Cloth Simulation

COMP 768 Presentation
Zhen Wei
Outline

• Motivation and Application
• Cloth Simulation Methods
  • Physically-based Cloth Simulation
    • Overview
    • Development
• References
Motivation

- Movies
- Games
- VR scene
- Virtual Try-on
- Fashion Design
Cloth Simulation Methods

• Geometric Method
  • Represent folds and creases by geometrical equations.
  • Aim at modeling the appearance of the cloth
  • Not focus on the physical aspects of cloth

• Physically-based Method
Physically-based Cloth Simulation

General Ideas

- Represent cloth as grids
- Vertices are points with finite mass
- Forces and energies of points are calculated from the relations with the other points
Physically-based Cloth Simulation

How does Cloth Simulation work
Differences among the methods

- Finding Governing Equation
- Solving the Equations
- Collision Detection / Handling
Governing Equation

\[ \ddot{x} = M^{-1} \left( -\frac{\partial E}{\partial x} + F \right) \]

- \( x \): vector, the geometric state
- \( M \): diagonal matrix, mass distribution of the cloth
- \( E \): a scalar function of \( x \), cloth’s internal energy
- \( F \): a function of \( x \) and \( x' \), other forces acting on cloth
Governing Equation

- Potential energy $E$ is related to deformations
  - Stretch
  - Shear
  - Bending
  - Other Forces

\[ \ddot{x} = M^{-1} \left( -\frac{\partial E}{\partial x} + F \right) \]
Solving the Equations

- Explicit Euler
- Implicit Euler
- Midpoint (Leapfrog)
- Runge-Kutta
- Crank-Nicolson
- Adams-Bashforth, Adams-Moulton
- Backward Differentiation Formula (BDF)

\[ \dot{x} = M^{-1} \left( -\frac{\partial E}{\partial x} + F \right) \]
Solving the Equations

• Crank-Nicolson

• If the partial differential equation is

\[
\frac{\partial u}{\partial t} = F\left(u, x, t, \frac{\partial u}{\partial x}, \frac{\partial^2 u}{\partial x^2}\right)
\]

• Solution:

\[
\frac{u_{i}^{n+1} - u_{i}^{n}}{\Delta t} = F_{i}^{n}\left(u, x, t, \frac{\partial u}{\partial x}, \frac{\partial^2 u}{\partial x^2}\right) \quad \text{(forward Euler)}
\]

\[
\frac{u_{i}^{n+1} - u_{i}^{n}}{\Delta t} = F_{i}^{n+1}\left(u, x, t, \frac{\partial u}{\partial x}, \frac{\partial^2 u}{\partial x^2}\right) \quad \text{(backward Euler)}
\]

\[
\frac{u_{i}^{n+1} - u_{i}^{n}}{\Delta t} = \frac{1}{2} \left[ F_{i}^{n+1}\left(u, x, t, \frac{\partial u}{\partial x}, \frac{\partial^2 u}{\partial x^2}\right) + F_{i}^{n}\left(u, x, t, \frac{\partial u}{\partial x}, \frac{\partial^2 u}{\partial x^2}\right) \right] \quad \text{(Crank--Nicolson)}.
\]

\[u(i\Delta x, n\Delta t) = u_{i}^{n}\]
Solving the Equations

• Linear multistep method
  • Single-step methods (such as Euler's method) refer to only one previous point and its derivative to determine current value. Methods such as Runge–Kutta take some intermediate steps (for example, a half-step) to obtain a higher order method
  • Discard all previous information before taking a second step
  • Multistep methods:
    • Use the information from previous steps: refer to several previous points and derivative values to get current value.
    • Linear multistep methods: linear combination of the previous points and derivative values
Solving the Equations

• Adams’ Method (Linear multistep method)
• One-step Euler

\[ y_{n+1} = y_n + hf(t_n, y_n). \]

\begin{align*}
y_1 &= y_0 + hf(t_0, y_0) = 1 + \frac{1}{2} \cdot 1 = 1.5, \\
y_2 &= y_1 + hf(t_1, y_1) = 1.5 + \frac{1}{2} \cdot 1.5 = 2.25, \\
y_3 &= y_2 + hf(t_2, y_2) = 2.25 + \frac{1}{2} \cdot 2.25 = 3.375, \\
y_4 &= y_3 + hf(t_3, y_3) = 3.375 + \frac{1}{2} \cdot 3.375 = 5.0625.
\end{align*}

• Two-step Adams–Bashforth

\[ y_{n+2} = y_{n+1} + \frac{3}{2} hf(t_{n+1}, y_{n+1}) - \frac{1}{2} hf(t_n, y_n). \]

\begin{align*}
y_2 &= y_1 + \frac{3}{2} hf(t_1, y_1) - \frac{1}{2} hf(t_0, y_0) = 1.5 + \frac{3}{2} \cdot \frac{1}{2} \cdot 1.5 - \frac{1}{2} \cdot \frac{1}{2} \cdot 1 = 2.375, \\
y_3 &= y_2 + \frac{3}{2} hf(t_2, y_2) - \frac{1}{2} hf(t_1, y_1) = 2.375 + \frac{3}{2} \cdot \frac{1}{2} \cdot 2.375 - \frac{1}{2} \cdot \frac{1}{2} \cdot 1.5 = 3.7812, \\
y_4 &= y_3 + \frac{3}{2} hf(t_3, y_3) - \frac{1}{2} hf(t_2, y_2) = 3.7812 + \frac{3}{2} \cdot \frac{1}{2} \cdot 3.7812 - \frac{1}{2} \cdot \frac{1}{2} \cdot 2.375 = 6.0234.
\end{align*}
Solving the Equations

- Adams’ Method (Linear multistep method)
  - Adams–Bashforth methods
    \[
    y_{n+1} = y_n + hf(t_n, y_n), \quad \text{(This is the Euler method)}
    \]
    \[
    y_{n+2} = y_{n+1} + h \left( \frac{3}{2} f(t_{n+1}, y_{n+1}) - \frac{1}{2} f(t_n, y_n) \right),
    \]
    \[
    y_{n+3} = y_{n+2} + h \left( \frac{23}{12} f(t_{n+2}, y_{n+2}) - \frac{4}{3} f(t_{n+1}, y_{n+1}) + \frac{5}{12} f(t_n, y_n) \right),
    \]
    \[
    y_{n+4} = y_{n+3} + h \left( \frac{55}{24} f(t_{n+3}, y_{n+3}) - \frac{59}{24} f(t_{n+2}, y_{n+2}) + \frac{37}{24} f(t_{n+1}, y_{n+1}) - \frac{3}{8} f(t_n, y_n) \right),
    \]
  - Adams–Moulton methods
    \[
    y_{n} = y_{n-1} + hf(t_n, y_n), \quad \text{(This is the backward Euler method)}
    \]
    \[
    y_{n+1} = y_n + \frac{1}{2} h \left( f(t_{n+1}, y_{n+1}) + f(t_n, y_n) \right), \quad \text{(This is the trapezoidal rule)}
    \]
    \[
    y_{n+2} = y_{n+1} + h \left( \frac{5}{12} f(t_{n+2}, y_{n+2}) + \frac{2}{3} f(t_{n+1}, y_{n+1}) - \frac{1}{12} f(t_n, y_n) \right),
    \]
    \[
    y_{n+3} = y_{n+2} + h \left( \frac{3}{8} f(t_{n+3}, y_{n+3}) + \frac{19}{24} f(t_{n+2}, y_{n+2}) - \frac{5}{24} f(t_{n+1}, y_{n+1}) + \frac{1}{24} f(t_n, y_n) \right),
    \]
Solving the Equations

- Backward Differentiation Formula (BDF)

\[
\begin{align*}
\text{BDF1: } y_{n+1} - y_n &= hf(t_{n+1}, y_{n+1}); \text{ (this is the backward Euler method)} \\
\text{BDF2: } y_{n+2} - \frac{4}{3}y_{n+1} + \frac{1}{3}y_n &= \frac{2}{3}hf(t_{n+2}, y_{n+2}); \\
\text{BDF3: } y_{n+3} - \frac{18}{11}y_{n+2} + \frac{9}{11}y_{n+1} - \frac{2}{11}y_n &= \frac{6}{11}hf(t_{n+3}, y_{n+3}) \\
\text{BDF4: } y_{n+4} - \frac{48}{25}y_{n+3} + \frac{36}{25}y_{n+2} - \frac{16}{25}y_{n+1} + \frac{3}{25}y_n &= \frac{12}{25}hf(t_{n+4}, y_{n+4}) \\
\text{BDF5: } y_{n+5} - \frac{300}{137}y_{n+4} + \frac{300}{137}y_{n+3} - \frac{200}{137}y_{n+2} + \frac{75}{137}y_{n+1} - \frac{12}{137}y_n &= \frac{60}{137}hf(t_{n+5}, y_{n+5}) \\
\text{BDF6: } y_{n+6} - \frac{360}{147}y_{n+5} + \frac{450}{147}y_{n+4} - \frac{400}{147}y_{n+3} + \frac{225}{147}y_{n+2} - \frac{72}{147}y_{n+1} + \frac{10}{147}y_n &= \frac{60}{147}hf(t_{n+6}, y_{n+6}).
\end{align*}
\]

Methods with \( s > 6 \) are not zero-stable so they cannot be used.

- vs. Adams–Moulton methods

\[
\begin{align*}
y_n &= y_{n-1} + hf(t_n, y_n), \quad \text{(This is the backward Euler method)} \\
y_{n+1} &= y_n + \frac{1}{2}h \left( f(t_{n+1}, y_{n+1}) + f(t_n, y_n) \right), \quad \text{(This is the trapezoidal rule)} \\
y_{n+2} &= y_{n+1} + h \left( \frac{5}{12}f(t_{n+2}, y_{n+2}) + \frac{2}{3}f(t_{n+1}, y_{n+1}) - \frac{1}{12}f(t_n, y_n) \right), \\
y_{n+3} &= y_{n+2} + h \left( \frac{3}{8}f(t_{n+3}, y_{n+3}) + \frac{19}{24}f(t_{n+2}, y_{n+2}) - \frac{5}{24}f(t_{n+1}, y_{n+1}) + \frac{1}{24}f(t_n, y_n) \right),
\end{align*}
\]
Jerry Weil (1986)

- Geometric model: $y = a \cosh(x/b)$
- A cable under self-weight forms a catenary curve at equilibrium
- A cloth hanging from a discrete number of points can be described by a system of these curves
- Limitation: only models the hanging clothes
Feynman (1986)

- Physically-based model
  - Represented cloth in a 3D space by using a 2D grid
  
  - Energy-based method:
    The energy for each point is calculated in relation to surrounding points

  - The final position of cloth was derived based on the minimization of energy

  \[
  E(P_{i,j}) = k_s E_{\text{elastic}(i,j)} + k_b E_{\text{bending}(i,j)} + k_g E_{\text{gravitational}(i,j)}
  \]

  - Limitation: only modeling cloth draped over rigid objects

- Each particle is based on thread-level interactions

- Energy: Stretching, bending, trellising (shear) & gravity

\[ E_{\text{total}_{ij}} = E_{\text{repel}_{ij}} + E_{\text{stretch}_{ij}} + E_{\text{bend}_{ij}} + E_{\text{trellis}_{ij}} + E_{\text{gravity}_{ij}} \]

- Minimize total energy (SGD), while maintaining collision constraints

- Fit functions to the measured data

- Limitation: No dynamics involved
Provot (1995)

- Spring-Mass System
- Internal Forces and External Forces
- Integration: Simple Euler method
- Dynamic Inverse Procedures
Spring-Mass System for Cloth

- Consider a rectangular cloth with \( m \times n \) particles

**Types of springs**

- **Structural**
  \([i, j] \rightarrow [i, j + 1]; [i, j] \rightarrow [i + 1, j]\)

- **Shear**
  \([i, j] \rightarrow [i + 1, j + 1]; [i + 1, j] \rightarrow [i, j + 1]\)

- **Flexion (bend)**
  \([i, j] \rightarrow [i, j + 2]; [i, j] \rightarrow [i + 2, j]\)

Provot (1995)
Spring-Mass System for Cloth

• Internal Forces: \( F = -k \cdot u \)

\[
F_{\text{int}}(P_{i,j}) = \sum_{(k,l) \in R} K_{i,j,k,l} \left[ l_{i,j,k,l}^0 - l_{i,j,k,l} \right] \frac{l_{i,j,k,l}}{l_{i,j,k,l}^0} \tag{1}
\]

where:

- \( R \) is the set regrouping all couples \((k, l)\) such as \( P_{k,l} \) is linked by a spring to \( P_{i,j} \),
- \( l_{i,j,k,l}^0 = \overrightarrow{P_{i,j}P_{k,l}} \),
- \( l_{i,j,k,l} \) is the natural length of the spring linking \( P_{i,j} \) and \( P_{k,l} \),
- \( K_{i,j,k,l} \) is the stiffness of the spring linking \( P_{i,j} \) and \( P_{k,l} \).

Notations:

- The system is the mesh of \( m \times n \) masses
- \( P_{i,j}(t) \): position at time \( t \)
- \( u \): deformation (displacement from equilibrium) of the elastic body subjected to the force \( F \)

Provot (1995)
Spring-Mass System for Cloth

• External Forces
  • Force of gravity
    \[ F_{gr}(P_{i,j}) = \mu g \]
  • Viscous damping
    \[ F_{dis}(P_{i,j}) = -C_{dis}v_{i,j} \]
  • Viscous fluid (wind)

Notations:
• \( \mu \): mass
• \( g \): the acceleration of gravity
• \( C_{dis} \): damping coefficient
• \( v_{i,j} \): velocity at point \( P_{i,j} \).

Provot (1995)
Spring-Mass System for Cloth

• External Forces

• Viscous fluid (wind)

Force: a viscous fluid moving at a uniform velocity $u_{\text{fluid}}$ exerts, on a surface of a body moving at a velocity $v$

$$F_{vi}(P_{i,j}) = C_{vi} [n_{i,j} \cdot (u_{\text{fluid}} - v_{i,j})] n_{i,j}$$

• $u_{\text{fluid}}$: a viscous fluid with uniform velocity
• $v_{i,j}$: velocity at point $P_{i,j}$
• $n_{i,j}$ is the unit normal at $P_{i,j}$
• $C_{vi}$ is the viscosity constant

Provot (1995)
Spring-Mass System for Cloth

• Integration: Simple Euler Method

• Dynamic Inverse Procedure

  • movement is not entirely caused by analytically computed forces (Contact problems: hanging)

  • compute displacement due to the force => we know displacement of a hanging point (=0), compute actual velocity and actual resulting force

• can be used in object collisions and self-intersection

Provot (1995)
Spring-Mass System for Cloth

• Discretization Problem

• Discretize our cloth more or less finely

• It takes a lot of effort to design discretization-independent schemes.

Provot (1995)
Baraff and Witkin (1998)

- **Triangle-based** representation
- Exploit **sparseness** of Jacobian
- **Implicit** integration
- Result - larger time steps, faster simulations (a few CPU-secs/frame)
- Used in Maya Cloth
Large Steps in Cloth Simulation

\[ \ddot{x} = M^{-1} \left( -\frac{\partial E}{\partial x} + F \right) \]

• Every particle has a changing position \( x_i \)

• Given a vector condition \( C(x) \) which we want to be zero

• Associate an energy function \( E_C(x) \) with \( C \), \( k \) is stiffness constant of our choice

\[ E_C(x) = \frac{k}{2} C(x)^T C(x) \]

• Assuming that \( C \) depends on only a few particle, \( C \) gives rise to a sparse force vector \( f \).

\[ f_i = -\frac{\partial E_C}{\partial x_i} = -k \frac{\partial C(x)}{\partial x_i} C(x) \]
Large Steps in Cloth Simulation

\[ f_i = -\frac{\partial E_c}{\partial x_i} = -k \frac{\partial C(x)}{\partial x_i} C(x) \]

- Derivative matrix \( K = \frac{\partial f}{\partial x} \)
- Nonzero entries of \( K \) are \( K_{ij} \) for all pairs of particles \( i \) and \( j \) that \( C \) depends on

\[ K_{ij} = \frac{\partial f_i}{\partial x_j} = -k \left( \frac{\partial C(x)}{\partial x_i} \frac{\partial C(x)}{\partial x_j}^T + \frac{\partial^2 C(x)}{\partial x_i \partial x_j} C(x) \right) \]

- \( K \) is symmetric. \( \partial^2 E/\partial x_i \partial x_j \)
- Also, \( K \) is sparse

Baraff and Witkin (1998)
Large Steps in Cloth Simulation

Stretch Forces

- Stretch can be measured by
  \[ w_u = \frac{\partial w}{\partial u} \quad \text{and} \quad w_v = \frac{\partial w}{\partial v} \]

- Material is unstretched wherever
  \[ \|w_u\| = 1 \quad \text{and} \quad \|w_v\| = 1 \]

- How to calculate?

- Every cloth particle has
  - Changing position \( x_i \) in world space
  - Fixed plane coordinate \((u_i, v_i)\)

- Suppose we have a single continuous function \( w(u, v) \) that maps from plane coordinates to world space
Large Steps in Cloth Simulation

Stretch Forces

- Stretch can be measured by
  \[ w_u = \partial w / \partial u \text{ and } w_v = \partial w / \partial v \]

- Material is unstretched wherever
  \[ \| w_u \| = 1 \quad \| w_v \| = 1 \]

- The condition for the stretch energy
  \[ C(x) = a \left( \frac{\| w_u(x) \| - b_u}{\| w_v(x) \| - b_v} \right) \]
  \( a \) is the triangle’s area in uv coordinates

\[ \begin{align*} 
\Delta x_1 &= w_u \Delta u_1 + w_v \Delta v_1 \\
\Delta x_2 &= w_u \Delta u_2 + w_v \Delta v_2 
\end{align*} \]

Solution:

\[ (w_u \quad w_v) = (\Delta x_1 \quad \Delta x_2) \left( \begin{array}{cc}
\Delta u_1 & \Delta u_2 \\
\Delta v_1 & \Delta v_2 
\end{array} \right)^{-1} \]

Baraff and Witkin (1998)
Large Steps in Cloth Simulation

Sheer Forces

• Approximation to the shear angle

\[ C(x) = a w_u(x)^T w_v(x) \]

a the triangle’s area in the uv plane.

Baraff and Witkin (1998)
Large Steps in Cloth Simulation

Bend Forces

\[ C(x) = \theta \]

\[ \sin \theta = (\mathbf{n}_1 \times \mathbf{n}_2) \cdot \mathbf{e} \quad \text{and} \quad \cos \theta = \mathbf{n}_1 \cdot \mathbf{n}_2 \]

- \( \mathbf{n}_1 \) and \( \mathbf{n}_2 \): the unit normals of the two triangles
- \( \mathbf{e} \): a unit vector parallel to the common edge

Baraff and Witkin (1998)
Large Steps in Cloth Simulation

Damping Forces

The force $f$ arising from the energy acts only in the direction $\partial C(x)/\partial x$.

So should the damping force.

damping force should depend on the component of the system’s velocity in $\partial C(x)/\partial x$ direction.

So the damping strength should depend on $(\partial C(x)/\partial x)^T \dot{x} = \dot{C}(x)$.

$$E_C(x) = \frac{k}{2} C(x)^T C(x)$$

$$f_i = -\frac{\partial E_C}{\partial x_i} = -k \frac{\partial C(x)}{\partial x_i} C(x)$$

$$d = -k_d \frac{\partial C(x)}{\partial x} \dot{C}(x).$$

Baraff and Witkin (1998)
Large Steps in Cloth Simulation

Constraints

Constraints determined by the user or contact constraints

- Reduced Coordinates
- Penalty Methods
- Lagrange Multipliers

Enforcing constraints by mass modification

Example: zero acceleration along z-axis

\[
\ddot{x}_i = \begin{pmatrix}
\frac{1}{m_i} & 0 & 0 \\
0 & 1/m_i & 0 \\
0 & 0 & 0
\end{pmatrix} \mathbf{f}_i
\]

Baraff and Witkin (1998)
Large Steps in Cloth Simulation

Solving Equations

- Resultant sparse linear system
  - solved using conjugate gradient
- Integration:
  - Backward Euler (implicit method)
- Adaptive time stepping

Baraff and Witkin (1998)
Large Steps in Cloth Simulation

Collision

• Collision detection:
  • cloth-cloth: particle-triangle and edge-edge intersection
  • cloth-solid: cloth particle against the faces of solid object

• Collision Response:
  • cloth-cloth: Insert a strong damped spring force to push the cloth apart
  • cloth-solid: If the relative tangential velocity is low, lock the particle onto the surface; If not allow the particle to slide on the surface.

Baraff and Witkin (1998)
Choi and Ko (2002)

- Cloth property
  - Weak resistance to bending
  - Strong resistance to tension
  - Need large compression forces for out-of-plane motion
- Use *column buckling* as their basic model
- Replace bend and compression forces with a single nonlinear model
- Semi-implicit cloth simulation technique (BDF2)
- Allows a large fixed time step

Stable but Responsible Cloth
Bridson, Marino, Fedkiw (2003)

- Clothing with many folds and wrinkle
- Accurate Model for Bending:
  - possibly nonzero rest angles for modeling wrinkles into the cloth
- Mixed explicit/implicit integration (Crank-Nicolson)
- Collisions: Forecasting collision response technique that promotes the development of detail in contact regions. Post-processing method for treating cloth-character collisions that preserves folds and wrinkles
- Dynamic constraint mechanism that helps to control large scale folding

Figure 5: Wrinkles and folds in this CG cloth from Terminator 3: Rise of the Machines are preserved even when tightly stretched over a level set collision volume.
English and Bridson (2008)

- Effective new discretization for deformable surfaces
- Constrained to not deform at all in-plane but free to bend out-of-plane
  - A triangle is rigid if and only if the distance between any two edge midpoints remains constant
    \[ c_{ij}(x) = \| x_i - x_j \|^2 - d_{ij}^2 = 0 \]
  - Lagrange multiplier constraint forces
    \[ F_c = \left( \frac{\partial C}{\partial x} \right)^T \lambda = J^T \lambda, \]
- Second order accurate multistep constrained mechanics time integration scheme (BDF2)

*Figure 5: A developable surface is dropped on a sphere, with immediate wrinkling and creasing patterns.*
References

• SIG GRAPH Courses
  • SIG GRAPH 2003 Course: Clothing Simulation and Animation
  • SIG GRAPH 2005 Course: Advanced Topics Clothing Simulation and Animation

• Some course notes and slides:
  • http://caig.cs.nctu.edu.tw/course/CA/Lecture/clothSimulation.pdf
  • http://caig.cs.nctu.edu.tw/course/CA/Lecture/clothSimulation2.pdf
  • graphics.ucsd.edu/courses/cse169_w05/CSE169_16.ppt
  • http://www.ics.uci.edu/~shz/courses/cs114/slides/mass_spring.pdf

• Some student presentation slides:
  • www.cs.unc.edu/~lin/COMP768-S09/LEC/cloth.pdf