Physically-Based Simulation Final Project: Paper Simulation

Group 8

Sofie Kamber, Jasmin Koller, Jonathan Manz

Overview

Discrete Shells

Eitan Grinspun, Caltech

Anil N. Hirani, Caltech

Mathieu Desbrun, USC

Peter Schröder, Caltech

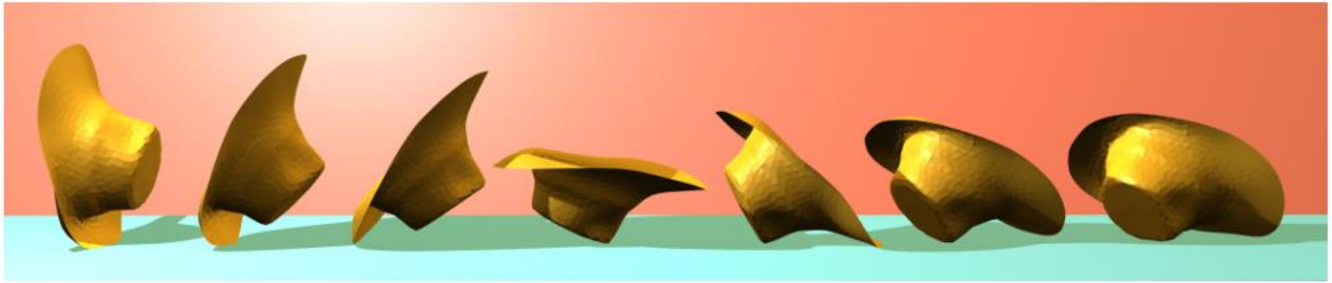


Figure 1: *Composite of 7 frames from a simulation with our thin-shell simulator as a hat is hitting the floor and tumbling to the right.*

Discrete Shells

Eitan Grinspun, Caltech
Anil N. Hirani, Caltech
Mathieu Desbrun, USC
Peter Schröder, Caltech

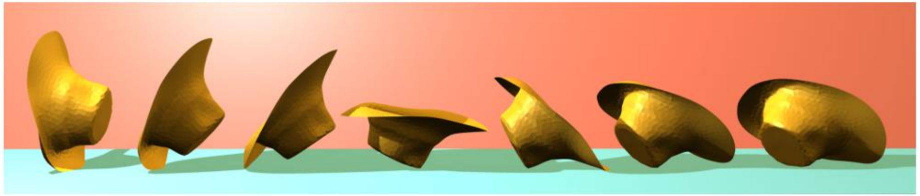


Figure 1: *Composite of 7 frames from a simulation with our thin-shell simulator as a hat is hitting the floor and tumbling to the right.*

Discrete Shells – Total Energy

- Stretching: Change of edge lengths

$$E_L = k_{\text{stretch}} \cdot \sum_{e_{ij} \in E} |\bar{e}_{ij}|^2 \left(\frac{|e_{ij}|}{|\bar{e}_{ij}|} - 1 \right)^2$$

- Stretching: Change of triangle areas

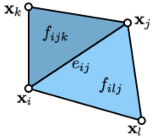
$$E_A = k_{\text{area}} \cdot \sum_{f_{ijk} \in F} |\bar{f}_{ijk}| \left(\frac{|f_{ijk}|}{|\bar{f}_{ijk}|} - 1 \right)^2$$

- Bending: Change of dihedral angles

$$E_B = k_{bend} \sum_{ij \in E} (\theta_{ij} - \bar{\theta}_{ij})^2 \frac{|\bar{e}_{ij}|}{\bar{h}_{ij}}$$

- Total energy

$$E_{DS} = E_L + E_A + E_B$$



Automatic Differentiation

Newmark Time Stepping

Progress since Milestone Presentation

- Implemented Membrane and Flexure Energies

Discrete Shells – Total Energy

- Stretching: Change of edge lengths

$$E_L = k_{\text{stretch}} \cdot \sum_{e_{ij} \in E} |\bar{e}_{ij}|^2 \left(\frac{|e_{ij}|}{|\bar{e}_{ij}|} - 1 \right)^2$$

- Stretching: Change of triangle areas

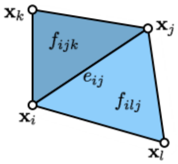
$$E_A = k_{\text{area}} \cdot \sum_{f_{ijk} \in F} |\bar{f}_{ijk}| \left(\frac{|f_{ijk}|}{|\bar{f}_{ijk}|} - 1 \right)^2$$

- Bending: Change of dihedral angles

$$E_B = k_{bend} \sum_{ij \in E} (\theta_{ij} - \bar{\theta}_{ij})^2 \frac{|\bar{e}_{ij}|}{\bar{h}_{ij}}$$

- Total energy

$$E_{DS} = E_L + E_A + E_B$$



Lecture slide 12 (Thin Shells)

```
import taichi as ti
import taichi.math as math
2 usages  Jonathan Manz
@ti.func
def flex_J(x1, y1, z1, x2, y2, z2, x3, y3, z3, x4, y4, z4, e_bar, theta_bar, h_bar):
    x0 = -y3
    x5 = x0 + y2
    x6 = x1 - x2
    x7 = -y4
    x8 = x7 + y1
    x9 = -x4
    x10 = x1 + x9
    x11 = y1 - y2
    x12 = x10*x11 - x6*x8
    x13 = x12*x5
    x14 = x7 + y2
    x15 = x0 + y1
    x16 = -x3
    x17 = x1 + x16
    x18 = x11*x17 - x15*x6
    x19 = x14*x18
    x20 = -z3
    x21 = x20 + z2
    x22 = -z4
    x23 = x22 + z1
    x24 = z1 - z2
    x25 = x10*x24 - x23*x6
    x26 = x21*x25
```

```
@ti.func
def flex_H(x1, y1, z1, x2, y2, z2, x3, y3, z3, x4, y4, z4, e_bar, theta_bar, h_bar):
    x0 = -y3
    x5 = x0 + y2
    x6 = x1 - x2
    x7 = -y4
    x8 = x7 + y1
    x9 = x6*x8
    x10 = -x4
    x11 = x1 + x10
    x12 = y1 - y2
    x13 = x11*x12
    x14 = -x13
    x15 = x14 + x9
    x16 = -x15
    x17 = x16*x5
    x18 = x7 + y2
    x19 = x0 + y1
    x20 = x19*x6
    x21 = -x3
    x22 = x1 + x21
    x23 = -x12*x22
    x24 = x20 + x23
    x25 = -x24
    x26 = x18*x25
    x27 = -z3
```

Progress since Milestone Presentation

- Implemented Newmark Integration Scheme

Newmark Time Stepping We adopt the Newmark scheme[24] for ODE integration,

$$\begin{aligned}
\mathbf{x}_{i+1} &= \mathbf{x}_i + \boxed{\Delta t_i} + \Delta t_i^2\left((1/2-\beta)\ddot{\mathbf{x}}_i + \beta\ddot{\mathbf{x}}_{i+1}\right), \\
\dot{\mathbf{x}}_{i+1} &= \dot{\mathbf{x}}_i + \Delta t_i\left((1-\gamma)\ddot{\mathbf{x}}_i + \gamma\ddot{\mathbf{x}}_{i+1}\right),
\end{aligned}$$

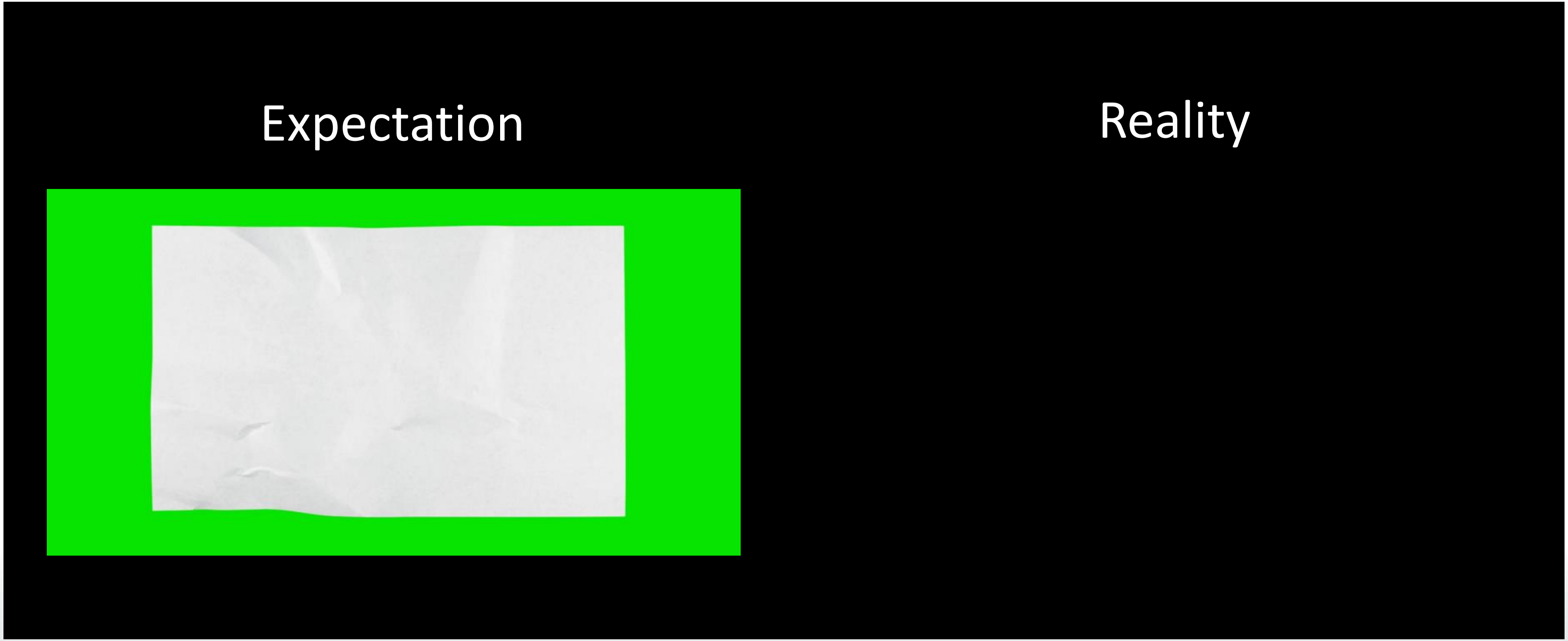
[24] N. M. Newmark. A method of computation for structural dynamics. ASCE J. of the Engineering Mechanics Division, 85(EM 3):67–94, 1959.

computer graphics laboratory

Targets

1. Minimal Target: Membrane and flexure energies
2. Desired Target: A scrunched paper including self collision, basic rendering
3. Bonus Target: Friction effects, viscosity

Demo



https://www.youtube.com/watch?v=bn6mS2_9tE4&ab_channel=10thElementGraphics

Difficulties / Take-aways

1. Don't use Taichi Lang
2. Debugging

Thanks for your attention!