Large Steps in Cloth Simulation

David Baraff, Andrew Witkin

SIGGRAPH 98, Orlando, July 19–24

Presented by: D. Rybarova

24. january 2011

Overview

- Realistic cloth simulation
- Implicit integration method instead of explicit integration method
- Conjugate gradient method for solving sparse linear systems
- Enforcing constraints
- Large time steps instead of small ones
- Simulation system significantly faster

Introduction

- Physically-based cloth simulation formulated as a time-varying partial differential equation

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

- Faster performance – choosing implicit integration method
- Cloth resists stretching motions, but not shearing and bending
- Computational costs of explicit methods limits realizable resolution of cloth

Introduction

Previous approaches

- Terzopoulos: cloth as rectangular mesh, implicit scheme, not very good damping forces
- Carignan: rectangular mesh, explicit integration scheme
- Volino: triangular mesh, collision detection, no damping forces, midpoint method

Introduction

- Cloth - triangular mesh; eliminates topological restrictions of rectangular meshes
- Deformation energies – quadratic, not quartic functions (Terzopoulos, Carignan)
- Directly imposing and mantaining constraints
- Dynamically varying time steps

Simulation overview

- Triangular mesh of *n* particles, x_i – position of *i*-th particle, x – geometric state of all particles
- The same with force f
- Rest state of cloth: each particle has an unchanging coordinates (u,v) in plane
- 3 internal forces (stretch, shear, bend), 3 damping forces, additional forces

Simulation overview

- Shear force and stretch formulated on a per triangle basis, bend force on a per edge basis
- Stretch force – high coefficient of stiffness
- Combining all forces into force vector f
- Acceleration of *i*-th particle: $\ddot{x}_i = f_i / m_i$
- Matrix of masses M: $\ddot{x} = M^{-1} f(x, \dot{x})$
- Constraints – user defined/automatic, in 1,2 or 3 dimensions

Implicit integration

- Position $x(t_0)$ and velocity $\dot{x}(t_0)$ in time t_0
- Goal: determine new position and velocity in time t_0+h
- Define the system's velocity as $v=\dot{x}$:

$$\frac{d}{dt}\begin{pmatrix} \mathbf{x} \\ \dot{\mathbf{x}} \end{pmatrix} = \frac{d}{dt}\begin{pmatrix} \mathbf{x} \\ \mathbf{v} \end{pmatrix} = \begin{pmatrix} \mathbf{v} \\ \mathbf{M}^{-1}\mathbf{f}(\mathbf{x}, \mathbf{v}) \end{pmatrix}$$

- change in notation $x_0=x(t_0)$, $v_0=v(t_0)$, $\Delta x=x(t_0+h)-x(t_0)$, $\Delta v=v(t_0+h)-v(t_0)$

Implicit integration

- Implicit backward Euler's method:

$$\begin{pmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{v} \end{pmatrix} = h \begin{pmatrix} \mathbf{v}_0 + \Delta \mathbf{v} \\ \mathbf{M}^{-1}\mathbf{f}(\mathbf{x}_0 + \Delta \mathbf{x}, \mathbf{v}_0 + \Delta \mathbf{v}) \end{pmatrix}$$

- Taylor series expansion to f and making first-order approximation

$$\mathbf{f}(\mathbf{x}_0 + \Delta \mathbf{x}, \mathbf{v}_0 + \Delta \mathbf{v}) = \mathbf{f}_0 + \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \Delta \mathbf{x} + \frac{\partial \mathbf{f}}{\partial \mathbf{v}} \Delta \mathbf{v}.$$

Implicit integration

- Derivative $\partial f/\partial x$ is evaluated for the state (x_0,y_0), similarly for $\partial f/\partial v$
- Substituting into equation, substituting $\Delta x = h(v_0 + \Delta v)$, considering identity matrix I:

$$\left(\mathbf{I} - h\mathbf{M}^{-1}\frac{\partial \mathbf{f}}{\partial \mathbf{v}} - h^2\mathbf{M}^{-1}\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)\Delta\mathbf{v} = h\mathbf{M}^{-1}\left(\mathbf{f}_0 + h\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\mathbf{v}_0\right)$$

- This is solved for Δv
- Then $\Delta x = h(v_0 + \Delta v)$ is computed

Forces

- Cloth's material behavior described in terms of a scalar potential energy function E(x)
- Force f arising from this energy: $f = -\partial E/\partial x$
- Expressing energy as single function – impractical
- Internal behavior defined by vector condition C(x)
- Associated energy: $k/2 C(x)^T C(x)$, k-stiffness constant

Stretch force

- Every cloth particle has changing position x_i in space and unchanging coordinates (u,v) in plane
- w(u,v) – mapping function from plane coordinates to world space
- Stretch measured by examining $w_u = \partial w / \partial u$ and $w_v = \partial w / \partial v$ at a point
- $|w_u|$ - stretch/compression in u direction

Stretch force

- Apply stretch/compression measure to a triangle: (vertices=particles i,j,k)

$$(\mathbf{w}_u \quad \mathbf{w}_v) = (\Delta\mathbf{x}_1 \quad \Delta\mathbf{x}_2)\begin{pmatrix}\Delta u_1 & \Delta u_2 \\ \Delta v_1 & \Delta v_2\end{pmatrix}^{-1}.$$

Δx1=xj-xi, Δx2=xk-xi, Δu1=uj-ui, Δu2=uk-ui, similarly for Δy1, Δy2

- Condition for a stretch energy:

$$\mathbf{C}(\mathbf{x}) = a\begin{pmatrix} \|\mathbf{w}_u(\mathbf{x})\| - b_u \\ \|\mathbf{w}_v(\mathbf{x})\| - b_v \end{pmatrix}$$

where we treat wu and wv as functions of x; they depend only on xi,xj,xk

Shear and bend forces

- Extent to which cloth has sheared is $w_u^T w_v$
- Condition for shearing: $C(x)=aw_u(x)^T w_v(x)$
- Bend – measured between pair of adjacent triangles
- Bend energy depends upon 4 particles defining two adjoining triangles
- $C(x) = \theta$

n_1 and n_2: unit normals of the two triangles, e: unit vector parallel to the common edge, angle θ between two faces defined by: $\sin \theta = (n_1 x n_2).e$ and $\cos \theta = n_1.n_2$

Damping

- Forces before – functions of position only
- Damping forces – functions of position and velocity
- I.e – strong stretch force must be accompanied by strong damping force (anomalous in-plane oscillations)
- Not formulated for E(x) by measuring velocity of the energy – nonsensical results
- Defined in terms of the condition C(x)

Damping

- Damping force d associated with a condition C:

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

- Add damping forces to internal forces, finding term that breaks symmetry, term omission
- Result:

$$\frac{\partial \mathbf{d}_i}{\partial \mathbf{v}_j} = -k_d \frac{\partial \mathbf{C}(\mathbf{x})}{\partial \mathbf{x}_i} \frac{\partial \dot{\mathbf{C}}(\mathbf{x})}{\partial \mathbf{v}_j}^T = -k_d \frac{\partial \mathbf{C}(\mathbf{x})}{\partial \mathbf{x}_i} \frac{\partial \mathbf{C}(\mathbf{x})}{\partial \mathbf{x}_j}^T$$

Constraints

- Automatically determined by user, or contact constraints generated by system
- At given step, particle is unconstrained/constrained in 1,2 or 3 dimensions
- 3 dimensions: explicitly setting velocity of particle
- 2 or 1 dimension: constraining velocity along either 2 or 1 mutually orthogonal axes

Constraints

Other enforcement mechanisms:

- Reduced coordinates – that are describing position and velocity, complicates system (size of matrices changes)
- Penalty method – stiff springs for preventing illegal motion; additional stiffness needed
- Lagrange multipliers – additional constraint forces; more variables

Constraints

- Build constraints directly into equation
- Inverse mass: M^{-1}, enforcing constraints by mass altering
- W = modified M , $W_{ii} = (1/M_i)*S_i$

$$\mathbf{S}_i = \begin{cases} \mathbf{I} & \text{if ndof}(i) = 3 \\ (\mathbf{I} - \mathbf{p}_i\mathbf{p}_i^T) & \text{if ndof}(i) = 2 \\ (\mathbf{I} - \mathbf{p}_i\mathbf{p}_i^T - \mathbf{q}_i\mathbf{q}_i^T) & \text{if ndof}(i) = 1 \\ \mathbf{0} & \text{if ndof}(i) = 0. \end{cases}$$

ndof(i) is number of degrees of freedom particle, pi and qi – prohibited directions; pi if ndof(i) = 2, qi if ndof(i)=1

Constraints

- For particle i, z_i = change in velocity we wish to enforce in the particle's constrained direction(s)
- Rewriting equation to directly enforce contraints

$$\left(\mathbf{I} - h\mathbf{W}\frac{\partial \mathbf{f}}{\partial \mathbf{v}} - h^2\mathbf{W}\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)\Delta\mathbf{v} = h\mathbf{W}\left(\mathbf{f}_0 + h\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\mathbf{v}_0\right) + \mathbf{z}$$

- Solving for Δv, completely constrained particle: $\Delta v_i = z_i$, partially: Δv_i whose component in the constrained direction(s) = z_i

Implementation

- For small test systems – former equation (with constraints) solved directly
- For larger – iterative method (conjugate gradient)
- Problem – CG method requires symmetrical matrices
- Transforming equation –without constraints – to symmetric system:

$$\left(\mathbf{M} - h\frac{\partial \mathbf{f}}{\partial \mathbf{v}} - h^2\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)\Delta\mathbf{v} = h\left(\mathbf{f}_0 + h\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\mathbf{v}_0\right)$$

Implementation

- Modify CG method so it can operate on equation from former slide
- Procedurally applying the constraints inherent in the matrix W
- Matrix A, vector b, residual vector r:

$$\mathbf{A} = \left(\mathbf{M} - h\frac{\partial \mathbf{f}}{\partial \mathbf{v}} - h^2\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)$$

$$\mathbf{b} = h\left(\mathbf{f}_0 + h\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\mathbf{v}_0\right) \quad \text{and} \quad \mathbf{r} = \mathbf{A}\Delta\mathbf{v} - \mathbf{b}.$$

- Component of r_i in the particle's unconstrained direction(s) will be = 0
- Component of Δv_i in the particle's constrained direction(s) will be = z_i

Modified CG method

- Takes matrix A, vector b, preconditioning matrix P and iteratively solves $A\Delta v=b$
- Termination criteria: $|b-A\Delta v| < e.|b|$
- P speeds convergence (P^{-1} approximates A)
- Effect of matrix W – filter out velocity changes in constrained direction
- Define an invariant - component of Δv_i in constrained direction(s) of particle i is equal to z_i

Modified CG method

- Filter – take vector a and perform filtering operation as multiplying by W
- Method always converges -> it works
- Tried to use not modified CG method with penalty term
- No substantial changes in number of iterations
- Similar convergence behavior

Constraint forces

- Contact constraint (cloth – solid object)
- Need to know actual force, in order to determine when to terminate a contraint
- Frictional forces
- Computed at the end of modified CG: (AΔv-b)
- Releasing constraint: constraint force between a particle and a solid switches from repulsive force to attractive one

Constraint forces

Friction:

- Cloth-solid object contact: particle locked onto a surface
- Monitor constraint force
- If tangential force exceed some fraction of normal force – sliding on the surface allowed

Collisions

- Cloth-cloth: detected by checking pairs (p,t) and (e1,e2) for intersections
- Coherency based bounding box approach
- Collision detected -> insert a strong damped spring force to push them apart
- Friction forces for cloth contact – not solved

Collisions

- Cloth-solid object: testing each cloth particle with faces of object
- Faces of solid object grouped into hierarchical bounding box tree
- Leaves of tree are individual faces of object
- Creation of tree - recursive splitting along coordinate axes

Collision and constraints

- Cloth-solid object collision: enforcing constraint
- Cloth-cloth collision: adding penalty force (enforcing constraints expensive)
- Discrete steps of simulator -> collision between one step and next step
- Cloth-solid object: particle can remain embedded below surface of solid object

Collision and constraints

- Solution – altering the position of cloth particles
- Because using one-step backward Euler method – no problem
- Simple position change – disastrous results
- Large deformation energies in altered particle's neighborhood

Position alteration

- Consider particle collided with solid object
- Particle's position in next step: $\Delta x_i = h(v_{0i} + \Delta v_i)$
- Changing position after this step - particle's neighbors receive no advance notification of the change in position
- $\Delta x_i = h(v_{0i} + \Delta v_i) + y_i$
- y_i – arbitrary correction term

Position alteration

- y_i - move a particle to a desired location during the backward Euler step
- Modify symmetric system:

$$\left(\mathbf{M} - h\frac{\partial \mathbf{f}}{\partial \mathbf{v}} - h^2\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)\Delta \mathbf{v} = h\left(\mathbf{f}_0 + h\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\mathbf{v}_0 + \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\mathbf{y}\right).$$

- Control over position and velocity of a constrained particle in one step
- Cloth-cloth collision: correction term can also be added

Adaptive time stepping

- Take sizeable steps forward, without loss of stability
- Still times when step reduction needed (to avoid divergence)
- Other methods – focused on simulation accurrancy, not stability
- Stiffness – potential instability arises from strong stretch forces

Adaptive time stepping

- Each step – take Δx as proposed change in cloth’s state
- Examine stretch term in every triangle in newly proposed state
- Drastic change in stretch -> discard proposed state, reduce time step, try again

Adaptive time stepping

- Parameter that indicates maximum allowable step size (less or equal to 1 frame)
- Simulator reduces time steps -> 2 successes -> try to increase time step
- Failure at larger step size ->waits for a longer time period -> retrying to increase time step

Results

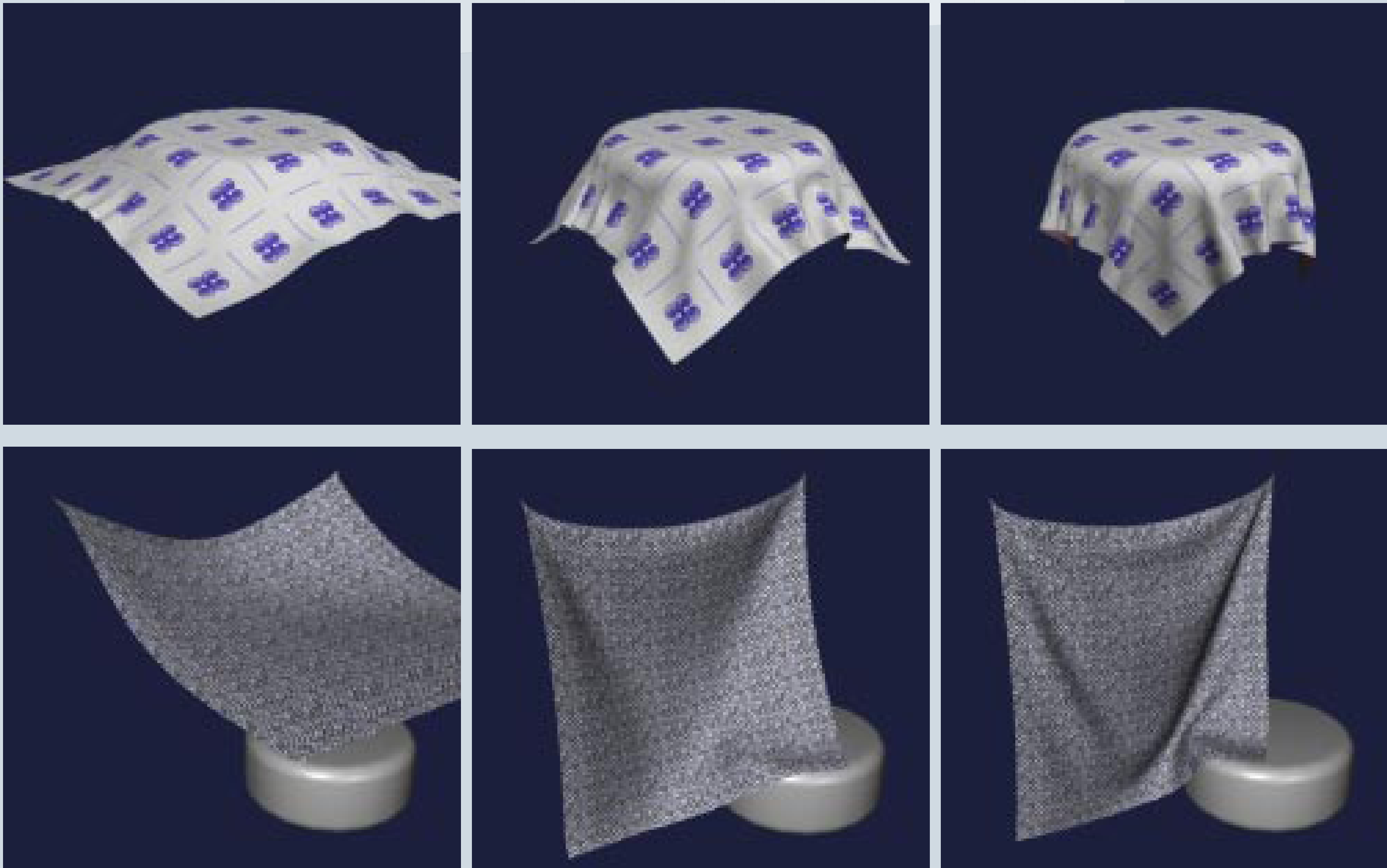
- Estimate their simulator's performance as function of n (number of cloth particles)
- Cloth resolution(Fig.1): 500, 900, 2602, 7359 particles
- Running times: 0.23, 0.46, 2.23, 10.3 seconds/frame
- Slightly better than $O(n^{1.5})$ performance (standard CG method)

Results

figure		no. vertices/no. triangles cloth	solid	time/frame (CPU sec.)	step size min/max (ms)	total frames/ total steps	task breakdown percentage EVAL	CG	C/C	C/S
1		2,602/4,9442	322/640	2.23	16.5/33	75/80	25.7	50.4	18.3	1.4
2		2,602/4,9442	322/640	3.06	16.5/33	75/80	17.9	63.6	15.3	0.2
3		6,450/12,654	9,941/18,110	7.32	16.5/33	50/52	18.9	37.9	30.9	2.6
4	(shirt)	6,450/12,654	9,941/18,110	14.5	2.5/20	430/748	16.7	29.9	46.1	2.2
	(pants)	8,757/17,352	9,941/18,110	38.5	0.625/20	430/1214	16.4	35.7	42.5	1.7
5	(skirt)	2,153/4,020	7,630/14,008	3.68	5/20	393/715	18.1	30.0	44.5	1.5
	(blouse)	5,108/10,016	7,630/14,008	16.7	5/20	393/701	11.2	26.0	57.7	1.3
6	(skirt)	4,530/8,844	7,630/14,008	10.2	10/20	393/670	20.1	36.8	29.7	2.6
	(blouse)	5,188/10,194	7,630/14,008	16.6	1.25/20	393/753	13.2	30.9	50.2	1.4

System performance for simulations in figures 1–6. Minimum and maximum time steps are in milliseconds of simulation time. Time/frame indicates actual CPU time for each frame, averaged over the simulation. Percentages of total running time are given for four tasks: EVAL—forming the linear system of equation (18); CG solving equation (18); C/C—cloth/cloth collision detection; and C/S—cloth/solid collision detection

Results



Results




Results