Homework 3
Position-based dynamics

Advanced computer graphics 2024/25

1 Introduction

The goal of this homework is to get familiar with position-based dynamics [MHHR07] for physically based animation. Your task is to implement and visualize a cloth simulation using a grid of particles subject to distance constraints.

1.1 Unconstrained particle dynamics

Particles will be modeled as point masses with position $\mathbf{x}(t)$, velocity $\mathbf{v}(t)$, and mass m. We will use the following system of differential equations (Newton's second law) to model the particles' motion:

$$\dot{\mathbf{v}} = \frac{\mathbf{F}}{m}, \tag{1}$$

$$\dot{\mathbf{x}} = \mathbf{v}, \tag{2}$$

where $\mathbf{F}$ is the sum of all forces acting on a particle. This system can be easily solved through Euler integration:

$$\mathbf{v}(t + \Delta t) = \mathbf{v}(t) + \Delta t\, \frac{\mathbf{F}}{m}, \tag{3}$$

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \Delta t\, \mathbf{v}(t). \tag{4}$$

In pseudocode, this becomes:

while simulating with time step Δt **do**
  for all particles i **do**
    $\mathbf{v}_i \leftarrow \mathbf{v}_i + \Delta t\, \frac{\mathbf{F}}{m}$
    $\mathbf{x}_i \leftarrow \mathbf{x}_i + \Delta t\, \mathbf{v}_i$

1.2 Constrained particle dynamics

Constraints can be added to force the simulation to add joints and attachments, handle contacts, and avoid impossible situations, such as particles falling through a solid surface.

In position-based dynamics, constraints are of the form $C(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n) \in \mathbb{R}$, i.e., it depends only on particle positions. An equality constraint is satisfied when $C = 0$, and an inequality constraint is satisfied when $C \geq 0$.

For example, this is a distance constraint that forces two particles $\mathbf{x}_1$ and $\mathbf{x}_2$ to be at a distance d from each other:

$$C(\mathbf{x}_1, \mathbf{x}_2) = \|\mathbf{x}_1 - \mathbf{x}_2\| - d. \tag{5}$$

To satisfy the given constraints, small corrections $\Delta\mathbf{x}_i$ are added to particle positions.

Finally, new velocities are assigned to the particles by taking the differences in positions over the time step. This is equivalent to taking the average velocities over the last time step, which ensures numerical stability, since the velocities are guaranteed to be small for small $\Delta\mathbf{x}_i$.

The simulation loop is changed appropriately:

while simulating with time step Δt **do**
  for all particles i **do**
    $\mathbf{x}_i^{\text{previous}} \leftarrow \mathbf{x}_i$
    $\mathbf{v}_i \leftarrow \mathbf{v}_i + \Delta t\, \frac{\mathbf{F}}{m}$
    $\mathbf{x}_i \leftarrow \mathbf{x}_i + \Delta t\, \mathbf{v}_i$
  for all constraints C_j **do**

for all particles i involved in C_j **do**
  calculate $\Delta\mathbf{x}_i$
  $\mathbf{x}_i \leftarrow \mathbf{x}_i + \Delta\mathbf{x}_i$
for all particles i **do**
  $\mathbf{v}_i \leftarrow (\mathbf{x}_i - \mathbf{x}_i^{\text{previous}})/\Delta t$

The tricky part is to calculate $\Delta\mathbf{x}_i$.

Let $\mathbf{x}$ be the concatenation of all particle positions, and C the joint constraint function. Then, since the corrections are small, a linear approximation is adequate:

$$C(\mathbf{x} + \Delta\mathbf{x}) \approx C(\mathbf{x}) + \nabla C(\mathbf{x}) \cdot \Delta\mathbf{x} = 0. \tag{6}$$

The corrections $\Delta\mathbf{x}$ should be minimal, and therefore the particles are moved in the direction of the gradient of the constraint:

$$\Delta\mathbf{x} = \lambda \nabla C(\mathbf{x}). \tag{7}$$

Solving for the Lagrange multiplier λ and substituting back gives:

$$\Delta\mathbf{x} = -\frac{C(\mathbf{x})}{\|\nabla C(\mathbf{x})\|^2} \nabla C(\mathbf{x}). \tag{8}$$

To account for different masses, weights $w_i = 1/m_i$ are assigned to particles, and the correction is redistributed proportionally to the weights.

Position-based dynamics calculates these corrections for each particle individually, so you only have to consider the gradient with respect to the given particle.

1.3 Distance constraint

The distance constraint forces two particles at positions $\mathbf{x}_2$ and $\mathbf{x}_2$ to be at distance d from each other:

$$C(\mathbf{x}_1, \mathbf{x}_2) = \|\mathbf{x}_1 - \mathbf{x}_2\| - d. \tag{9}$$

The gradients with respect to $\mathbf{x}_1$ and $\mathbf{x}_2$ are $+\frac{\mathbf{x}_1 - \mathbf{x}_2}{\|\mathbf{x}_1 - \mathbf{x}_2\|}$ and $-\frac{\mathbf{x}_1 - \mathbf{x}_2}{\|\mathbf{x}_1 - \mathbf{x}_2\|}$, respectively. The corrections are therefore:

$$\Delta\mathbf{x}_1 = -(\|\mathbf{x}_1 - \mathbf{x}_2\| - d)\frac{\mathbf{x}_1 - \mathbf{x}_2}{\|\mathbf{x}_1 - \mathbf{x}_2\|}, \tag{10}$$

$$\Delta\mathbf{x}_2 = +(\|\mathbf{x}_1 - \mathbf{x}_2\| - d)\frac{\mathbf{x}_1 - \mathbf{x}_2}{\|\mathbf{x}_1 - \mathbf{x}_2\|}. \tag{11}$$

1.4 Substepping

In the basic formulation, the constraints are iterated over only once and satisfied one by one. The satisfaction of a later constraint may, in fact, destroy the satisfaction of an earlier constraint. Doing more iterations improves the result:

while simulating with time step Δt **do**
  for all particles i **do**
    update $\mathbf{x}_i$ and $\mathbf{v}_i$
  for n iterations **do**
    for all constraints C_j **do**
      solve C_j
  for all particles i **do**
    calculate $\mathbf{v}_i$

However, a better solution is to make smaller time steps to begin with. You can specify the number of substeps n and use $\Delta t_n = \Delta t/n$:

while simulating with time step Δt **do**
  for n substeps with Δt_n **do**
    for all particles i **do**
      update $\mathbf{x}_i$ and $\mathbf{v}_i$

for all constraints C_j **do**
  solve C_j
for all particles i **do**
  calculate $\mathbf{v}_i$

2 Task

Your task is to implement a cloth simulation using position-based dynamics. The cloth will be an $N \times N$ grid (with $(N+1) \times (N+1)$ particles). Adjacent particles will be connected with distance constraints. There will be only one external force, gravity. To prevent the cloth falling into infinity, you must pin two corners of the cloth to fixed positions. This can be added as a distance constraint.

Initially, the cloth will be horizontal. As it starts falling, the two fixed edges will cause the cloth to make a sweeping motion.

Visualize the scene.

Measure the performance and plot the simulation time with respect to cloth resolution and with respect to the number of substeps.

2.1 *Optional: interaction*

Implement interaction so that the user can drag particles around with the mouse.

2.2 *Optional: tearing*

Implement tearing by removing a distance constraint if the distance between particles is to large.

3 Outputs

The expected output of this homework is a real-time simulation and visualization of a physically based cloth, and a short report on performance with respect to the number of simulation substeps and the resolution of the cloth.

4 Grading

This assignment is worth 10 (*+2*) points:

- 2 points for Euler integration,
- 3 points for the constraints,
- 3 points for the position-based dynamics algorithm and substepping,
- 1 point for visualization,
- 1 point for the report,
- *1 point for interaction*
- *1 point for tearing*

References

[MHHR07] Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. Position based dynamics. *Journal of Visual Communication and Image Representation*, 18(2):109–118, 2007.