SIGGRAPH2011

Solid Simulation with Oriented Particles

Matthias Müller

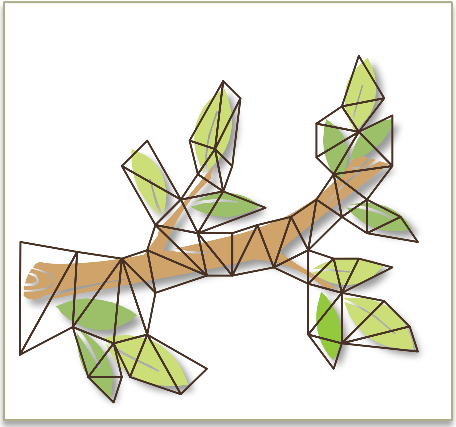
Nuttapong Chentanez

NVIDIA

Motivation

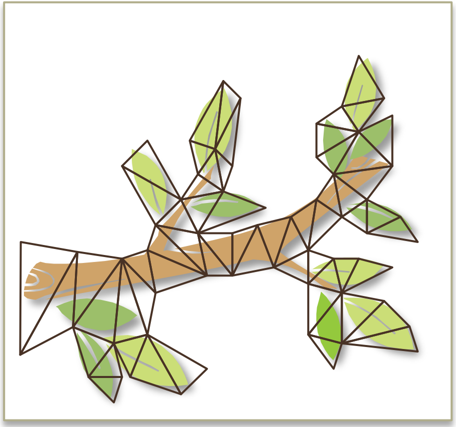
Traditional Deformable Simulation

- Embed visual mesh in tetrahedral mesh
- Deform visual mesh using barycentric interpolation



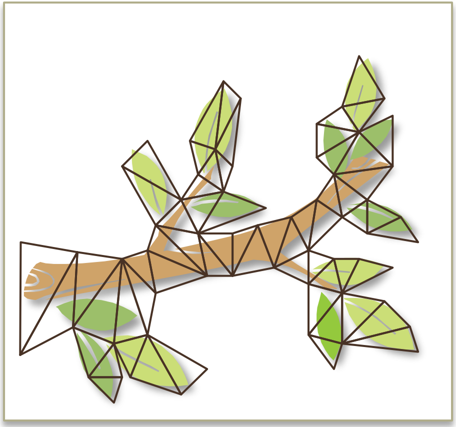
Tetrahedral Mesh

- Need enough tetrahedra to
  - Resolve separate parts
  - Hide piecewise linear deformation
  - Good approximation for collision handling
- Mesh creation non-trivial

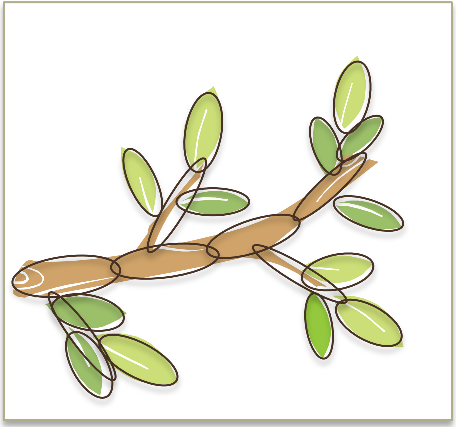


New Approach

- Approximate the visual mesh with a sparse set of oriented particles



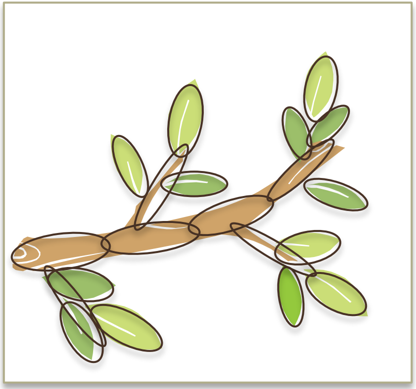
60 triangles (~ 200 tetras)



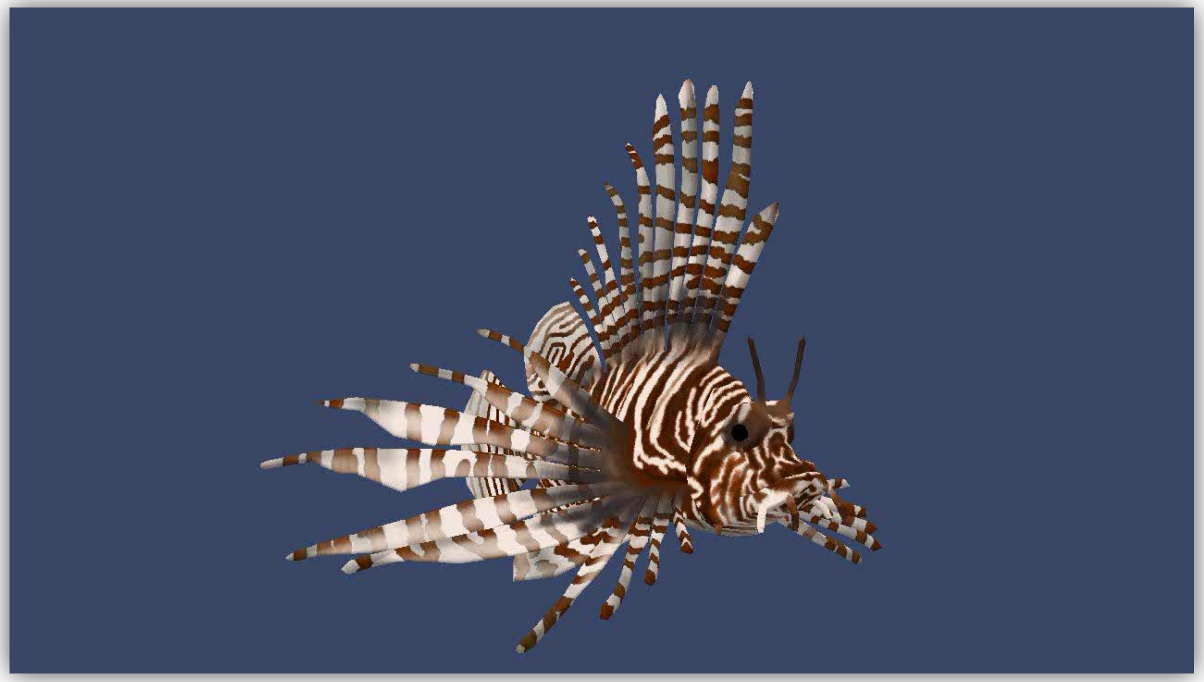
20 ellipsoids

Oriented Particles

- Orientation information is used
  - To position anisotropic collision shapes (ellipsoids)
  - To make the simulation stable in sparse regions
  - For rubust skinning of the visual mesh



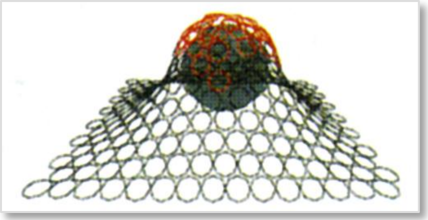
Example



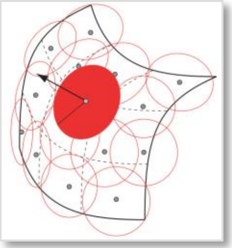
Related Work

Oriented Particles

- Term introduced by



[Szeliski et al., 1992]

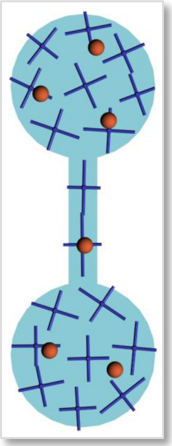


[Pfister et al., 2000]

- Used for surface modeling and rendering

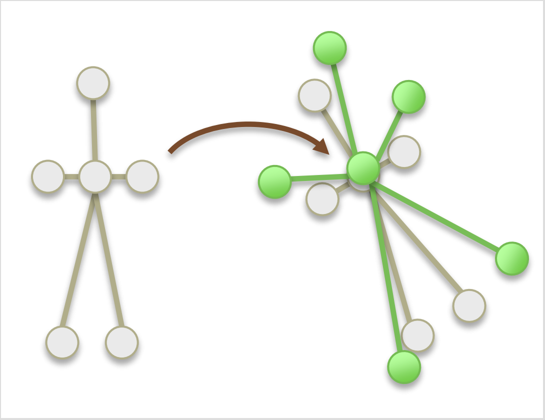
Elastons

- [Martin et al., 2010]
  - 1D, 2D and 3D structures
  - Energy integration points with orientation
  - Accurate: Continuum mechanics based
  - Non-real-time: Seconds / frame



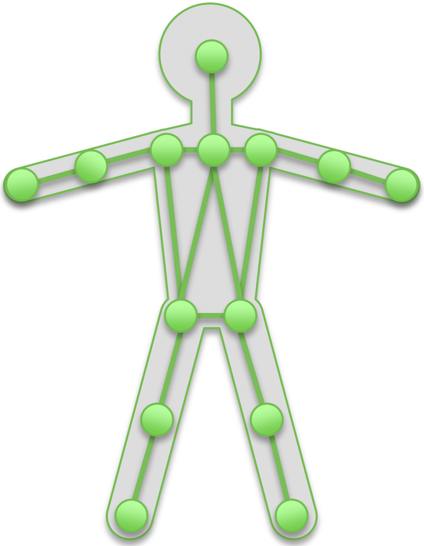
Shape Matching

- [Müller et al., 2005]
  - Geometry based model
  - Simple and fast
  - Fails in sparsely sampled regions



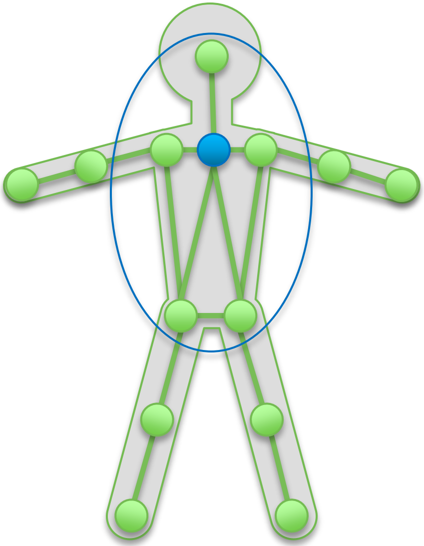
Simulation Method

Mesh Creation

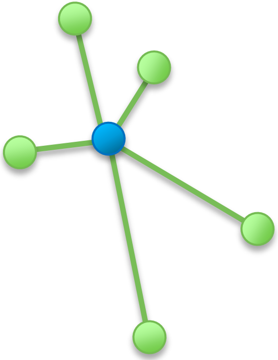


- Cover the visual mesh with particles
- Create arbitrary connectivity
- Manual and automatic tools

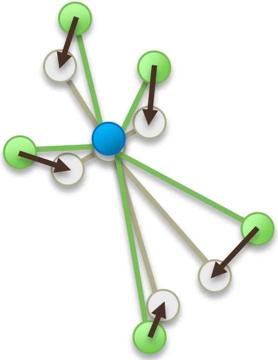
Shape Matching Simulation



one constraint per particle

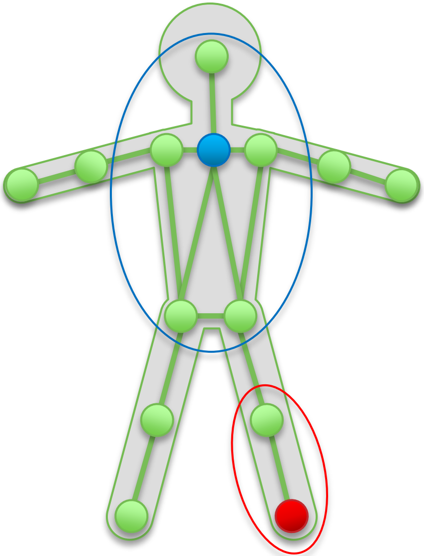


deformed state



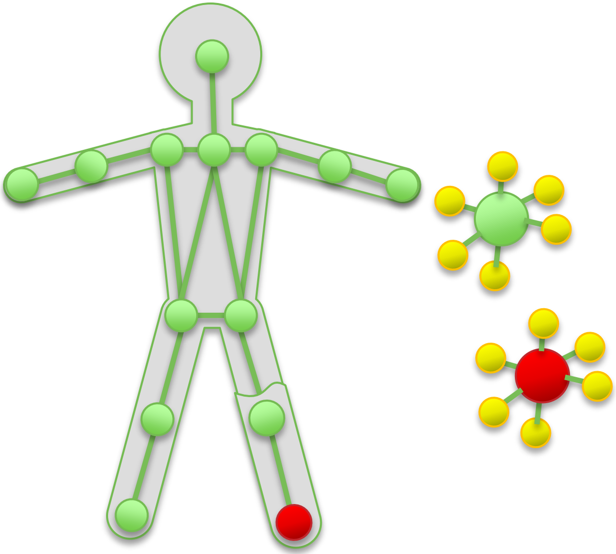
move towards
matched
rest configuration

Singularity Problem



- Region under-sampled in 1D and 2D structures
- Rest state transformation not unique

Our Solution



- Replace existing particles with 6 virtual particles (conceptually)
- Distance and relative arrangement fixed
- Need particle orientation!
- Orientation influences other parts
  → must be properly simulated

Simulate Orientation State

Prediction

$$\mathbf{x}_p \leftarrow \mathbf{x} + \mathbf{v}\Delta t$$

$$\mathbf{q}_p \leftarrow \left[\frac{\omega}{|\omega|}\sin\left(\frac{|\omega|\Delta t}{2}\right), \cos\left(\frac{|\omega|\Delta t}{2}\right)\right]\mathbf{q}$$

Integration

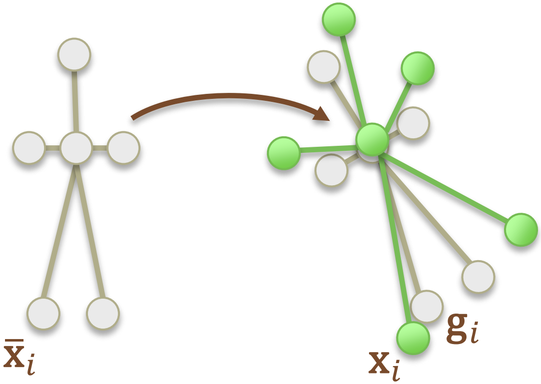
$$\mathbf{v} \leftarrow (\mathbf{x}_p - \mathbf{x})/\Delta t$$

$$\mathbf{x} \leftarrow \mathbf{x}_p$$

$$\omega \leftarrow axis(\mathbf{q}_p\mathbf{q}^{-1}) \cdot angle(\mathbf{q}_p\mathbf{q}^{-1})/\Delta t$$

$$\mathbf{q} \leftarrow \mathbf{q}_p$$

Shape Matching



$$\mathbf{g}_i = \mathrm{R}(\bar{\mathbf{x}}_i - \bar{\mathbf{c}}) + \mathbf{c}$$

$$\mathbf{c} = \sum_i m_i \mathbf{x}_i \,/\, \sum_i m_i$$

$$\bar{\mathbf{c}} = \sum_i m_i \bar{\mathbf{x}}_i \,/\, \sum_i m_i$$

$$\boxed{\mathrm{A} = \sum_i m_i (\mathbf{x}_i - \mathbf{c})(\bar{\mathbf{x}}_i - \bar{\mathbf{c}})^T}$$

$\mathrm{A} = \mathrm{RS}$ (polar decomposition)

Oriented Particle

$$\mathbf{A} = \sum_i m_i(\mathbf{x}_i - c)(\overline{\boldsymbol{x}}_i - \overline{\boldsymbol{c}})^T$$

Moment matrix of a single spherical particle with radius *r* :

$$\boldsymbol{A}_{sphere} = \int_{V_r} \rho(\mathrm{R}\mathbf{x})\mathbf{x}^T dV = \rho\mathrm{R}\int_{V_r} \mathbf{x}\mathbf{x}^T dV = \frac{1}{5}mr^2\mathrm{R}$$

$$\boldsymbol{A}_{ellipsoid} = \frac{1}{5}m\begin{bmatrix} a^2 & 0 & 0 \\ 0 & b^2 & 0 \\ 0 & 0 & c^2 \end{bmatrix}\mathrm{R}$$

Generalized Shape Matching

Particle A_i are evaluated w.r.t. origin.

Factored out center: [Rivers and James, 2007]

$$\sum_i m_i(\mathbf{x}_i - \mathbf{c})(\bar{\mathbf{x}}_i - \bar{\mathbf{c}})^T = \sum_i m_i\, \mathbf{x}_i \bar{\mathbf{x}}_i{}^T - M\mathbf{c}\bar{\mathbf{c}}^T$$

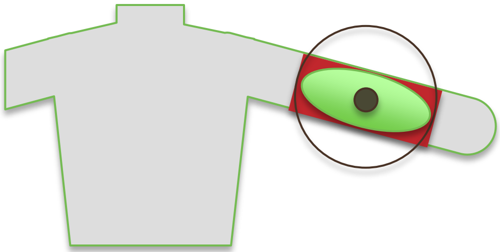
$$\underbrace{\int_{V_r} \rho(\mathrm{R}\mathbf{x})\mathbf{x}^T dV}_{\mathbf{A}_i} = \underbrace{\int_{V_r} \rho(\mathrm{R}\mathbf{x}+\mathbf{x}_i - \boldsymbol{c})(\mathbf{x}+\bar{\mathbf{x}}_i - \bar{\mathbf{c}})^T dV}_{\mathbf{A}_i{}^{global}} - m_i(\mathbf{x}_i - c)(\bar{\boldsymbol{x}}_i - \bar{\boldsymbol{c}})^T$$

$$\boxed{\mathbf{A} = \sum_i (\mathbf{A}_i + m_i(\mathbf{x}_i - c)(\bar{\boldsymbol{x}}_i - \bar{\boldsymbol{c}})^T)}$$

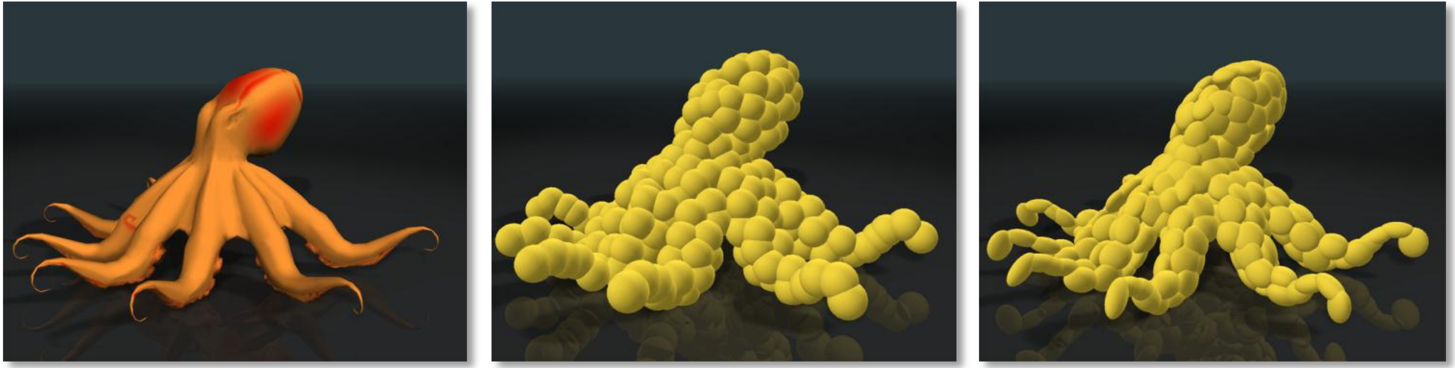
Collision Handling and Skinning

Collision Primitives

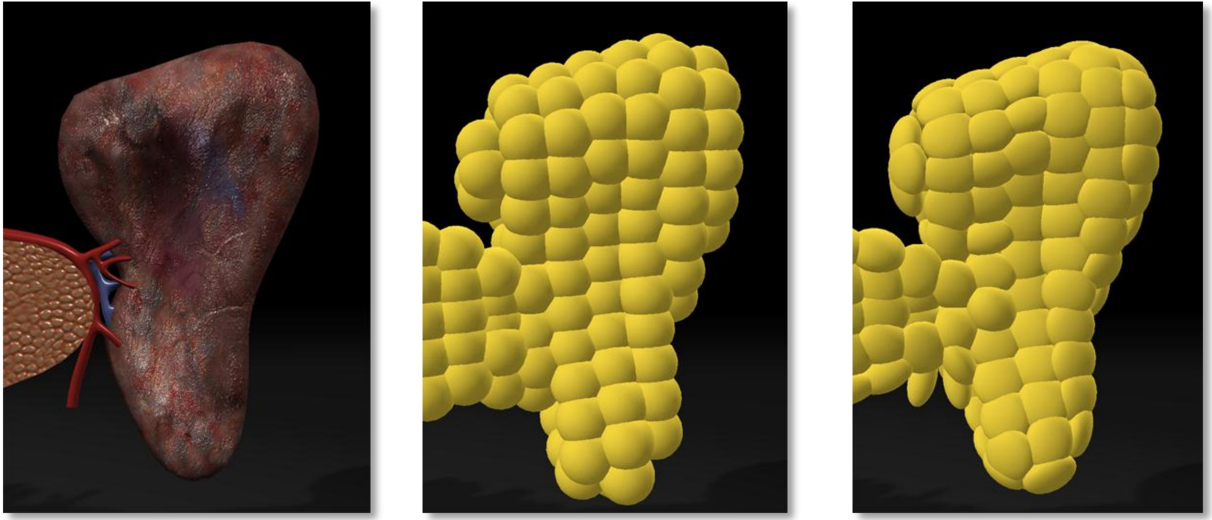
- Orientation information allows anisotropic particle shapes (ellipsoids)
- Initial radii and pose by OBB of mesh neighborhood



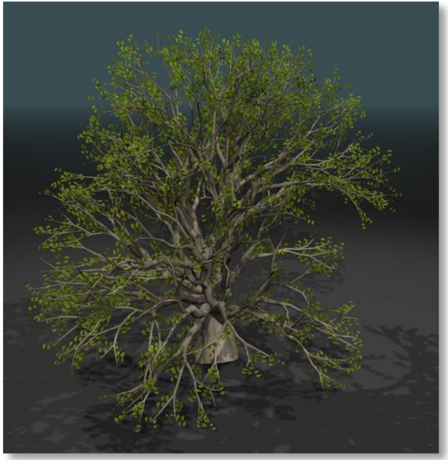
Ellipsoid Example: Octopus

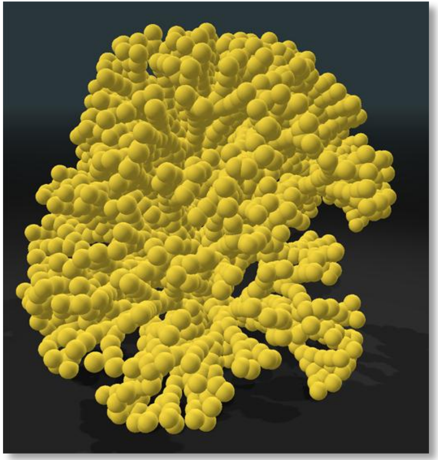


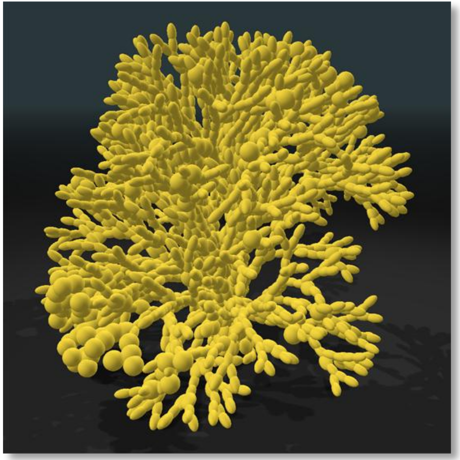
Ellipsoid Example: Pancreas



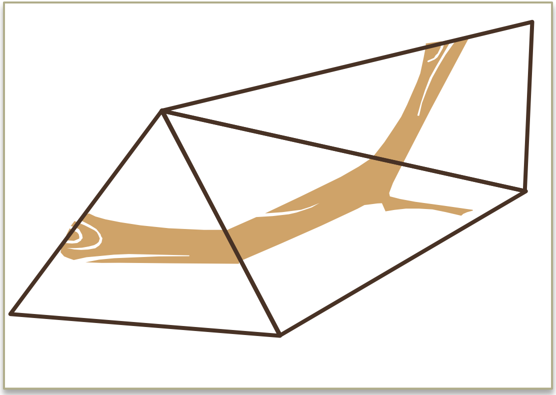
Ellipsoid Example: Tree



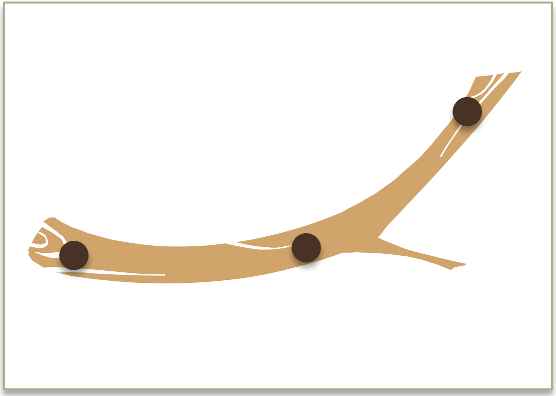




Skinning Methods

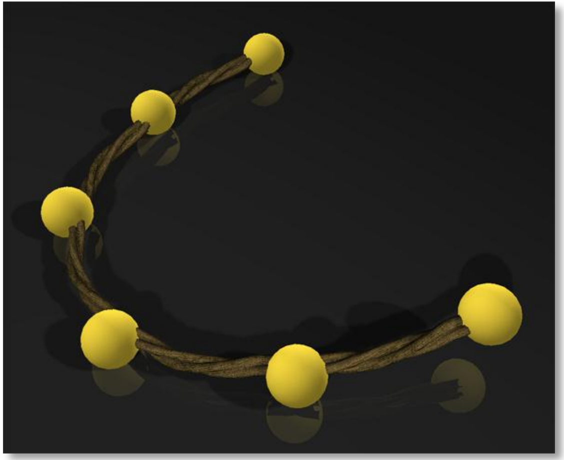


- Barycentric interpolation w.r.t. surrounding tetrahedron
- Piecewise linear



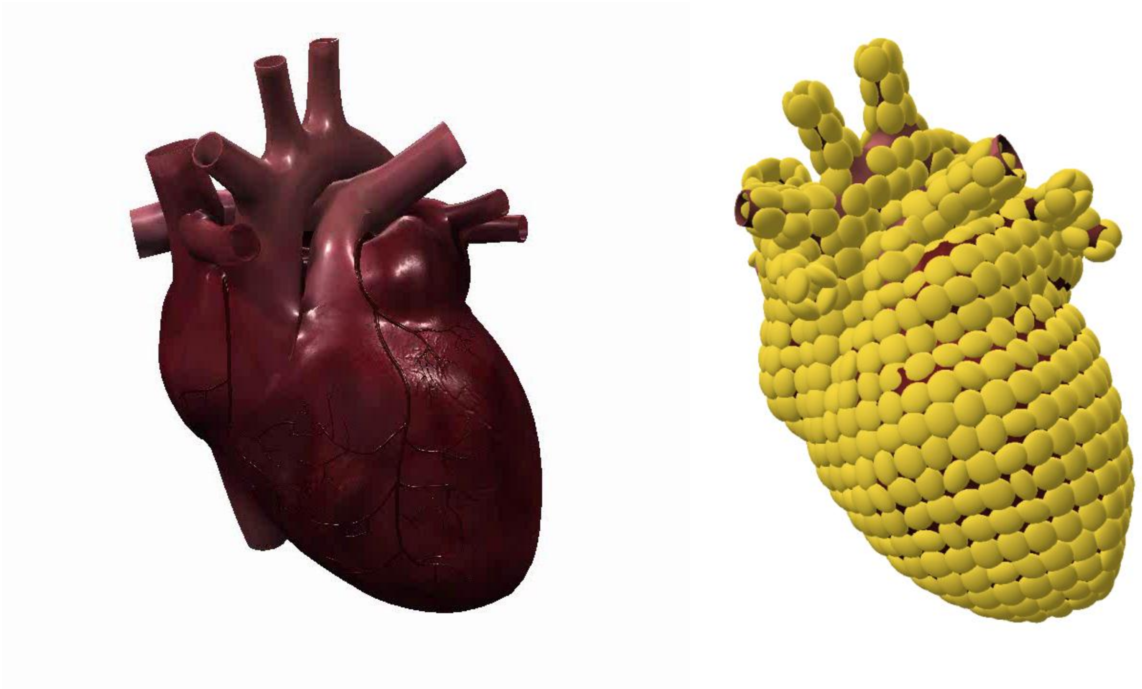
- Linear blend skinning w.r.t. k closest oriented particles
- Curved

Curved Interpolation



Results

Intel Core i7 CPU @ 3 GHz (simulation)
GeForce GTX 480 (skinning)



900 particles, 63k triangles, 60 fps

3000 particles, 90k triangles, 25 fps

Arbitrary Shape Match Groups

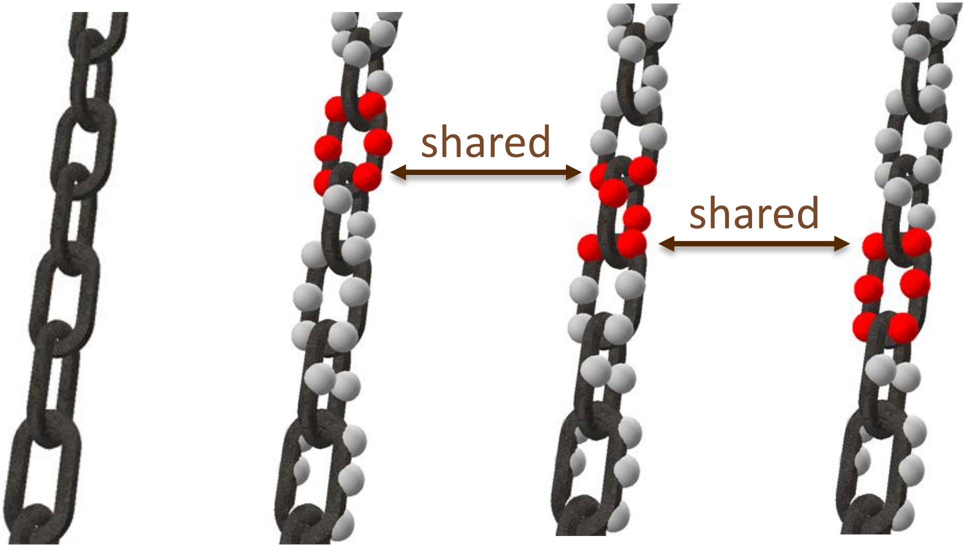
- Rigid parts
- Joints via shared particles
  - Free rotation only if shared particles non-oriented!

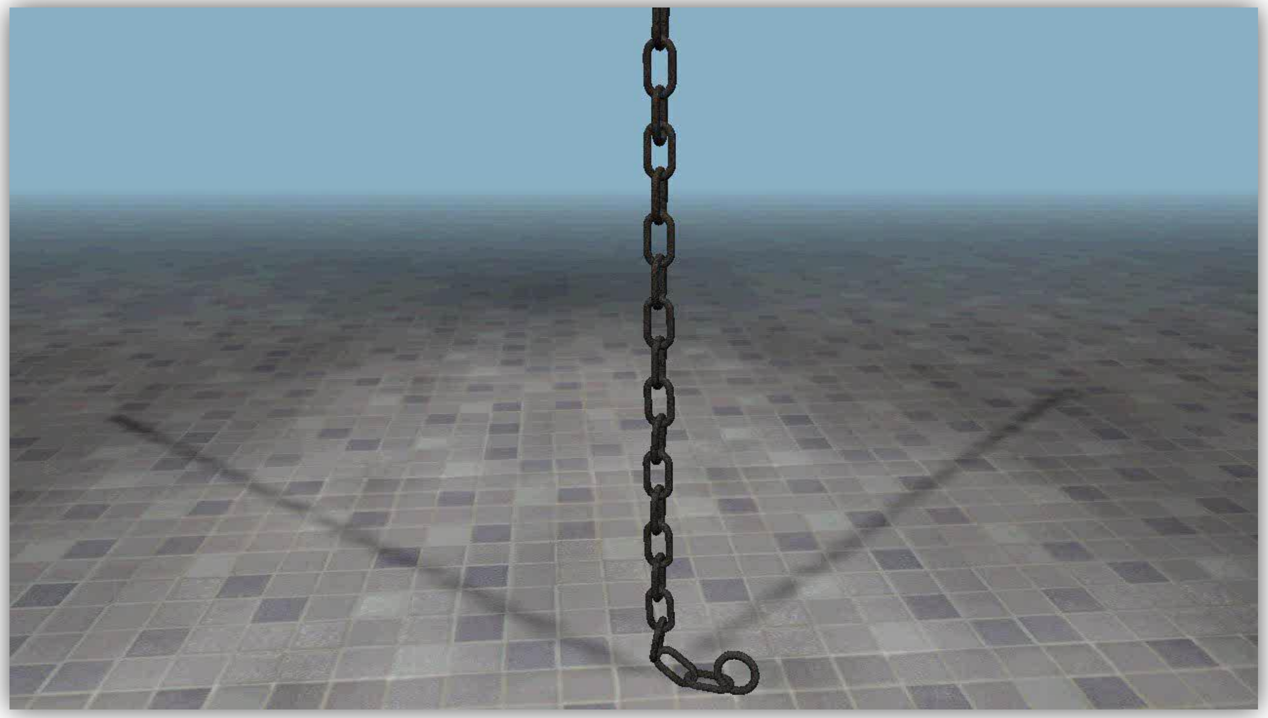
$$\mathbf{A} = \sum_i \left(\mathbf{A}_i + m_i(\mathbf{x}_i - c)(\bar{\boldsymbol{x}}_i - \bar{\boldsymbol{c}})^T\right)$$

omit for shared particles

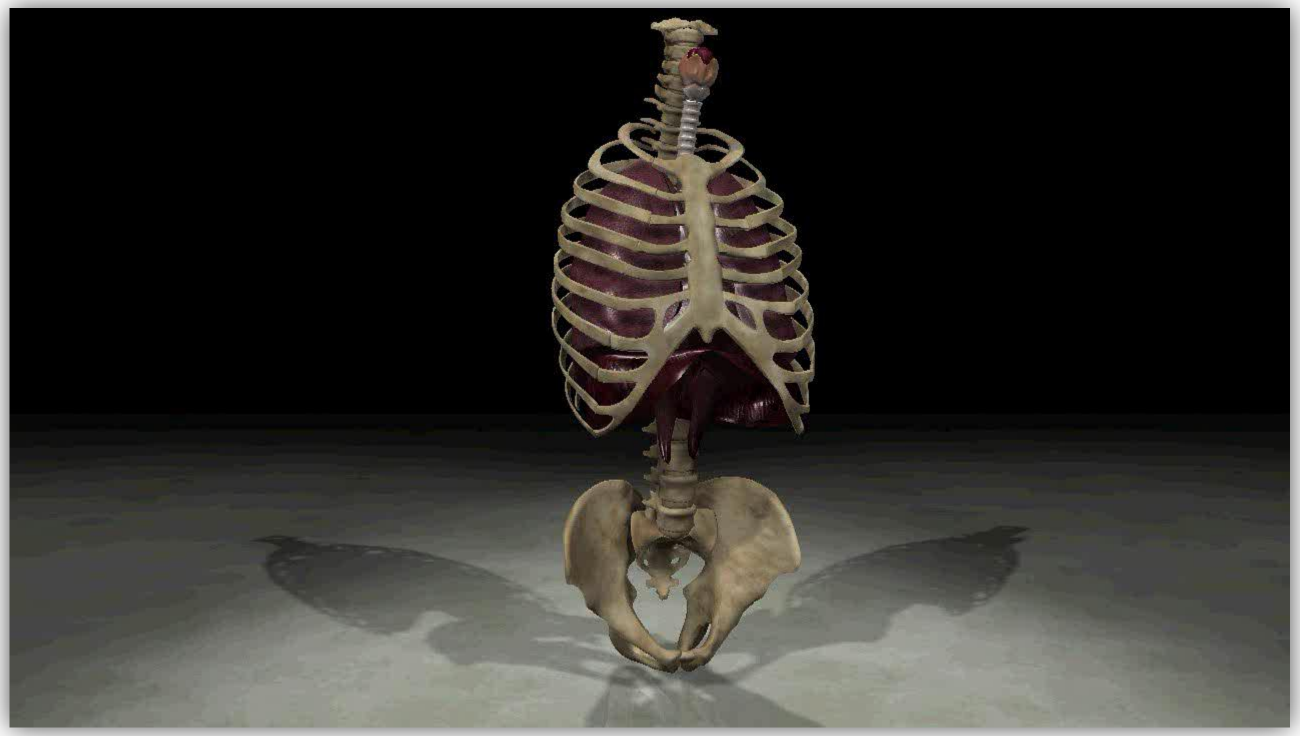
2000 particles, 240k triangles, 40 fps

Simplified Chain





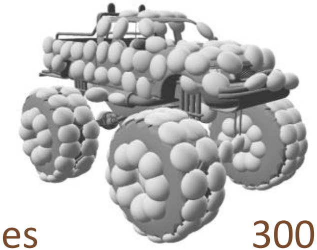
600 chain links @ 45 fps (simulation + skinning)



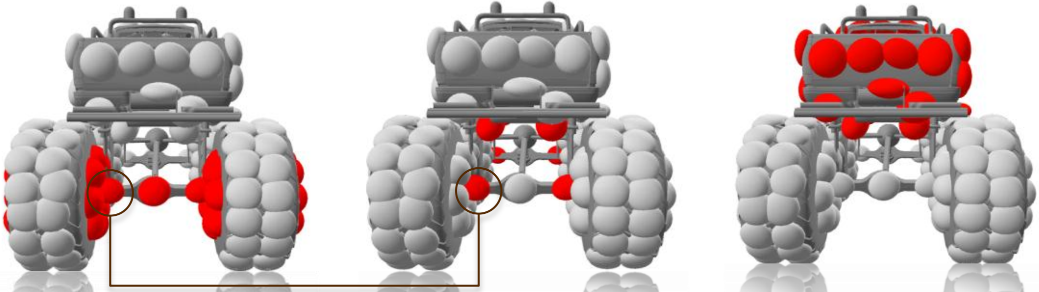
1000 particles, 100k triangles, 35 fps

Monster Truck

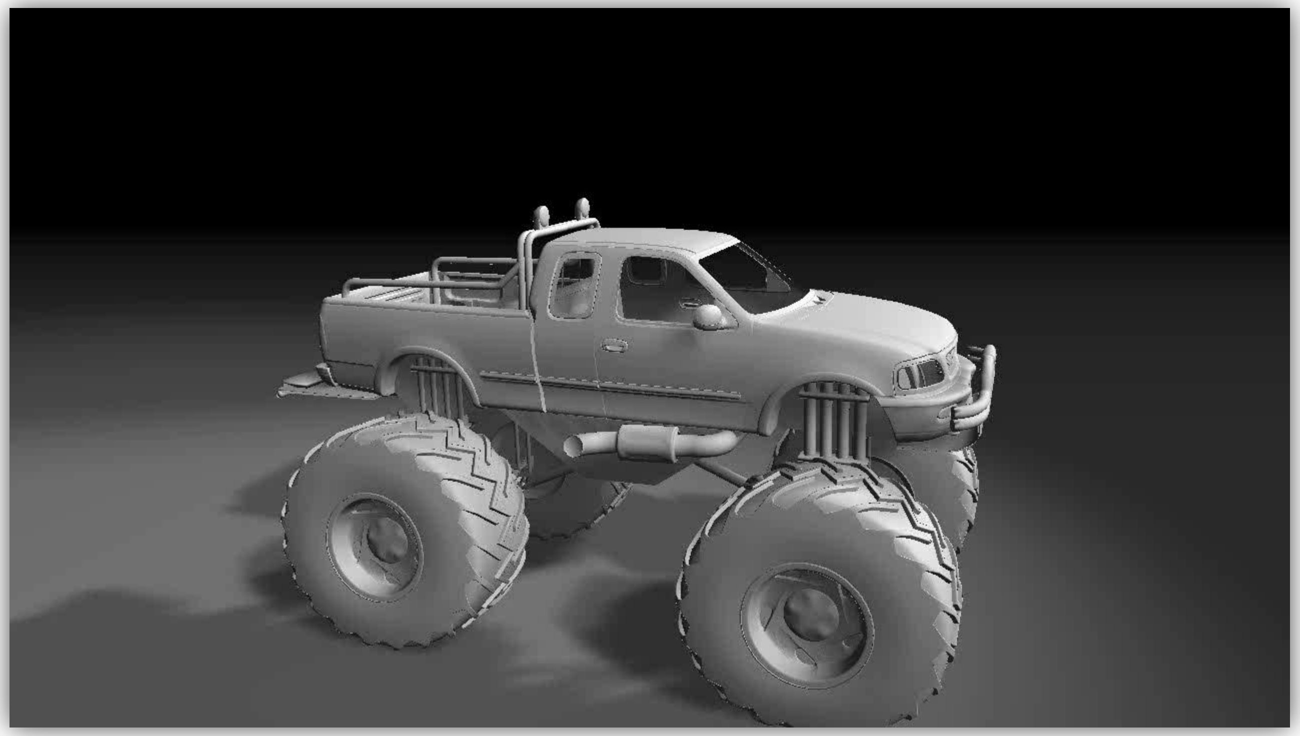
100k triangles



300 particles



2 x axle (rigid) chassis (rigid) body (plastic)



10 instances @ 20 fps (simulation + skinning)

Conclusion

- Oriented particles for simulation
- Stabilization, tighter collision volumes, skinning
- Future
  - Volume conservation
  - GPU implementation + game engine integration

Thank you for your attention!