# Cloth and Hair Collisions

Fig. 27. Real-time rendering of long, moving hair using recent graphics hardware [83]. Image Courtesy of NVIDIA Corporation, 2004

Adrien Treuille

# Course Evaluations

- More emphasis on current challenges.
- More papers / reading.
- More emphasis on realtime.
- Too fast.
- More emphasis on implementation.
- More videos + real-world examples.

# Question

- What kind of collisions are there in cloth?
- How can we detect such collisions?
- How do we prevent cloth from self-intersecting?

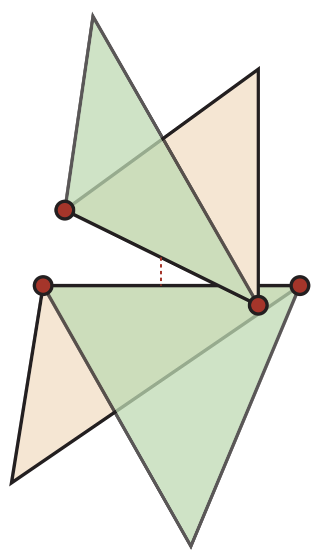
# Outline

- Cloth Collisions
  - Collision Detection
  - Collision Handling
- Hair Collisions
  - Like Cloth
  - Continuum Approaches

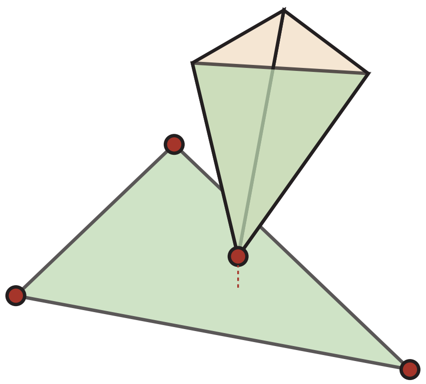
# Collision Detection

## Collision detection

- We already covered this for deformable bodies
- Many of the same methods work, especially acceleration methods
- Generally need to do triangle-triangle collision checks:



Edge-edge collision

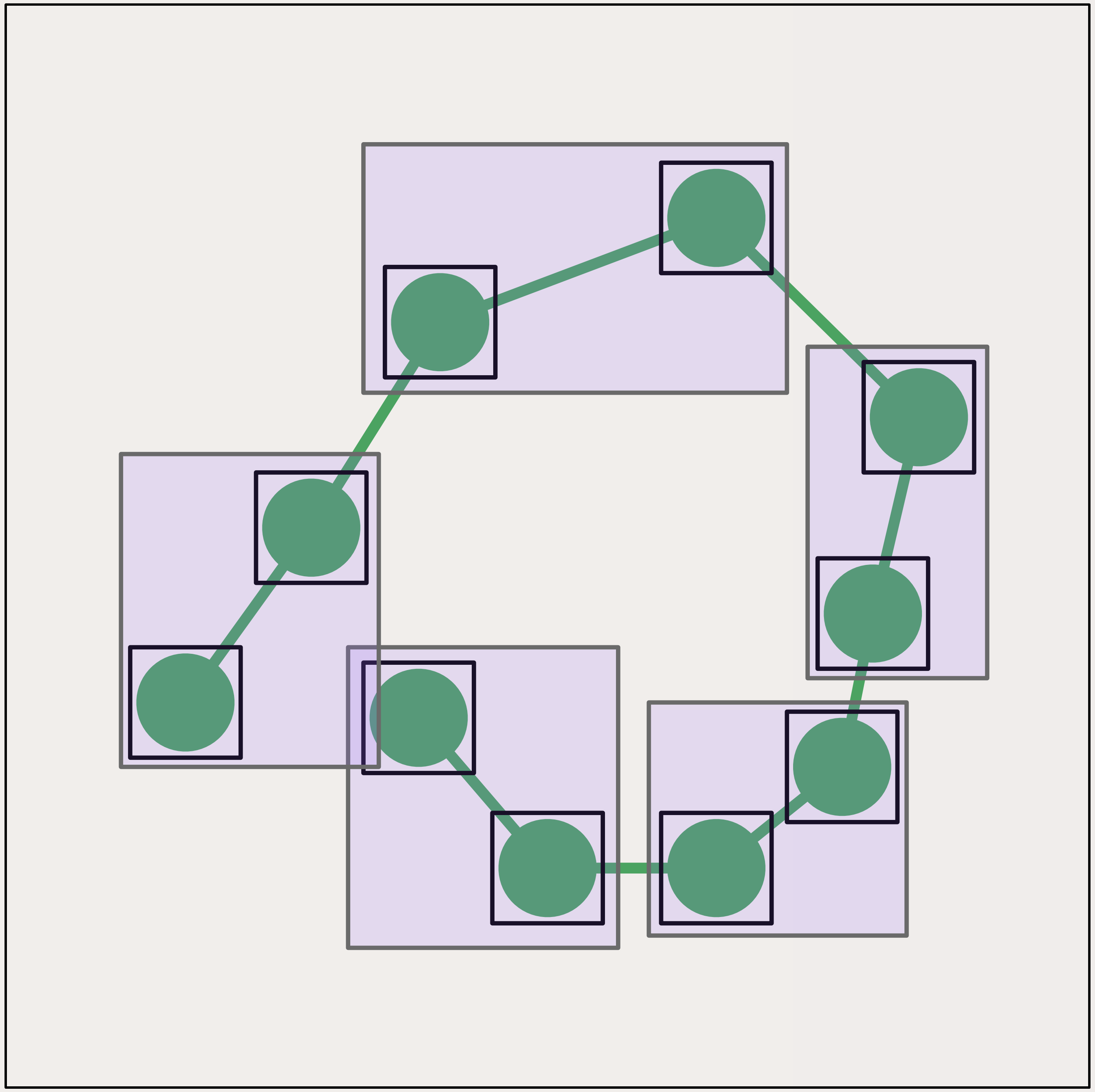


Point-face collision

Cloth Animation

Christopher Twigg
March 4, 2003

# Cloth Collision Detection



# Curvature Optimization

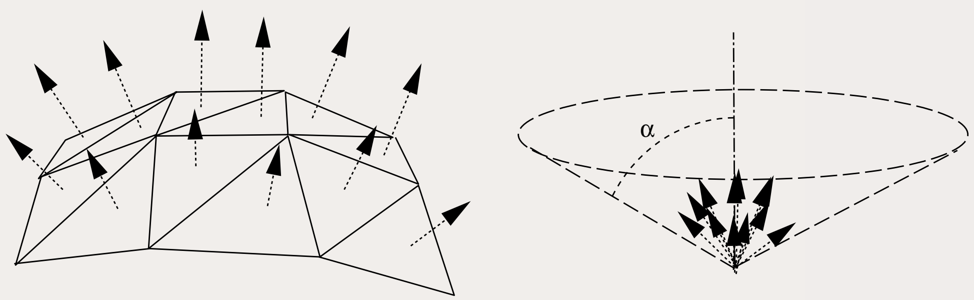



Figure 1: Cone including normals to triangles of a zone of the cloth surface.

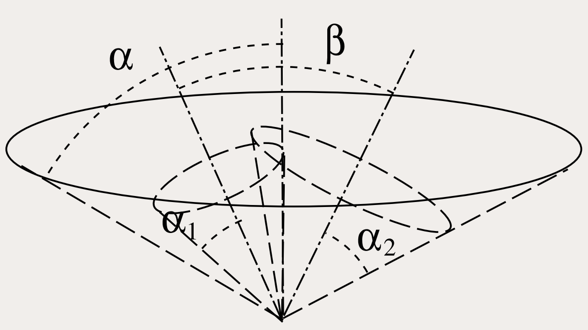


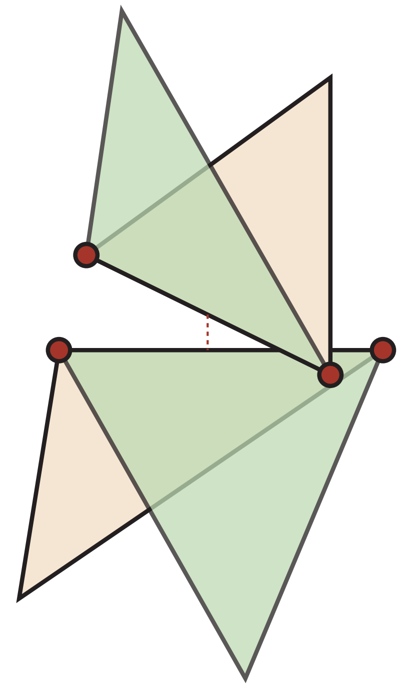

Figure 2: Cone (angle $\alpha$) enclosing its two "descendant" cones in the hierarchical tree (angles $\alpha_1$ and $\alpha_2$).

**Collision and self-collision handling in cloth model dedicated to design garments**

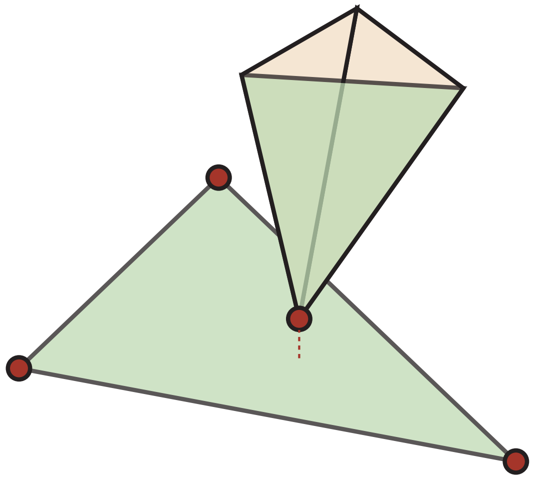
Xavier PROVOT
Institut National de Recherche en Informatique et Automatique (INRIA)
B.P. 105, 78153 Le Chesnay Cedex, France
Xavier.Provot@inria.fr

# Collision detection

- We already covered this for deformable bodies
- Many of the same methods work, especially acceleration methods
- Generally need to do triangle-triangle collision checks:

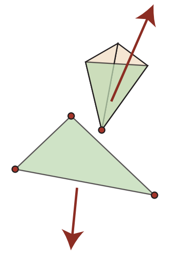


Edge-edge collision



Point-face collision

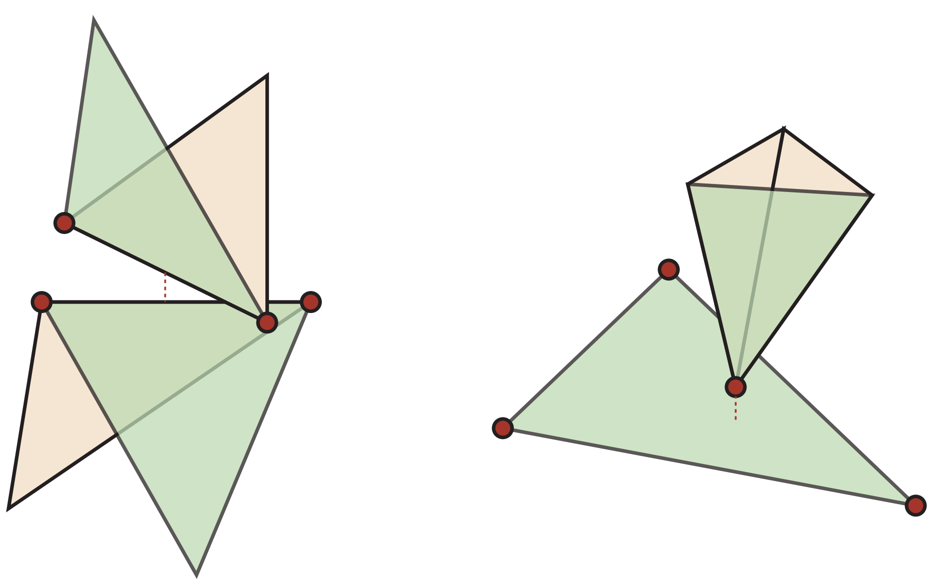
# Robust collision detection



If triangles are moving too fast, they may pass through each other in a single timestep.

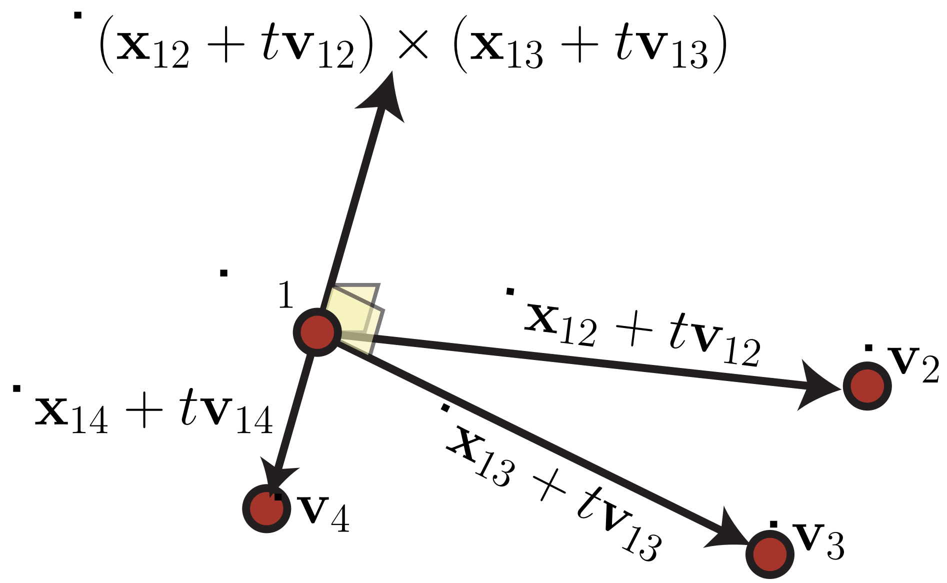
We can prevent this by checking for *any* collisions during the timestep (Provot [1997])

Note first that both point-face and edge-edge collisions occur when the appropriate 4 points are *coplanar*



# Robust collision detection (2)

Detecting time of coplanarity - assume linear velocity throughout timestep:

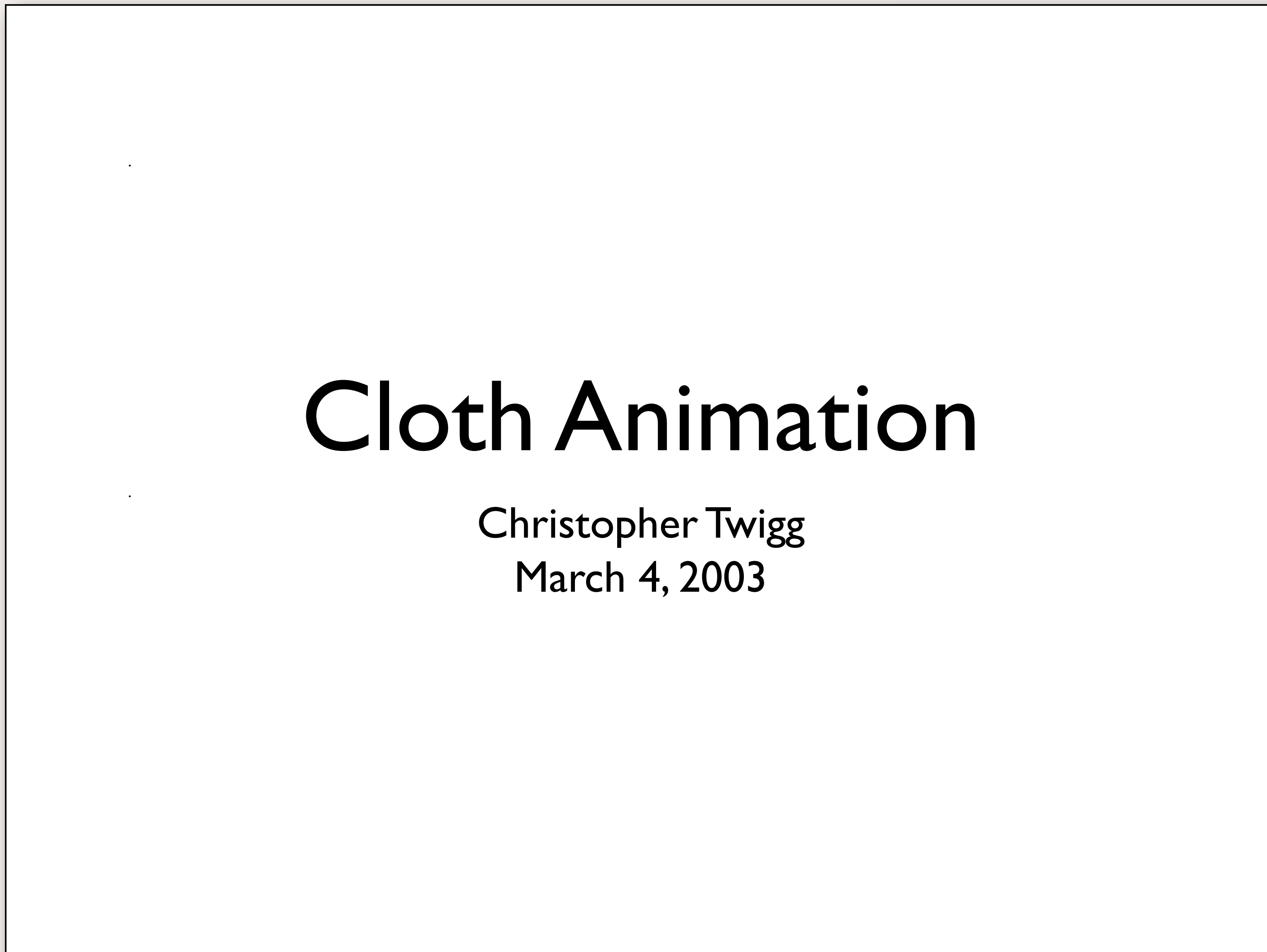



So the problem reduces to finding roots of the cubic equation

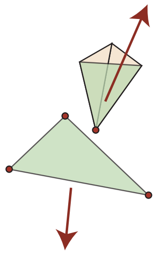
$$\left((\mathbf{x}_{12} + t\mathbf{v}_{12}) \times (\mathbf{x}_{13} + t\mathbf{v}_{13})\right) \cdot (\mathbf{x}_{14} + t\mathbf{v}_{14})$$

Once we have these roots, we can plug back in and test for triangle adjacency.

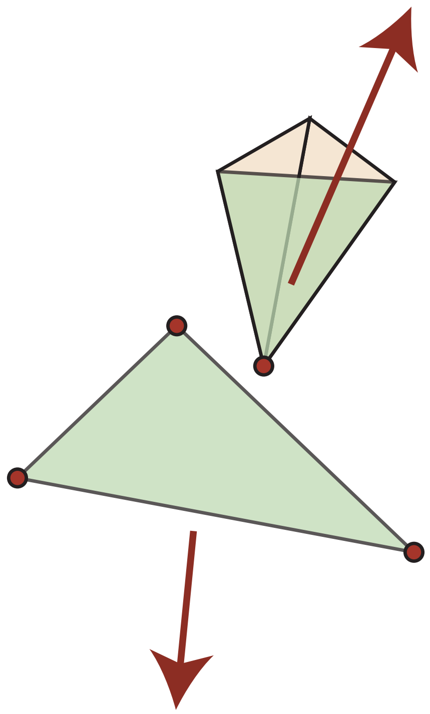
# Collision Response



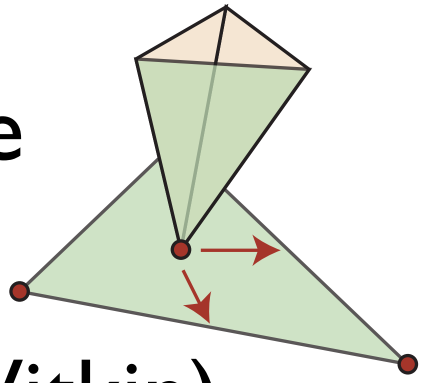
# Collision response



- 4 basic options:
  - Constraint-based
  - Penalty forces
  - Impulse-based
  - Rigid body dynamics (will explain)



# Constraint-based response

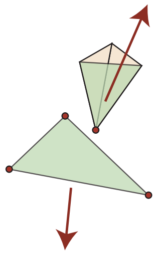


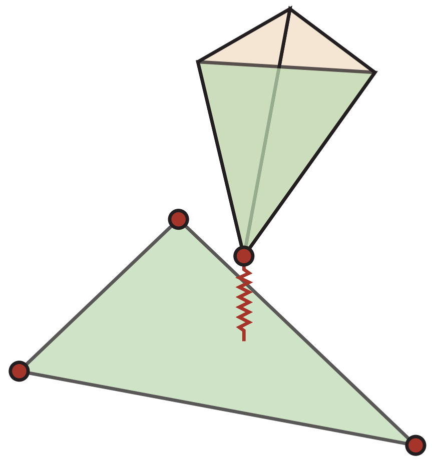
- Assume totally inelastic collision
- Constrain particle to lie on triangle surface
- Benefits:
  - Fast, may not add stiffness (e.g., Baraff/Witkin)
  - No extra damping needed
- Drawbacks
  - Only supports point-face collisions
  - Constraint attachment, release add discontinuities (constants hard to get right)
  - Doesn’t handle self-collisions (generally)
- Conclusion: a good place to start, but not robust enough for heavy-duty work

# Constraint-based response (4)

- Must keep track of constraint forces in the simulator -- that is, the force the simulator is applying to maintain the constraint
- If constraint force opposes surface normal, need to release particle

# Penalty forces





- Apply a spring force that keeps particles away from each other
- Benefits:
  - Easy to fit into an existing simulator
  - Works with all kinds of collisions (use barycentric coordinates to distribute responses among vertices)
- Drawbacks:
  - Hard to tune: if force is too weak, it will sometimes fail; if force is too strong, it will cause the particles to “float” and “wiggle”

# Penalty forces (2)

- In general, penalty forces are not inelastic (springs store energy)
- Can be made less elastic by limiting force when particles are moving away
- Some kind of additional damping may be needed to control deformation rate along surface

# Impulses

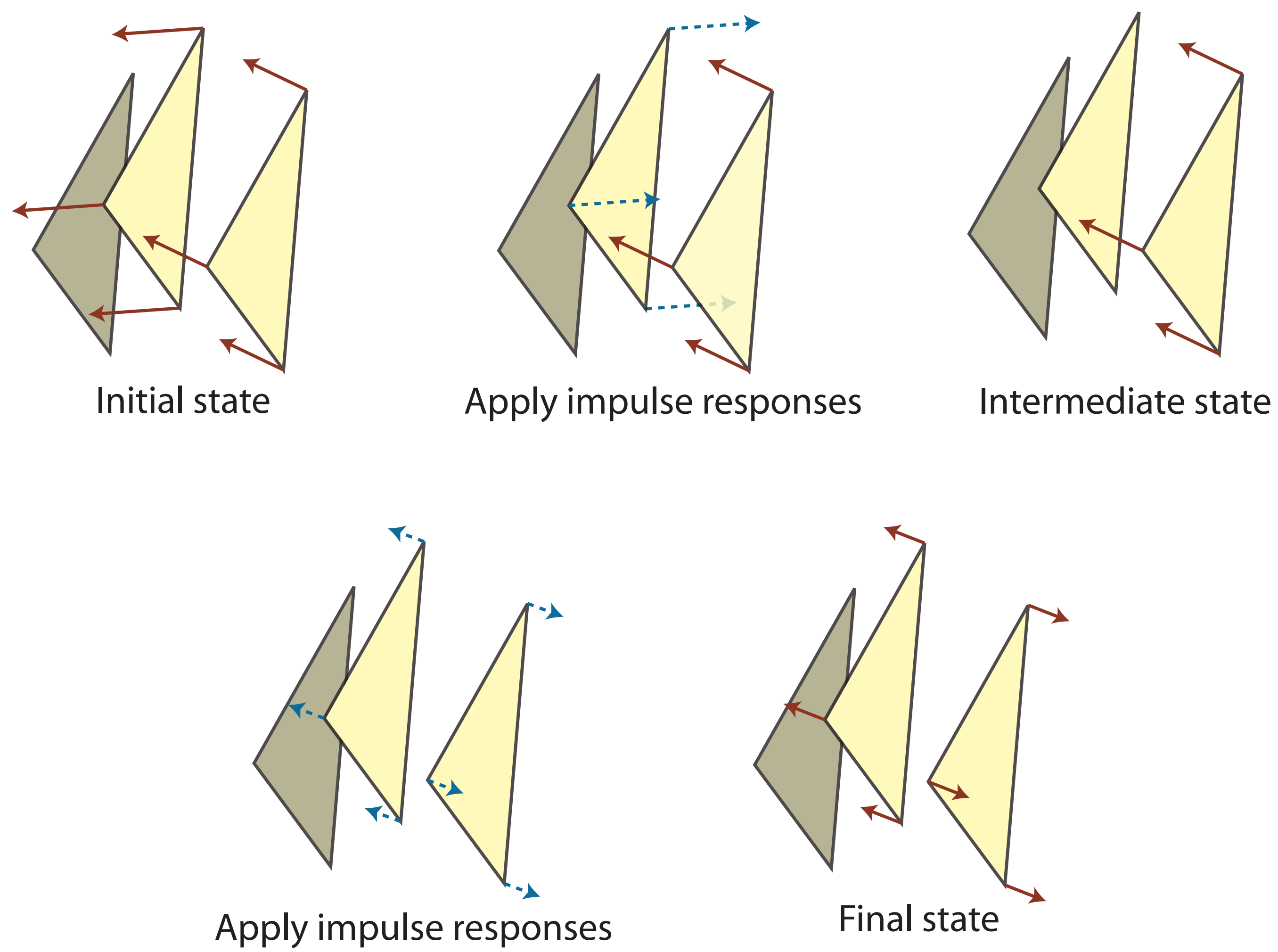
- "Instantaneous" change in momentum

$$\mathbf{J} = \int_{t_i}^{t_f} \mathbf{F}\, dt = \mathbf{p}_f - \mathbf{p}_i$$

- Generally applied outside the simulator timestep
- Benefits
  - Correctly stops all collisions (no sloppy spring forces)
- Drawbacks
  - Can have poor numerical performance
  - Handles persistent contact poorly

# Impulses (2)

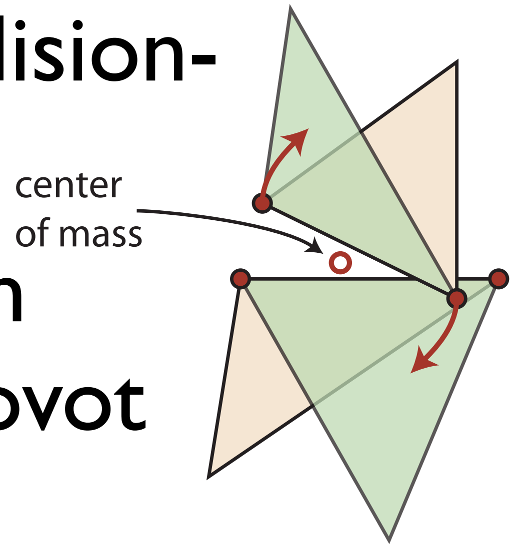
Iteration is generally necessary to remove all collisions.




Convergence may be slow in some cases.

# Rigid collision impact zones

- Basic idea: if a group of particles *start* timestep collision-free, and move as a rigid body throughout the timestep, then they will *end* timestep collision-free.
- We can group particles involved in a collision together and move them as a rigid body (Provot [1997] -- error?, Bridson [2002])




$$x_{CM} = \frac{\sum_i m_i \mathbf{x}_i}{m_i} \qquad v_{CM} = \frac{\sum_i m_i \mathbf{v}_i}{m_i}$$

Center of mass frame

$$\mathbf{L} = \sum_i m_i(\mathbf{x}_i - \mathbf{x}_{CM}) \times (\mathbf{v}_i - \mathbf{v}_{CM})$$

Momentum

$$\mathbf{I} = \sum_i m \left( |\mathbf{x}_i - \mathbf{x}_{CM}|^2 \delta - (\mathbf{x}_i - \mathbf{x}_{CM}) \otimes (\mathbf{x}_i - \mathbf{x}_{CM}) \right)$$

Inertia tensor

$$\omega = \mathbf{I}^{-1}\mathbf{L}$$

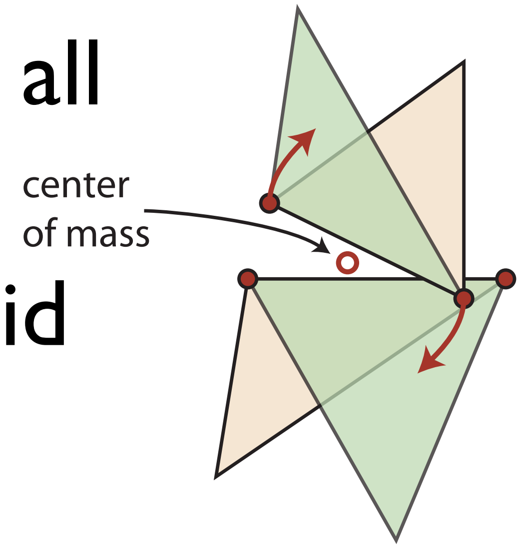
Angular velocity

$$\mathbf{v}_i = \mathbf{v}_{CM} + \omega \times (\mathbf{x}_i - \mathbf{x}_{CM})$$

Final velocity

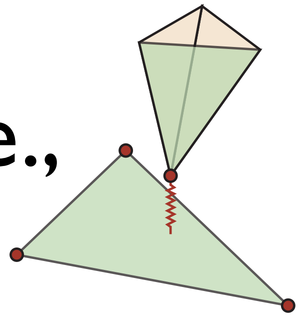
# Rigid collision impact zones (2)

- Note that this is totally failsafe
- We will need to iterate, and merge impact zones as we do (e.g. until the impact zone includes all colliding particles)
- This is best used as a last resort, because rigid body cloth can be unappealing.

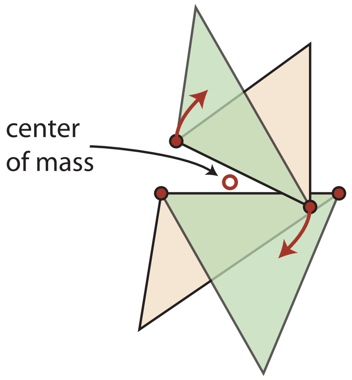


# Combining methods

- So we have:
  - penalty forces - not robust, not intrusive (i.e., integrates with solver)



  - impulses - robust (esp. with iteration), intrusive - but may not converge
  - rigid impact zones - completely robust, guaranteed convergence, but very intrusive



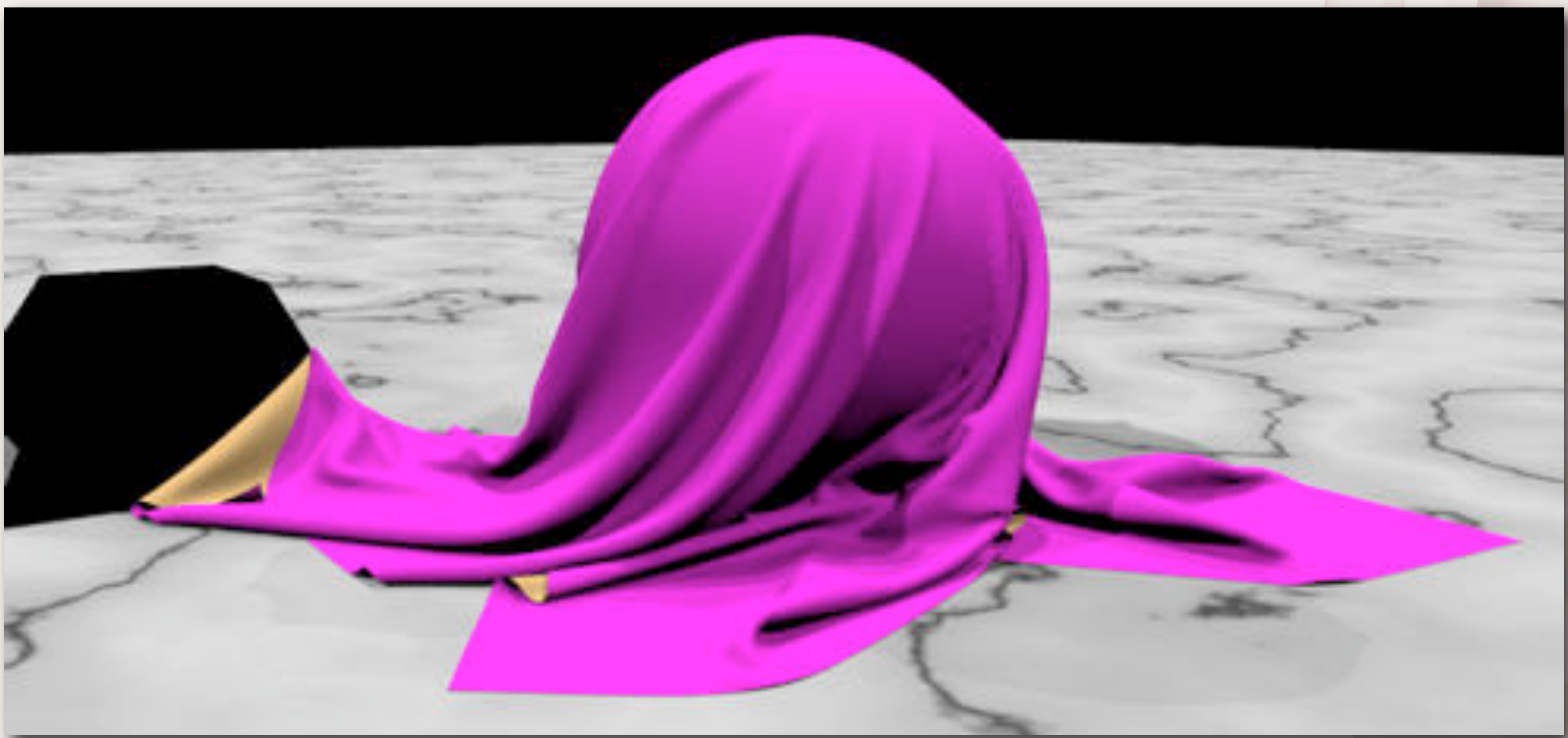

Solution? Use all three! (Bridson et al [2002])

# Combining methods (2)

Basic methodology (Bridson et al [2002]):

1. Apply penalty forces (implicitly)
2. While there are collisions left
   1. Check robustly for collisions
   2. Apply impulses
3. After several iterations of this, start grouping particles into rigid impact zones
4. 

Objective: guaranteed convergence with minimal interference with cloth internal dynamics



# Efficient Simulation of Inextensible Cloth

Rony Goldenthal
The Hebrew University
Columbia University

David Harmon
Columbia University

Raanan Fattal
UC Berkeley

Michel Bercovier
The Hebrew University

Eitan Grinnspun
Columbia University

# Outline

- Cloth Collisions
  - Collision Detection
  - Collision Handling
- Hair Collisions
  - Like Cloth
  - Continuum Approaches

# Hair Collisions

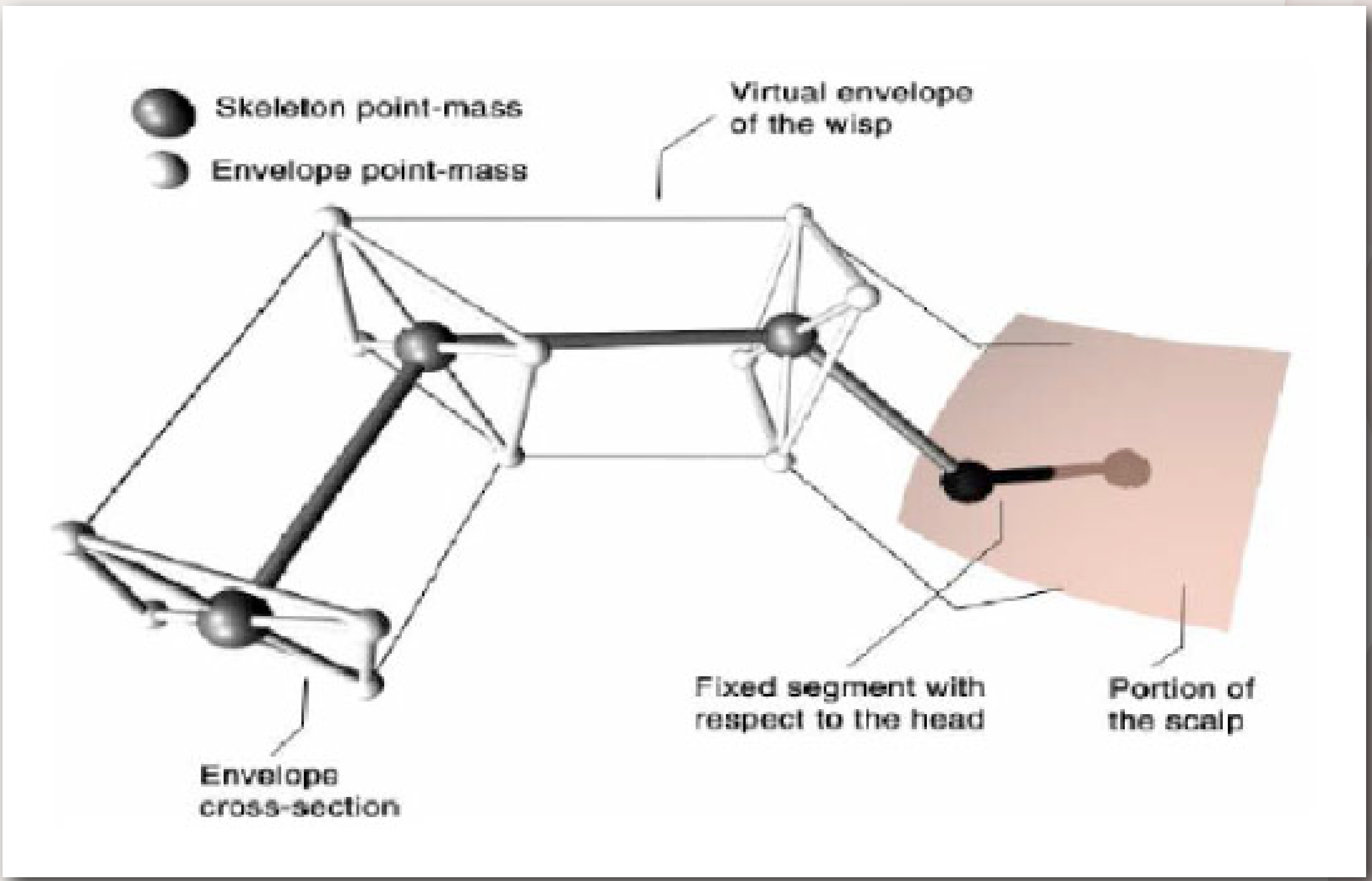
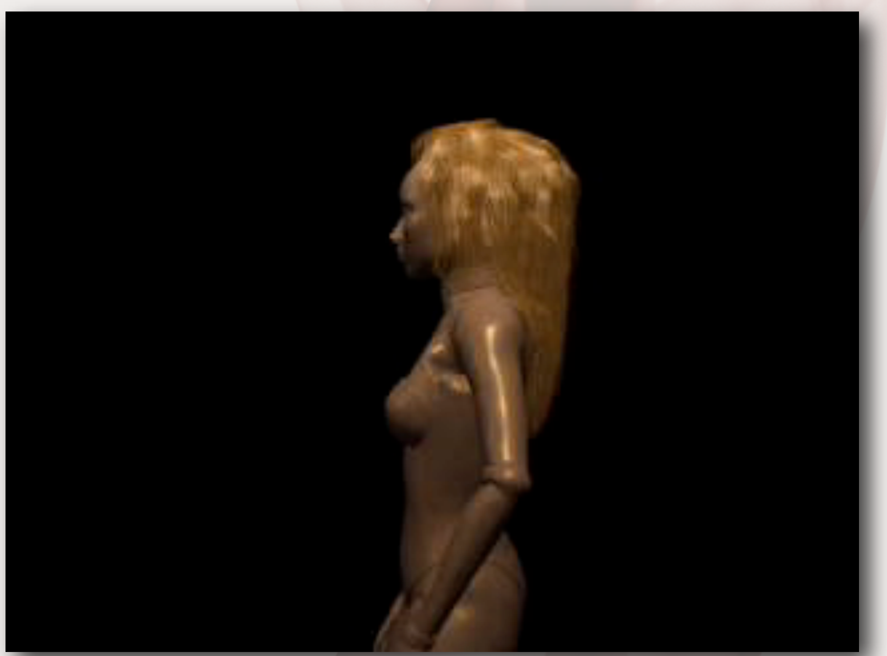



Figure 2.2: Elements defining a deformable volumetric wisp [PCP01a].

# Examples

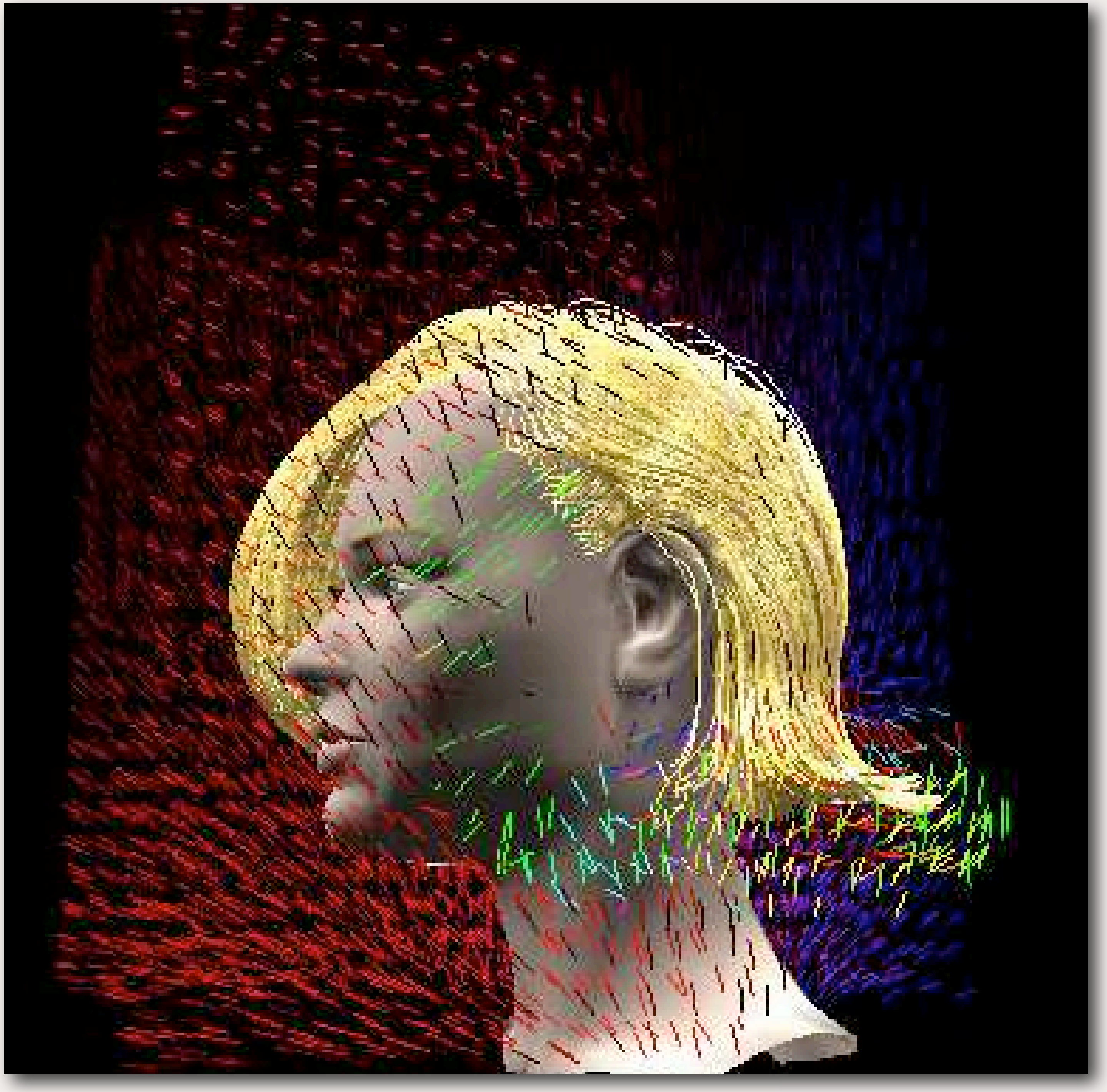
Short

Medium



Long

# Continuum Hair



# Examples

# Current Challenges

- Cloth:
  - Different kinds of cloth.
  - Realtime.
  - Are we doing too much work?
- Hair:
  - Beards and Mustaches.

# Question

- What are the relevant properties of crowds?
- Can crowds be modeled as particles?
- How?
- What phenomena does your algorithm capture, what doesn't it?