

IFT 6112

11 – MESH DEFORMATION

<http://www-labs.iro.umontreal.ca/~bmpix/teaching/6112/2018/>

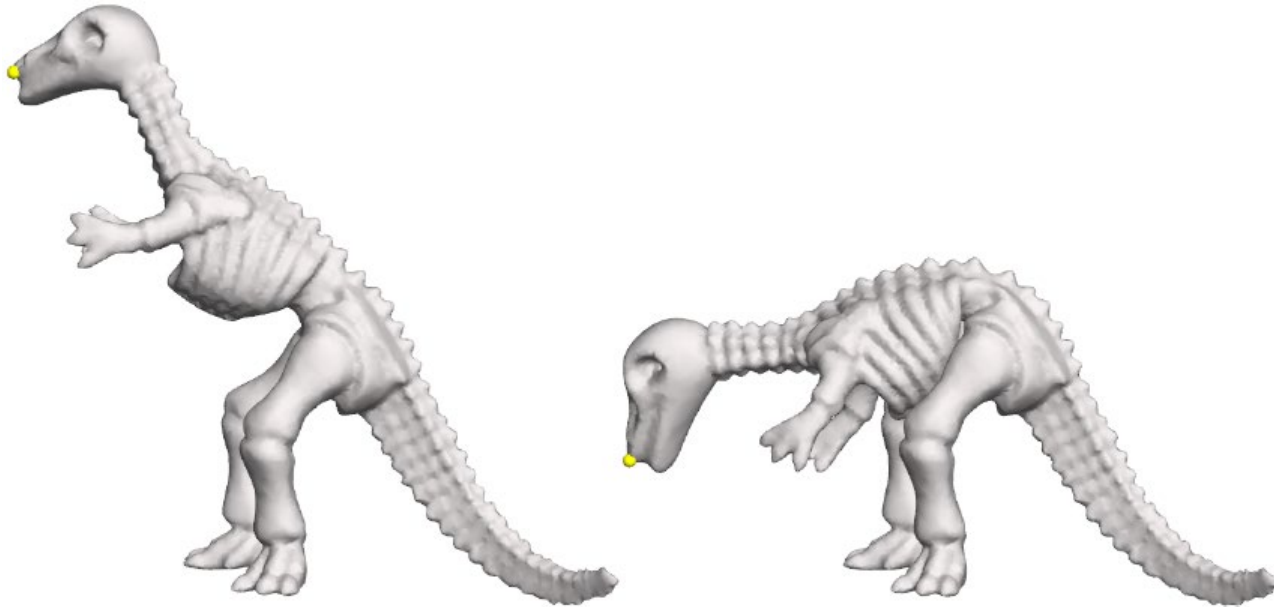


Image from 'As-Rigid-As-Possible Shape Modeling' by Sorkine & Alexa, 2007

Mikhail Bessmeltsev

Administrative

- Paper title due tonight!
- A3 due on Nov, 8th
- Paper presentations on Nov, 20th
 - Make sure you understand the inner workings
 - Ask tough questions!
- Face-to-face grading next time

What for?

- Animation!
- Mesh editing
- Image warping (2D)

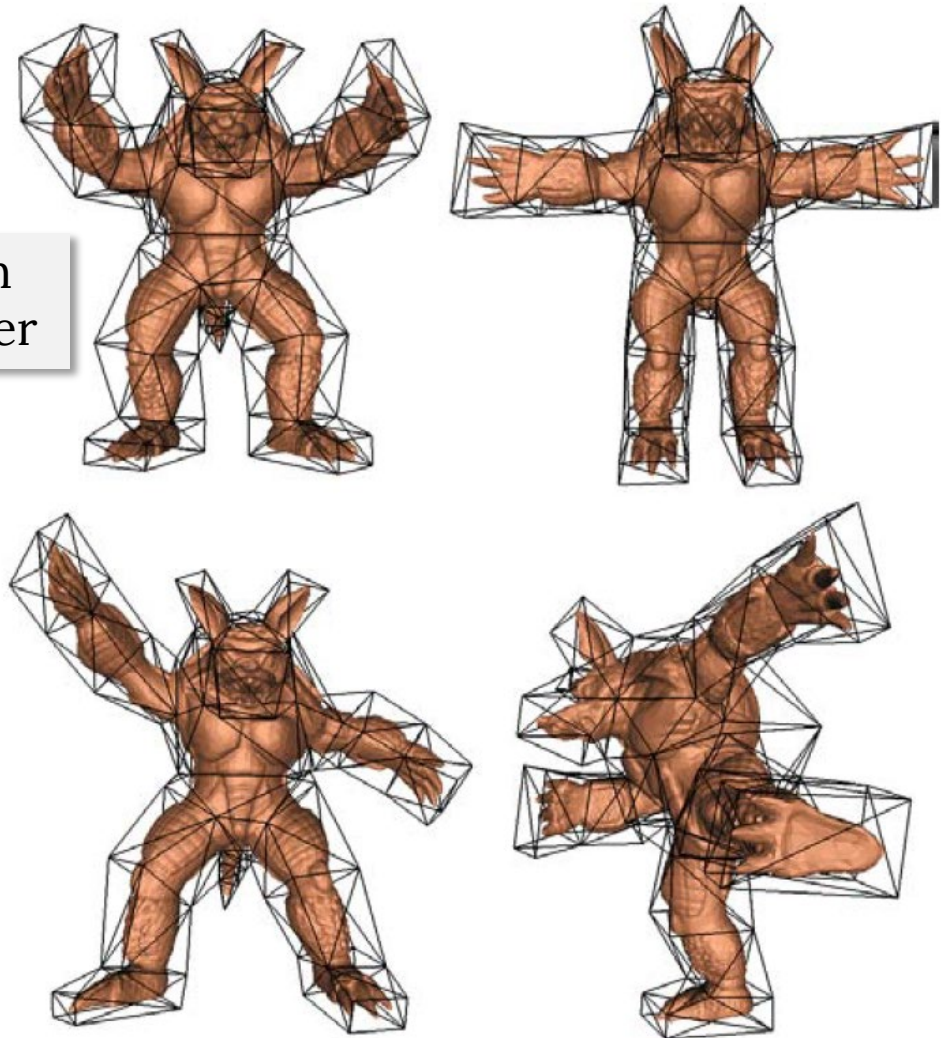


This, and many other images in this presentation are from 'Polygon Mesh Processing' textbook by Botsch et al. or their website

Deformation: user interface

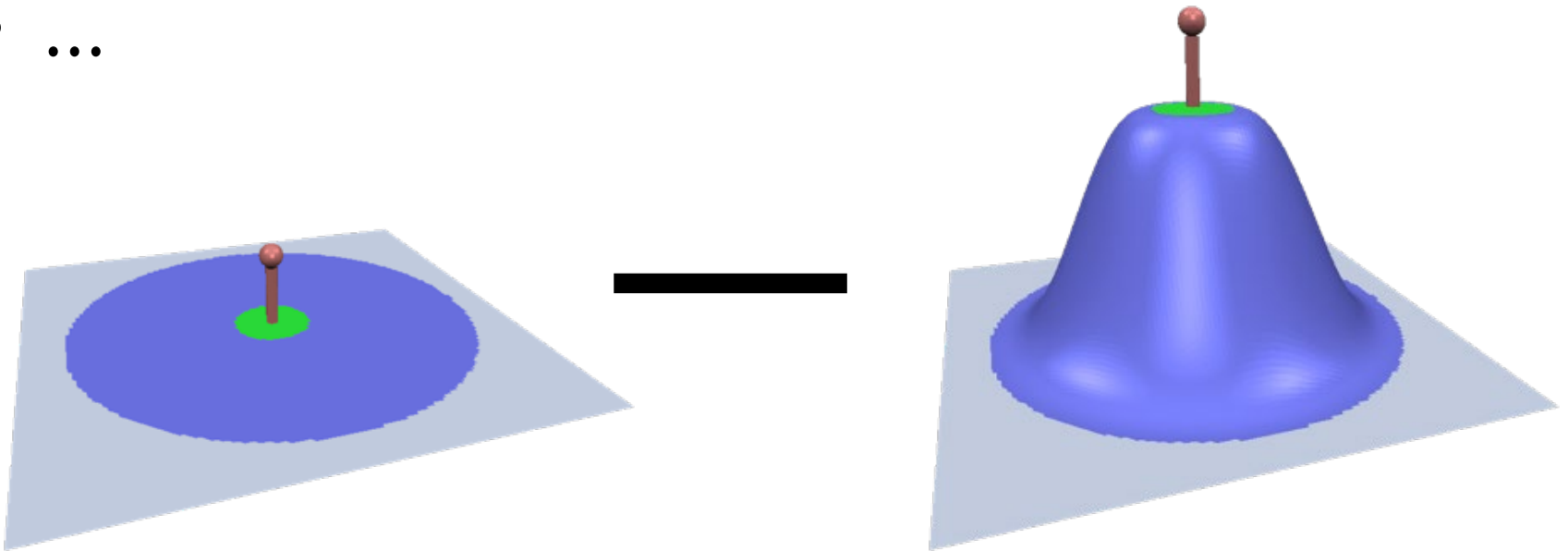
- Handles
- Cages
- Skeletons
- ...

More on
those later



Deformation: user interface

- **Handles**
- Cages
- Skeletons
- ...



Modeling

Paint three surface areas:

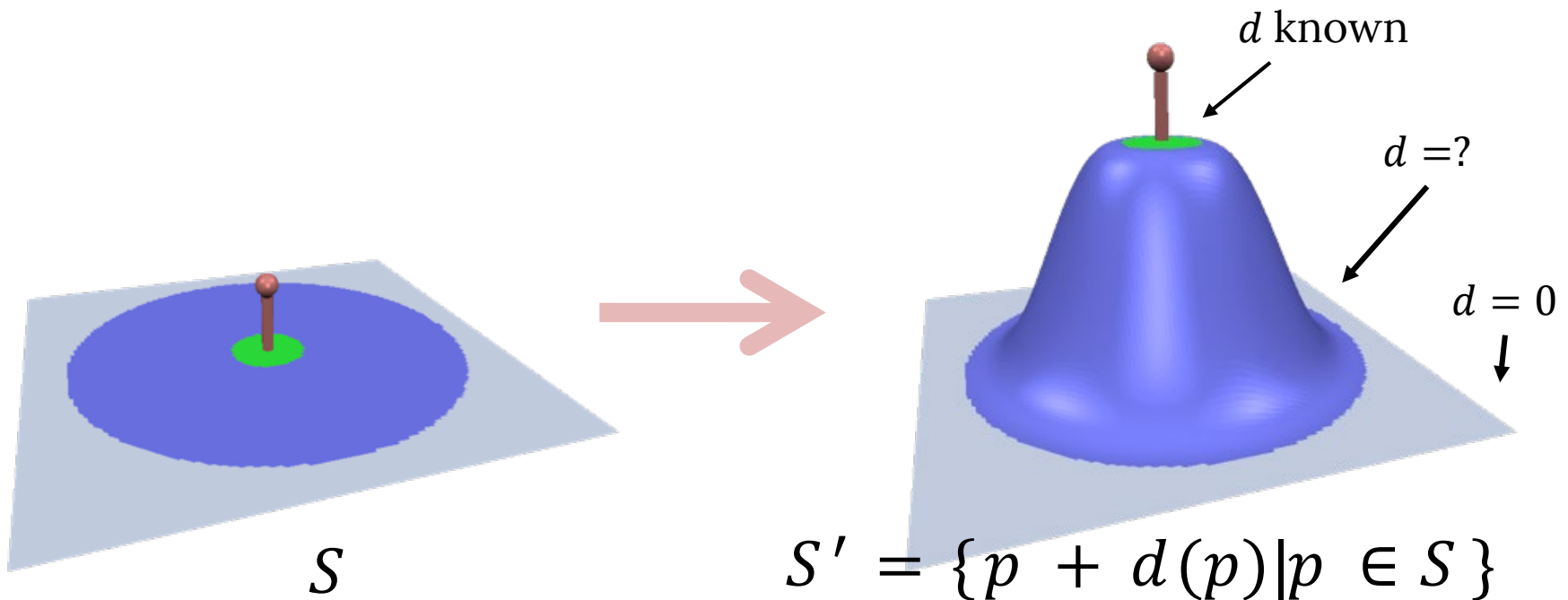
- Constrained
- Smooth falloff
- Fixed



Formulation

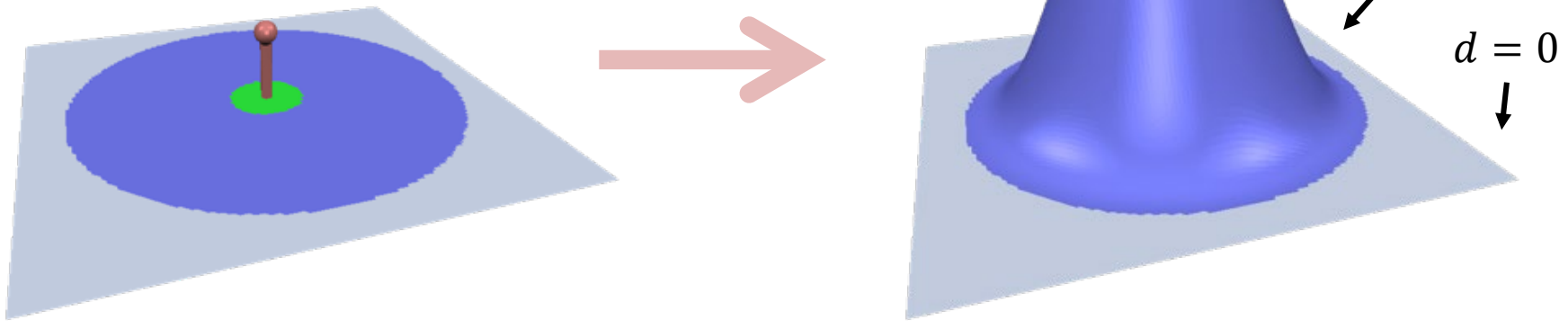
Find displacement vector field d

- Smooth
- Satisfies constraints



Simplest idea

- $d = s(p) \cdot D$
- $s(p)$ is a smooth function:
 - 1 on green vertices
 - 0 on grey ones



How to find $s(p)$?

- Something inversely proportional to geodesic distance
- Or our favorite:

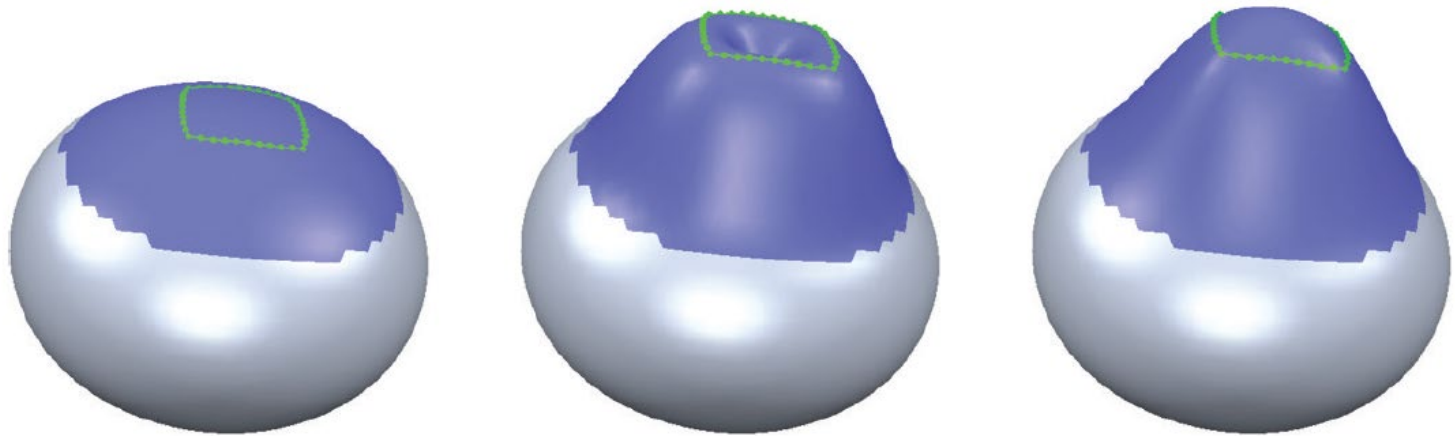
$$\Delta s(\mathbf{p}_i) = 0, \quad \mathbf{p}_i \in \mathcal{R},$$

$$s(\mathbf{p}_i) = 1, \quad \mathbf{p}_i \in \mathcal{H},$$

$$s(\mathbf{p}_i) = 0, \quad \mathbf{p}_i \in \mathcal{F}.$$

Solved?

Reality vs Expectation



Physically-Based

Find a deformation that preserves both fundamental forms

Express the fundamental forms of S' via vector field d

$$\int_{\Omega} k_s \left\| \mathbf{I} - \mathbf{I}' \right\|_F^2 + k_b \left\| \mathbf{II} - \mathbf{II}' \right\|_F^2 \, dudv$$

Expensive to optimize!

stretching bending

Physically-Based

Find a deformation that preserves both
fundamental forms

Linearize—Express the fundamental forms of S' via vector
field d

$$\int_{\Omega} k_s \left(\|d_u\|^2 + \|d_v\|^2 \right) + k_b \left(\|d_{uu}\|^2 + 2 \|d_{uv}\|^2 + \|d_{vv}\|^2 \right) dudv$$

stretching bending

Physically-Based

$$\int_{\Omega} k_s \left(\|\mathbf{d}_u\|^2 + \|\mathbf{d}_v\|^2 \right) + k_b \left(\|\mathbf{d}_{uu}\|^2 + 2 \|\mathbf{d}_{uv}\|^2 + \|\mathbf{d}_{vv}\|^2 \right) dudv$$

stretching bending

Gateaux derivative =>

$$-k_s \Delta d + k_b \Delta^2 d = 0$$

Physically-Based

$$\int_{\Omega} k_s \left(\|\mathbf{d}_u\|^2 + \|\mathbf{d}_v\|^2 \right) + k_b \left(\|\mathbf{d}_{uu}\|^2 + 2 \|\mathbf{d}_{uv}\|^2 + \|\mathbf{d}_{vv}\|^2 \right) dudv$$

stretching bending

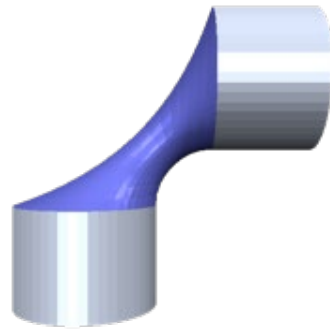
Gateaux derivative =>

$$-k_s \Delta d + k_b \Delta^2 d = 0$$

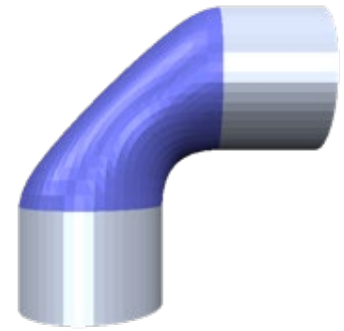
$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

Bi-Laplacian

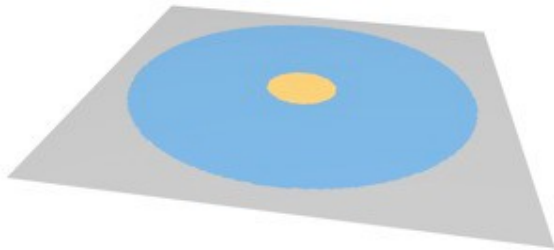
Deformation Energies



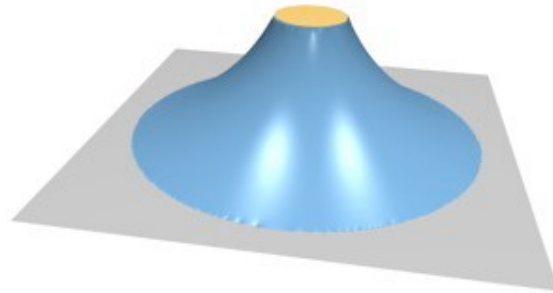
$$\Delta p = 0$$



$$\Delta^2 p = 0$$

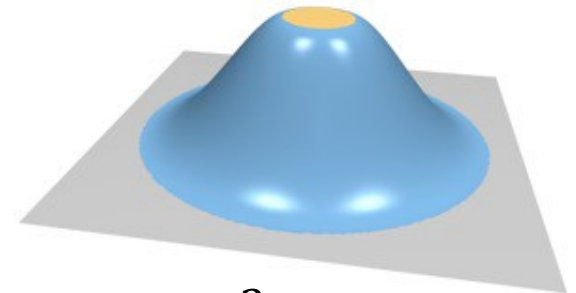


Initial state



$$\Delta d = 0$$

(Membrane)

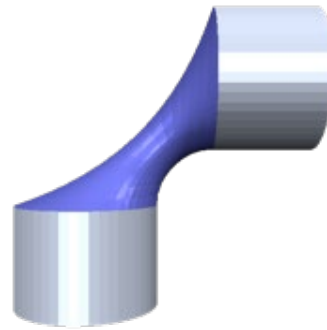


$$\Delta^2 d = 0$$

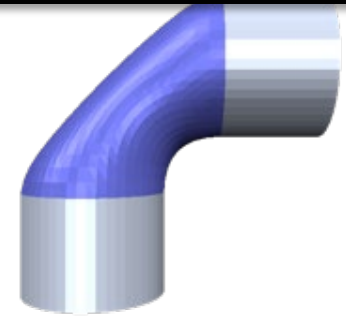
(Thin plate)

Deformation Energy

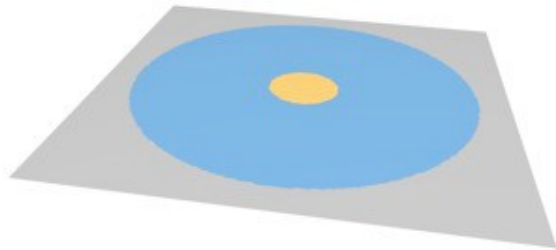
Higher order =>
more boundary
conditions



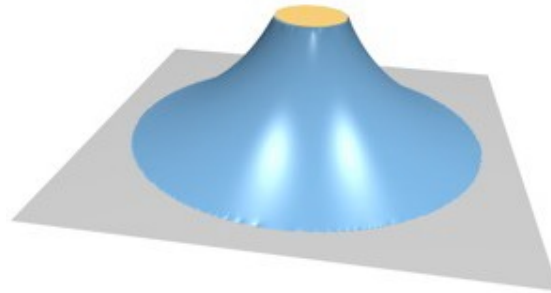
$$\Delta p = 0$$



$$\Delta^2 p = 0$$

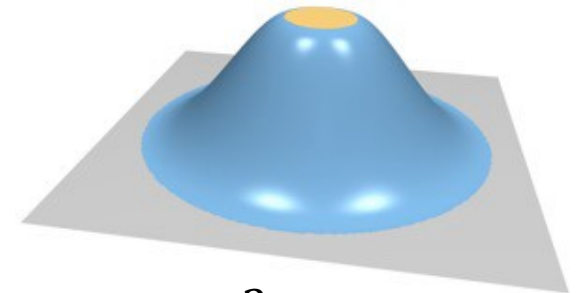


Initial state



$$\Delta d = 0$$

(Membrane)

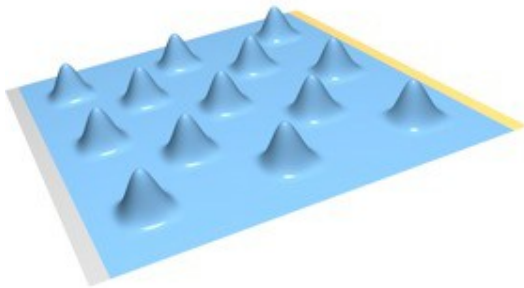


$$\Delta^2 d = 0$$

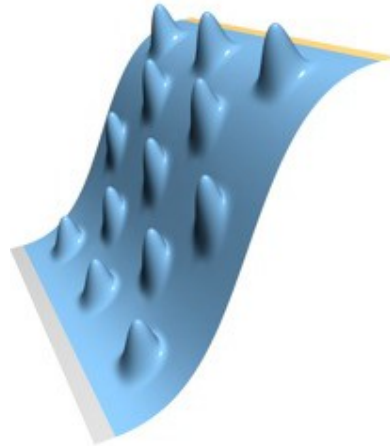
(Thin plate)

Solved?

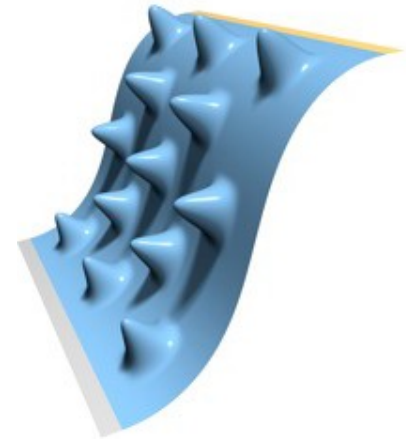
- Very fast
 - One linear solve!
- Physically-based
- Linearization => **lose details**



Original



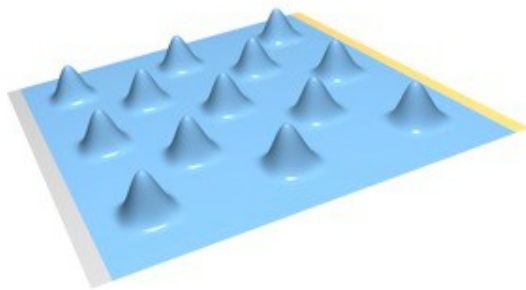
Linear
deformation



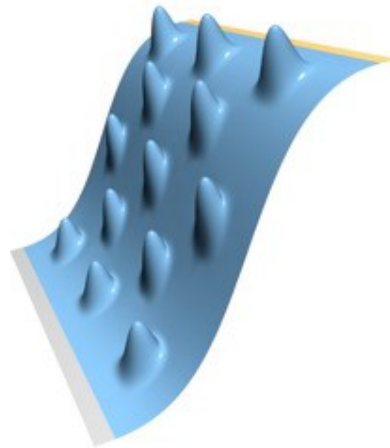
Non-linear
deformation

Issue

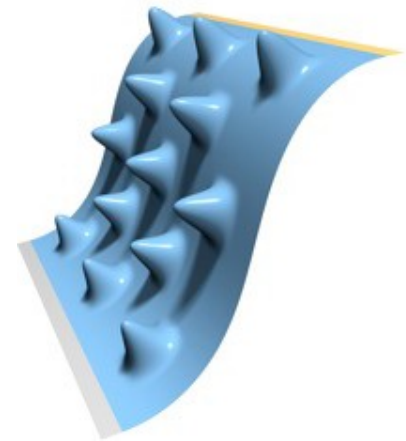
- We need to rotate details
 - Local rotation is nonlinear!
- Can we still survive with linear solves?



Original

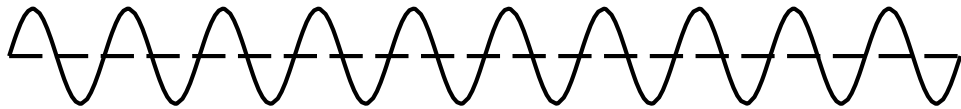


Linear
deformation



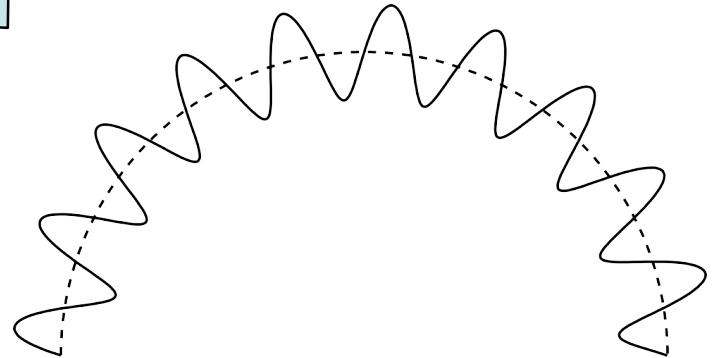
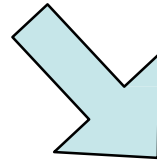
Non-linear
deformation

Multiresolution Editing



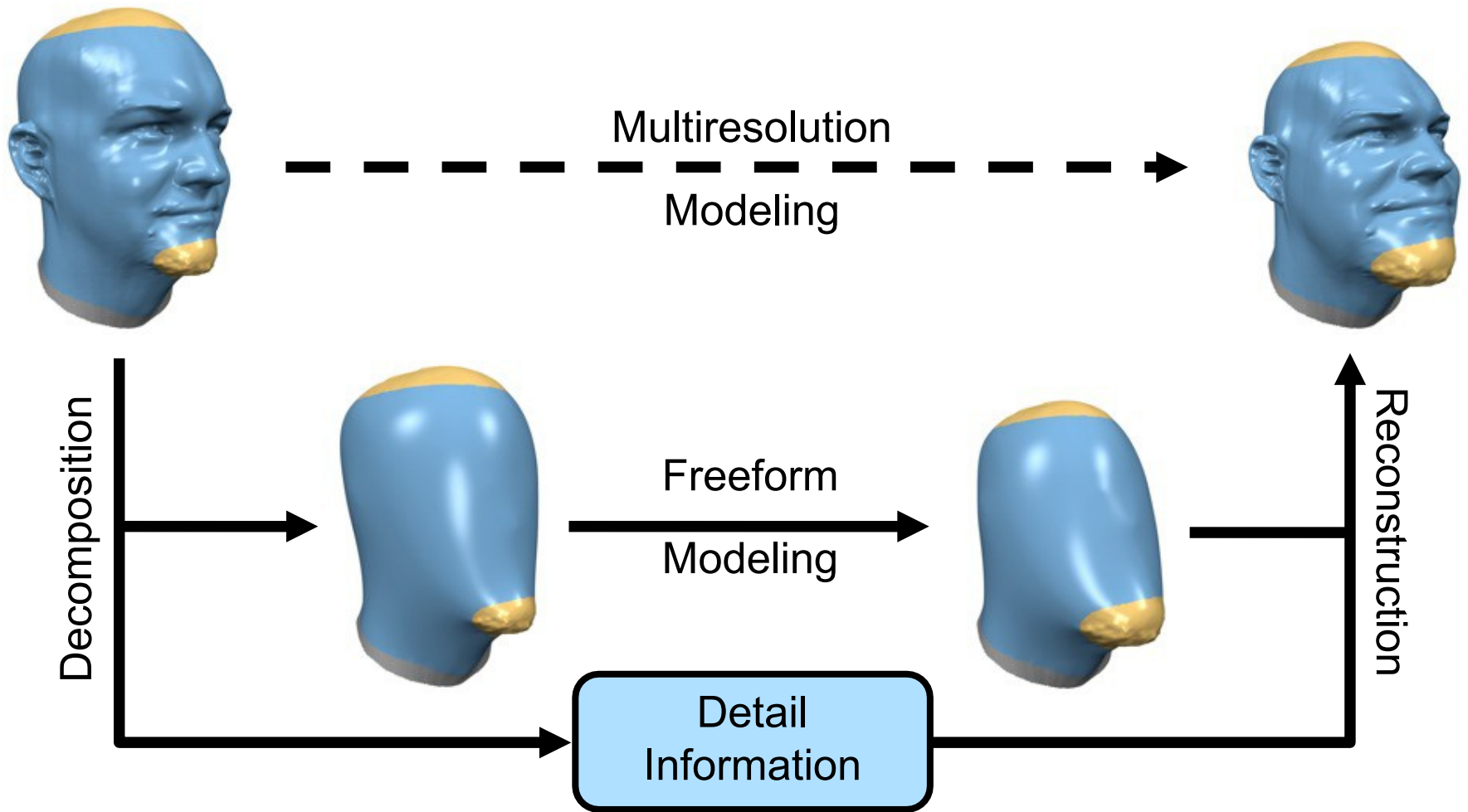
Frequency decomposition

Change low
frequencies



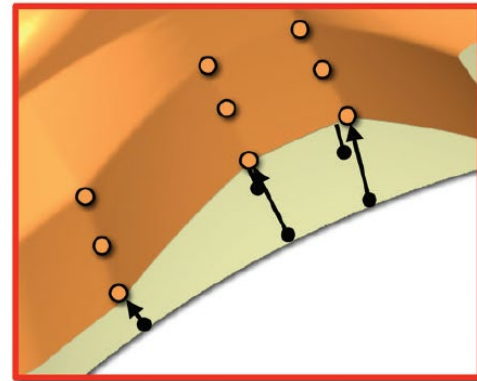
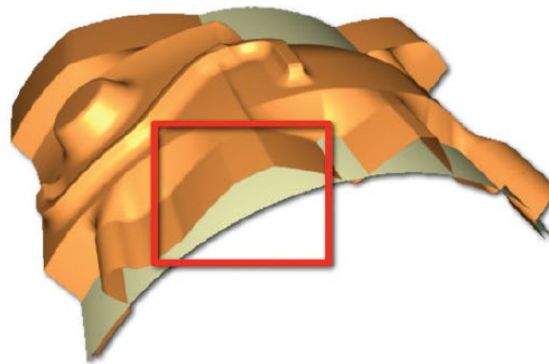
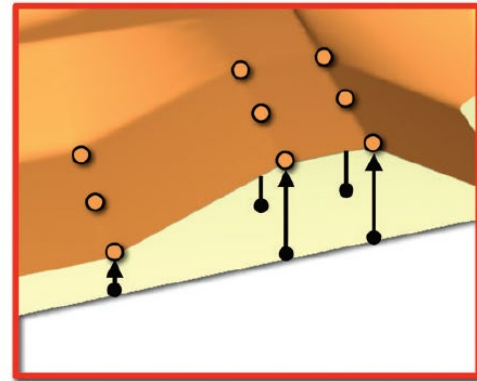
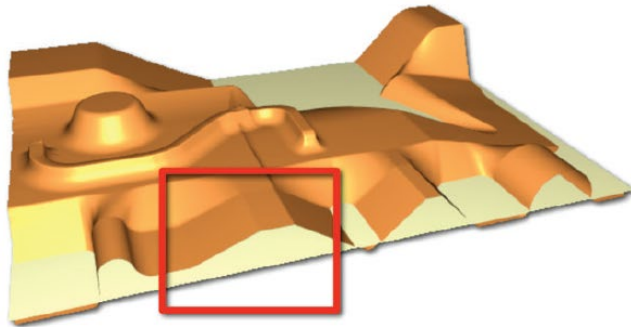
Add high frequency details,
stored in local frames

Multiresolution Editing

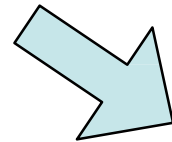
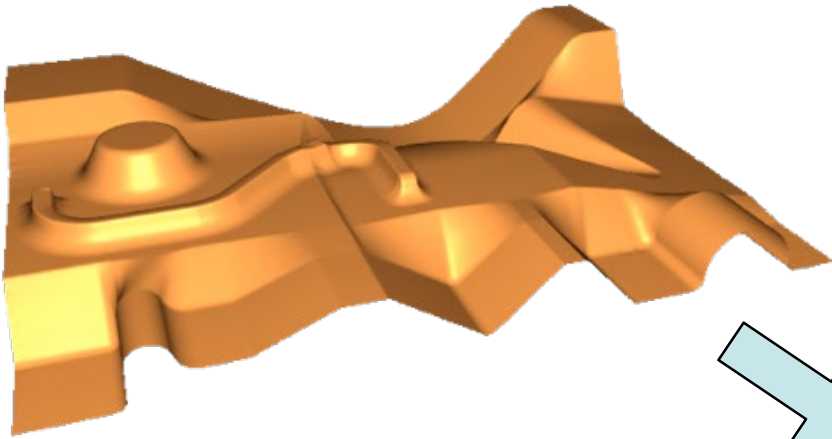


How to represent details?

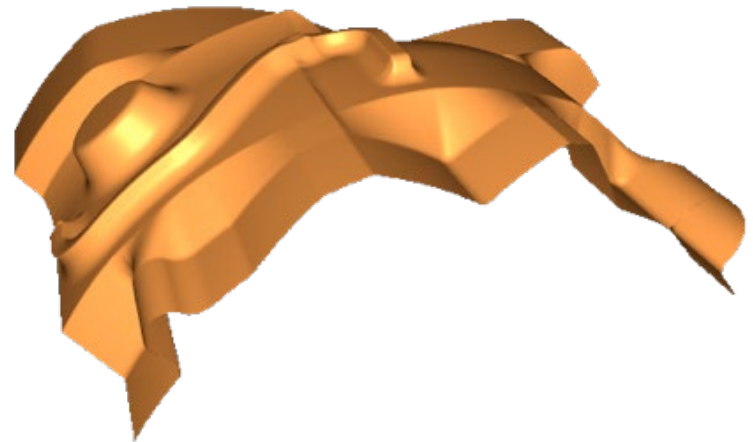
- For example, normal displacements



Result



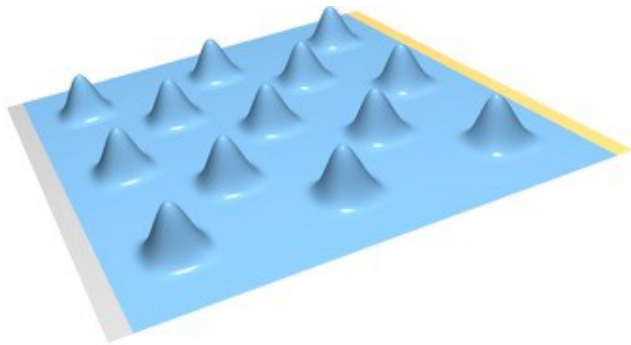
Global deformation
with intuitive detail
preservation



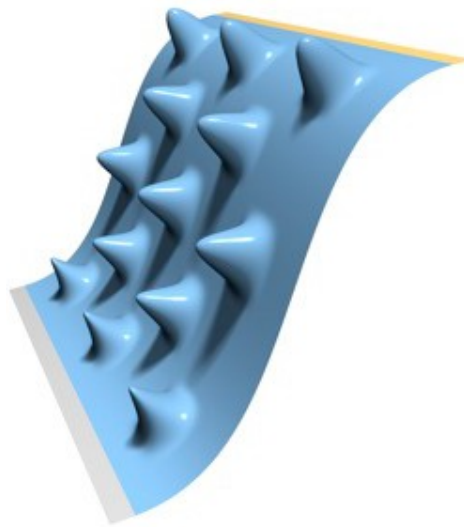
Limitations

Neighboring displacements are not coupled

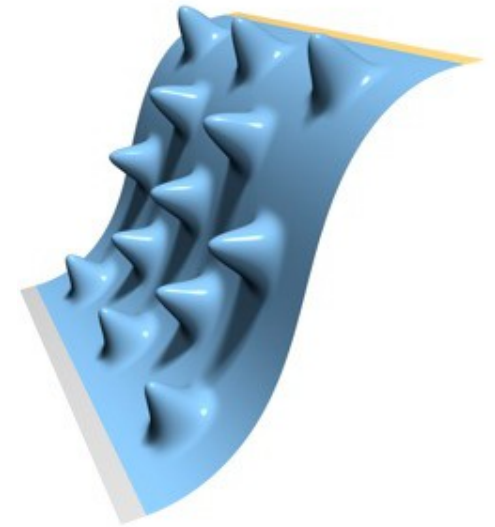
- Surface bending changes their angle
- Leads to volume changes or self-intersections



Original



Normal Displ.

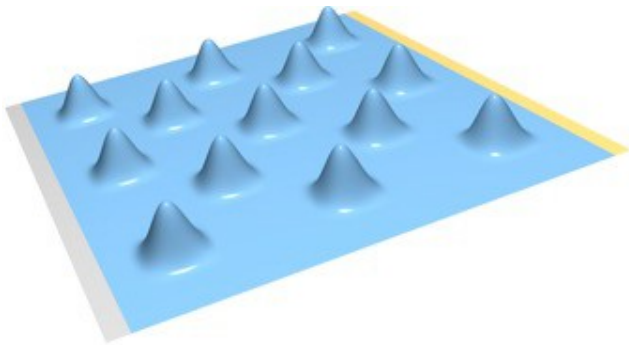


Nonlinear

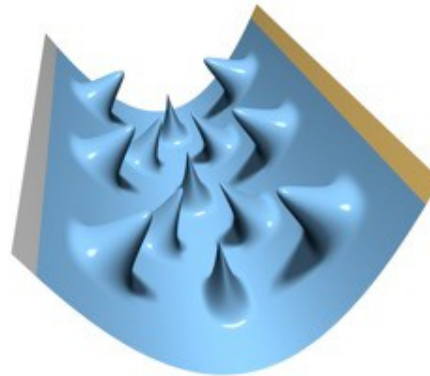
Limitations

Neighboring displacements are not coupled

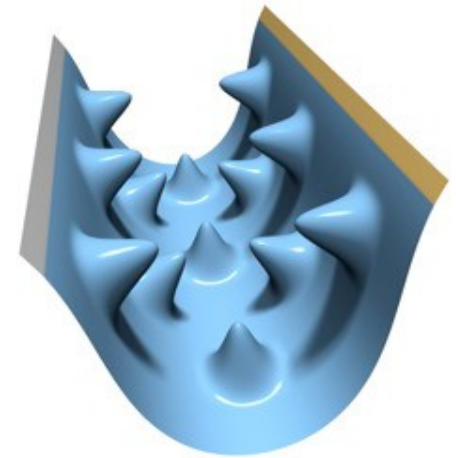
- Surface bending changes their angle
- Leads to volume changes or self-intersections



Original



Normal Displ.



Nonlinear

Laplacian Mesh Editing

- Avoid explicit multi-scale decomposition
- Express shape in *differential coordinates*
- Transform those, then reconstruct the new shape

Long time ago:

Mean Value Property

$$L_{vw} = A - D = \begin{cases} 1 & \text{if } v \sim w \\ -\text{degree}(v) & \text{if } v = w \\ 0 & \text{otherwise} \end{cases}$$

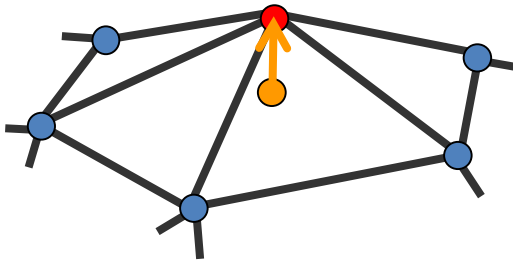
$$(Lx)_v = 0$$



Value at v is average of neighboring values

Laplacian Mesh Editing

(Graph Laplacian):

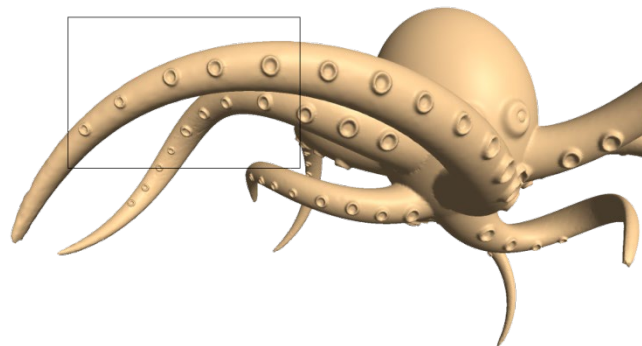


$$\delta_i = \mathbf{v}_i - \frac{1}{d_i} \sum_{j \in N(i)} \mathbf{v}_j$$

$$\delta_i = \sum_{j \in N(i)} \frac{1}{d_i} (\mathbf{v}_i - \mathbf{v}_j)$$

Local coordinates!

$$Lv = \delta$$



Laplacian Mesh Editing

- Do all transformations on δ
- Find a surface whose Laplacian coordinates are as close as possible to δ

$$\int_{\mathcal{S}} \|\Delta \mathbf{p}' - \delta'\|^2 d\mathcal{S} \rightarrow \min$$

- Gateaux derivative \Rightarrow

$$\Delta^2 \mathbf{p}' = \Delta \delta'$$

Before:

Physically-Based

$$\int_{\Omega} k_s \left(\|\mathbf{d}_u\|^2 + \|\mathbf{d}_v\|^2 \right) + k_b \left(\|\mathbf{d}_{uu}\|^2 + 2\|\mathbf{d}_{uv}\|^2 + \|\mathbf{d}_{vv}\|^2 \right) dudv$$

stretching bending

Gateaux derivative =>

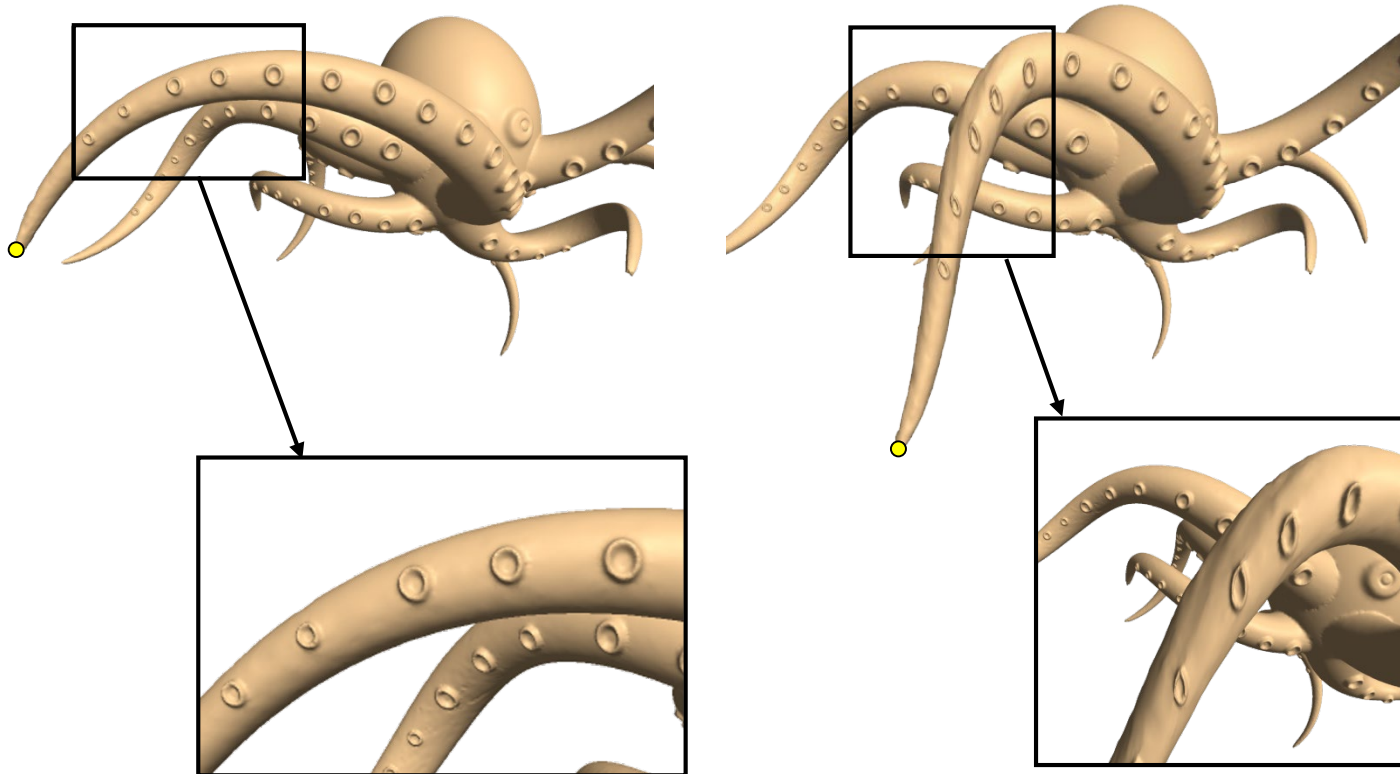
$$-k_s \Delta d + k_b \Delta^2 d = 0$$

(almost) the same
equation?

Bi-Laplacian

Issue

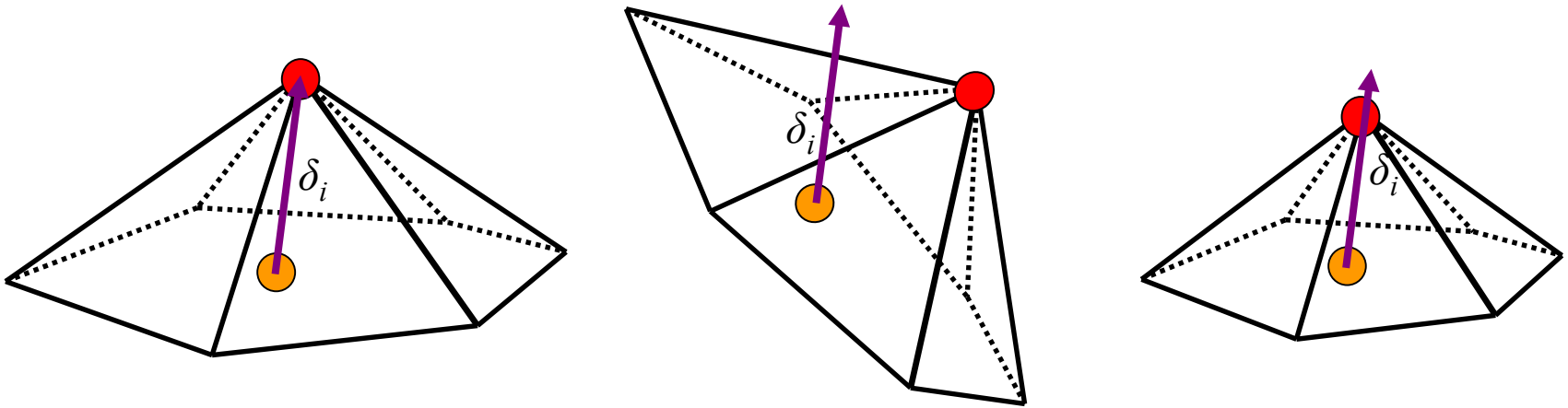
Only translation invariant



Translating a handle induces local rotations!

Laplacian Coordinates

- Translation invariant
- Not rotation/scale invariant



$$\delta_i = L(\mathbf{v}_i) = L(\mathbf{v}_i + \mathbf{t}); \forall \mathbf{t} \in \mathbb{R}^3$$

Solutions

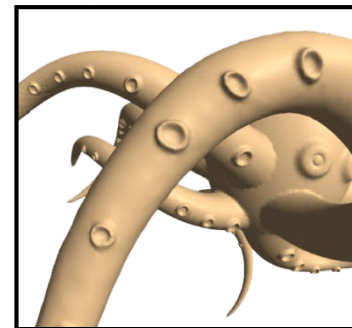
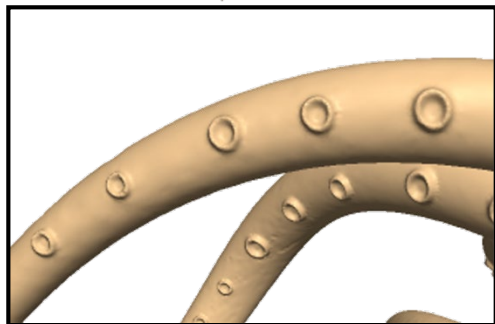
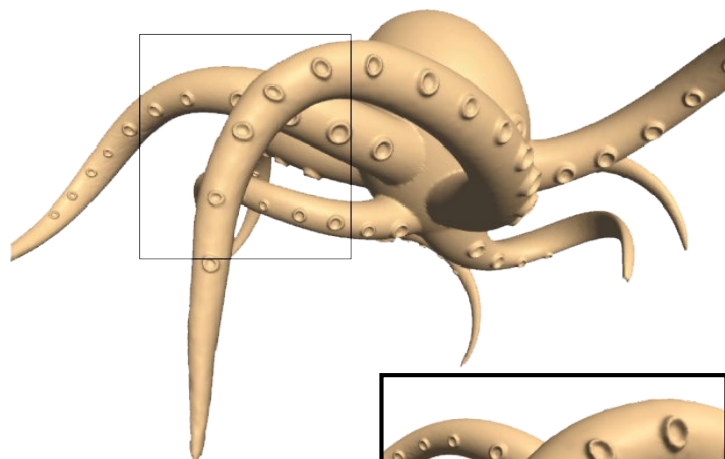
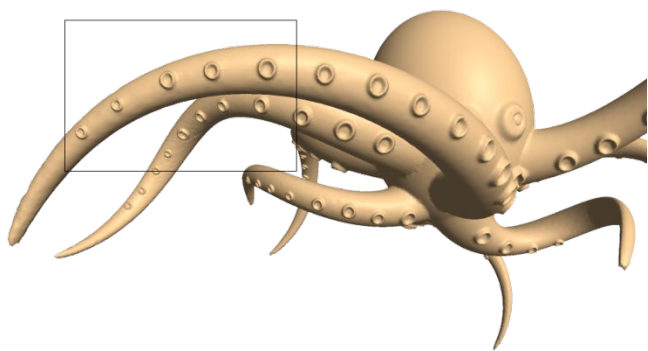
- Explicit transformation of the differential coordinates prior to surface reconstruction
 - Estimate rotations
- Add rotations into framework

$$E(\mathbf{V}') = \sum_{i=1}^n \|R_i \delta_i - L(\mathbf{v}'_i)\|^2 + \sum_{i \in C} \|\mathbf{v}'_i - \mathbf{u}_i\|^2$$

- Interleave rotate/position (local/global) iterations

Solutions

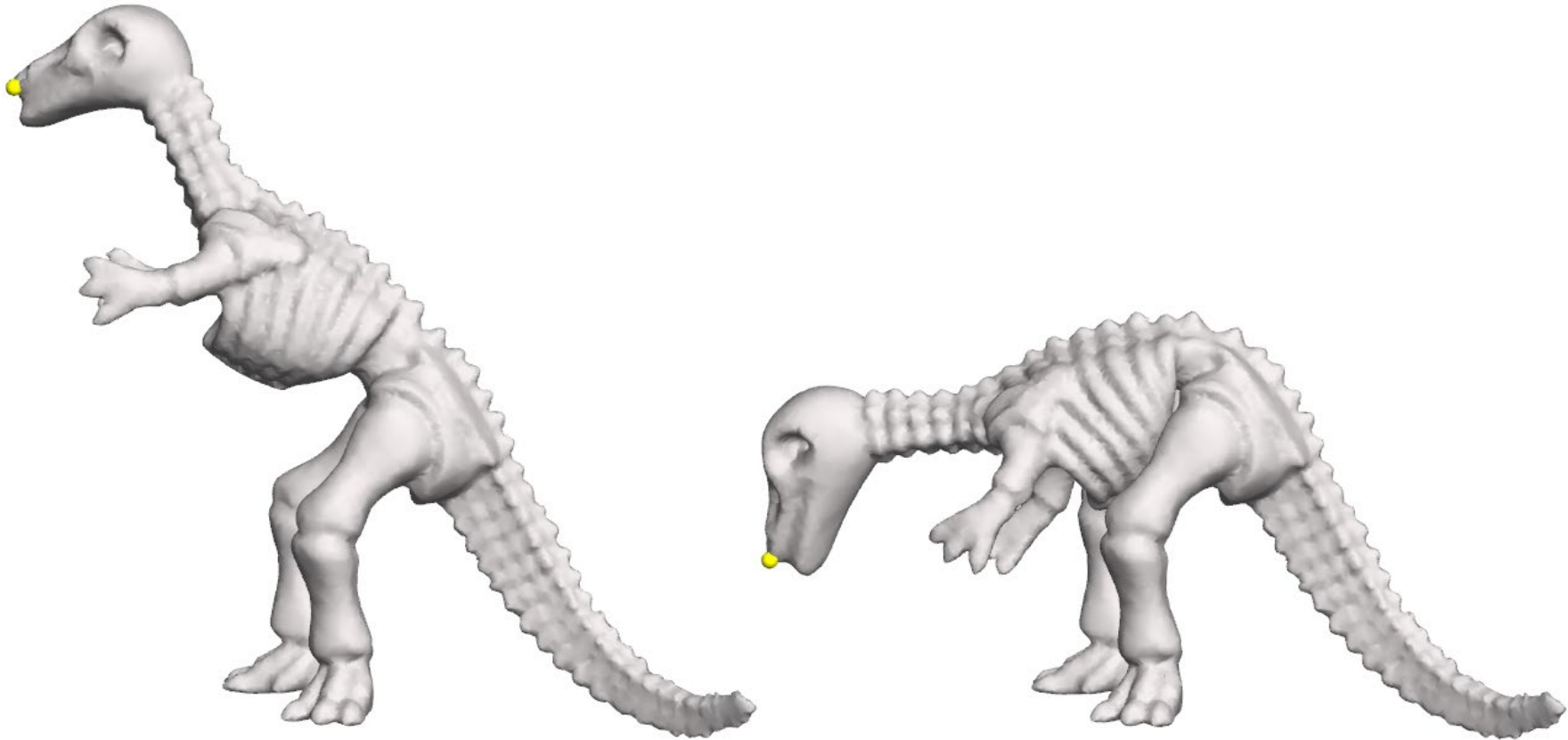
- [Lipman et al. 2004] estimate rotation of local frames
 - Reconstruct the surface with the original Laplacians
 - Estimate the normals of the underlying smooth surface
 - Rotate the Laplacians and reconstruct again



Rotation Propagation

- For each vertex in ROI define rotation (new normal) as combination of boundary & anchor rotations
 - Use “distance” from boundary/anchors as weight
 - distance = geodesic, harmonic (Laplacian),...
 - Rotation algebra
 - quaternions
 - simple vector algebra with re-normalization...
 - ...

As-Rigid-As-Possible Surface Modelling





As-rigid-as-possible (ARAP)



ARAP in a nutshell...

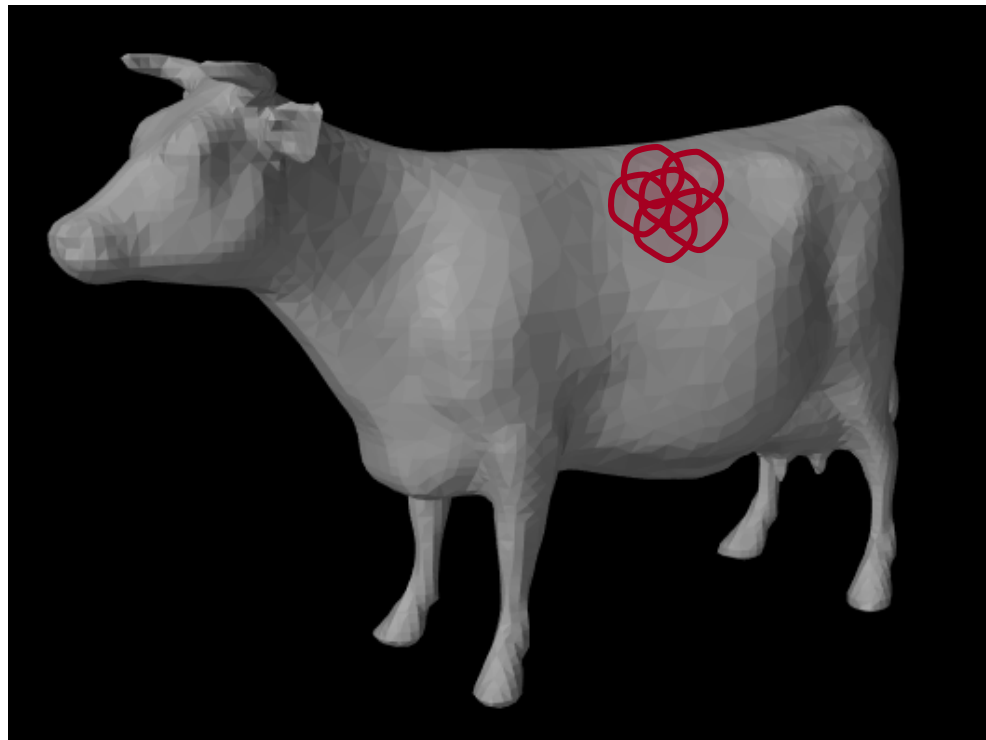
- Decompose surface into small overlapping “cells”
- Define a metric for measuring local rigidity
- Create a global energy function representing global “rigidity”
- Minimize “shell energy” function to determine “optimal” vertex locations



Cell Construction

Desired Properties:

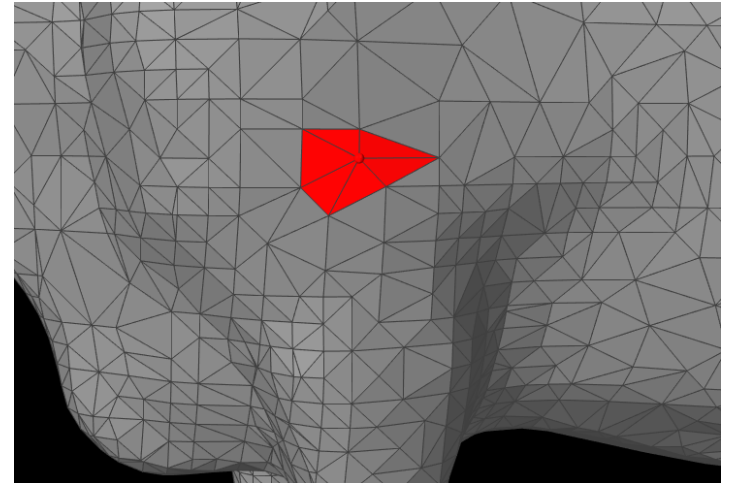
- Characterize local shape
- Used to enforce local rigidity constraints
- Overlapping, to prevent shearing/stretching at cell boundaries



Cell Construction

Simplest logical choice for cells?

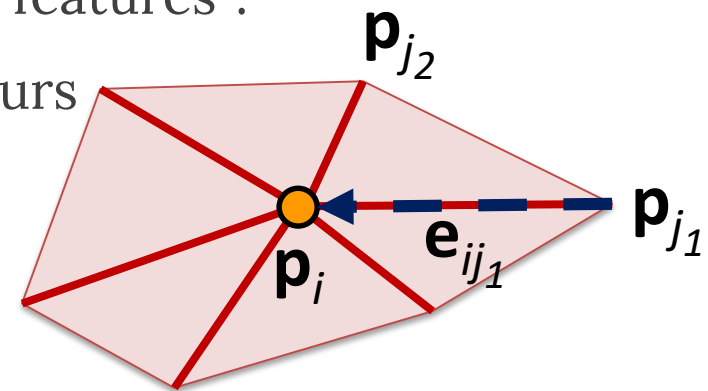
- Vertex Umbrella
 - Covers entire surface
 - One cell per vertex
 - All triangles exist in 3 cells



Within cell, define translation-invariant “features”:

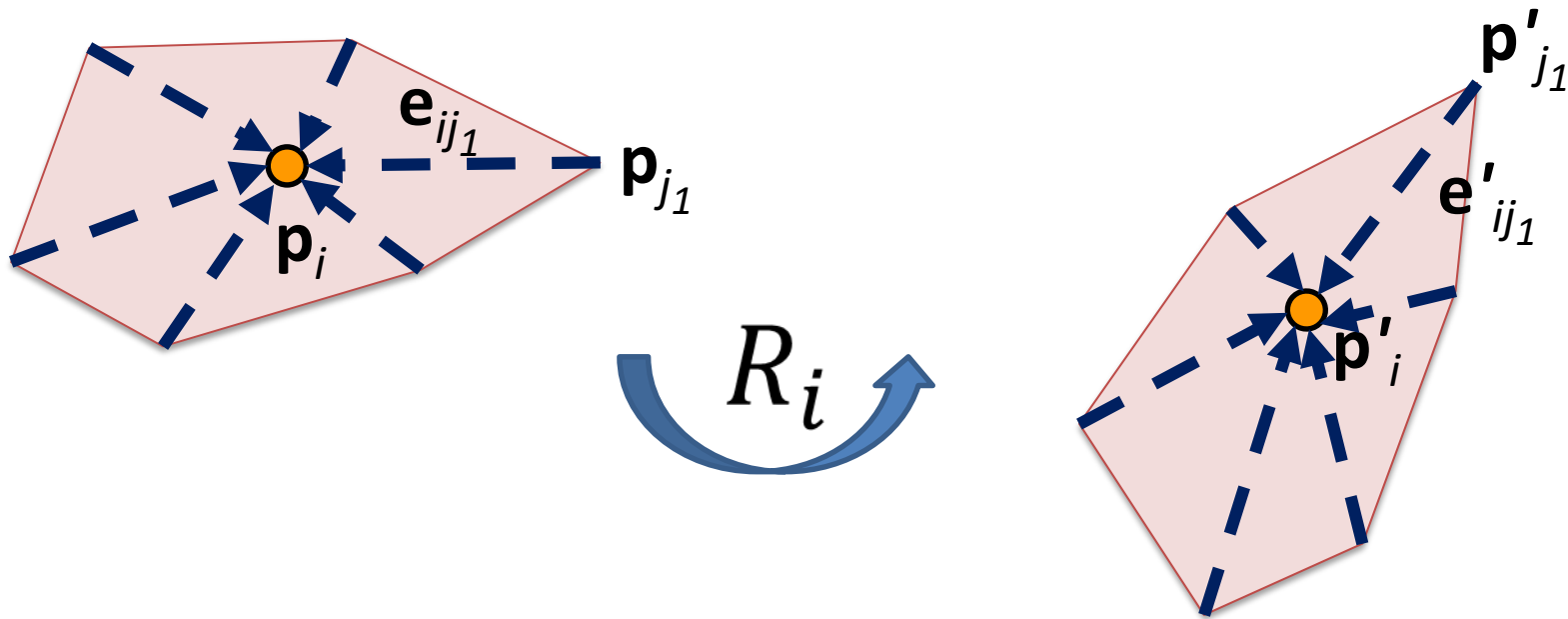
- Vectors from central vertex to neighbours

$$\mathbf{e}_{ij} = \mathbf{p}_i - \mathbf{p}_j$$



Local “Rigidity”

If cell i moved as a rigid unit, we **could** write: $\mathbf{e}'_{ij} = R_i \mathbf{e}_{ij} \quad \forall j \in N(i)$

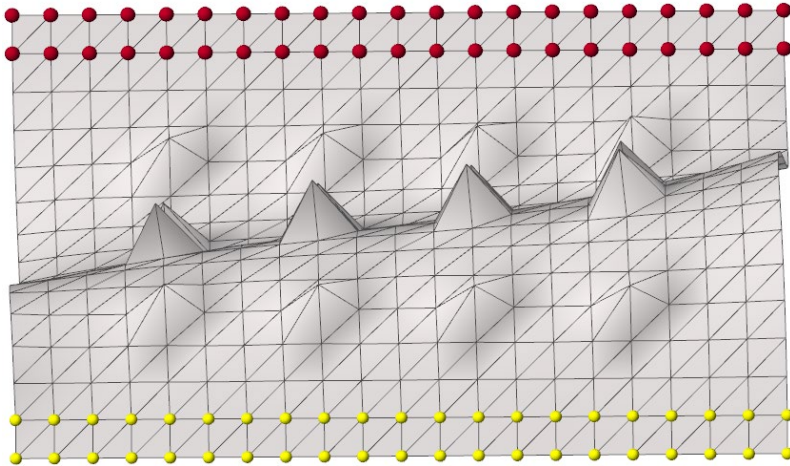


- What if they didn't move as a rigid unit?

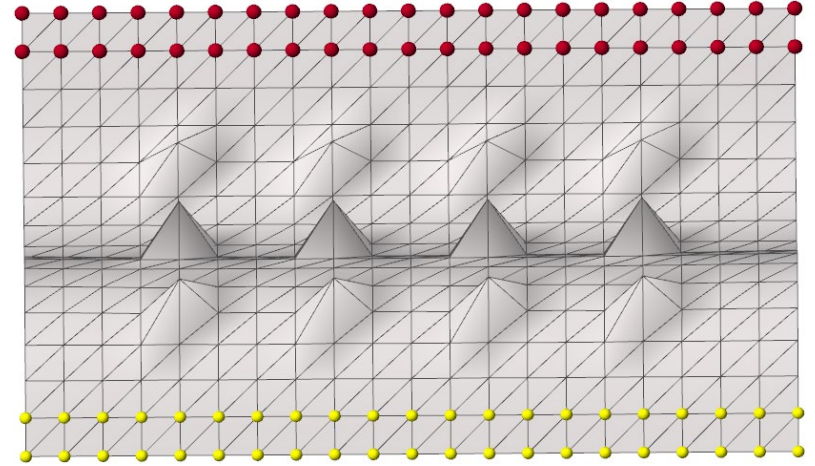
$$\text{Rigid Error}^2 (C_i, C'_i) = \sum_{j \in N(i)} \|\mathbf{e}'_{ij} - R_i \mathbf{e}_{ij}\|^2$$

Edge Weights

Should all edge vectors be weighted equally?



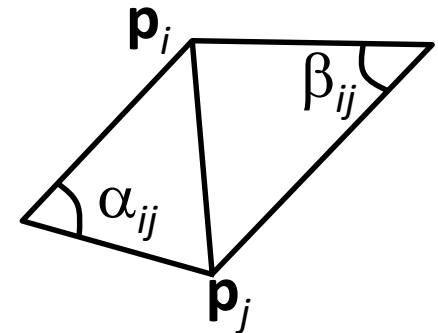
Uniform Weights



Cotangent Weights

Cotangent weights:

$$w_{ij} = \frac{1}{2} (\cot \alpha_{ij} + \cot \beta_{ij})$$



Global Rigidity

Local Rigidity:

$$\text{Rigid Error}^2 (C_i, C'_i) = \sum_{j \in N(i)} w_{ij} \|(\mathbf{p}'_i - \mathbf{p}'_j) - R_i(\mathbf{p}_i - \mathbf{p}_j)\|^2$$

- Given $\{\mathbf{p}_i\}$ and $\{\mathbf{p}'_i\}$, what are the optimal rigid transforms $\{R_i\}$?
 - Least squares + SVD!!!

Global Rigidity:

- All local cells as-rigid-as-possible \rightarrow global shape is as-rigid-as-possible
- Global energy function:

$$\text{Energy} = \sum_i \sum_{j \in N(i)} w_{ij} \|(\mathbf{p}'_i - \mathbf{p}'_j) - R_i(\mathbf{p}_i - \mathbf{p}_j)\|^2$$

Mesh Deformation

- Set the locations of some points (constraints): $\mathbf{p}'_i = \mathbf{u}_i, i \in C$
- Determine locations $\{\mathbf{p}'_i\}$ for “free” points by minimizing global energy

$$\text{Energy} = \sum_i \sum_{j \in N(i)} w_{ij} \|(\mathbf{p}'_i - \mathbf{p}'_j) - R_i(\mathbf{p}_i - \mathbf{p}_j)\|^2 + \underbrace{\mu \sum_{i \in C} w_{ij} \|\mathbf{p}'_i - \mathbf{u}_i\|^2}_{\text{soft constraints}}$$

Caveats:

- $\{\mathbf{p}'_i\}$ and $\{R_i\}$ are unknown
- Non-linear optimization problem

Mesh Deformation

Solution:

- Start with initial guess of $\{\mathbf{p}'_i\}$, solve for $\{R_i\}$
 - Compute for each cell independently (L.S. + SVD)
 - Embarrassingly parallel
- Given $\{R_i\}$, minimize energy to find $\{\mathbf{p}'_i\}$

$$\sum_{j \in N(i)} w_{ij} (\mathbf{p}'_i - \mathbf{p}'_j) = \sum_{j \in N(i)} \frac{w_{ij}}{2} (R_i + R_j) (\mathbf{p}_i - \mathbf{p}_j)$$

$$L\mathbf{p}' = \mathbf{b}$$



Advantages

“ L ” is only a function of the cotangent weights

- Depends only on original mesh
- Only needs to be factored once!

FAST!!

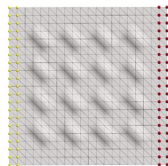
Rotations can be computed in parallel

- Each iteration reduces energy
 - Updating rotations guaranteed to reduce cell-error
 - Updating positions guaranteed to reduce global error

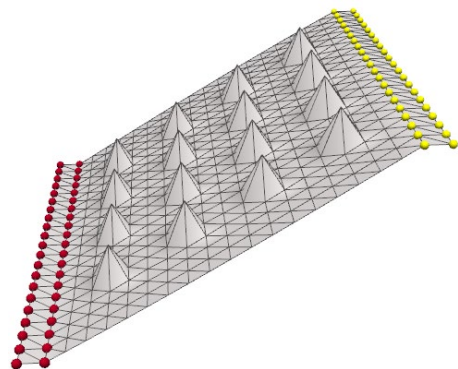
Guaranteed Convergence!!



Results (vs Poisson)



Poisson:



ARAP:

