Computing Hard Contacts with heterogeneous Materials for Medical Simulators

Maximilian Kaluschke

Computer Graphics and Virtual Reality, University of Bremen

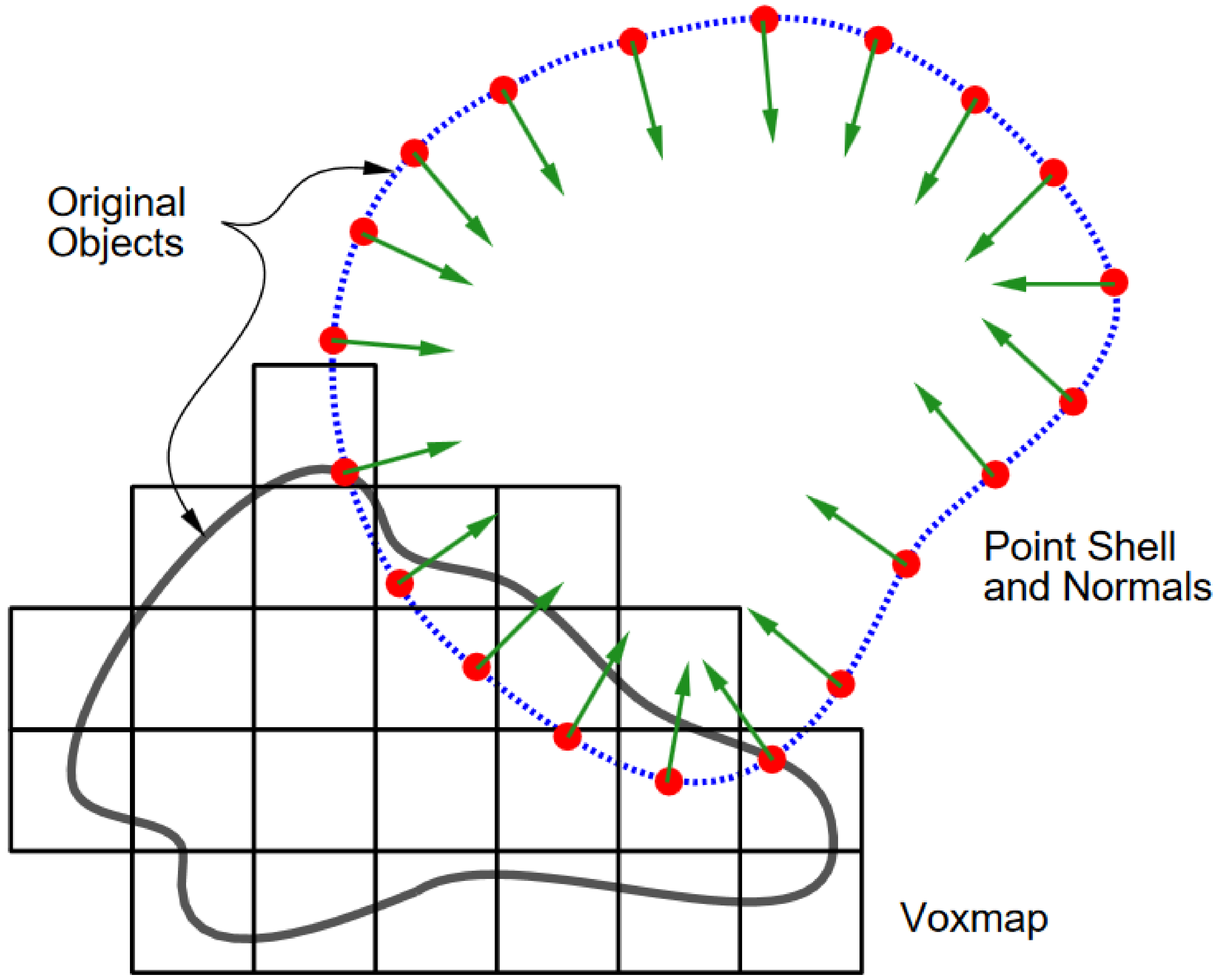
Introduction

- Arthritis: 9% of 20+ y/o $\longrightarrow$ >90% of 65+ y/o
- Hip replacement
  - Prevalence
    - >140.000 procedures in 2016 in Germany
    - Cost-effective quality of life improvement
  - Hip socket milling is difficult to learn
  - Surgeon‘s skill major factor

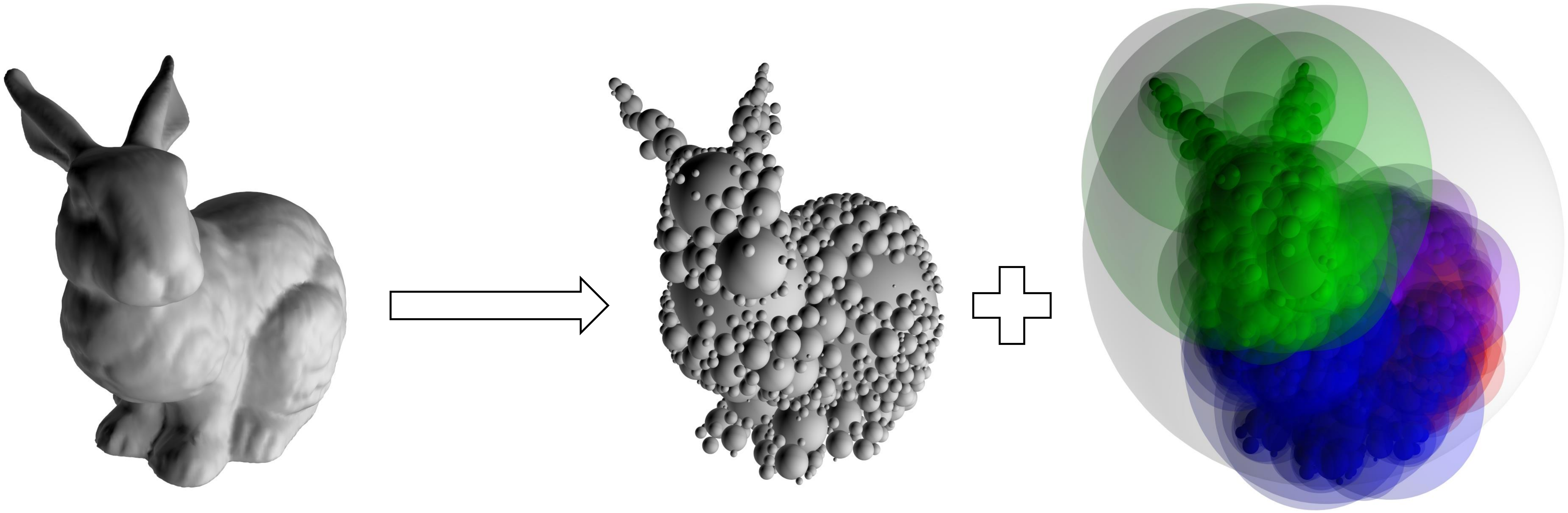
Introduction

Previous Work

- Six Degree-of-Freedom Haptic Rendering Using Voxel Sampling; *McNeely, 1999*
- A GPU-implemented physics-based haptic simulator of tooth drilling; *Razavi, 2015*
- Inner sphere trees and their application to collision detection; *Weller, 2008*



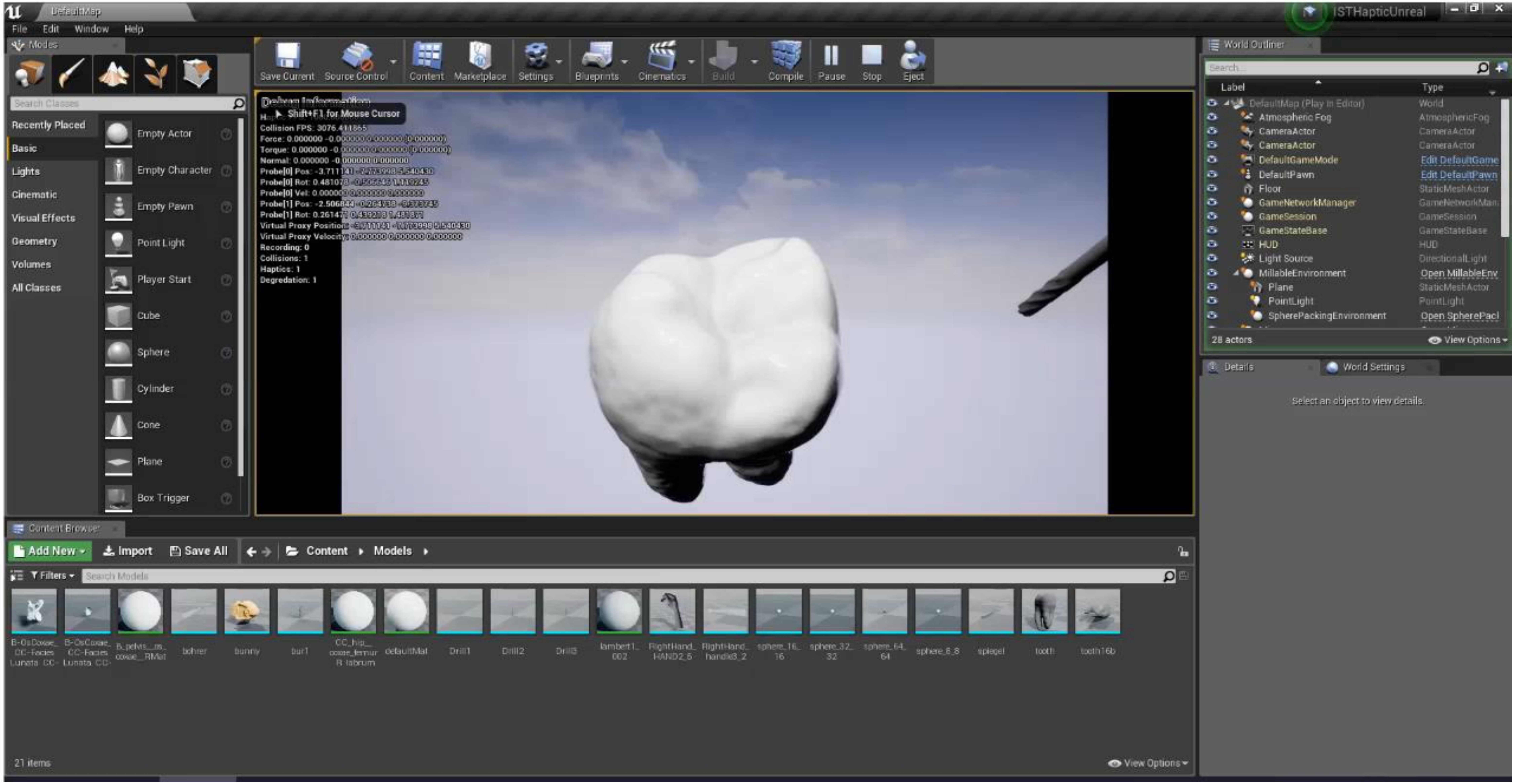
Collision Detection — Inner Sphere Trees



Container Inner spheres Bounding volume hierarchy

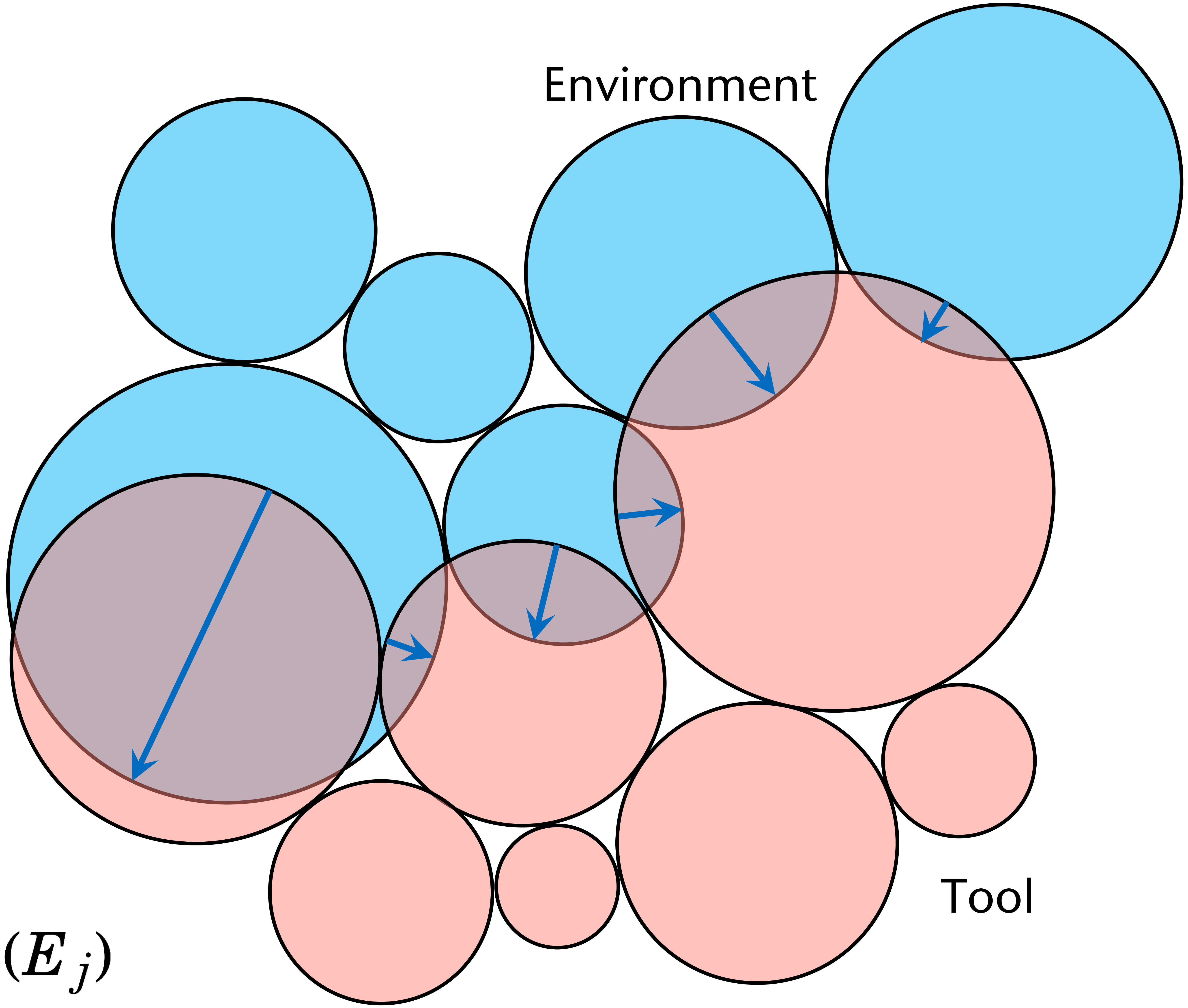
Teaser

Teaser



Collision Detection — Force Computation

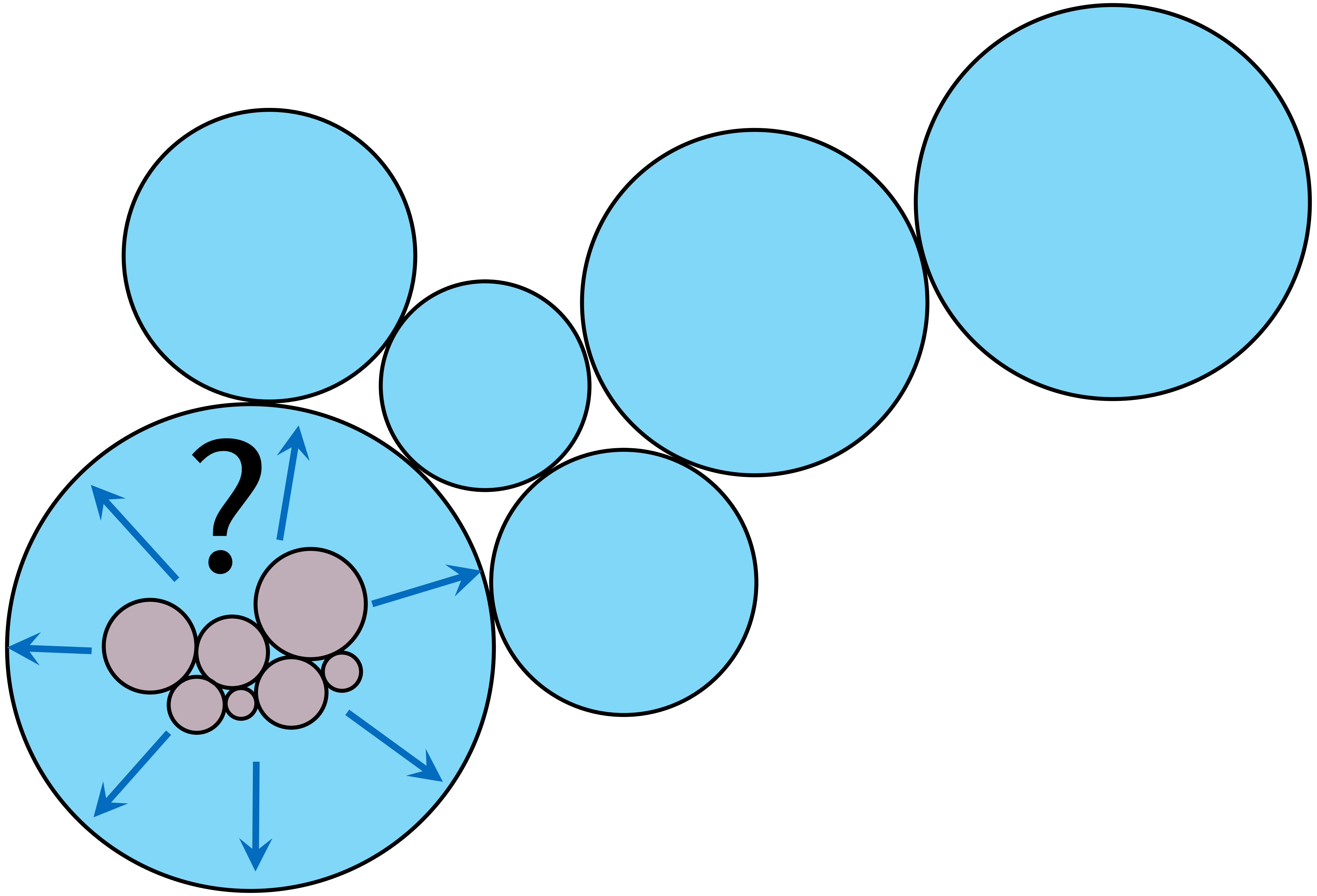
- Penalty-based forces
  - Overlap defines direction and magnitude of force
  - Easy to implement
  - Good performance



$$F_p = k \sum_{T_i \cap E_j \neq \emptyset} \mathrm{Vol}(T_i \cap E_j) \cdot n(E_j)$$

Collision Detection — Force Computation

- Penalty-based forces
  - Overlap defines direction and magnitude of force
  - Easy to implement
  - Good performance
  - Introduces potential problems
    - Solution: Normal cones

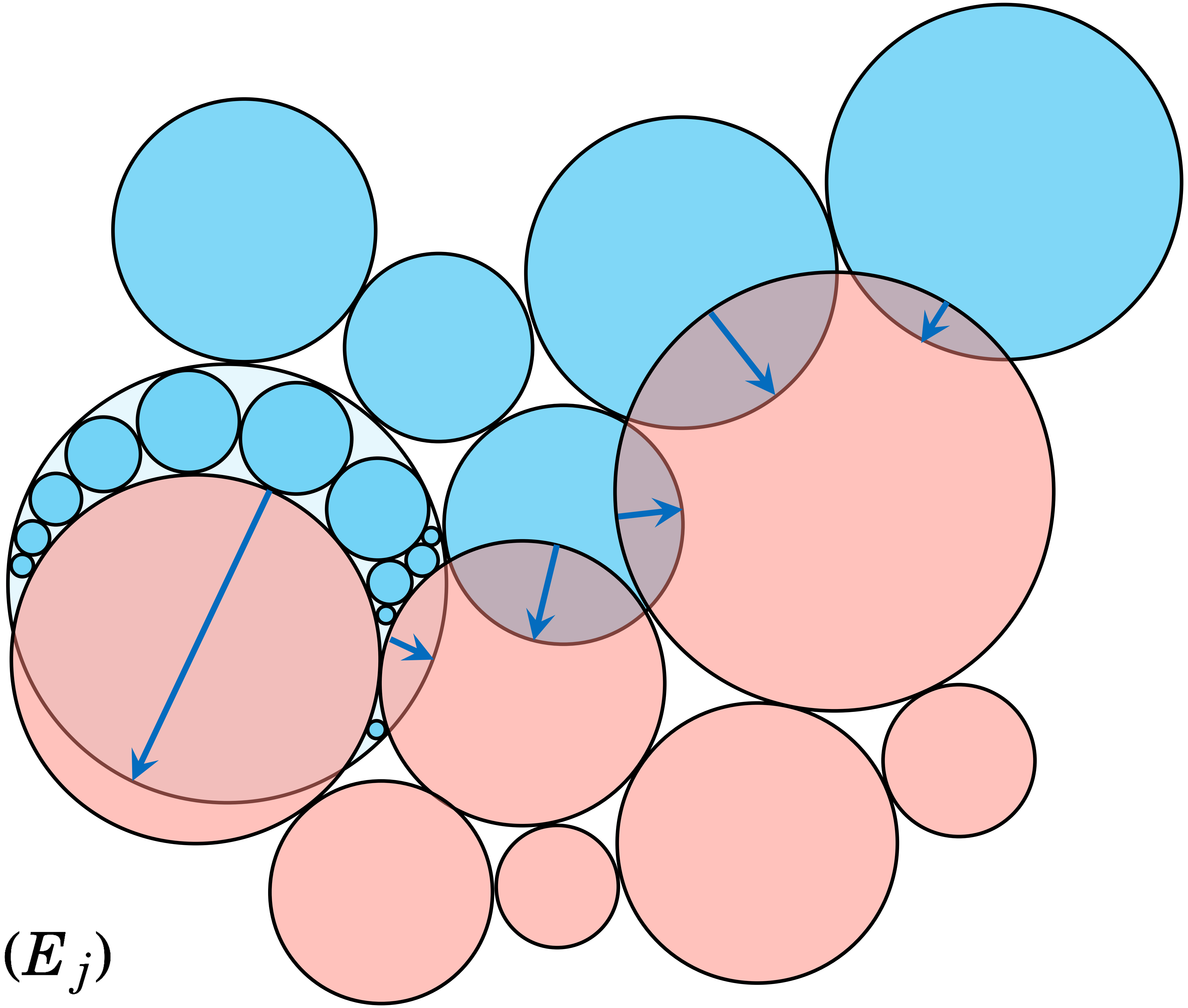


$$F_p = k \sum_{T_i \cap E_j \neq \emptyset} \mathrm{Vol}(T_i \cap E_j) \cdot n(E_j)$$

Collision Detection — Force Computation

- Penalty-based forces
  - Overlap defines direction and magnitude of force
  - Easy to implement
  - Good performance
  - Introduces potential problems
    - Solution: Normal cones
  - Discrete drill simulation

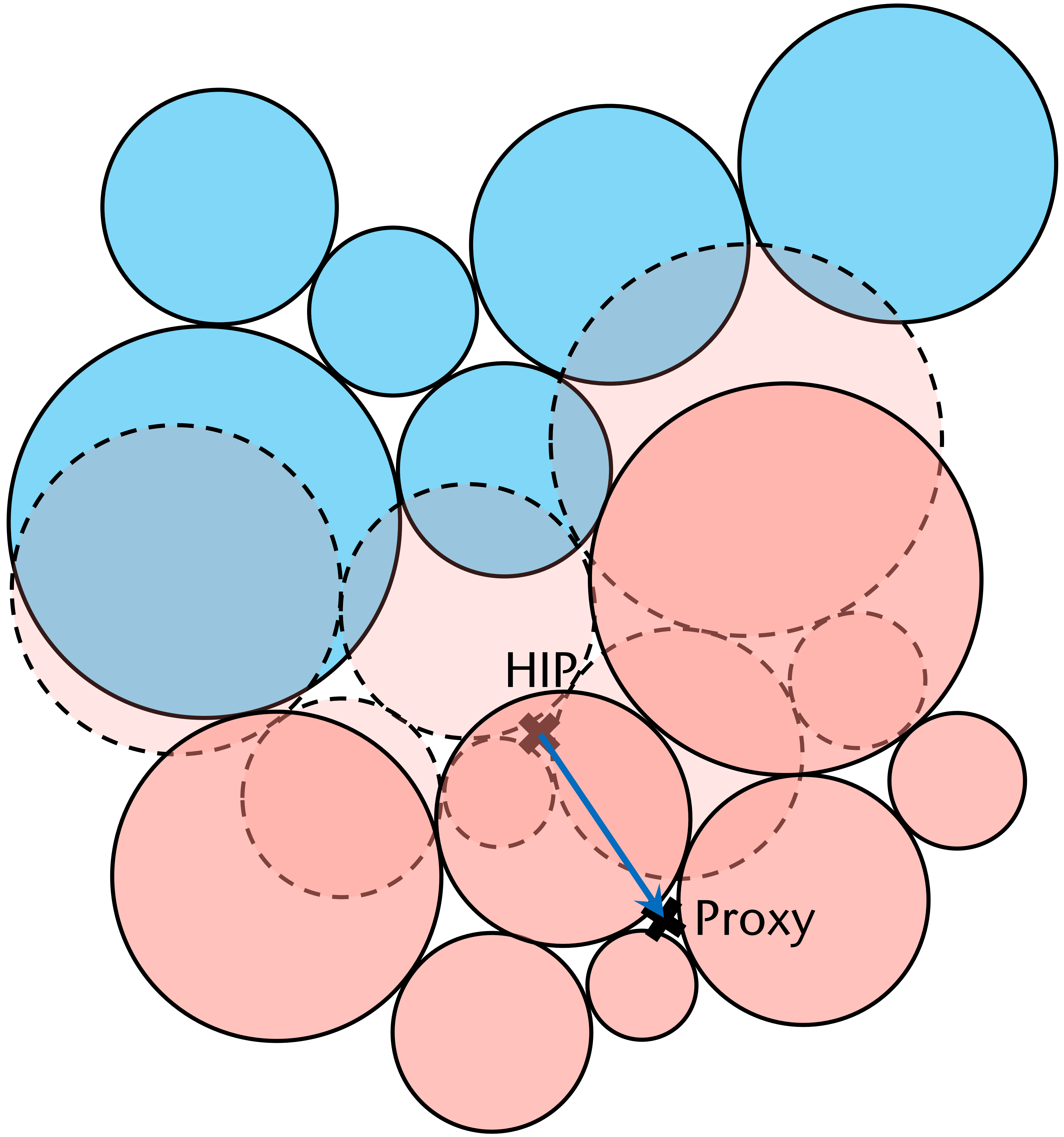
$$F_p = k \sum_{T_i \cap E_j \neq \emptyset} \mathrm{Vol}(T_i \cap E_j) \cdot n(E_j)$$



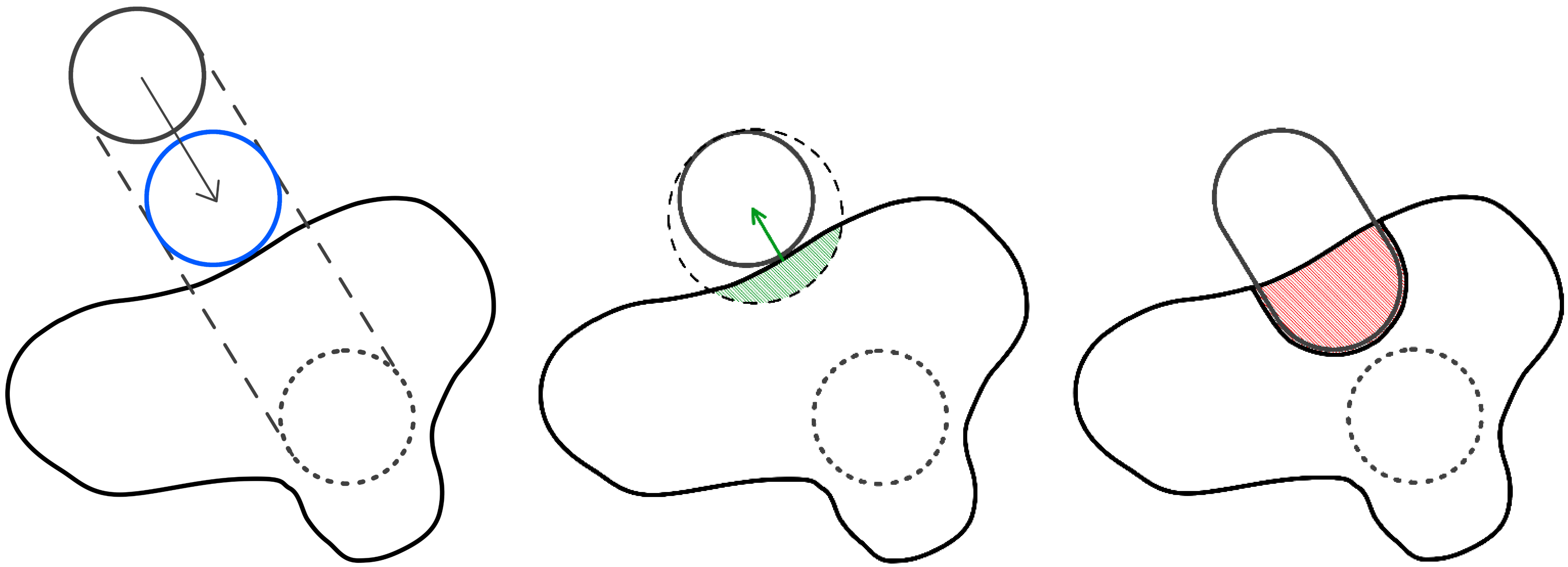
Collision Detection — Force Computation

- Constraint-based forces
  - Well-defined force direction (at all times)
  - Extra instance of tool (called „proxy“)
    - Constraint by environment surface
  - Not *much* harder than penalty method for 3 DoF

$$F_c = k \cdot (P_{Proxy} - P_{HIP})$$



Simulation — Overview



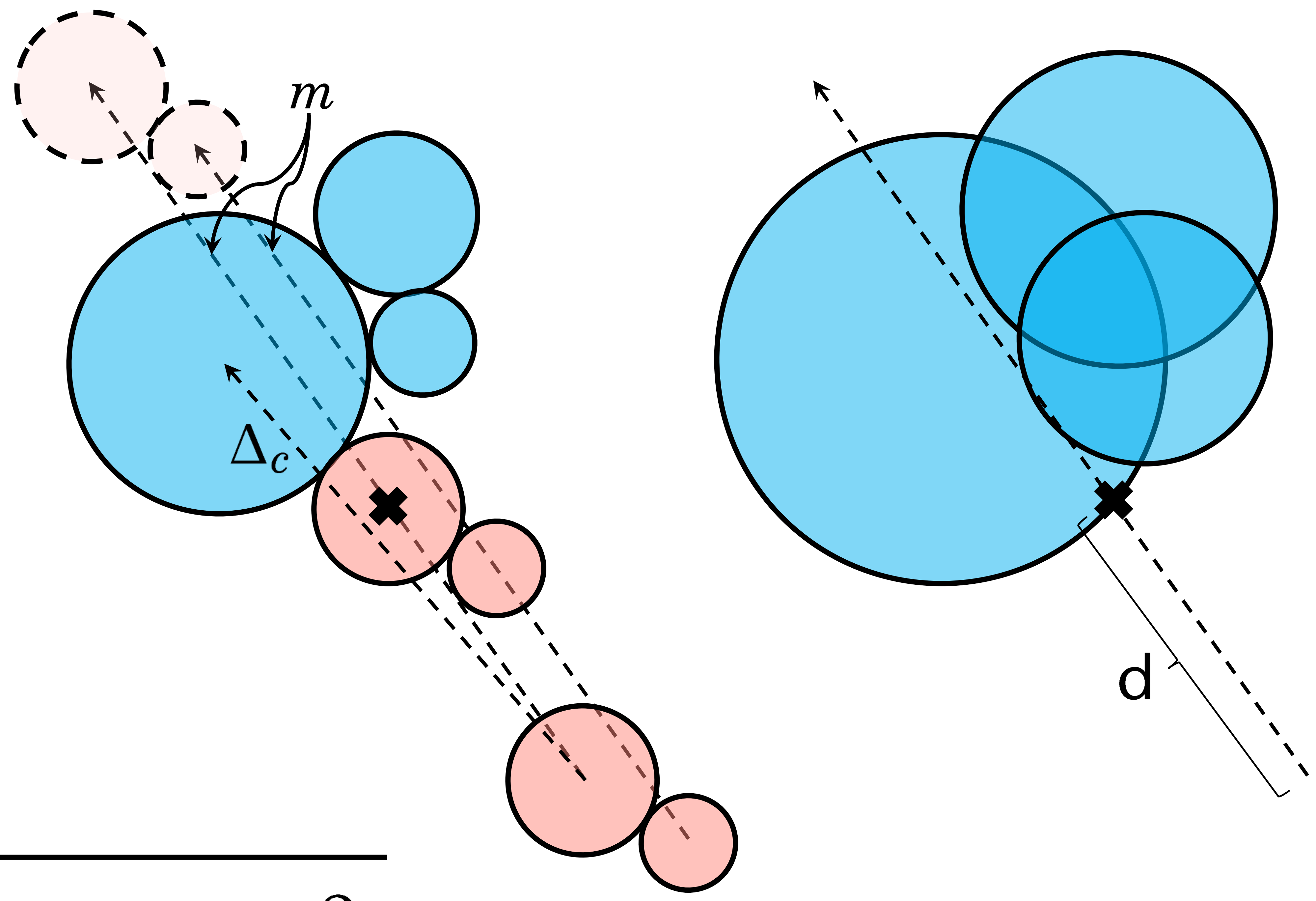
1. Surface contact 2. Surface estimation 3. Drilling

Simulation — 1. Surface Contact

- Regard proxy ⟶ HIP as movement
- Continuous collision detection
  - For individual spheres
  - Reduce to line-sphere intersection

$$d_{t,e} = \frac{2(\Delta_c \cdot m) - \sqrt{4(m\Delta_c)^2 - 4m^2(\Delta_c^2 - (r_t + r_e)^2)}}{2m^2}$$

  - Global minimum of d = constraint movement



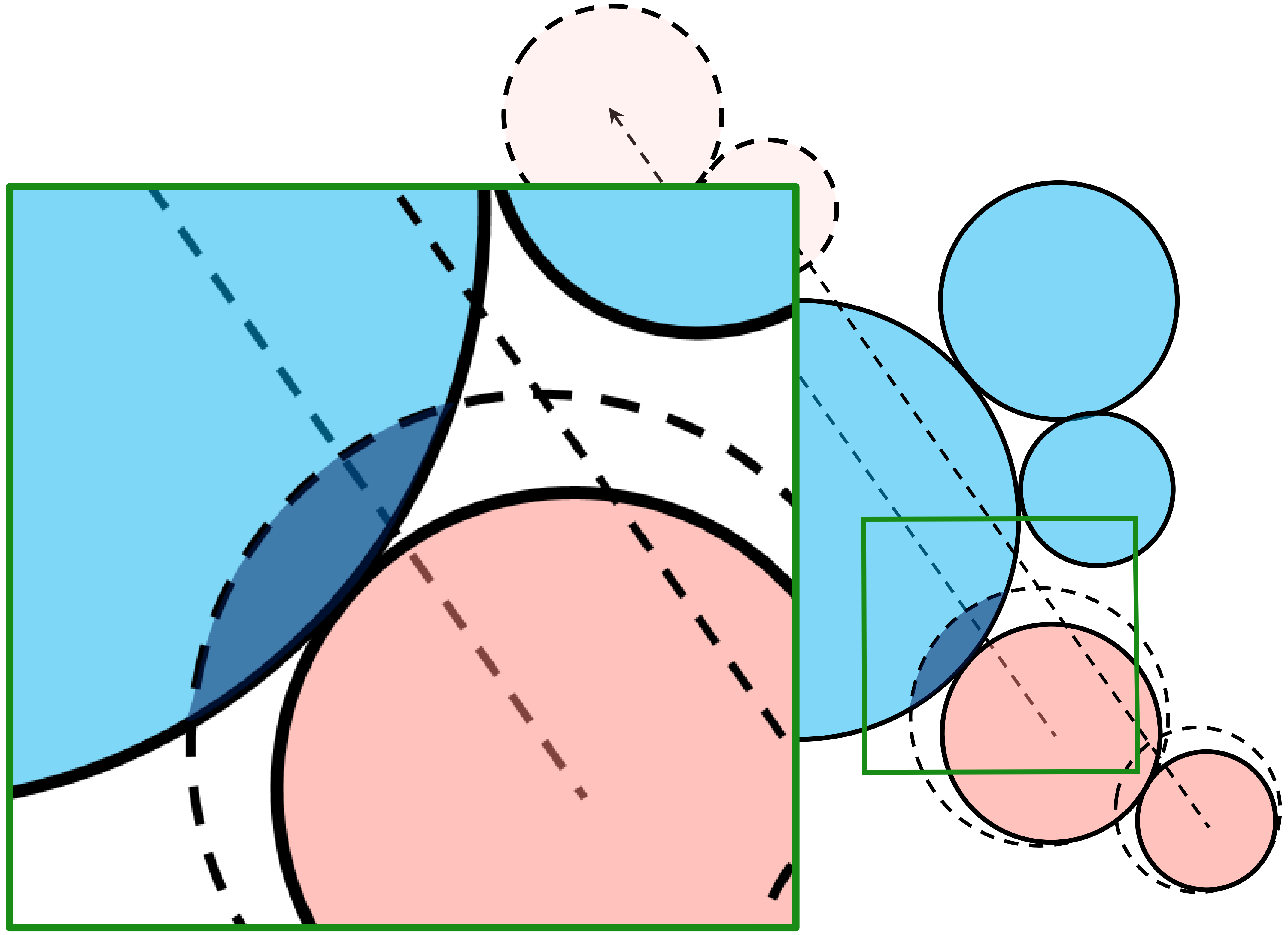
Simulation — 2. Surface Estimation

- Store material properties in spheres
  - Material density and friction

$$\vec{n} = \frac{\sum_{T_i \cap E_j \neq \emptyset} \mathrm{Vol}(T_i \cap E_j) \cdot n(E_j)}{\sum_{T_i \cap E_j \neq \emptyset} \mathrm{Vol}(T_i \cap E_j)}$$

$$h = \sum_{T_i \cap E_j \neq \emptyset} \mathrm{Vol}(T_i \cap E_j) \cdot h(E_j)$$

$$F_c := hF_c$$



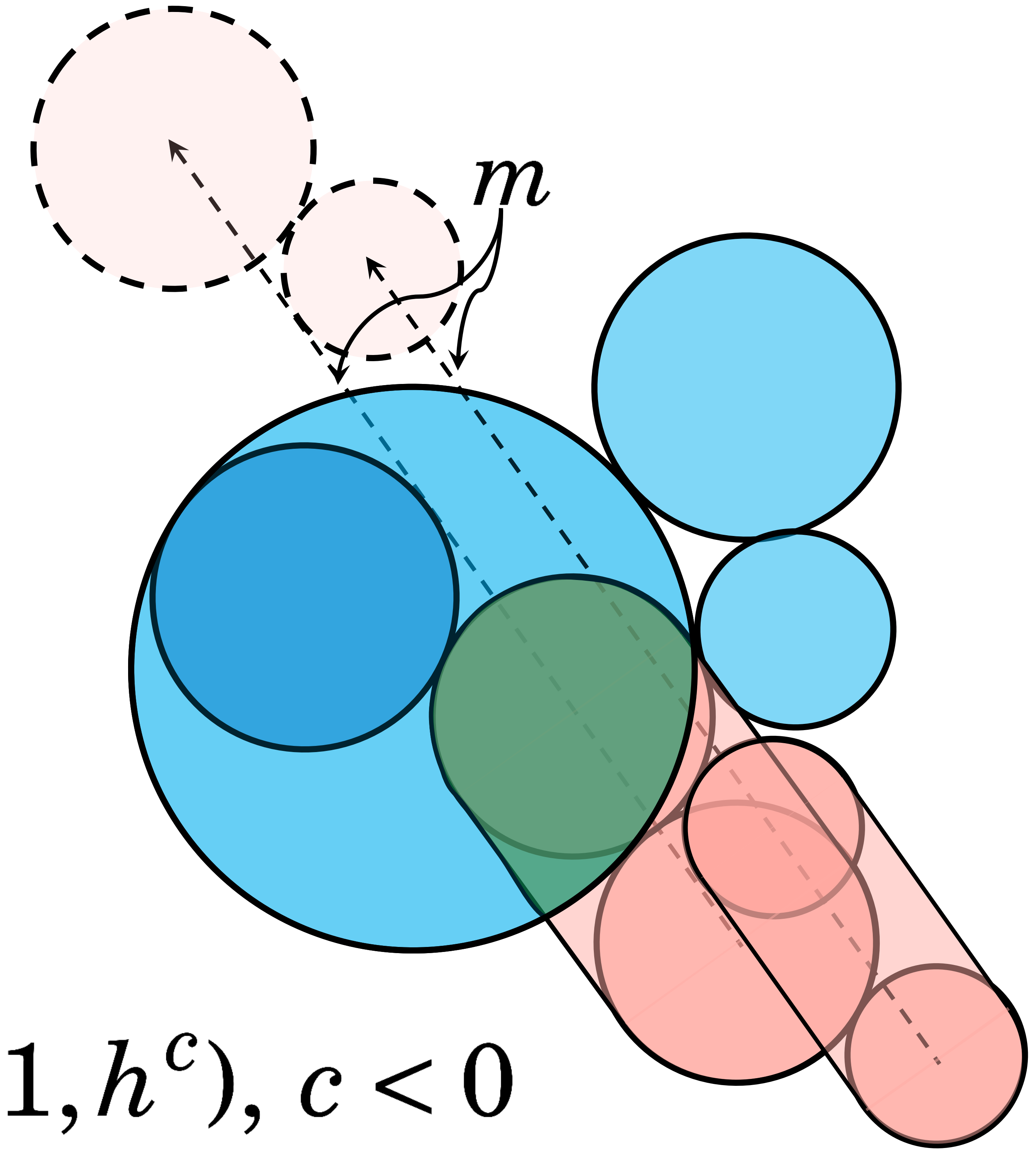
- Coulomb friction
  - Restrict lateral movment unless:

$$\vec{n}(F_c \cdot \vec{n})\mu_s < F_c - (\vec{n} \cdot F_c)$$

$$\vec{n}(F_c \cdot \vec{n})\mu_k < F_c - (\vec{n} \cdot F_c)$$

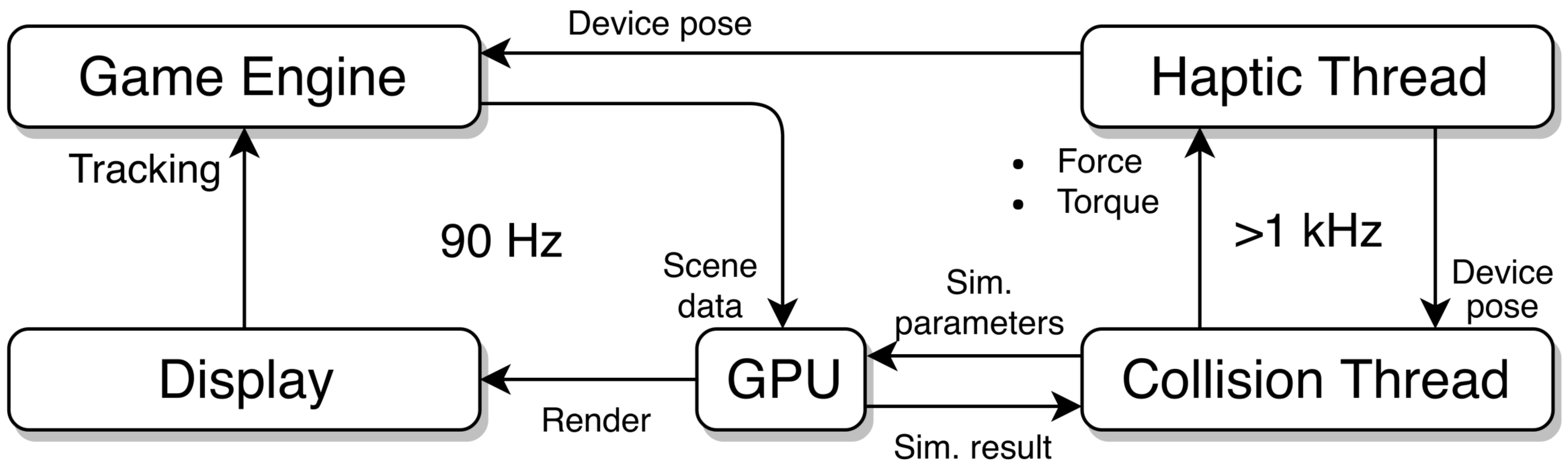
Simulation — 3. Drilling

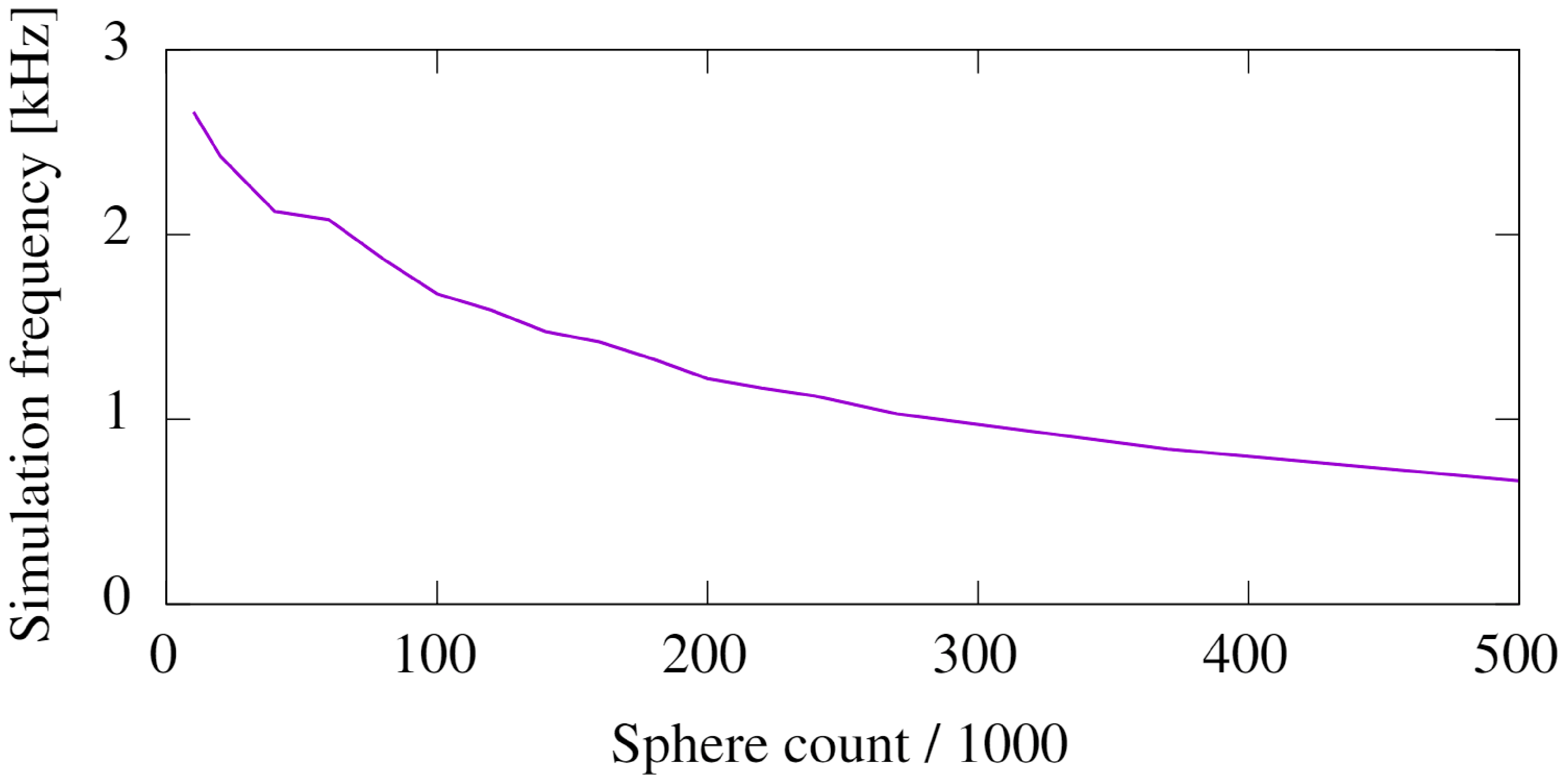
- Start drilling at new proxy (Pass 1)
- Estimate frame time
- Tool's drill speed v_d
  - Scale with penetation depth updated m (Pass 1)
  - Scale inversely with density (Pass 2) $v_d := v_d \cdot \min(1, h^c),\ c < 0$
- Volume updates
  - Small spheres are shrunk
  - Large spheres are subdivided



Results

- Dynamic library
  - GPU accelerated implementation
  - Game engine plugins
- Simulation performance
  - Interaction recording
  - 1 kHz for nearly 300.000 spheres





Conclusion & Future Work

- Simulation method with continuous drilling & stable haptic feedback
  - Fast implementation
  - Integration in popular game-engines
- Approximate 6 DoF constraints