

CS277 - Experimental Haptics  
Lecture 3

# Proxy-Based Rendering



# Outline

- ▶ Announcements
- ▶ A note on timing...
- ▶ Limitations with potential fields
- ▶ God-object algorithm
- ▶ Finger-proxy algorithm

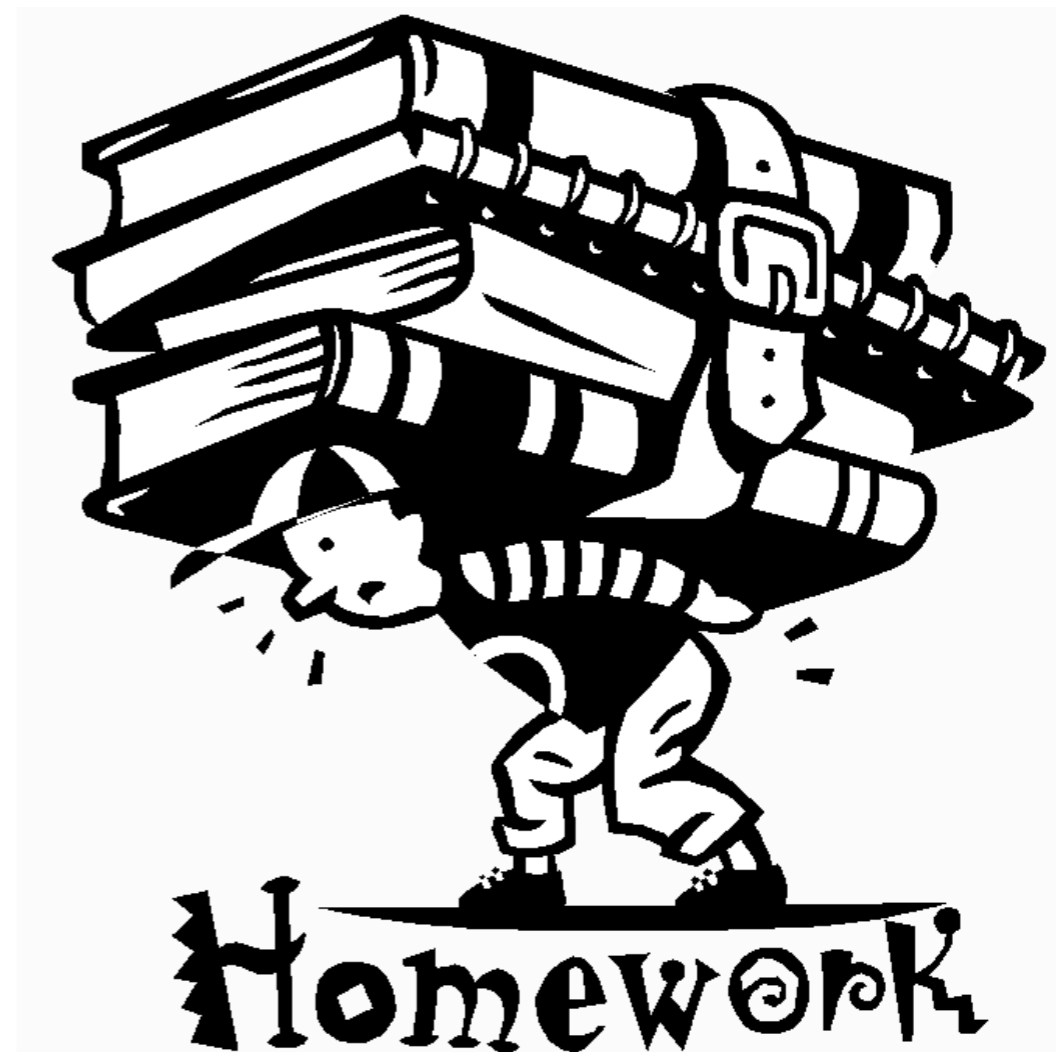
# Course Project Showcase



**Afternoon of Friday, May 30th?**

# Programming Assignments

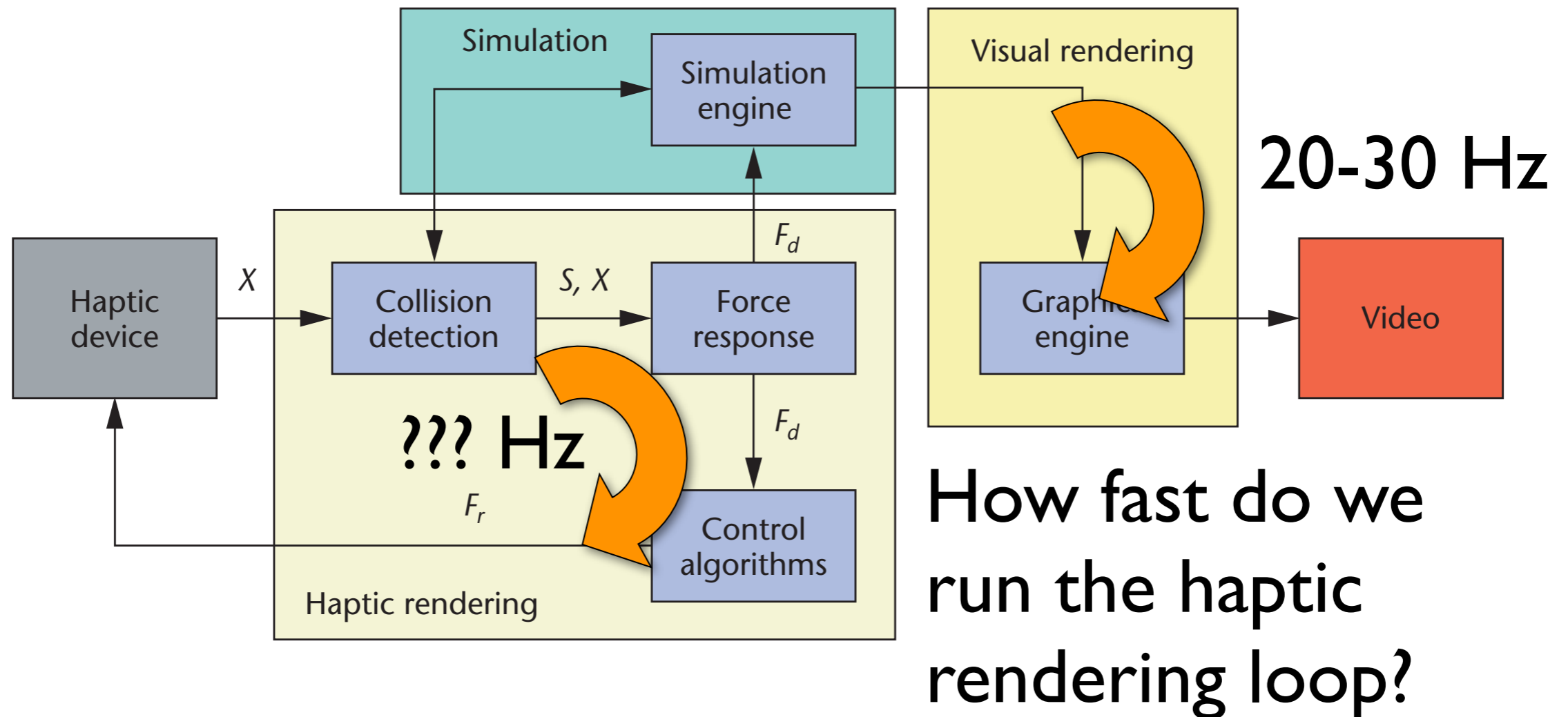
- ▶ CHAI3D libraries and application template posted on CS277 site
- ▶ Assignment #1 will be posted later today
- ▶ Picked up your Falcon?



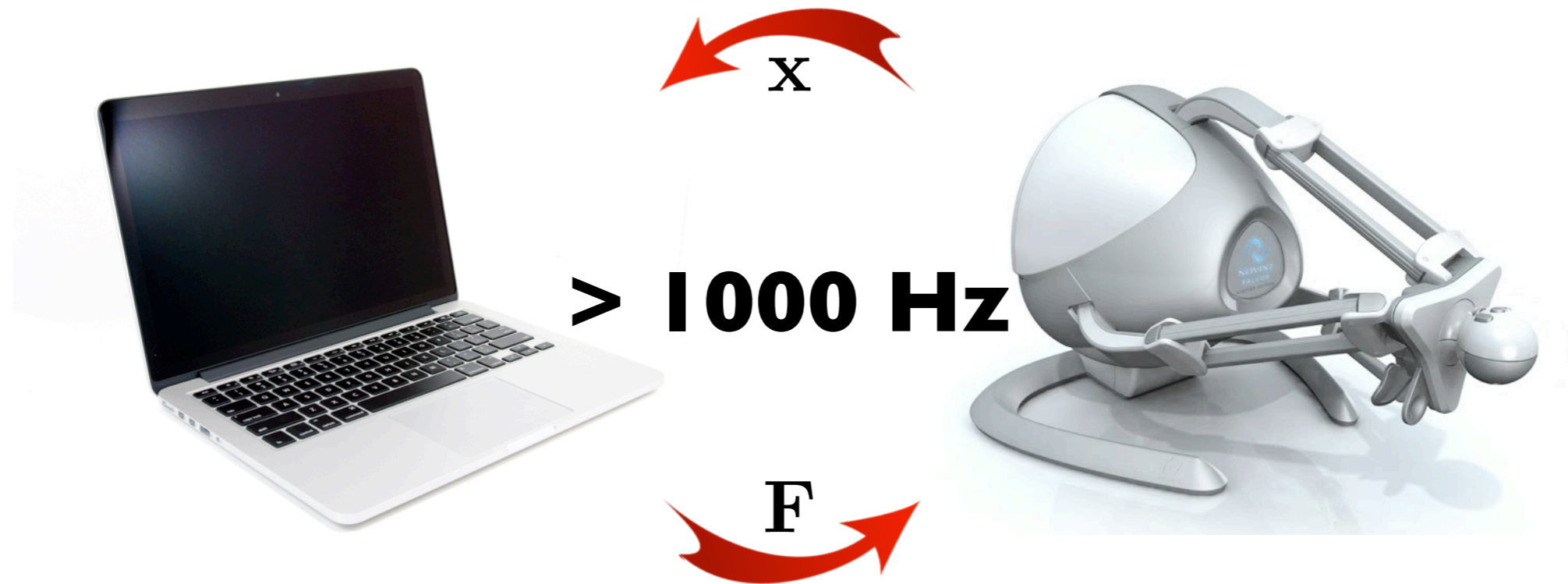


# A Note on Timing

# Visual-Haptic Simulation



# Haptic Rendering Loop



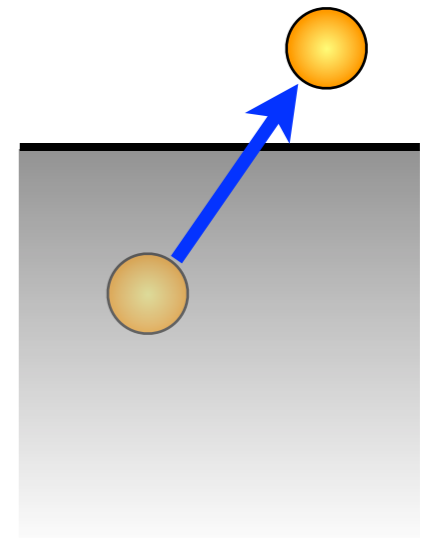
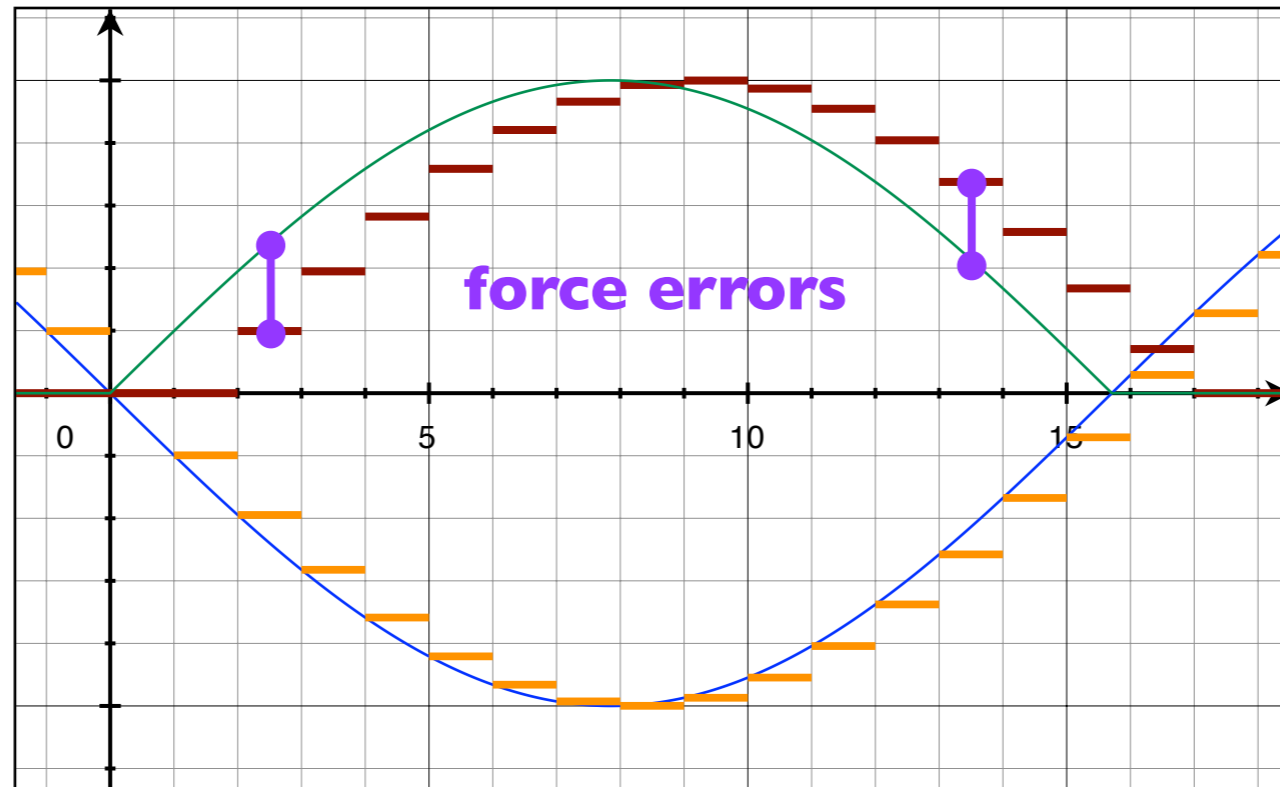
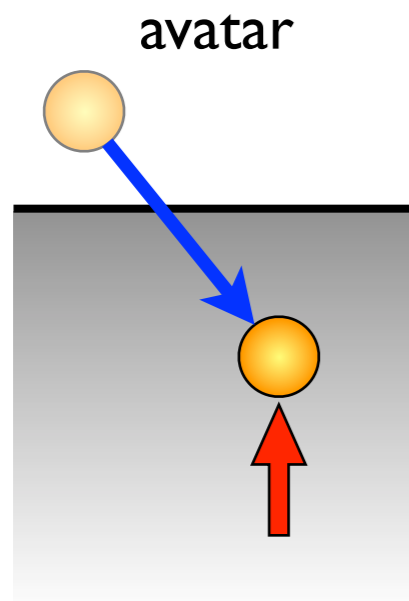
The faster the better!

# Human Perception

- ▶ Mechanoreceptors in our skin can detect vibrations up to ~400 Hz
- ▶ We are generating a discrete force signal
  - sampling rate should be above Nyquist

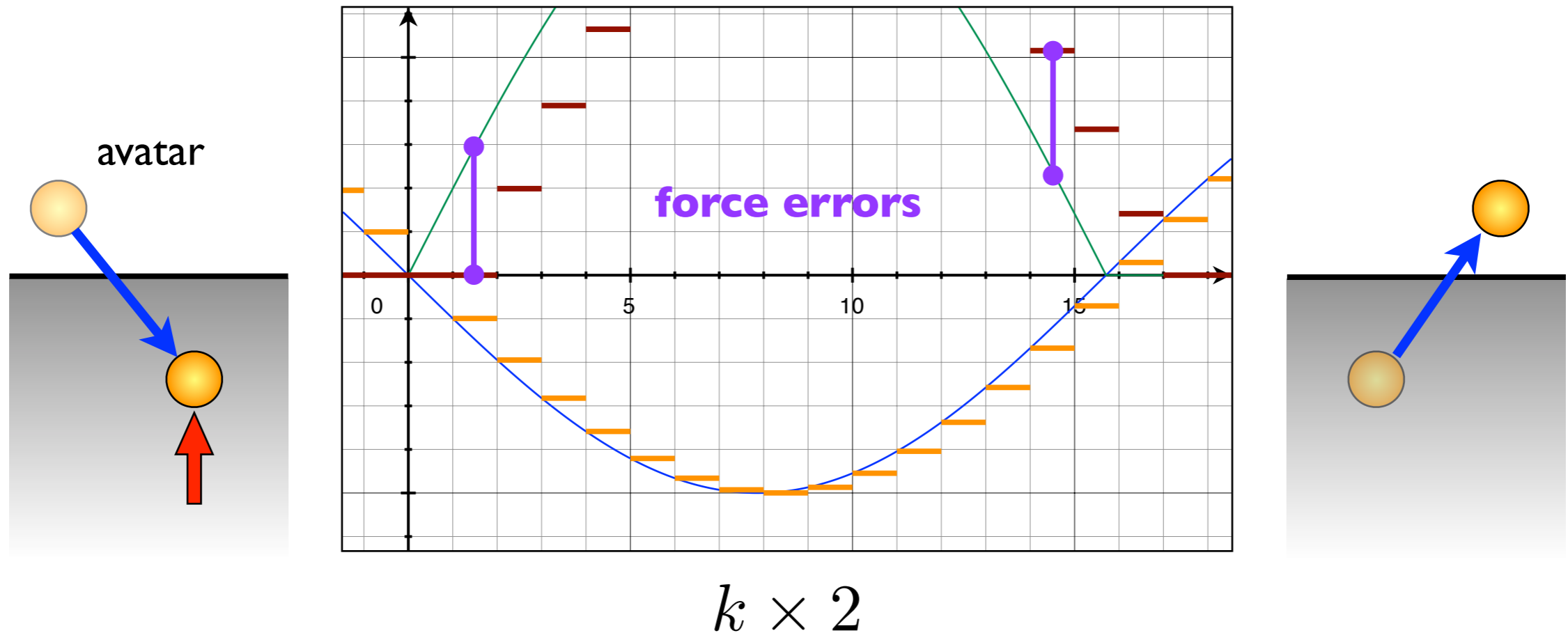


# Sampled-Data System



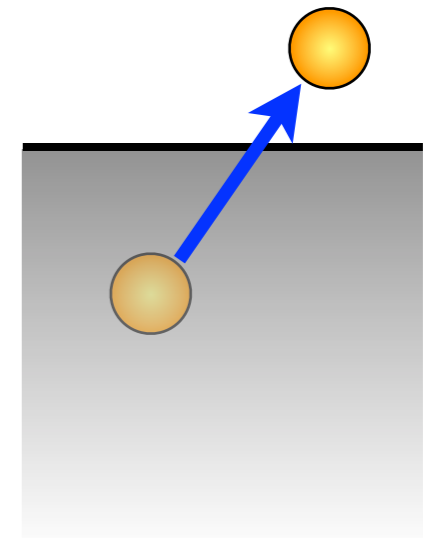
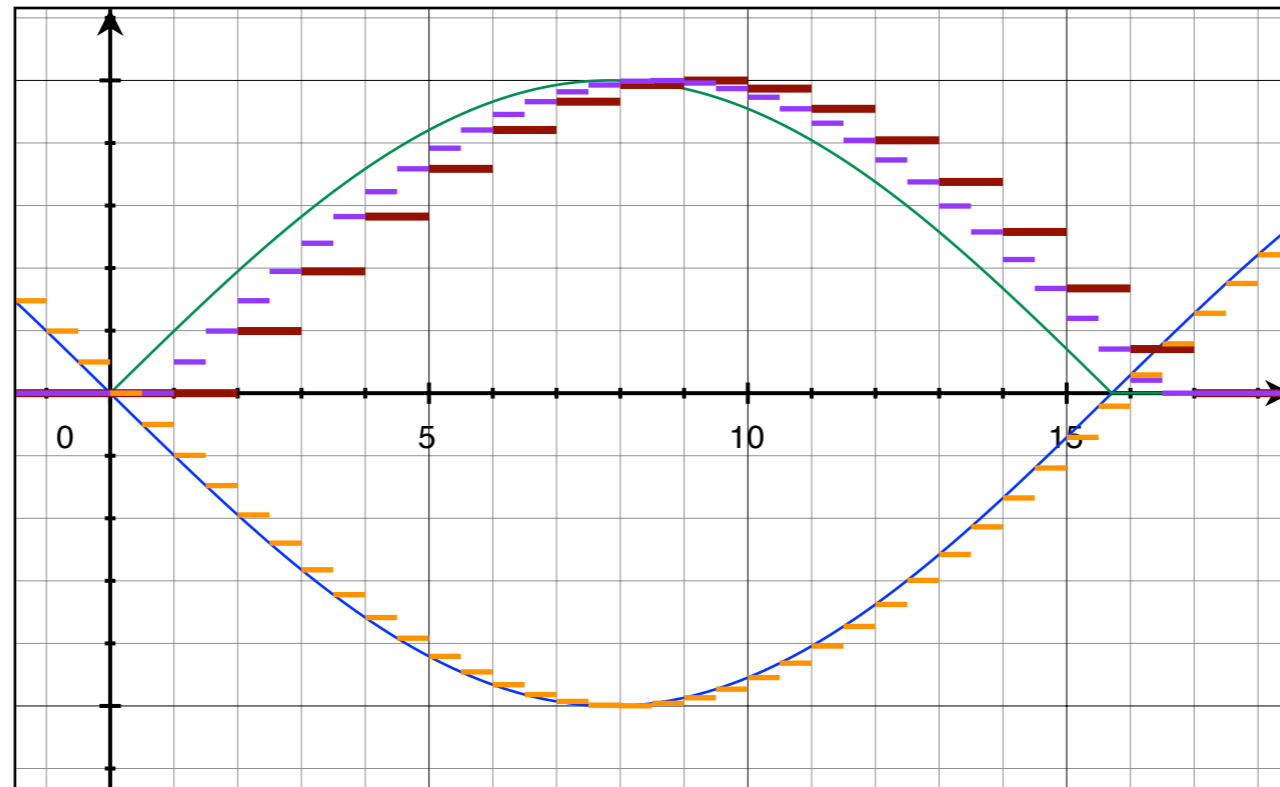
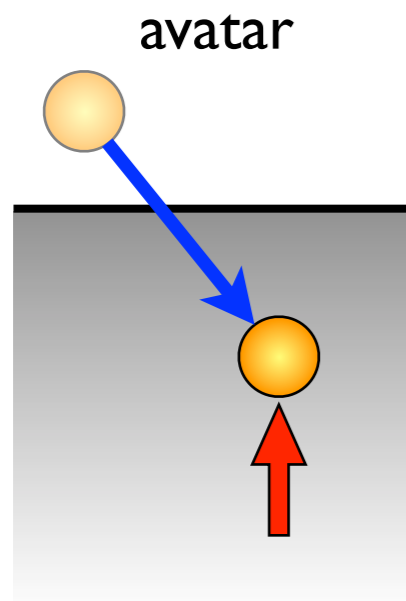
- $x$  position
- $F$  ideal force
- $\underline{x}$  sampled position
- $\underline{F}$  commanded force

# Effect of Stiffness



**Errors get worse as stiffness is increased!**

# Servo Rate Effects



Errors are reduced with an increased servo rate

# Summary of Challenges

- ▶ High stiffness and low servo rate can cause excess energy to be generated
- ▶ Energy must be dissipated through mechanical device
- ▶ Advanced control algorithms may also help

# On Stiffness...



700 N/m

> 100 N/m



> 1000 N/m



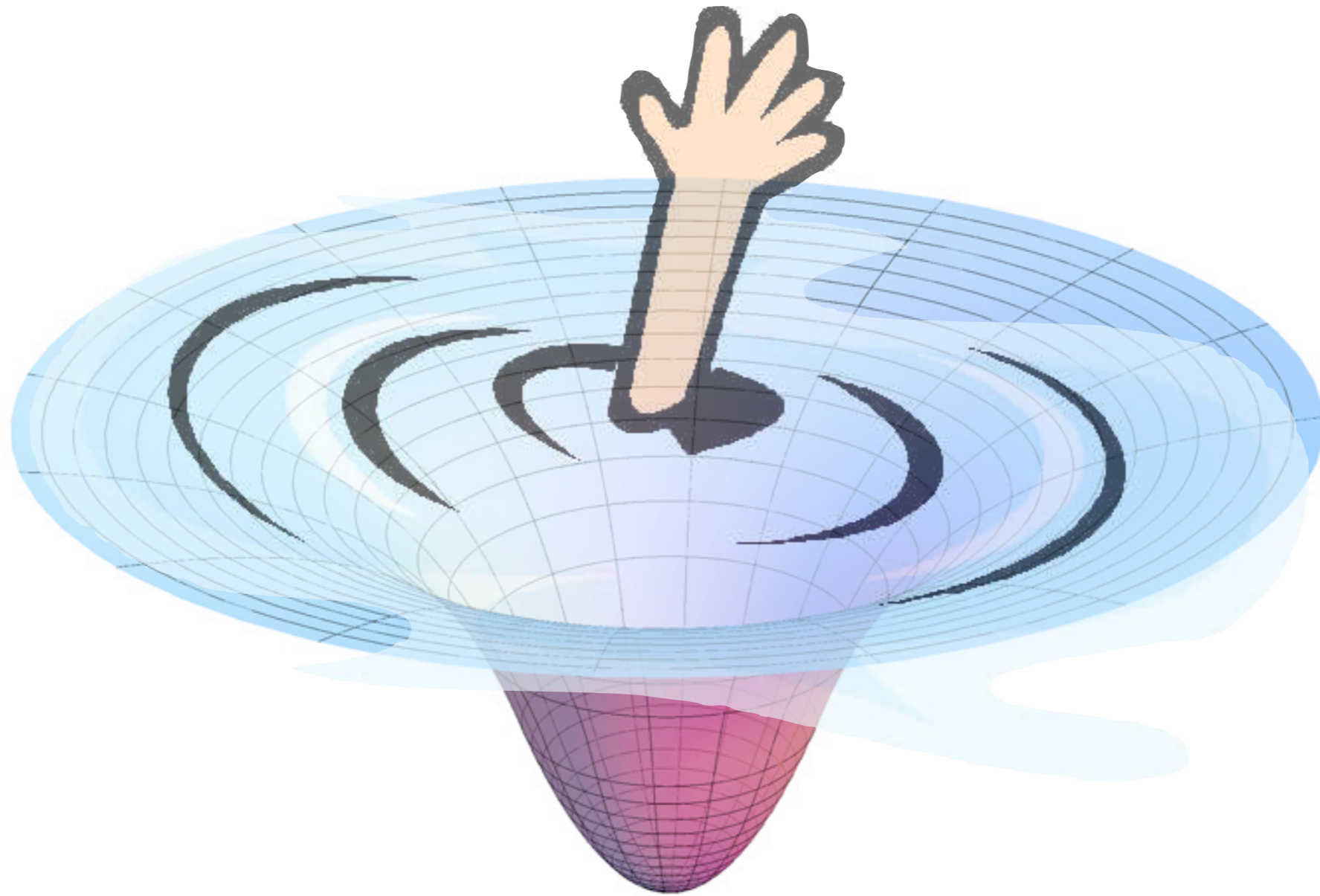
> 10,000 N/m



5000 N/m

> 100,000 N/m

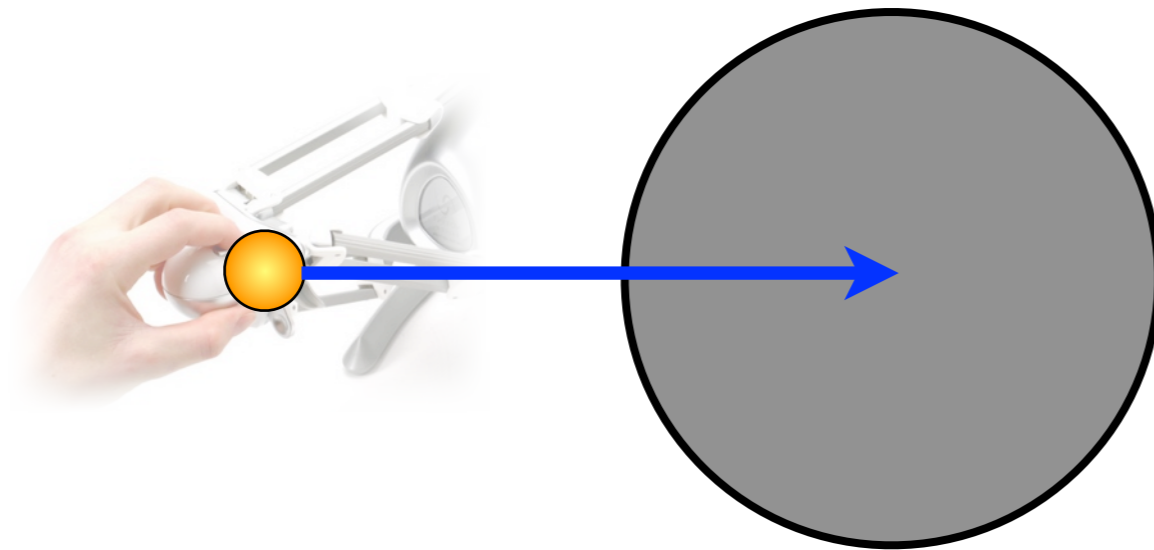
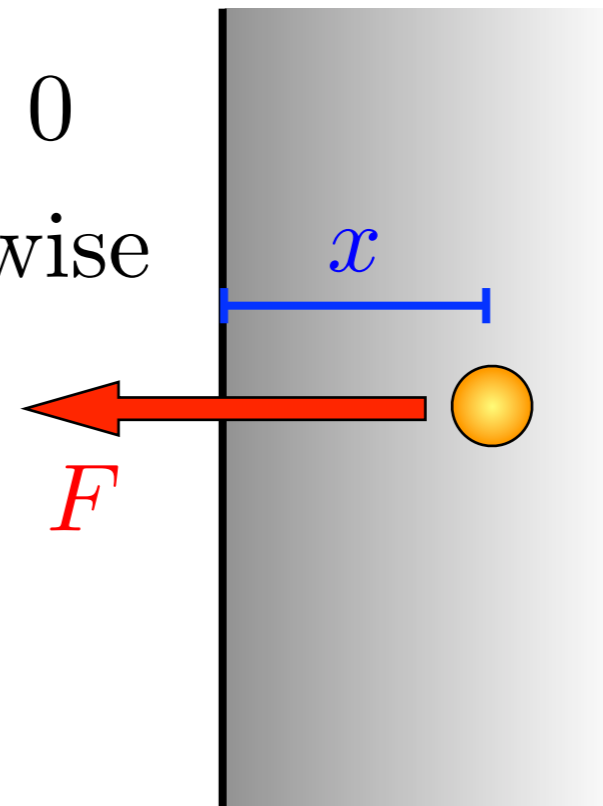




# Potential Field Issues

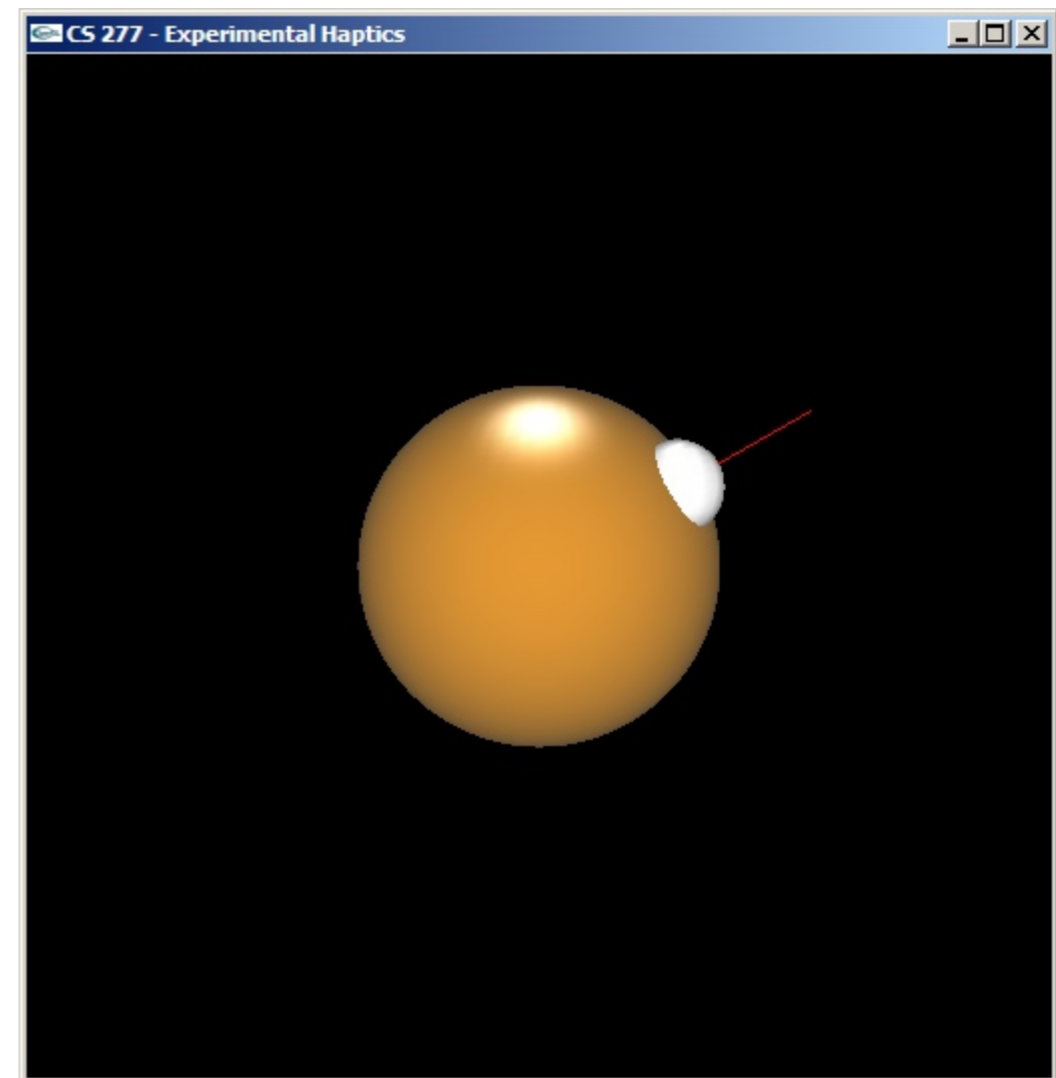
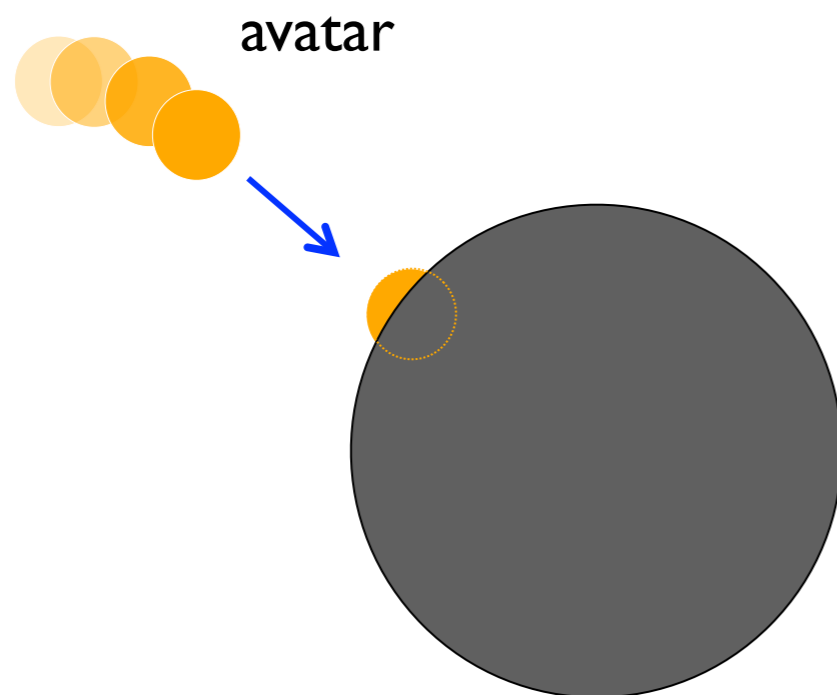
# Recall Potential Fields

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



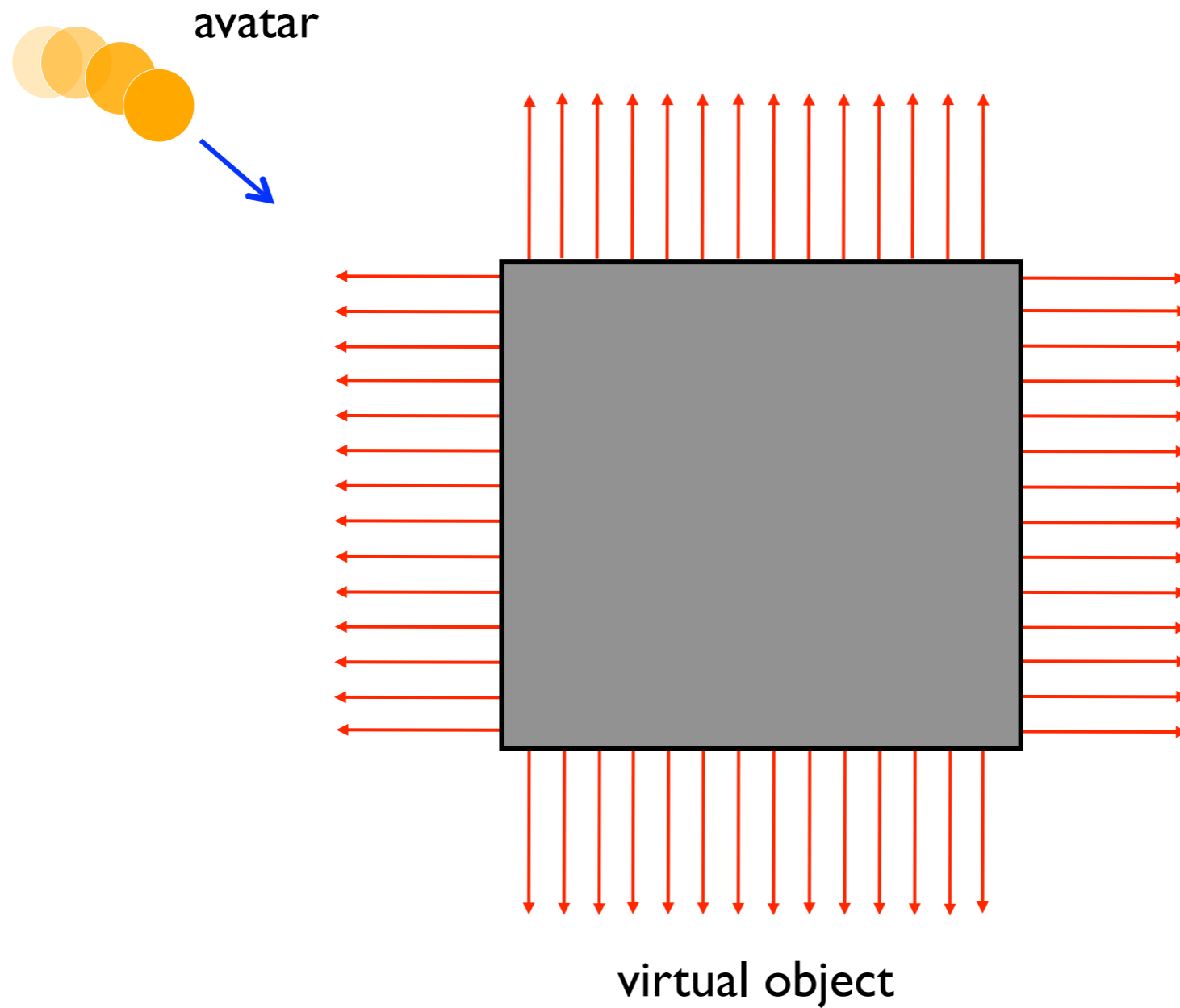
$$F(x, y, z) = \begin{cases} -k(x^2 + y^2 + z^2 - r^2) & \text{if } x^2 + y^2 + z^2 < r^2 \\ 0 & \text{otherwise} \end{cases}$$

# The Sinking Avatar

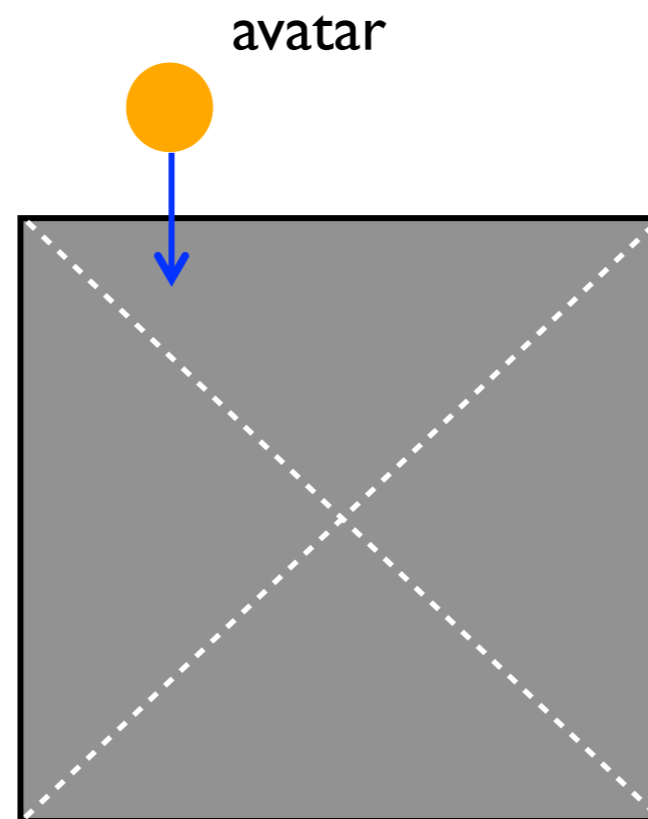




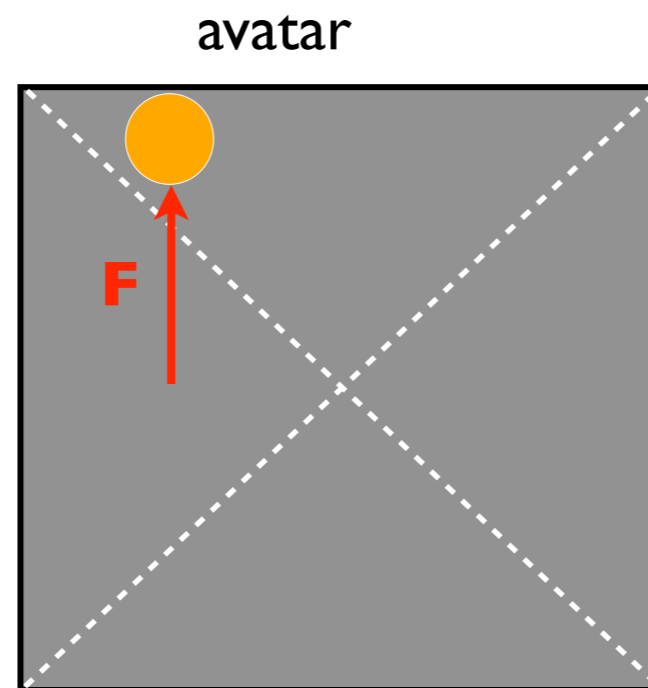
# Virtual Box



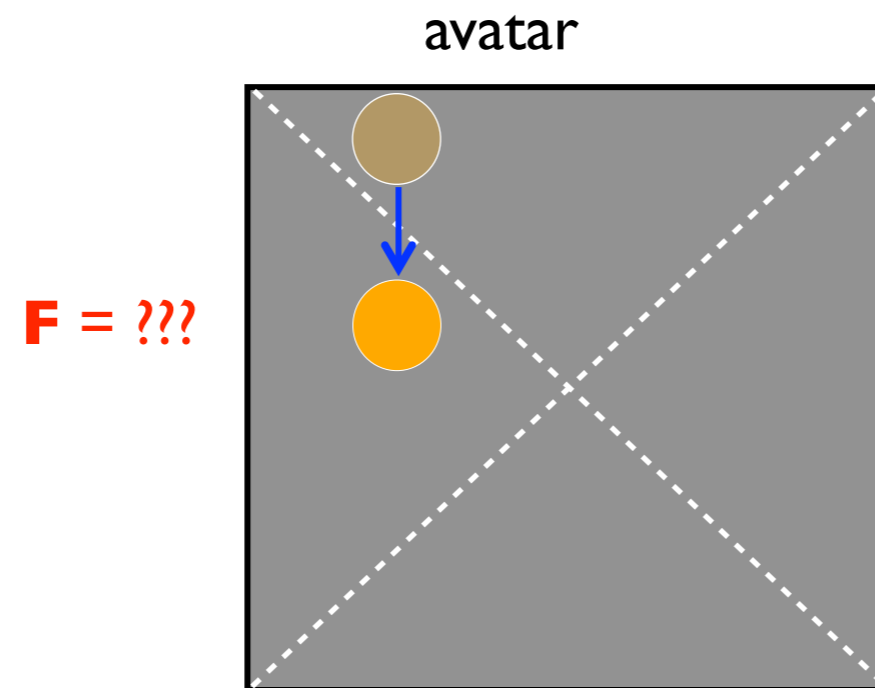
# Virtual Box



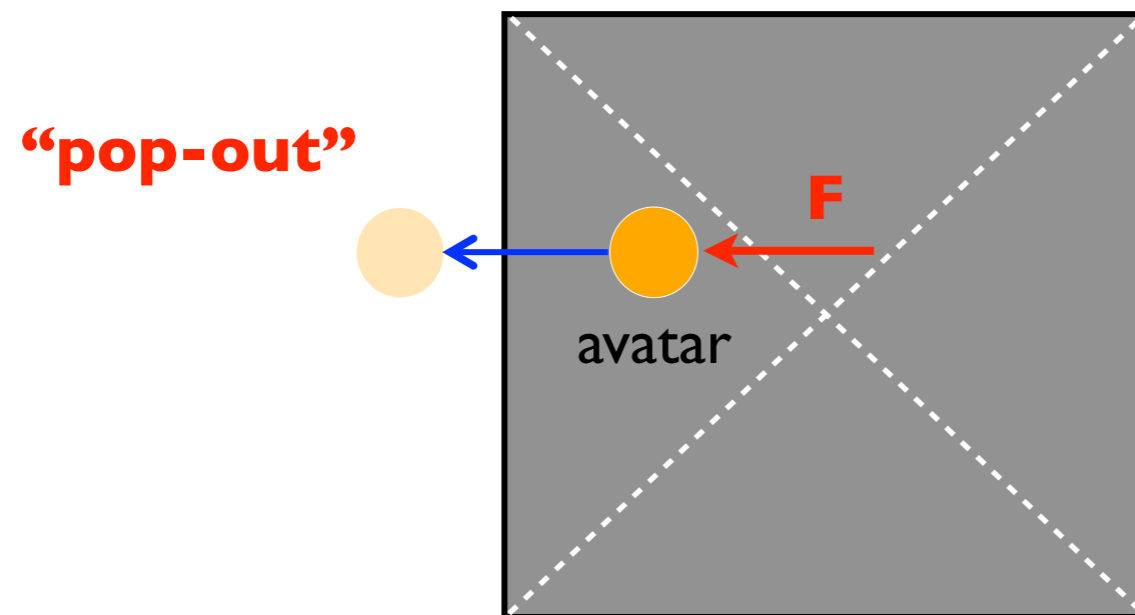
# Virtual Box



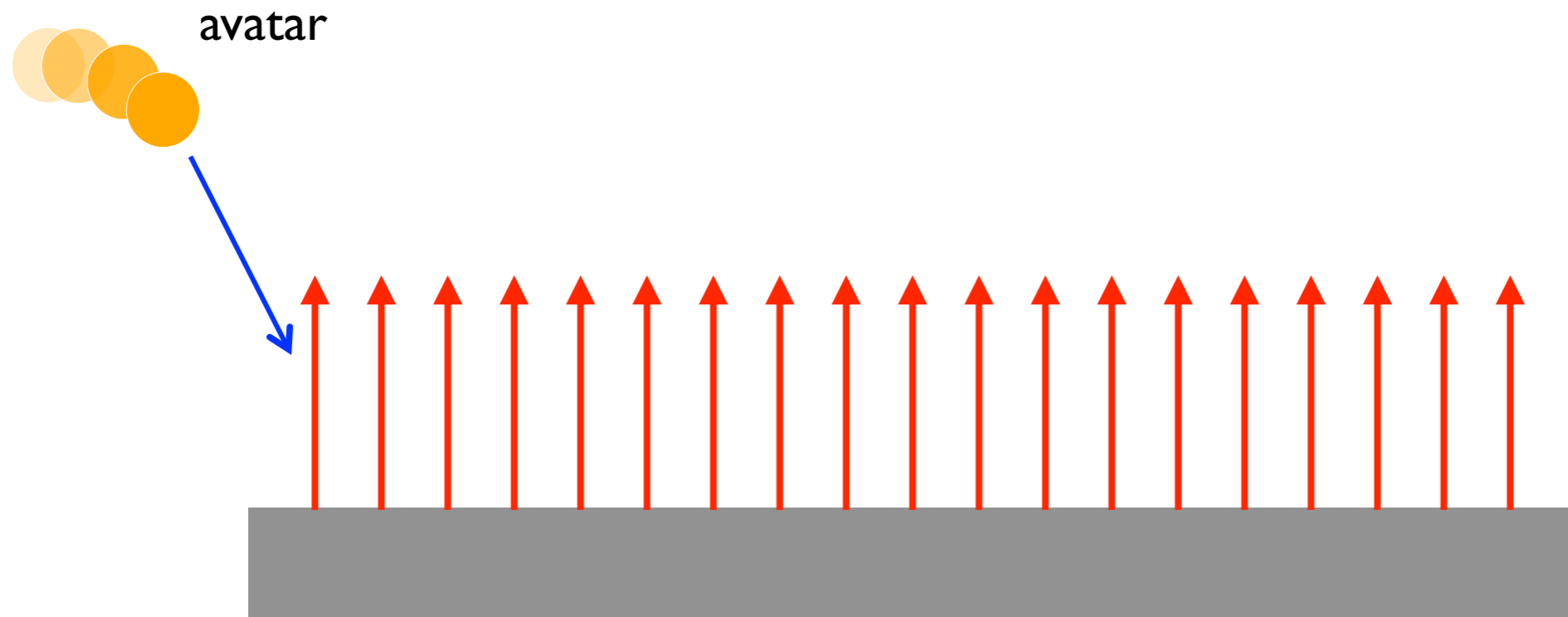
# Virtual Box



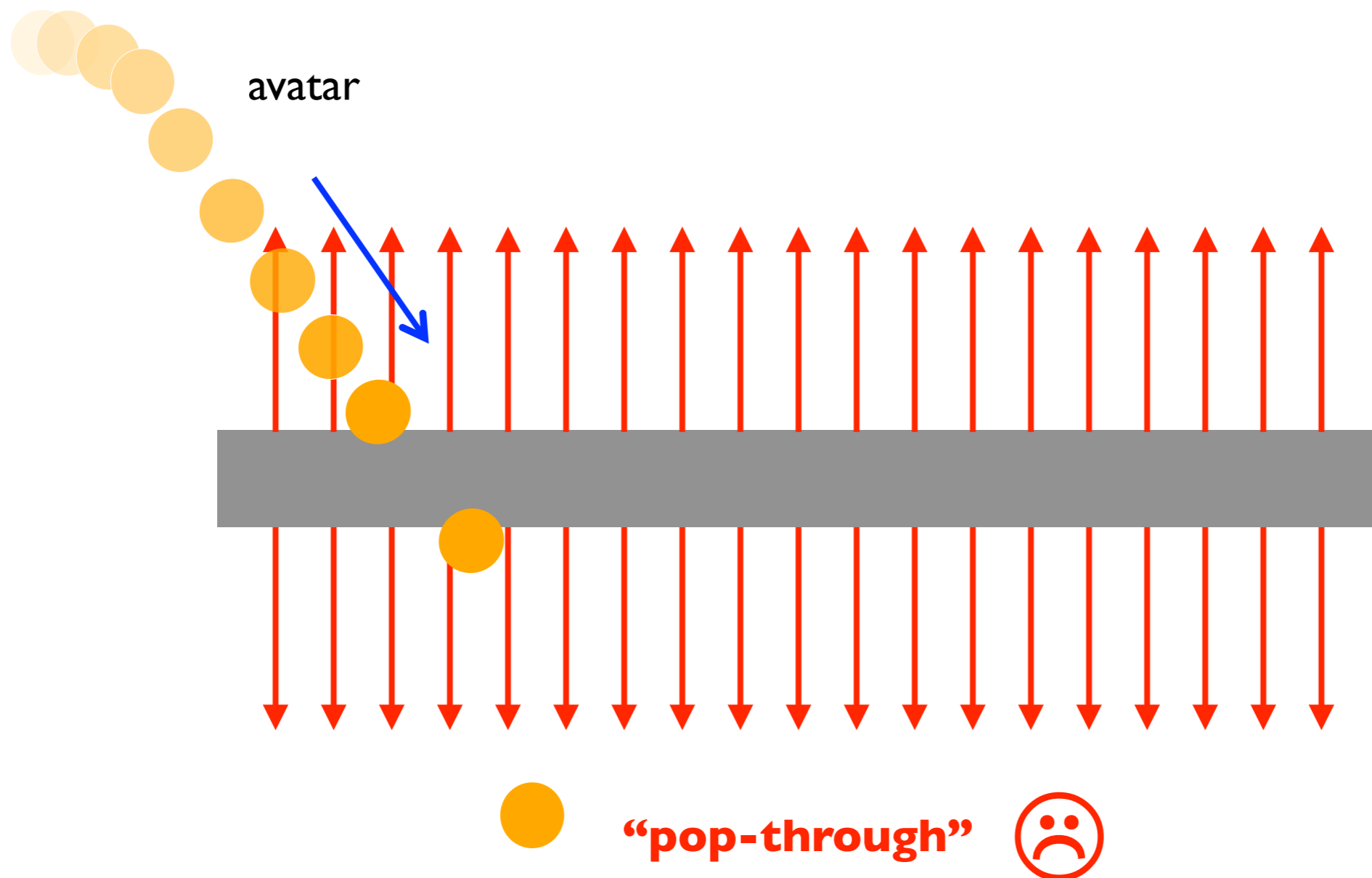
# Virtual Box



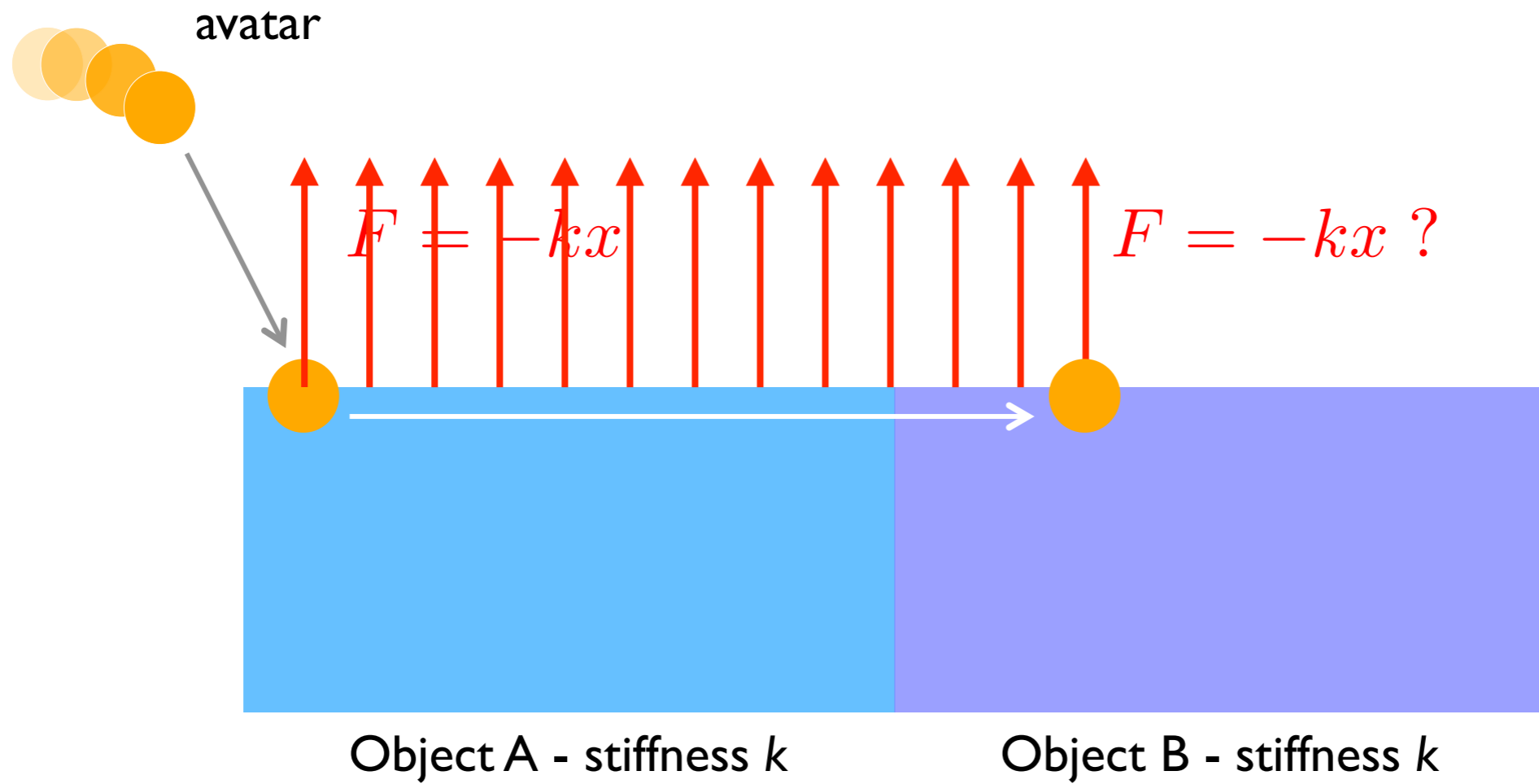
# Thin Objects



# Thin Objects

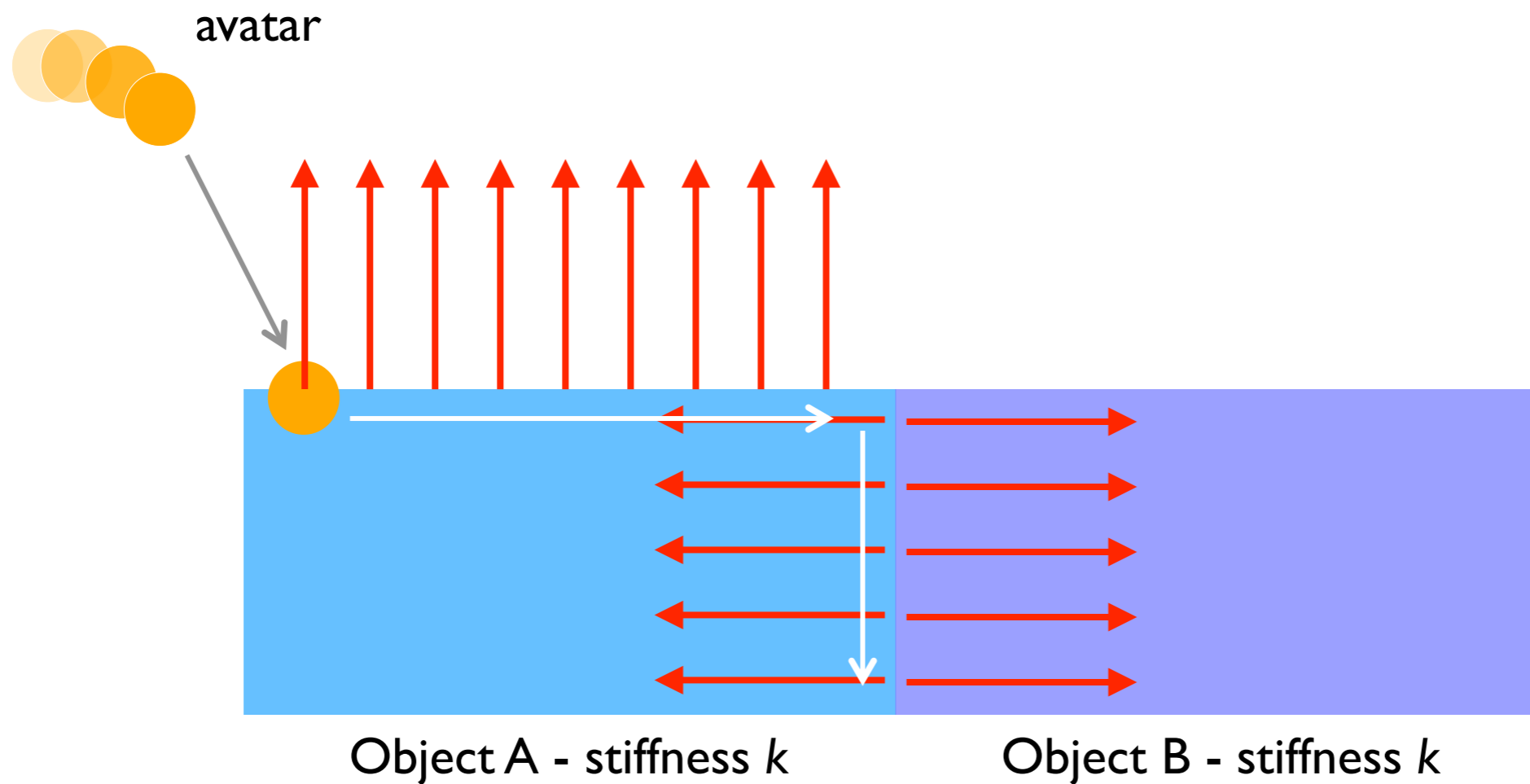


# Building Blocks

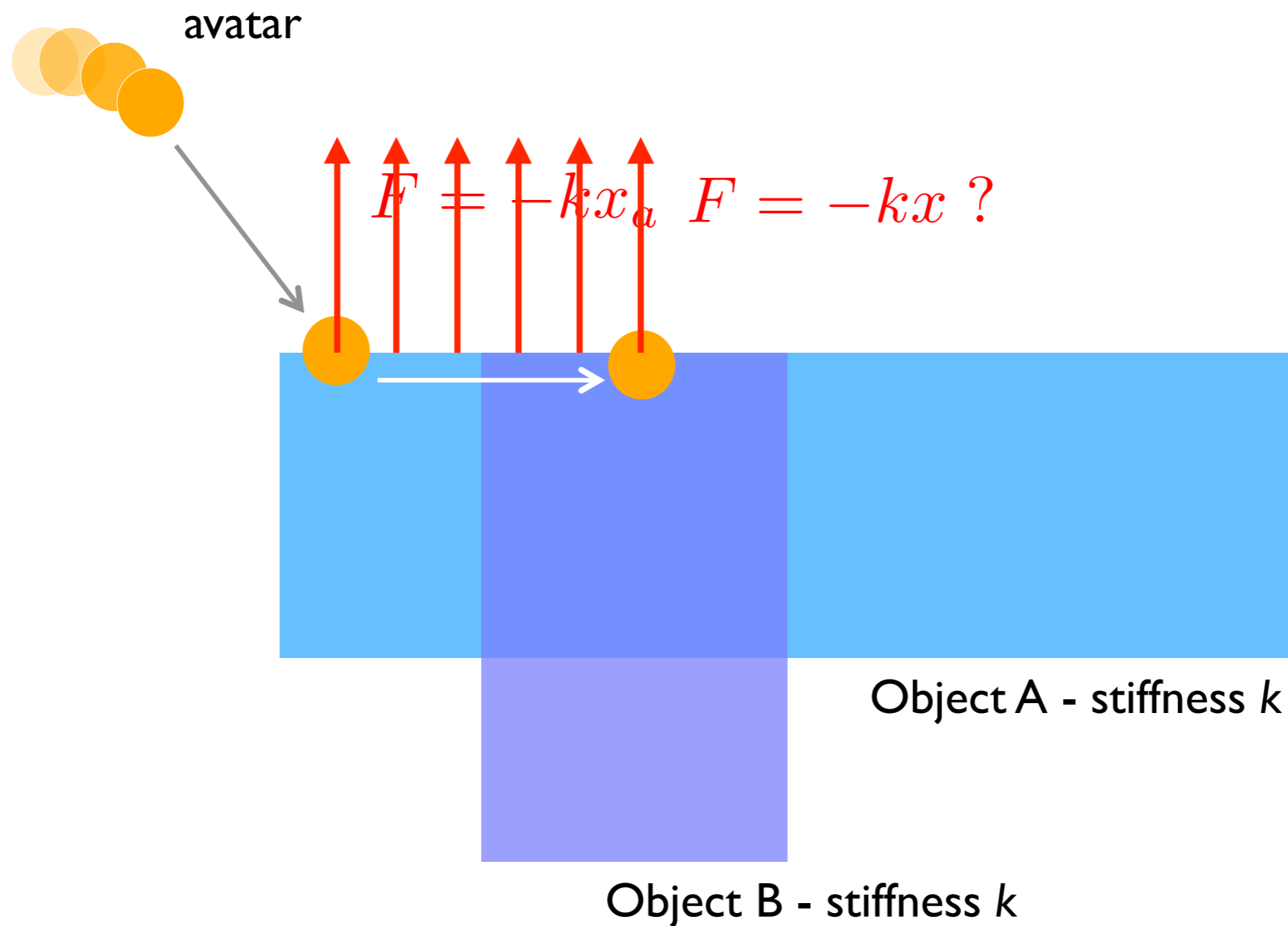




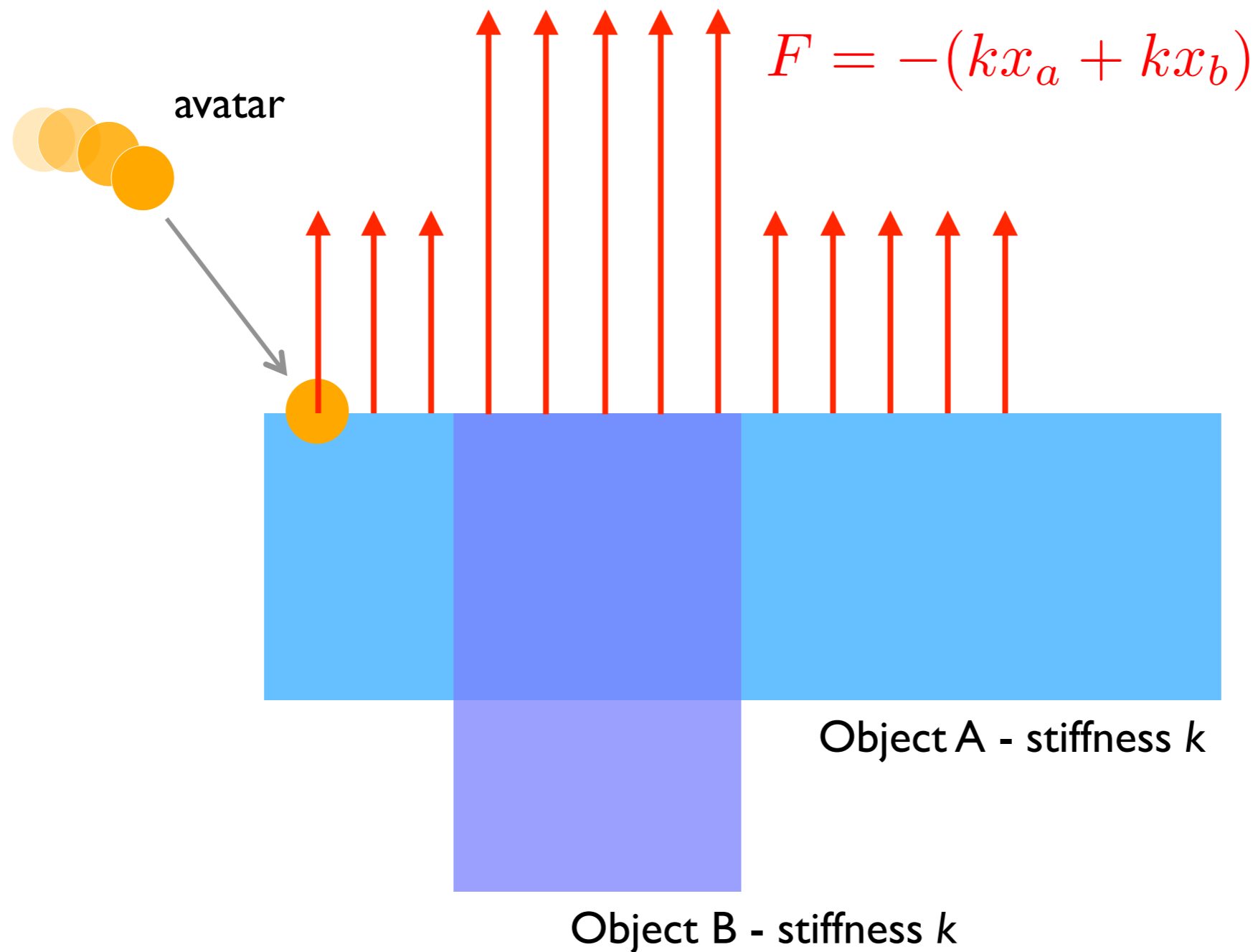
# Gap in the Middle!



# Overlapping Blocks



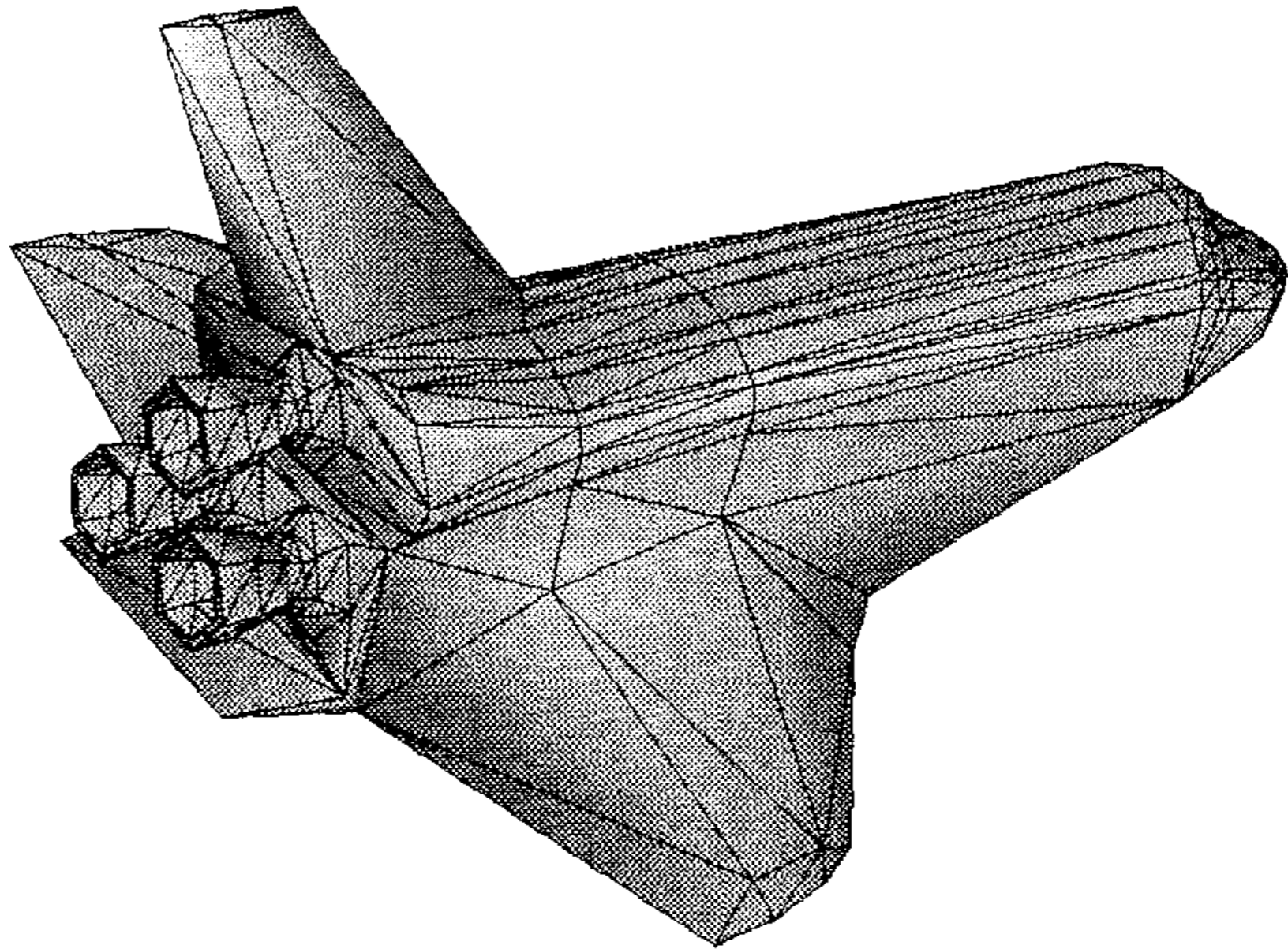
# Stiffness Variation!



# Potential Field Problems

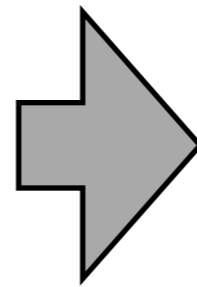
- ▶ Pop-out and pop-through
- ▶ Gaps between objects
- ▶ Stiffness variation with overlapping objects
  
- ▶ How do we deal with these???





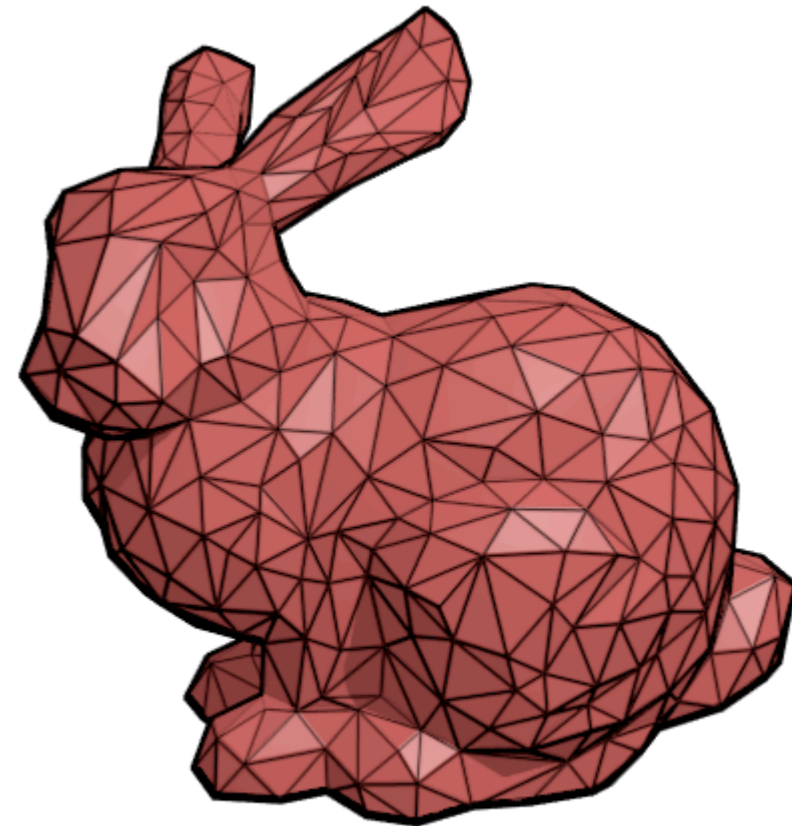
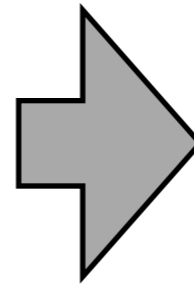
# God-Object Algorithm

# On Geometric Representation

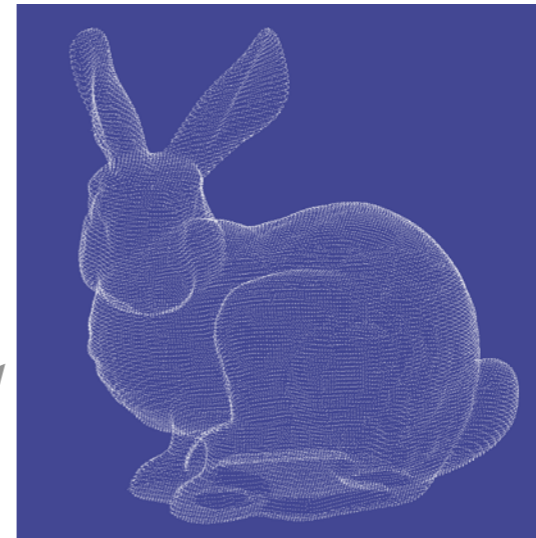


01001011 ???

# Polygonal Meshes



# Other Representations



point cloud



volumetric

potential field???  
implicit surface???



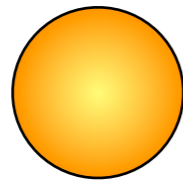
# God-Object Main Idea

- ▶ Simulate an avatar whose position and motion are independent of that of the physical device
- ▶ Avatar tries to follow the physical position while obeying laws of quantum mechanics
  - Pauli exclusion principle

# What it looks like

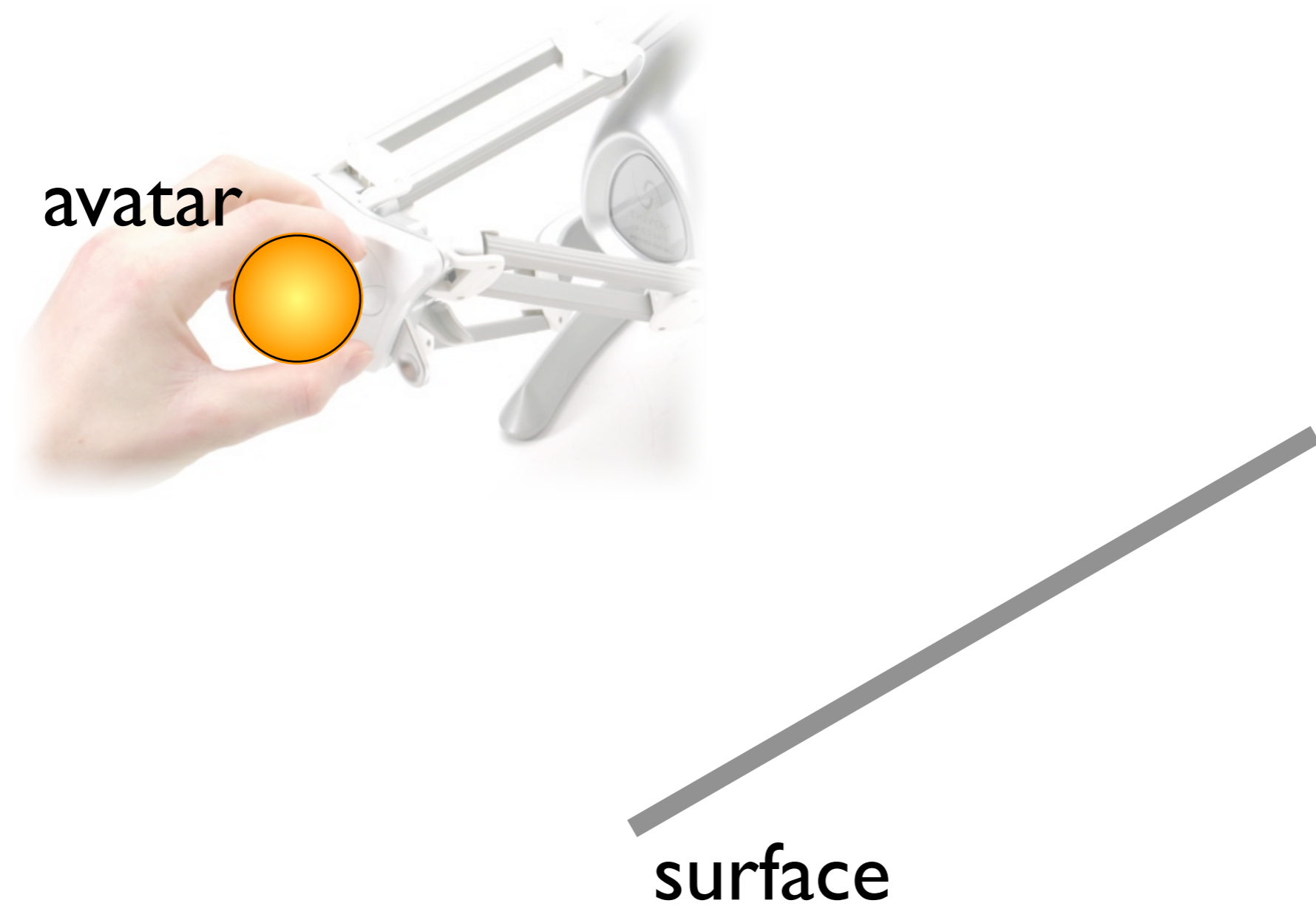
**avatar**

(a.k.a. god-object, proxy, virtual tool, HIP)

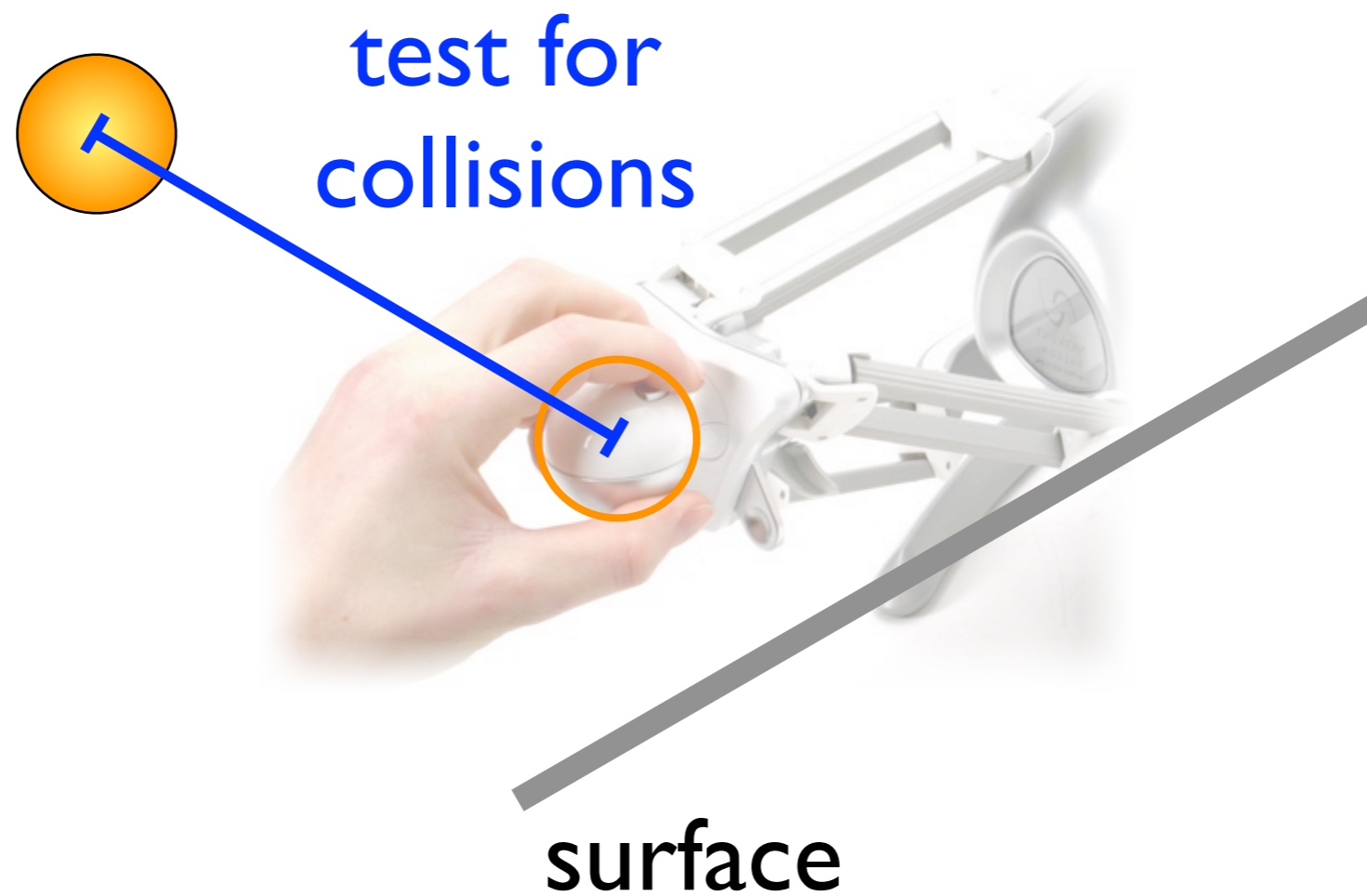


**device position**

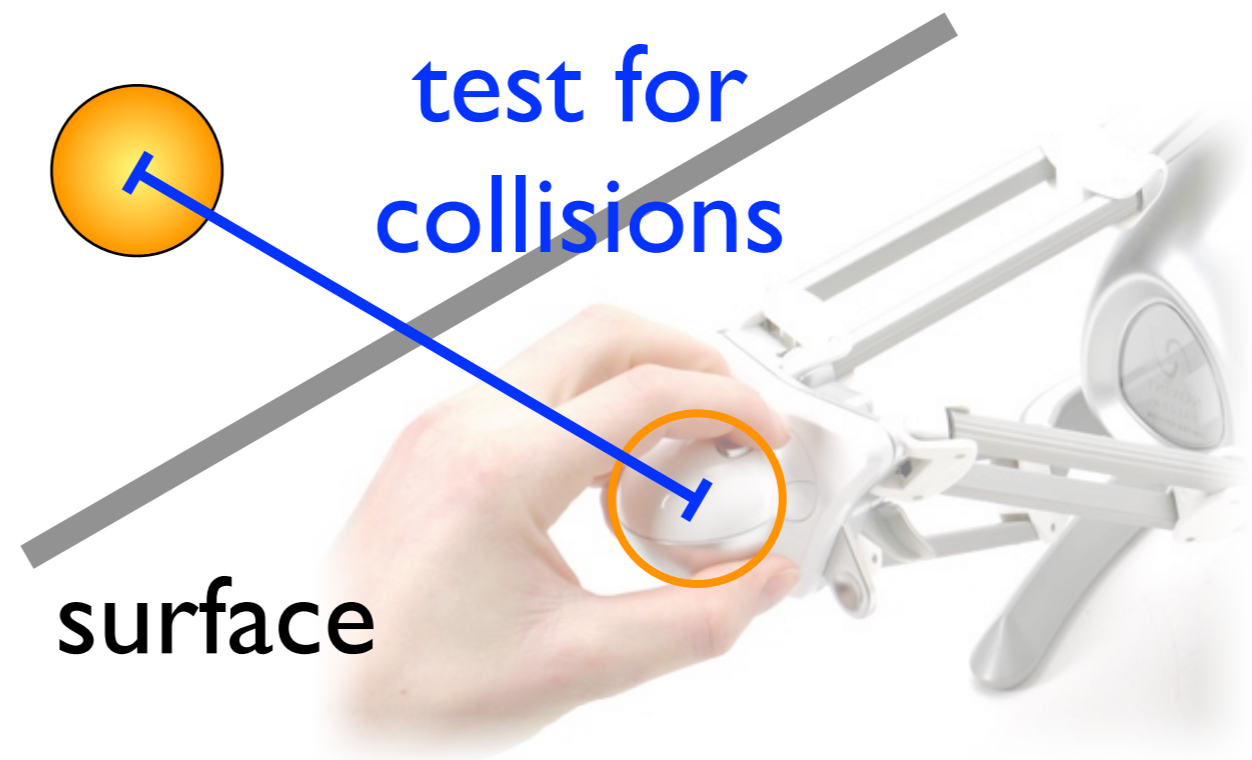
# Contact With a Surface



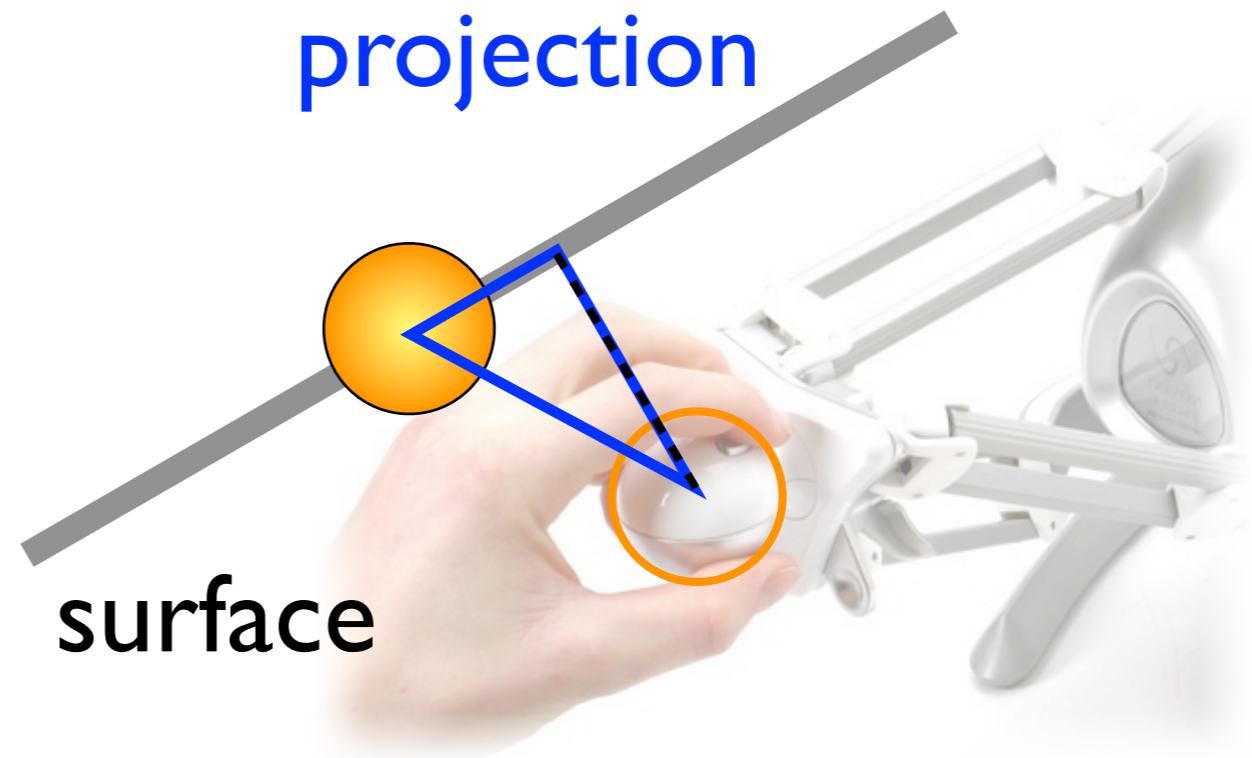
# Contact With a Surface



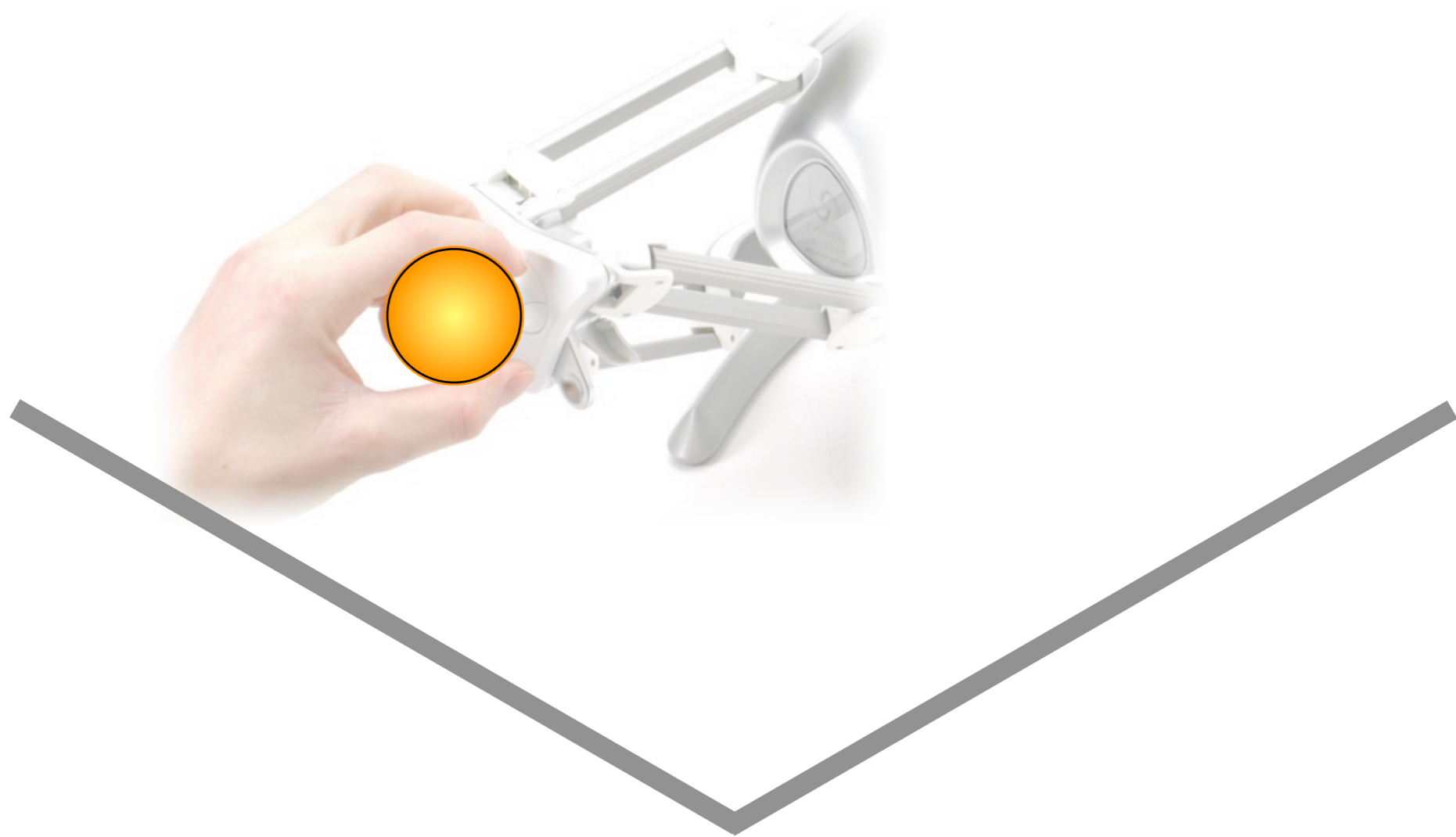
# Contact With a Surface



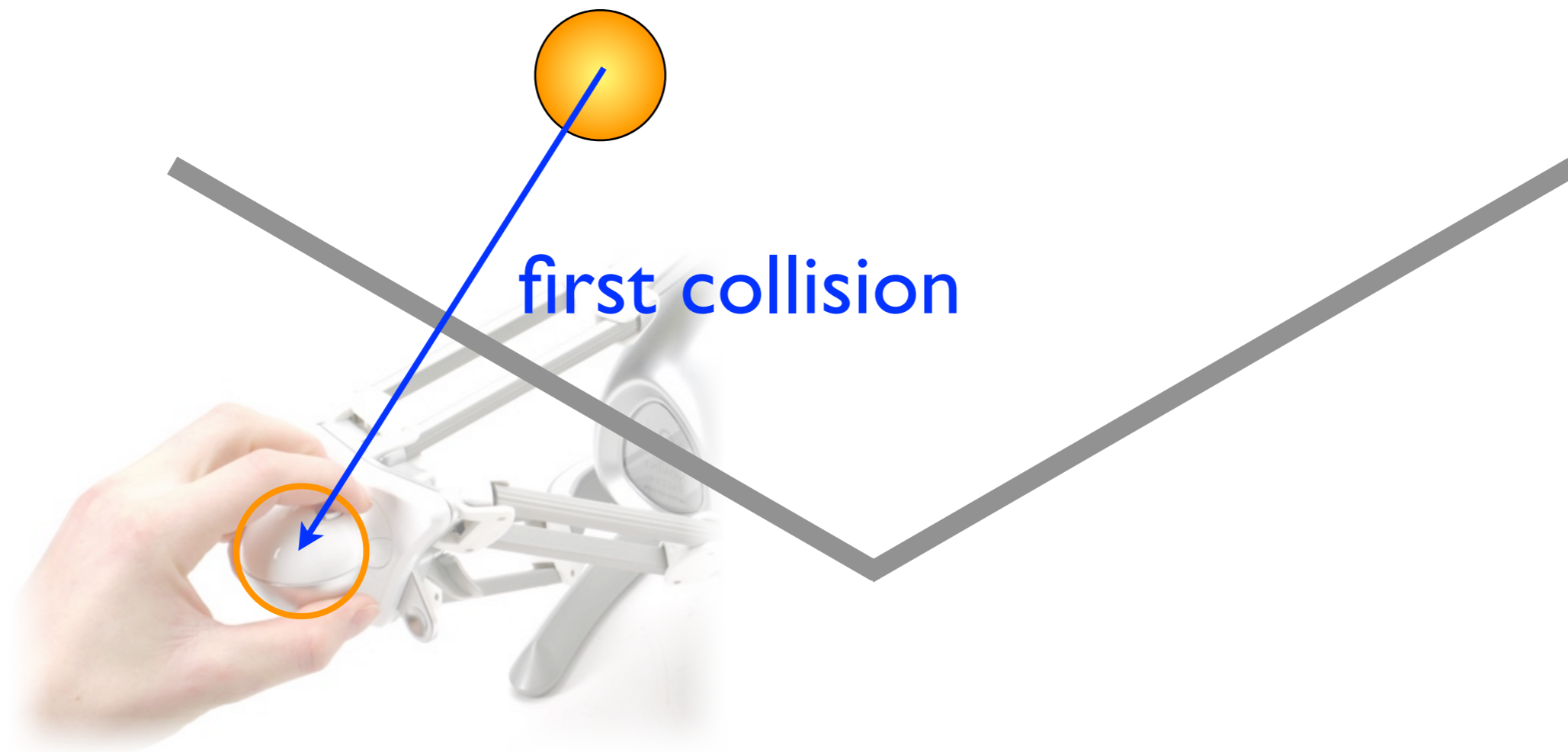
# Contact With a Surface



# Two Constraints

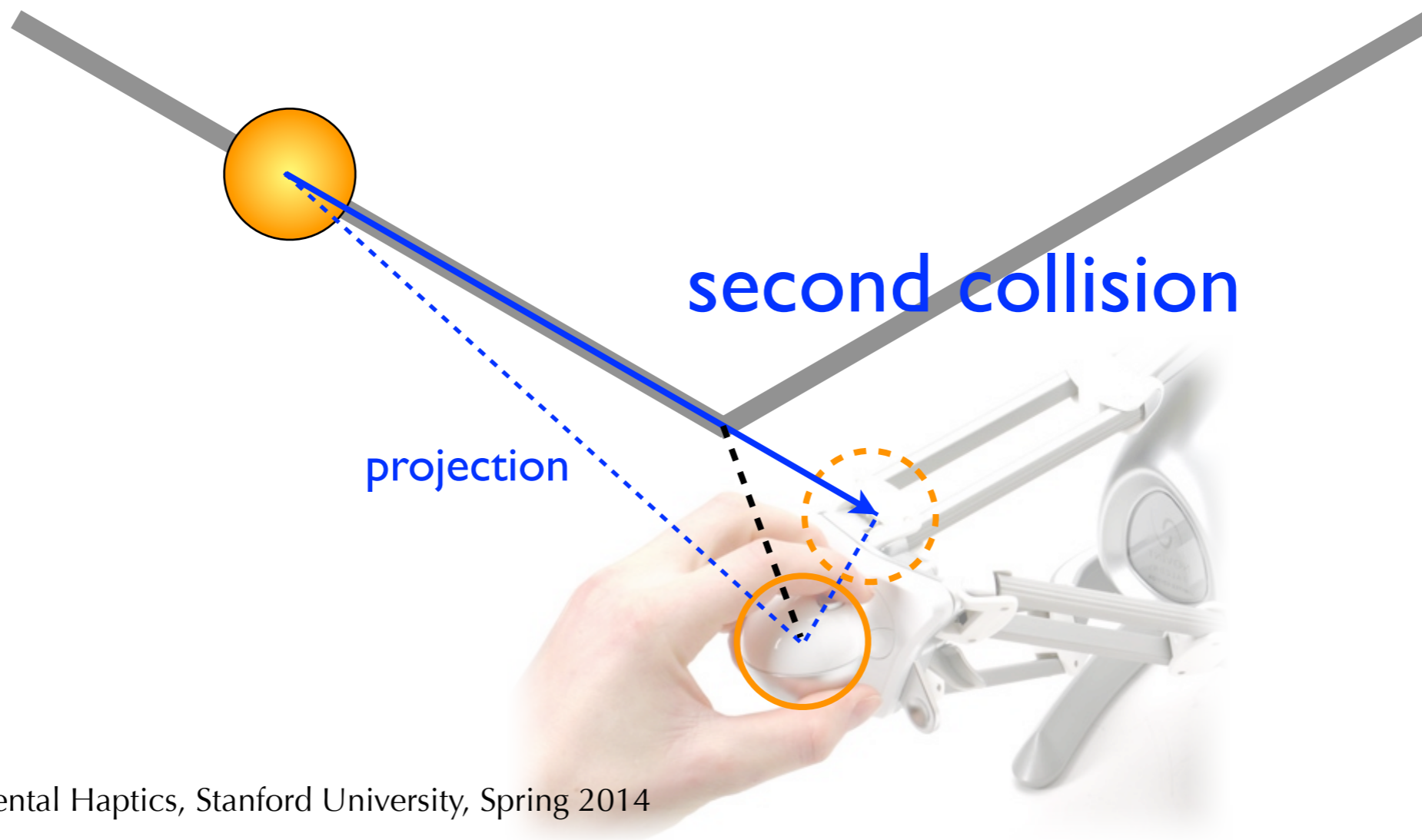


# Two Constraints

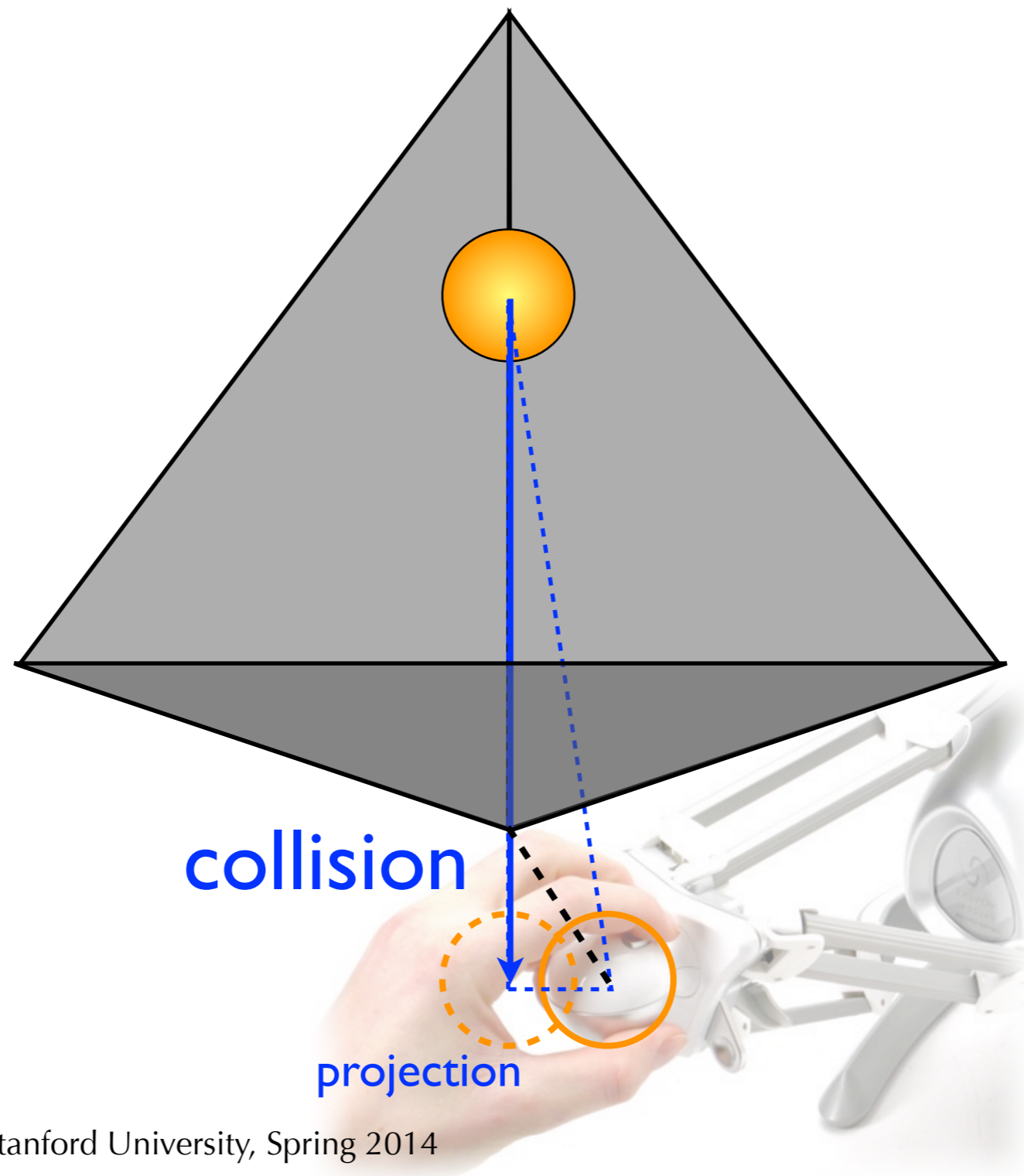


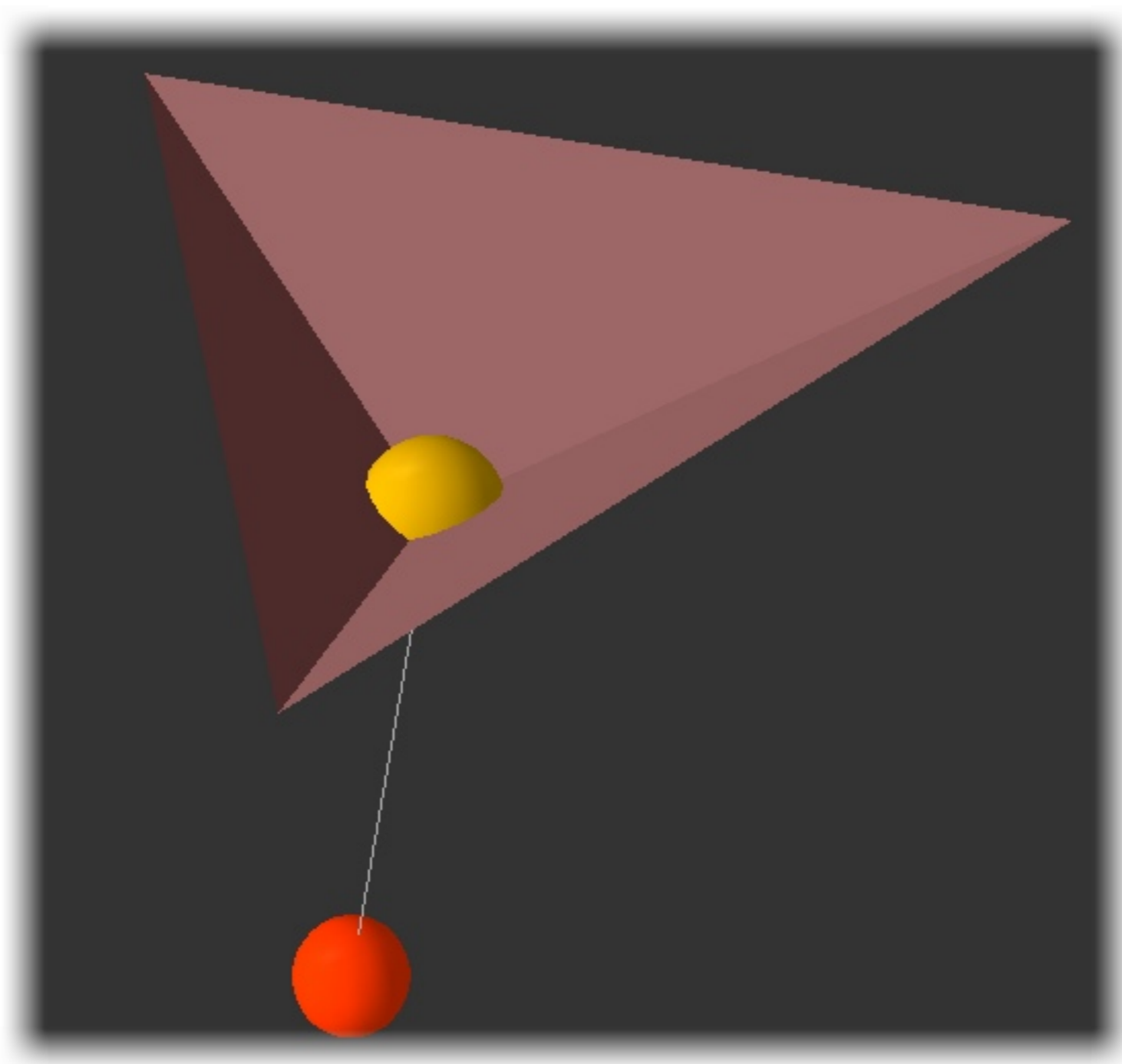


# Two Constraints



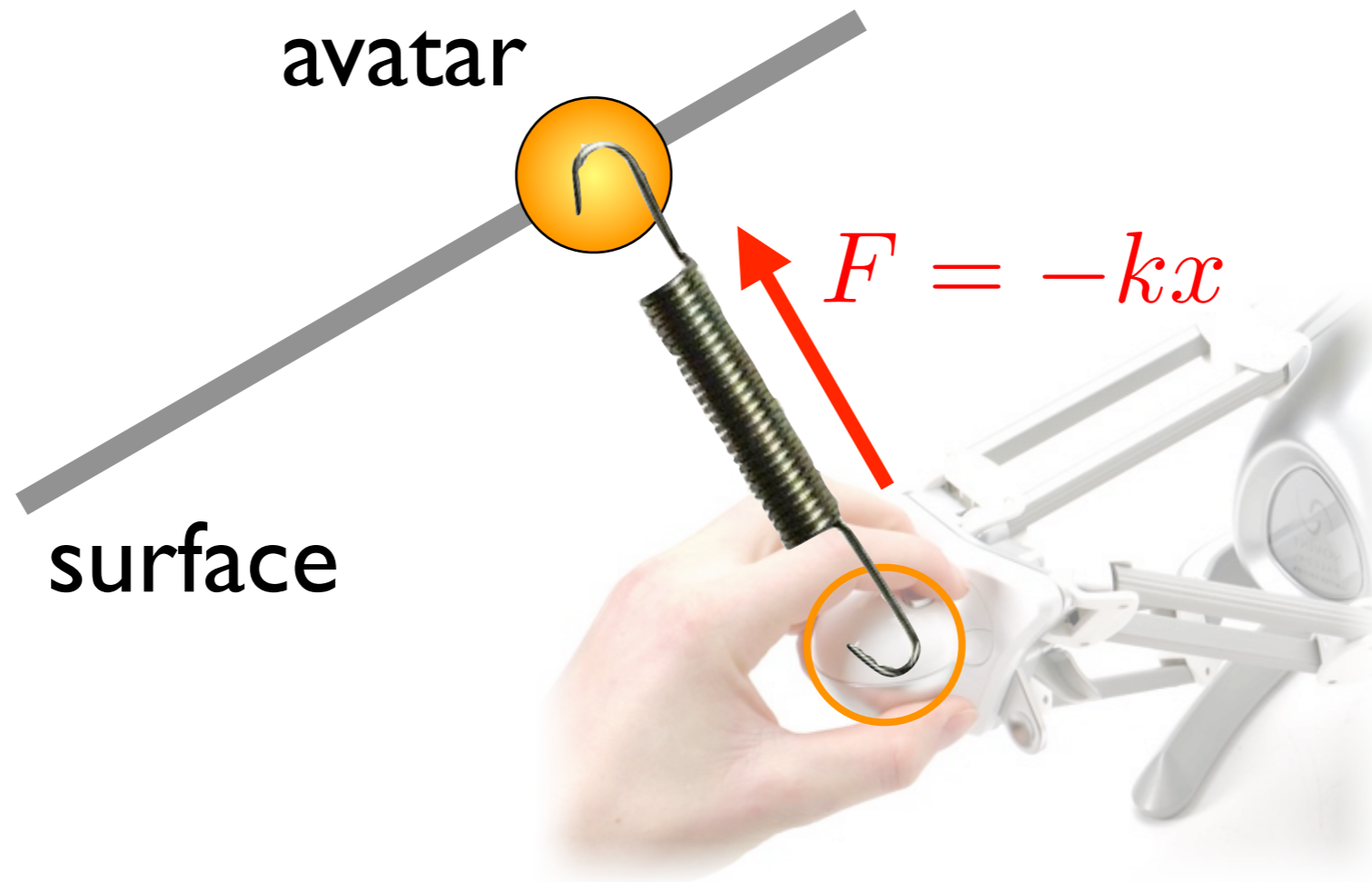
# Three Constraints



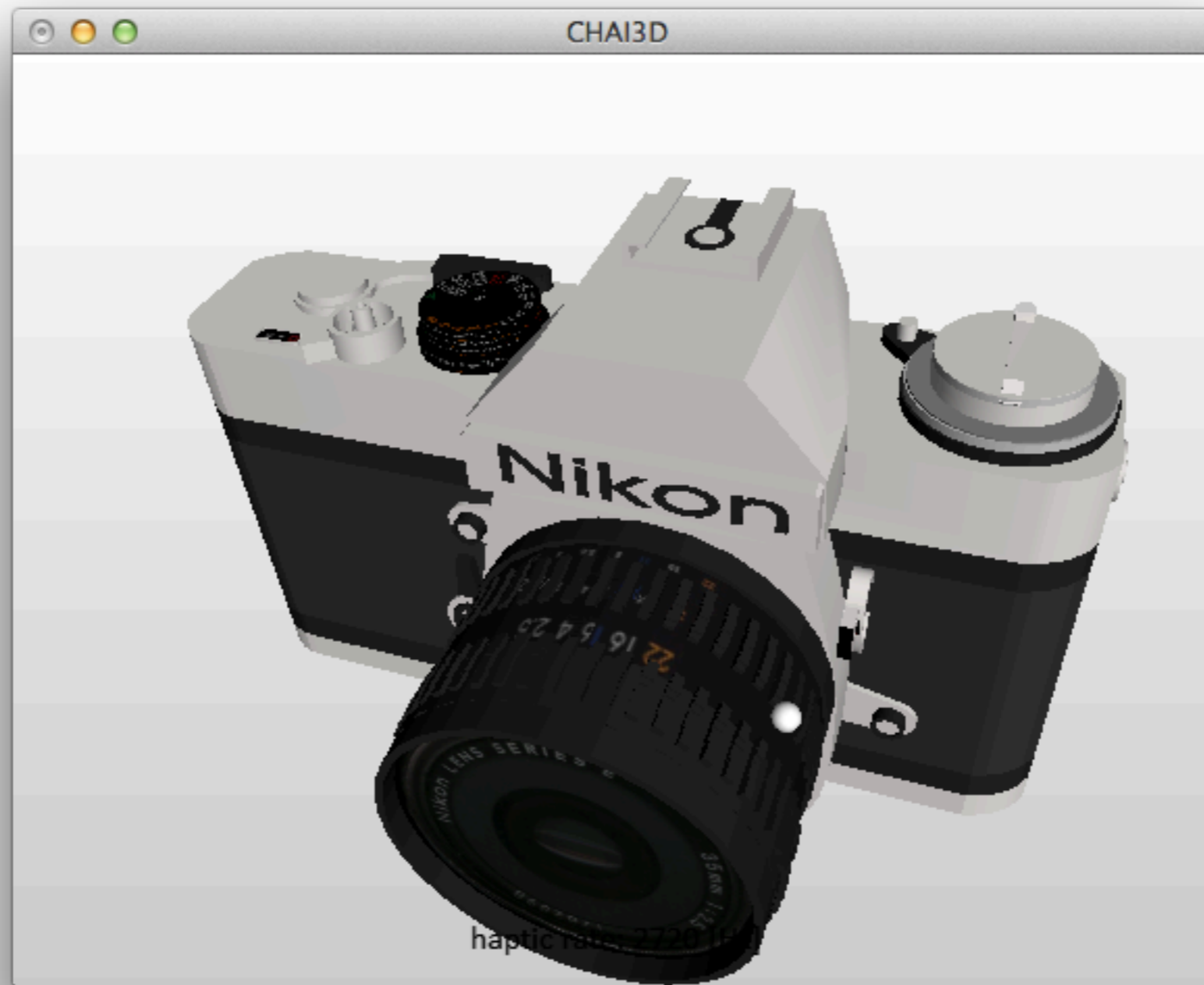


**What about cases with more than three constraints?**

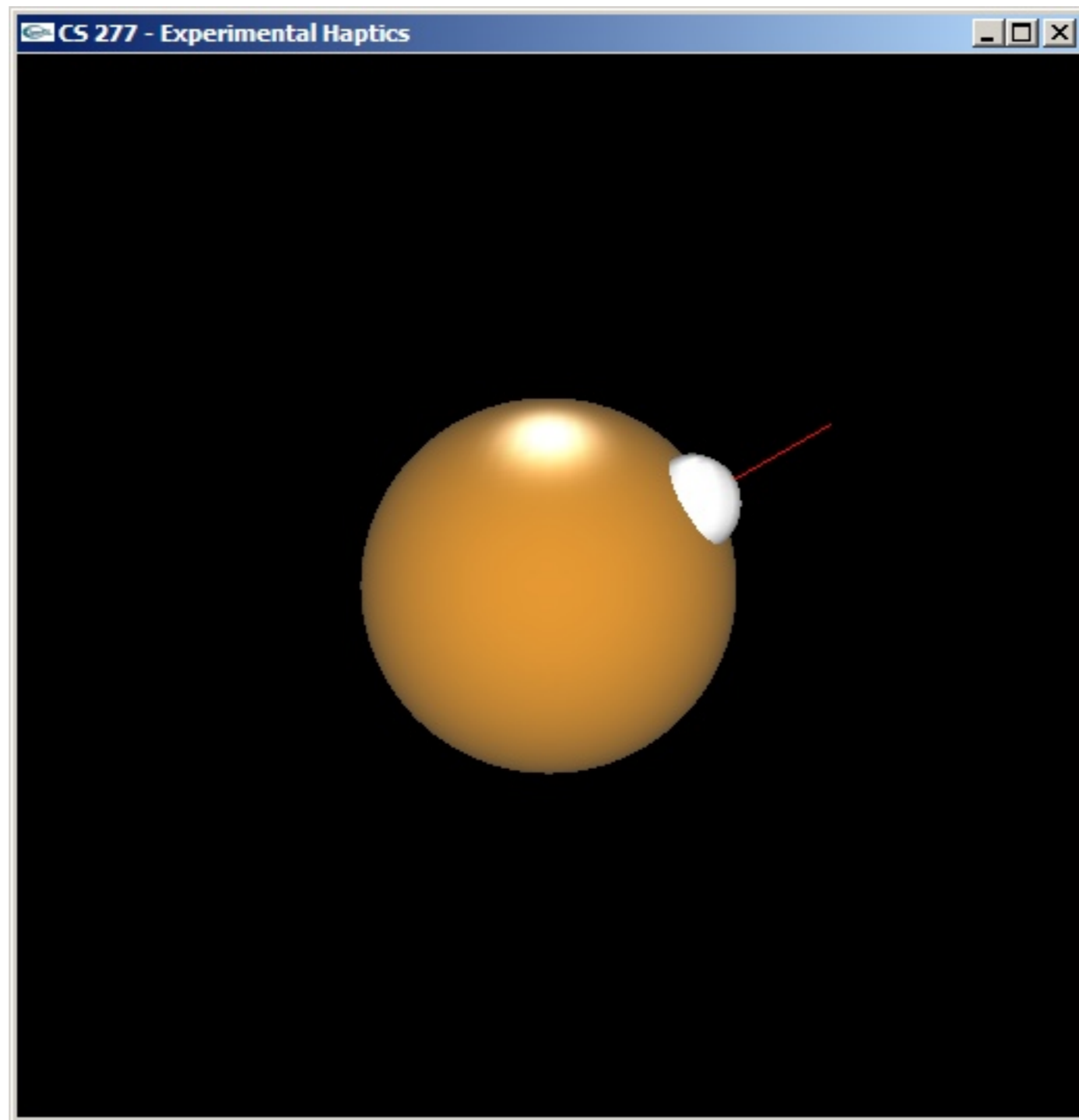
# Force Feedback

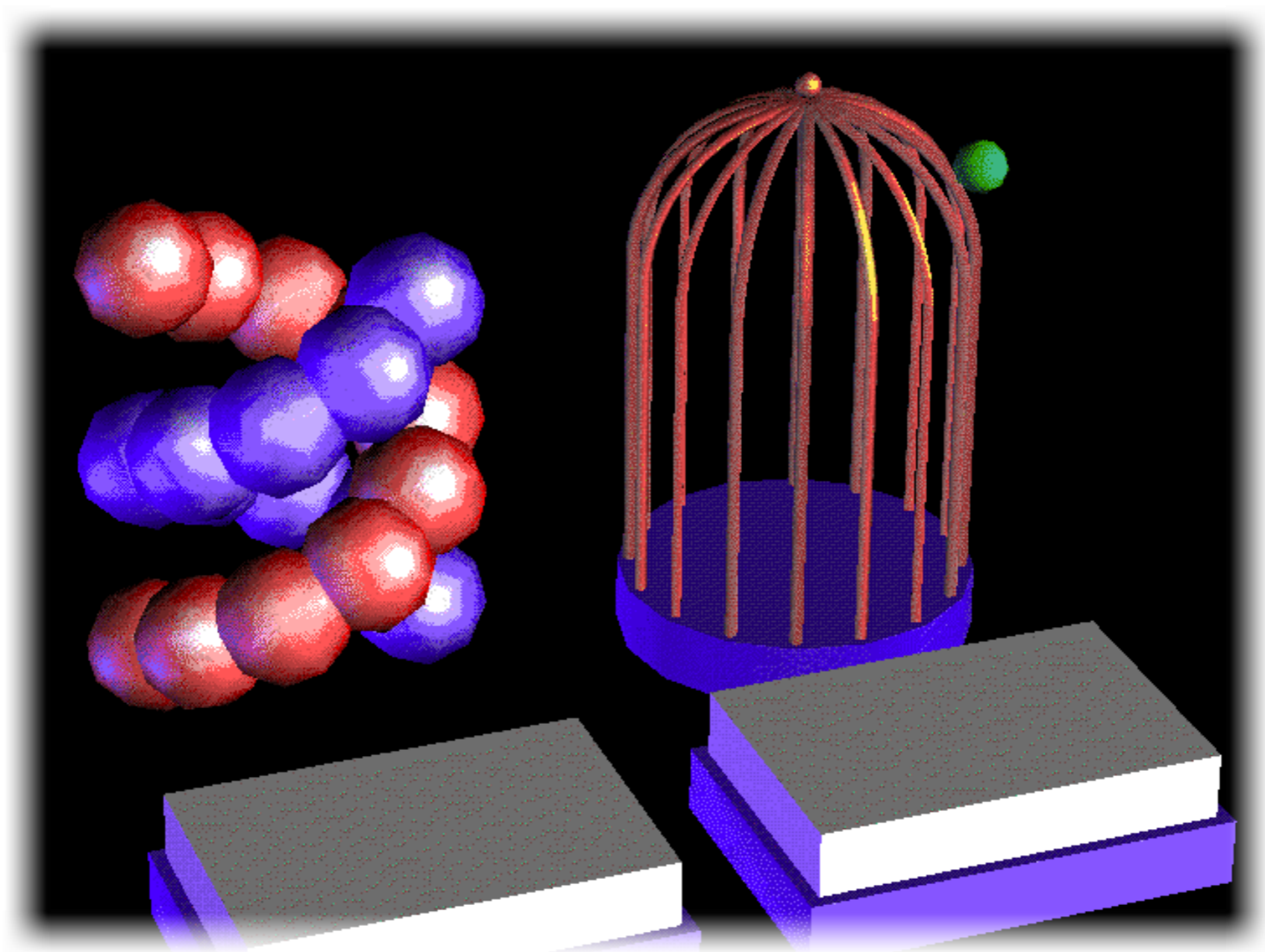


# God-Object Demo



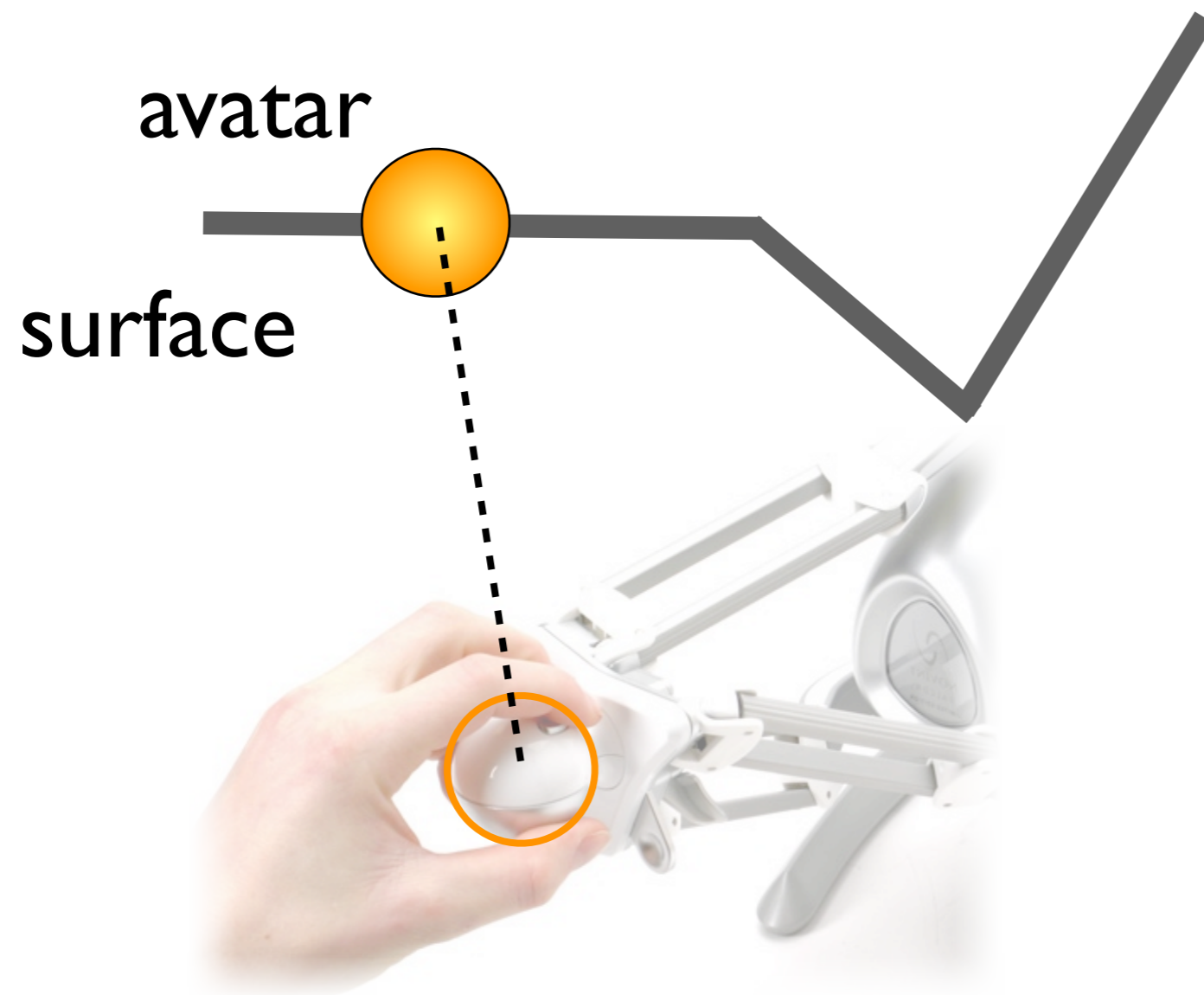
# The Sinking Avatar?





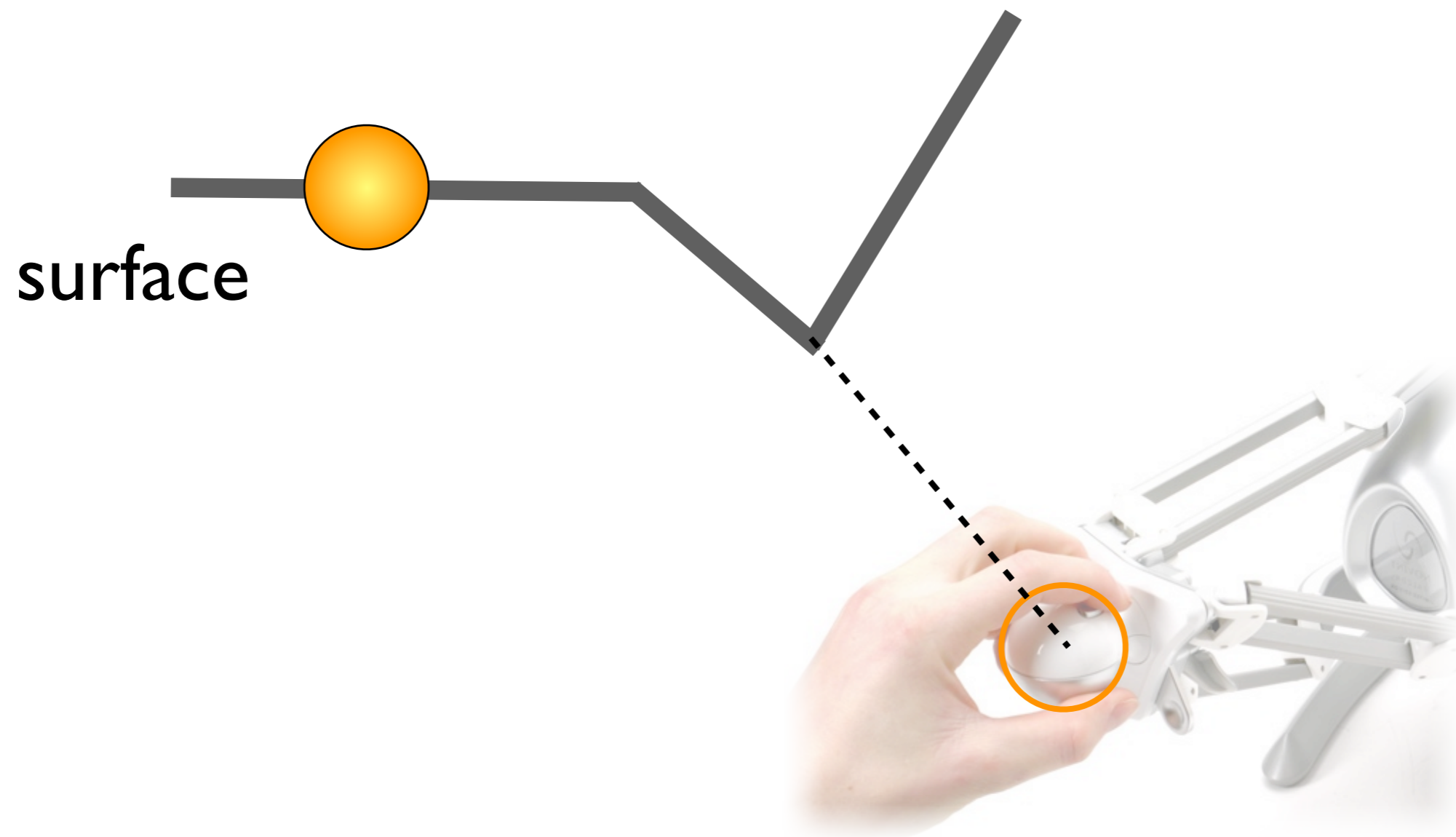
# Finger-Proxy Algorithm

# God-Object Algorithm

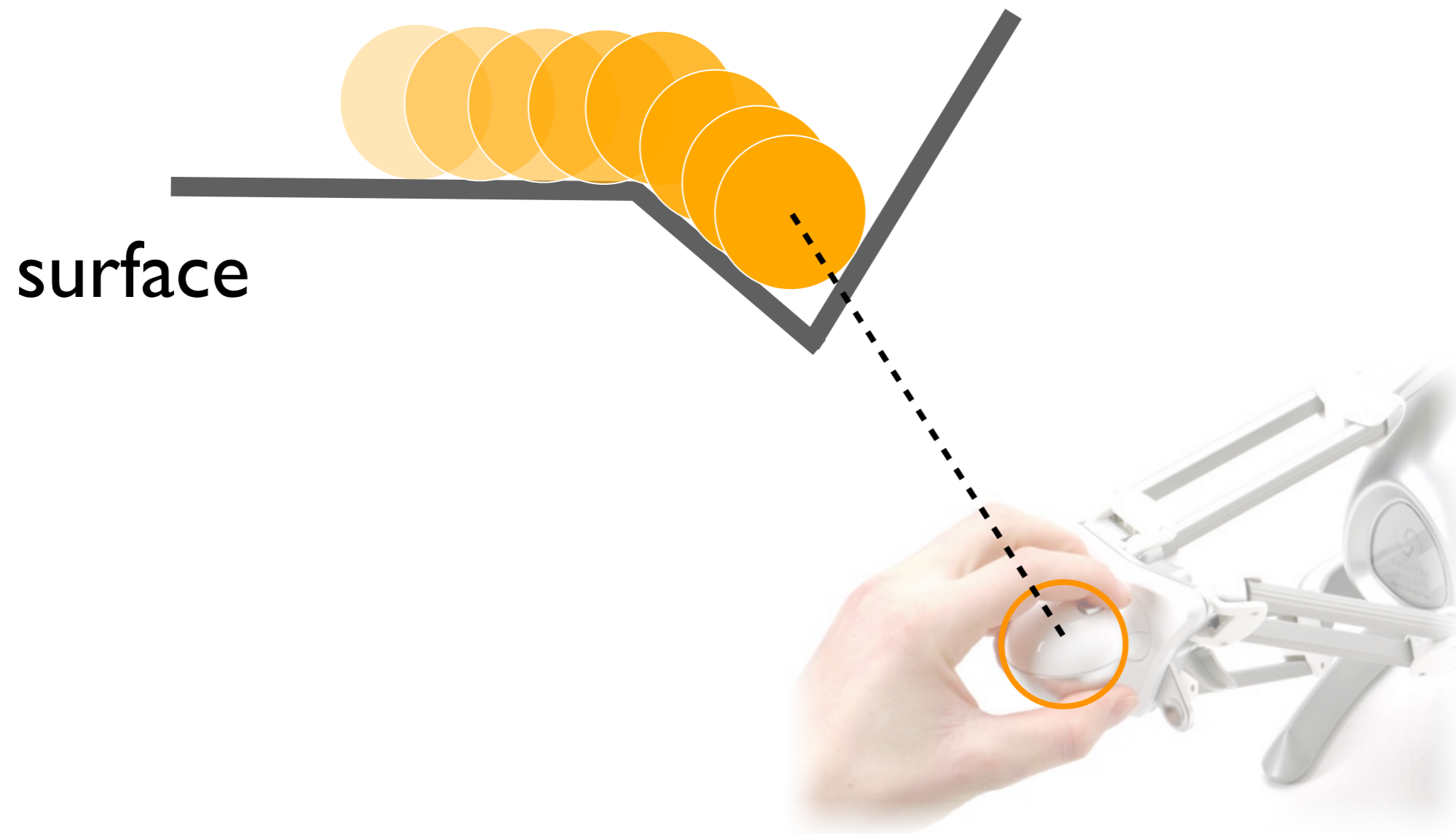




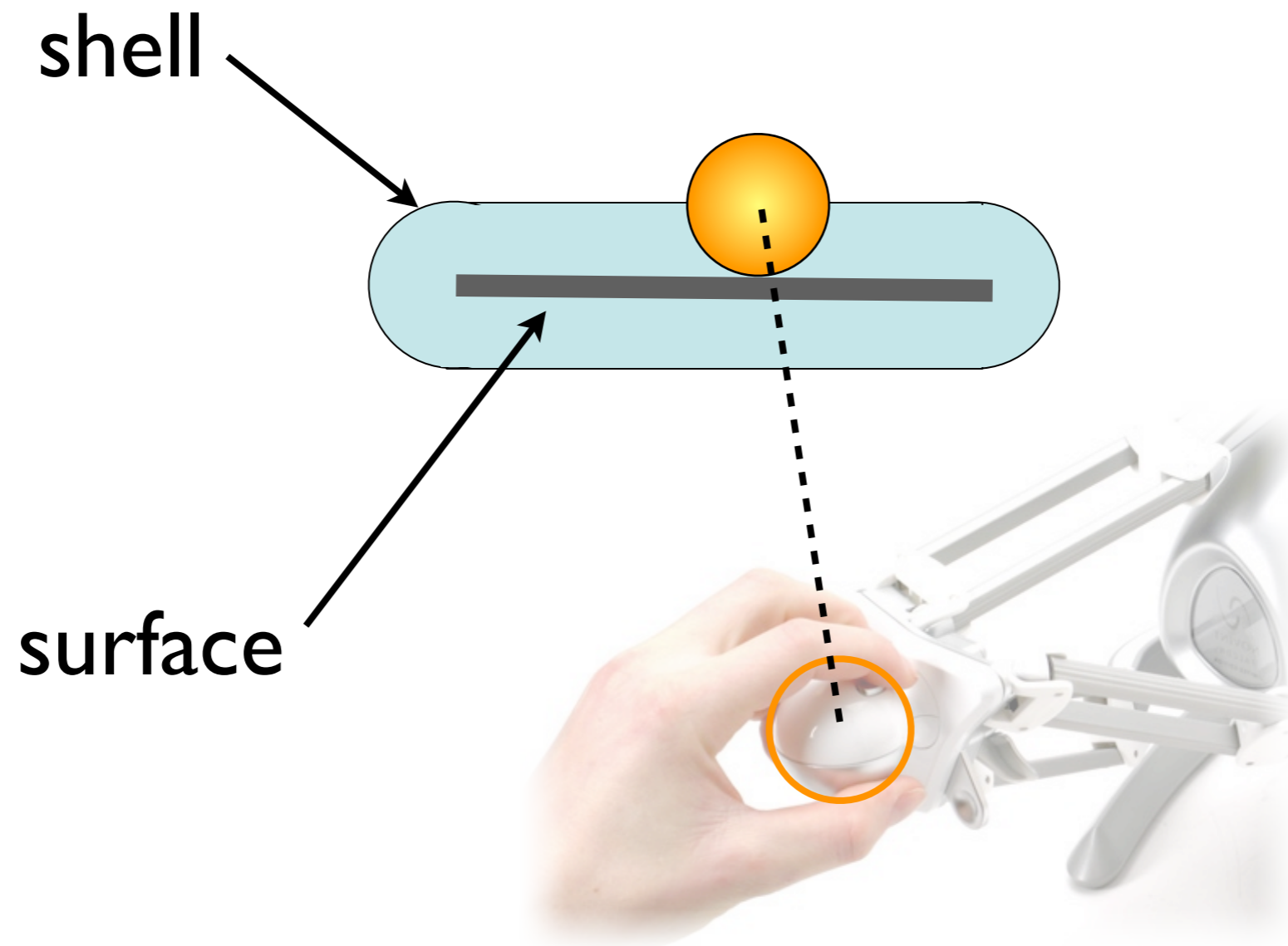
# God-Object Algorithm



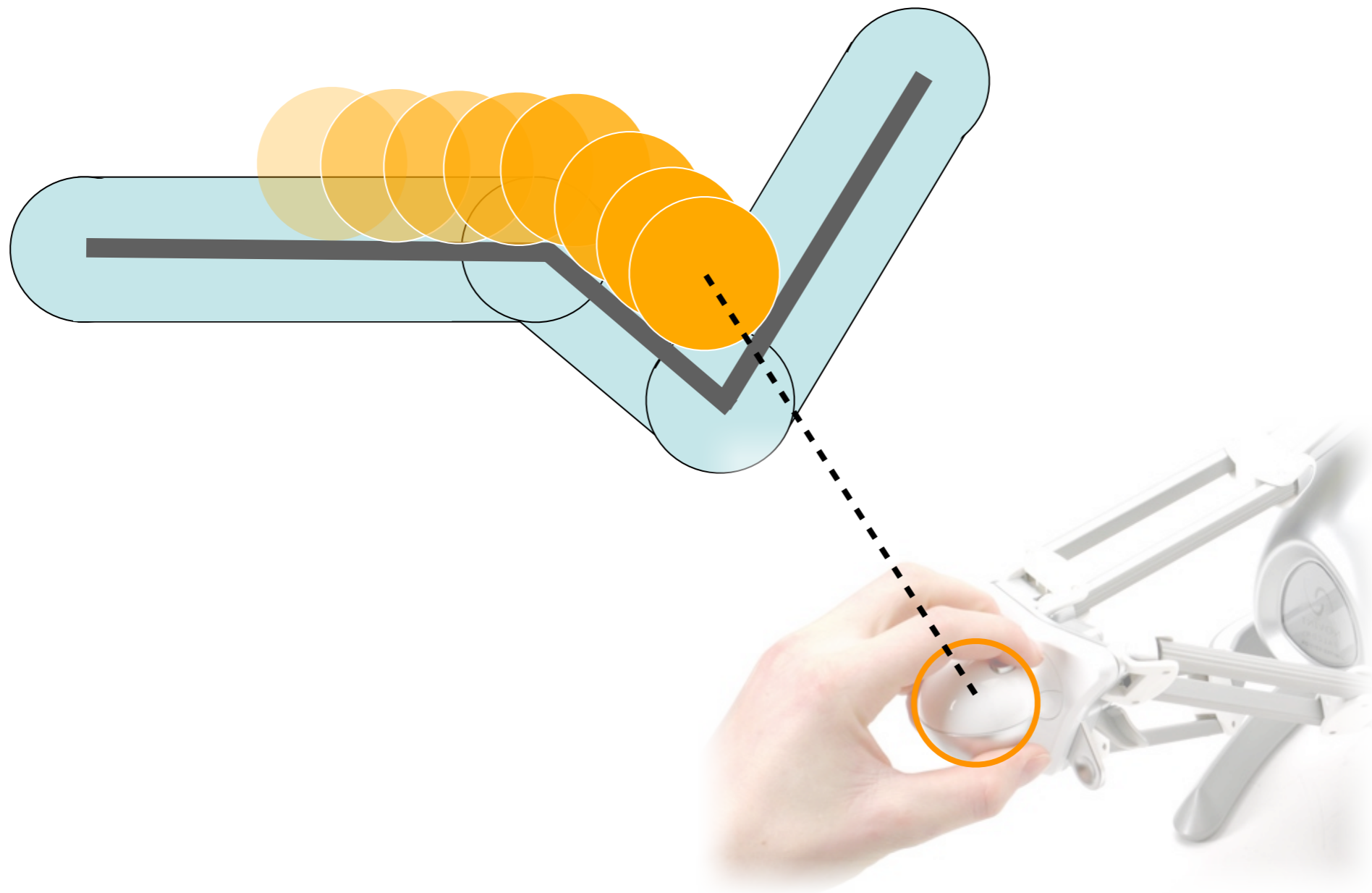
# Desired Behavior



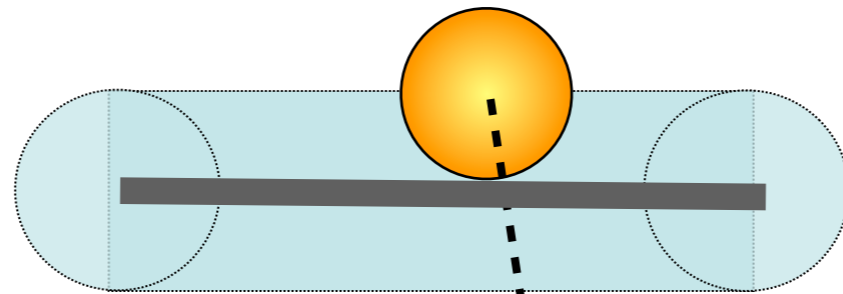
# Finger-Proxy Idea



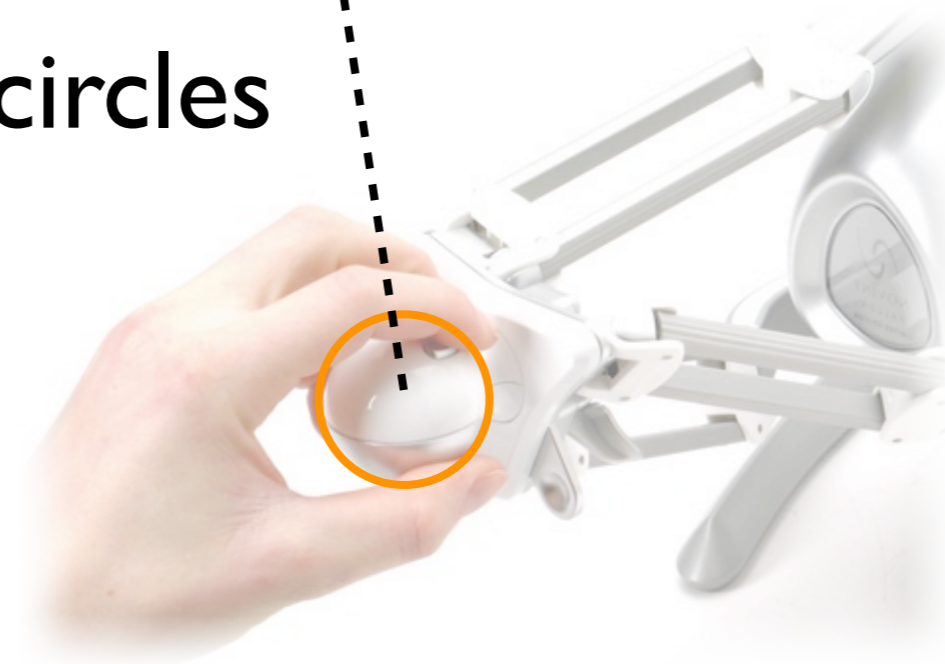
# Complex Shell



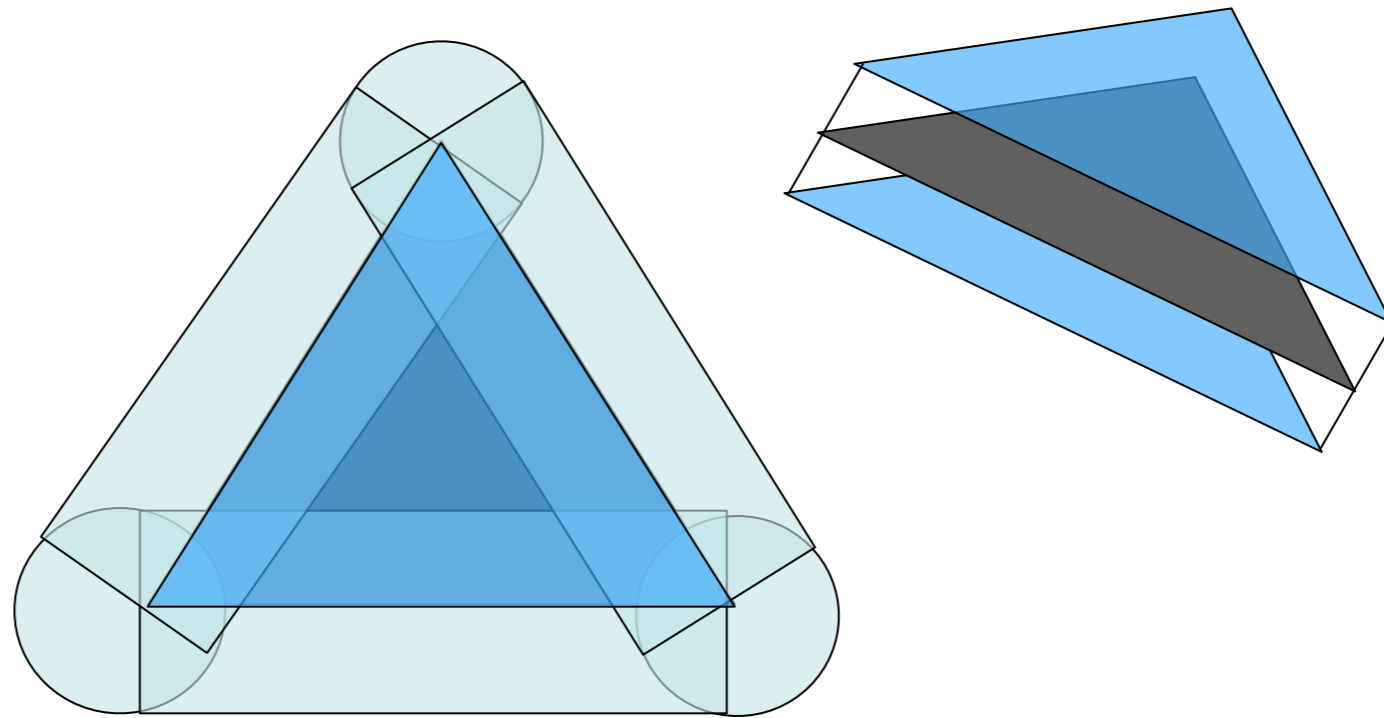
# In Two Dimensions



shell = 1 box + 2 circles

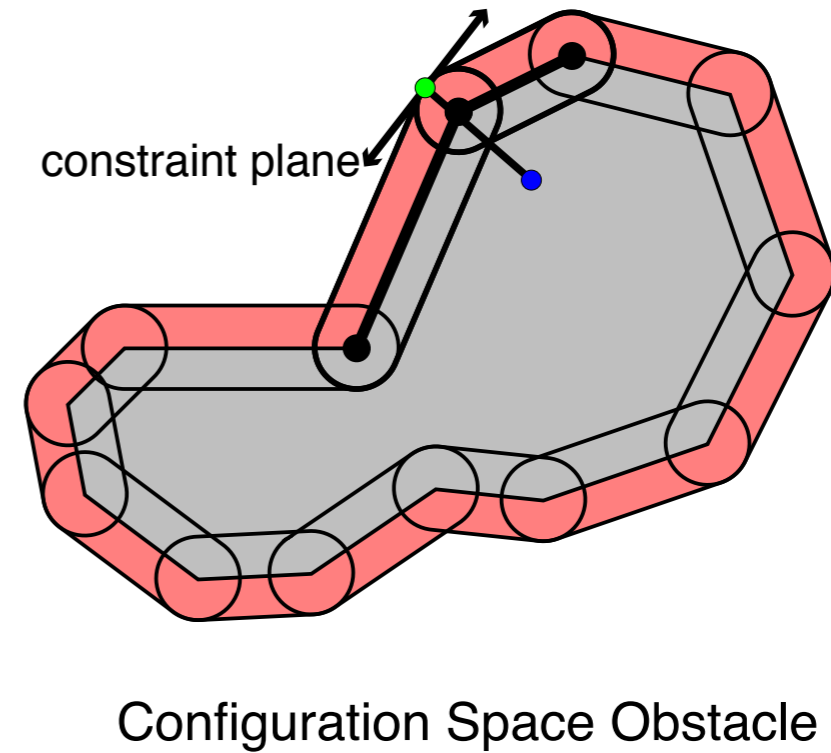
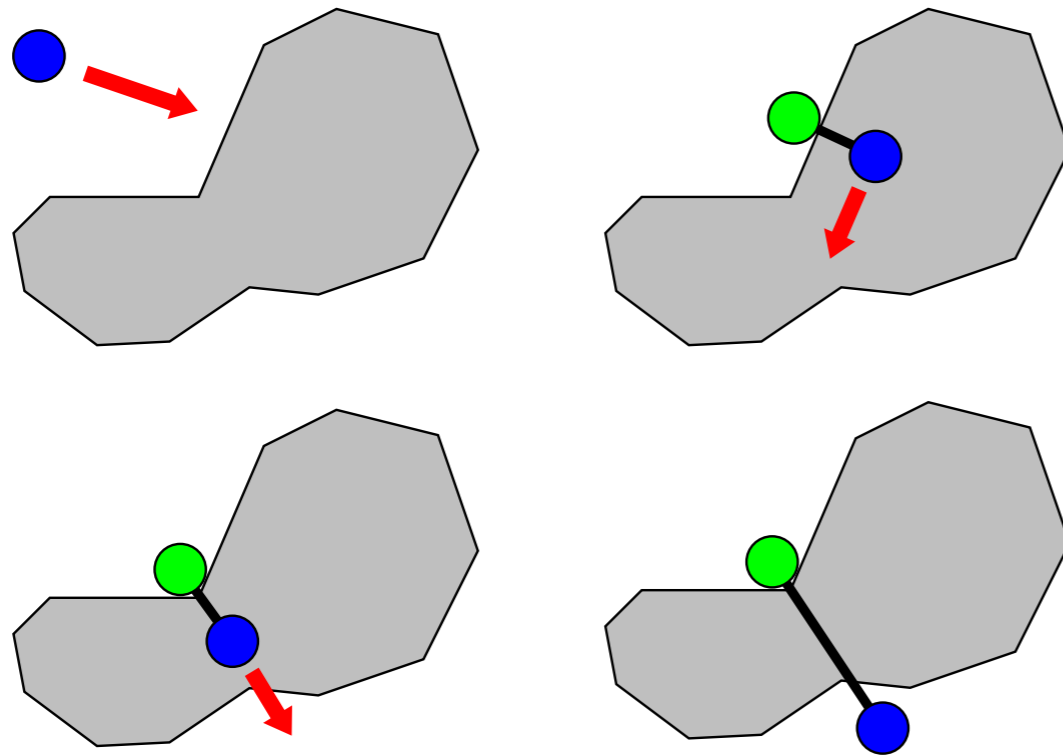


# In Three Dimensions

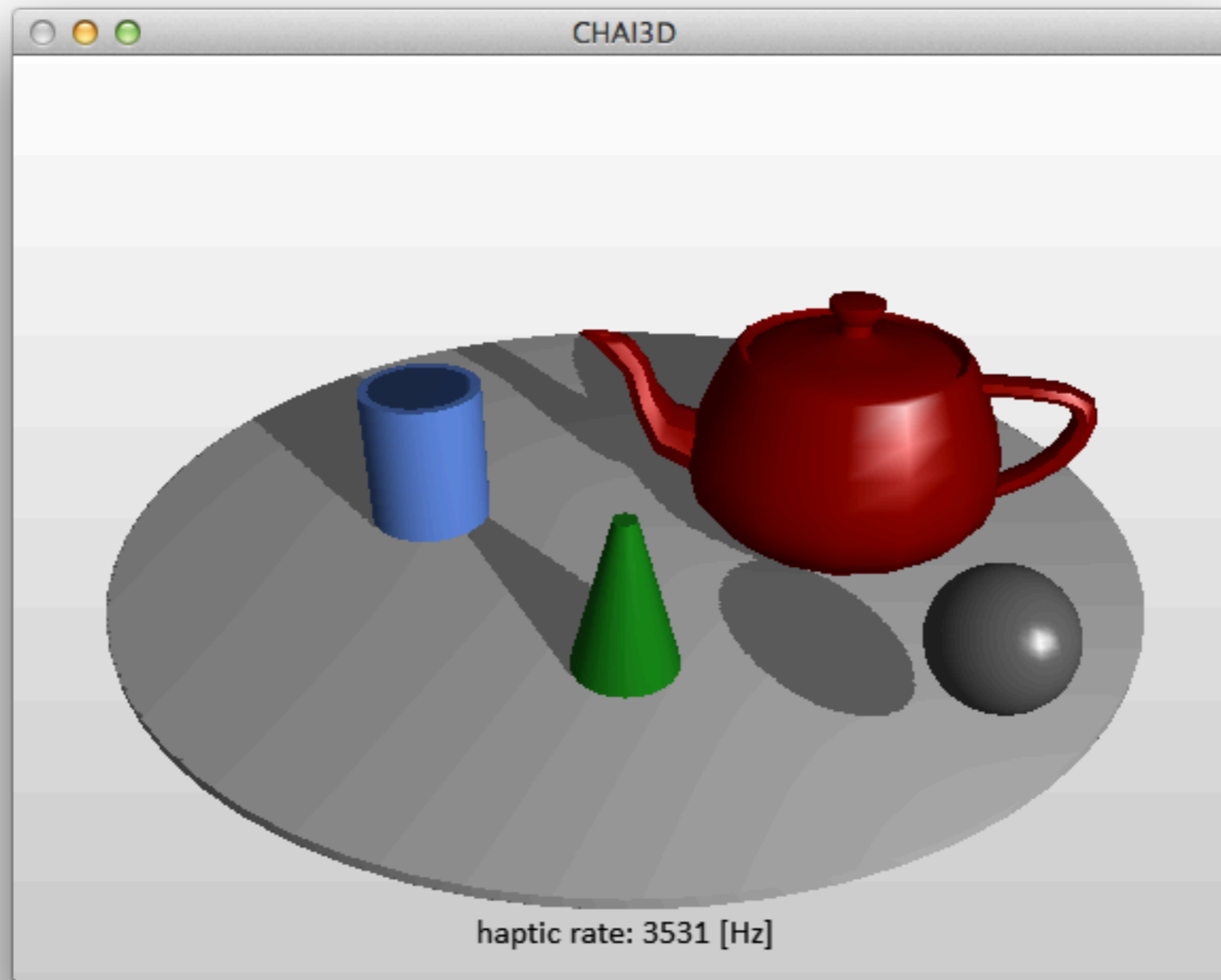


**shell = 3 spheres + 3 cylinders + 2 triangles**

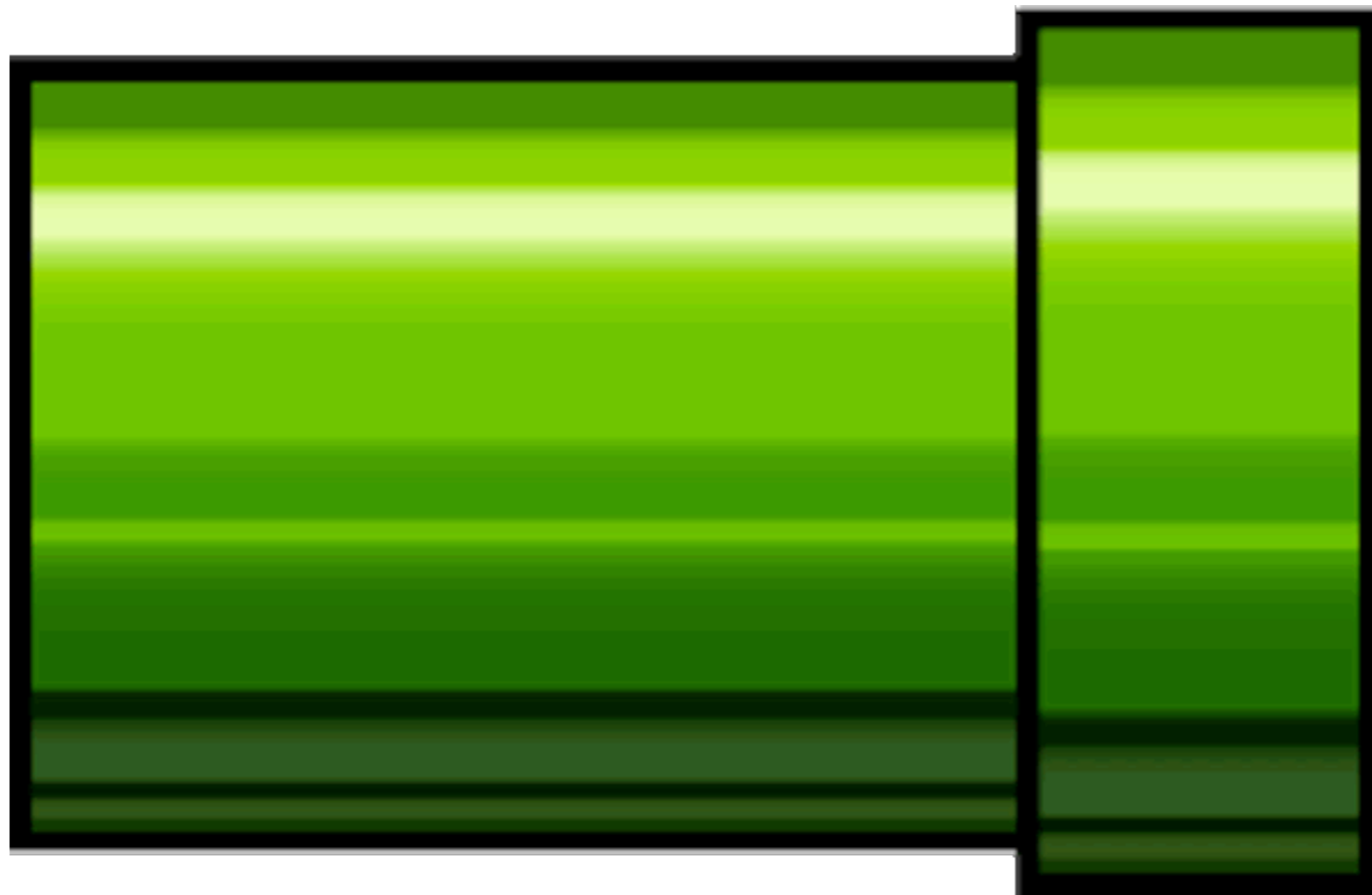
# Finger-Proxy Algorithm



# Finger-Proxy Demo

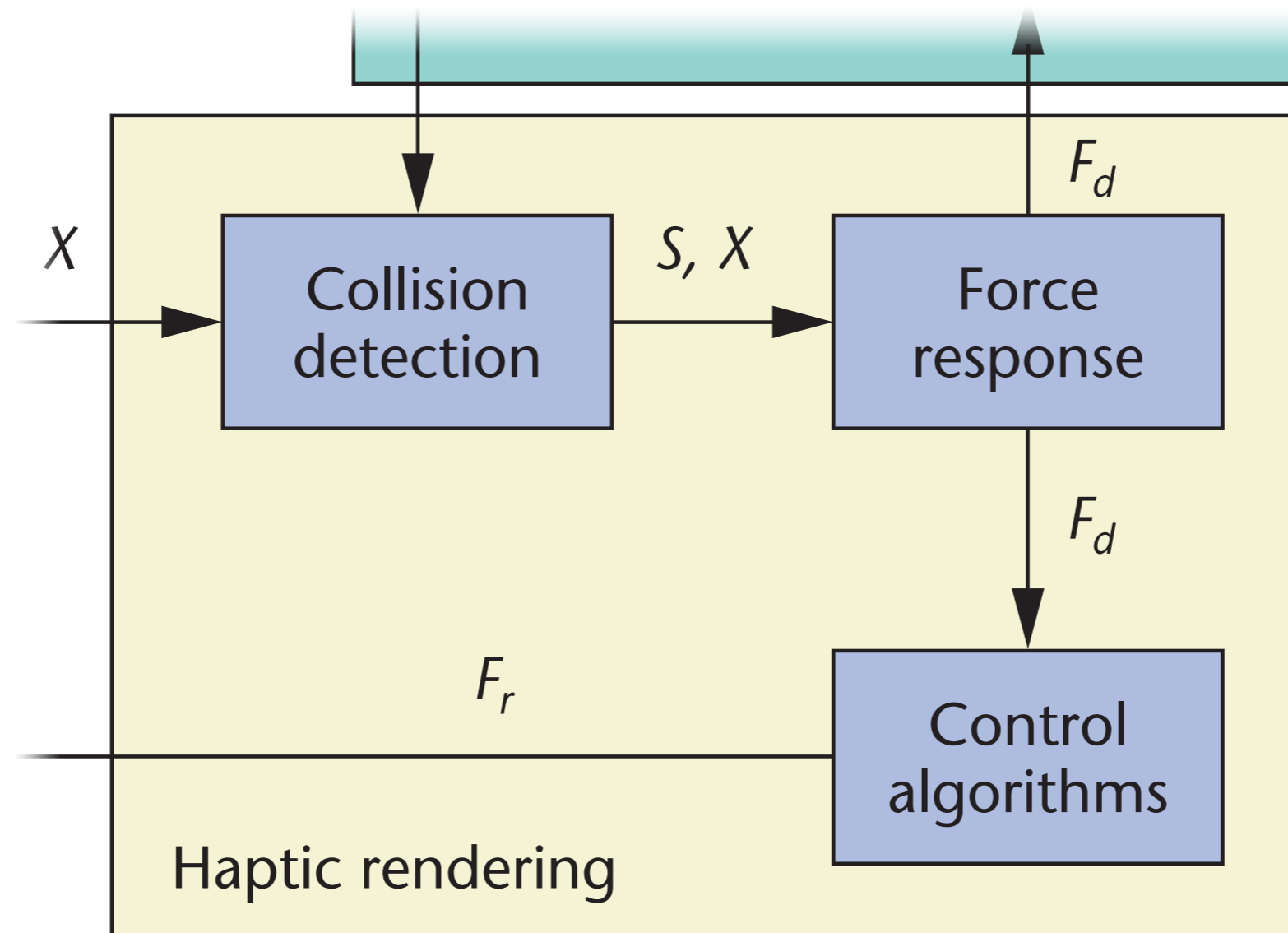




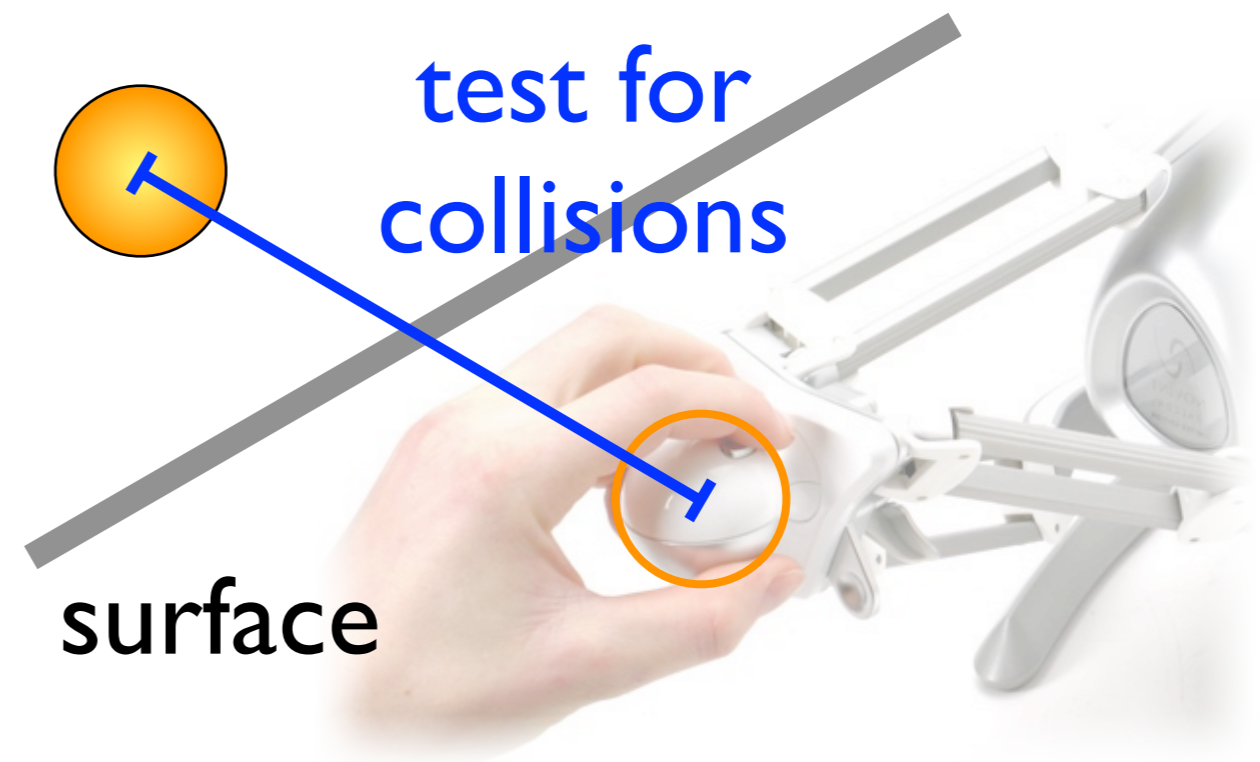
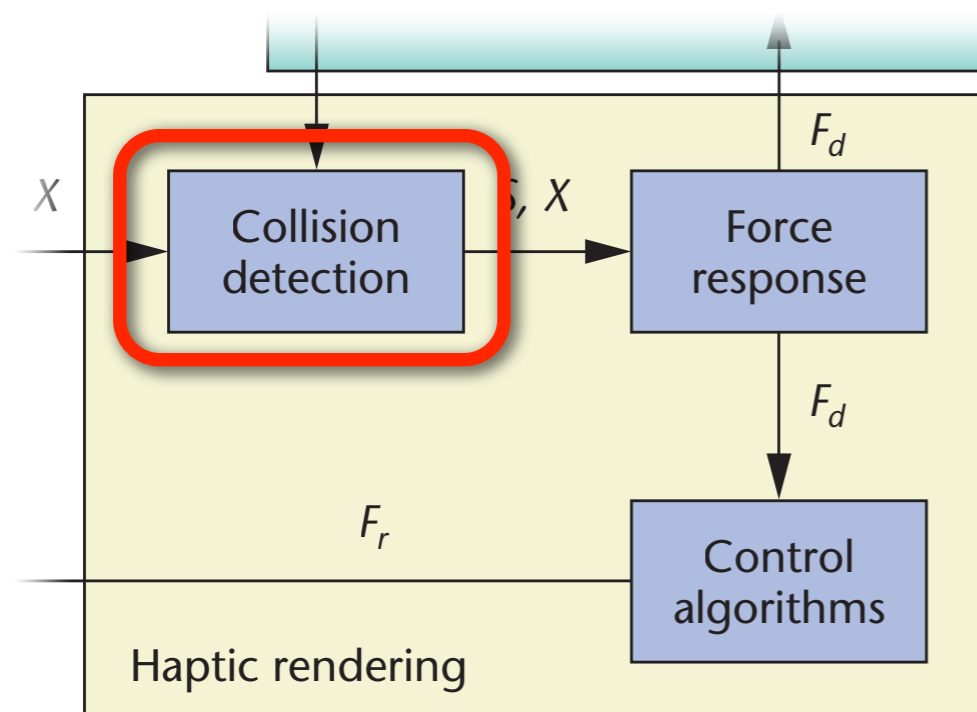


# Revisiting the Pipeline

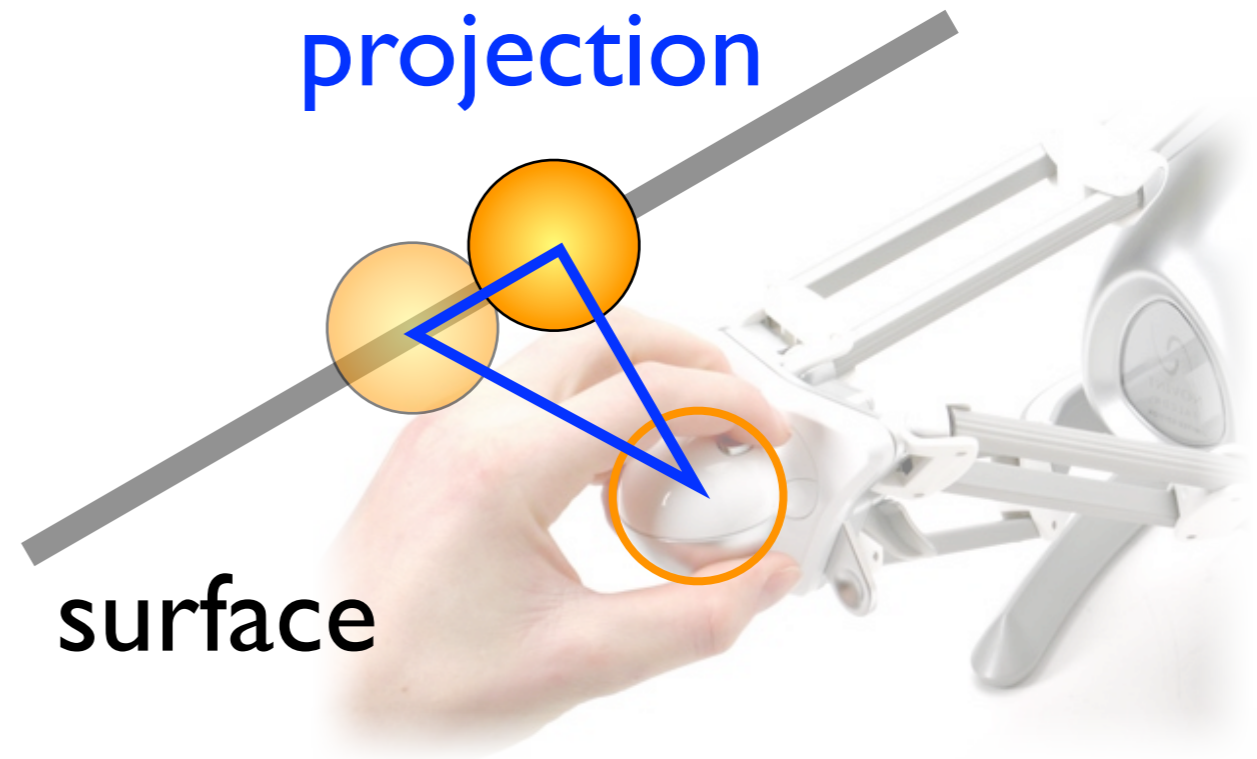
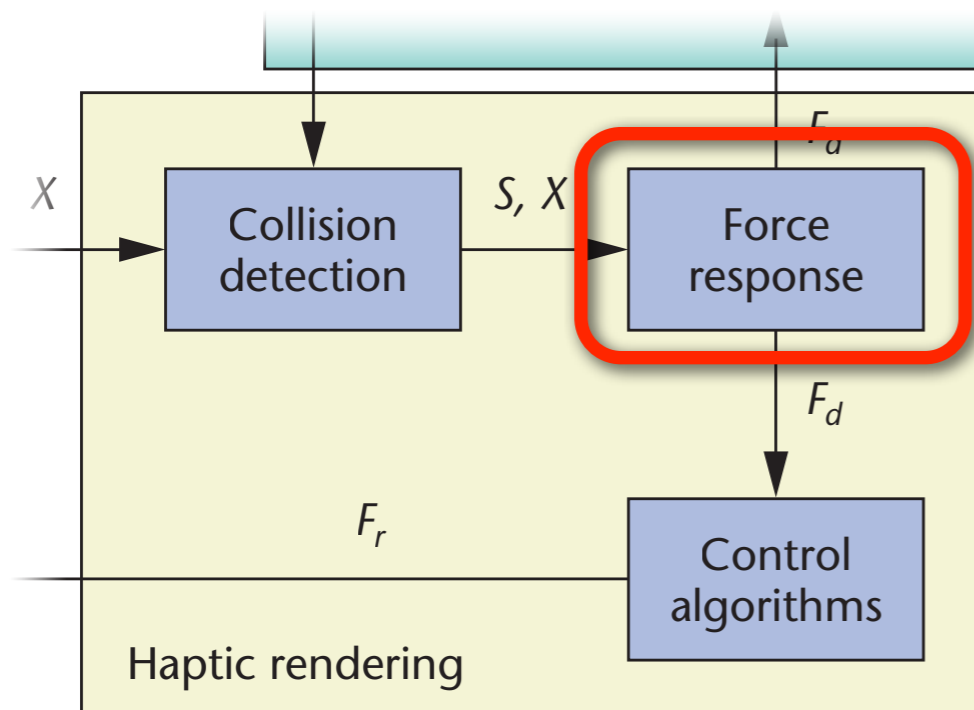
# Haptic Rendering



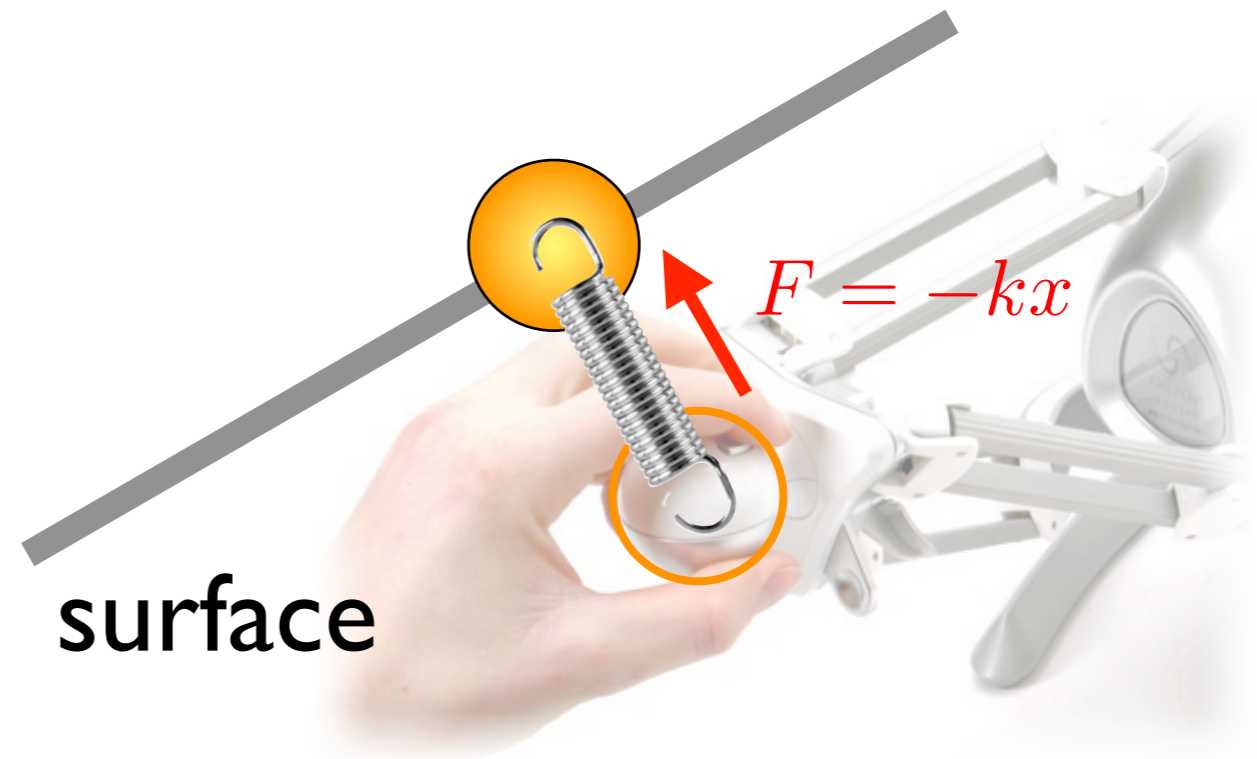
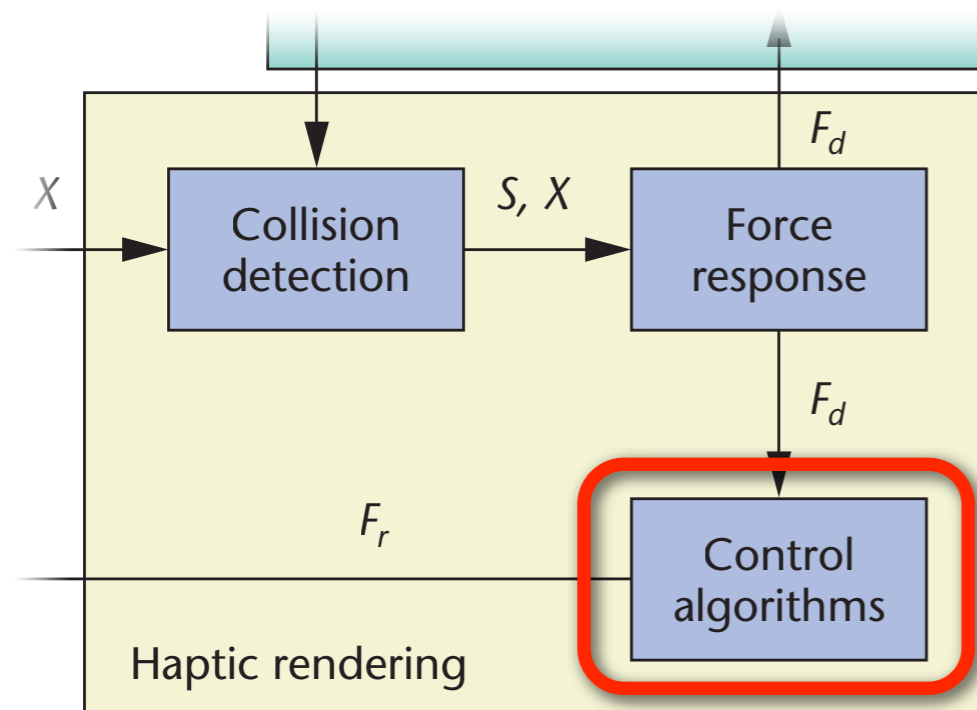
# Collision Detection



# Simulation / Optimization



# Force Feedback



# Summary

- ▶ Time is of the essence!
- ▶ Limitations with potential fields
- ▶ God-object algorithm
- ▶ Finger-proxy algorithm
- ▶ Revisit of the haptic rendering pipeline