Meshless Deformations Based on Shape Matching

M. Müller, B. Heidelberger, M. Teschner, M. Gross

Proceedings of SIGGRAPH'05, 2005
The Goal

To simulate deformable objects in a computationally efficient and stable way

Oriented towards the gaming and animation industries
Existing methods

Not applicable because they are either:

– Difficult to code (FEM, BEM)
– Too slow (FEM)
– Not stable (explicit mass spring)
– Memory consuming
  (Eigenmodes, stiffness matrices, Green‘s functions, motion databases)
Mickey Mouse physics

To simplify the problem, the authors:

• Relax the requirement of realism
• Do not model material properties
• Avoid using the connectivity information
Simple Model

• Solid represented by points with mass
  – Simple to generate
  – Animated as a particle system

• No connectivity needed
  – Volumetric meshes not available in games
  – Each point path is calculated independently

• Explicit integration
  – Simple to code
  – Efficient to compute
  – Unconditionally stable
Basic Algorithm

• Given $n$ points with masses $m_i$ and original positions $x_{0_i}$.
Basic Algorithm

• Animate the points as unconnected particle system considering only external forces

\[ m_i, x^0_i \]
Basic Algorithm

- At every time step, match the original configuration \( x^0_i \) to the actual configuration \( x_i \) yielding goals \( g_i \)
Basic Algorithm

• Pull the actual points towards the goal positions $g_i$
Algorithm

Has two main components:

1. Finding the optimal transformation
   (shape matching to find the goal points)

2. Moving the particles towards those points
   (modeling acceleration and velocity)

Trade Off:

Subside to external forces vs. Retain the original configuration
Shape Matching

• Given \( \mathbf{x}_i^0, \mathbf{x}_i \) and weights \( w_i \)
• Find rigid body transform \( t_0, t, R \) minimizing

\[
\sum_i w_i \left( R(\mathbf{x}_i^0 - t_0) + t - \mathbf{x}_i \right)^2
\]

• Select \( w_i = m_i \)
• Optimal \( t_0 \) is center of mass of the \( \mathbf{x}_i^0 \)
• Optimal \( t \) is center of mass of the \( \mathbf{x}_i \)

[Kanatani 1994], [Umeyama 1991] and [Lorusso et al. 1995]
Optimal Rotation

- Define $p_i = x_i - x_{cm}$ and $q_i = x_i^0 - x_{cm}^0$

- Minimize $\sum_i m_i (A q_i - p_i)^2$

- With optimal linear transform

$$A = \left( \sum_i m_i p_i q_i^T \right) \left( \sum_i m_i q_i q_i^T \right)^{-1} = A_{pq} A_{qq}$$

- Optimal $R$ from polar decomposition of $A_{pq}$
Goal positions

\[
A = \left( \sum_i m_i p_i q_i^T \right) \left( \sum_i m_i q_i q_i^T \right)^{-1} = A_{pq} A_{qq}
\]

\[
R = A_{pq} S^{-1} \quad \text{where} \quad S = \sqrt{A_{pq}^T A_{pq}}
\]

\[
g_i = R(x_i^0 - x_{cm}^0) + x_{cm}
\]
Stable Explicit Integration

\[ \mathbf{v}_i(t + h) = \mathbf{v}_i(t) + \alpha \frac{\mathbf{g}_i(t) - \mathbf{x}_i(t)}{h} + \frac{h}{m_i} \mathbf{f}_{\text{ext}}(t) \]

\[ \mathbf{x}_i(t + h) = \mathbf{x}_i(t) + h \mathbf{v}_i(t + h) \]

- Unconditionally stable
- Controls stiffness (rigid for \( \alpha = 1 \))
- Pulled out of a hat
Compare to Euler Integration

\[ f = -k(x(t) - l_0) \]

\[ \Delta x = -\frac{h^2 k}{m}(x(t) - l_0) \]

\[
\begin{align*}
  v(t + h) &= v(t) + h \frac{-k(x(t) - l_0)}{m} \\
  x(t + h) &= x(t) + hv(t + h),
\end{align*}
\]

\[
A = \begin{bmatrix}
  1 & -\frac{kh}{m} \\
  h & 1 - \frac{h^2 k}{m}
\end{bmatrix}
\]

\[ e_0 = 1 - \frac{1}{2m}(h^2 k - \sqrt{-4mh^2 k + h^4 k^2}) \]

\[ e_1 = 1 - \frac{1}{2m}(h^2 k + \sqrt{-4mh^2 k + h^4 k^2}) \]

\[ e_0 < 1 \text{ for } h^2 k \to \infty \]

\[ e_1 < 1 \text{ iff } h < 2\sqrt{\frac{m}{k}} \]
Compare to Euler Integration

\[ f = -k(x(t) - l_0) \]

\[
\begin{align*}
v_i(t + h) &= v_i(t) + \alpha \frac{g_i(t) - x_i(t)}{h} + hf_{ext}(t)/m_i \\
x_i(t + h) &= x_i(t) + hv_i(t + h),
\end{align*}
\]

\[
\begin{bmatrix} v(t + h) \\ x(t + h) \end{bmatrix} = \begin{bmatrix} 1 & -\alpha/h \\ h & 1 - \alpha \end{bmatrix} \begin{bmatrix} v(t) \\ x(t) \end{bmatrix} + \begin{bmatrix} \alpha l_0/h \\ \alpha l_0 \end{bmatrix}
\]

\[ e_{0,1} = (1 - \alpha/2) \pm i\sqrt{4\alpha - \alpha^2}/2 \]

\[ \clubsuit e_{0,1} \clubsuit / \]

1
Extensions
(Shape matching cont’d)

1. Linear Deformations
2. Quadratic Deformations
3. Cluster-based Deformations
4. Plasticity
Linear Deformations

Recall that \( \mathbf{R} \) is the polar decomposition of \( \mathbf{A}_{pq} \) \( \left( \mathbf{A}_{pq} = RS, \text{ where } S = \sqrt{\mathbf{A}^T_{pq} \mathbf{A}_{pq}} \right) \)

Now, instead of using only \( \mathbf{R} \) to compute the goal points, also use the matrix \( \mathbf{A} \)

\[
g_i = (\beta \mathbf{A} + (1 - \beta) \mathbf{R})(\mathbf{x}_i^0 - \mathbf{x}_{cm}^0) + \mathbf{x}_{cm}
\]

Objects are allowed to shear and stretch
Quadratic Deformations

Further modify the matrix $\mathbf{A}$ by setting

$$\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{A} & \mathbf{Q} & \mathbf{M} \end{bmatrix}$$

and minimizing $\sum_i m_i (\tilde{\mathbf{A}} \mathbf{q}_i - \mathbf{p}_i)^2$ where

$$\mathbf{q} = [q_x, q_y, q_z, q_x^2, q_y^2, q_z^2, q_x q_y, q_x q_z, q_y q_z]$$

$$\mathbf{g}_i = (\beta \tilde{\mathbf{A}} + (1 - \beta) \mathbf{R})(\mathbf{x}_i^0 - \mathbf{x}_{cm}^0) + \mathbf{x}_{cm}$$

Objects are allowed to twist and bend
Cluster Based Deformations

The set of particles is divided into overlapping clusters (cubes in space)

Run shape matching for each cluster

Average goal positions for each point
Plasticity

Recall $A = RS$ where $R$ is the rotational part of the transformation

In each cluster, if the unrotated deformation $\|S - I\|_2$ exceeds a threshold $c_{\text{yield}}$, incorporate it into the original shape
Conclusions

• Pros
  – Straight forward to implement
  – Stable and scalable
  – Time and memory efficient

• Cons
  – Geometric - **not** physically derived
  – Tuned via non-physical (geometric) parameters