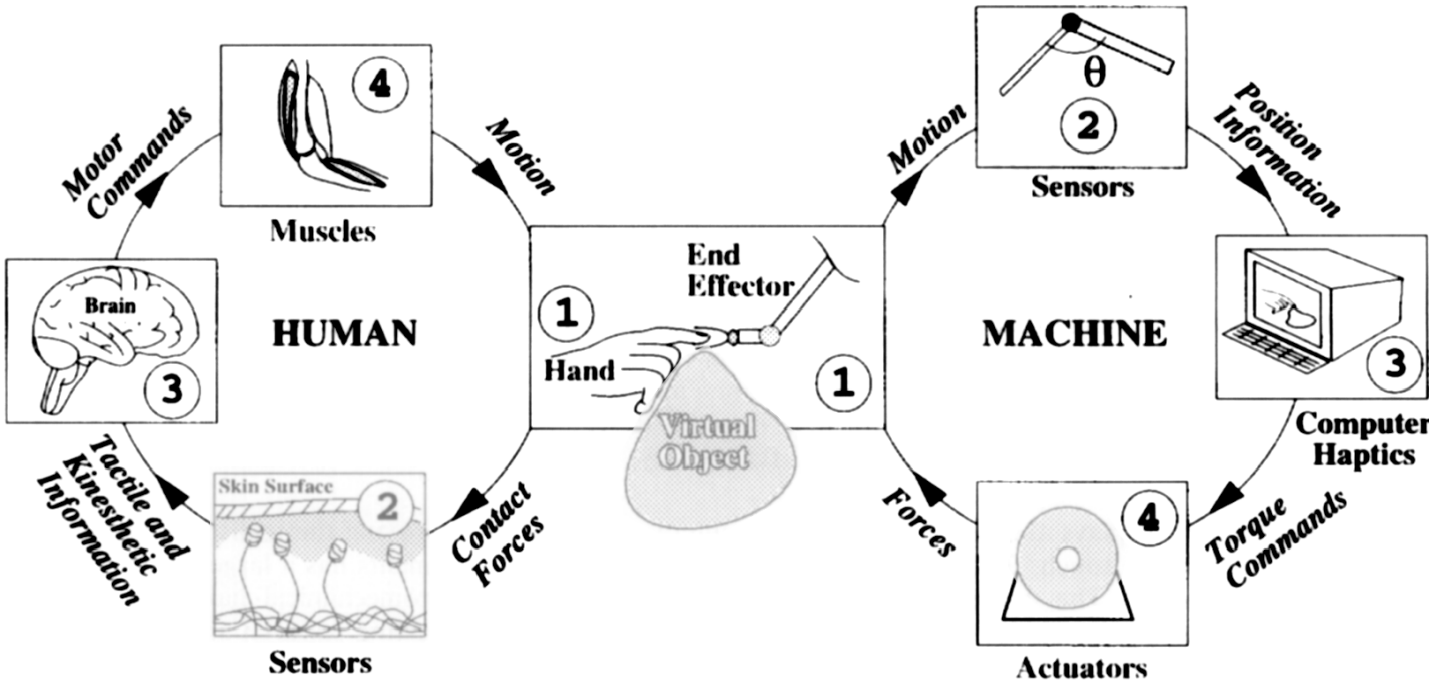
(c) Fingertip devices (e.g., 3-DoF cable-driven device [5])

C. Pacchierotti, S. Sinclair, M. Solazzi, A. Frisoli, V. Hayward, D. Prattichizzo. “Wearable haptic systems for the fingertip and the hand: taxonomy, review, and perspectives” IEEE Transactions on Haptics, 10(4):580-600, 2017.

Today's Outline

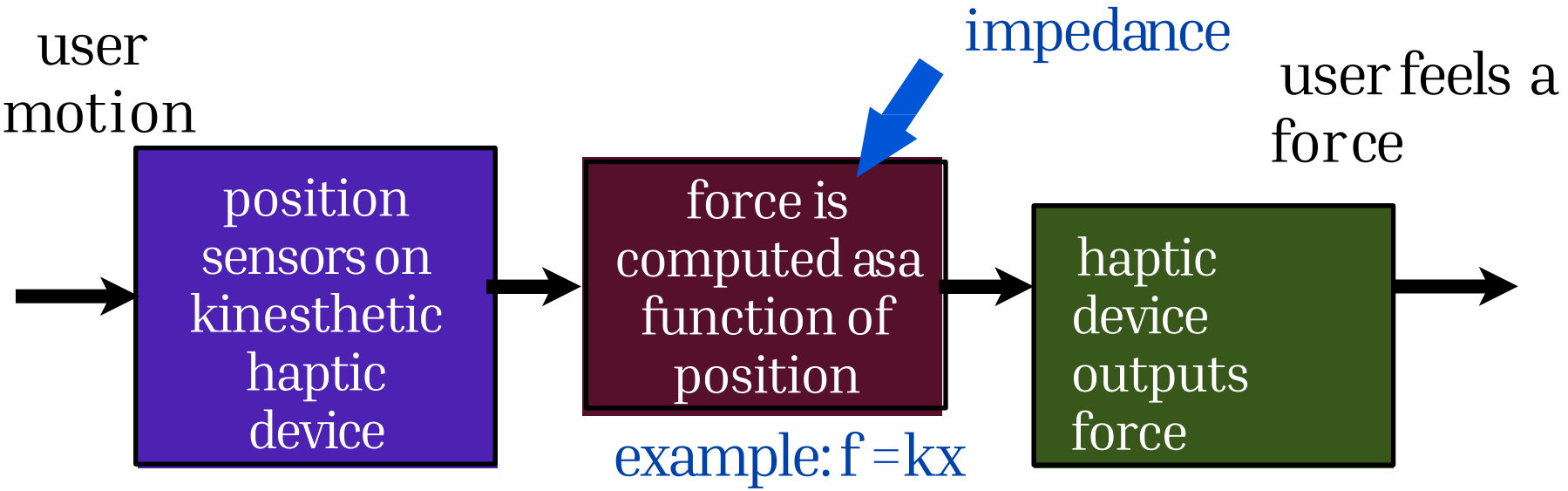
- **The haptic loop**
- God-object/Proxy model
- Examples of rendering of a wall and a sphere
- Data-driven rendering
- Personalization of haptic rendering techniques

Information & Power Flows



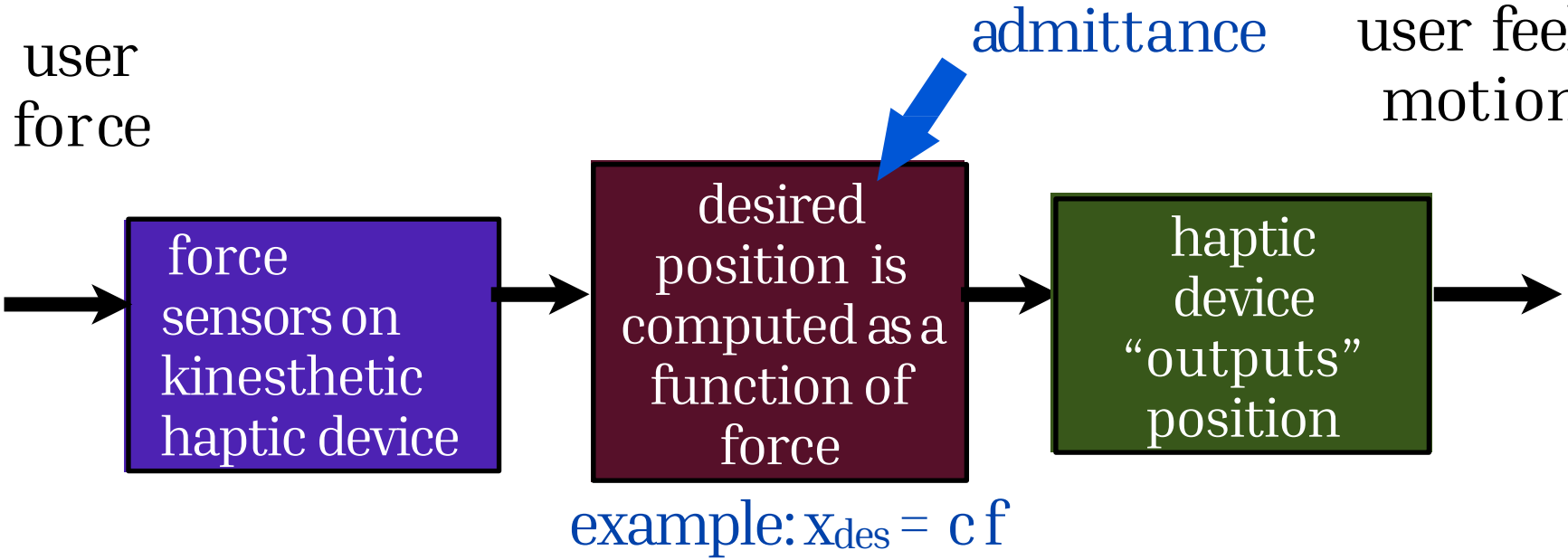
[From M. Srinivasan and C. Basdogan, *Computers & Graphics* 21(4), 1997.]

Impedance-type kinesthetic devices



most force feedback devices are of the "impedance" type

Admittance-type kinesthetic devices



"admittance"-type devices are not as common

Impedance vs. Admittance Control

- Impedance devices

  sensed position

  commanded force

- Admittance devices

  sensed force

  commanded position

Impedance vs. Admittance Devices

- Impedance haptic devices

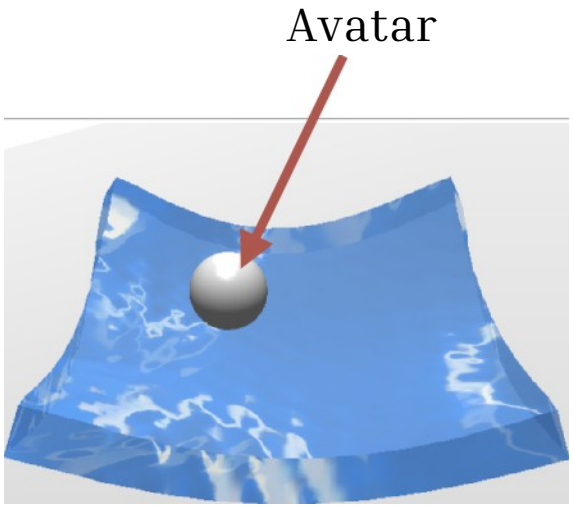
  cheaper to build, back-drivable

- Admittance haptic devices

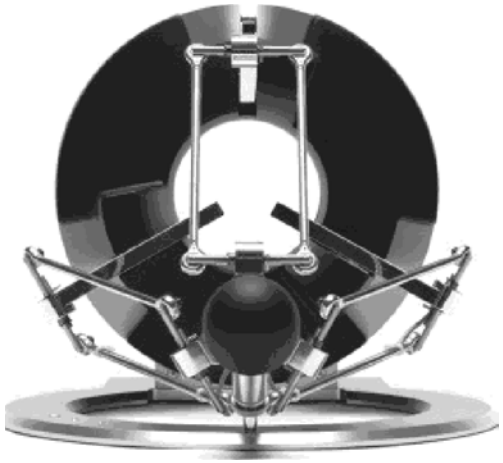
higher range of forces, requires force sensor, generally less common

The Basics

How does a basic visual-haptic simulation work?



Virtual Environment



Haptic Interface

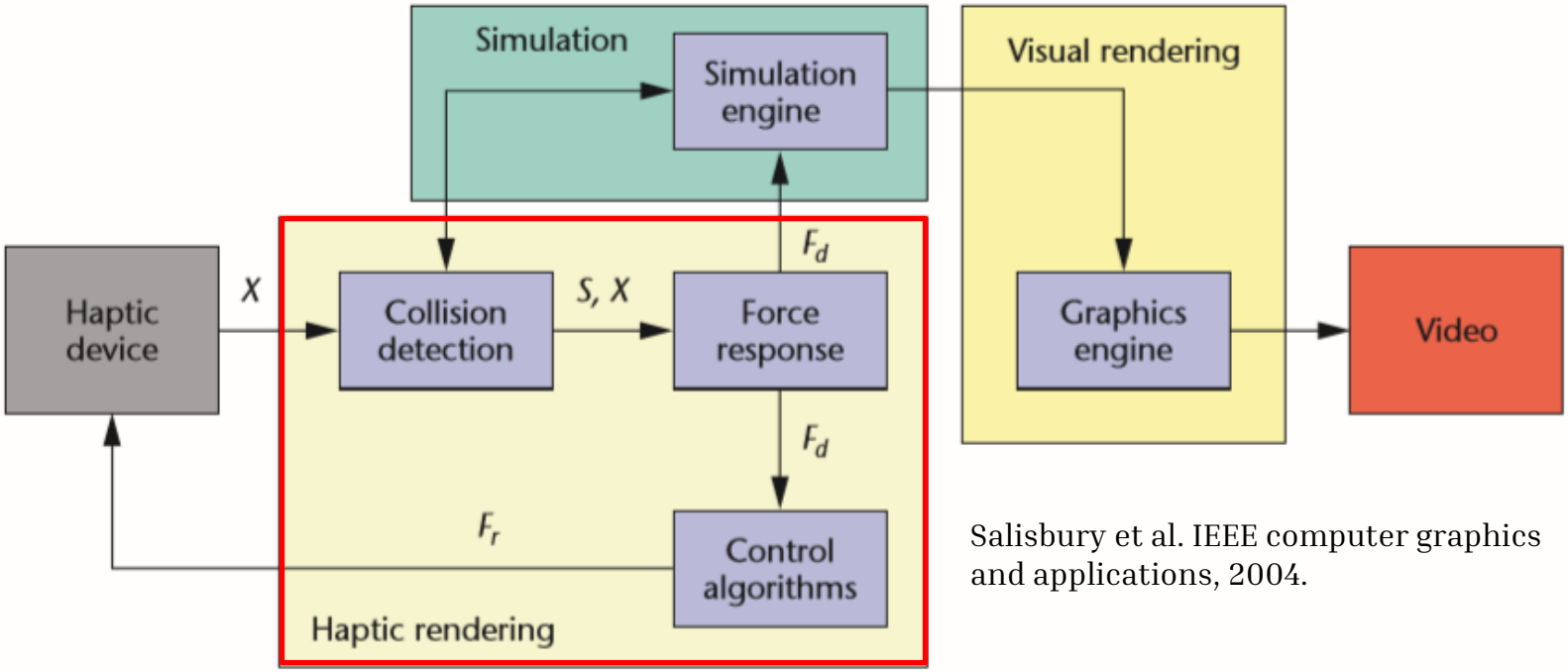
The Basics

How does a basic visual-haptic simulation work?

Today's Outline

- The haptic loop
- **God-object/Proxy model**
- Examples of rendering of a wall and a sphere
- Data-driven rendering
- Personalization of haptic rendering techniques

Components of the haptic simulation



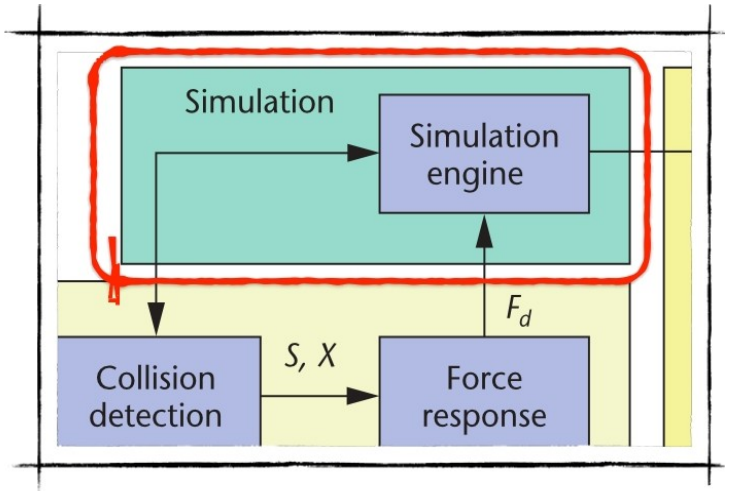
Salisbury et al. IEEE computer graphics and applications, 2004.

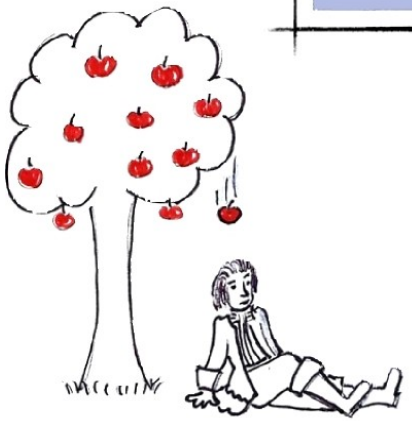
Collision-detection algorithms provide information about contacts **S** occurring between an avatar at position **X** and objects in the virtual environment. Force-response algorithms return the *ideal* interaction force F_d between avatar and virtual objects. Control algorithms return a force F_r to the user, approximating the ideal interaction force to the best of the device's capabilities.

Components of the haptic simulation

The Virtual Environment

- representations of virtual objects
- real-time simulation of physical behaviour

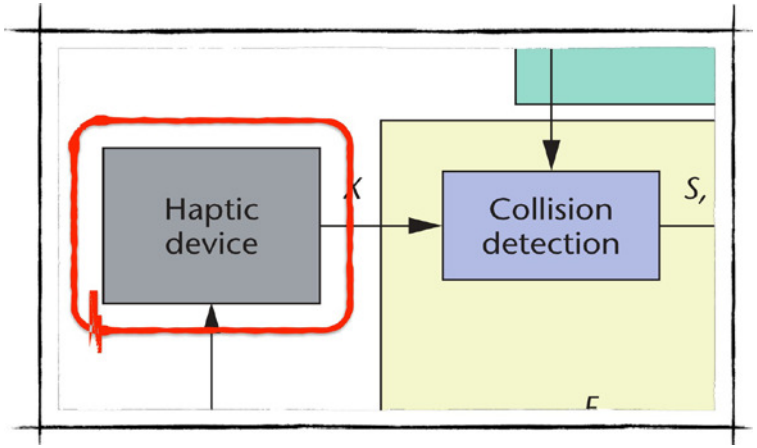




Components of the haptic simulation

Haptic Device

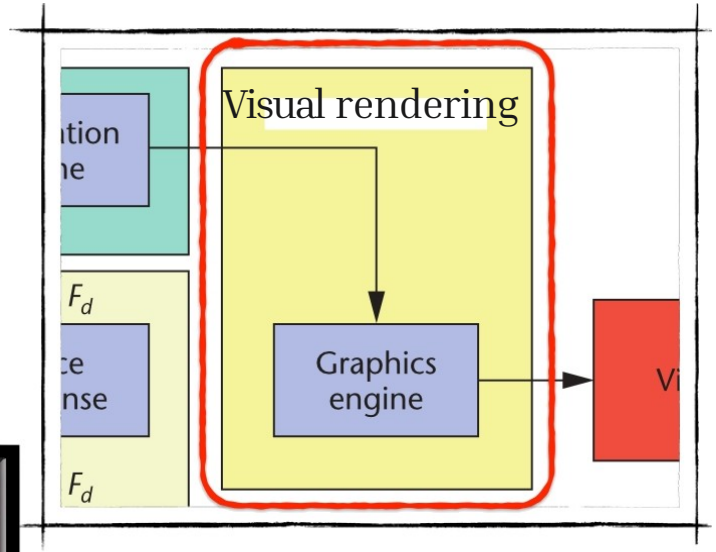
We discussed them yesterday



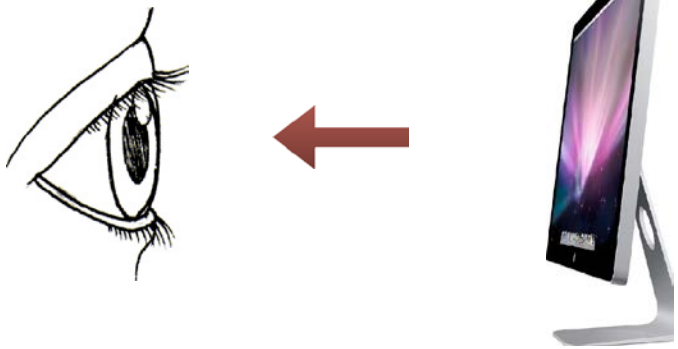
Components of the haptic simulation

Visual Rendering

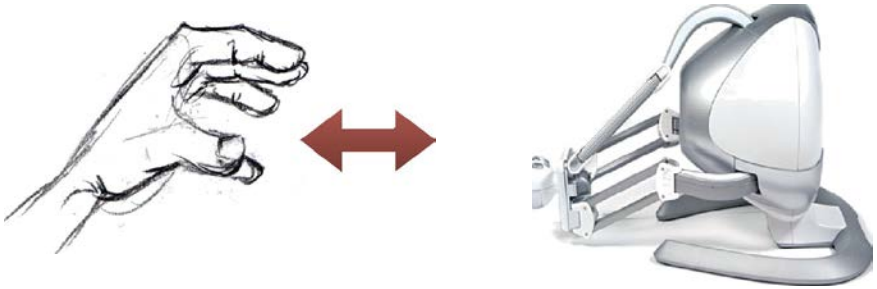
- Given a virtual environment, render its state on the screen (in real time)
- CHAI3D is framework to design haptic interactions in VR



Haptic vs. Visual Rendering



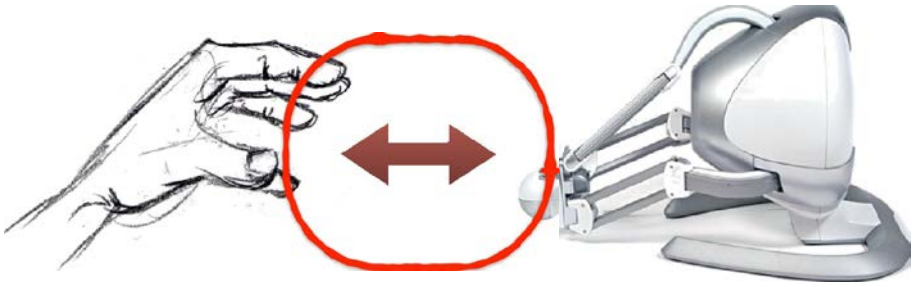
Visual Rendering



Haptic Rendering

Bi-DirectionaIity

Bi-directional information flow is a distinguishing feature of haptic interfaces.



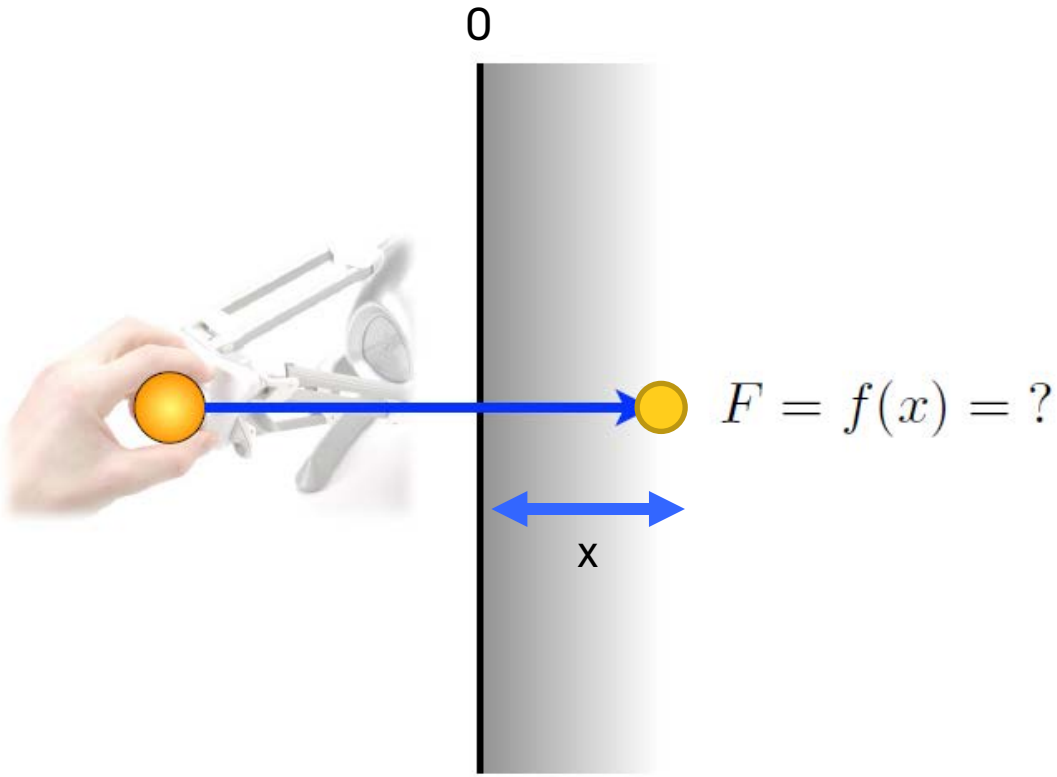
Haptic Rendering

This feature has other consequences that we will not, however, cover in details in this class.

Simple interaction: virtual wall

- A plane is one of the simplest virtual environments we can conceive and render

How can we render such a "virtual wall"?



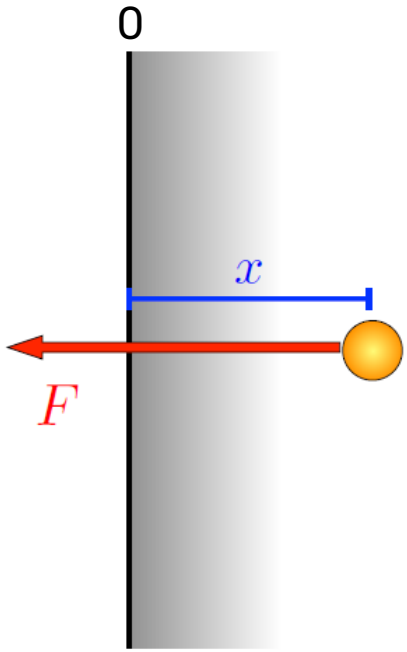
Try!

Simple interaction: virtual wall

- A plane is one of the simplest virtual environments we can conceive and render

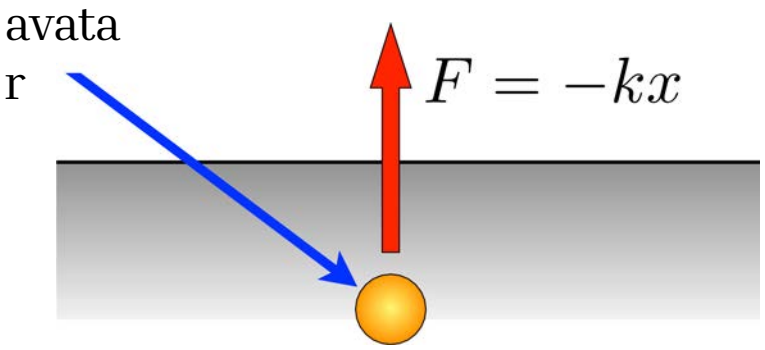
A first idea is to consider a spring

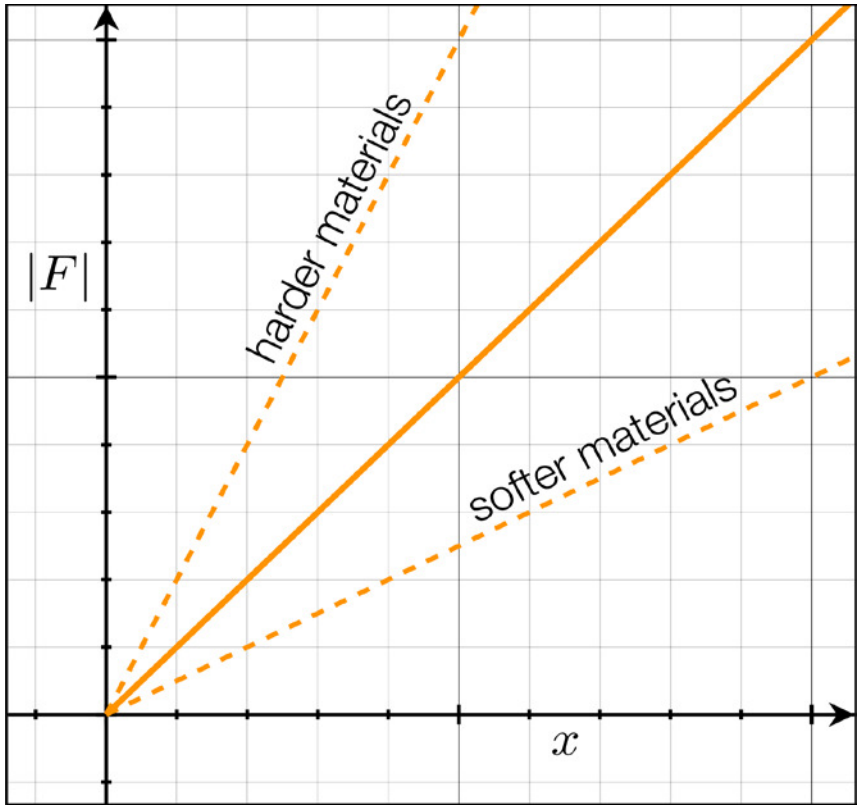
$$F(x) = \begin{cases} -kx & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$



Simple interaction: virtual wall

The elastic constant (the stiffness k) affects how the virtual wall feels.





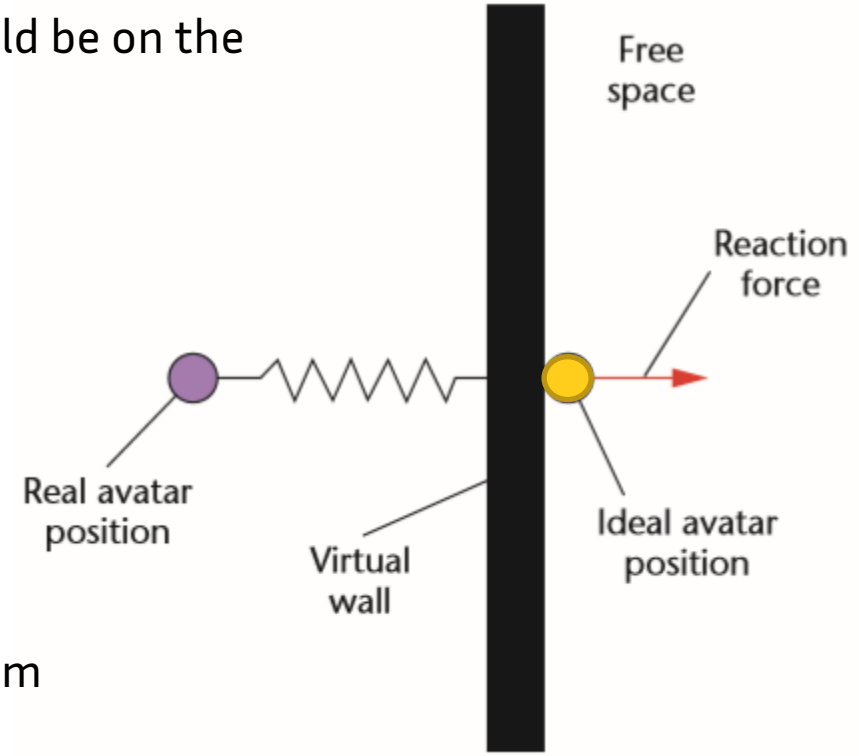
Simple interaction: virtual wall

The ideal position of the avatar should be on the surface, but it is not.

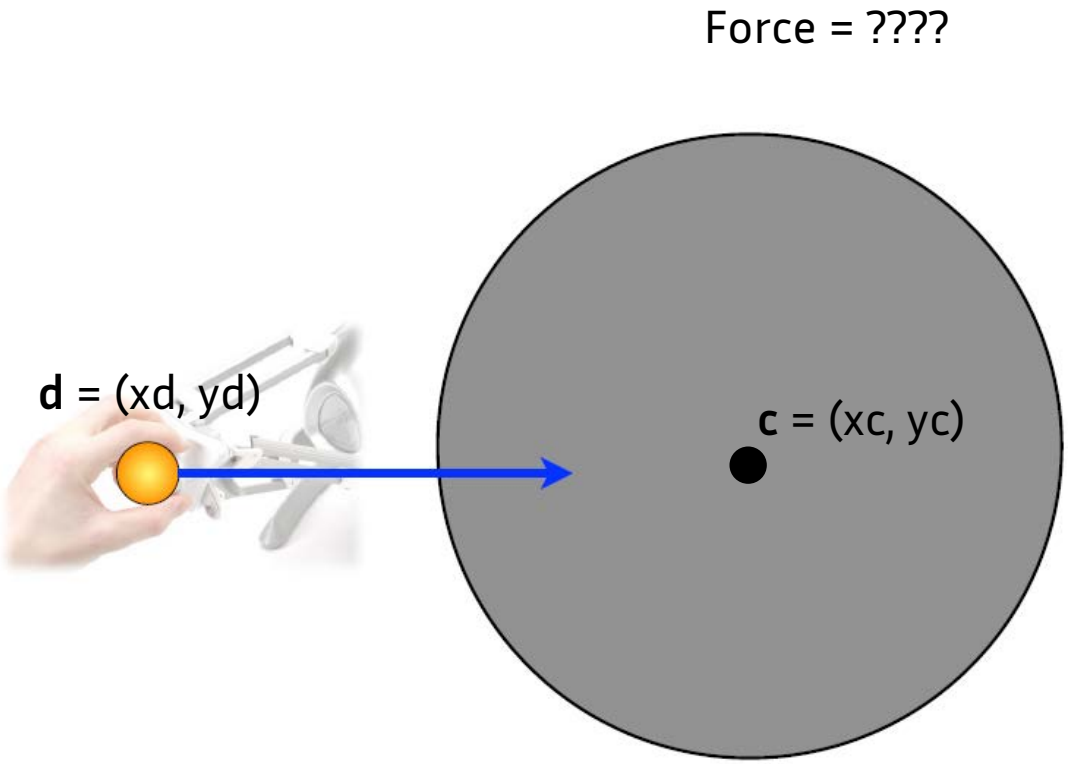
Why?

Well, regardless, we can still visually show the avatar where it should be.

Issues of this approach can come from angles/corners.

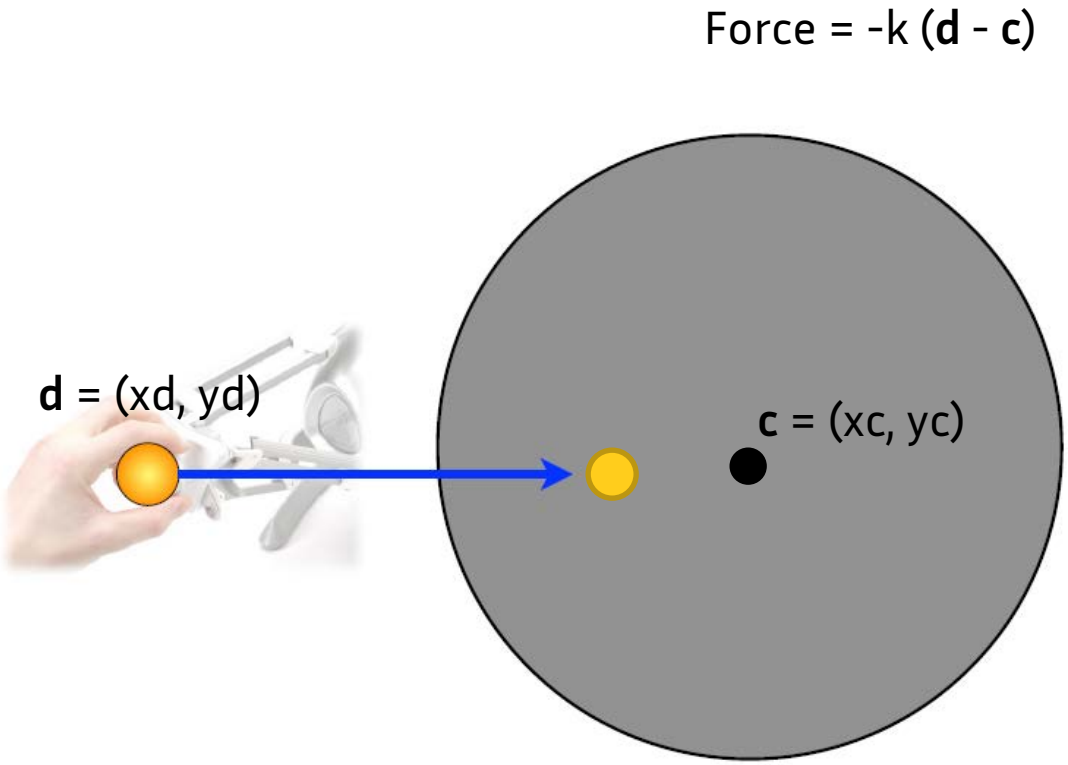


What about another shape?

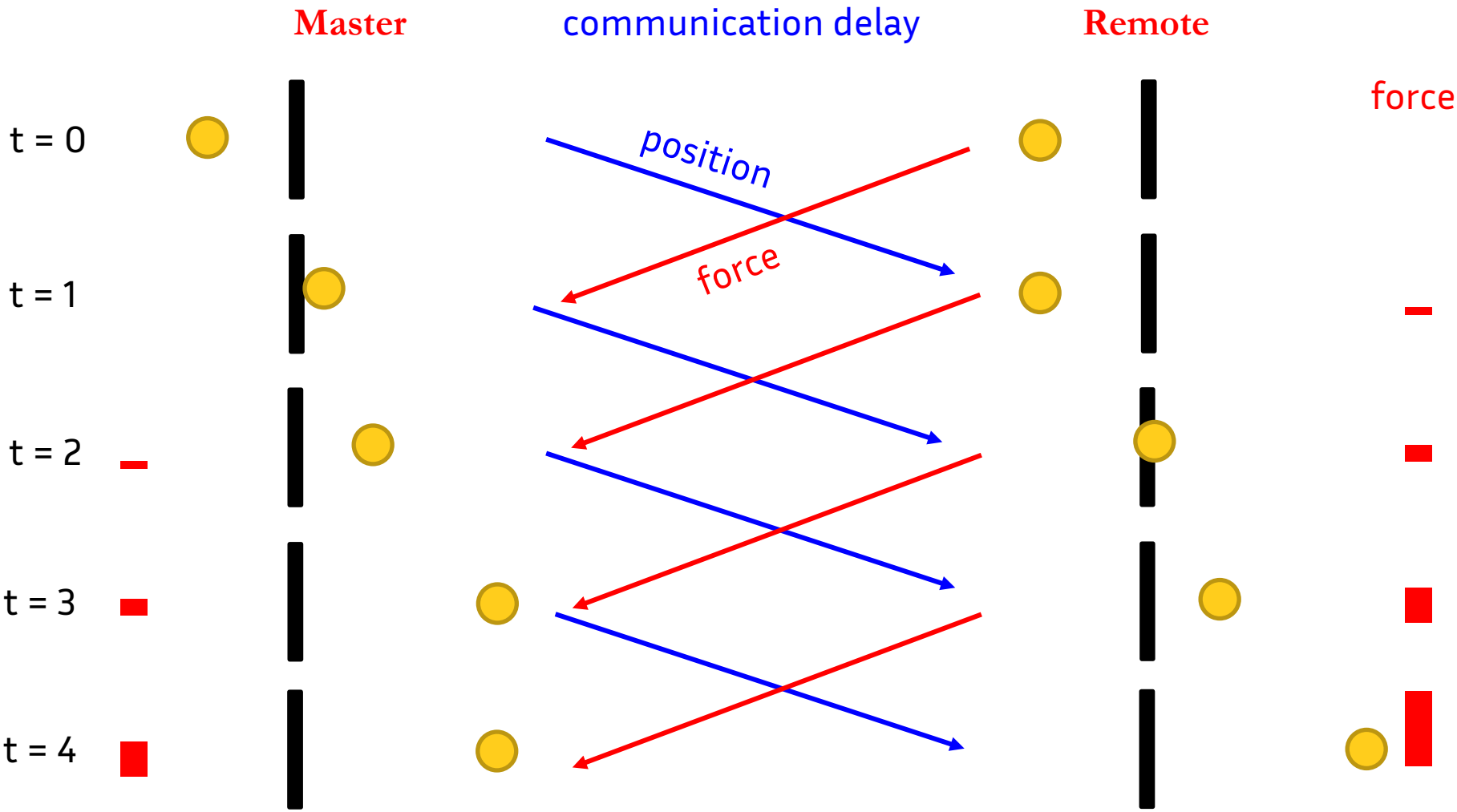


Try!

What about another shape?

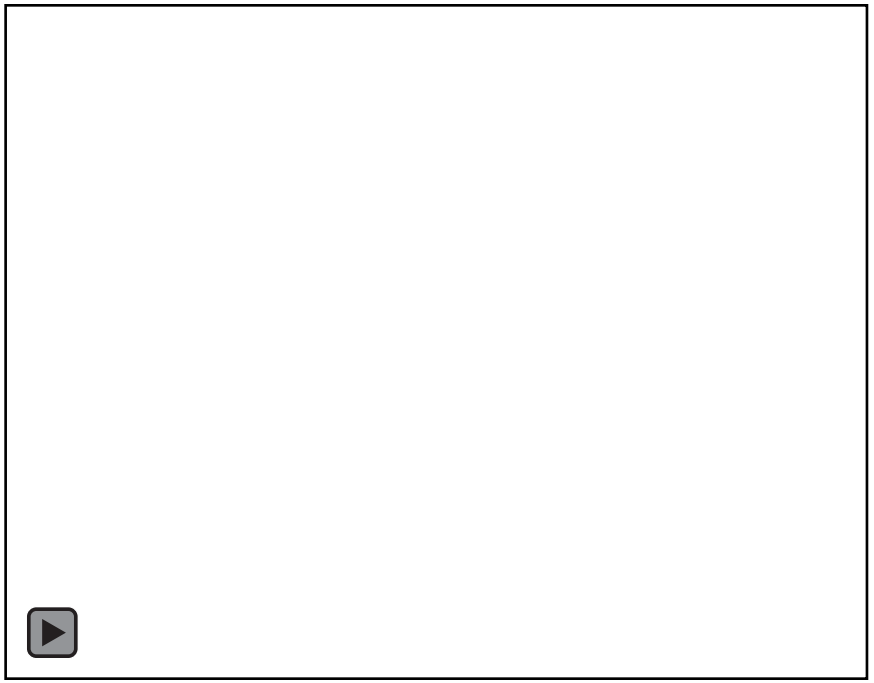


Issues with spring-only interactions



Issues with spring-only interactions

Hard contact
(Seo et al., 2011)



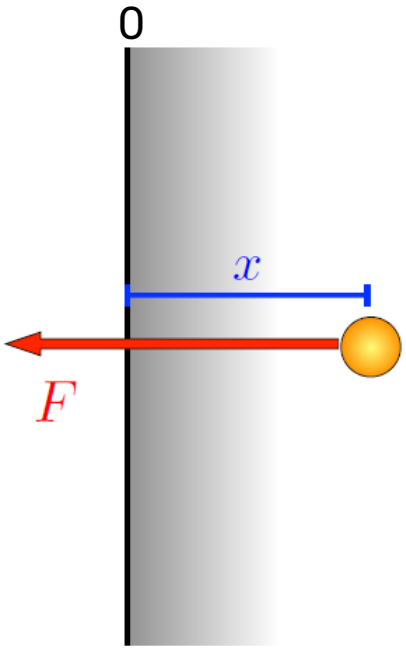
Communication delay
(Jazayeri and Tavakoli, 2011)

Simple interaction: adding a damper

Considering only a spring might feel too reactive and even introduce severe safety issues in the presence of communication delays.

We can add a damper to the system.

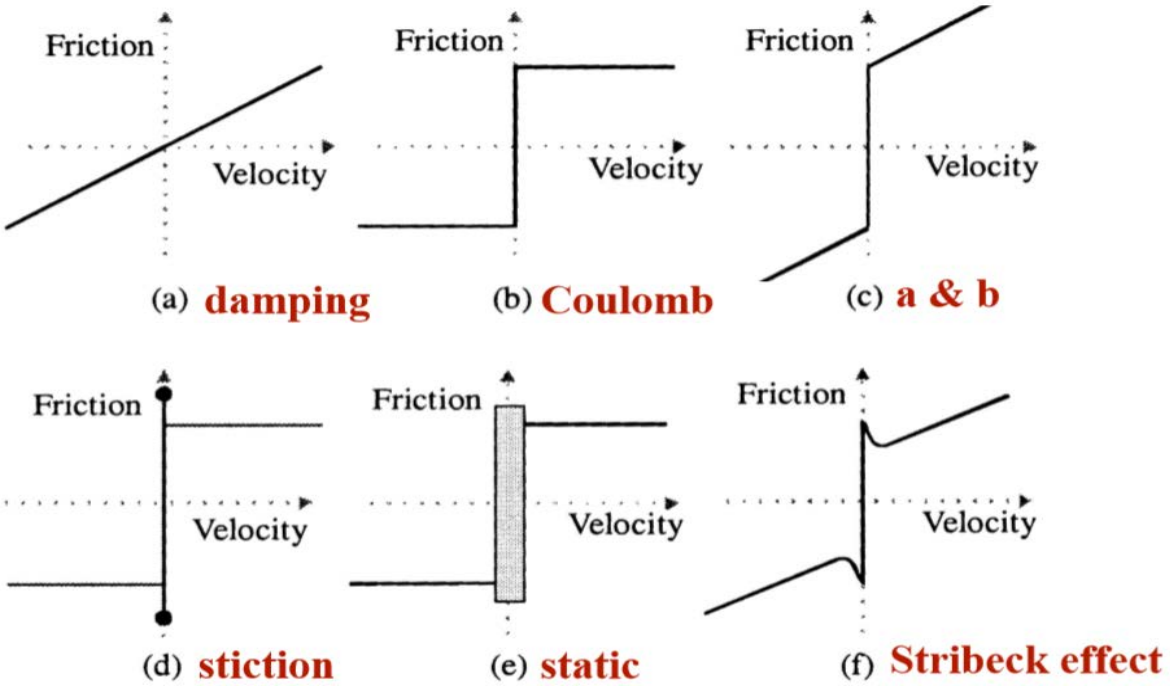
$$F(x) = \begin{cases} -kx - b\dot{x} & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$



Simple interaction: “frictional” damping

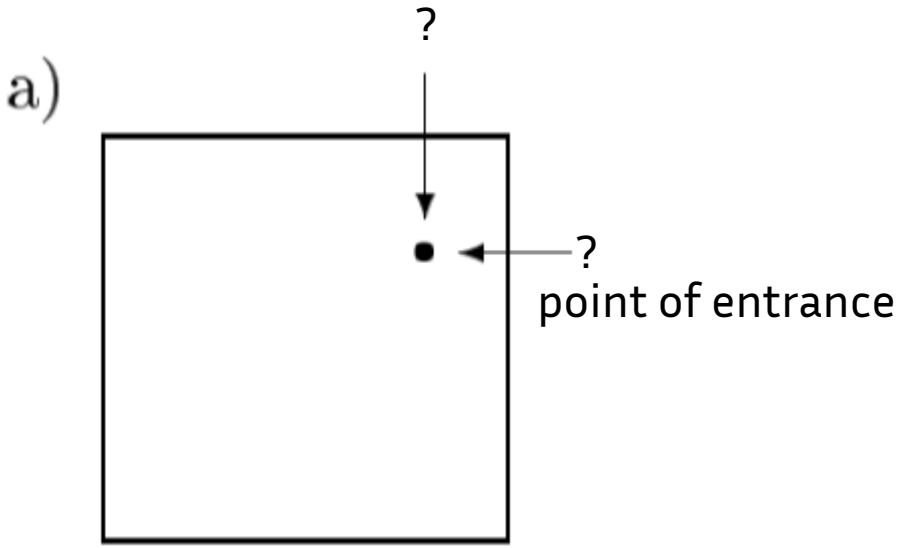
However, doing all this, surfaces can feel unnaturally slippery. Of course, friction would help, but it can be difficult to implement.

A good idea is to add a damping to motions *parallel* to the surface.

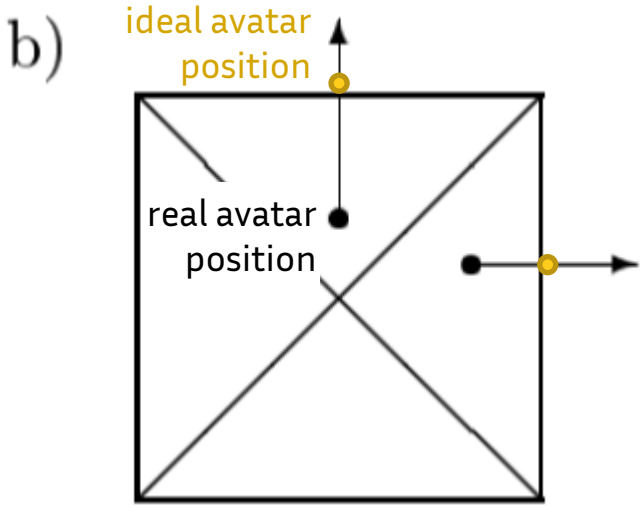


Richard & Cutkosky
Okamura

Issues with more complex objects

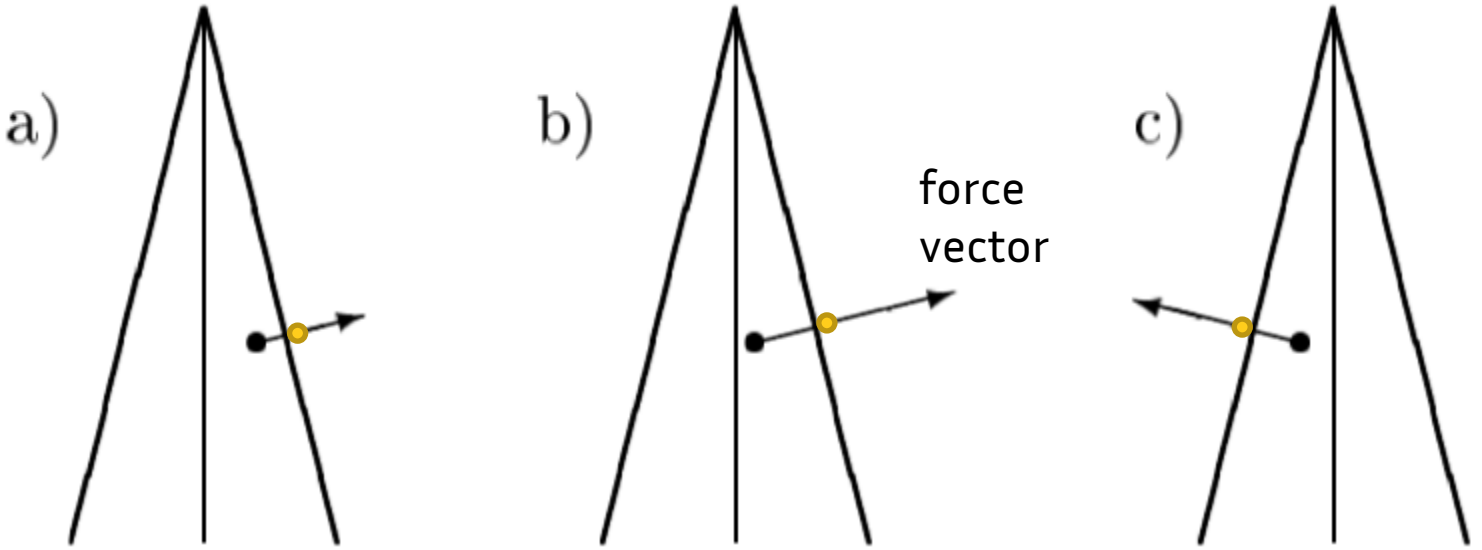


Two possible paths to reach the same location in a square; without a history we do not know which path was taken by the user.



Vector field method: subdivide the square's area and assume that the user entered from the closest edge. The force vectors are normal to the edge and proportional to distance penetrated.

Issues with more complex objects

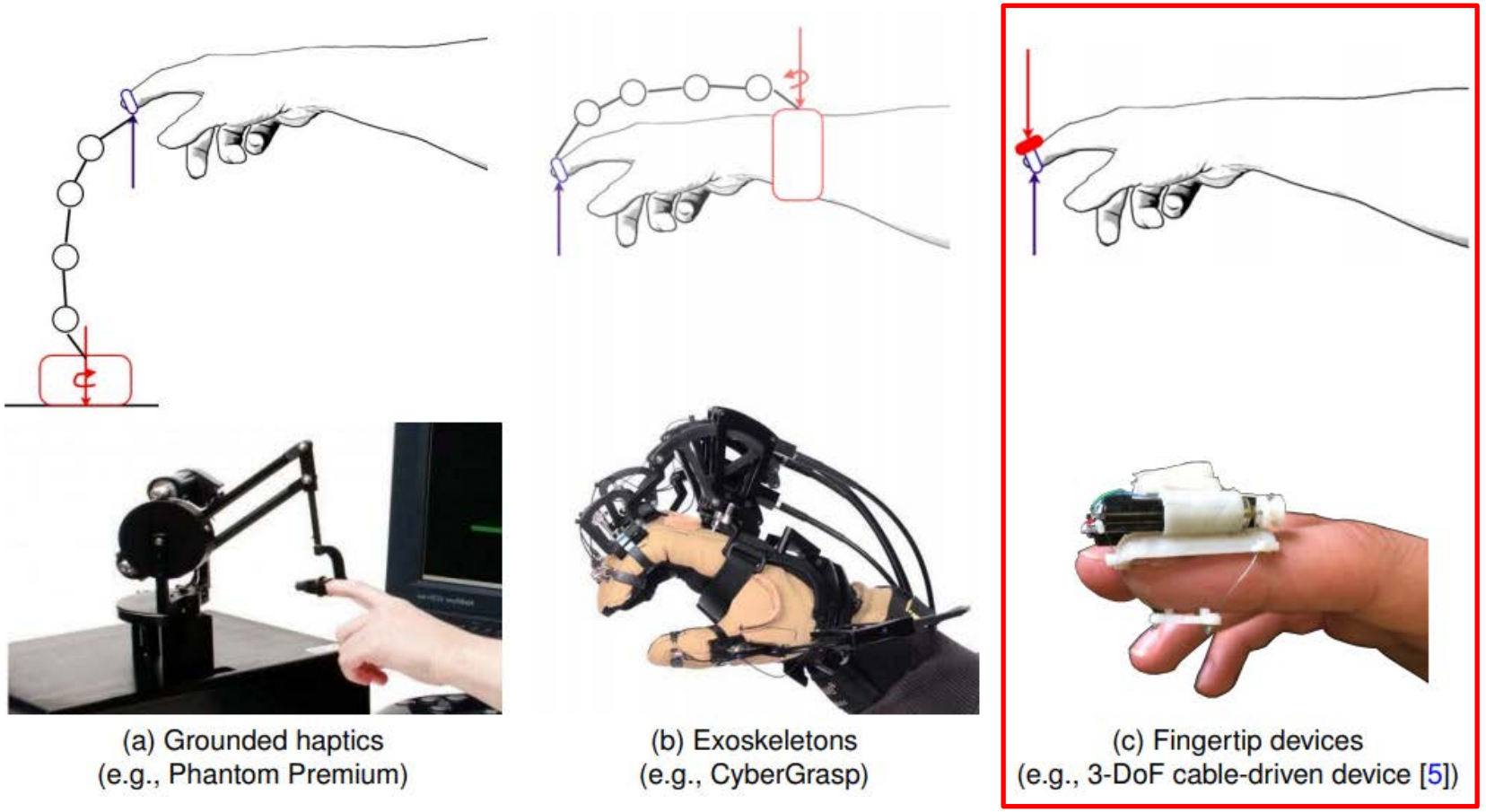


Push through of thin objects. a) user touches surface and feels a small force, b) as he/she pushes harder he/she penetrates deeper into the object, until c) he/she passes more than halfway through the object where the force vector changes direction and pushes him/her out the other side.

Today's Outline

- The haptic loop
- God-object/Proxy model
- Examples of rendering of a wall and a sphere
- **Data-driven rendering**
- Personalization of haptic rendering techniques

From grounded to wearable haptics



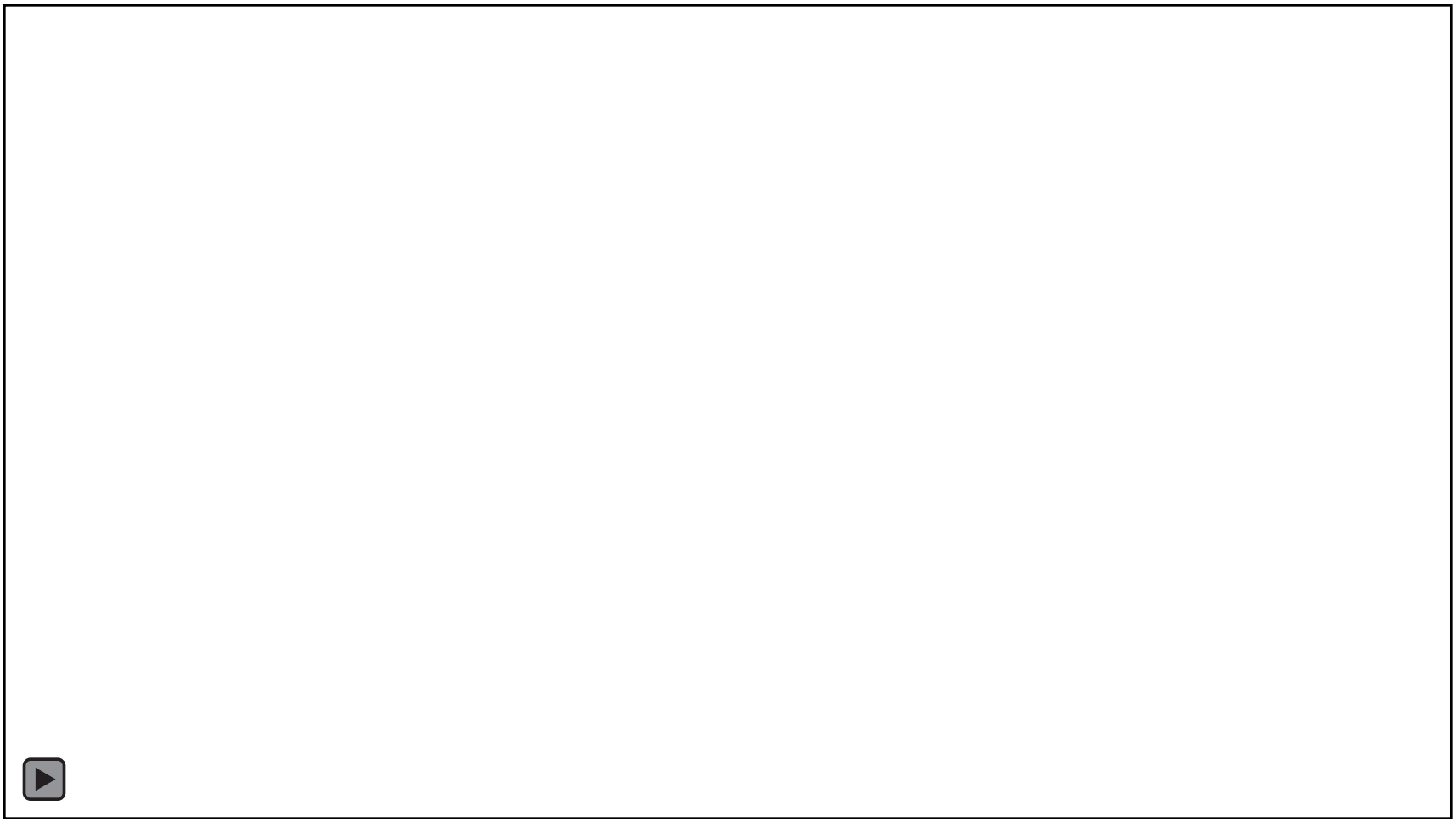
(a) Grounded haptics
(e.g., Phantom Premium)

(b) Exoskeletons
(e.g., CyberGrasp)

(c) Fingertip devices
(e.g., 3-DoF cable-driven device [5])

C. Pacchierotti, S. Sinclair, M. Solazzi, A. Frisoli, V. Hayward, D. Prattichizzo. “Wearable haptic systems for the fingertip and the hand: taxonomy, review, and perspectives” IEEE Transactions on Haptics, 10(4):580-600, 2017.

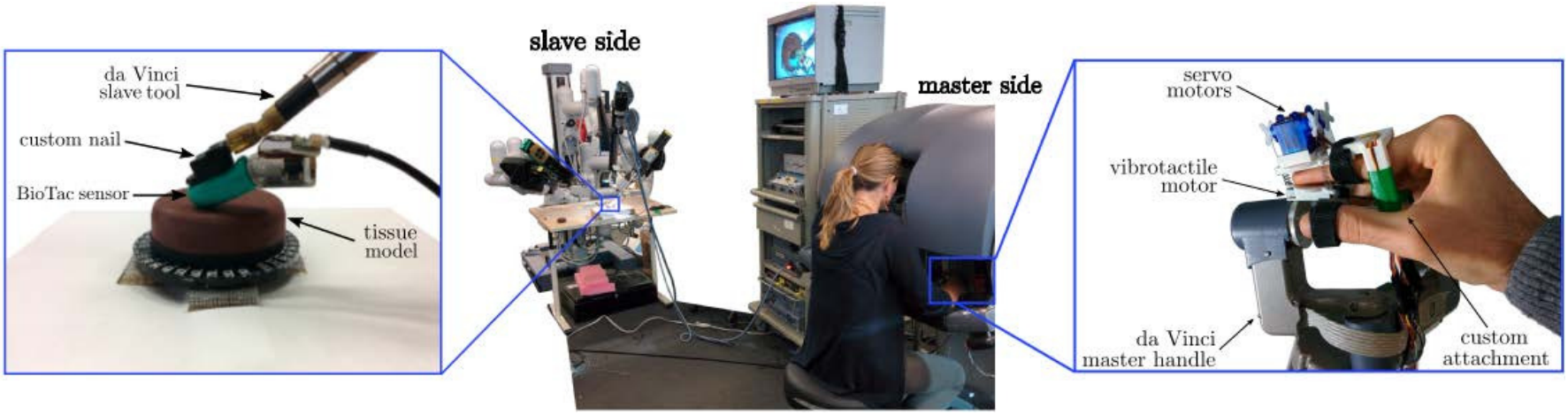
Cutaneous feedback device



Let's try to imagine an interesting application!

- Interaction in Virtual Reality
- Gaming
- E-commerce
- Medical diagnosis
- Surgical robotics

Palpation using the da Vinci Surgical System

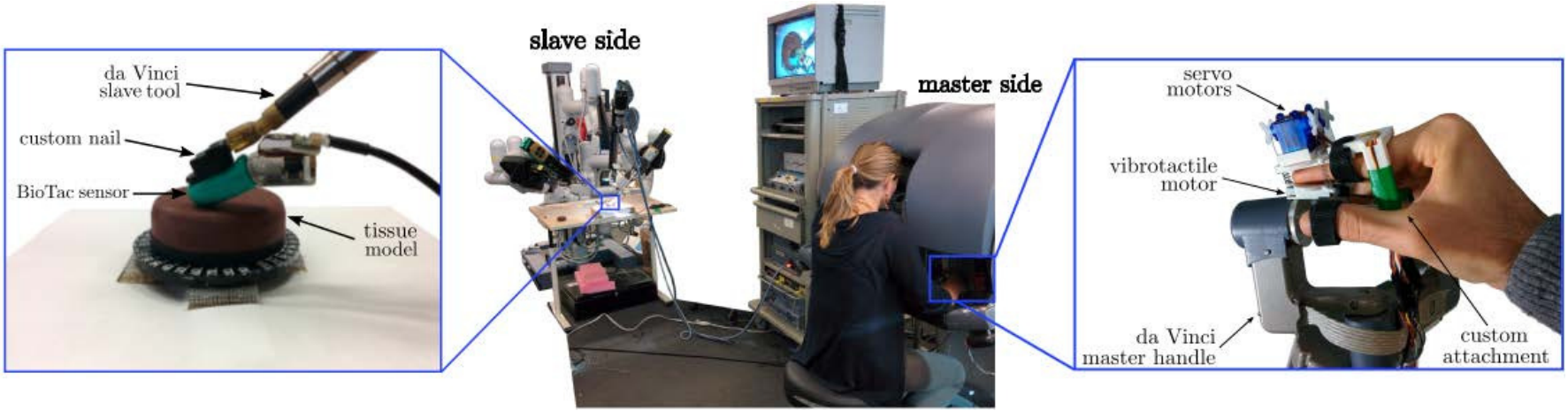


The haptic system is composed of a

(1) BioTac tactile sensor, in charge of registering contact forces and vibrations at the operating table, and

(2) a cutaneous feedback device, in charge of applying contact forces and vibrations to the surgeon.

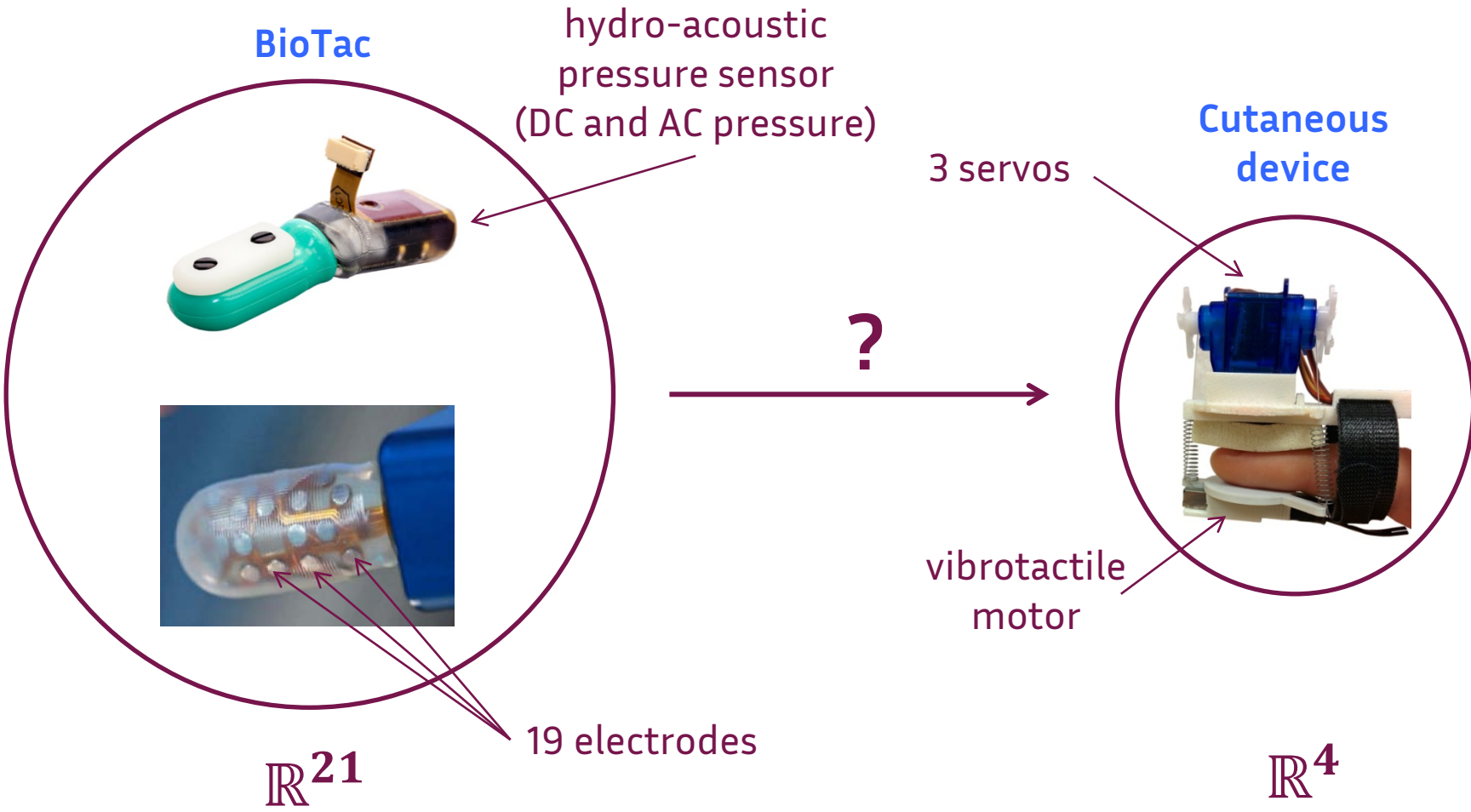
Palpation using the da Vinci Surgical System



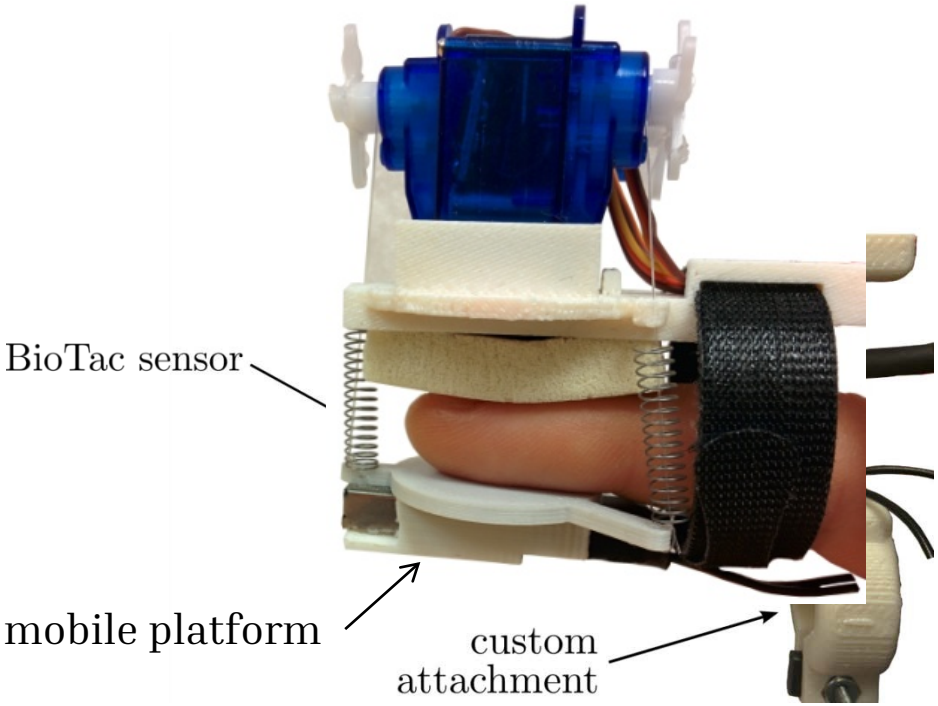
Contact deformations and vibrations sensed by the BioTac are directly mapped to input commands for the cutaneous device's motors using a model-free data-driven algorithm.

How?

Mapping tactile sensations from the BioTac to the cutaneous device



Data collection (servo motors)



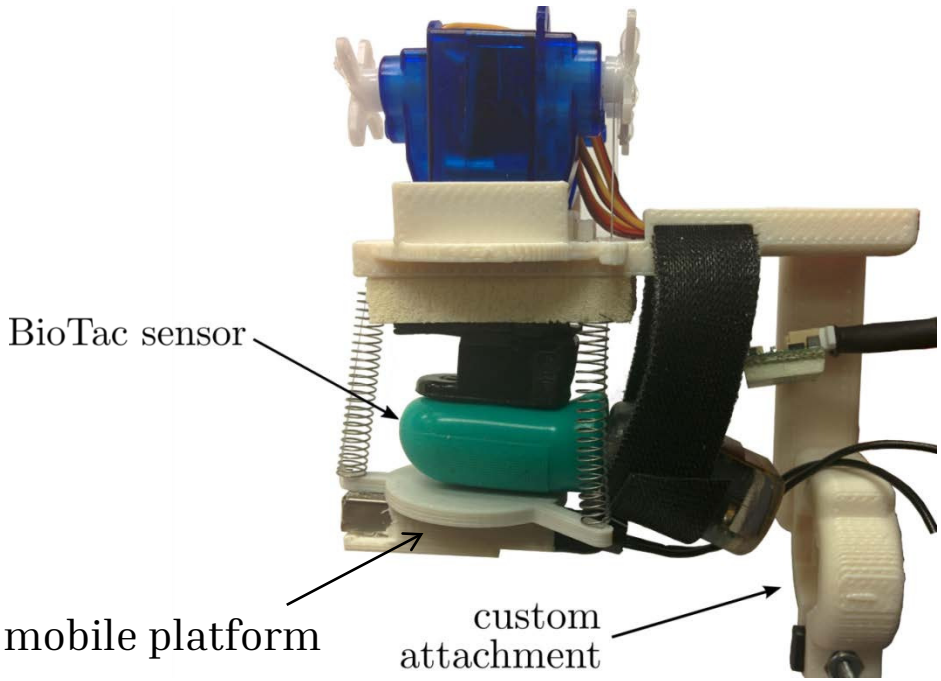
Platform configuration (input for the servo motors)

↓

Tactile sensation on the BioTac (19 electrodes + DC pressure)

We can define $\mu_d(\cdot)$, that maps a given BioTac sensation to the corresponding platform configuration.

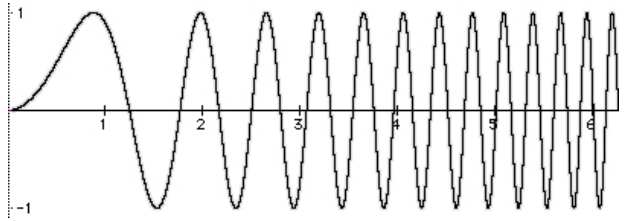
Data collection (vibrotactile motor)



Platform configuration (input for the servo motors)

↓

Transfer function between the vibrations sensed by the BioTac sensor (AC pressure) and the ones played by the vibrotactile motor



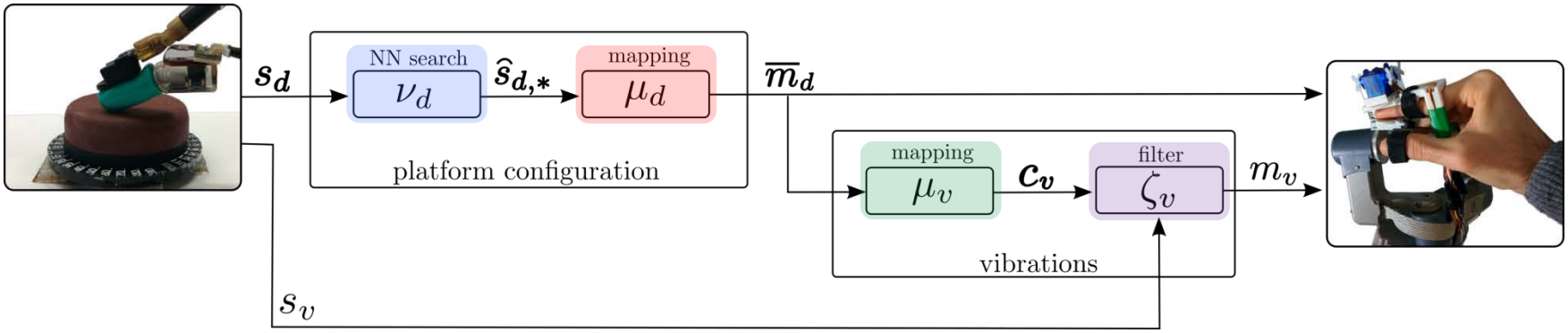
2-seconds-long sweep sine

We can define $\mu_V(\cdot)$, that maps a given platform configuration to the corresponding transfer function coefficients.

Mapping (simplified)

Search for the closest sensations experienced by the BioTac during the data collection

Map those sensations to the corresponding platform configuration

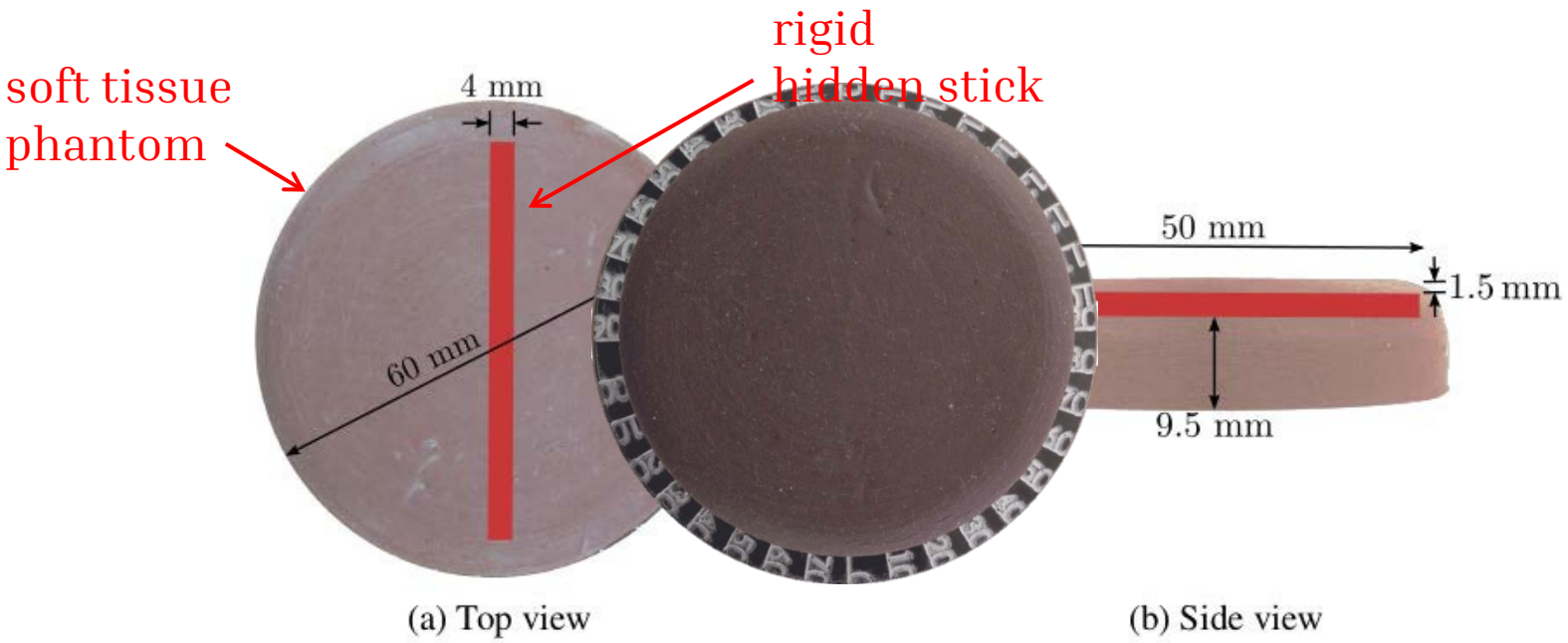


Map that platform configuration to the corresponding filter coefficients

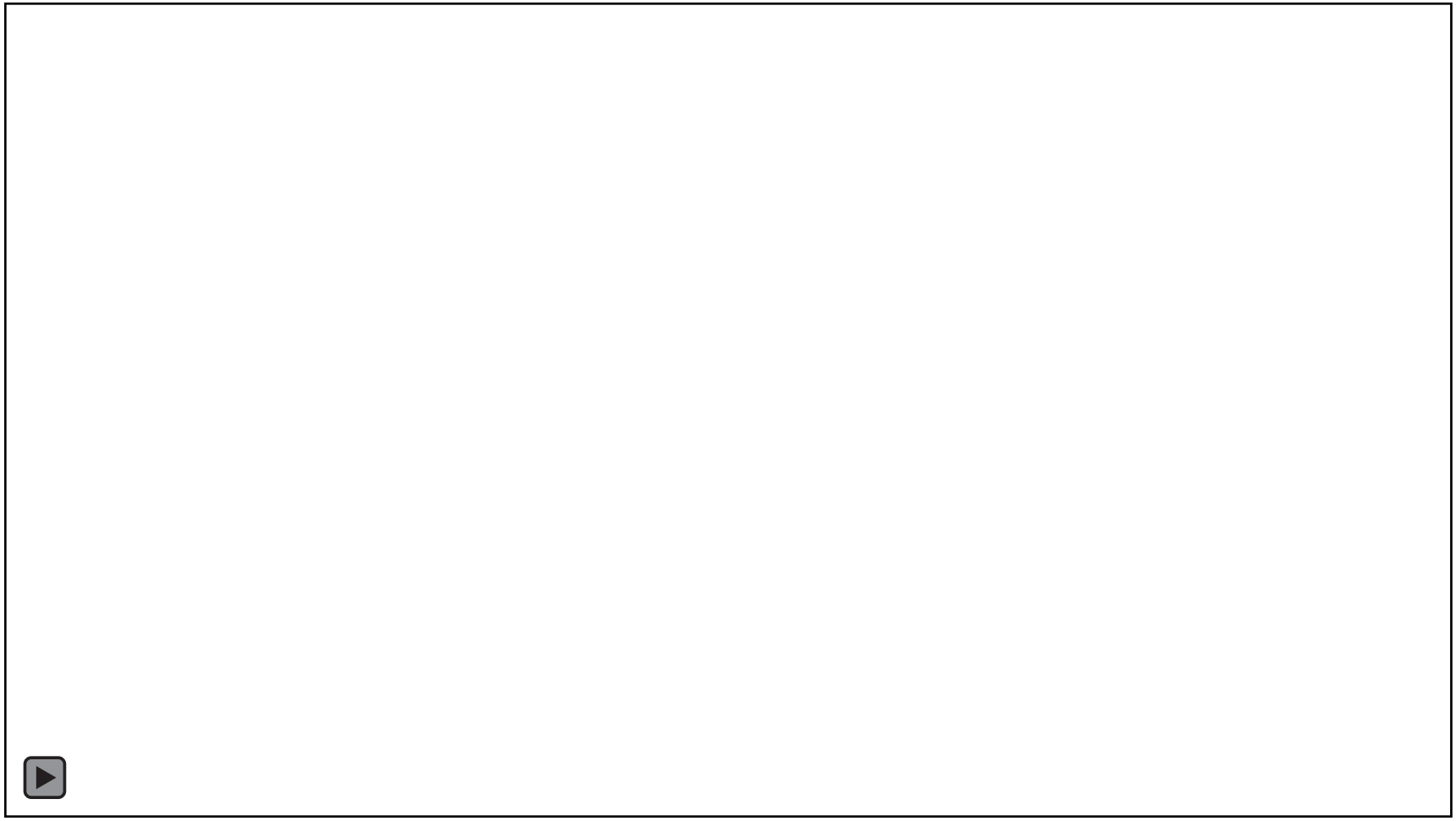
Filter AC pressure signal

Experimental evaluation

The remote environment is composed of a tissue phantom heart model. A plastic stick is embedded into the tissue model at 1.5 mm from the surface, and it is not visible from the outside. The plastic stick simulates the presence of a calcified artery.



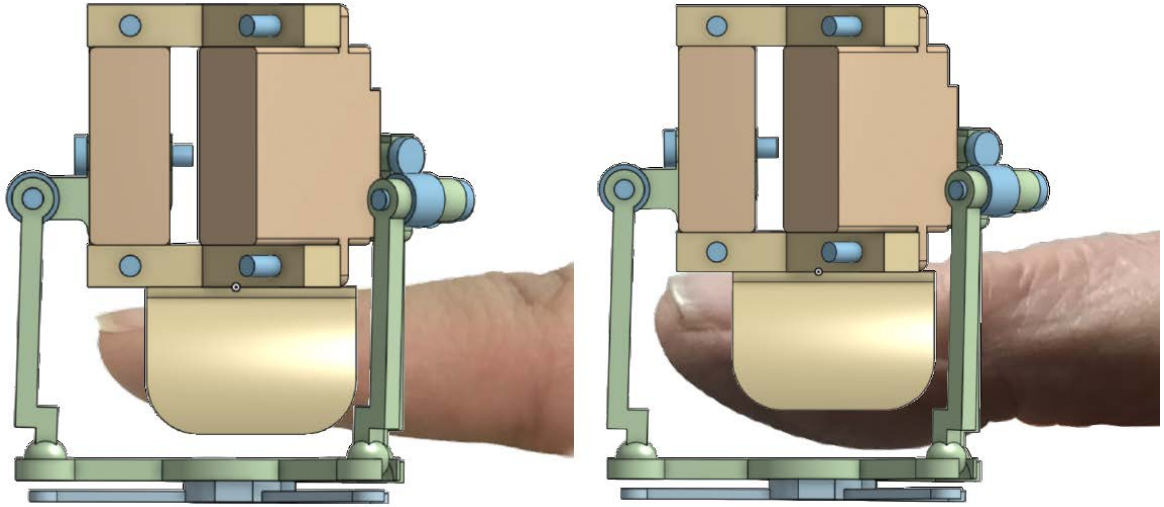
Experiment



Today's Outline

- The haptic loop
- God-object/Proxy model
- Examples of rendering of a wall and a sphere
- Data-driven rendering
- **Personalization of haptic rendering techniques**

Personalizing haptic interfaces (hardware)



(a) Small fingertip wearing the device.

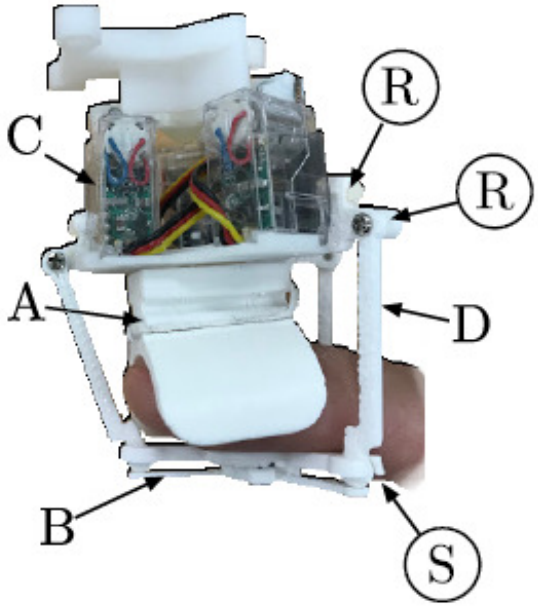
(b) Large fingertip wearing the device.

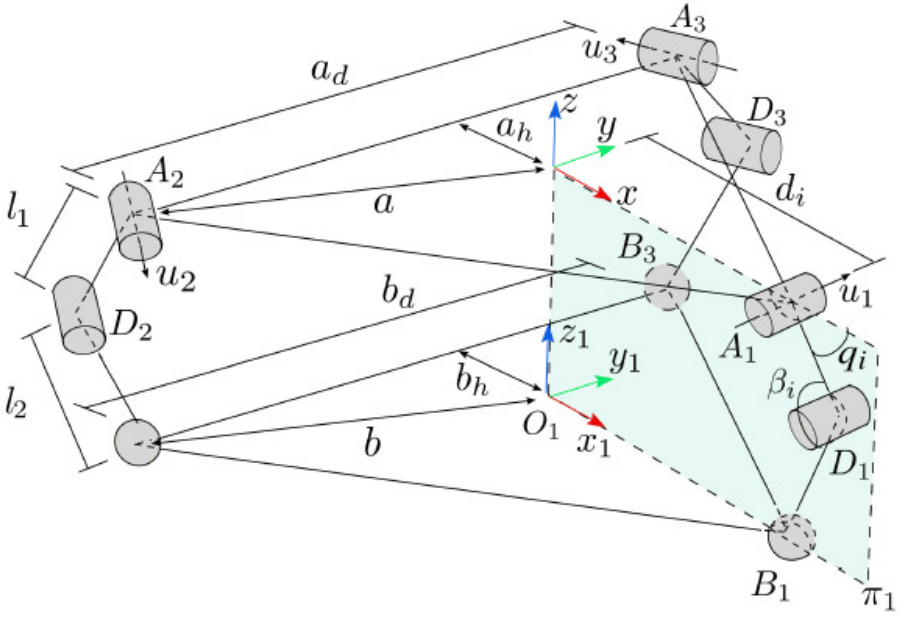
Representative problem: **the same fingertip haptic device will elicit different sensations on fingertips having different size and shape.**

Personalized haptics optimizes the device design for a target fingertip, so as to to always elicit the desired haptic sensation.

Personalizing haptic interfaces (hardware)

Let's consider the example of this fingertip haptic device.



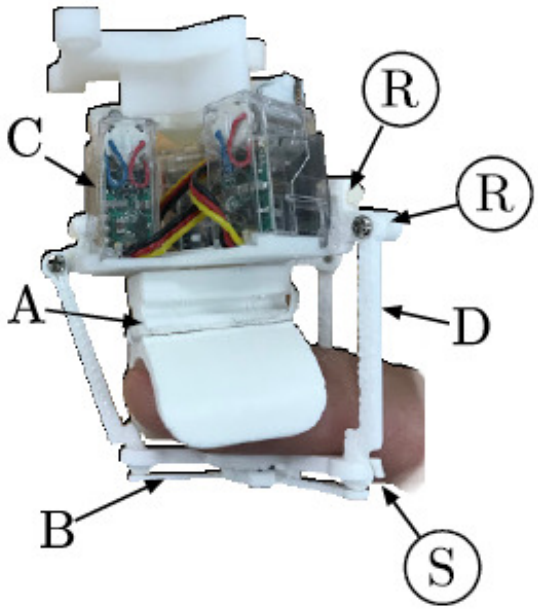


It is composed of a static upper body (A) and a mobile platform (B): the body is located above the nail, supporting three servo motors (C), while the mobile platform contacts the finger pulp. Three legs (D) connect the mobile platform with the static body.

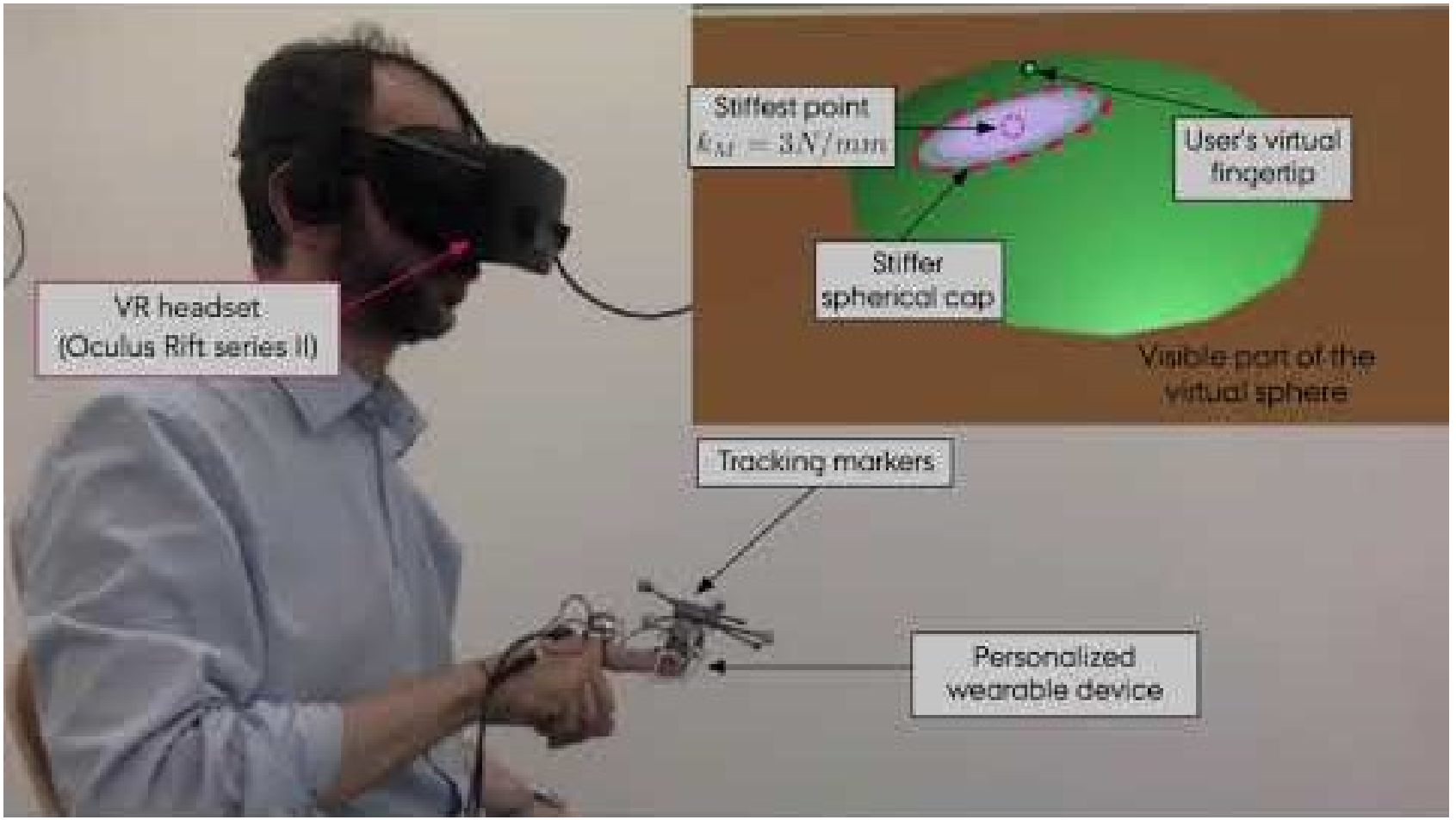
Personalizing haptic interfaces (hardware)

Malvezzi et al. (2020) came up with a system to personalize the design of this wearable tactile device for a given fingertip, considering three sub-problems in sequence:

- mobile platform (end-effector) dimensions are defined on the basis of the user's finger dimensions and the device target workspace, i.e., the surface of the finger that will be involved in the cutaneous stimulation.
- static platform dimensions are consequently defined so that, during the cutaneous stimuli application, only the mobile platform interacts with the fingertip and no undesired contacts with the legs occur.
- articulated legs lengths are defined to avoid kinematic singularities in any of the device operative configurations.

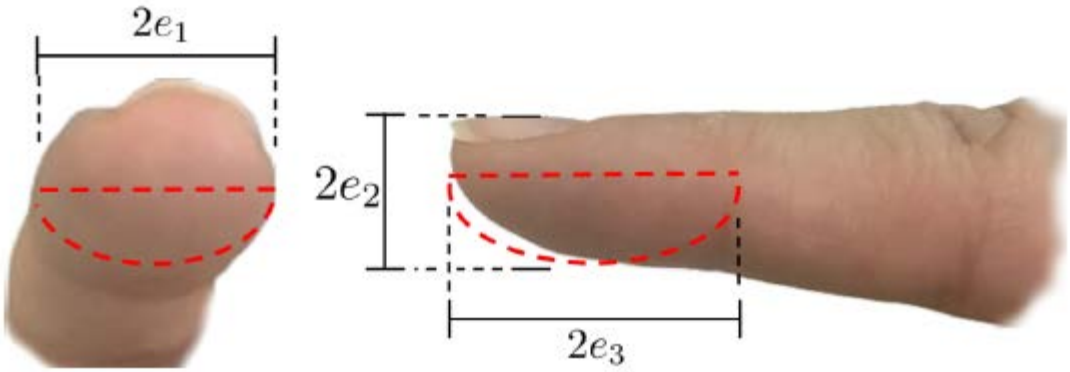


Personalizing haptic interfaces (hardware)



Personalizing haptic interfaces (hardware)

Let's try!



My right index finger has

e1 = 9.5 mm; e2 = 8.5 mm; e3 = 18 mm.

Personalizing haptic interfaces (hardware)

Demonstration

Personalized haptics: the rendering approach

We can personalized the behavior of wearable haptic interfaces to fit ones specific characteristics. We can modify the design of the haptic interface or adjust the rendering technique.

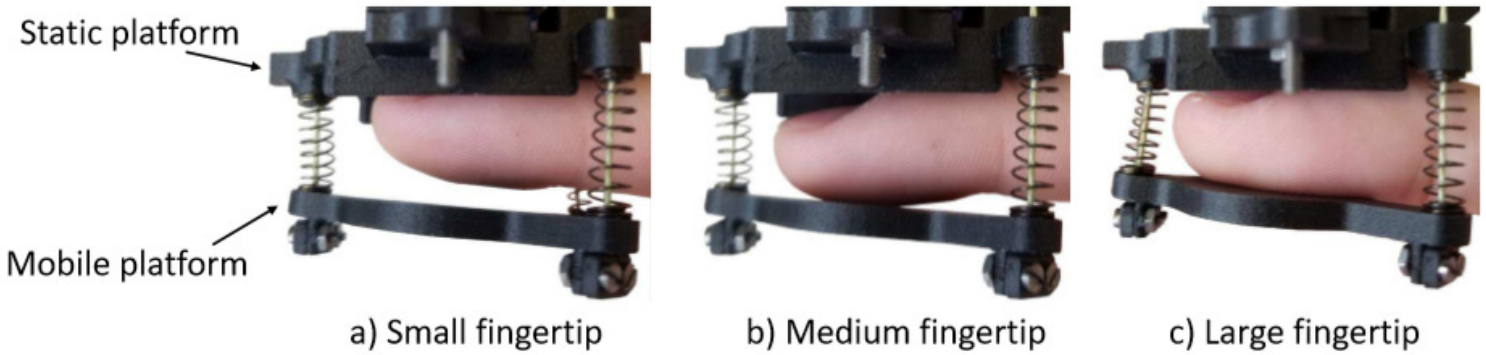
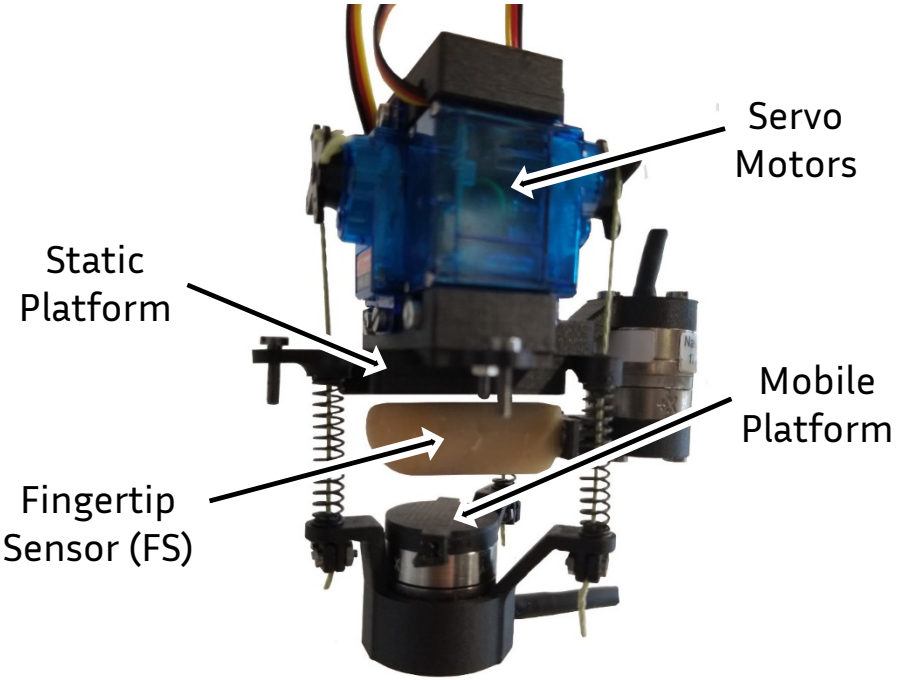


Fig. 1. Problem: the same end-effector configuration elicits different sensations for fingertips of different size, including users with (a) small, (b) medium, and (c) large fingertips. Our personalization methods aim to account for the user's fingertip shape so as to always elicit the desired haptic sensation.

Personalized haptics: the rendering approach

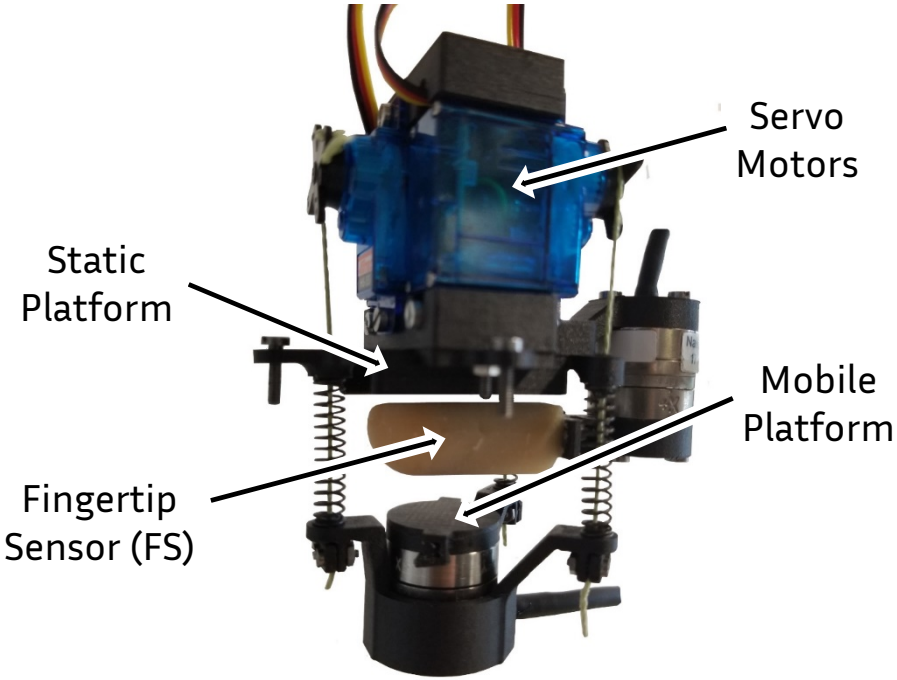
Let's recall the data-driven technique in this context.



First, we place the fingertip sensor (FS) in the device.

Personalized haptics: the rendering approach

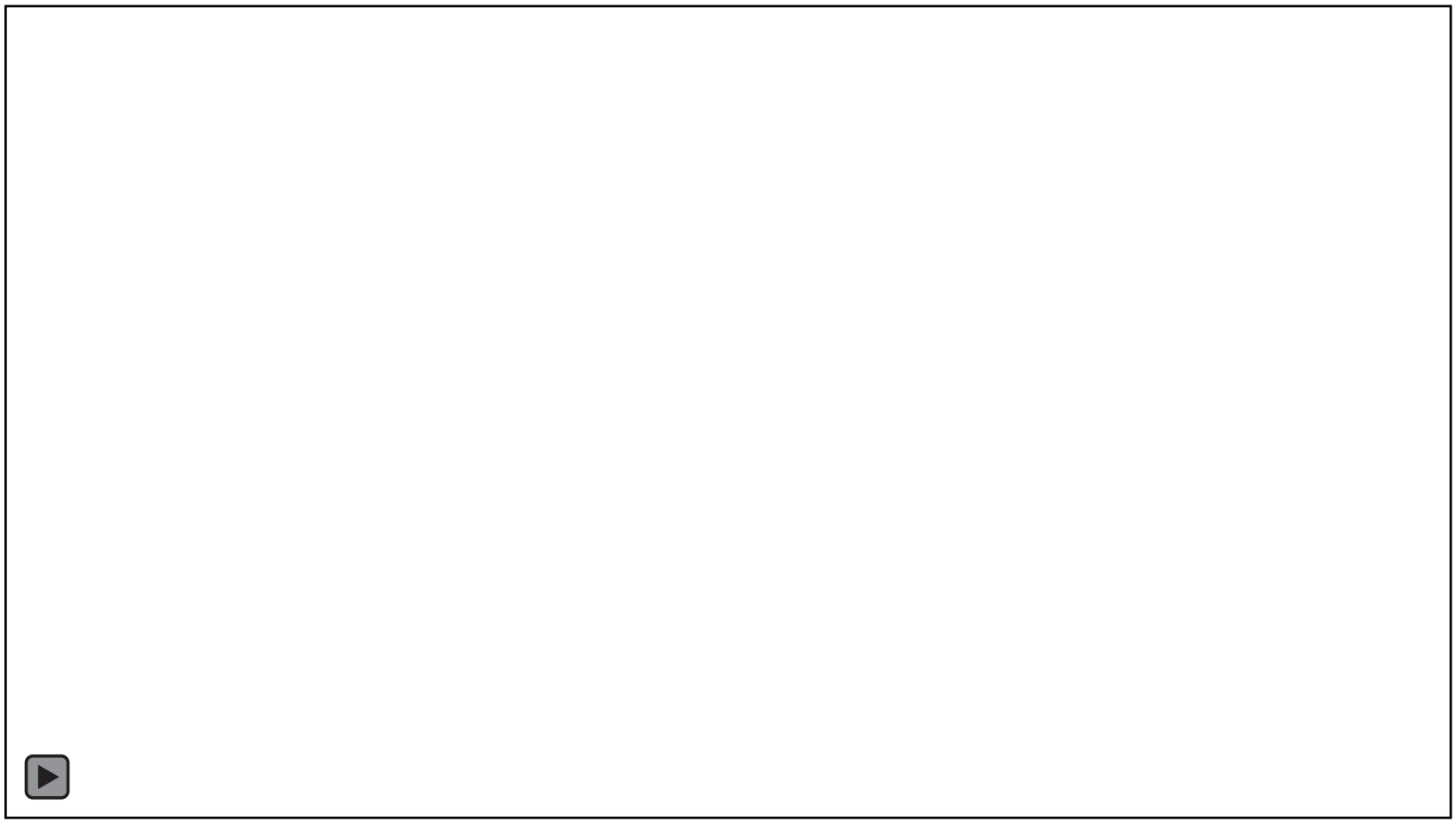
Let's recall the data-driven technique in this context.



We then move the end-effector of the device to a wide range of configurations, and the effect of each of these configurations is registered on the FS.

Personalized haptics: the rendering approach

Let's recall the data-driven technique in this context.



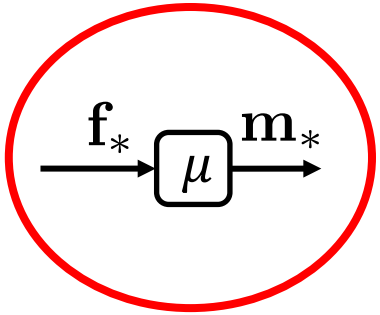
Personalized haptics: the rendering approach

Let's recall the data-driven technique in this context.

With this data, we define the mapping

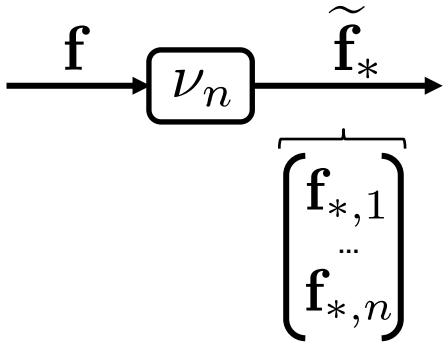
$$\mu(\mathbf{f}_*) = \mathbf{m}_*,$$

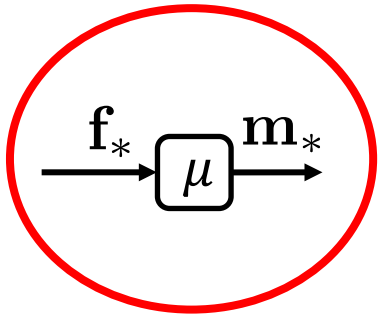
where $\mathbf{m}_*$ is a vector of motor inputs
and $\mathbf{f}_*$ is the corresponding FS output.



Personalized haptics: the rendering approach

Let's recall the data-driven technique in this context.

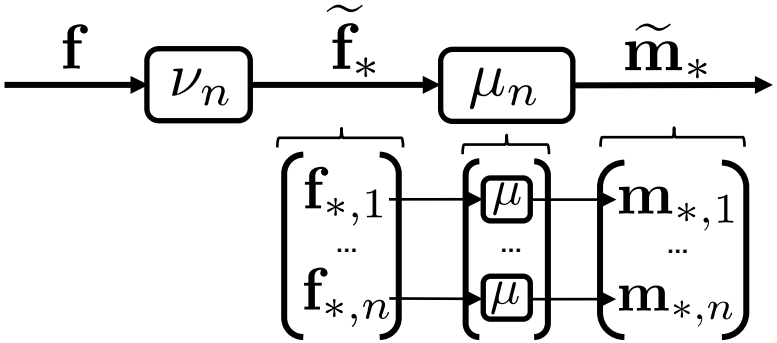


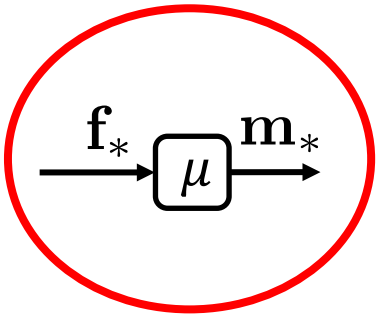


To render a new remote sensation, $\mathbf{f}$…
we first find the n most similar sensor outputs
measured during data collection, $\widetilde{\mathbf{f}}_*$.

Personalized haptics: the rendering approach

Let's recall the data-driven technique in this context.

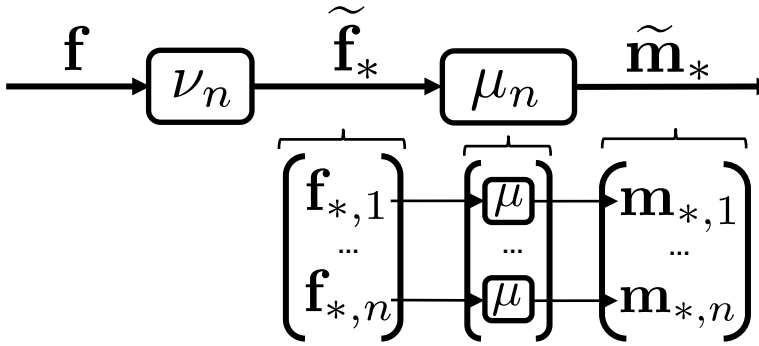


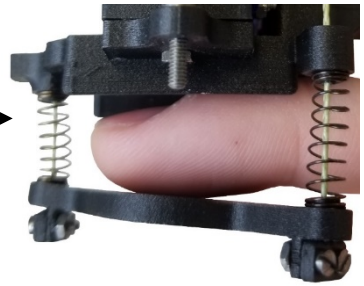


We then convert each element in $\widetilde{\mathbf{f}}_*$ back to its corresponding motor inputs, using the lookup table.

Personalized haptics: the rendering approach

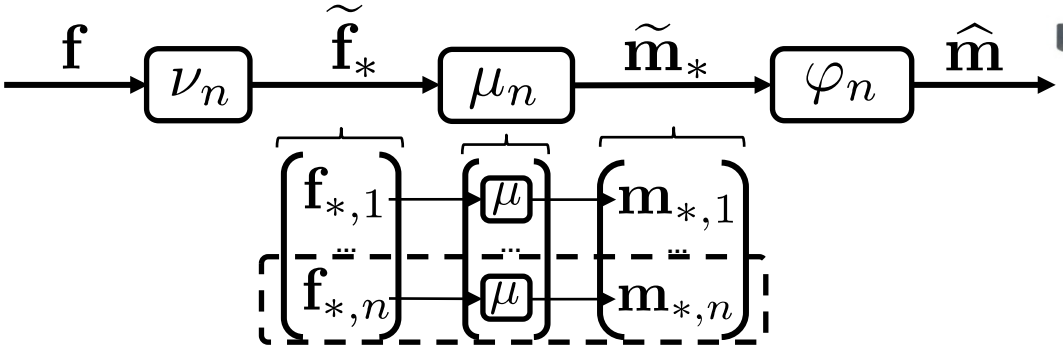
Let's recall the data-driven technique in this context.

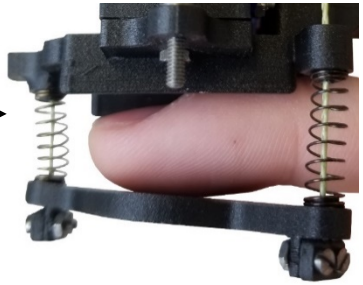




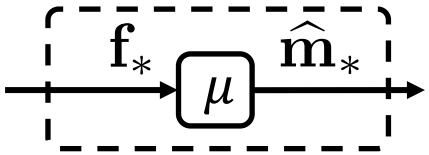
Finally, the n motor inputs, $\widetilde{\mathbf{m}}_*$, are averaged into one motor input, $\widehat{\mathbf{m}}$, weighting each input according to the distance between the original sensation, $\mathbf{f}$, and the one elicited by $\mathbf{m}_{*,i}$ during data collection, $\mathbf{f}_{*,i}$.

Personalized haptics: the rendering approach



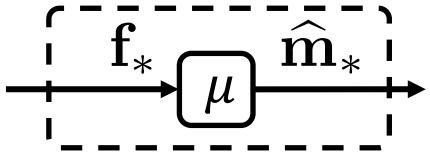


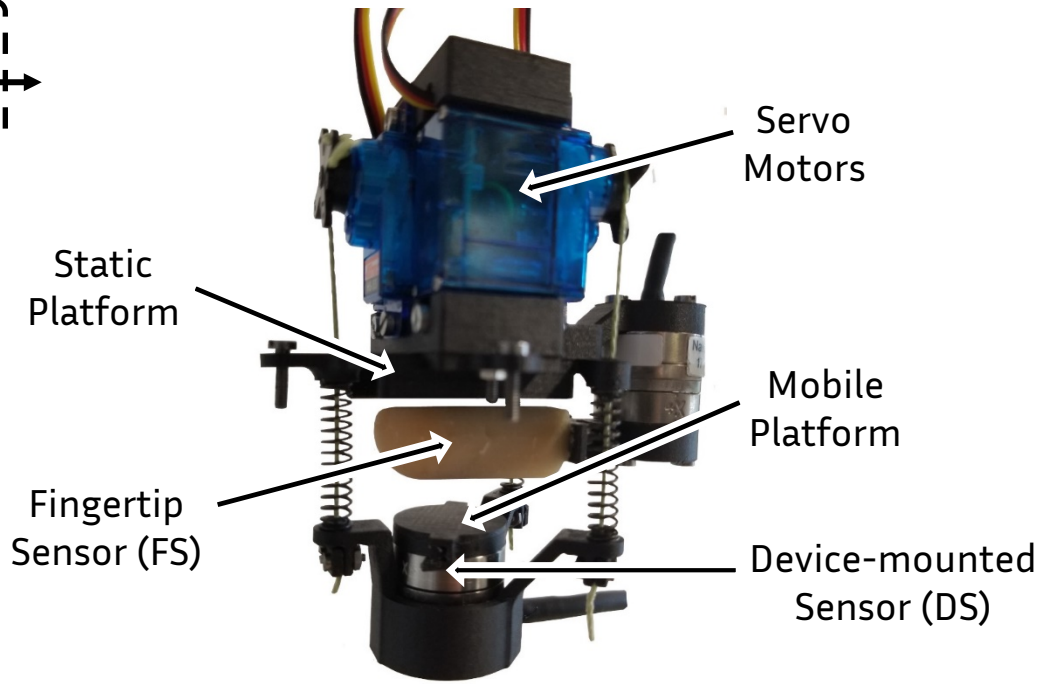
Both of our personalization methods adjust the vector of motor inputs, $\mathbf{m}_*$, associated with a sensation, $\mathbf{f}_*$, in this FS-centered calibration.



Personalized haptics: the rendering approach

Data-driven personalization:

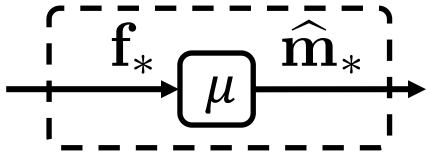




First, we adjust our FS-centered calibration procedure to include data from a device-mounted sensor (DS).

Personalized haptics: the rendering approach

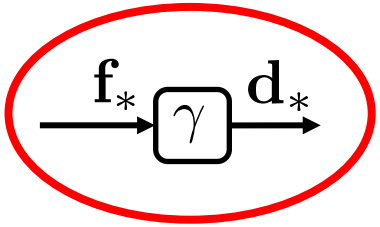
Data-driven personalization:



With this data, we define an additional mapping

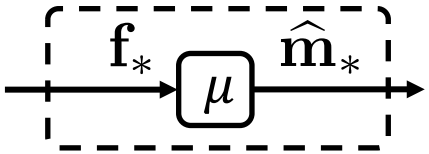
$$\gamma(\mathbf{f}_*) = \mathbf{d}_*,$$

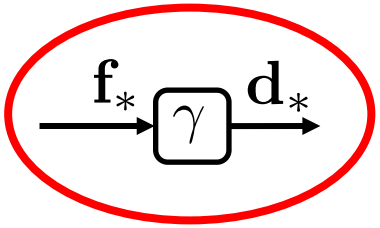
where $\mathbf{f}_*$ is an FS output and $\mathbf{d}_*$ is the corresponding DS output.

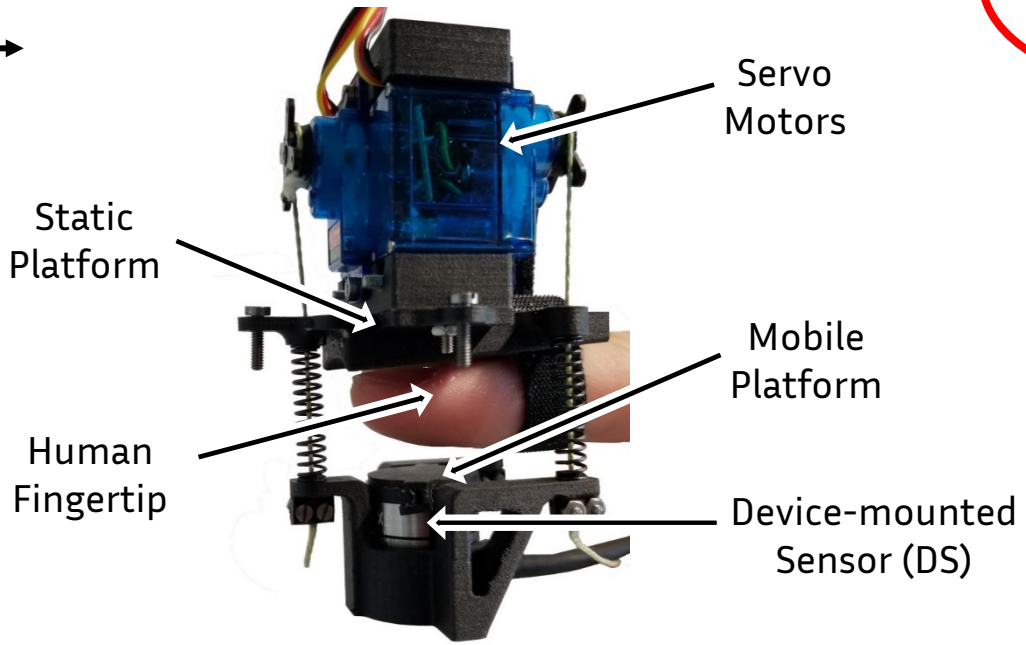


Personalized haptics: the rendering approach

Data-driven personalization:



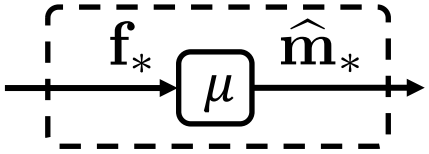


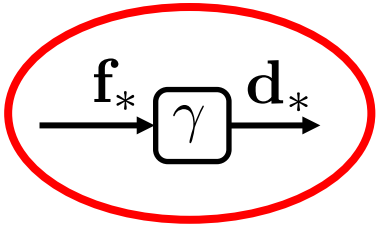


Then we have the human wear the device, and we create a user-centered lookup table.

Personalized haptics: the rendering approach

Data-driven personalization:

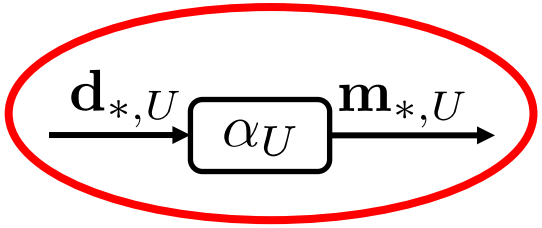




With this data, we define the mapping

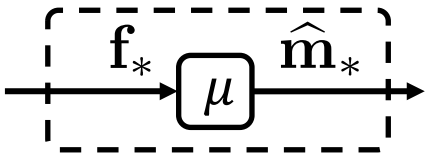
$$\alpha_U(\mathbf{d}_{*,U}) = \mathbf{m}_{*,U},$$

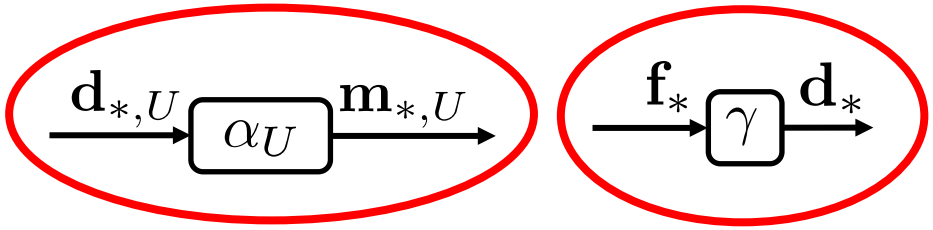
where $\mathbf{m}_{*,U}$ is a vector of motor inputs and $\mathbf{d}_{*,U}$ is the corresponding DS output, for a particular user, U.



Personalized haptics: the rendering approach

Data-driven personalization:

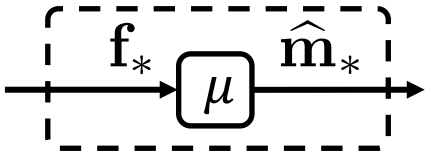


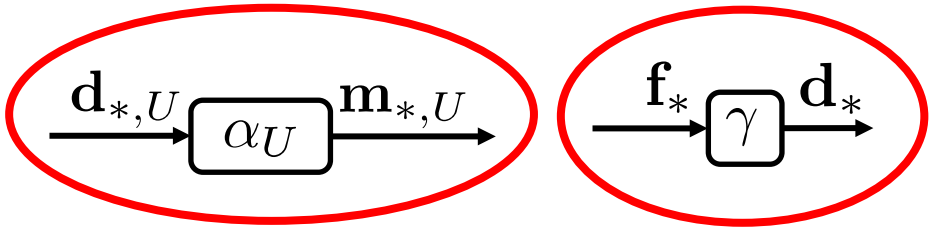


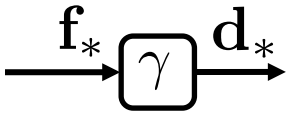
We can now use these functions to personalize the FS-centered calibration data.

Personalized haptics: the rendering approach

Data-driven personalization:



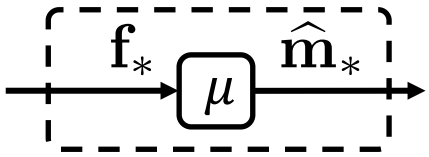


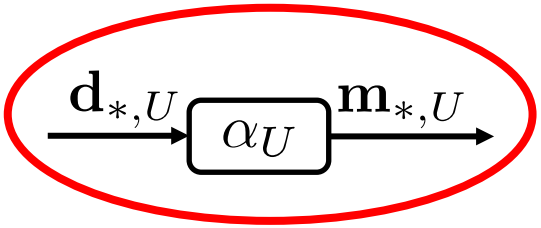


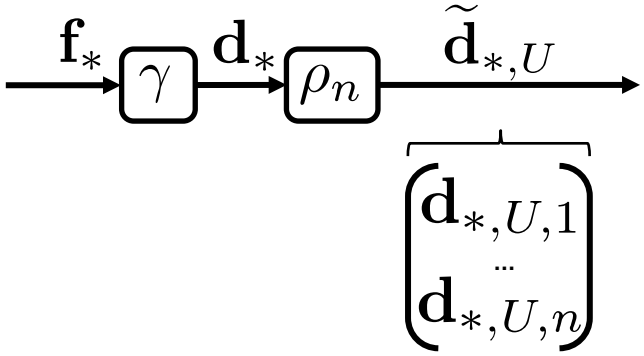
First, we take every FS sensation registered during the FS-centered calibration, $\mathbf{f}_*$, and find the corresponding DS sensation, $\mathbf{d}_*$.

Personalized haptics: the rendering approach

Data-driven personalization:



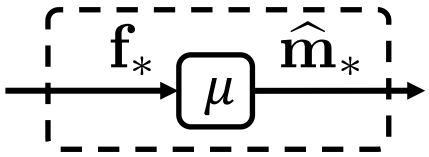


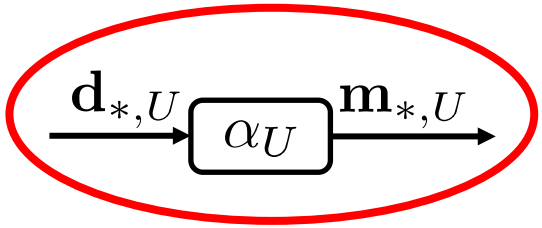


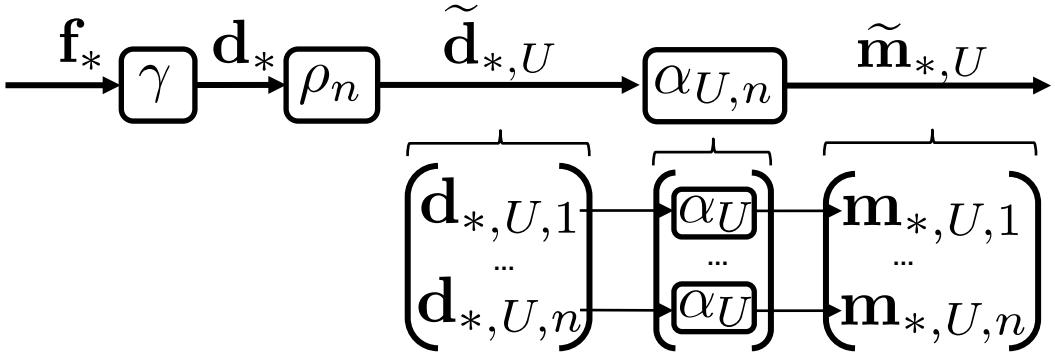
Then, we find the n nearest DS sensations observed during the user-centered calibration, $\widetilde{\mathbf{d}}_{*,U}$.

Personalized haptics: the rendering approach

Data-driven personalization:



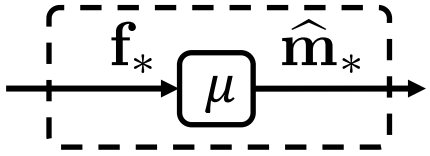


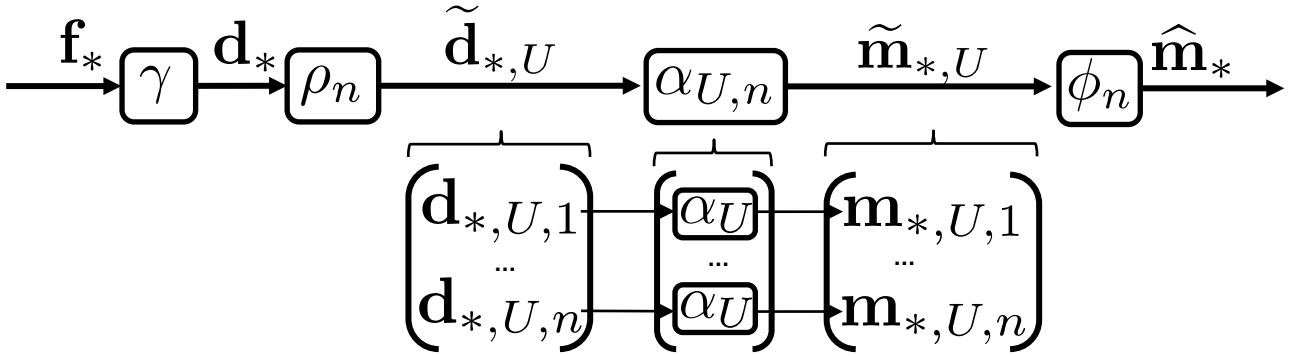


We then convert each element in $\widetilde{\mathbf{d}}_{*,U}$ back to its corresponding motor inputs, $\widetilde{\mathbf{m}}_{*,U}$, using the lookup table.

Personalized haptics: the rendering approach

Data-driven personalization:

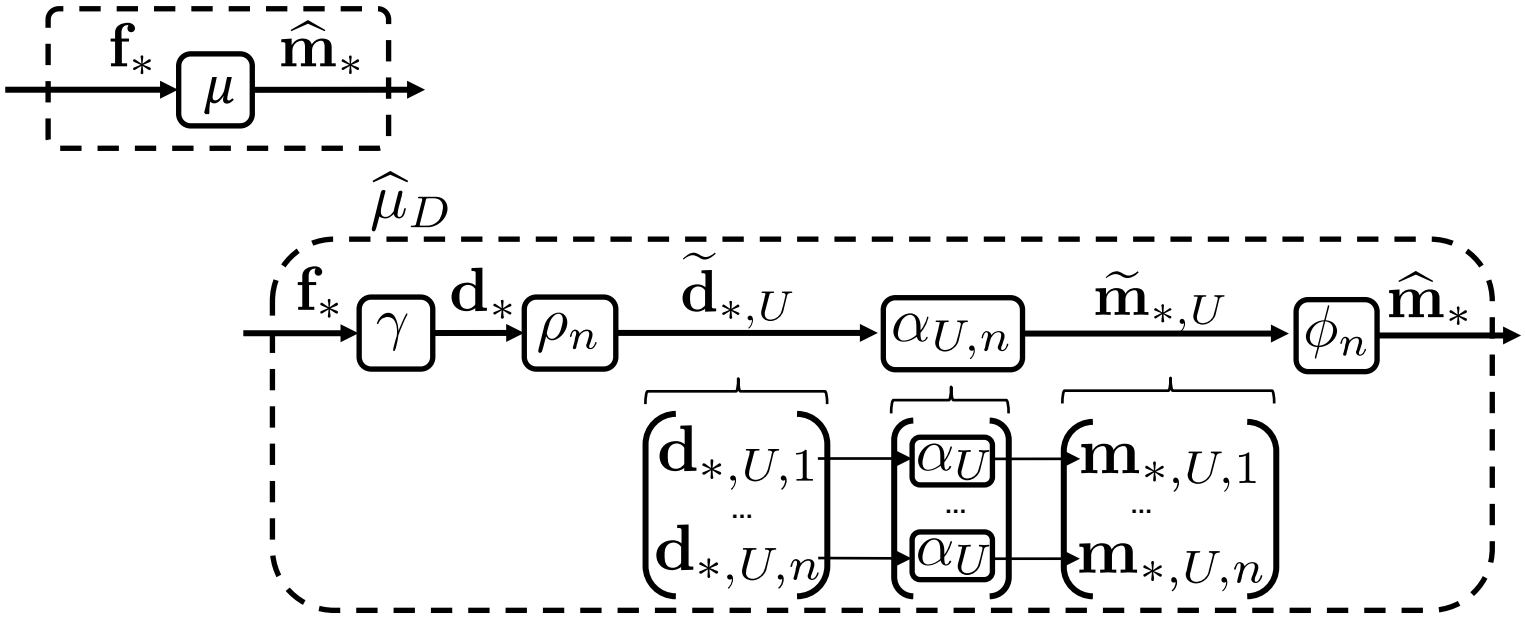




Finally, we find the weighted average of these motor inputs, $\widehat{\mathbf{m}}_*$.

Personalized haptics: the rendering approach

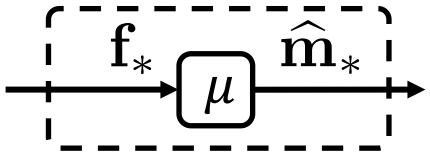
Data-driven personalization:



This procedure gives us the complete adjustment for our data-driven personalization approach.

Personalized haptics: the rendering approach

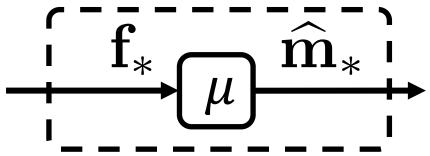
Data-driven personalization:



After calibrating our device for both the FS and the user, we can remove the DS and replace our end-effector with the original flat surface.

Personalized haptics: the rendering approach

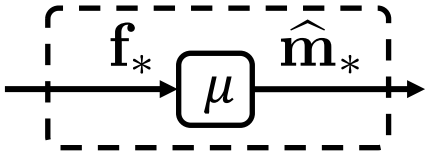
Geometric personalization:



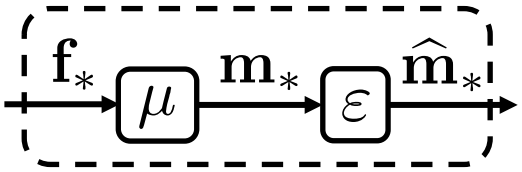
Now let's look at our geometric personalization approach.

Personalized haptics: the rendering approach

Geometric personalization:



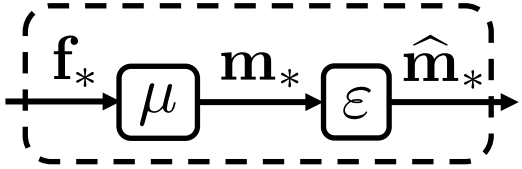
We define our geometric adjustment by

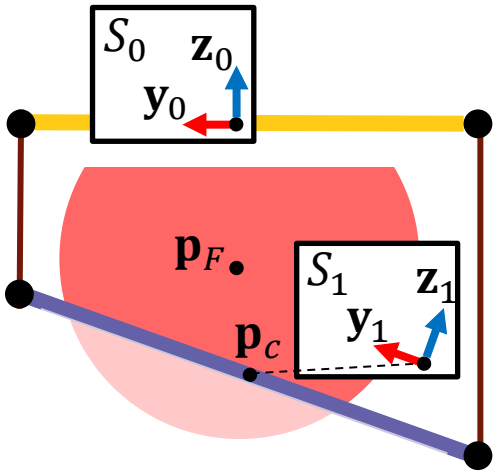


where ε adjusts each motor input registered during the FS-centered data collection based on the geometry of the target finger.

Personalized haptics: the rendering approach

Geometric personalization:

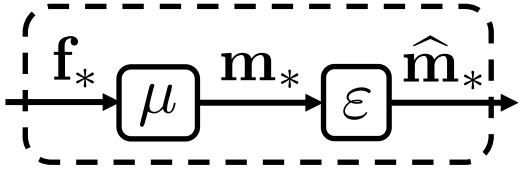


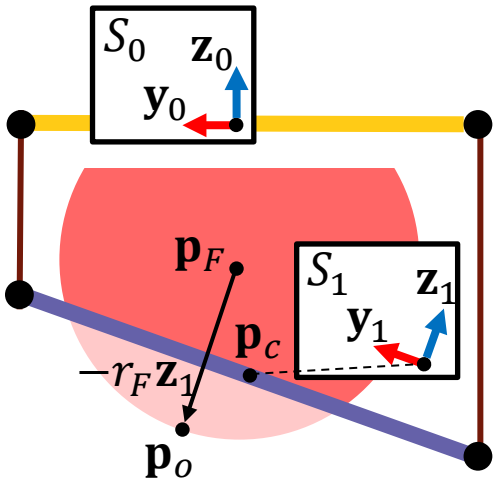


For each motor command $\mathbf{m}_*$ in the FS-centered calibration, we first find the corresponding mobile platform position and orientation, given by $\mathbf{p}_c$ and S_1, respectively.

Personalized haptics: the rendering approach

Geometric personalization:



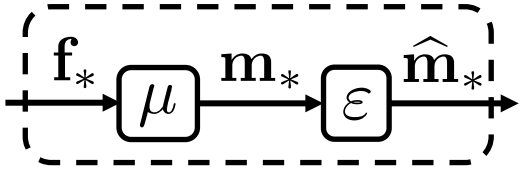


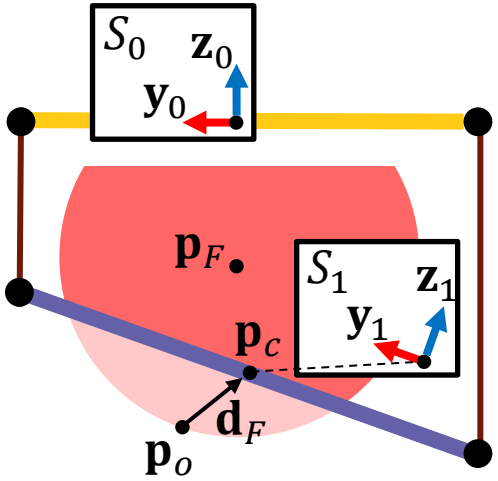
We then compute the point $\mathbf{p}_o$ at which the undeformed surface of the FS is parallel to our mobile platform, given by

$$\mathbf{p}_o = \mathbf{p}_F - r_F\mathbf{z}_1$$

Personalized haptics: the rendering approach

Geometric personalization:



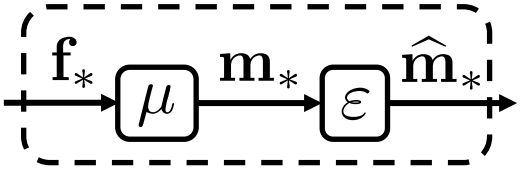


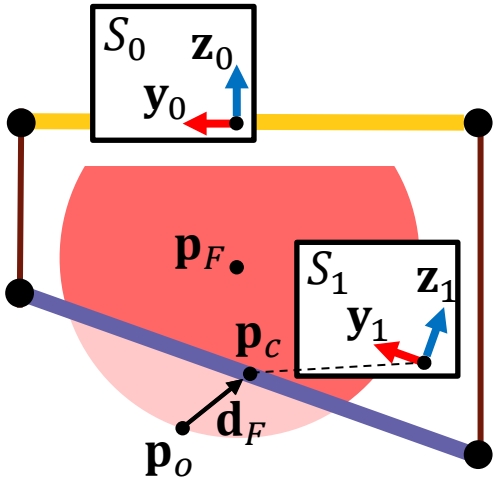
We are then able to estimate how much and in which direction the mobile platform deforms the sensor

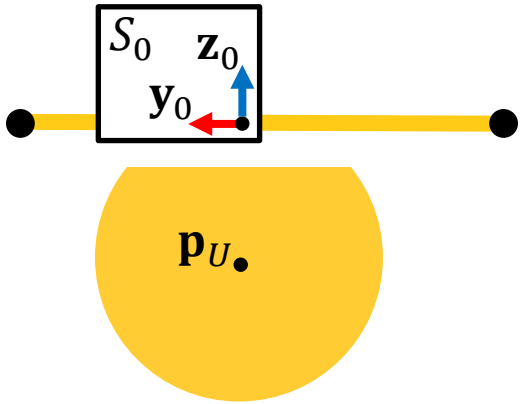
$$\mathbf{d}_F = \mathbf{p}_c - \mathbf{p}_o$$

Personalized haptics: the rendering approach

Geometric personalization:





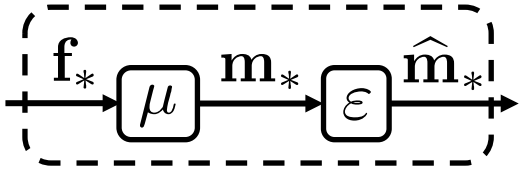


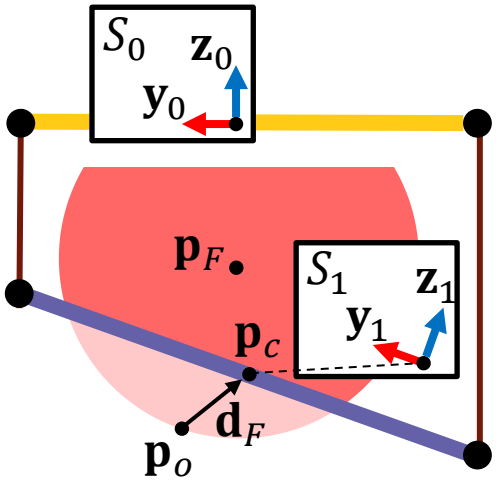
Now, we want to render to a second fingertip…
the same deformation.

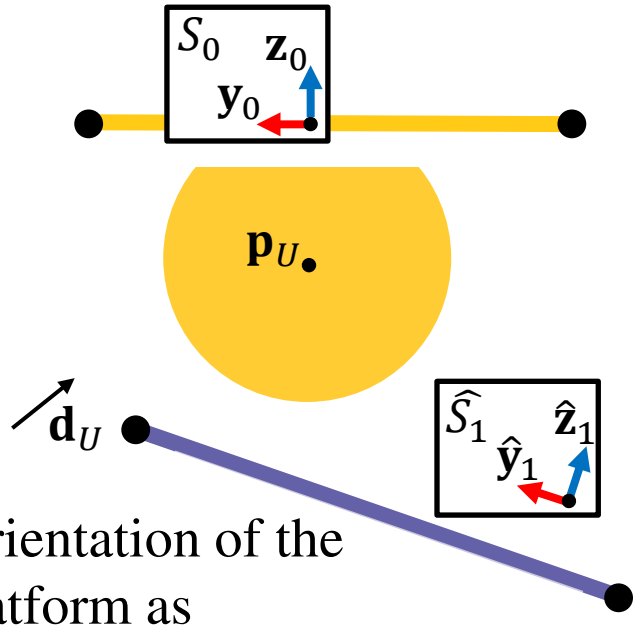
$$\mathbf{d}_U = \mathbf{d}_F$$

Personalized haptics: the rendering approach

Geometric personalization:





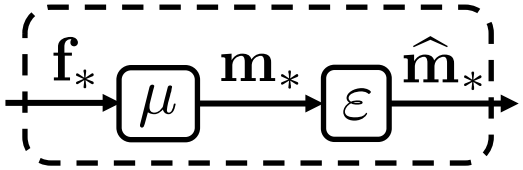


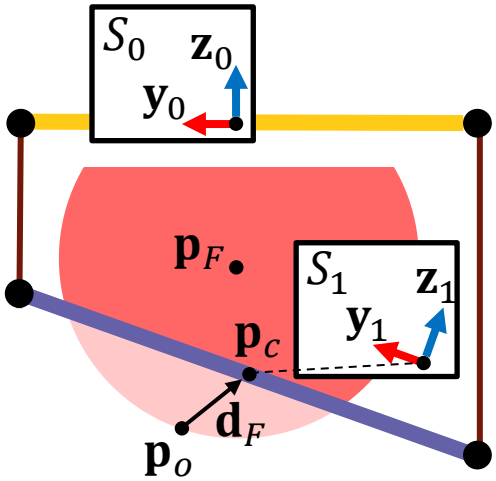
We estimate the orientation of the adjusted platform as

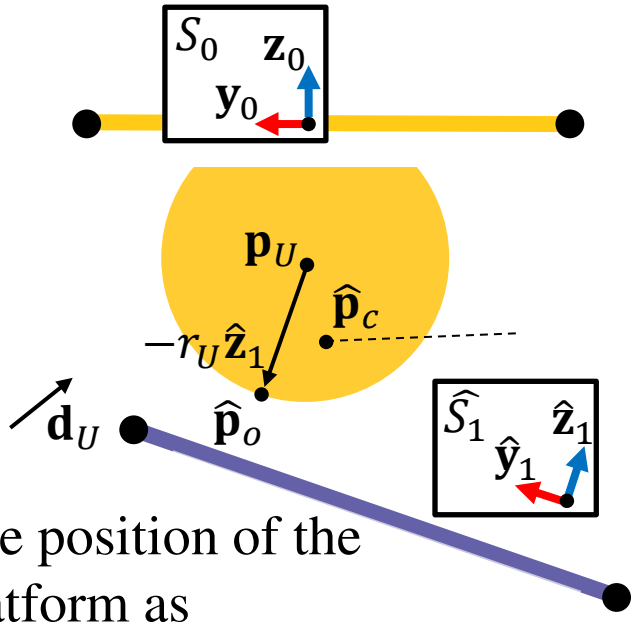
$$\widehat{S_1} = S_1$$

Personalized haptics: the rendering approach

Geometric personalization:





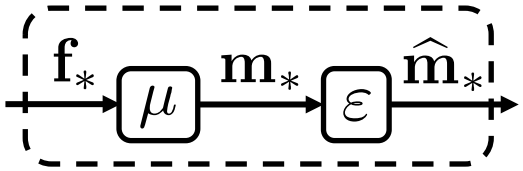


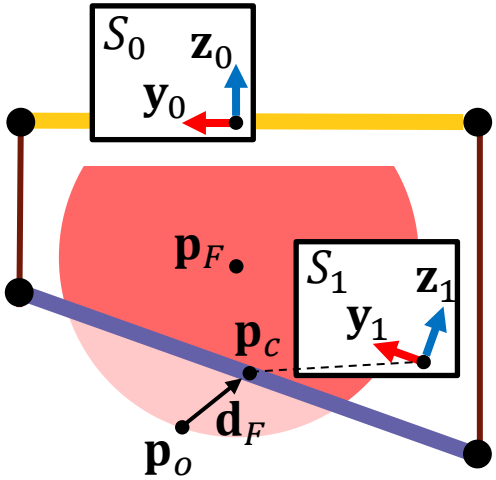
And we estimate the position of the adjusted platform as

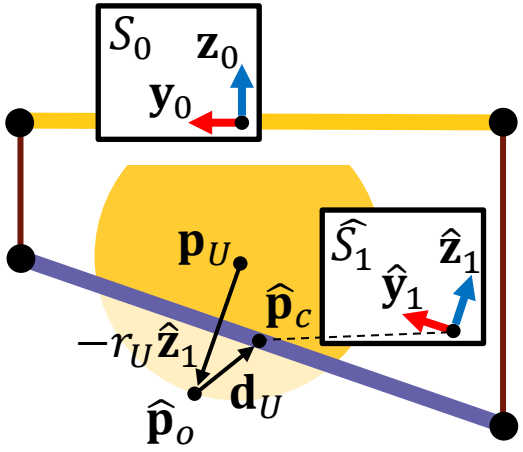
$$\hat{\mathbf{p}}_c = \mathbf{p}_U - r_U\hat{\mathbf{z}}_1 + \mathbf{d}_U$$

Personalized haptics: the rendering approach

Geometric personalization:







Finally, we convert our adjusted mobile platform orientation and position into adjusted motor inputs $\hat{\mathbf{m}}_*$.

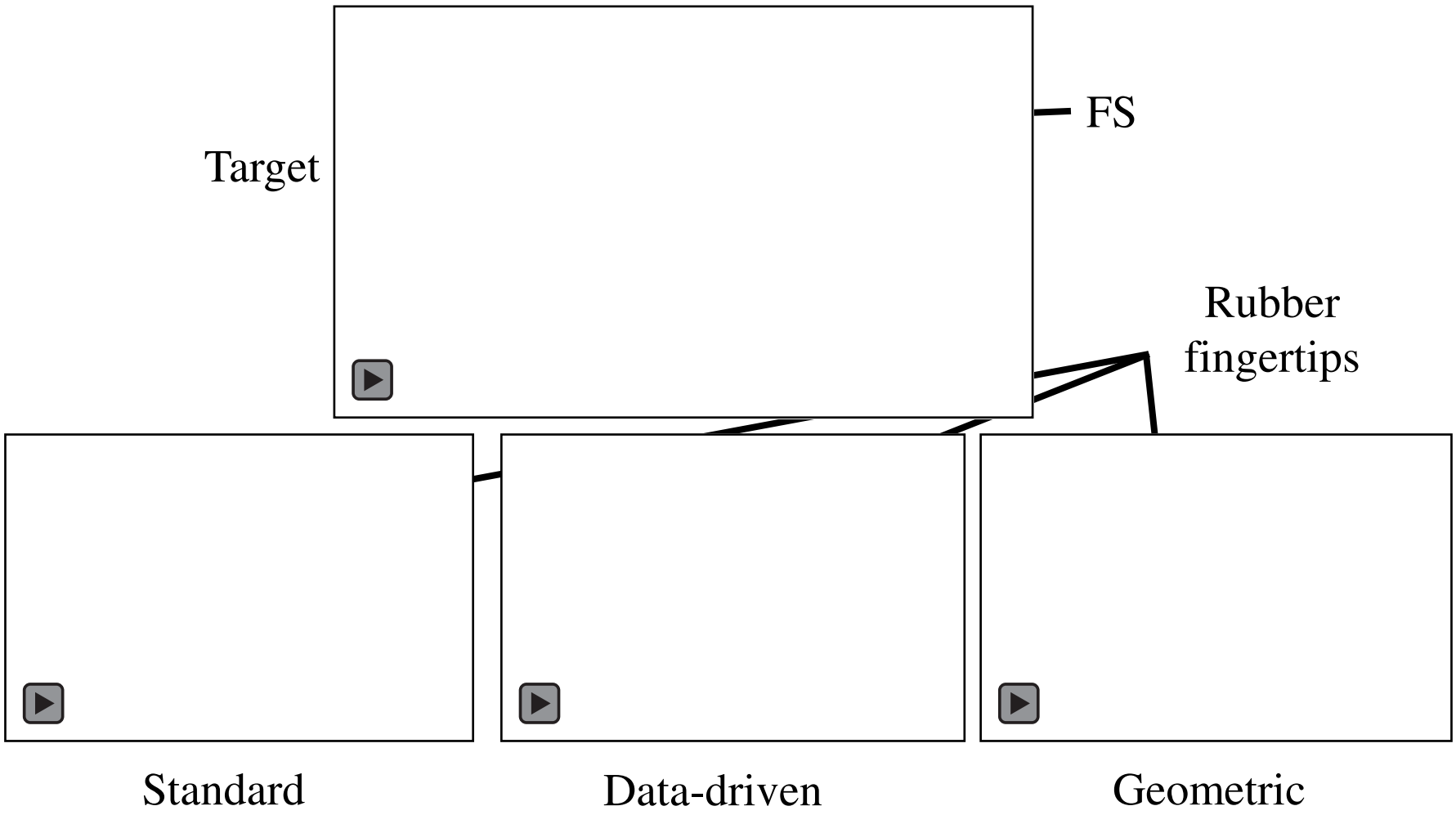
Personalized haptics: the rendering approach

Evaluation example

Render of a set of predefined tactile interactions to rubber or human fingertip using either

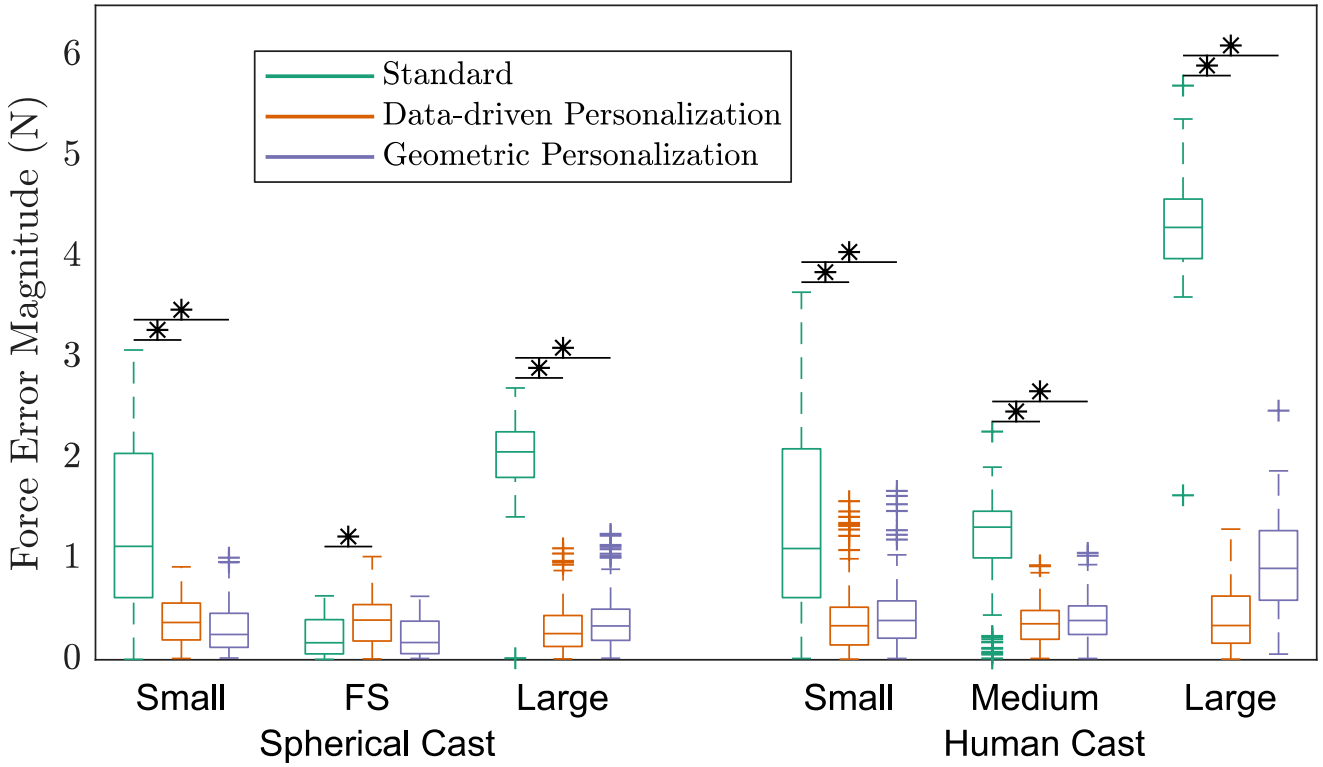
the **standard, non-personalized** method**,**
the **data-driven personalization** method**,** or
the **geometric personalization** method**.**

Personalized haptics: the rendering approach



Personalized haptics: the rendering approach

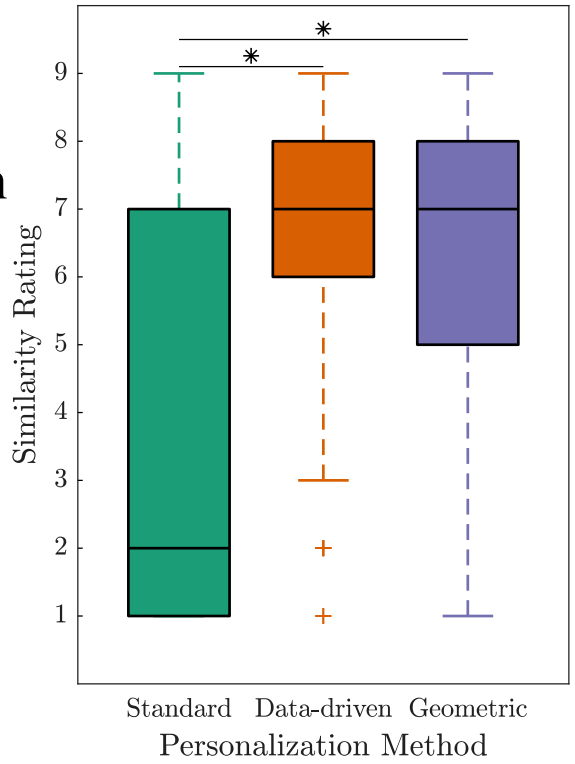
Objective Evaluation

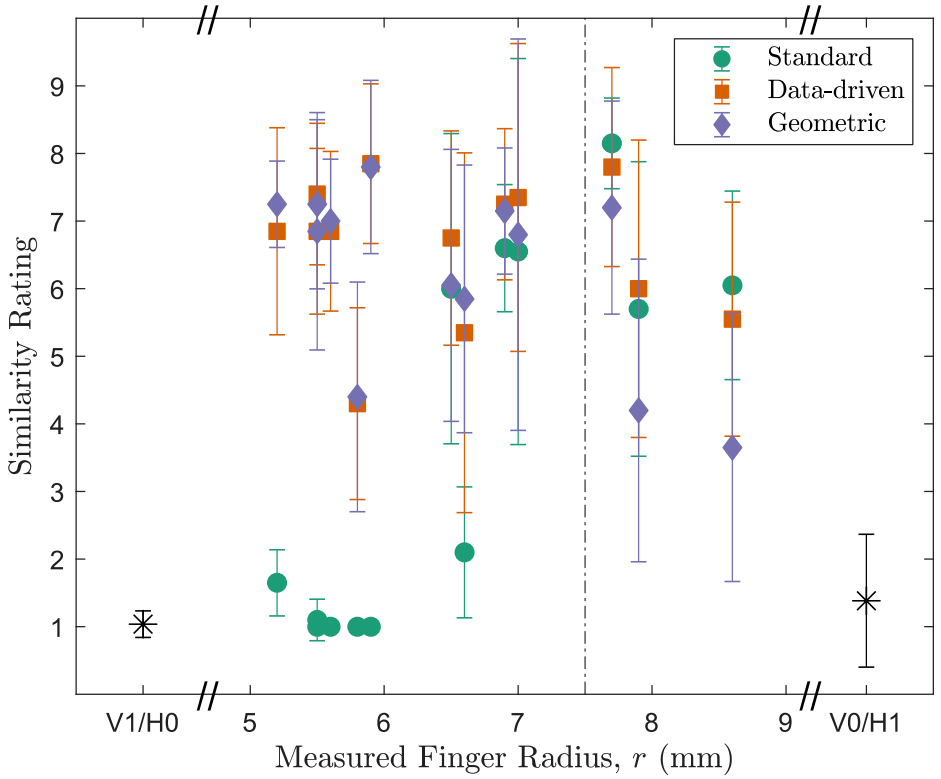


The data-driven and geometric personalizations significantly reduced the force rendering error compared to the standard approach for all of the rubber fingertips other than the FS.

Personalized haptics: the rendering approach

Human-subject Evaluation





We found that similarity ratings were significantly higher for both the data-driven and geometric personalizations than for the standard approach.

Thank you!

Questions?