CS277 - Experimental Haptics
Lecture 4

Implicit Surfaces & Friction

Outline

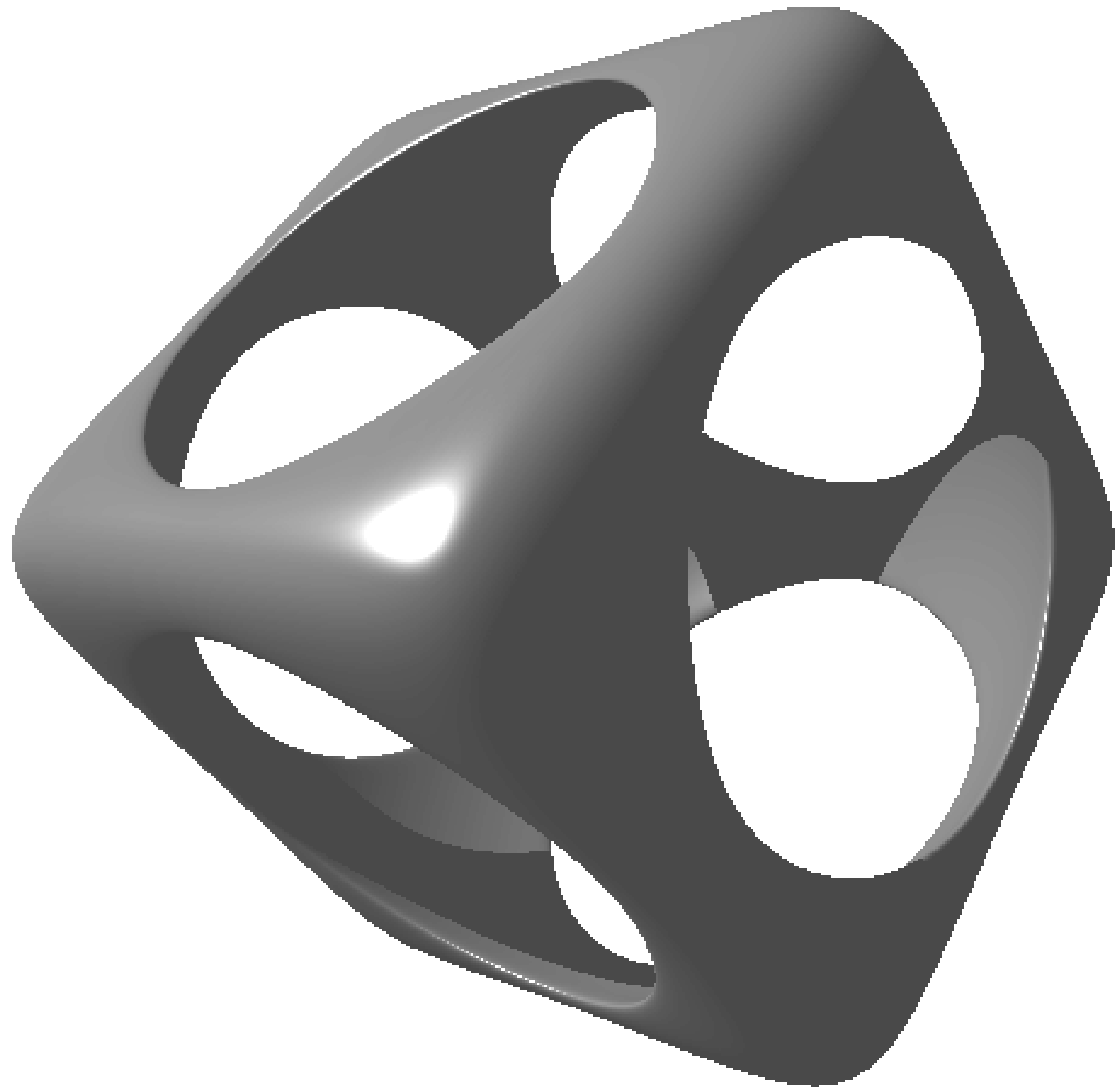
- Announcements
- Implicit surface rendering algorithm
- Rendering friction
- Rendering volumetric data (if I have time)

Office Hours

- Democracy worked! (?)
  - Mondays, 4-6pm, Clark E100 (Sonny)
  - Tuesdays, 4-6pm, Gates AI lab (François)
  - Thursdays, 2-4pm, Clark E100 (Sonny)
  - Fridays, 2-4pm, Gates AI lab (François)
- Will post times to course page and piazza

Haptic Devices

- We are now in stock!
- Pick up after class today, or during Thursday/Friday office hours
- Clark Center E1.3 (Salisbury Robotics)



Implicit Surfaces

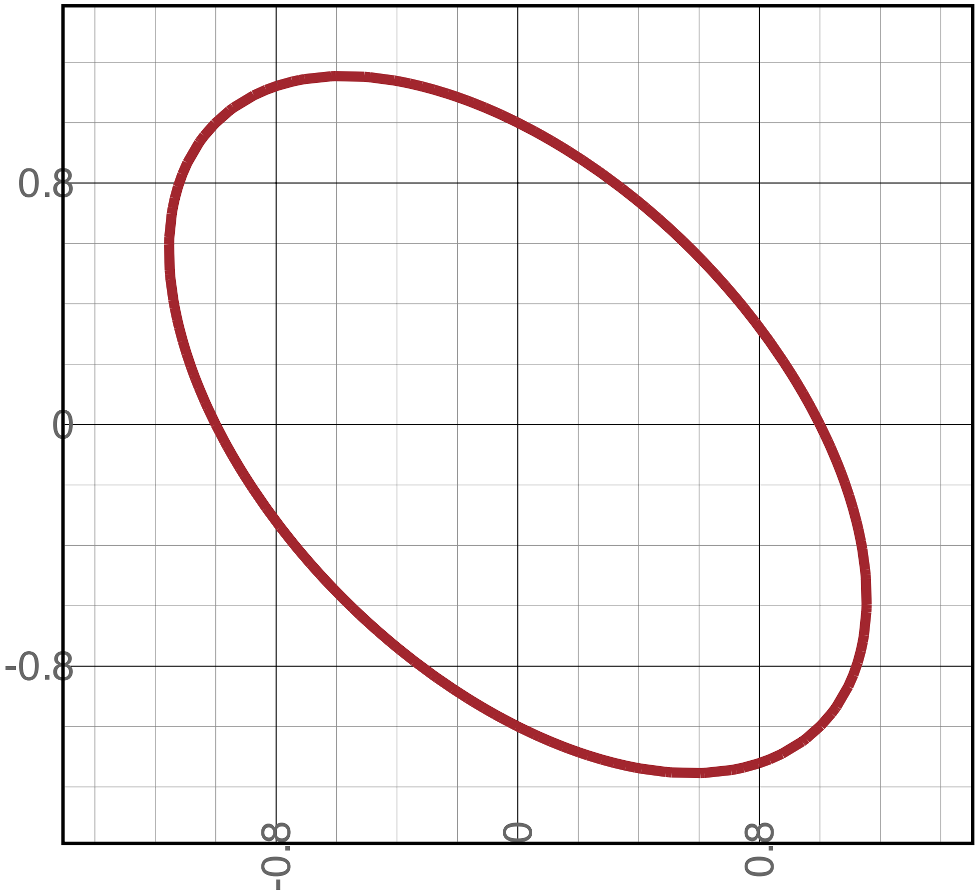
[from K. Salisbury & C. Tarr, *Proc. ASME Haptics Symposium*, 1997.]

Rendering Implicit Surfaces

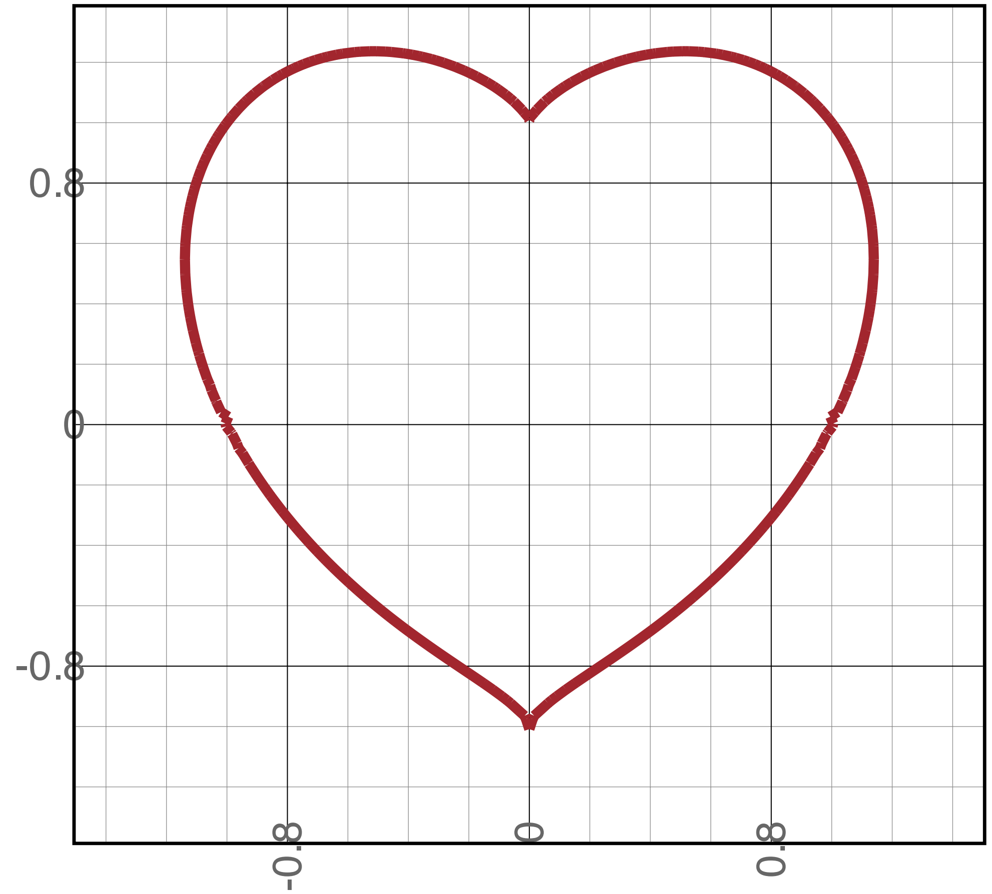
- A surface defined by an implicit equation:
  - $S(x, y, z) = 0$
- Can be rendered using the same proxy-based algorithm.

Review: Implicit Surfaces

- "Graph" the equation, as you learned how to do in high school...

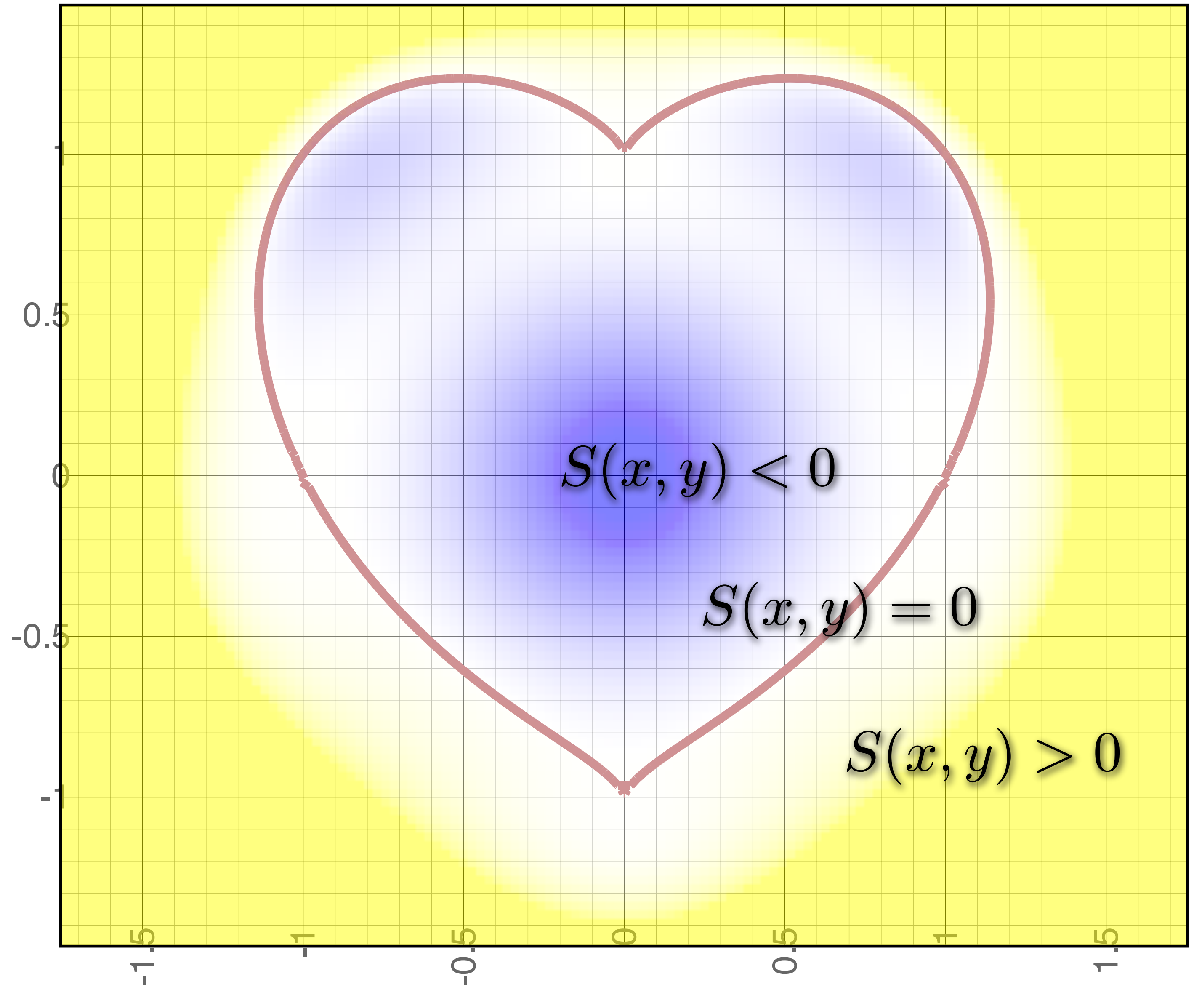


$$x^2 + xy + y^2 = 0$$

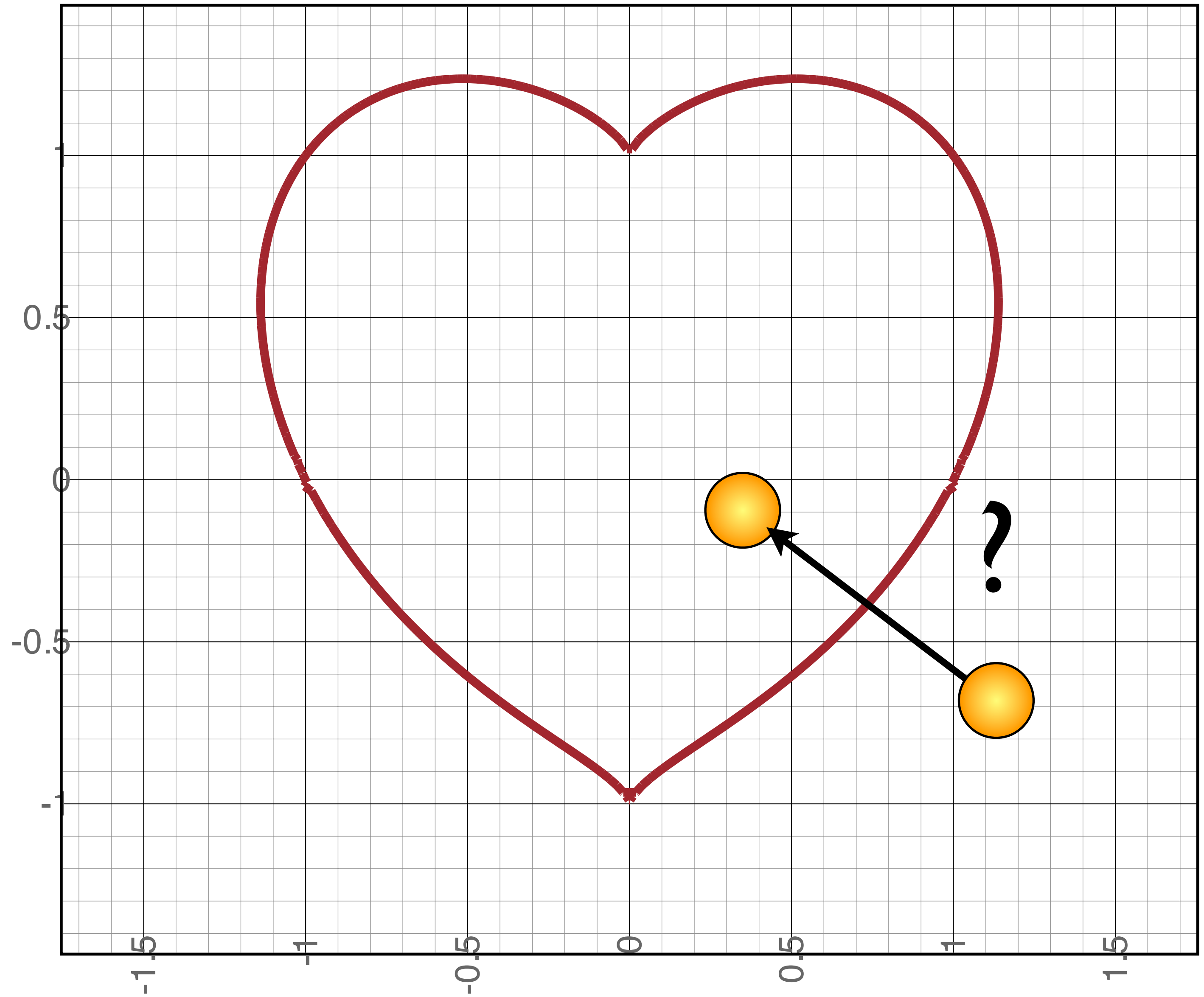


$$(x^2 + y^2 - 1)^3 - x^2y^3 = 0$$

The Landscape



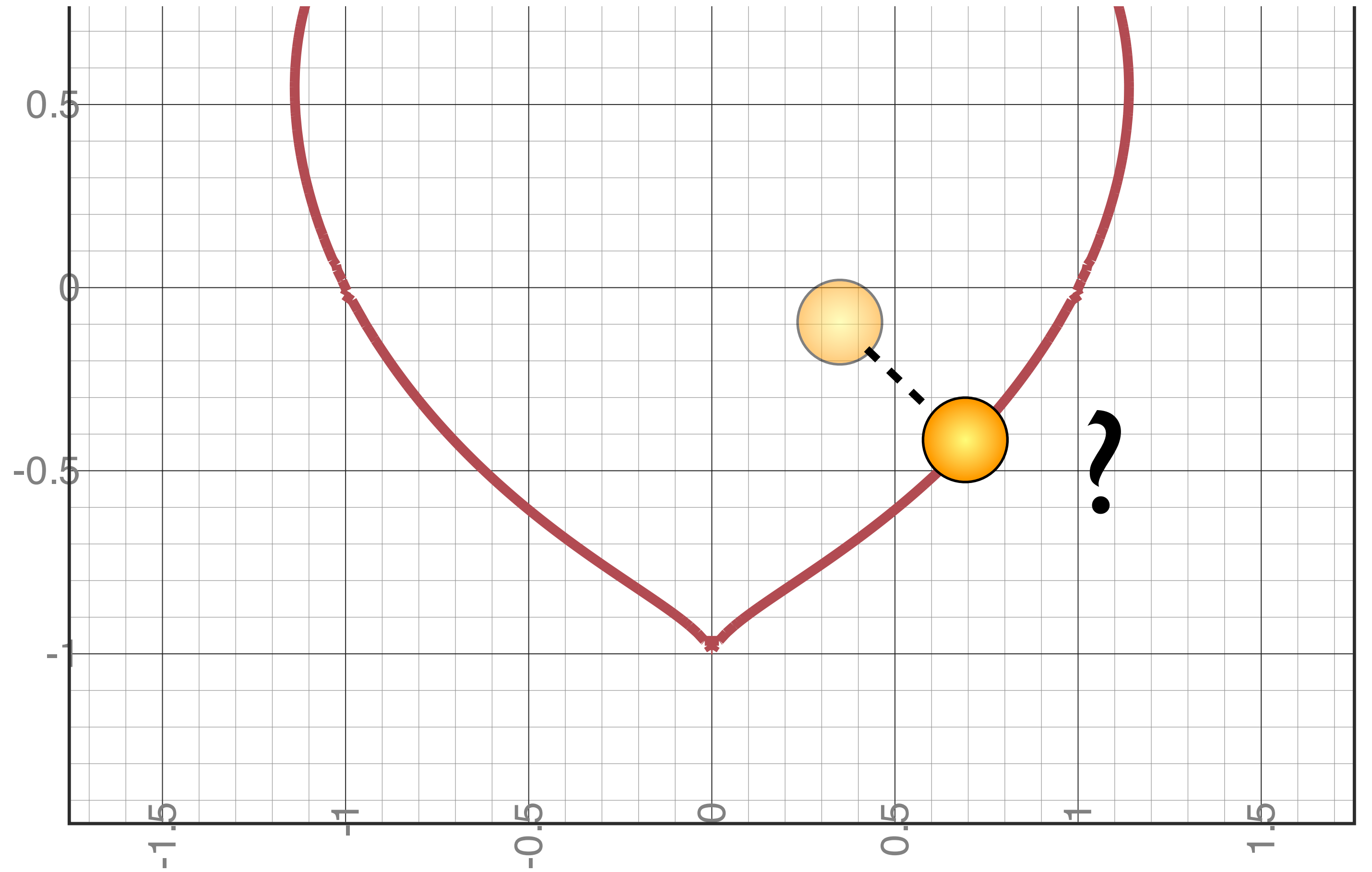
Step 1: Detecting Collision



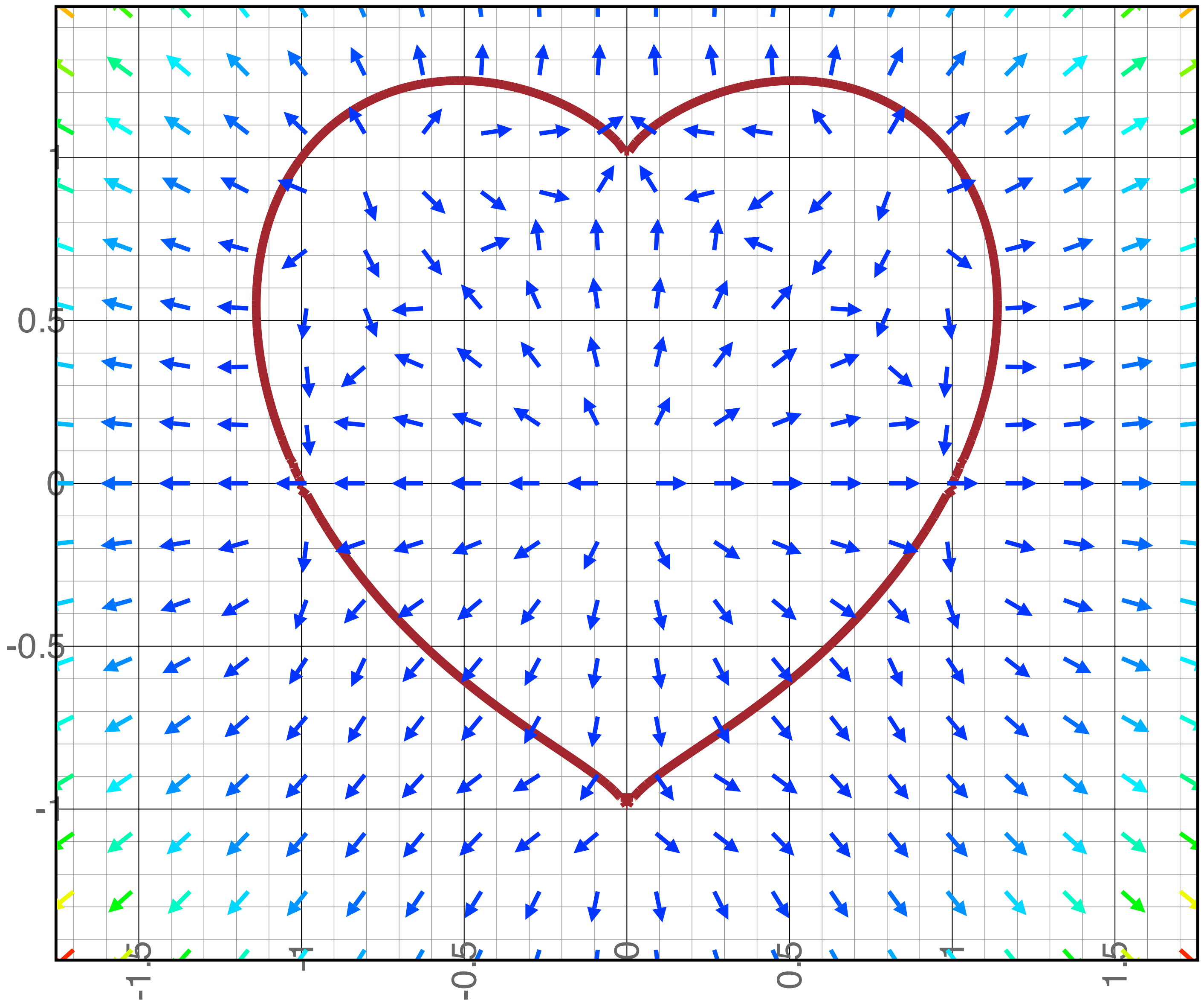
How do we detect the first contact with the object?

Step 2: Finding a Surface Point

- Once contact is made, we need to keep track of a point on the surface
- First, how do we find this point?

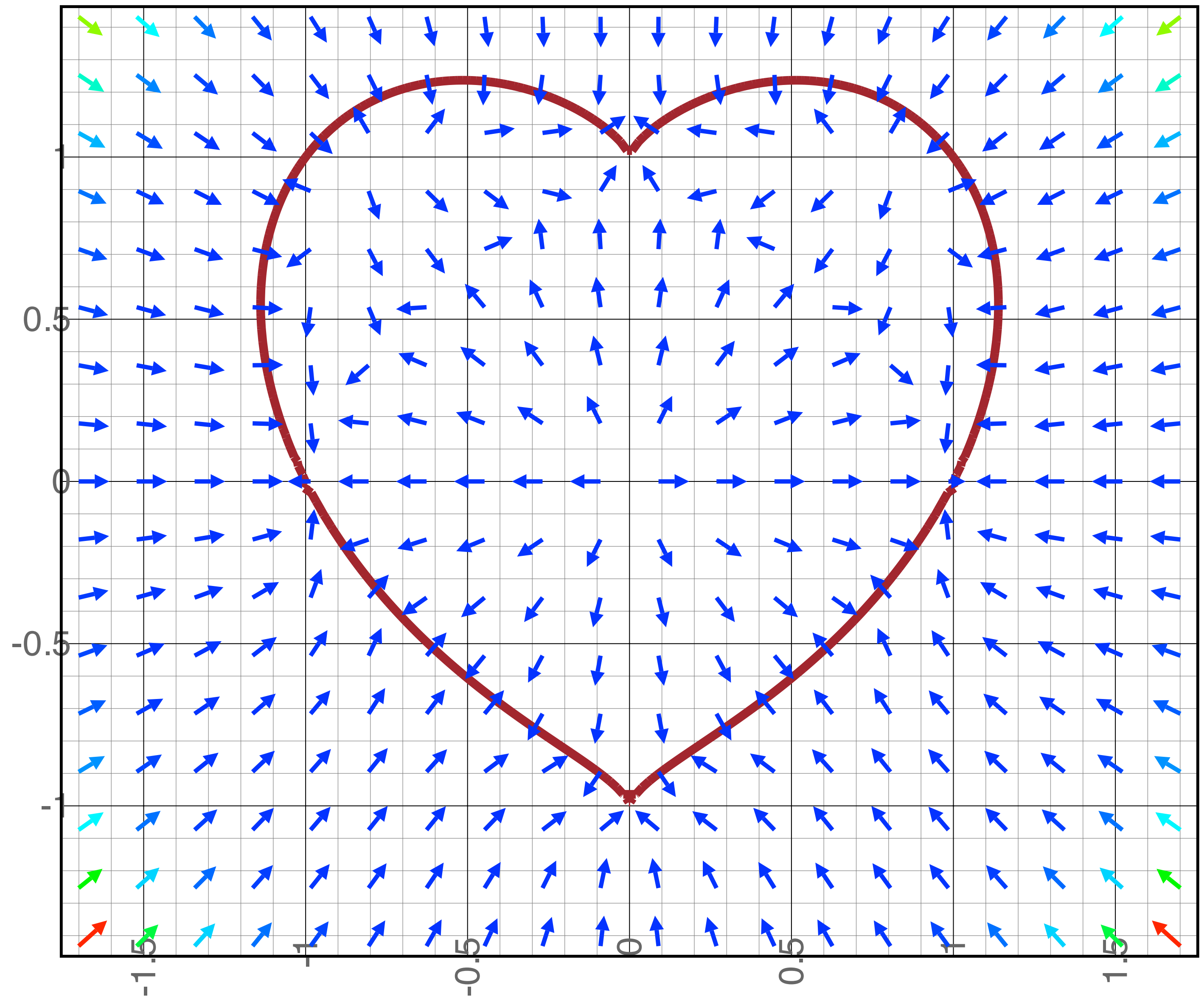


The Gradient



$$\nabla S(x,y) = \begin{pmatrix} \frac{\partial S}{\partial x} \\ \frac{\partial S}{\partial y} \end{pmatrix}$$

Direction to the Surface

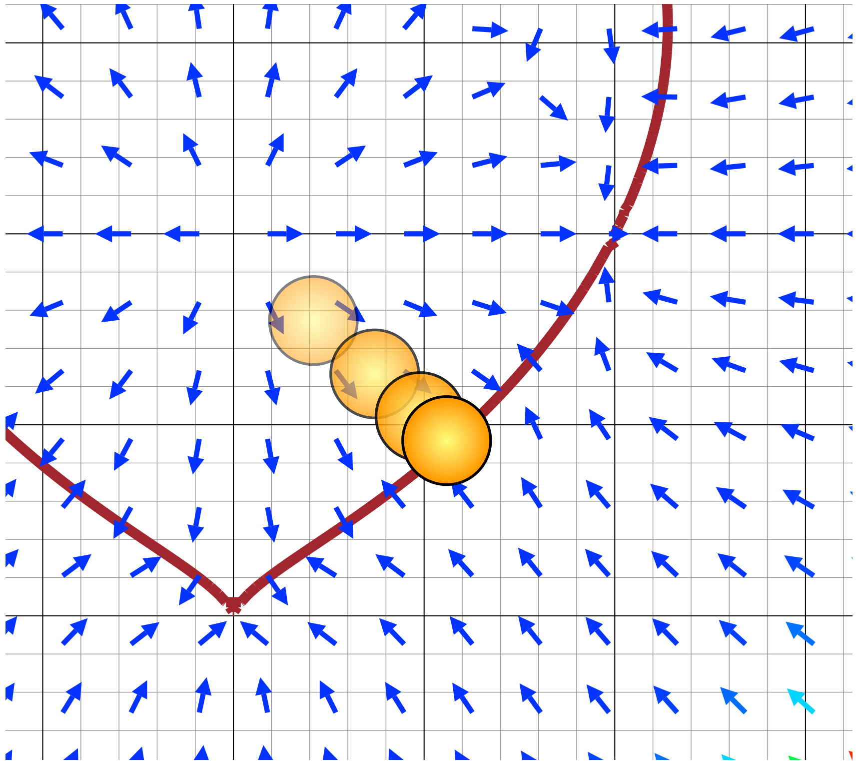


$$-S(x,y)\nabla S(x,y)$$

The "Seeding" Algorithm

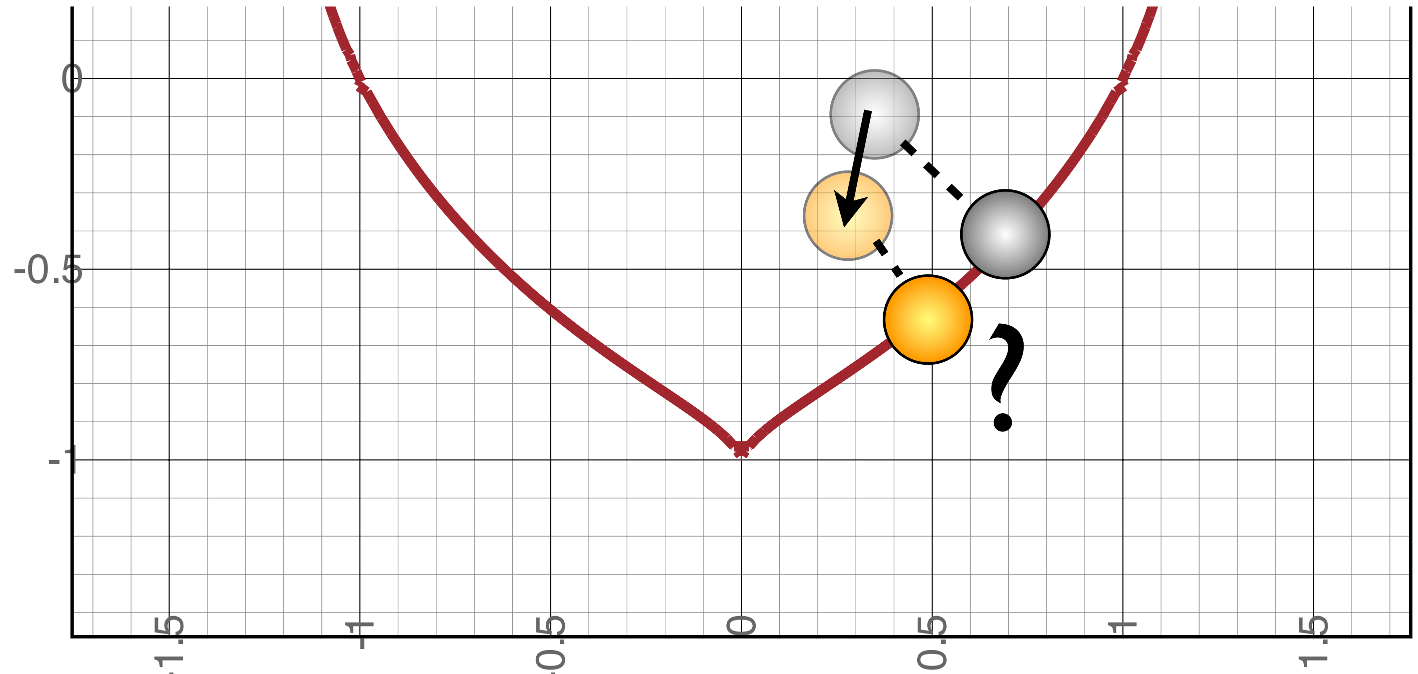
- Given a seed point, find the nearest point on the surface (within a certain tolerance)
- Exploit the condition that the seed is known to start close to the surface

$$
\begin{aligned}
&\mathbf{p} \leftarrow \mathbf{p}_{seed} \\
&\text{do} \\
&\quad \delta\mathbf{p} \leftarrow -\frac{S(\mathbf{p})\nabla S(\mathbf{p})}{\nabla S(\mathbf{p}) \cdot \nabla S(\mathbf{p})} \\
&\quad \mathbf{p} \leftarrow \mathbf{p} + \delta\mathbf{p} \\
&\text{until } (||\delta\mathbf{p}|| < \epsilon)
\end{aligned}
$$



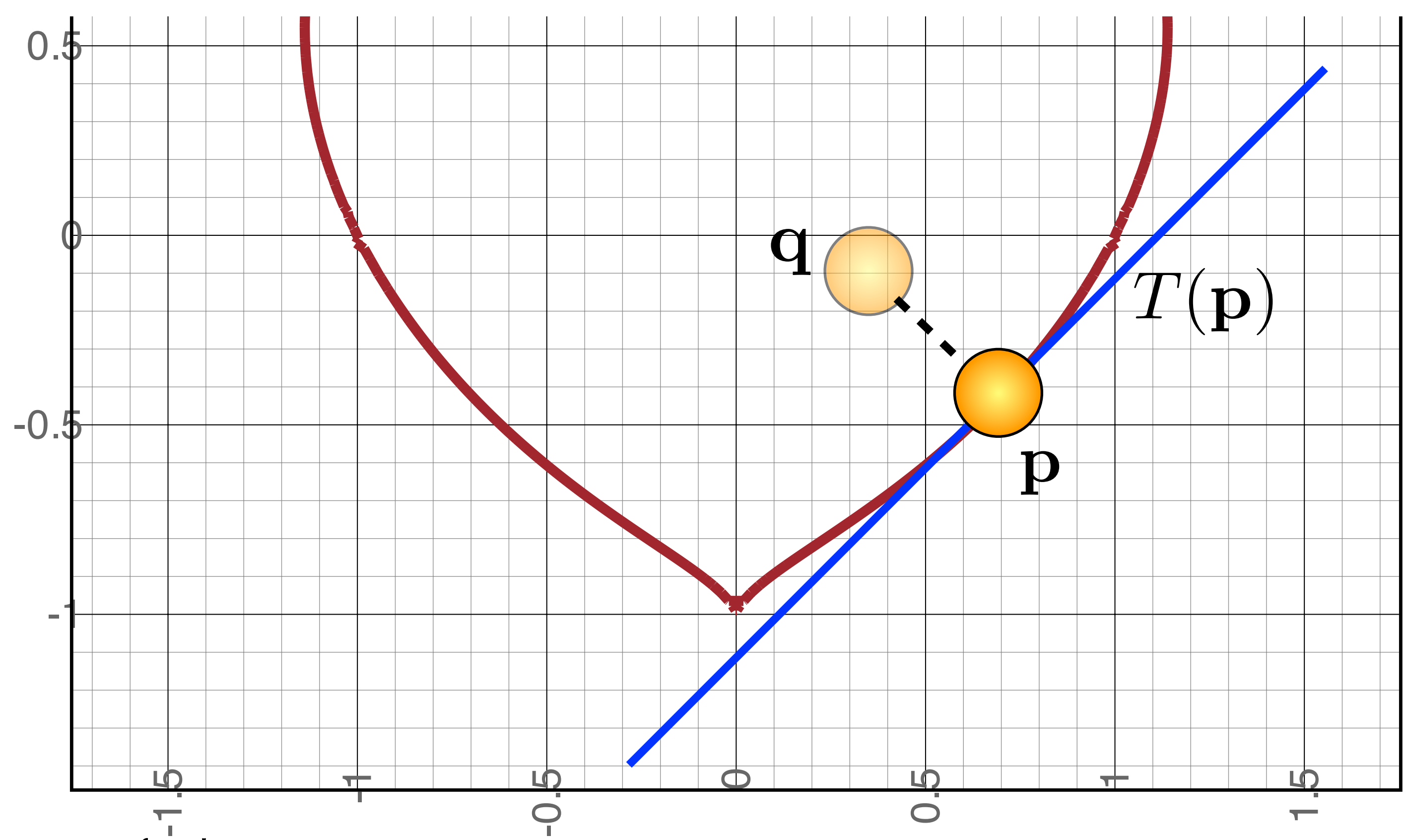
Step 3: Tracking the Surface Point

- As the device moves, we need to update our surface point, subject to constraints
- What are these constraints?
- Can we use the seeding algorithm again?

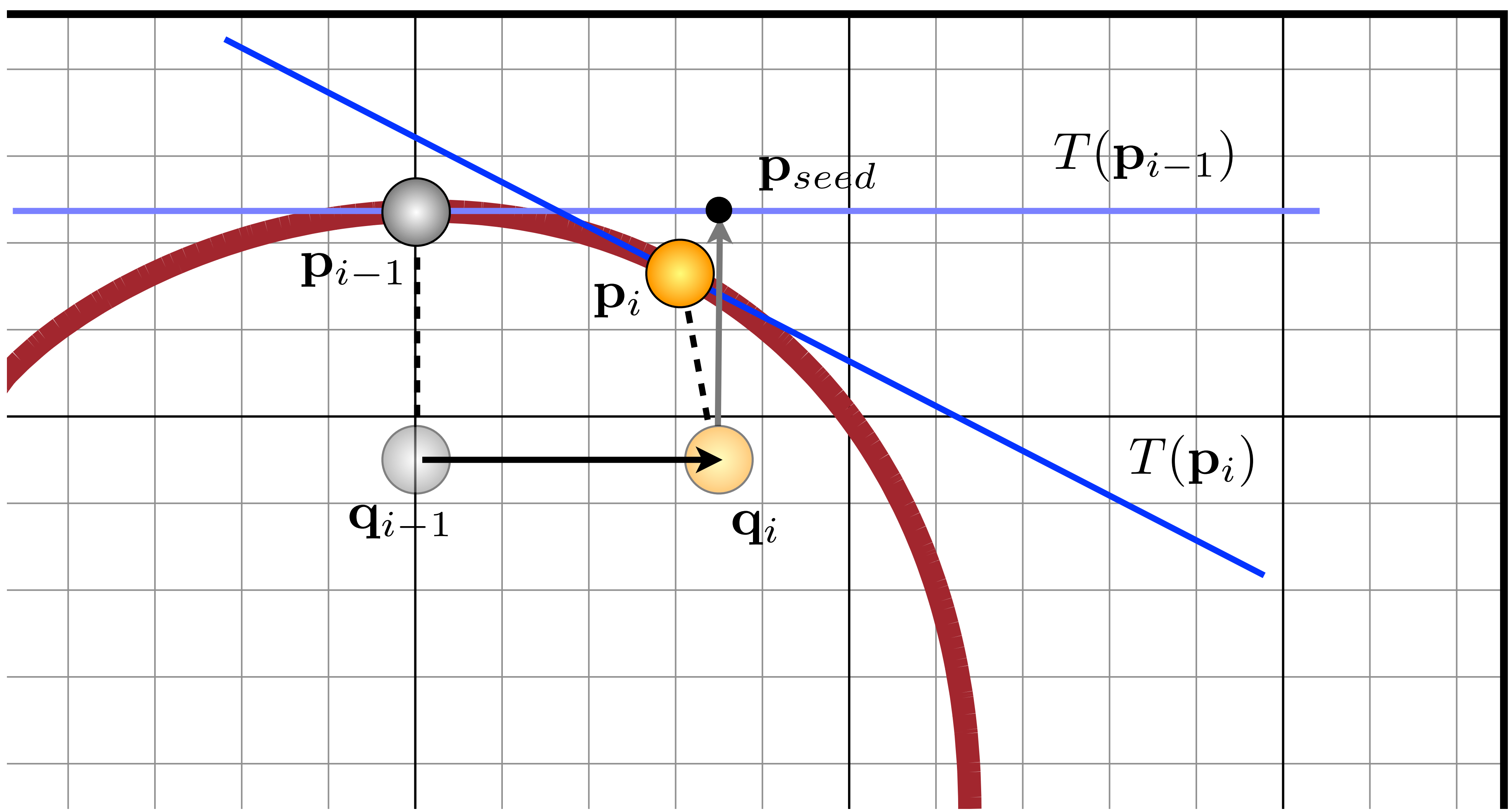


Constrained by a Plane

- We have a point on the surface...
- We have the surface normal (gradient)...
- The answer is to use a tangent plane!

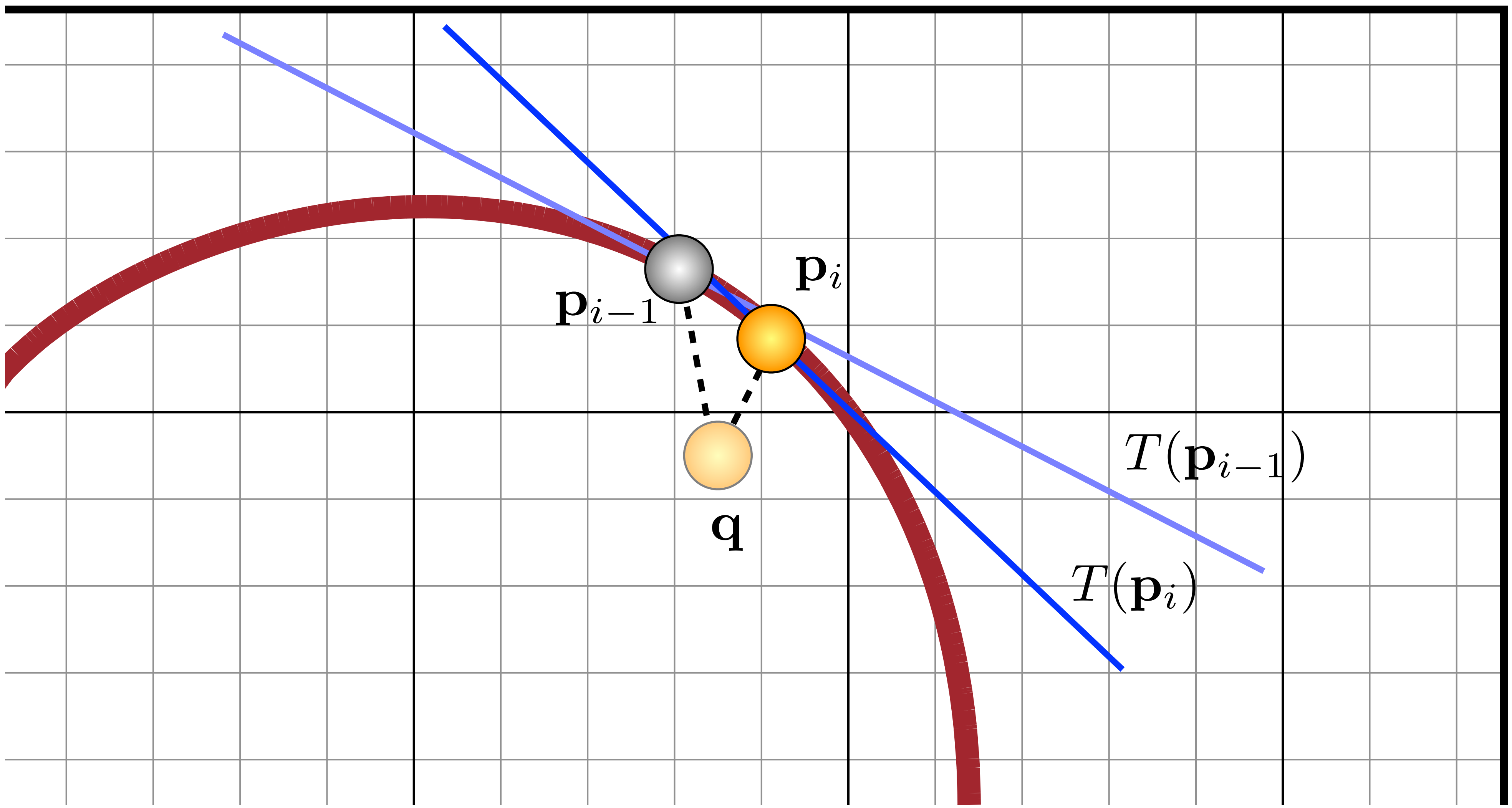


Surface Tracking



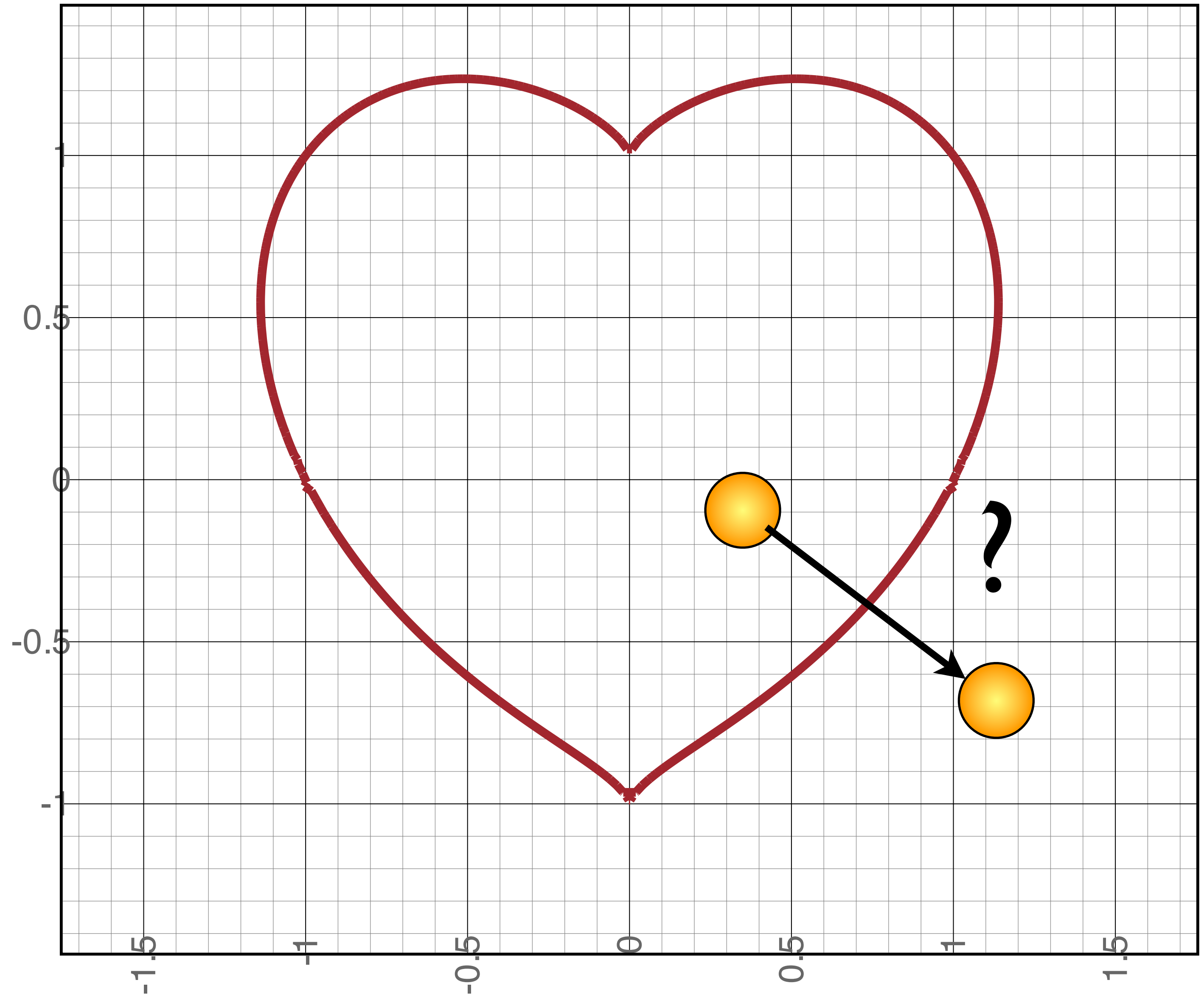
One time step: Is p the nearest surface point?

Surface Tracking



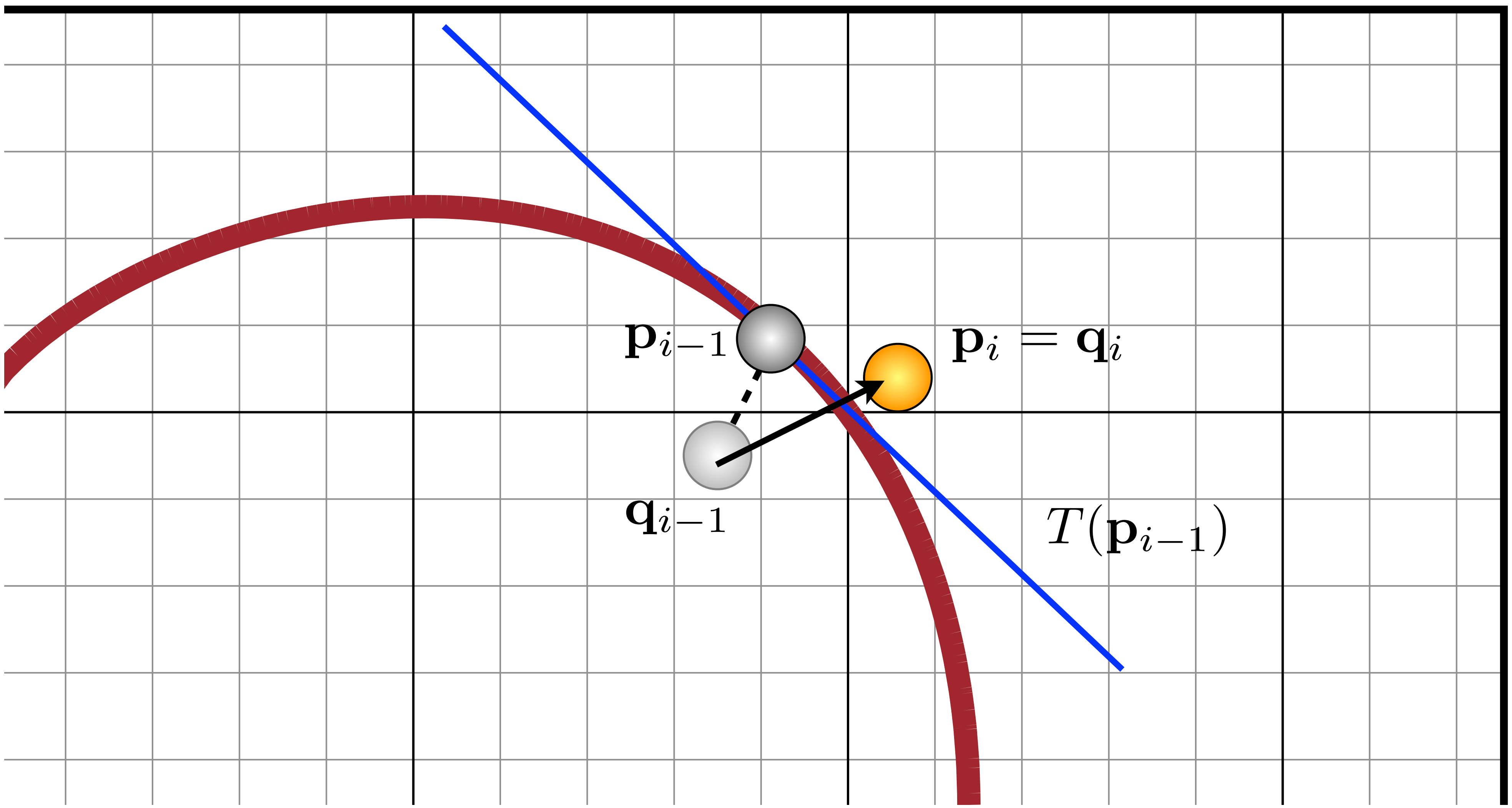
One more time step: A lot closer now!

Step 4: Breaking Contact



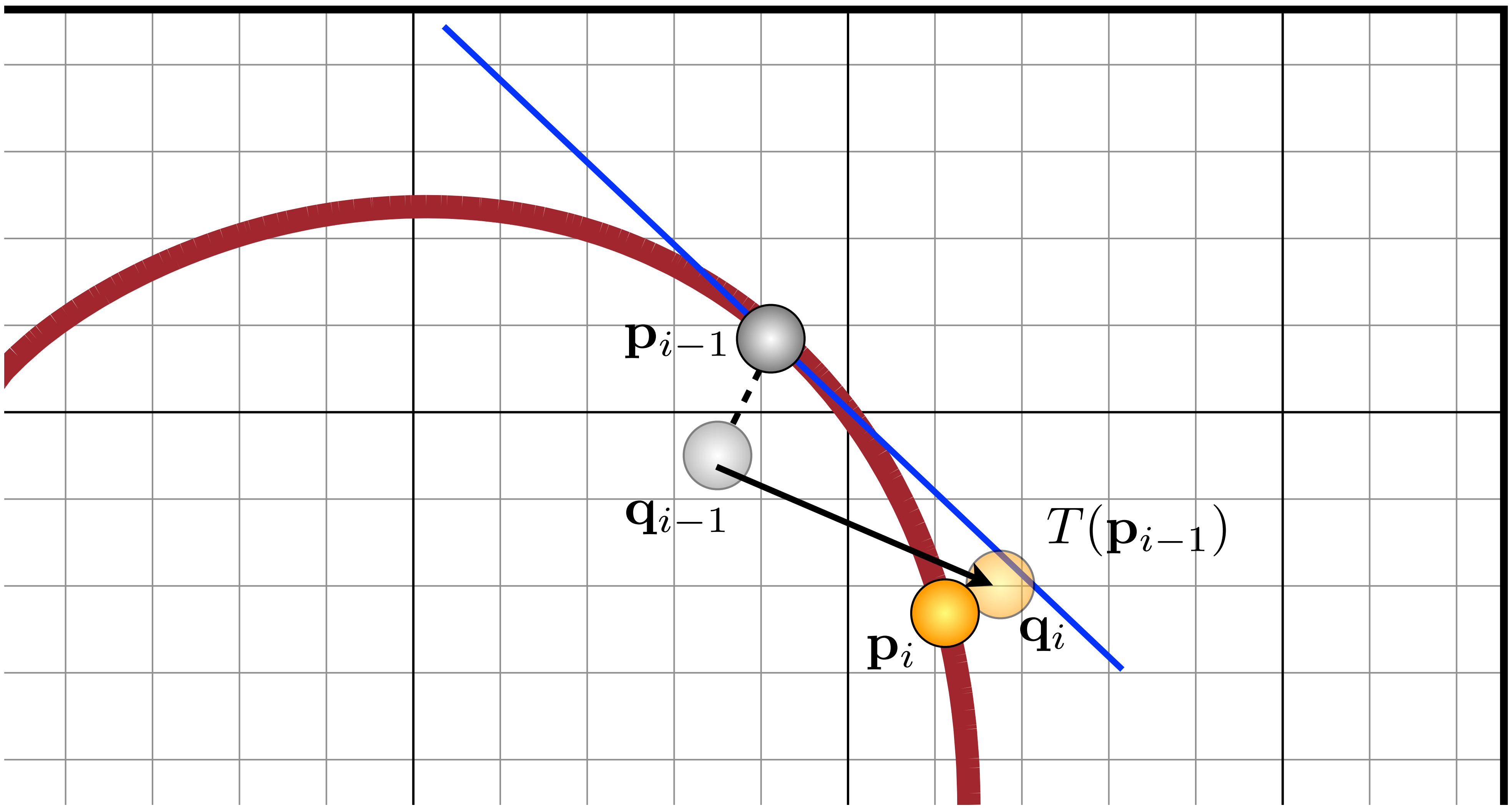
How do we know when to stop tracking?

Tangent to the Rescue



Contact is broken when q moves to the outside of the constraining plane (same direction as normal).

Incorrect Break?



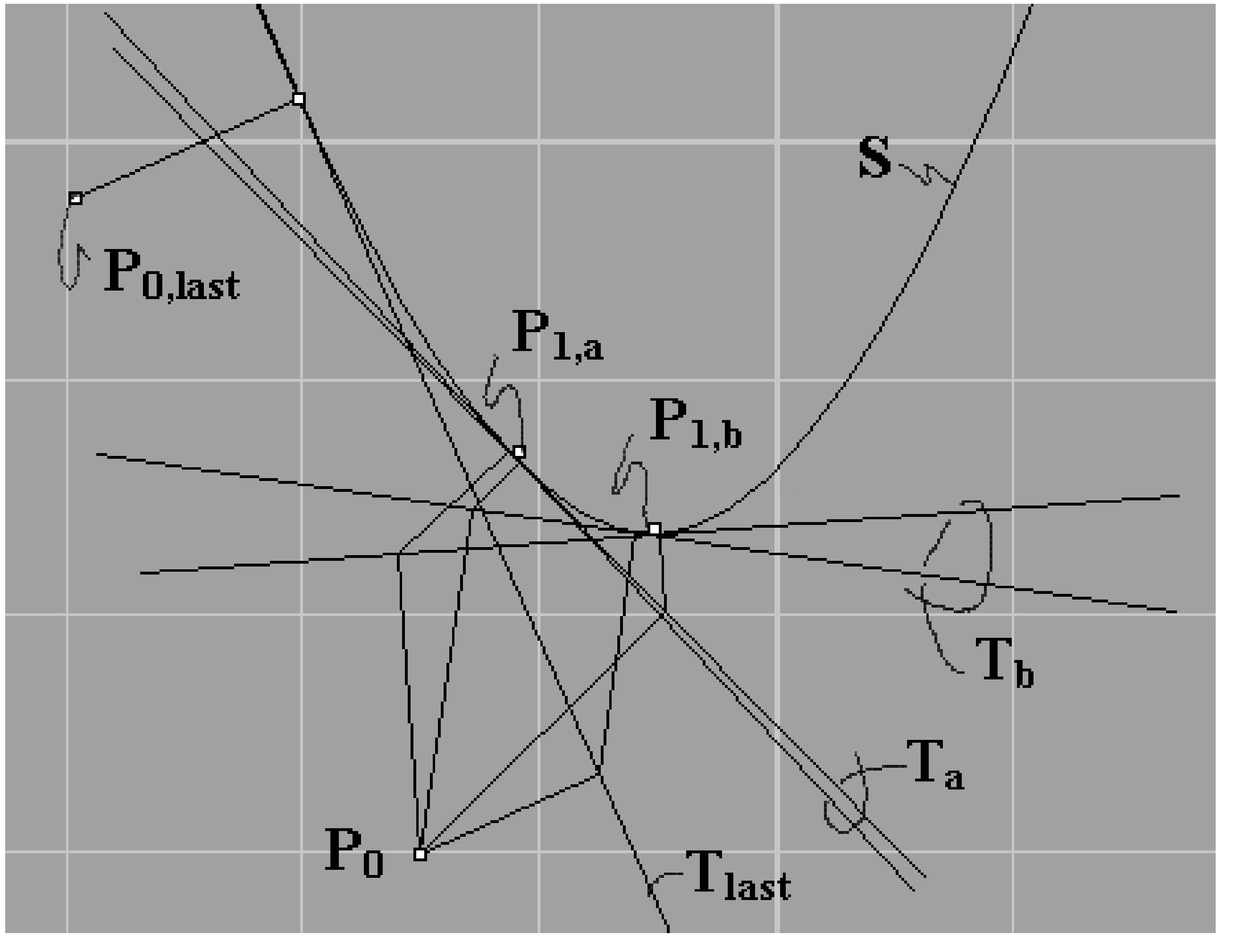
What happens here?

Summary

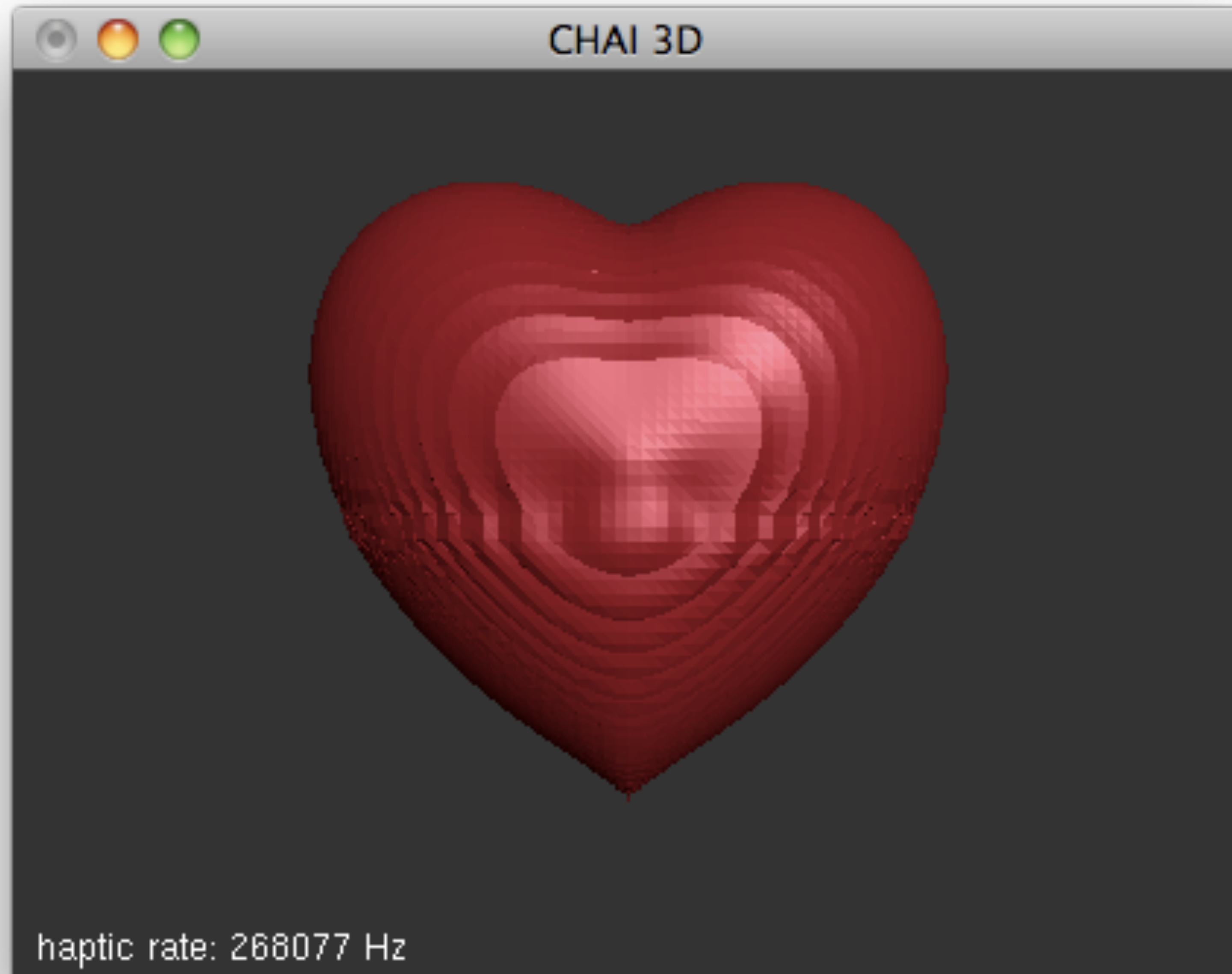
- The full implicit surface rendering algorithm:
  - Detect initial contact when $S(p) < 0$
  - Find surface point using initial point as seed
  - Update the surface point as the device moves by using the tangent plane as a constraint
  - Contact breaks when device is moved outside the constraining plane
- Repeat from start...

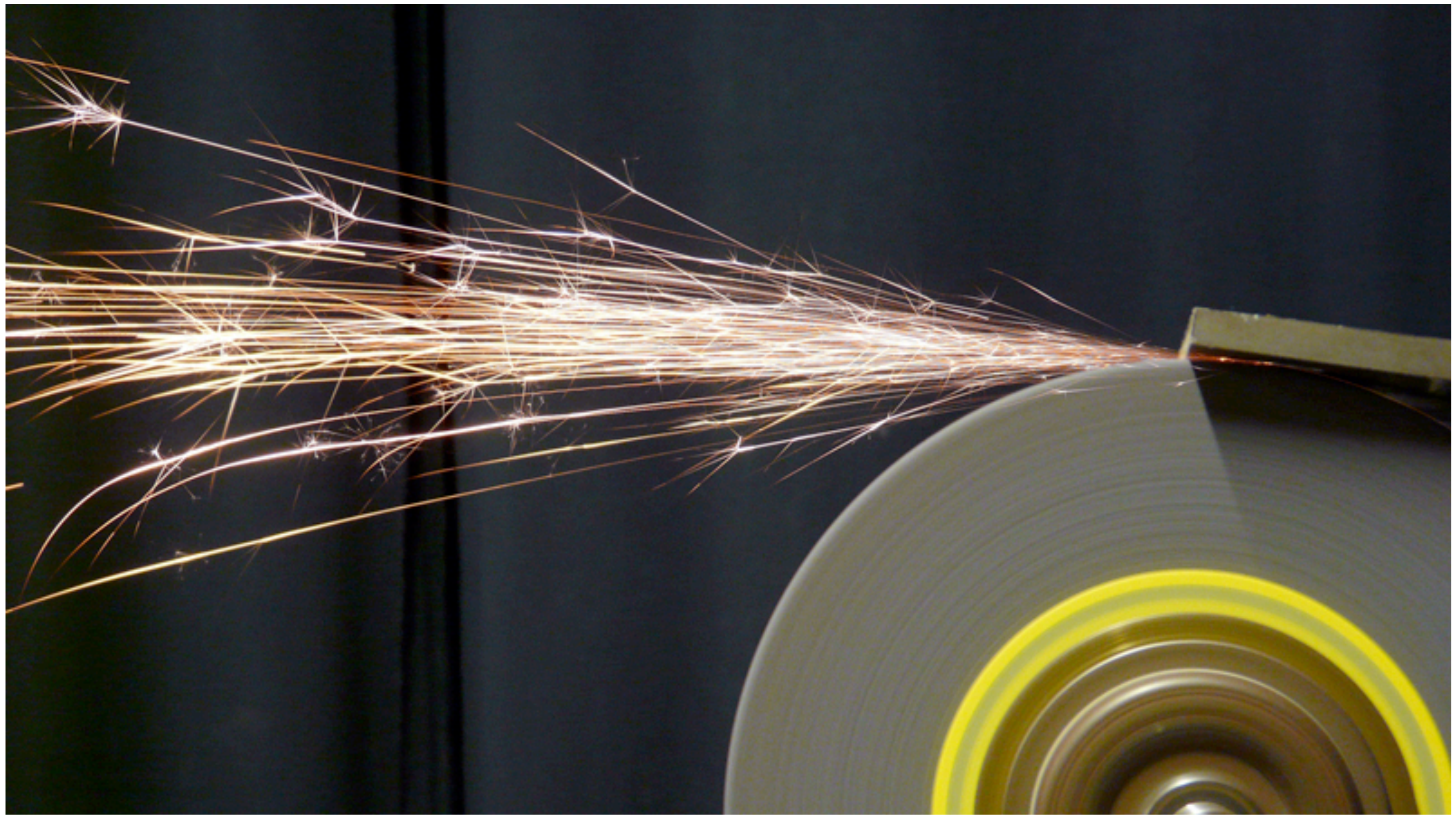
Potential Limitations

- Can this algorithm handle thin objects?
- A limit cycle?!



Implicit Surface Demo





Friction

Coulomb Friction

- Friction force proportional to normal force

$$F_f = \mu F_N$$

- Static (sticking) friction:

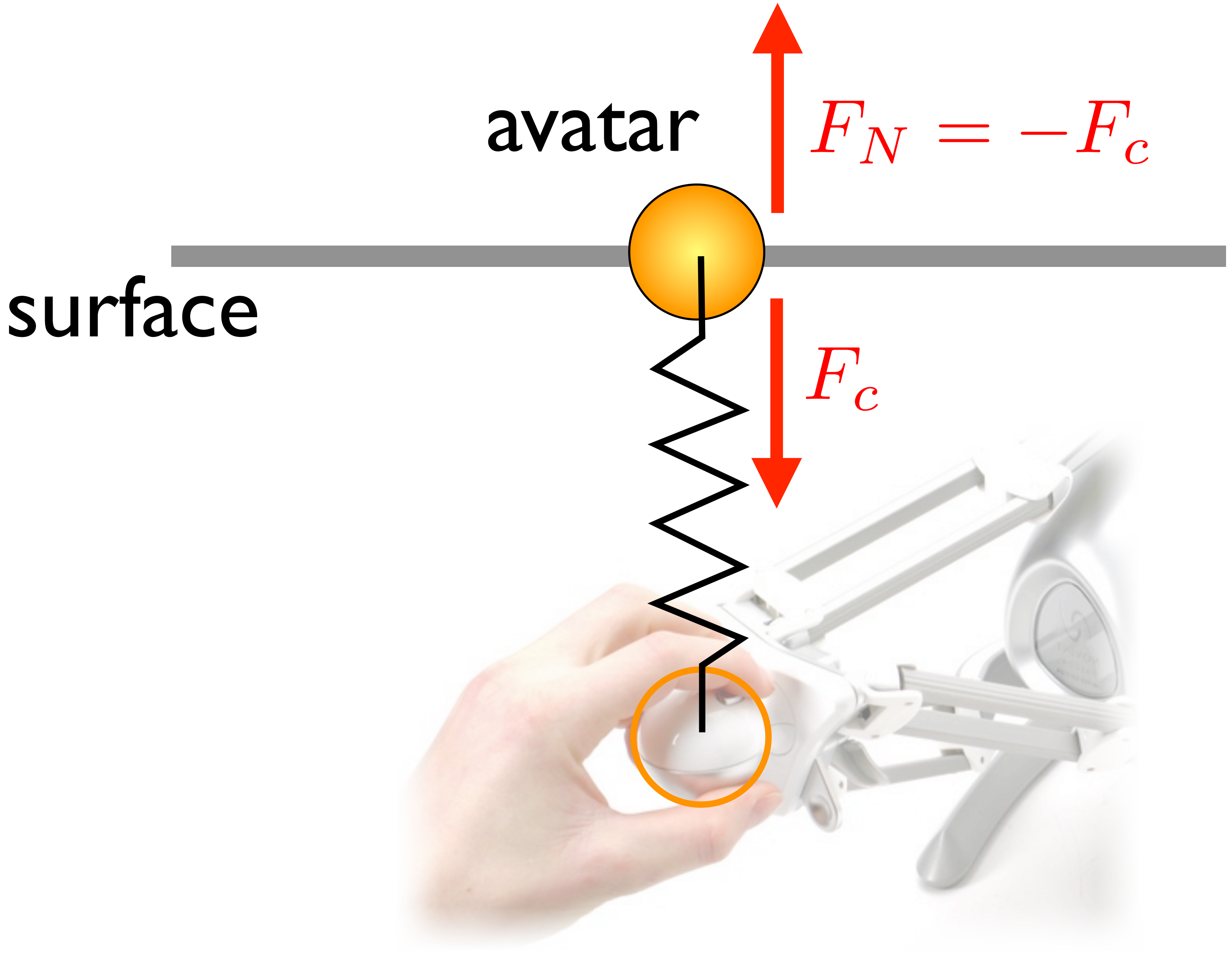
$$F_s \leq \mu_s F_N$$

- Kinetic (sliding) friction:

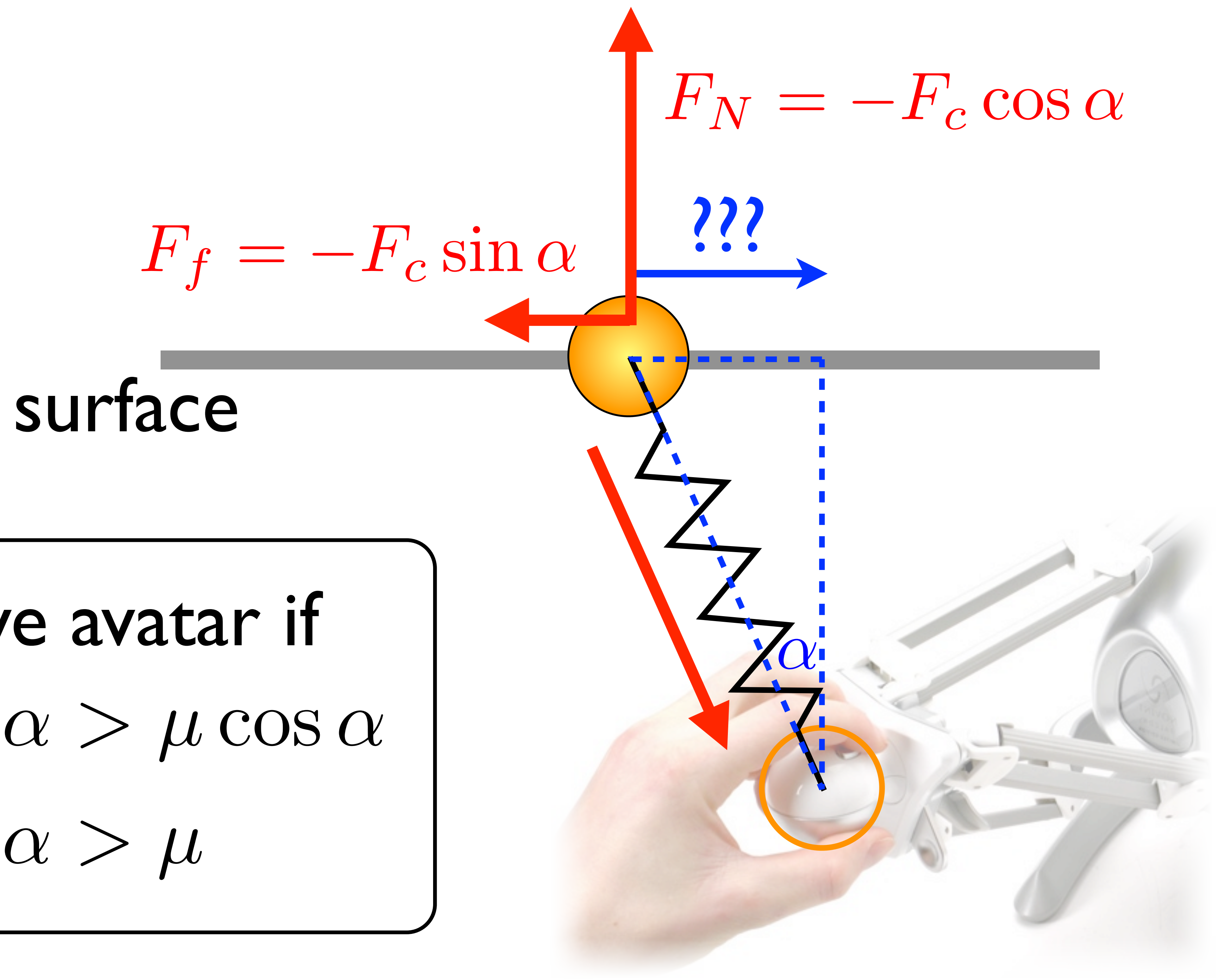
$$F_k = \mu_k F_N$$

Rendering Friction

- Basic case: $\mu = \mu_s = \mu_k$



Rendering Friction

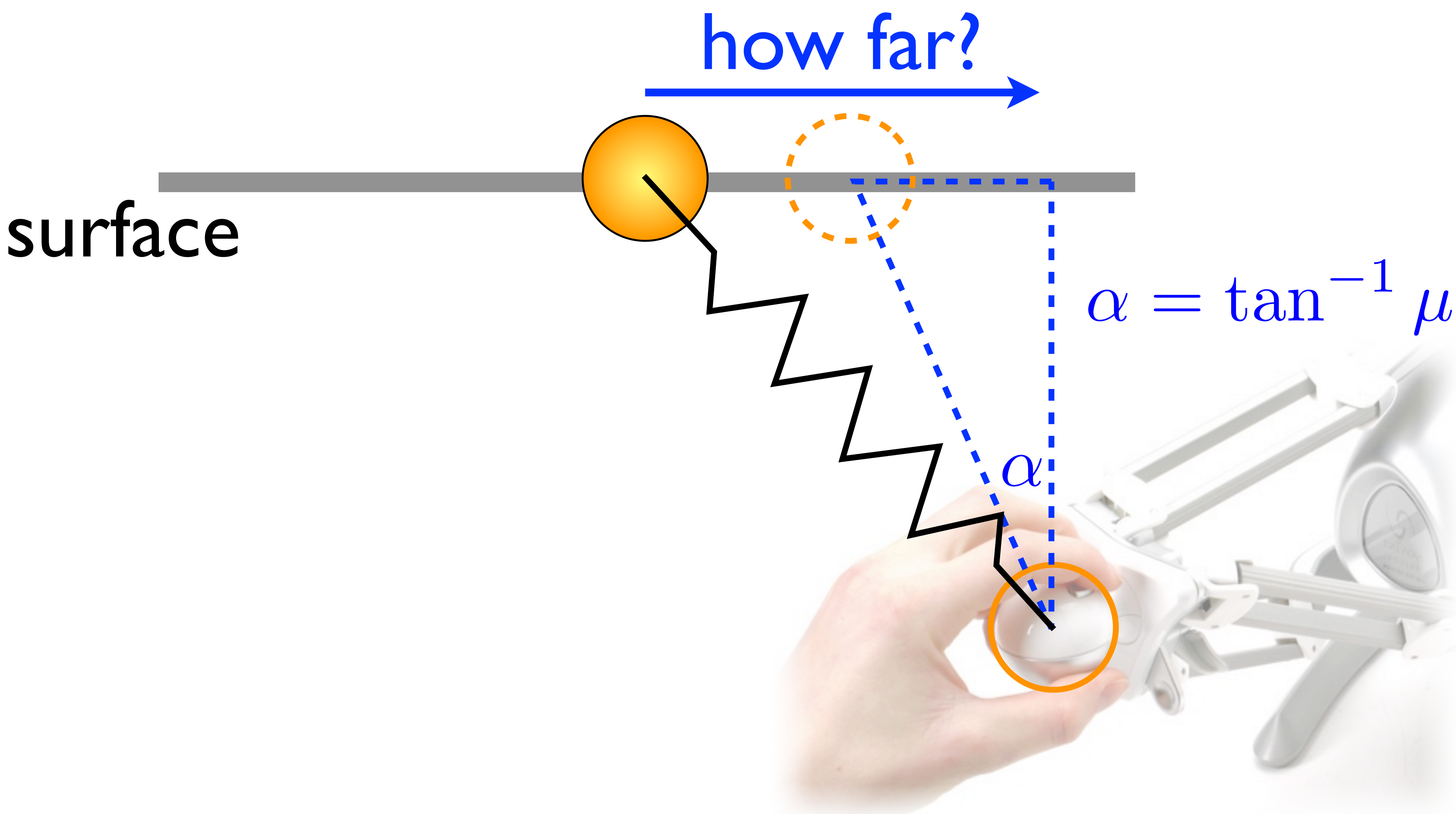


Move avatar if

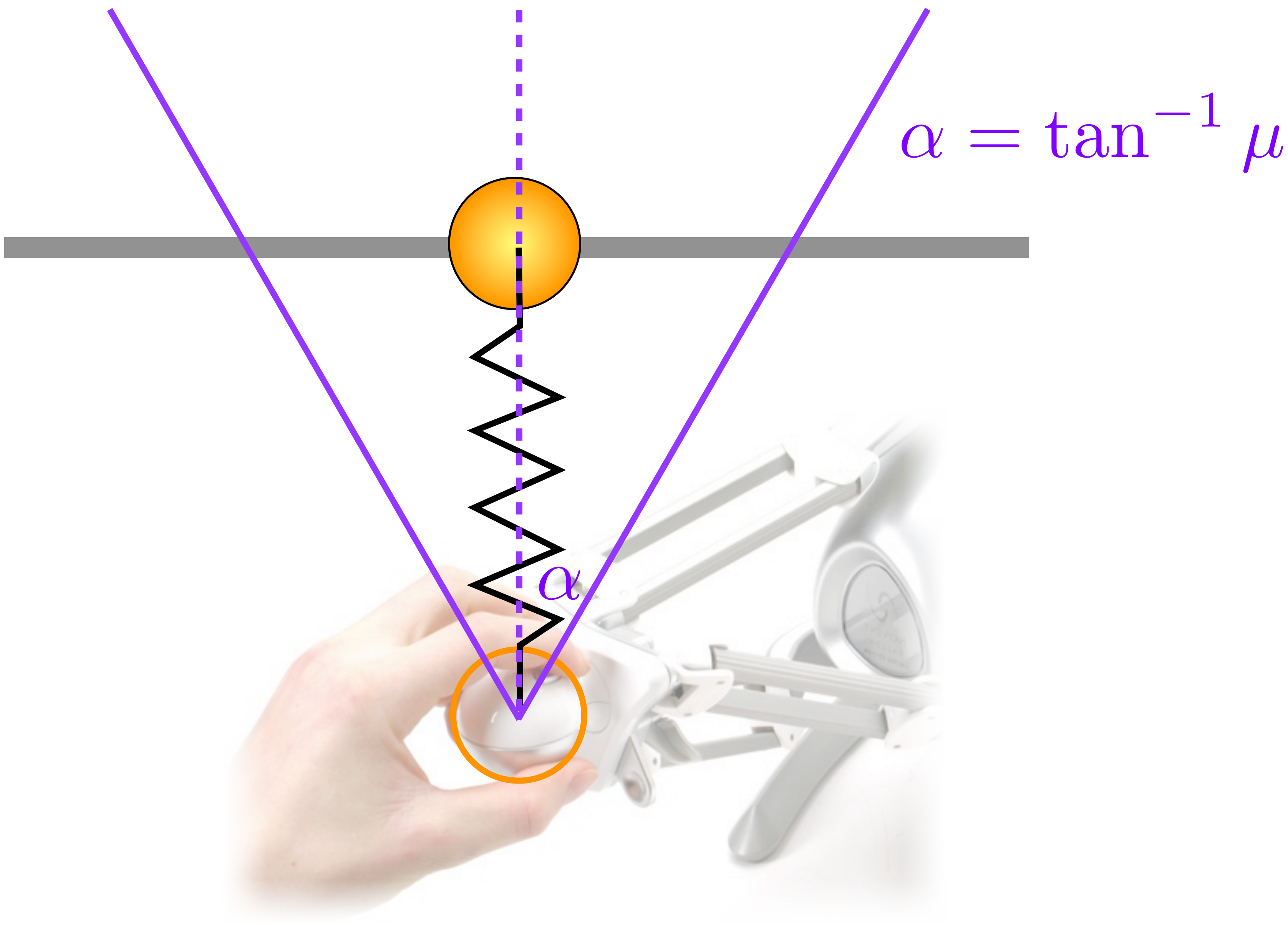
$$\sin\alpha > \mu\cos\alpha$$

$$\tan\alpha > \mu$$

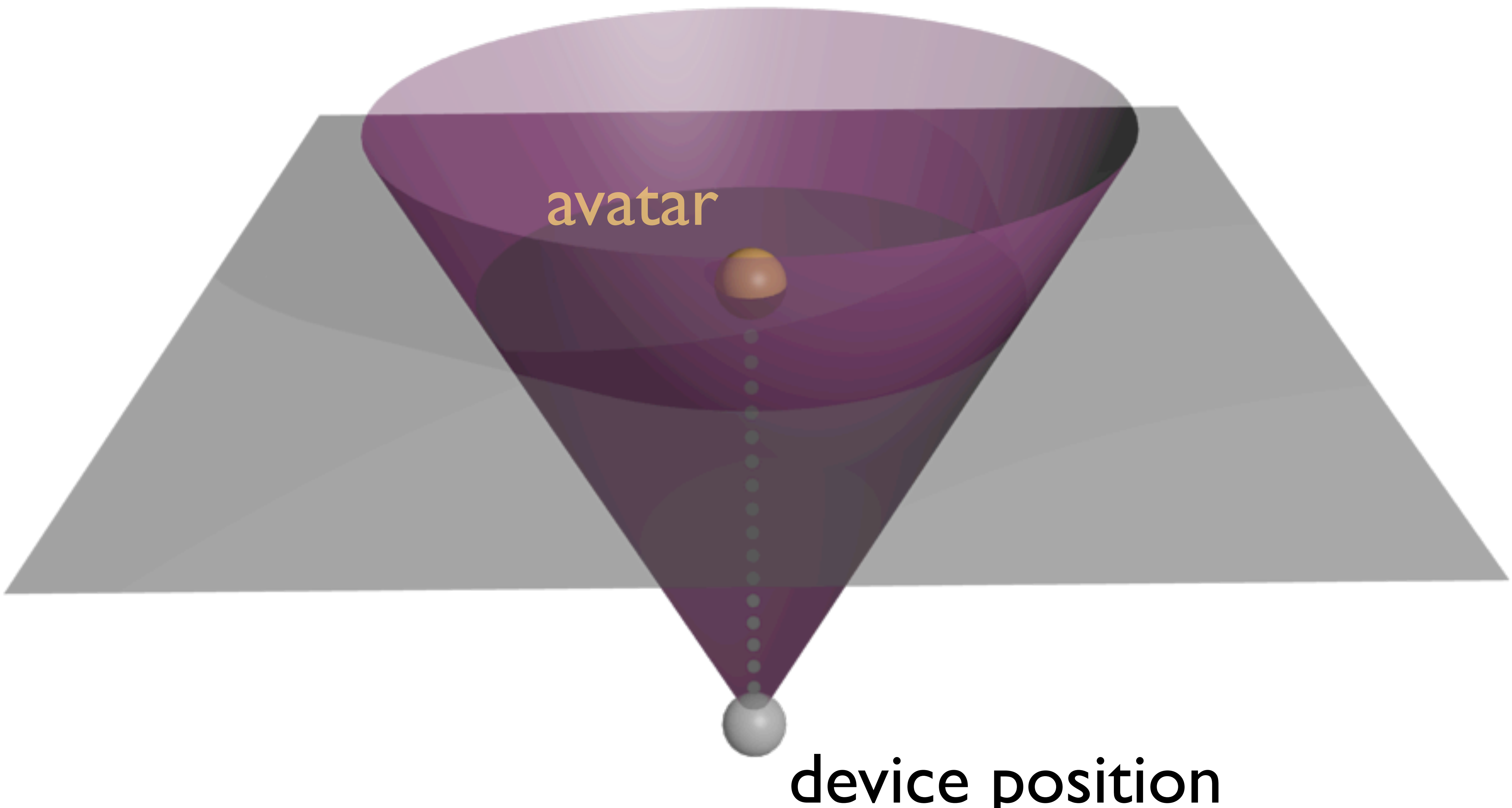
Rendering Friction



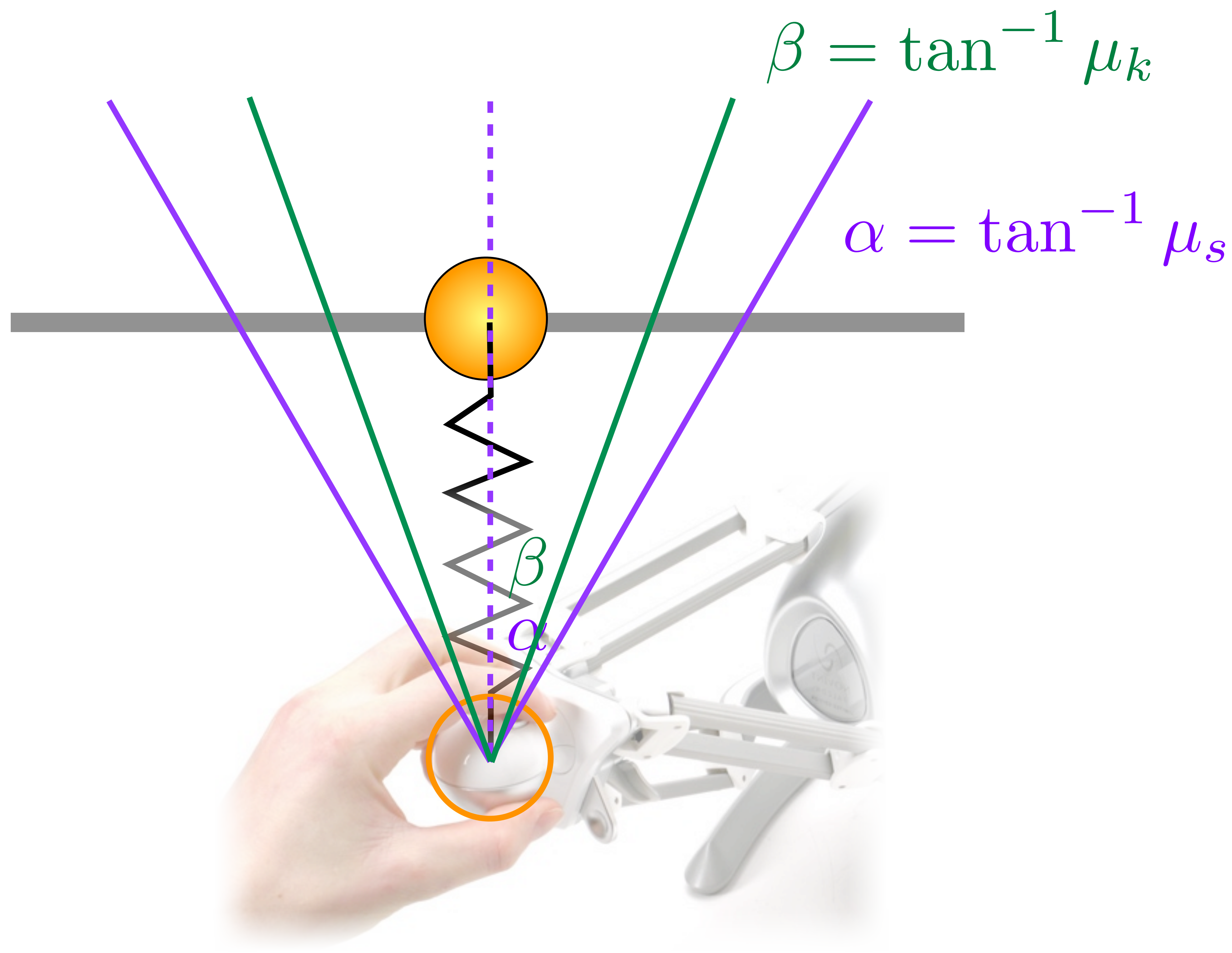
Friction Cone



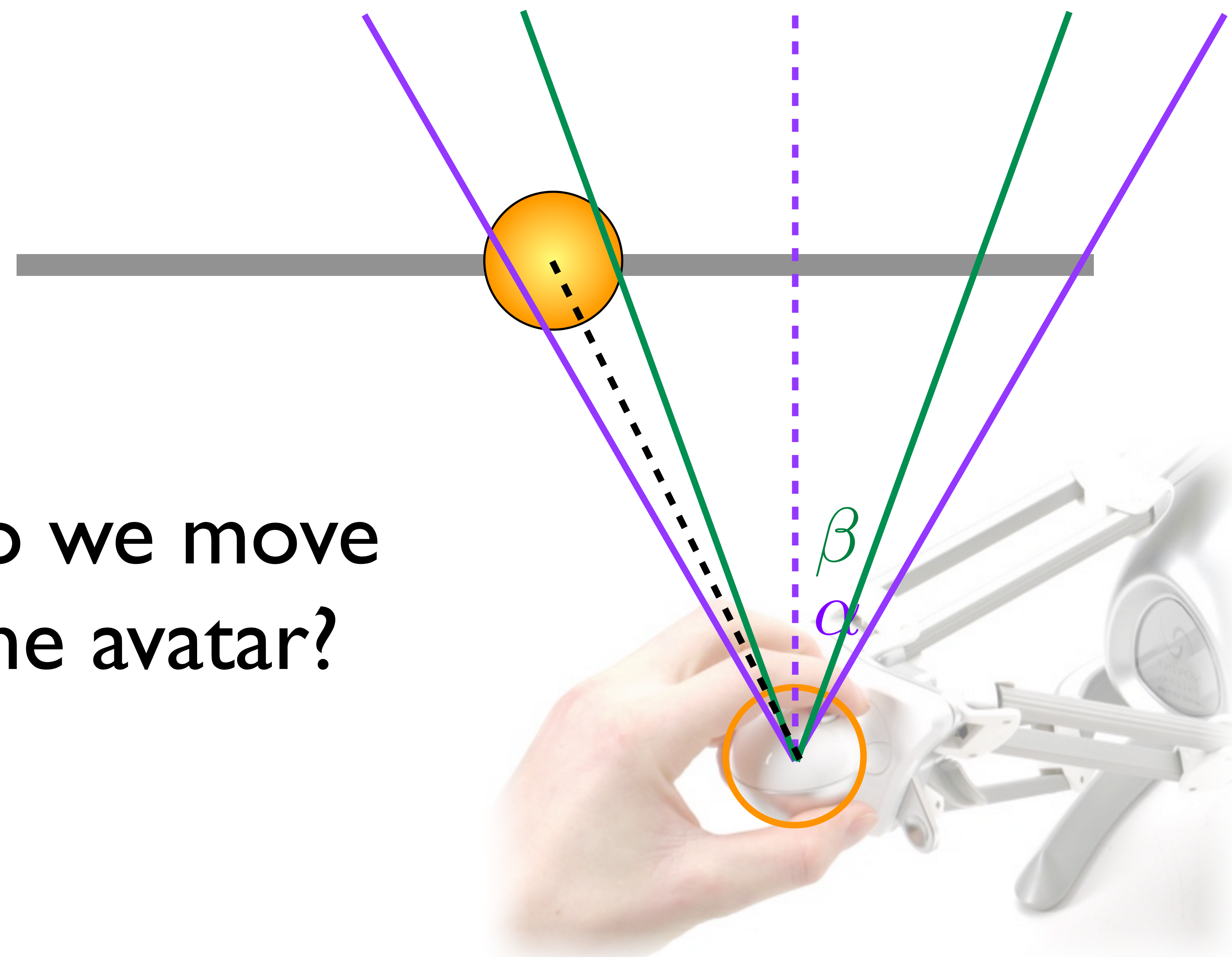
Friction Cone in 3D



Static & Kinetic Friction

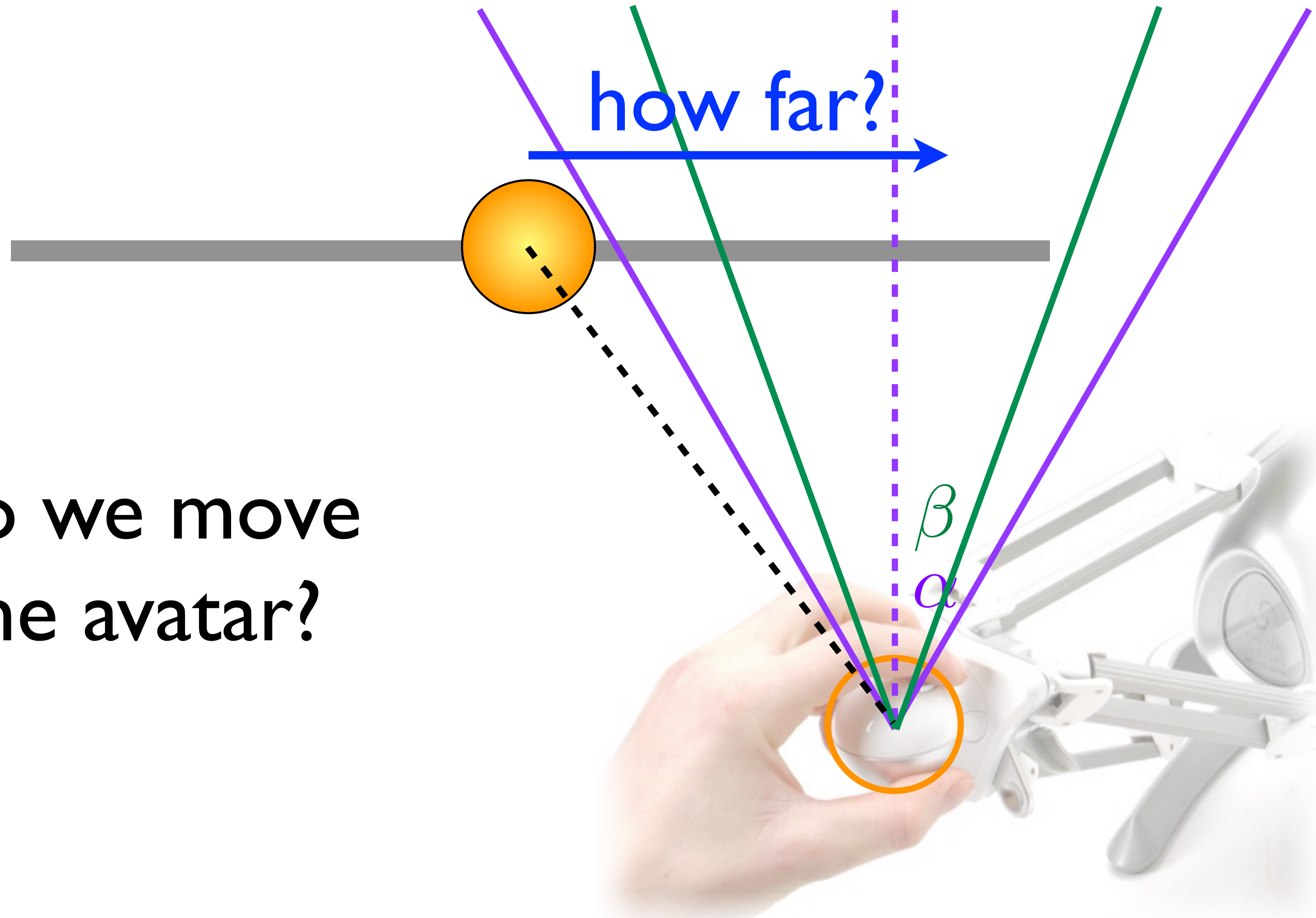


Static & Kinetic Friction



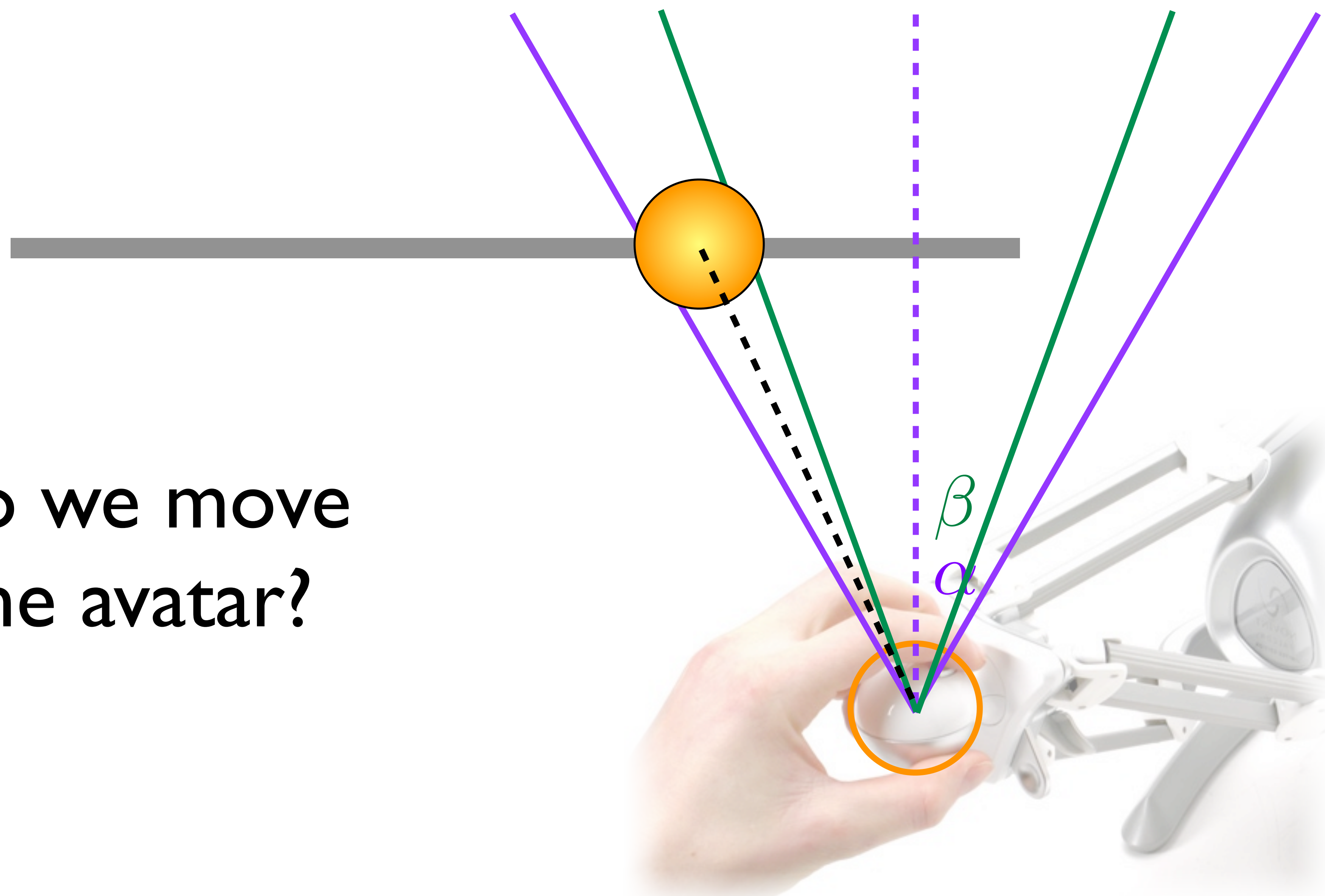
Do we move the avatar?

Static & Kinetic Friction



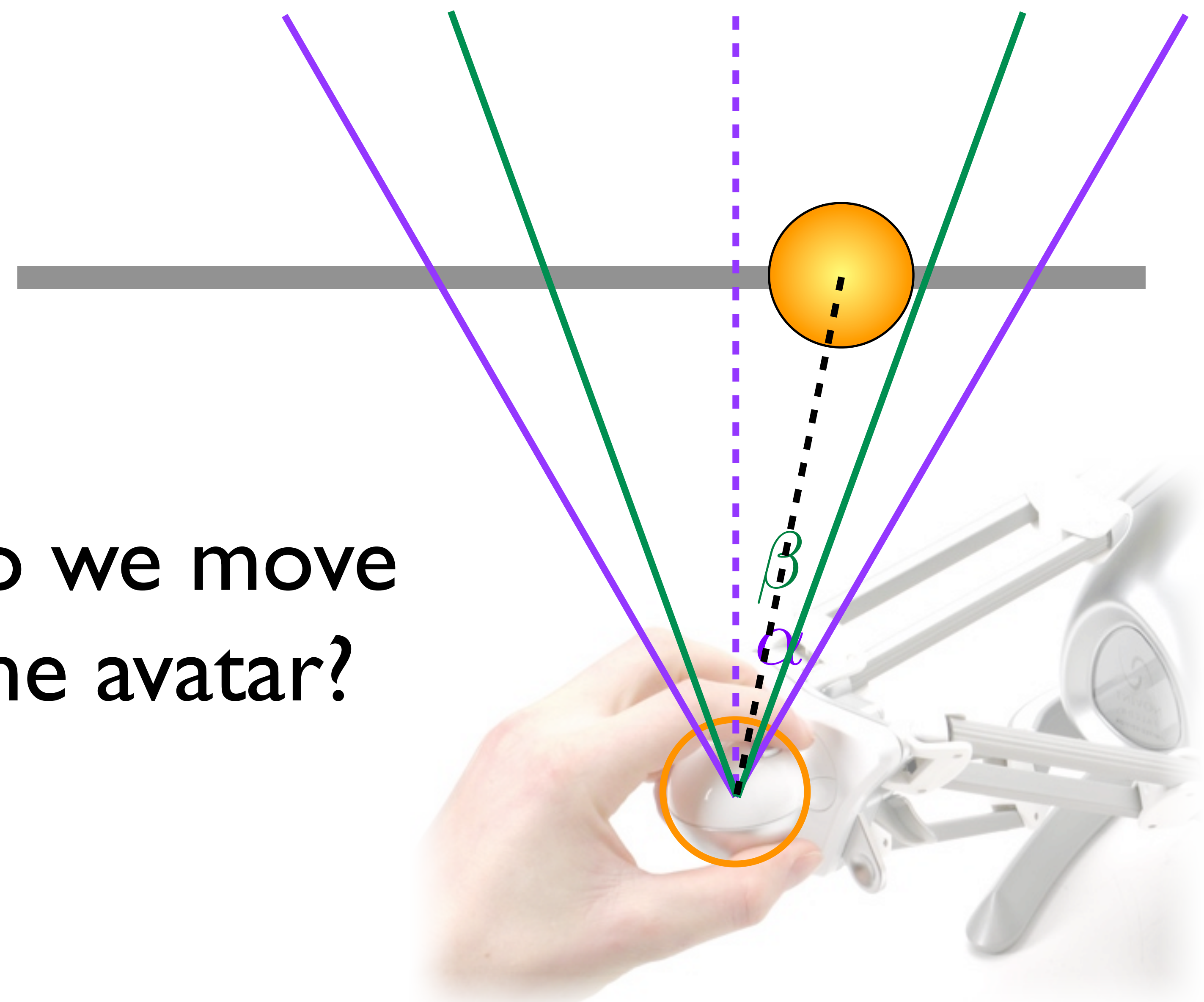
Do we move
the avatar?

Static & Kinetic Friction



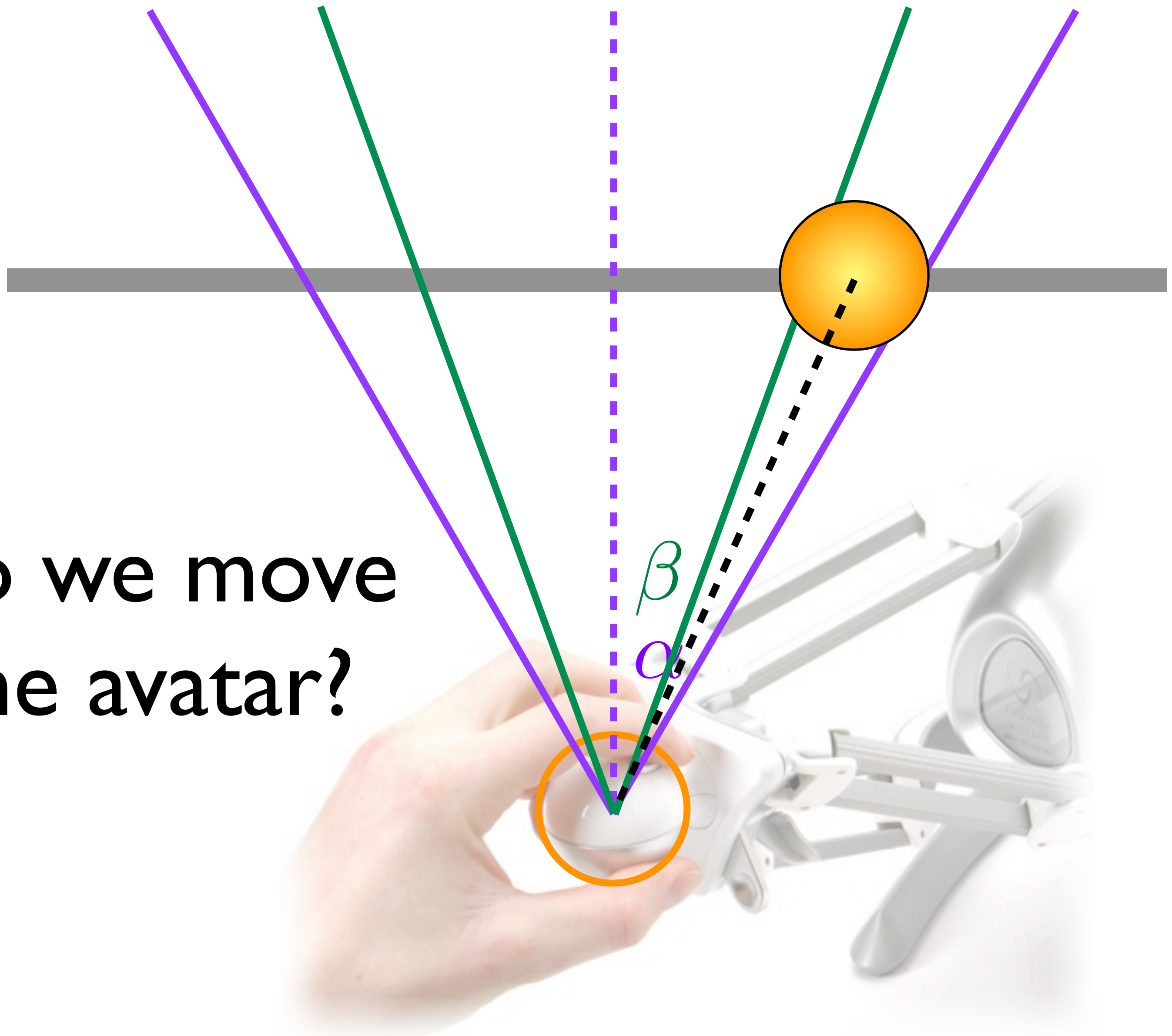
Do we move
the avatar?

Static & Kinetic Friction



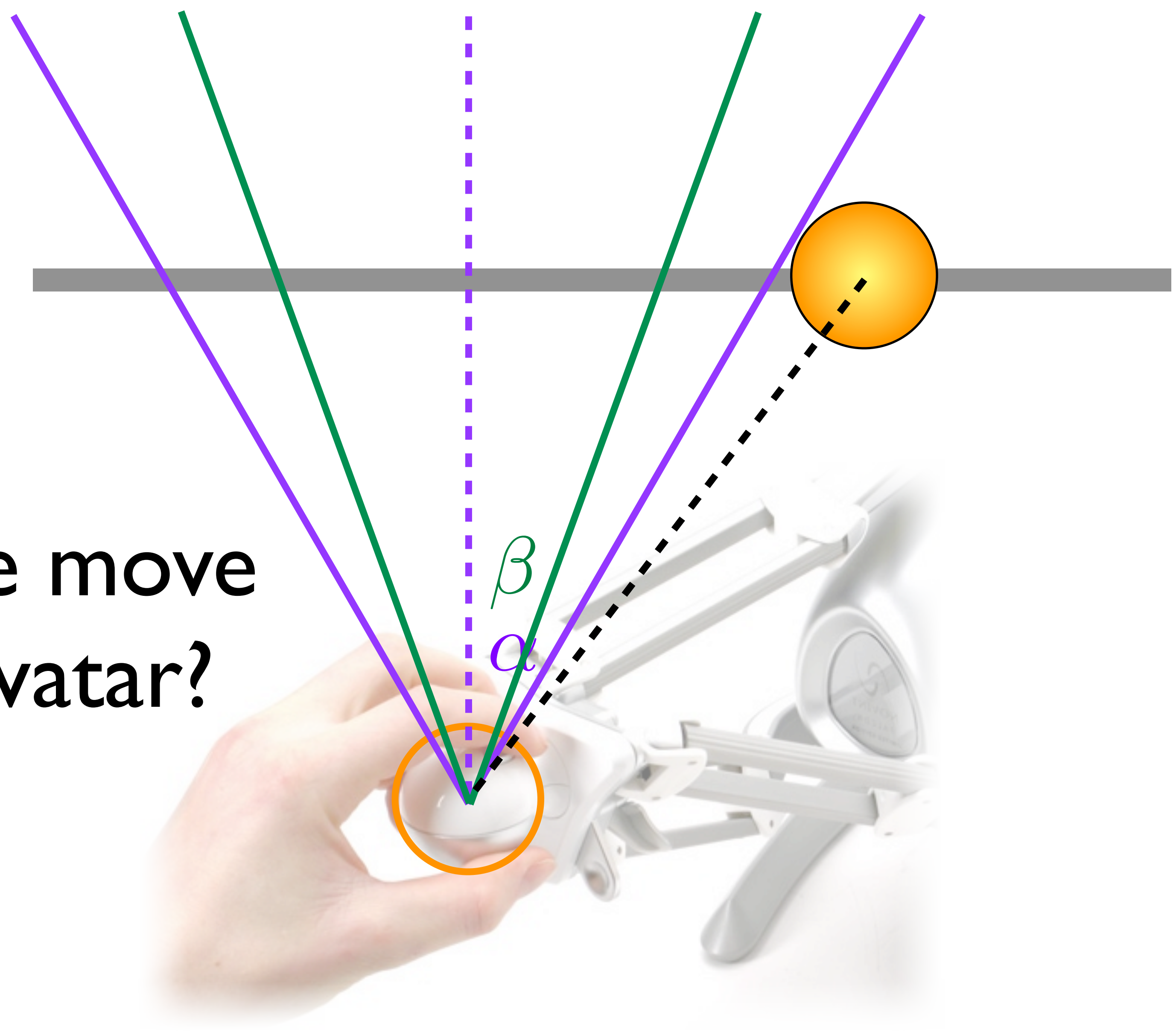
Do we move the avatar?

Static & Kinetic Friction



Do we move the avatar?

Static & Kinetic Friction

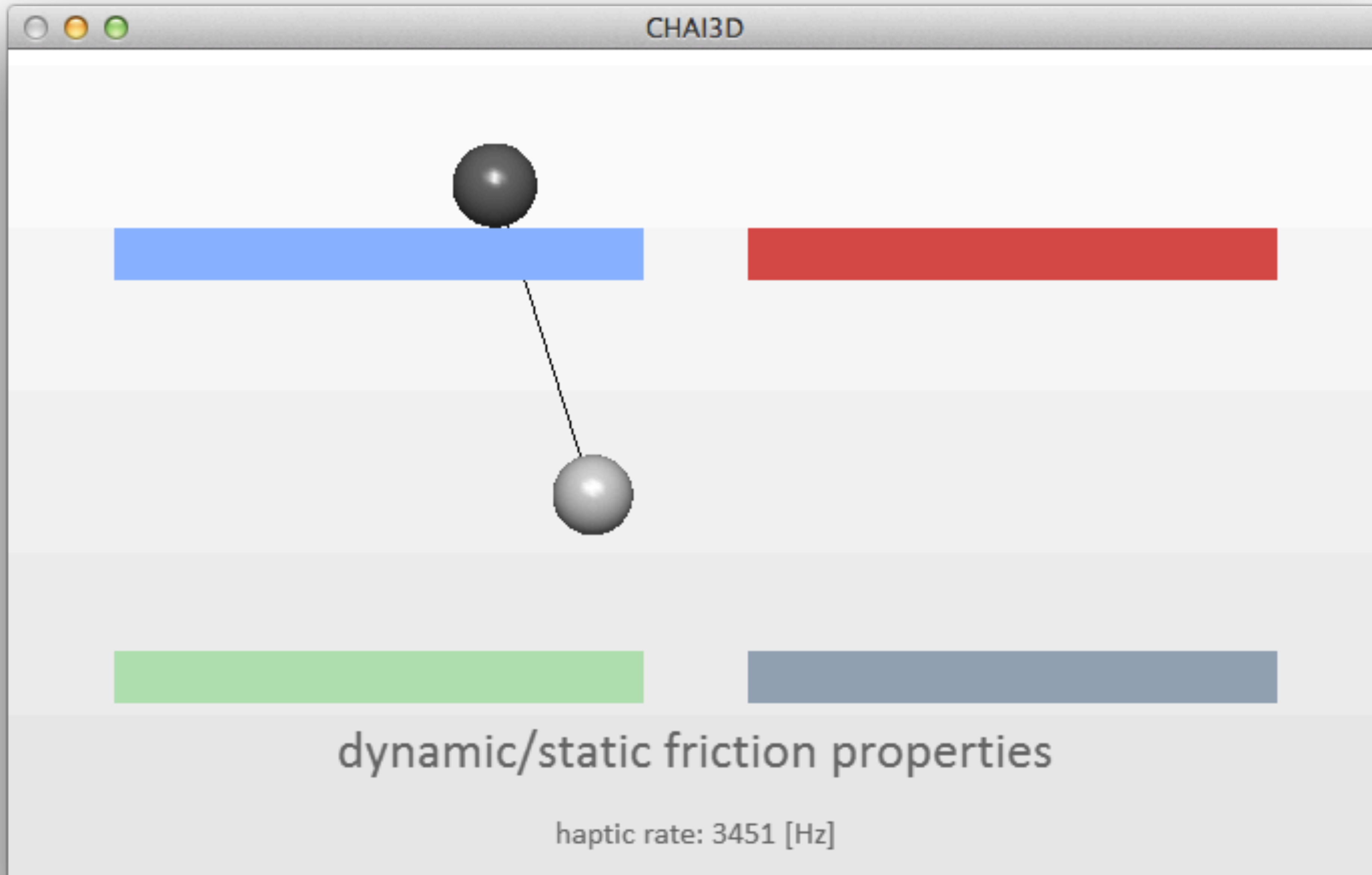


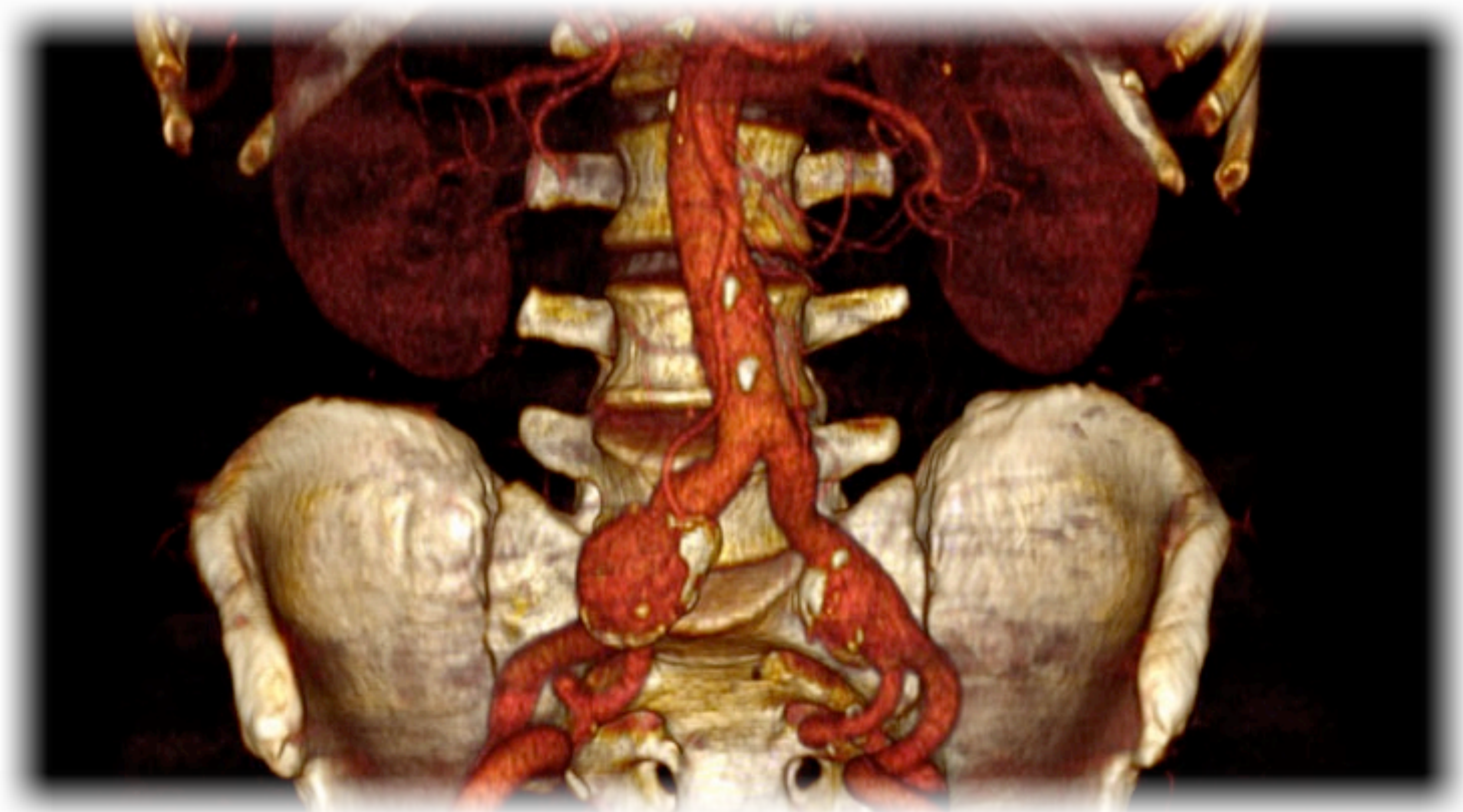
Do we move the avatar?

Coulomb Friction

- Friction force proportional to normal force
- Construct friction cone(s) from coefficents
- Can render effects of static and kinetic friction, and in general $\mu_s \geq \mu_k$
- When do we switch between static and kinetic friction cones?

Friction Demo





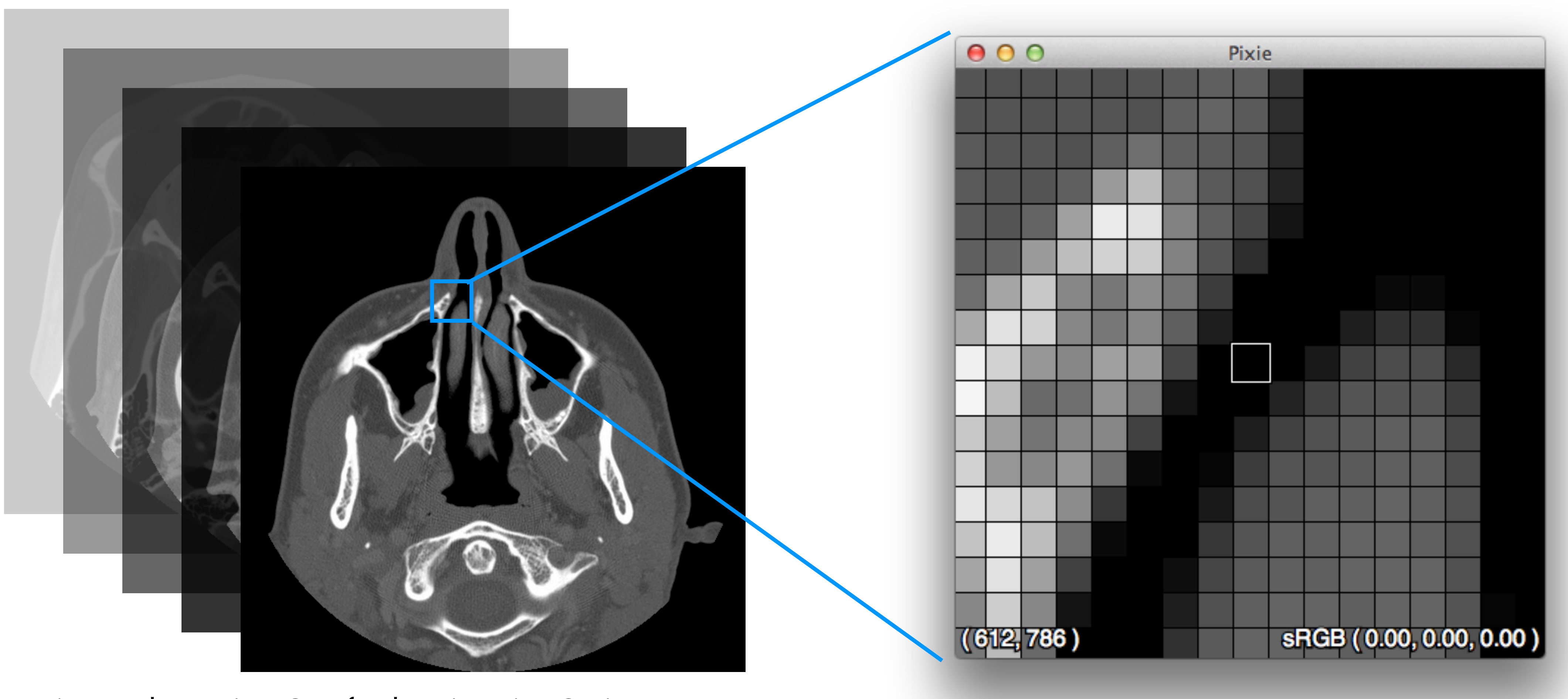
Volumetric Isosurfaces

Volume Rendering

- Implicit representations are now uncommon, but...
- 3D medical imaging (CT, MRI, etc.) has resulted in an abundance of volume data
- Can be rendered with (almost) the same algorithm!

Sampled Volume Data

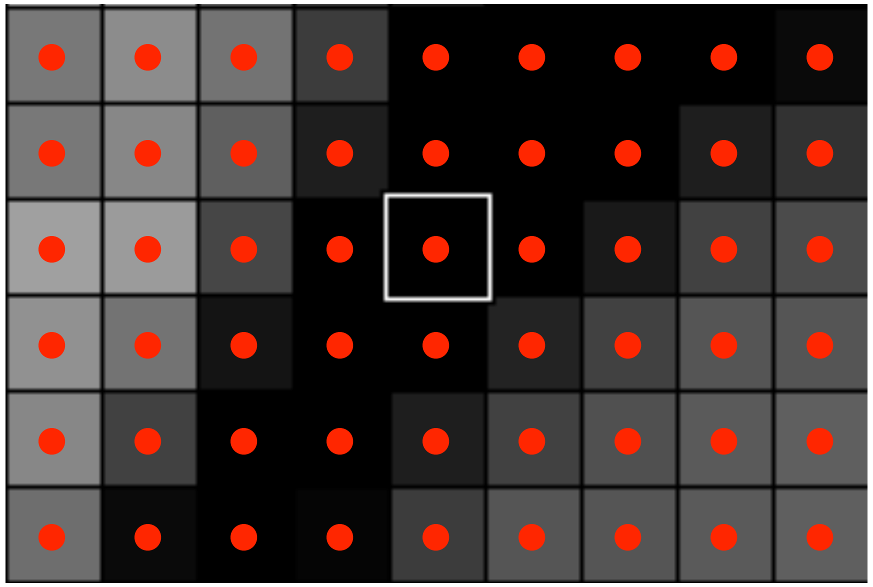
- Image values sampled on rectilinear grid
- CT scans measure *radiodensity* - Hounsfield Units



Implicit Representation

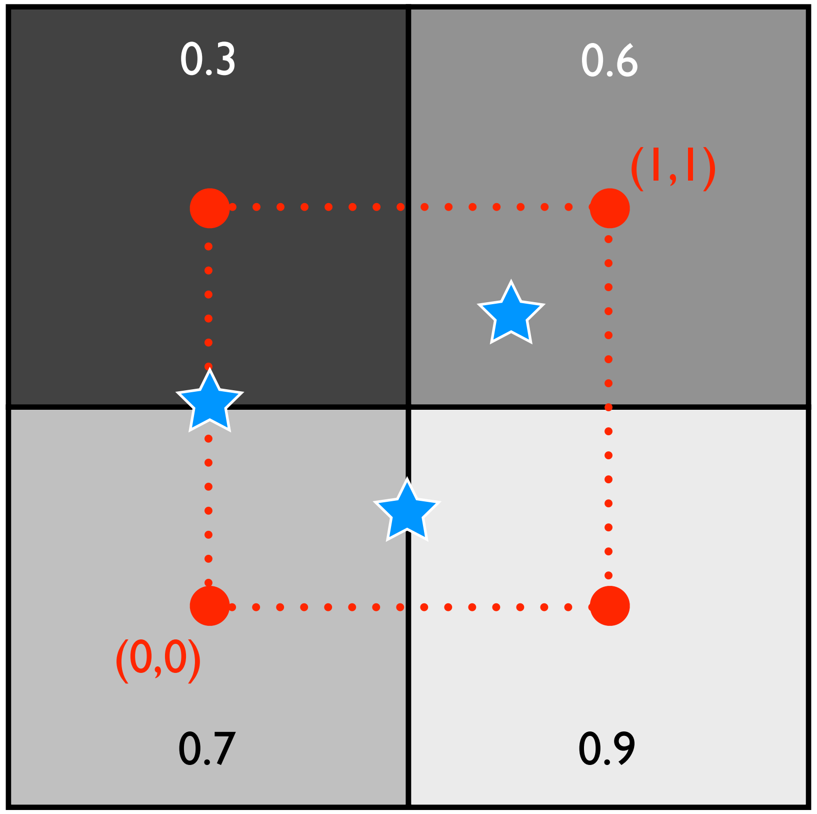
- We only have samples at integer locations:

$$I(x, y, z) = v_{\text{samp}} \quad \text{for} \quad x, y, z \in \mathbb{Z}$$



- How do we create a continuous implicit function, S(x, y, z)?

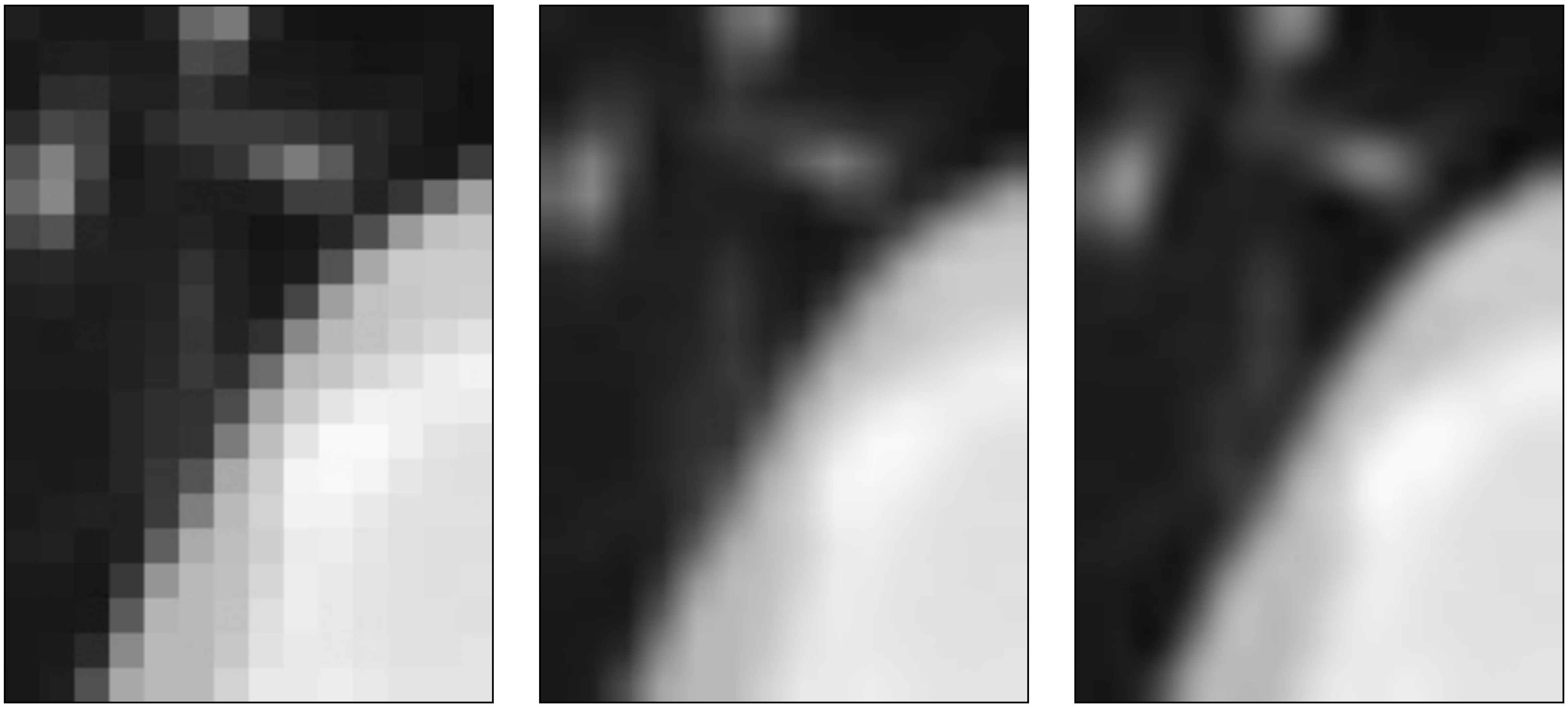
Interpolation



- What's the value here?

$$F(x,y) = (1 - x + \lfloor x \rfloor)(1 - y + \lfloor y \rfloor)\ I(\lfloor x \rfloor, \lfloor y \rfloor)$$
$$+ (x - \lfloor x \rfloor)(1 - y + \lfloor y \rfloor)\ I(\lceil x \rceil, \lfloor y \rfloor)$$
$$+ (1 - x + \lfloor x \rfloor)(y - \lfloor y \rfloor)\ I(\lfloor x \rfloor, \lceil y \rceil)$$
$$+ (x - \lfloor x \rfloor)(y - \lfloor y \rfloor)\ I(\lceil x \rceil, \lceil y \rceil) \quad \text{for} \quad x, y, z \in \mathbb{R}$$

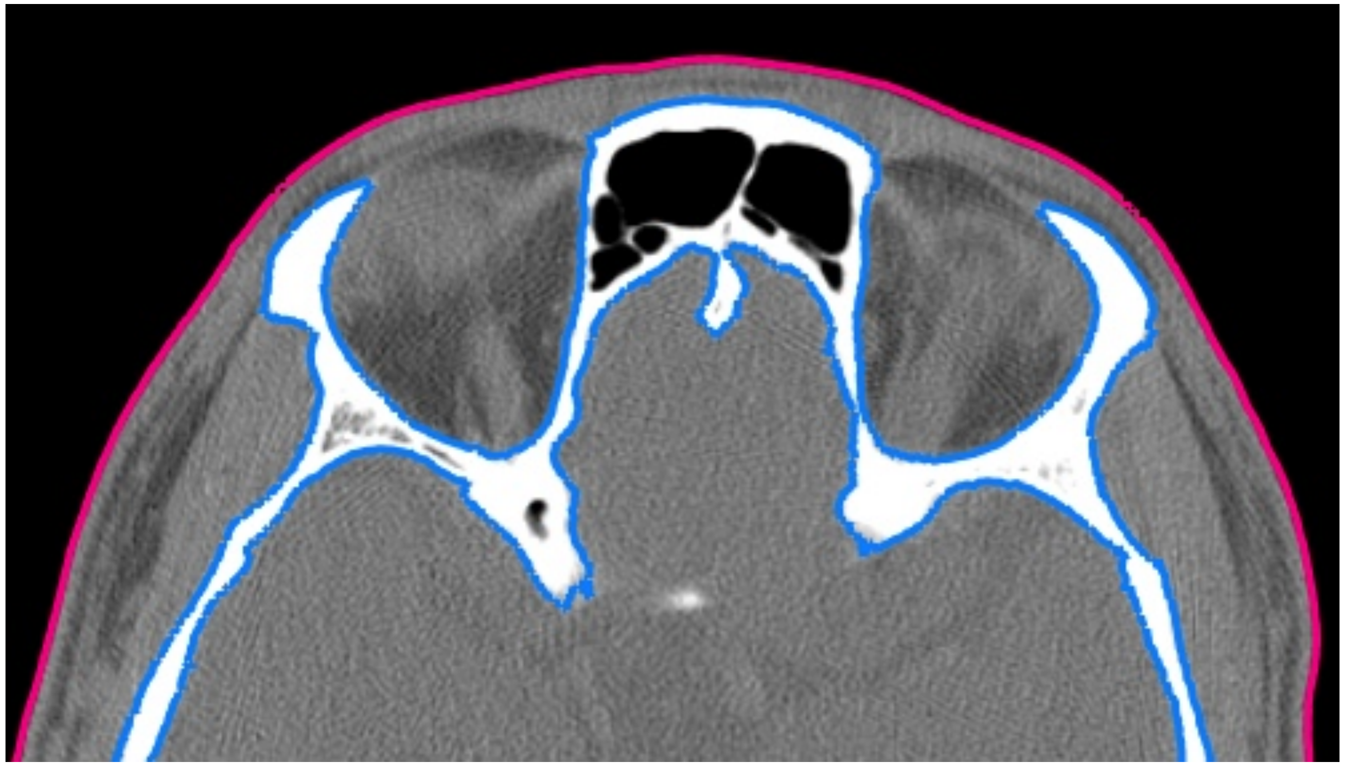
Interpolation Functions



nearest linear sinc

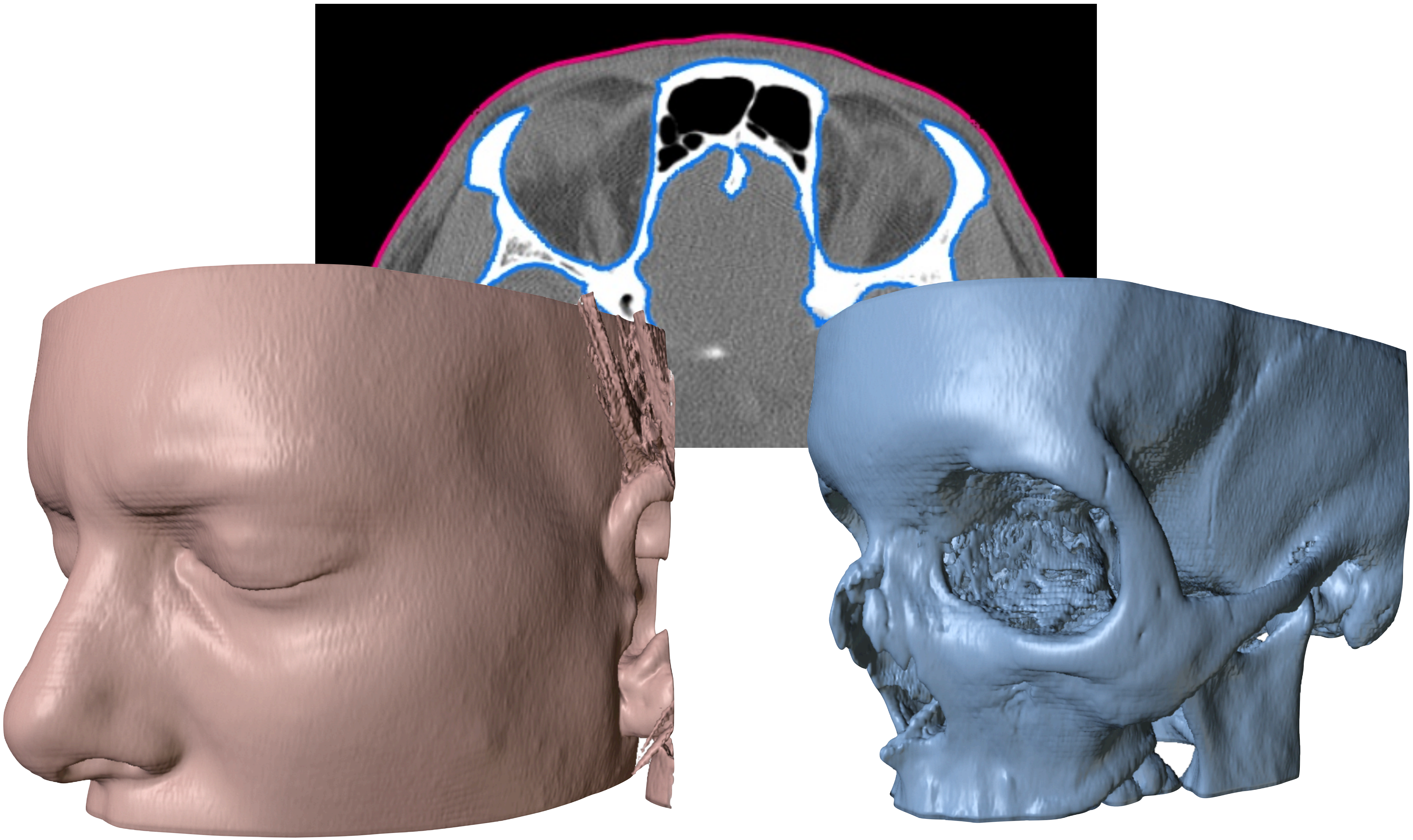
Isocontours & Isosurfaces

- Choose a threshold value, T, to determine surface function: $S(x, y, z) = T - F(x, y, z)$



T = -600 HU   T = 300 HU

Isosurfaces in 3D



Rendering Algorithm

- Our implicit surface rendering algorithm has two specific requirements:

  - Inside-outside function for S(p)

    $$S(x, y, z) = T - F(x, y, z)$$

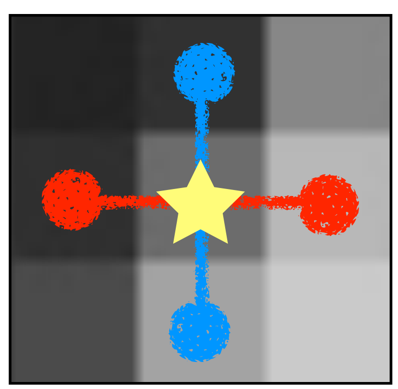
  - The gradient of S(p)

    $$\nabla S(x, y, z) = ???$$

Central Differencing

- Estimate gradient using central difference:

$$\nabla S(x,y) = \begin{pmatrix} \frac{\partial S}{\partial x} \\ \frac{\partial S}{\partial y} \end{pmatrix} \approx \begin{pmatrix} \frac{S(x+\delta,y)-S(x-\delta,y)}{2\delta} \\ \frac{S(x,y+\delta)-S(x,y-\delta)}{2\delta} \end{pmatrix}$$

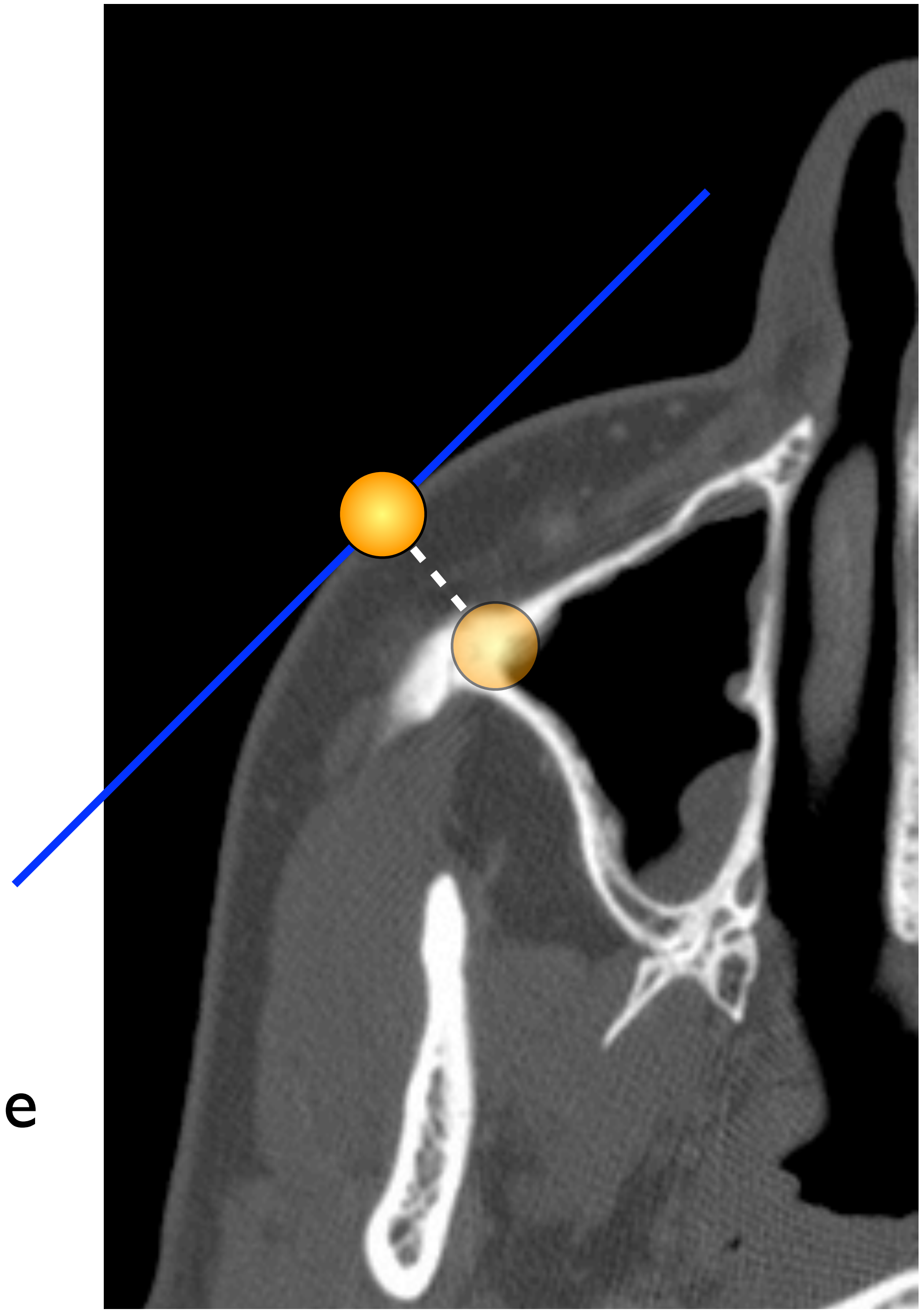


∂S/∂x
∂S/∂y

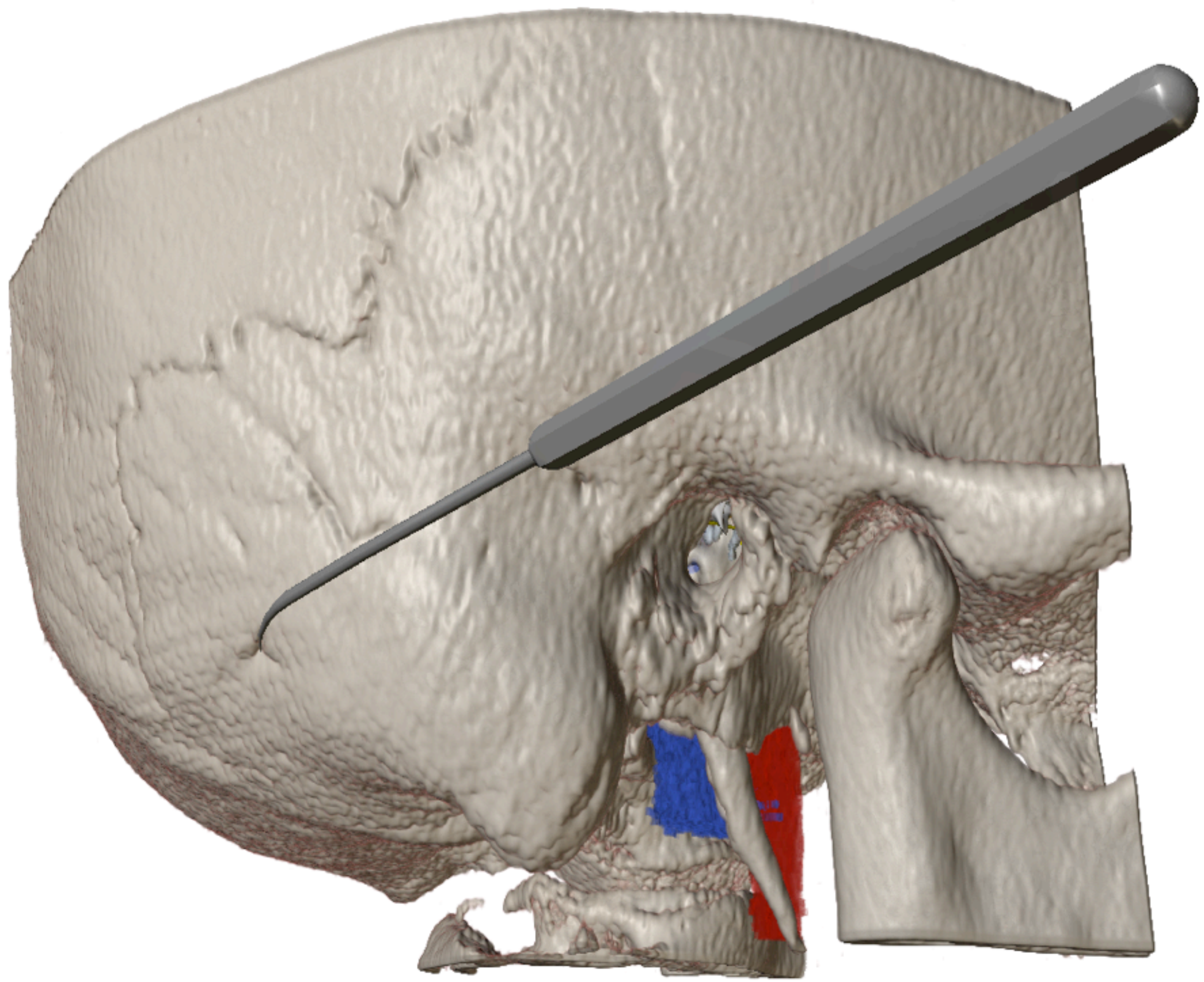
- What's the best choice for the δ value?

Recap Again, Issues?

- The full isosurface rendering algorithm:
  - Detect initial contact when S(p) < 0
  - Find surface point using initial point as seed
  - Update the surface point as the device moves by using the tangent plane as a constraint
  - Contact breaks when device is moved outside the constraining plane
  - Repeat from start...



Demo?



Summary

- Implicit surface rendering algorithm
- Rendering friction
  - Static and kinetic varieties
  - Friction cone
- Rendering volumetric data
  - Using a variant of the implicit surface algorithm