Point-Based Animation of Elastic, Plastic, and Melting Objects

M. Muller, R. Keiser, A. Nealen, M. Pauly, M. Gross and M. Alexa

Introduction

Point Based Simulation is an alternative to:

- Mass-Spring Systems
- Meshed FEM methods

Benefits

Allows for Complex effects like:

- Melting
- Solidifying
- Splitting
- Fusion

…and omits explicit connectivity information.

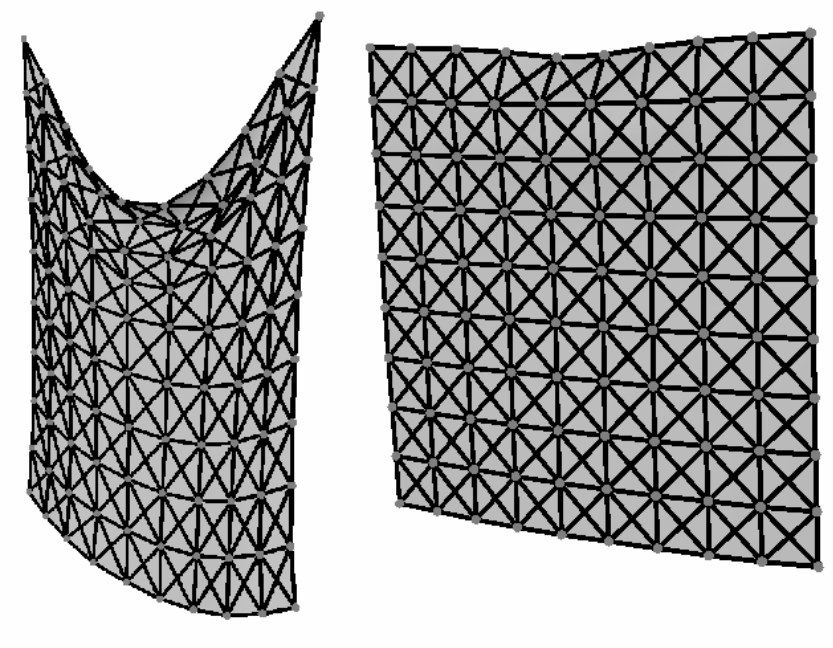
Related Work

- Mesh Based Physical Models
- Mesh Free Physical Models
- Point Based Surface Modeling

Mesh Based Physical Models

- Early Work: Dynamics of Deformable model calculated using finite difference discretization (FD).

- Mass-Spring Systems
- Boundary Element Method (BEM)
- Finite Element Method (FEM)

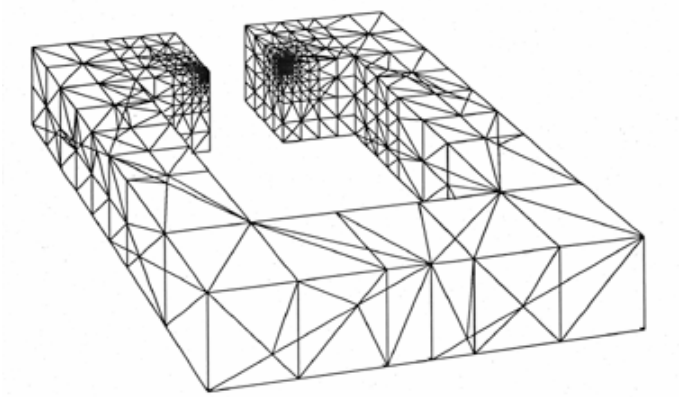


Mesh Free Physical Models

- Desbrun and Cani '95
  - Particle system with simple inter-particle forces
- Smoothed Particle Hydrodynamics (SPH)
  - Use discrete particles to approximate physical values and spatial derivatives.
- Others
  - Particle Volumes, using Lennard-Jones potential energy functions.

Point Based vs. Mesh Based

- Mesh Based Approach



- Point Based Approach

Model Overview

1. Physical Model, Continuum Equations
2. Discretization of Model and Simulation
3. Surfacing

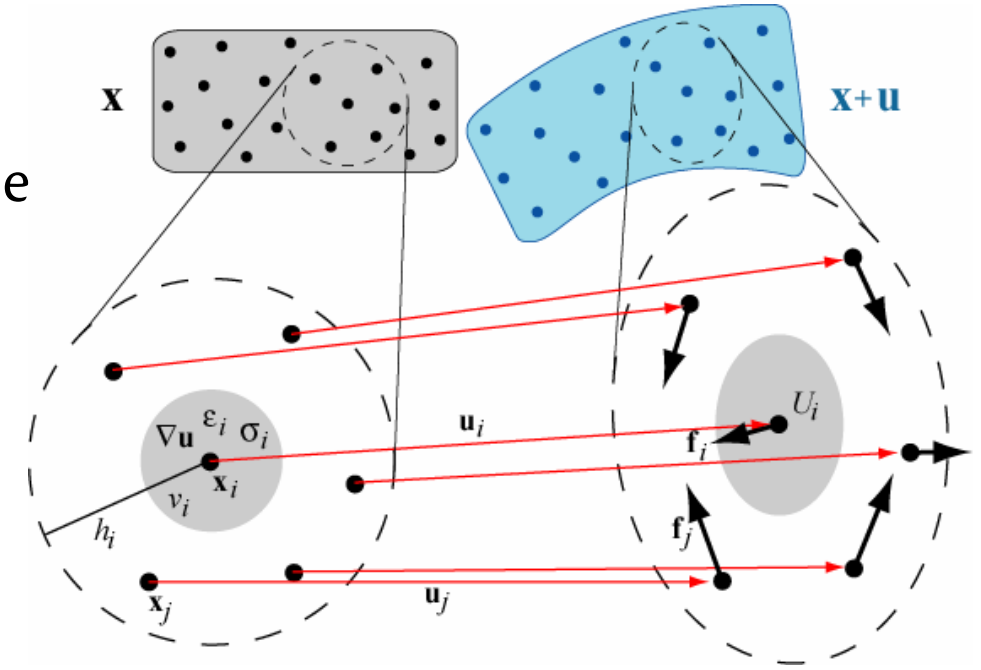
Physical Model (1/4)

- Coordinate Setup

  $\mathbf{x} = (x,y,z)^T$ Material

  $\mathbf{u} = (u,v,w)^T$ Displacement

- A point at **x** is located at **x+u** in the deformed model.



Physical Model (2/4)

- Physics Equations:
  - Green-Saint-Venant Strain Tensor:

$$\boldsymbol{\varepsilon}_{\text{Green}} = \nabla \mathbf{u}^T + \nabla \mathbf{u} + \nabla \mathbf{u}^T \nabla \mathbf{u}$$

  - Model assumes a Hookean material:

$$\boldsymbol{\sigma} = \mathbf{C}\boldsymbol{\varepsilon}$$

∇u = gradient of displacement vector field

σ = strain, ε = stress, C = material properties

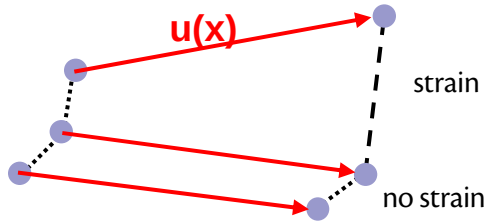
Physical Model (3/4)

- Strain Energy Density

$$U = \frac{1}{2}\sum_{i=1}^{3}\sum_{j=1}^{3}\varepsilon_{ij}\sigma_{ij}$$

- Finally, the force for each phyxel i and neighbors j:

$$\mathbf{f}_j = -\sigma\nabla u_i\varepsilon \qquad \mathbf{f}_i = -\sum_j \mathbf{f}_j$$



  - So, what we need to find the force is ∇u.

Physical Model (4/4)

- Volume Conservation
  - Achieved through an additional “volume conserving force”
  - Penalizes deviations in the determinant of the Jacobian from 1.

Discretization

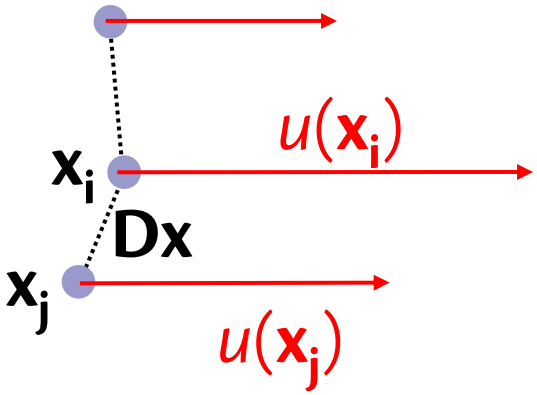
- Phyxel (PHYsical ELement)
  - Each Phyxel $phxl_i$ carries:
    - Location x_i
    - Density ρ_i
    - Deformation u_i
    - Velocity v_i
    - Strain ε_i
    - Stress σ_i

Discretization

- Calculating Gradient:
  - Moving Least Squares Approximation

- Moving Least Squares
  - Find a low degree polynomial which fits data in a least squares sense in the region around each point.
  - Replace data value at that point by the polynomial at that point.

Finding ∇u



$$u(\mathbf{x}_i + \Delta\mathbf{x}) = u(\mathbf{x}_i) + \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} \cdot \Delta\mathbf{x} + O(\|\Delta\mathbf{x}\|^2)$$

$$\nabla \mathbf{u} = \begin{bmatrix} u_x & u_y & u_z \\ v_x & v_y & v_z \\ w_x & w_y & w_z \end{bmatrix}$$

$$\tilde{u}_j = u_i + \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} \cdot (\mathbf{x}_j - \mathbf{x}_i)$$

Minimize error: $\sum_j (\tilde{u}_j - u_j)^2 \cdot w_{ij}$

MLS approximation of derivatives.

Simulation

- Process:
  1. Displacements (u_t)
  2. Derivatives (∇u_t)
  3. Strains (ε_t)
  4. Stresses (σ_t)
  5. Forces (f_t)
  6. Integration ($u_{t+\Delta t}$)

Updating Strains and Stresses

- Store Jacobian, Strain, Stress

$$\mathbf{J}_i \leftarrow \begin{bmatrix} \nabla u\,|_{x_i}^T \\ \nabla v\,|_{x_i}^T \\ \nabla w\,|_{x_i}^T \end{bmatrix} + \mathbf{I}, \quad \varepsilon_i \leftarrow (\mathbf{J}_i^T \mathbf{J}_i - \mathbf{I}), \quad \sigma_i \leftarrow (\mathbf{C}\varepsilon_i)$$

σ = strain, ε = stress, J = Jacobian

Calculating Forces

- Plug gradient into equations from earlier:

$$\mathbf{f}_j = -\sigma \nabla u_i \varepsilon \qquad \mathbf{f}_i = -\sum_j \mathbf{f}_j$$

Time Integration

- Implicit Integration is used to apply forces to the model.
- The paper also mentions Leap-Frog integration.

Plastic Strain Model

- Store a plastic strain tensor.

- Strain equals:

$$\tilde{\varepsilon}_i = \varepsilon_i - \varepsilon_i^p$$

Simulation

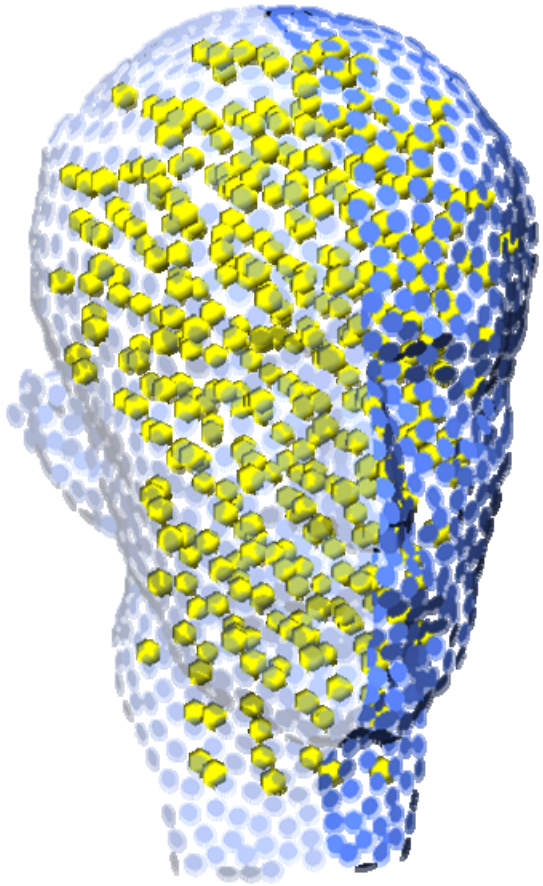
- After all values are calculated, apply!

- For each Phyxel:
  - $\varepsilon_i^p \leftarrow \varepsilon_i^p - \varepsilon_i$
  - $x_i \leftarrow x_i + u_i$
  - $u_i \leftarrow 0$

Surfaces

- How do we draw the surface over the phyxels?

  Surfels (SURFace ELement), are used as points in an implicit surface rendering.



How do we move the Surfels?

- Displacement Approach
- Multi-Representation Approach
  - Implicit Representation
  - Detail Representation
  - Contact Representation

Displacement Approach

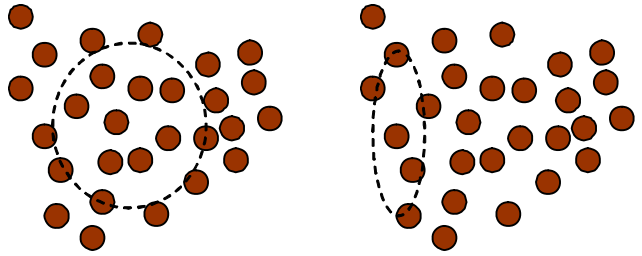
- Use displacement vector field from the nearby phyxels to move the surfels, Gaussian-weighted by distance.
- Split and merge surfels as necessary.

Multi-Representational Approach (1/3)

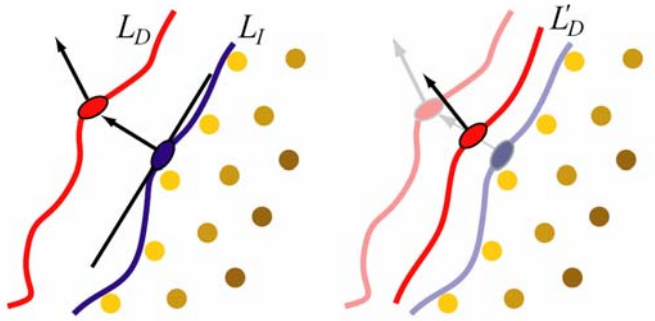
- Implicit Representation
  - Find iso-surface L_I from phyxels, and project surfels to this surface.
  - Only good for blobby surfaces.

Multi-Representational Approach (2/3)

- Detail Representation
  1. Categorize inside phyxels and outside phyxels using PCA.



  2. Project surfels onto outer phyxel boundaries, taking into account displacements to represent detail surface.



Multi-Representational Approach (3/3)

- Contact Representation
  - Object melting into mold will be projected onto MLS surface of mold as it contacts, setting normals equal to mold's MLS surface.
  - Perfectly takes shape of mold.

Resampling

- To ensure optimal surfel density:
  - Count neighboring surfels
  - If surfel count exceeds max threshold, delete surfel.
  - If surfel count is below min threshold, split surfel.

Results

- Simulations render offline at 5-8s/frame, with realtime simulations possible as well.

- <videos>

Limitations

- Assume Hookean Material
- MLS requires each phyxel to have at least 3 neighbors in non degenerate locations, so no 2D layers or 1D strings.
- Fractures aren't handled too well.