CLOTH SIMULATION

Graphics Course Final Project 2015-1

BHBros. @POSTECH

20090402 이다열

20100472 김태규

20100234 최준혁

Before the interim presentation, we did …

- Mass-Spring Model
  - ☑Structural forces
  - ☑Internal deformation forces
  - ☑External forces
- Collision
  - ☐External collision
  - ~~☒Internal collision~~
- Improvement
  - ☑Shading
  - ☑Texture mapping
  - ☑Normal mapping
  - ☐Deformation constraint
  - ☐Friction
  - ☐Other objects

How to represent cloth?

- Mass-Spring Model
  - spring force

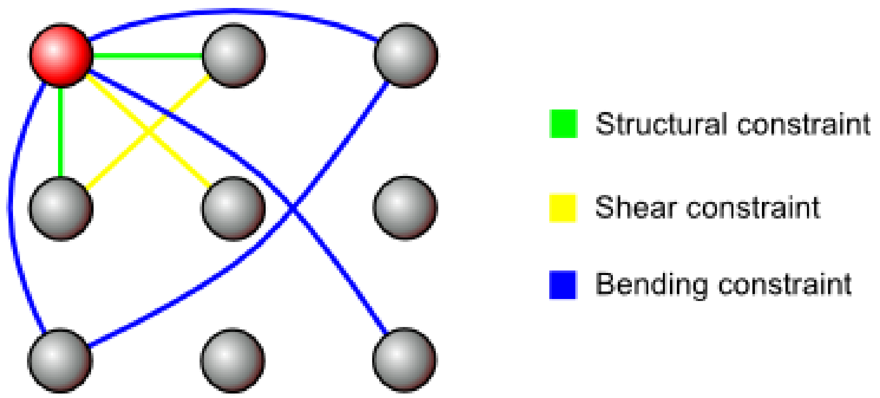
$$\mathbf{f}_i = \mathbf{f}^s(\mathbf{x}_i, \mathbf{x}_j) = k_s \frac{\mathbf{x}_j - \mathbf{x}_i}{|\mathbf{x}_j - \mathbf{x}_i|}(|\mathbf{x}_j - \mathbf{x}_i| - l_0)$$

  - damping force

$$\mathbf{f}_i = \mathbf{f}^d(\mathbf{x}_i, \mathbf{v}_i,\ \mathbf{x}_j, \mathbf{v}_j) = k_d(\mathbf{v}_j - \mathbf{v}_i) \cdot \frac{\mathbf{x}_j - \mathbf{x}_i}{|\mathbf{x}_j - \mathbf{x}_i|}$$

  - total force

$$\mathbf{f}(\mathbf{x}_i, \mathbf{v}_i,\ \mathbf{x}_j, \mathbf{v}_j) = \mathbf{f}^s(\mathbf{x}_i, \mathbf{x}_j) + \mathbf{f}^d(\mathbf{x}_i, \mathbf{v}_i,\ \mathbf{x}_j, \mathbf{v}_j)$$



How to represent cloth?

- Particle position (from F=ma)

$$\mathbf{v}(t) = \mathbf{v}_0 + \int_{t_0}^{t} \mathbf{f}(t)/m \, dt$$
$$\mathbf{x}(t) = \mathbf{x}_0 + \int_{t_0}^{t} \mathbf{v}(t) dt.$$

- Discrete time step (approximately)

$$\mathbf{v}^{t+1} = \mathbf{v}^t + \Delta t \, \mathbf{f}(\mathbf{x}^t, \mathbf{v}^t)/m$$
$$\mathbf{x}^{t+1} = \mathbf{x}^t + \Delta t \, \mathbf{v}^t.$$

How to represent cloth?

- Particle position (from F=ma)

$$\mathbf{v}(t) = \mathbf{v}_0 + \int_{t_0}^{t} \mathbf{f}(t)/m \, dt$$

$$\mathbf{x}(t) = \mathbf{x}_0 + \int_{t_0}^{t} \mathbf{v}(t) dt.$$

- Discrete time step (approximately)

$$\mathbf{v}^{t+1} = \mathbf{v}^t + \Delta t \, \mathbf{f}(\mathbf{x}^t, \mathbf{v}^t)/m$$

$$\mathbf{x}^{t+1} = \mathbf{x}^t + \Delta t \, \mathbf{v}^t.$$

Do we have to store all of these values?

Do we have to update new velocity every time?

How to represent cloth?

- Verlet Integration
  - Simplest and popular for real-time applications
  - Tailor expansion of $x(t-\Delta\mathrm{t})$, $x(t+\Delta\mathrm{t})$

$$\mathbf{x}(t+\Delta t) = \mathbf{x}(t) + \dot{\mathbf{x}}(t)\Delta t + \frac{1}{2}\ddot{\mathbf{x}}(t)\Delta t^2 + \frac{1}{6}\dddot{\mathbf{x}}(t)\Delta t^3 + O(\Delta t^4)$$

$$\mathbf{x}(t-\Delta t) = \mathbf{x}(t) - \dot{\mathbf{x}}(t)\Delta t + \frac{1}{2}\ddot{\mathbf{x}}(t)\Delta t^2 - \frac{1}{6}\dddot{\mathbf{x}}(t)\Delta t^3 + O(\Delta t^4)$$

Adding two yields

$$\begin{aligned}\mathbf{x}(t+\Delta t) &= 2\mathbf{x}(t) - \mathbf{x}(t-\Delta t) + \ddot{\mathbf{x}}(t)\Delta t^2 + O(\Delta t^4) \\ &= \mathbf{x}(t) + [\mathbf{x}(t) - \mathbf{x}(t-\Delta t)] + \mathbf{f}(t)\Delta t^2/m + O(\Delta t^4).\end{aligned}$$

- Finally, we get

$$x^{t+1} = x^t + v^t\Delta t + f(x^t)\Delta t^2/m$$

where $\quad v^t = (x^t - x^{t-1})/\Delta t.$

Keep only x^{t-1}

Implementation (1)

- Based on our assignment…
    - Camera
    - Model-View stack
    - Object class
    - Shading
    - Texture Mapping
    - Normal Mapping

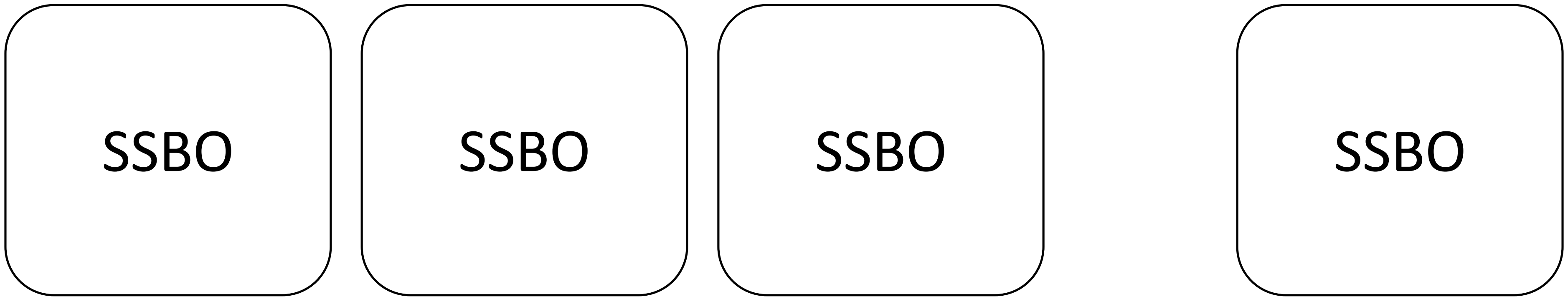
Implementation (2)

- Compute Shader
  - OpenGL >= 4.3
  - Computing arbitrary information

- Shader Storage Buffer Object (SSBO)
  - slower but large, writable buffer

Implementation (3)

- Used four SSBOs



position * 2 + output normal

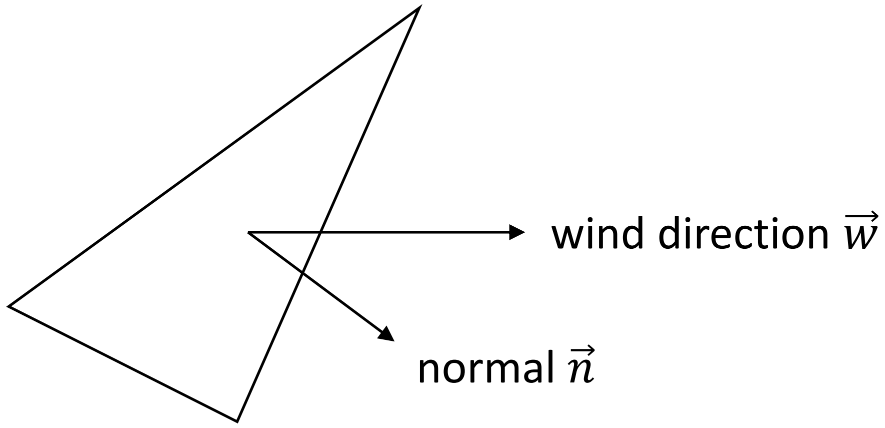
- Role Rotation
  - previous position, current position, next position(output)
  - the output will be used twice as an input

Implementation (4)

- Shader Programs
  - Floor shader: to render simple objects
  - Compute shader: to calculate particle position and normal
  - Render shader: to render the cloth
- Calculating Position
  - force = internal forces + gravity force + wind force
  - verlet integration to get next position
- Calculating Normal
  1. Calculate tangent, bitangent vector from <u>adjacent vertices</u>
  2. Calculate normal by cross product of tangent and bitangent.

Implementation (5)

- Wind force

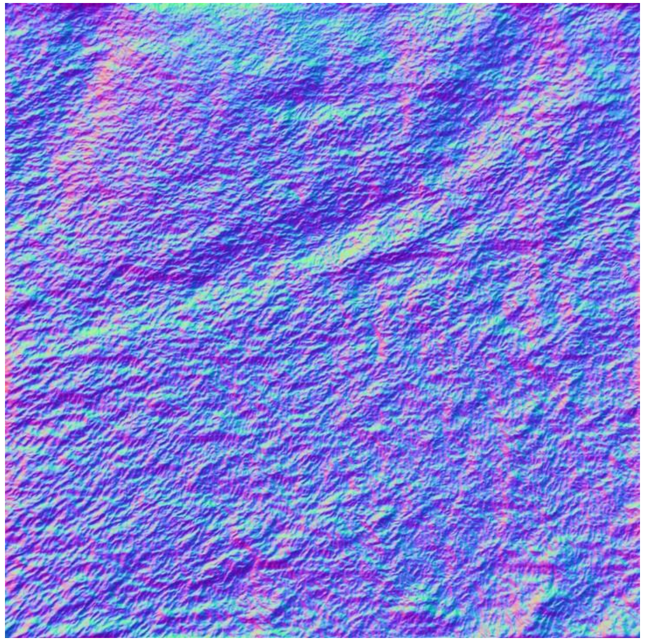


- $f_w = c_{rand} k_w \vec{w} \cdot \vec{n} + d_w$

Implementation (6)

- Texture mapping and normal mapping (wiper)

Color Map Texture



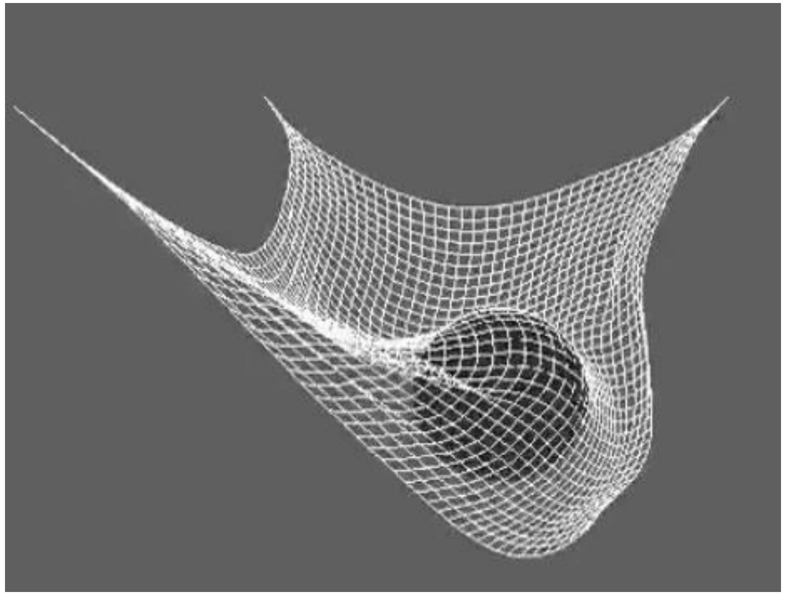
Normal Map Texture

To Do List (for final presentation!)

- Mass-Spring Model
  - ☑ Structural forces
  - ☑ Internal deformation forces
  - ☑ External forces
- Collision
  - ☑ External collision
  - ~~☒ Internal collision~~
- Improvement
  - ☑ Shading
  - ☑ Texture mapping
  - ☑ Normal mapping
  - ☑ Deformation constraint
  - ☑ Friction
  - ☑ Other objects
  - ☑ Simple shadow
  - ☑ Skybox mapping

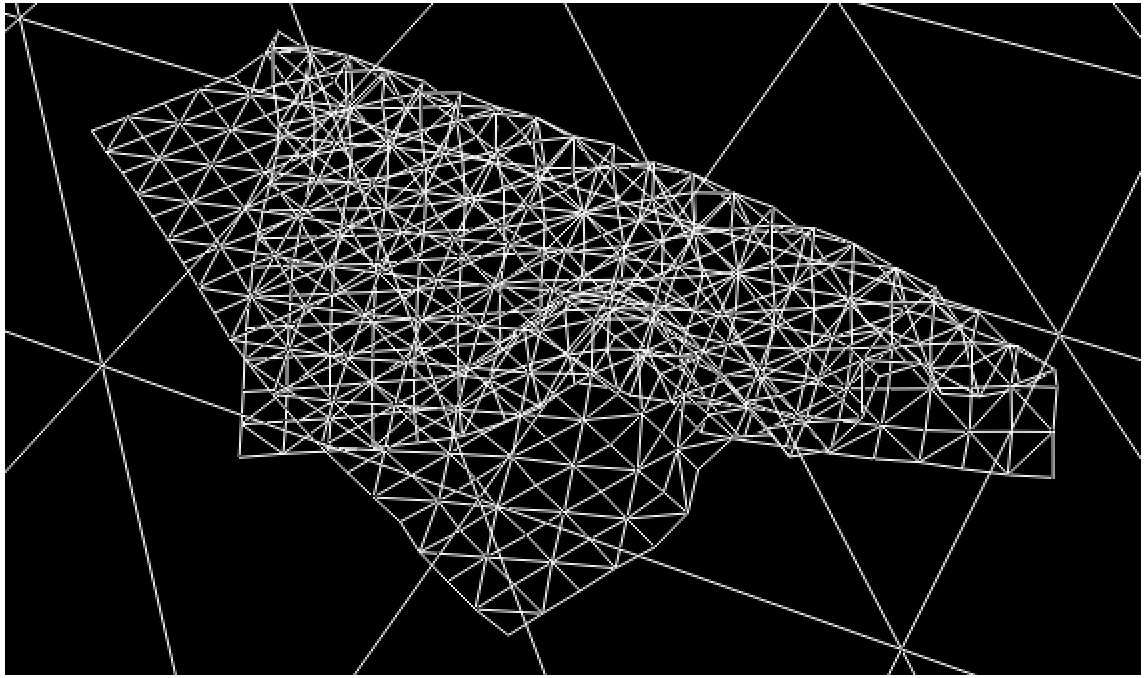
Collision Detection

- Very common challenge: How to detect & respond collision?
- **Sphere**, **Floor**, ~~Square, arbitrary polyhedron…, Deformable, multiple, hierarchical…~~



Floor Detection

- Floor: super easy!
- $if\ x(t + \Delta t) > floorZ\ then\ x(t + \Delta t) = x(t)$



Sphere Detection

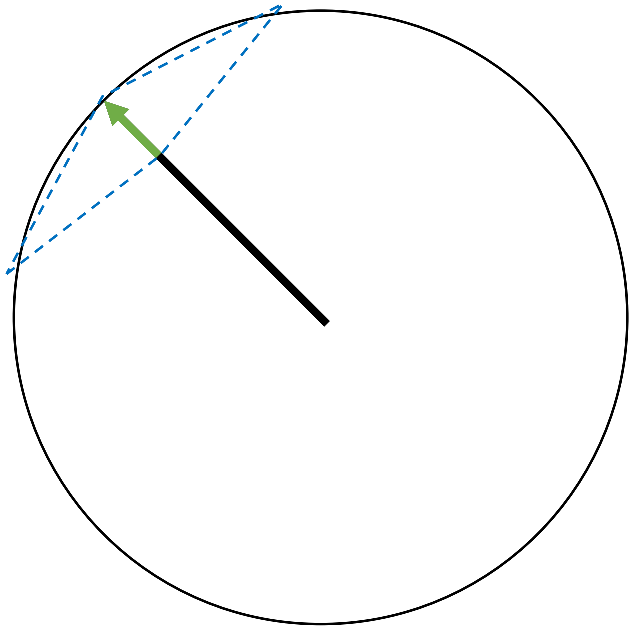
- Sphere center P, radius r

$$|x(t) - P| < r$$

- Response

$$\delta = \frac{x(t)-P}{|x(t)-P|}(r - |x(t) - P|) = \frac{x(t)-P}{|x(t)-P|}\epsilon$$

$$x(t) = x(t) + \delta$$



Sphere Detectio

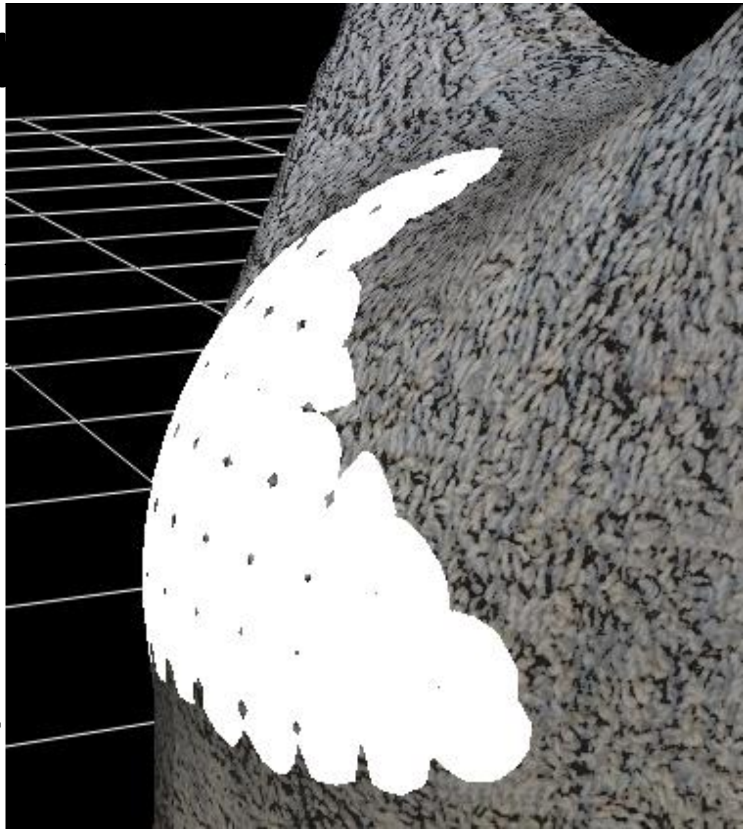
- Sphere center P, radius

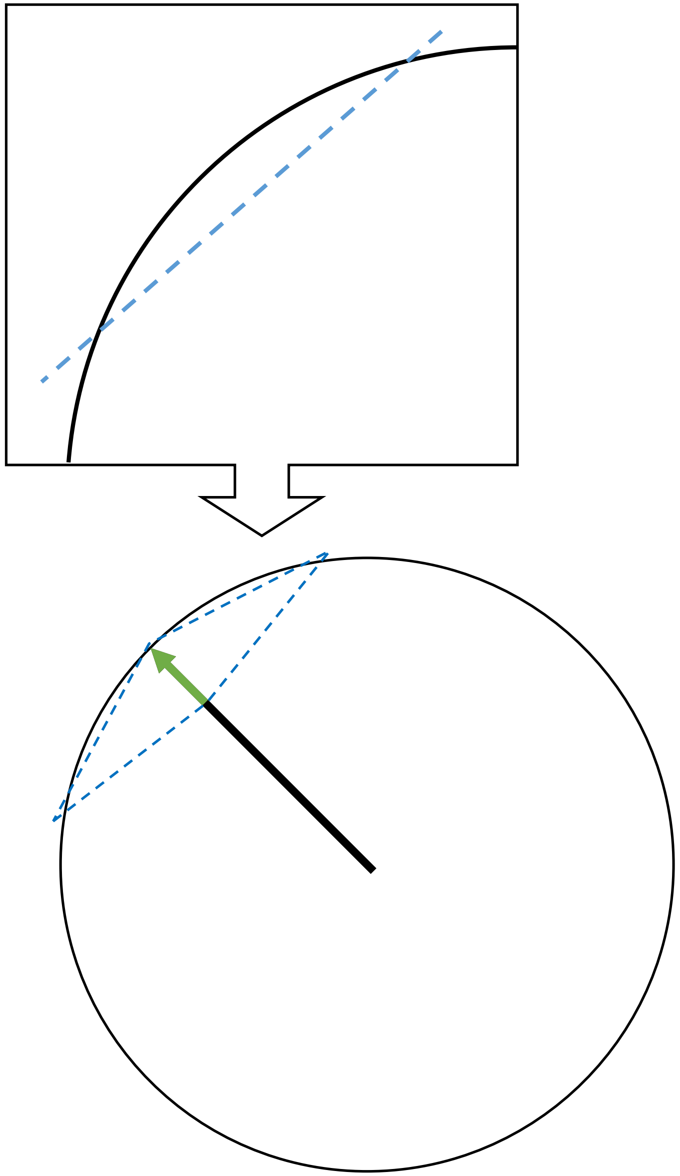
$|x(t) - P| < r$

- Response

$\delta = \frac{x(t)-P}{|x(t)-P|}(r - |x(t) -$

$x(t) = x(t) + \delta$





Sphere Detection

- Sphere center P, radius r

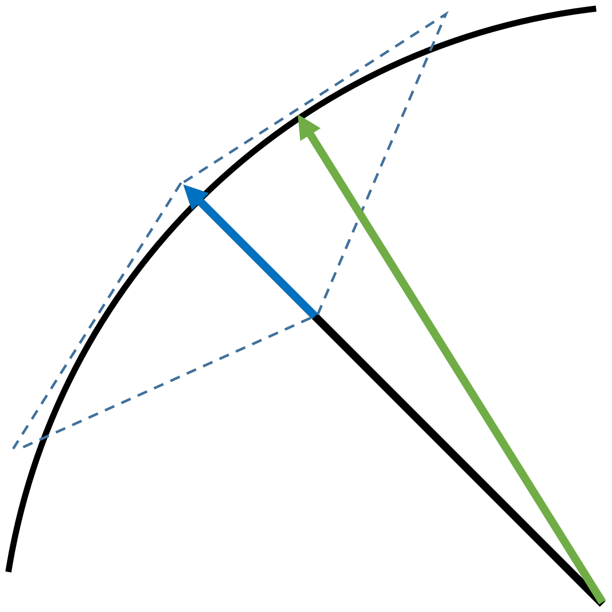
$$|x(t) - P| < \alpha$$

$$\alpha = \sqrt{(l_0)^2 + r^2}$$

- Response

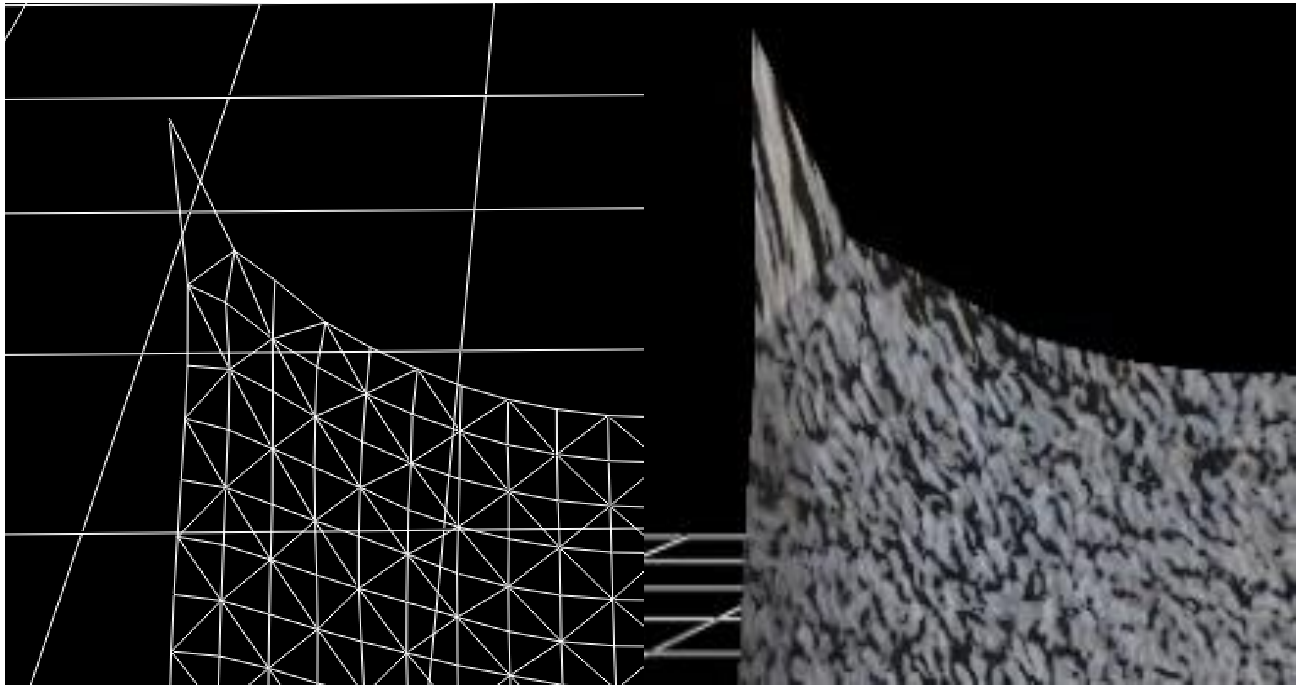
$$\delta = \frac{x(t)-P}{|x(t)-P|}(\alpha - |x(t) - P|) = \frac{x(t)-P}{|x(t)-P|}\epsilon_\alpha$$

$$x(t) = x(t) + \delta$$



Deformation Constraint

- Super-elongated spring (Super-elastic effect)
- Real cloth: limited elongation and shearing (about 10%)



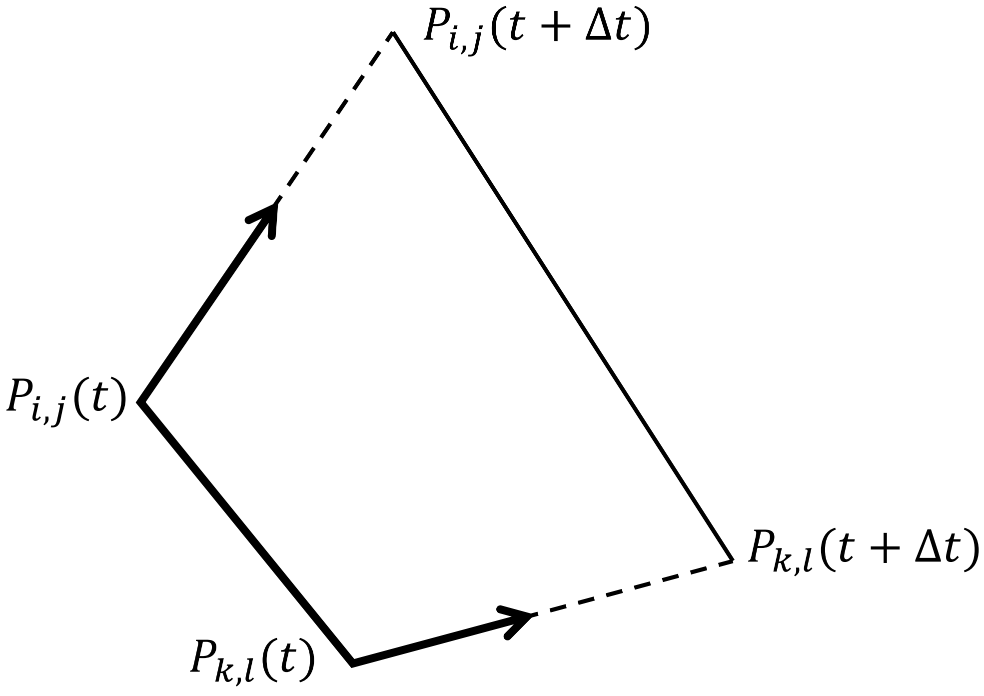
Deformation Constraint(1)

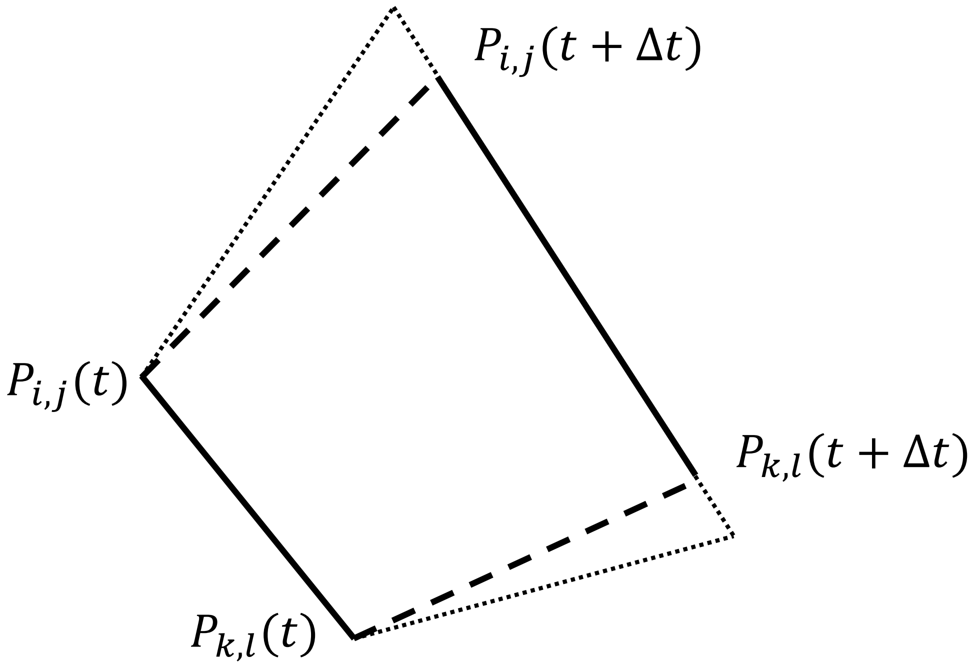
- Dynamic Inverse Constraints on Deformation Rates
  - Proposed by Xavier Provot (1995)

- If the deformation rate of a spring is greater than a critical deformation rate τ_c $(i.e.\ 0.1)$, adjust "*dynamic inverse procedure*"

Deformation Constraint(2)

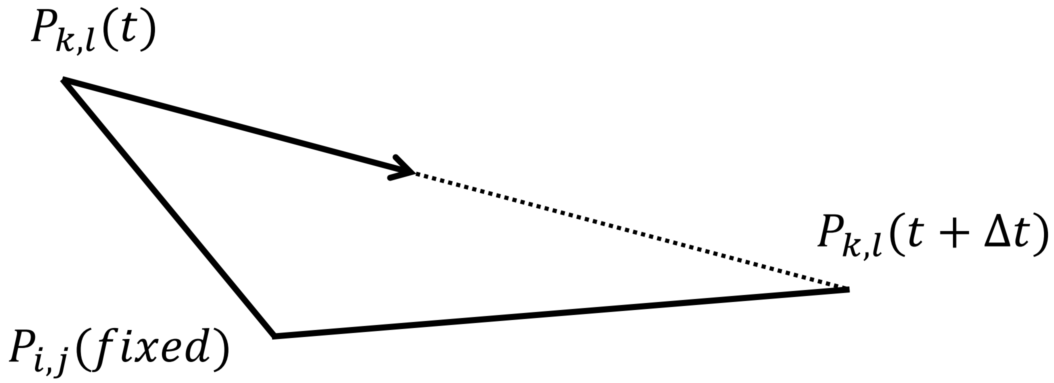
- Dynamic Inverse Procedure

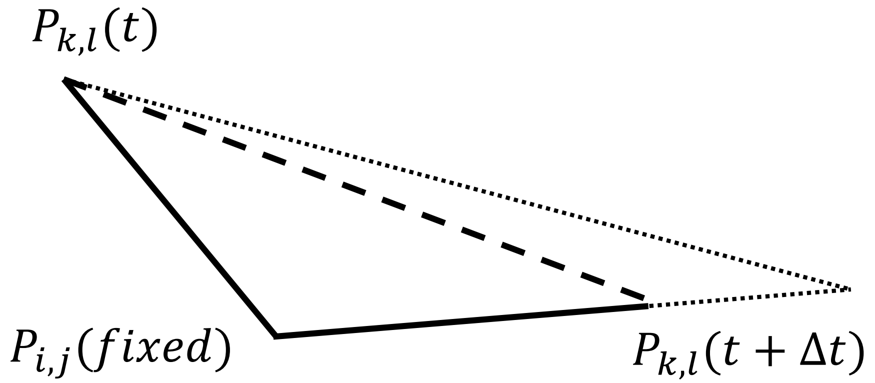




Deformation Constraint(3)

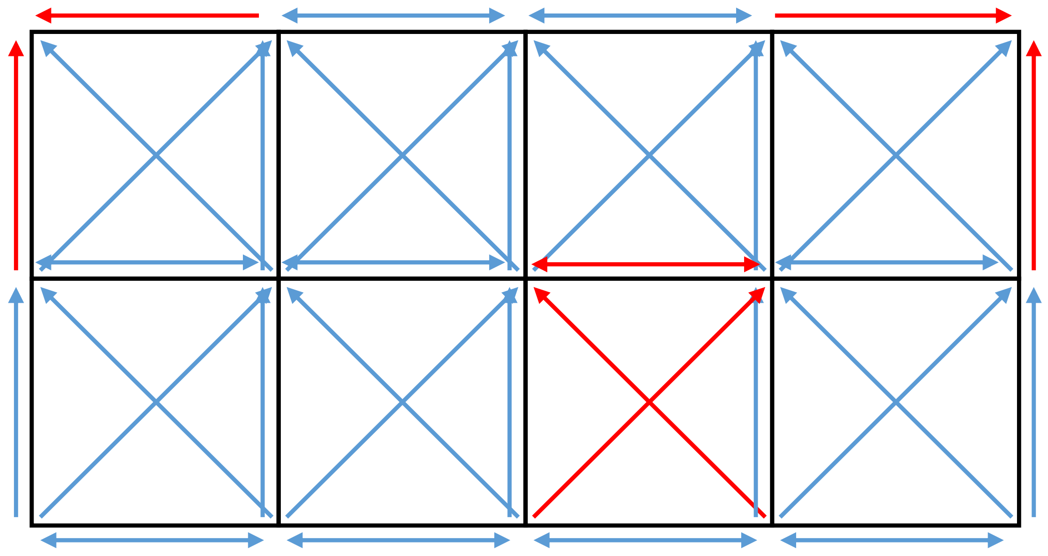
- Dynamic Inverse Procedure (fixed mass)





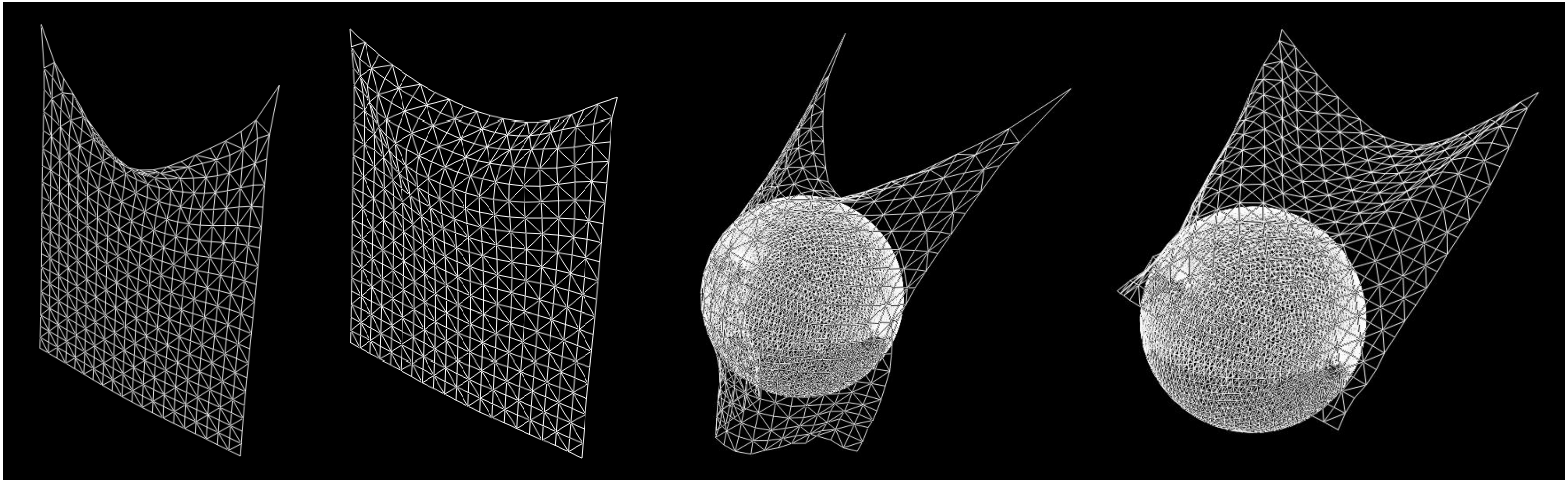
Deformation Constraint(4)

- Dynamic Inverse Procedure (mesh)
- Inverse vector size (bidirection = ½)



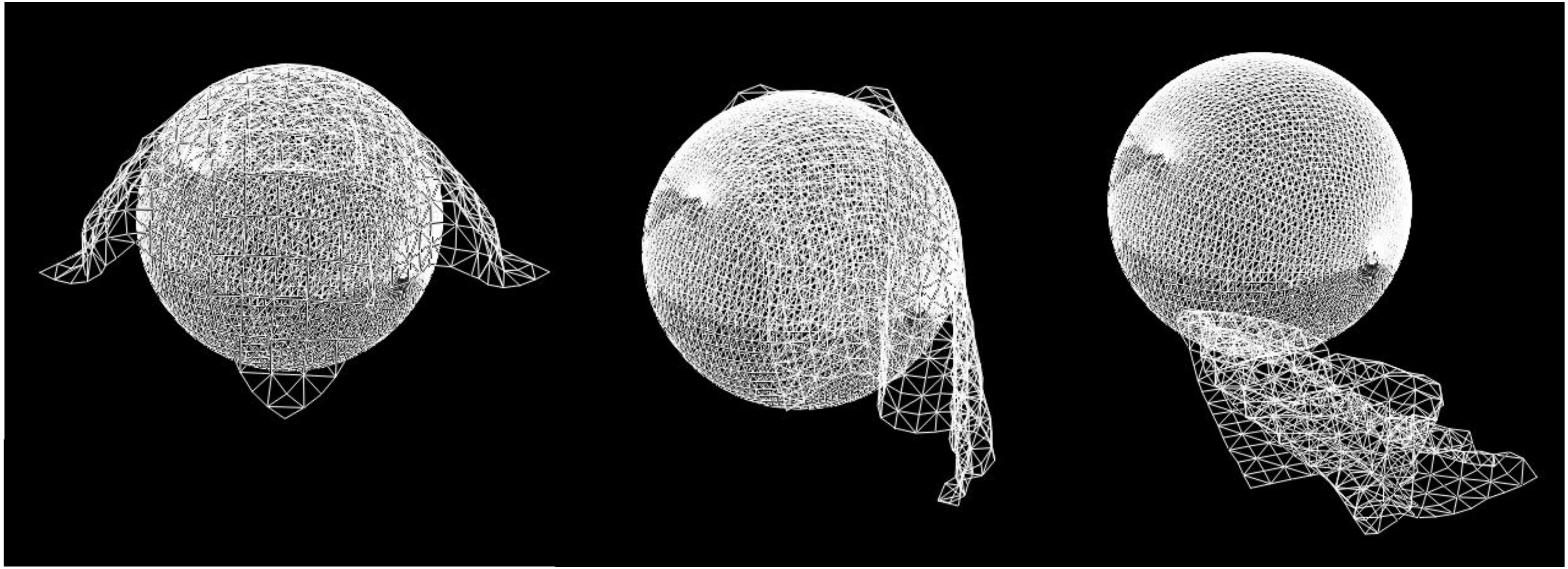
Deformation Constraint(5)

- Results



Frictional Force(1)

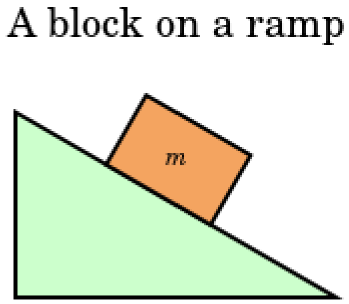
- Additional force from collision! (See below, It’s so sad…)

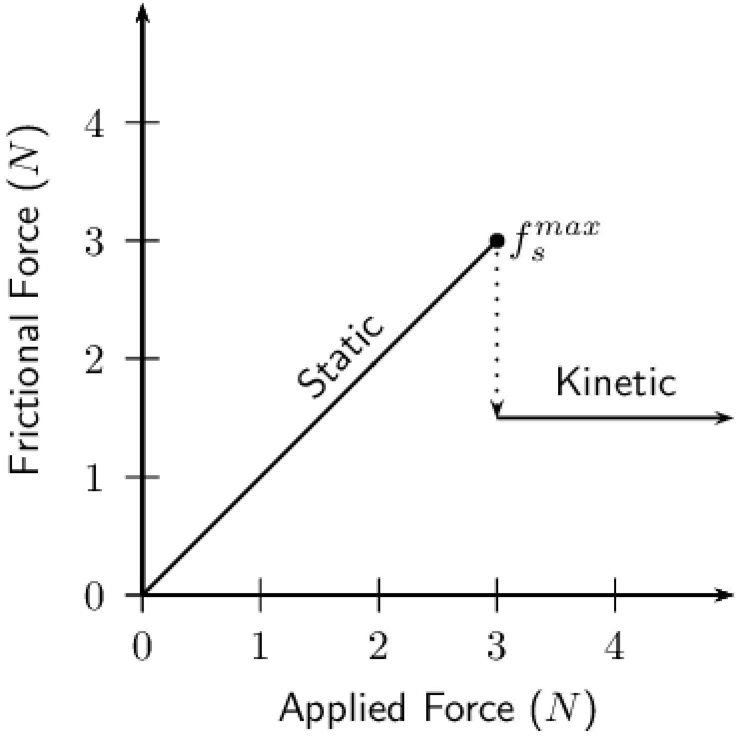


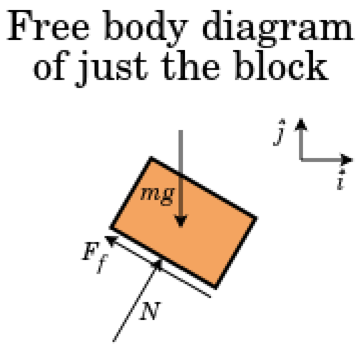
Frictional Force(2)

- Friction offsets all forces when $F_f \leq \mu F_n$
- If $F_f > \mu F_n, F = F_f - \mu F_n$

- Static vs. Kinetic?
  - Significant if F_n is large
  - Clothes are light-weighted
  - Thus, we won't care about it







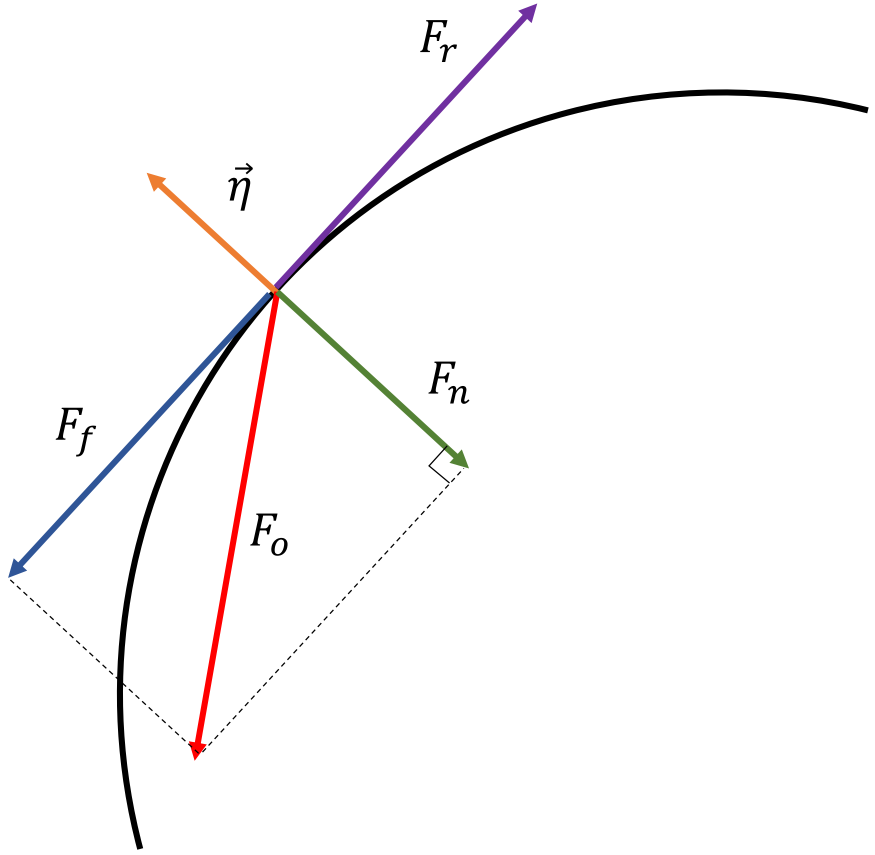
Frictional Force(3)

- Getting forces

$$\vec{\eta} = p_{vertex} - p_{sphere}$$
$$\overrightarrow{v_t} = \frac{(\vec{\eta} \times F) \times \vec{\eta}}{|(\vec{\eta} \times F) \times \vec{\eta}|}$$

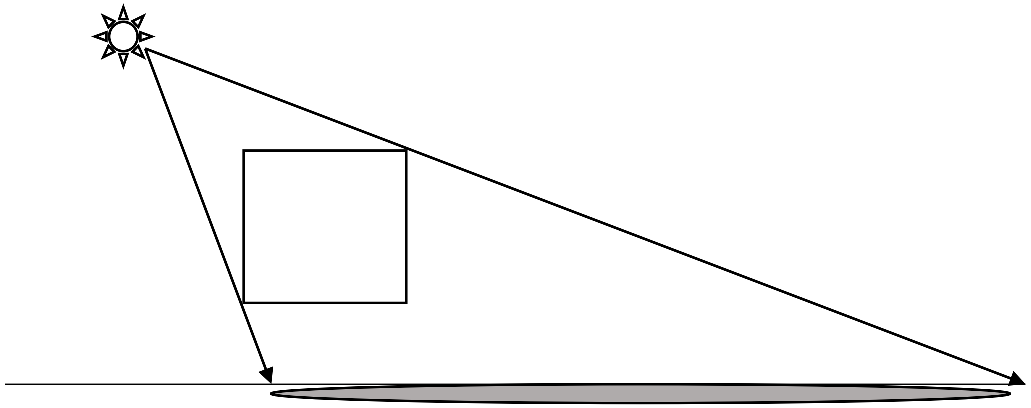
$$|F_n| = \max((-\vec{\eta}) \cdot F_o, 0)$$
$$|F_r| = \mu |F_n|$$
$$|F_f| = \max(\overrightarrow{v_t} \cdot F_o, 0)$$

$$F = \max(|F_f| - |F_r|, 0)(-\overrightarrow{v_t})$$

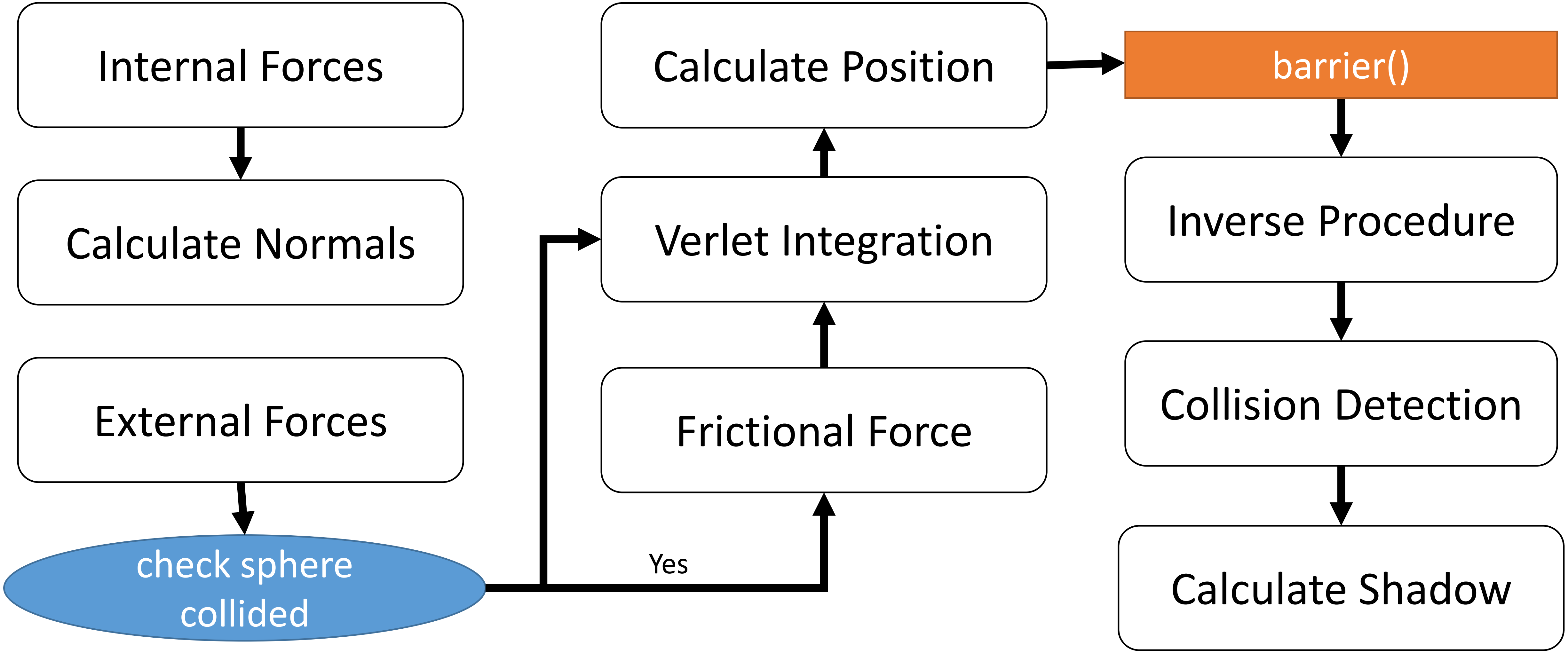


Simple Shadow

- Point Light Projection $$\frac{(x-x_0)}{a}=\frac{(y-y_0)}{b}=\frac{(z-z_0)}{c}\,,\quad (a,b,c\neq 0)$$

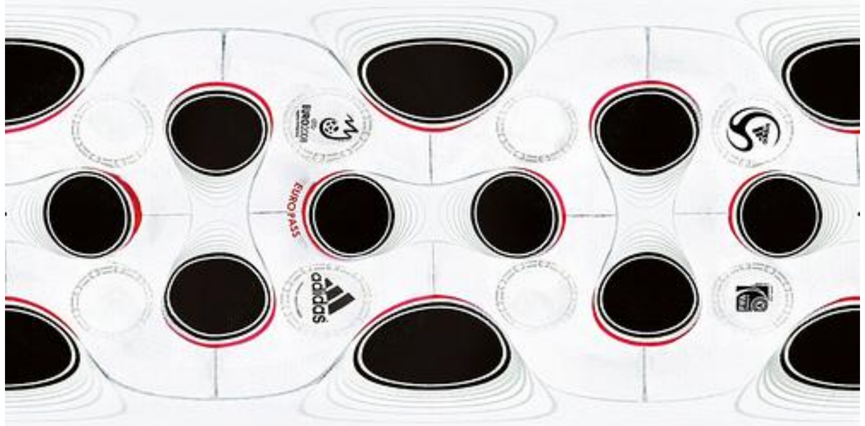


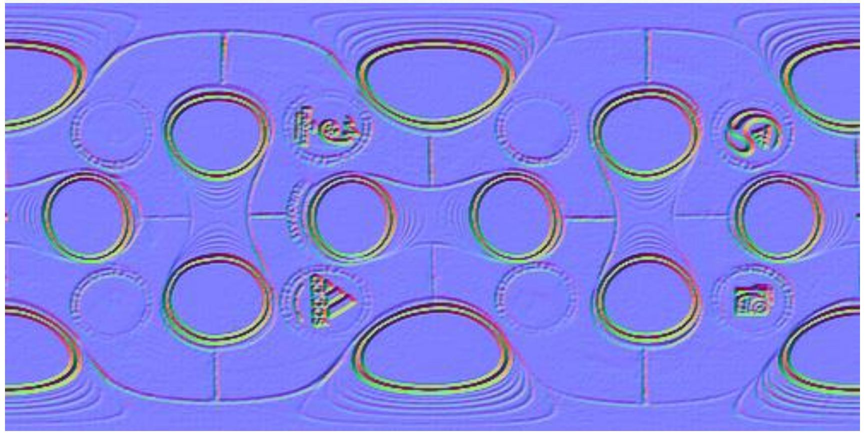
Mass-Spring Shader Program Summary



Sphere, Floor

- Texture/Normal mapping



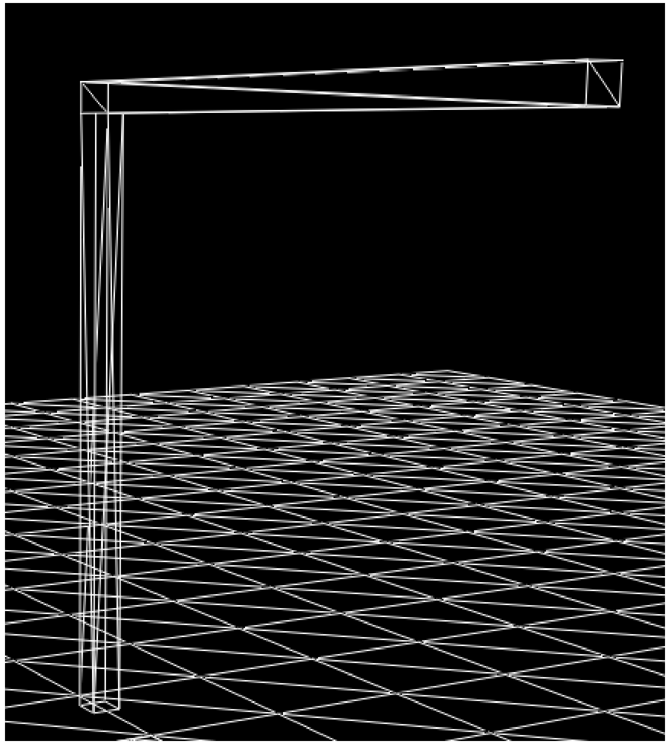


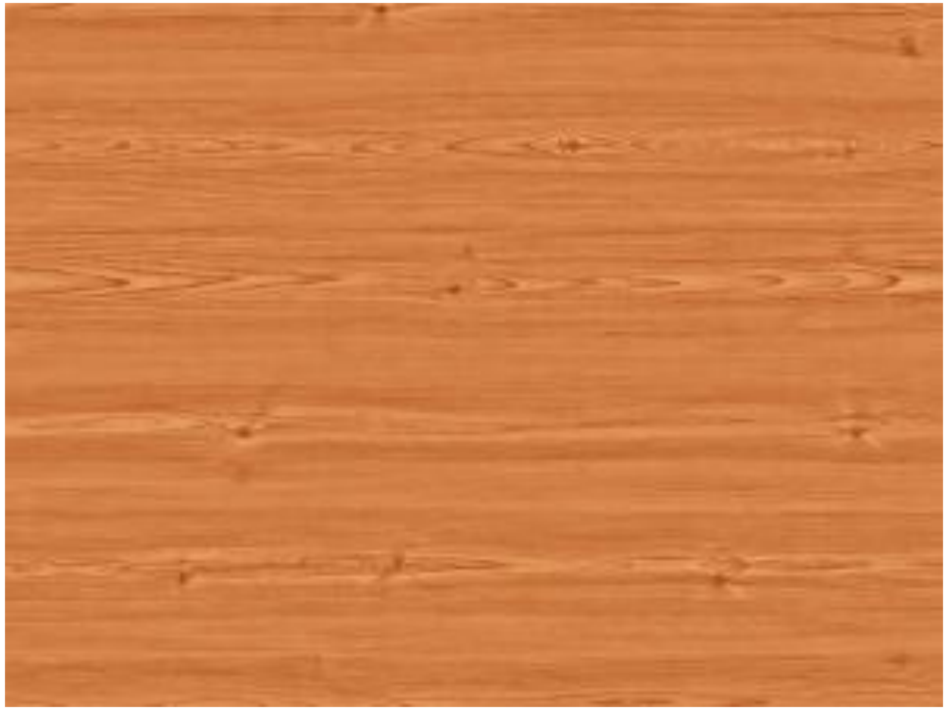
Skybox Mapping

- Skybox cube map

Wooden Bar

- Simple self-made object





Demo

- Shadow, Skybox, …
- Collision Detection (Floor/Sphere)
- Wind force with collision
- Deformation Constraint (Dynamic Inverse Procedure)
- Friction
- Kicking the ball through the cloth

Reference

- http://matthias-mueller-fischer.ch/realtimephysics/coursenotes.pdf
- http://cg.alexandra.dk/?p=147
- http://gamedevelopment.tutsplus.com/tutorials/simulate-tearable-cloth-and-ragdolls-with-simple-verlet-integration--gamedev-519
- https://www.opengl.org/wiki/Shader_Storage_Buffer_Object
- https://www.opengl.org/wiki/Compute_Shader
- Provot, X. (1995, May). Deformation constraints in a mass-spring model to describe rigid cloth behaviour. In *Graphics interface* (pp. 147-147). Canadian Information Processing Society.

Q&A

Thank you for listening!