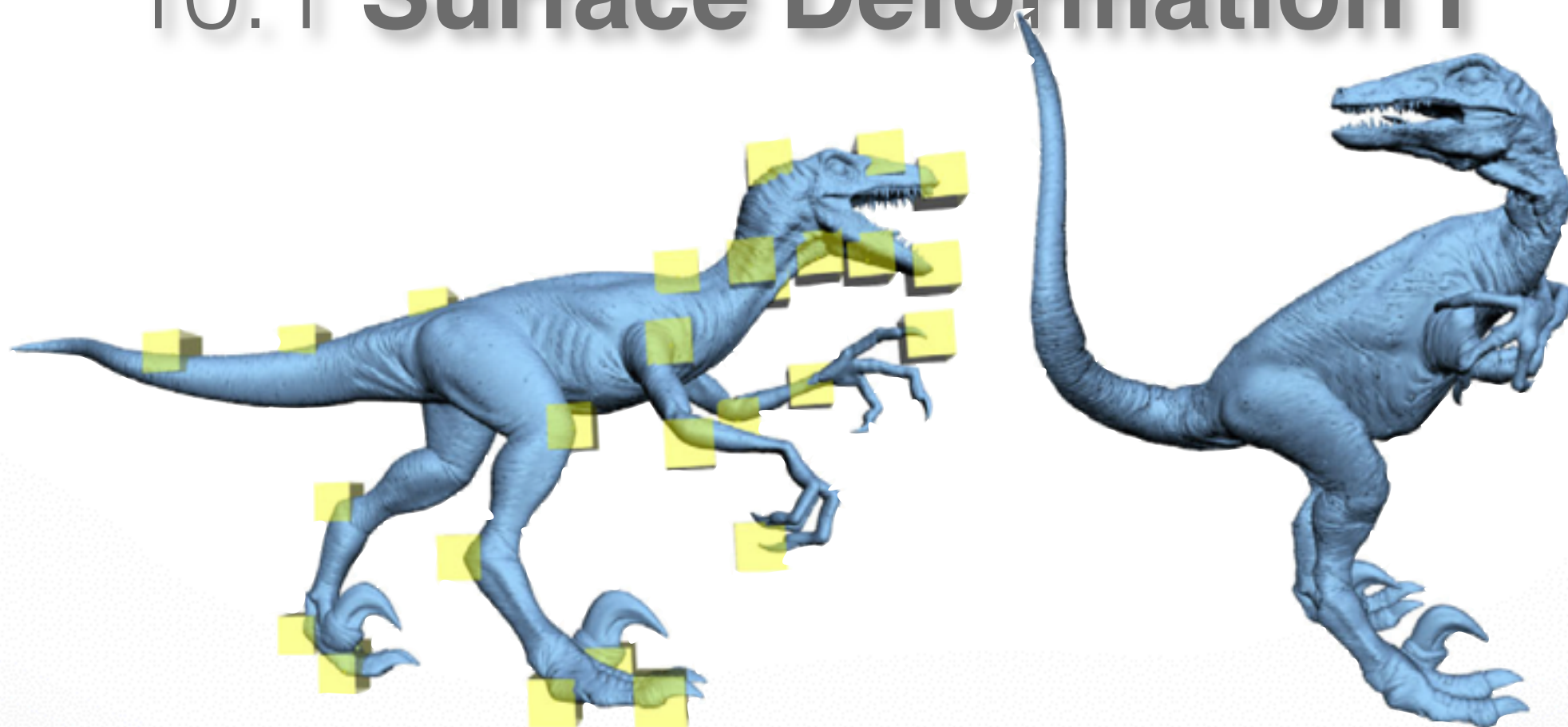


Spring 2015

CSCI 599: **Digital Geometry Processing**

10.1 **Surface Deformation I**



Hao Li

<http://cs599.hao-li.com>

Acknowledgement

Images and Slides are courtesy of

- Prof. Mario Botsch, Bielefeld University
- Prof. Olga Sorkine, ETH Zurich



Shapes & Deformation

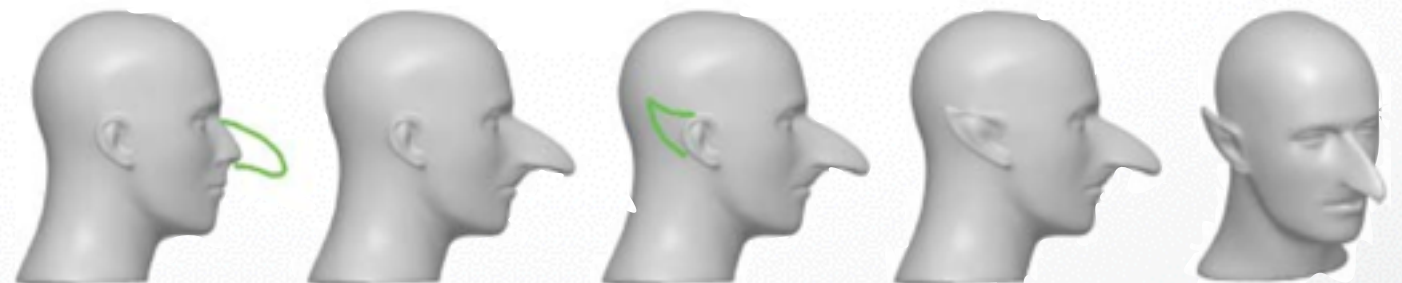
Why deformations?

- Sculpting, customization
- Character posing, animation



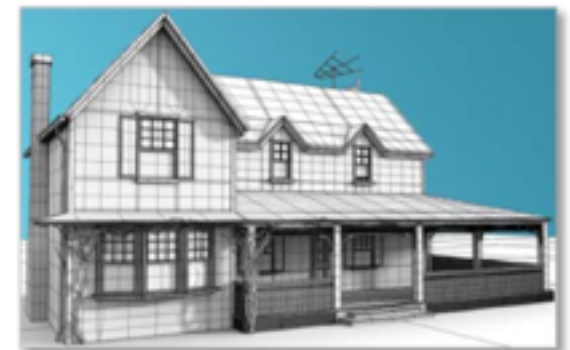
Criteria?

- Intuitive behavior and interface
- semantics
- Interactivity



Shapes & Deformation

- Manually modeled and scanned shape data
- Continuous and discrete shape representations



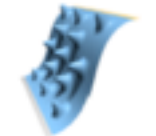
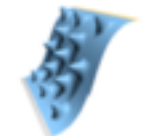
Goals

State of research in shape editing

Discuss practical considerations

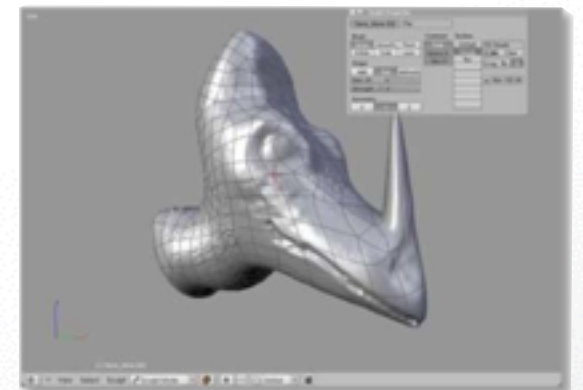
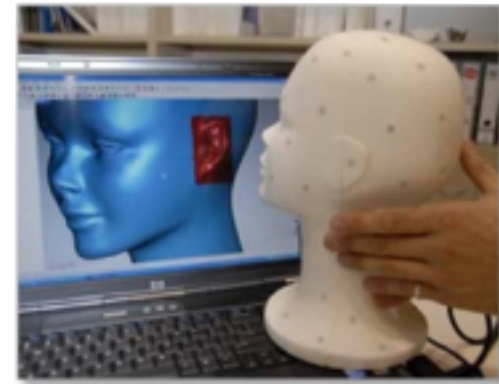
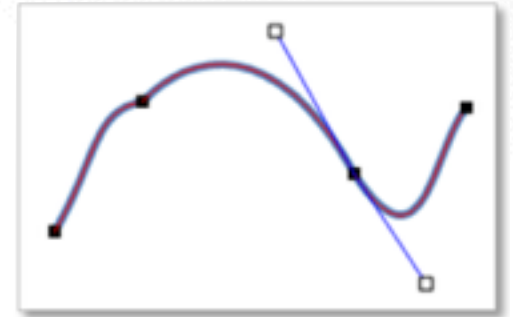
- Flexibility
- Numerical issues
- Admissible interfaces

Comparison, tradeoffs

Approach	Pen Translation	120° bend	135° twist	70° bend
Original model				
Non-linear prism-based modeling [12]				
Thin shells [10] + deformation transfer [14]				
Gradient-based editing [6]				
Implicit Laplacian-based editing [5]				
Rotation invariant coordinates [4]				

Continuous/Analytical Surfaces

- Tensor product surfaces (e.g. Bézier, B-Spline, NURBS)
- Subdivision Surfaces
- Editability is inherent to the representation

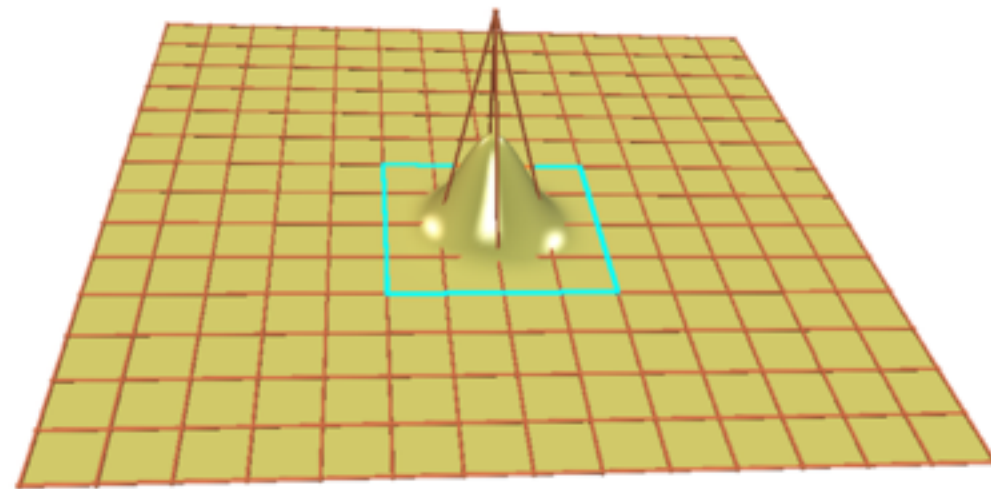


Spline Surfaces

Tensor product surfaces (“curves of curves”)

- Rectangular grid of control points

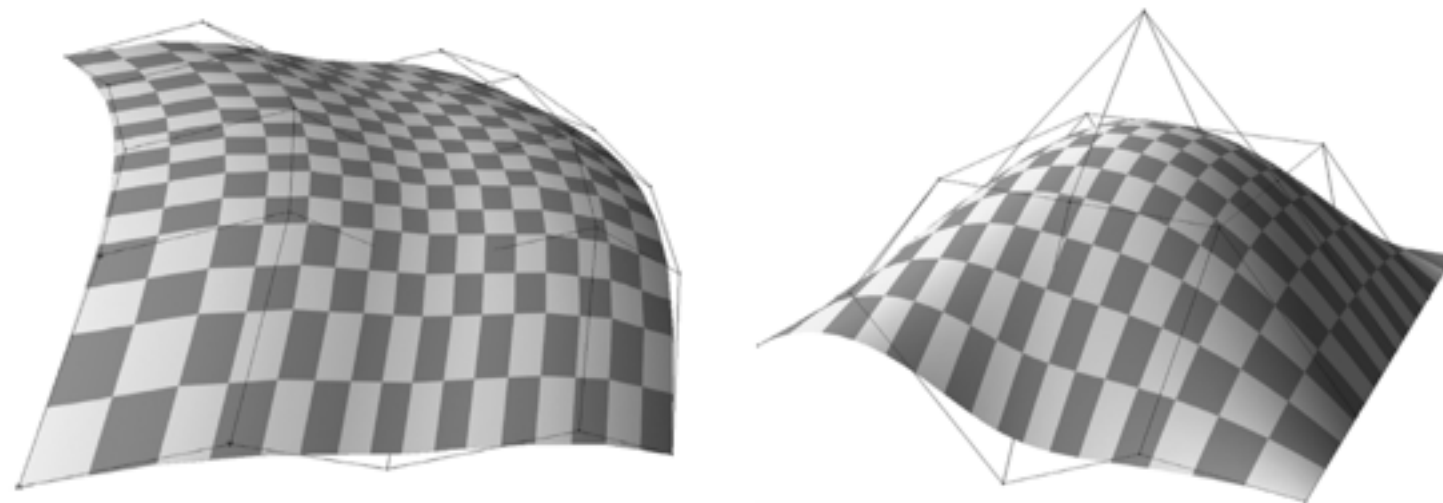
$$\mathbf{p}(u, v) = \sum_{i=0}^k \sum_{j=0}^l \mathbf{p}_{ij} N_i^n(u) N_j^n(v)$$



Spline Surfaces

Tensor product surfaces (“curves of curves”)

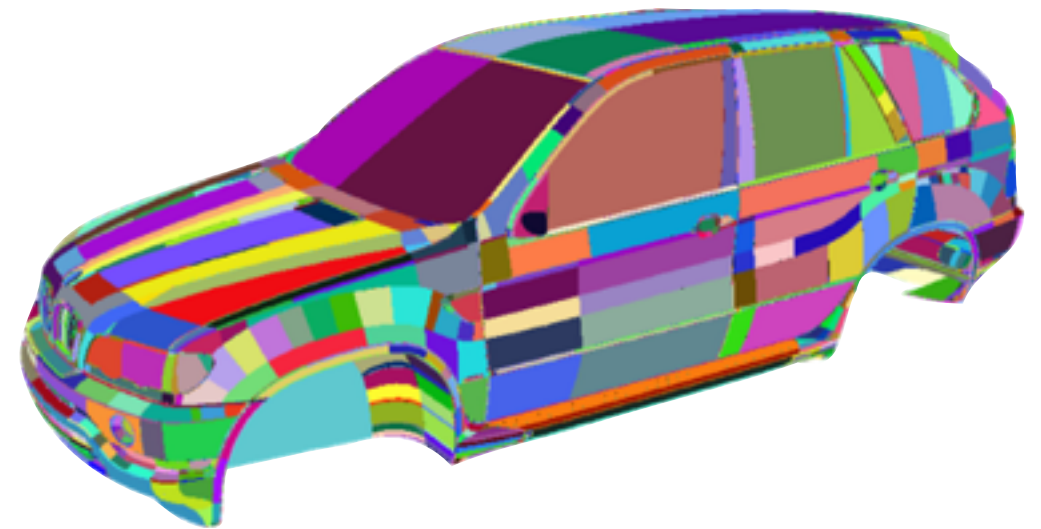
- Rectangular grid of control points
- Rectangular surface patch



Spline Surfaces

Tensor product surfaces (“curves of curves”)

- Rectangular grid of control points
- Rectangular surface patch



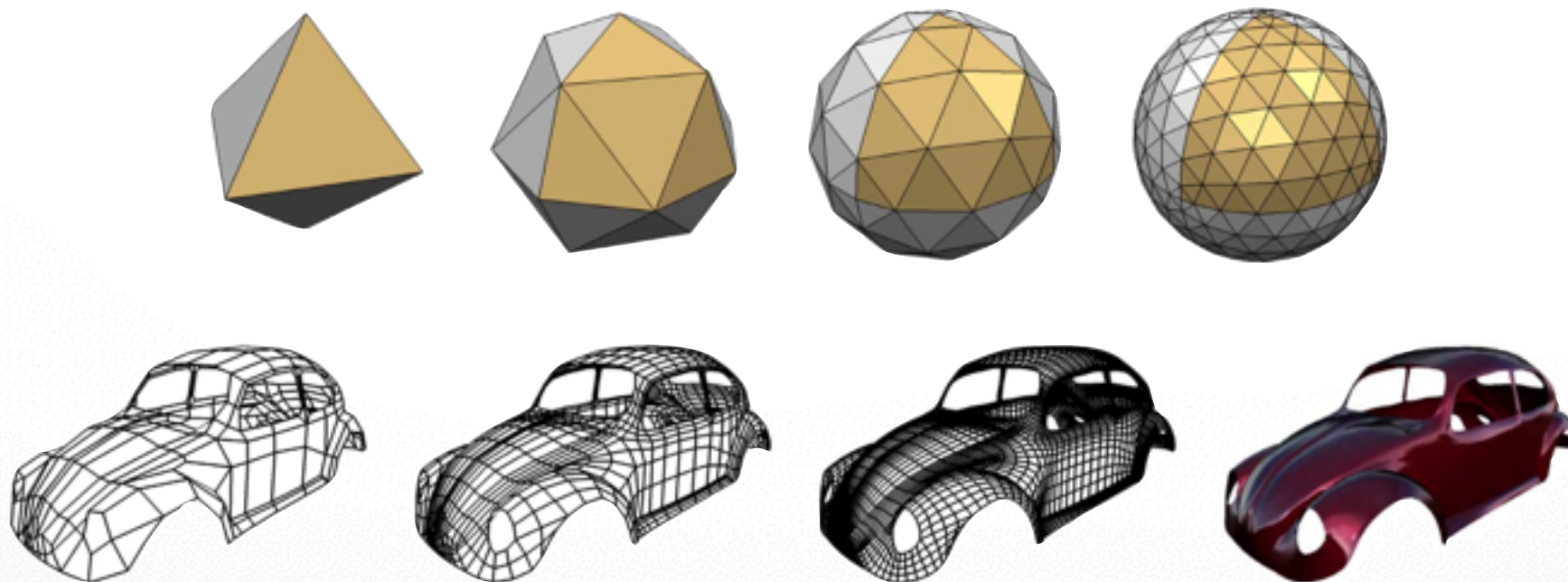
Problems:

- Many patches for complex models
- Smoothness across patch boundaries
- Trimming for non-rectangular patches

Subdivision Surfaces

Generalization of spline curves/surfaces

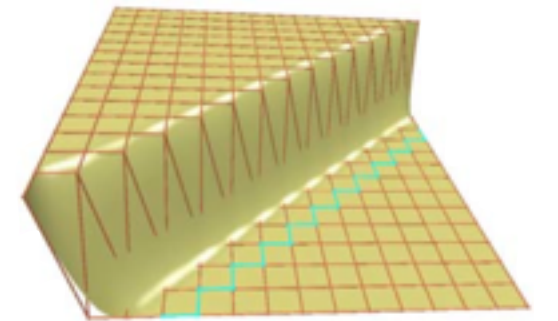
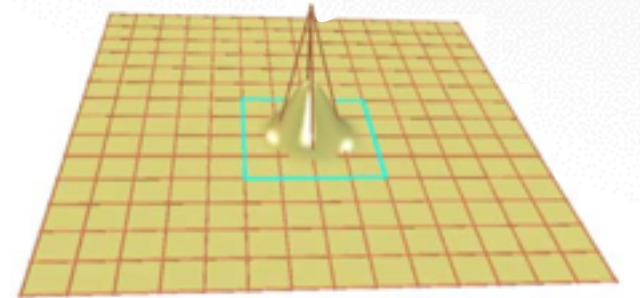
- Arbitrary control meshes
- Successive refinement (subdivision)
- Converges to smooth limit surface
- Connection between splines and meshes



Spline & Subdivision Surfaces

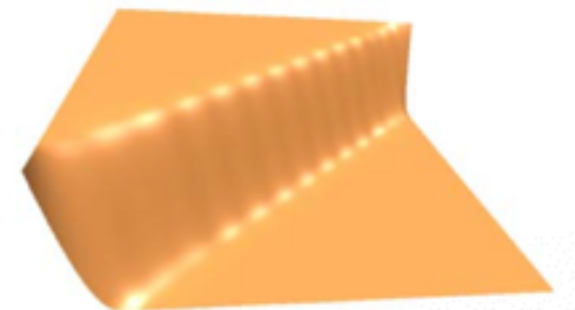
Basis functions are smooth bumps

- Fixed support
- Fixed control grid



Bound to control points

- Initial patch layout is crucial
- Requires experts!



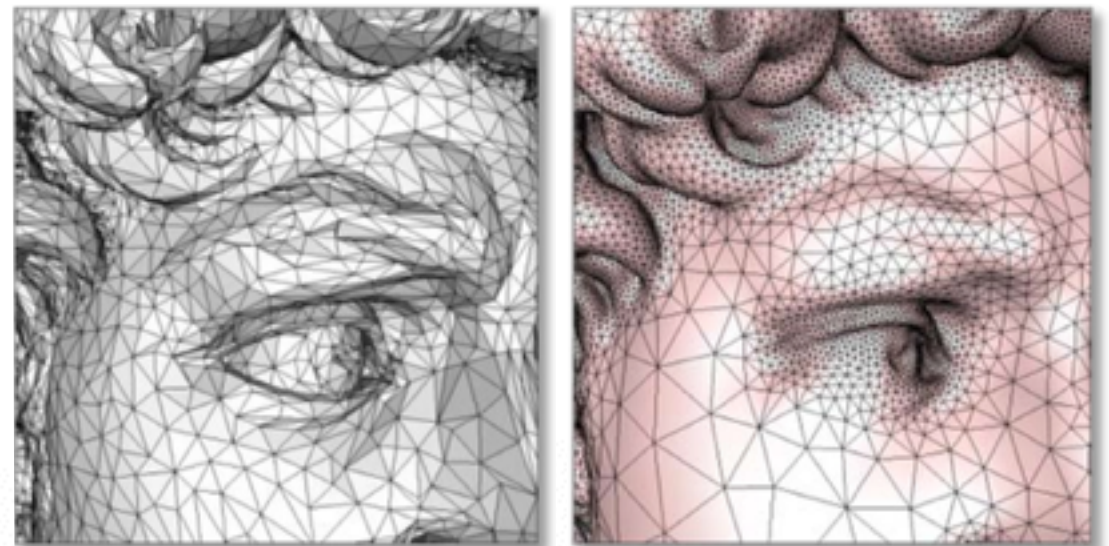
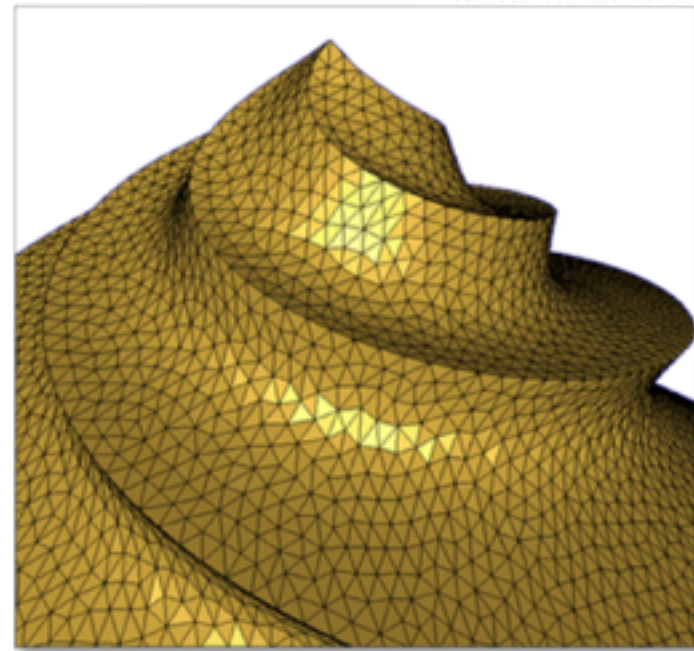
De-couple deformation from surface representation!

Discrete Surfaces: Point Sets, Meshes

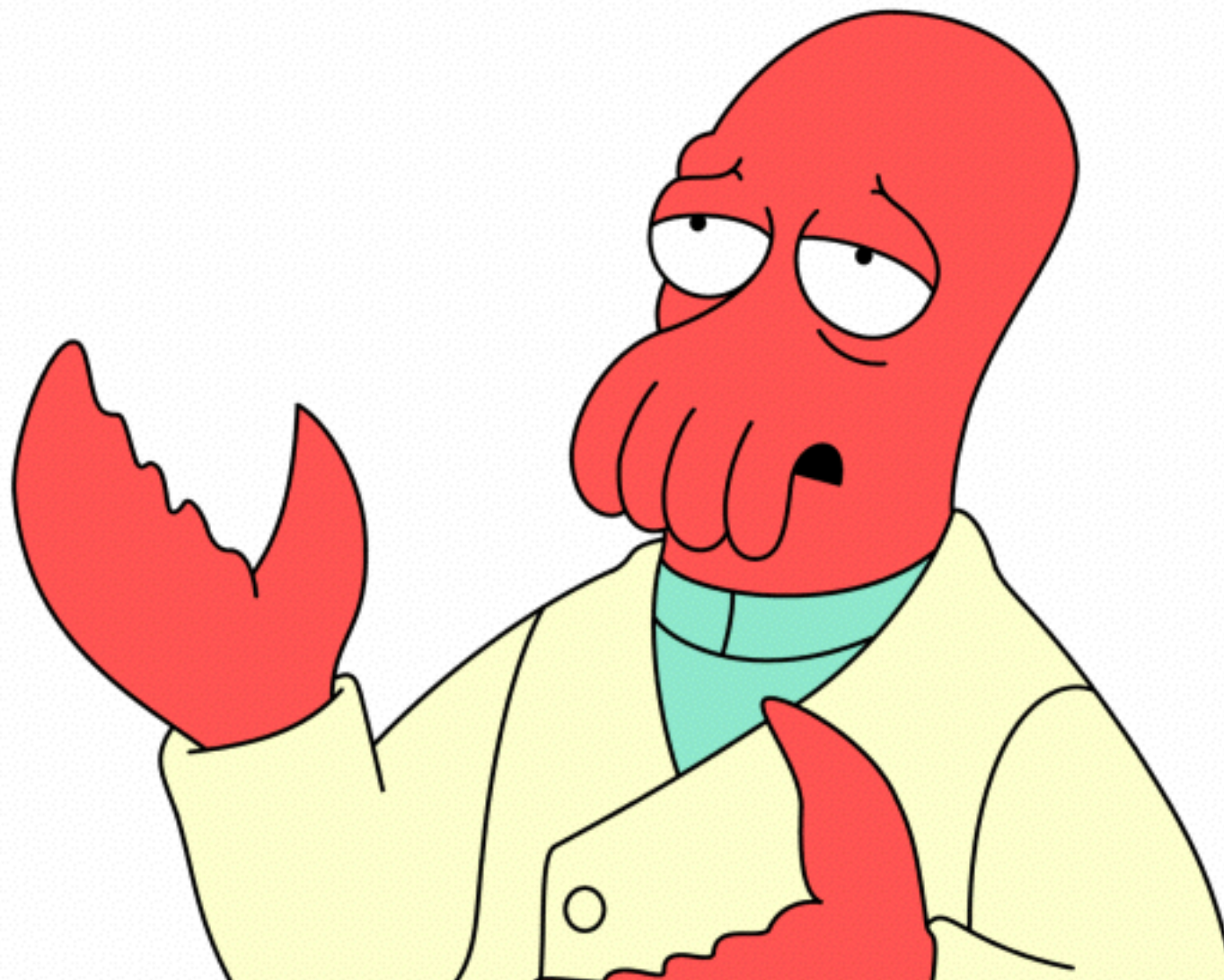
- Flexible
- Suitable for highly detailed scanned data
- No analytic surface
- No inherent “editability”



Mesh Editing



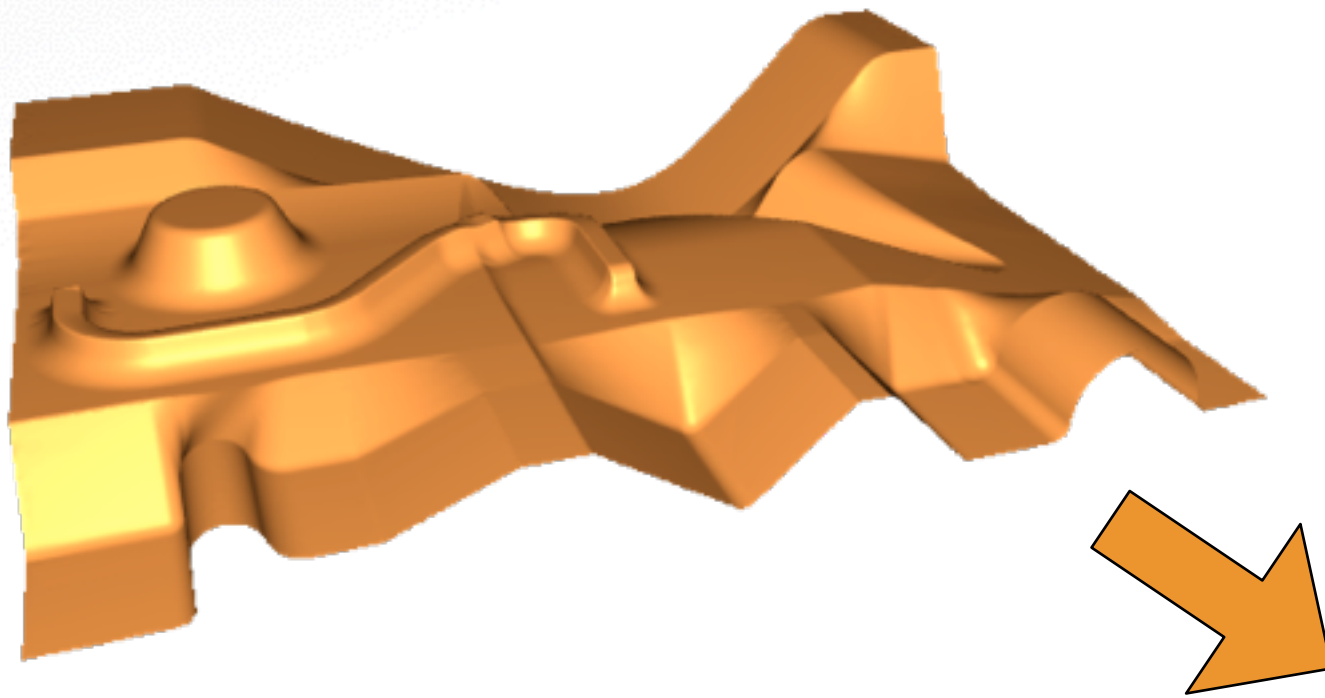
Demo



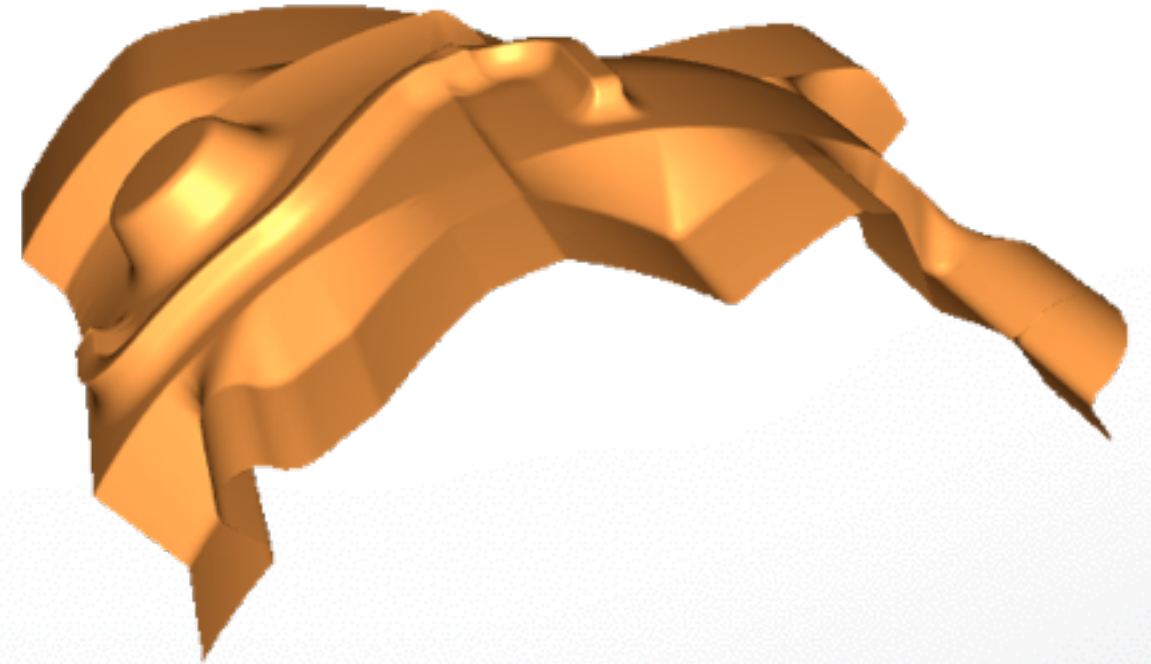
Outline

- **Surface-Based Deformation**
 - **Linear Methods**
 - Non-Linear Methods
- Spatial Deformation

Mesh Deformation

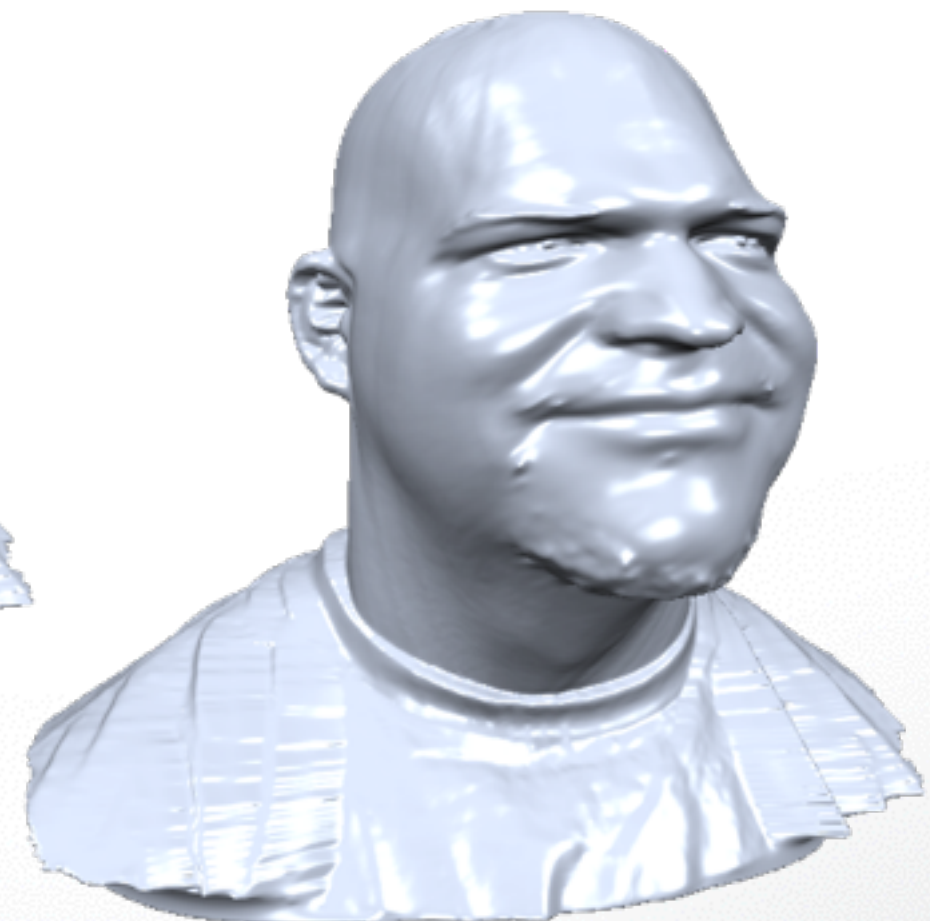
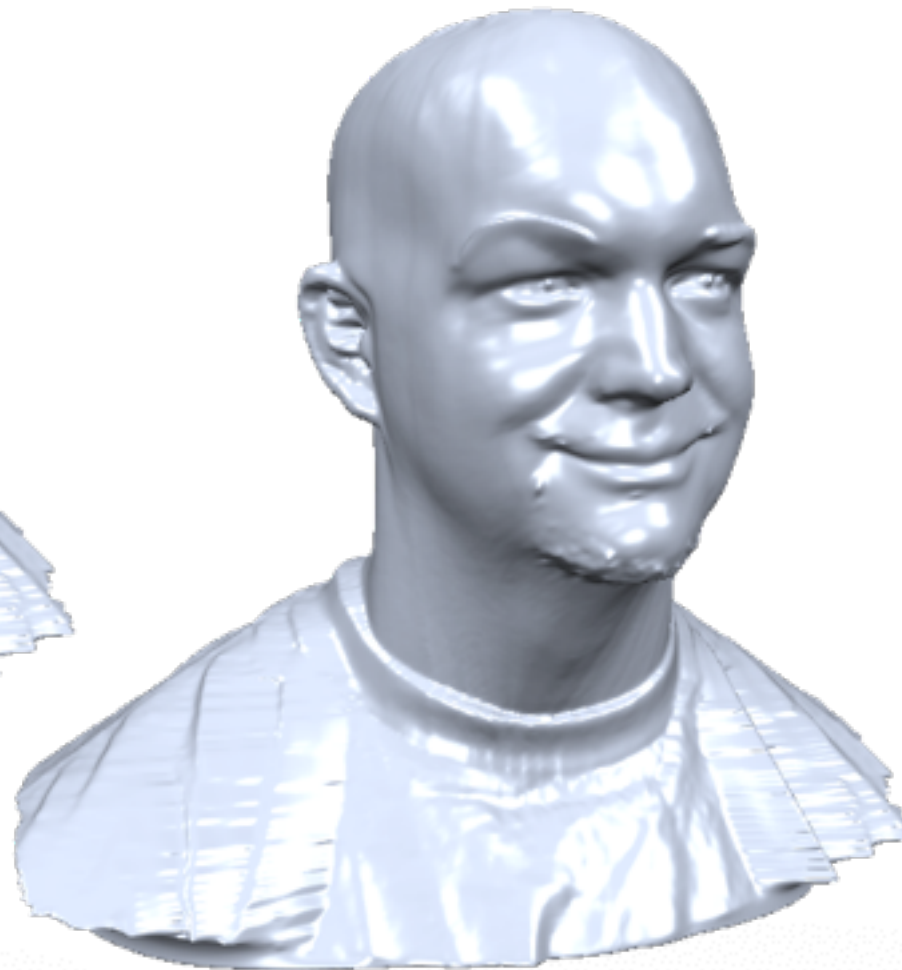
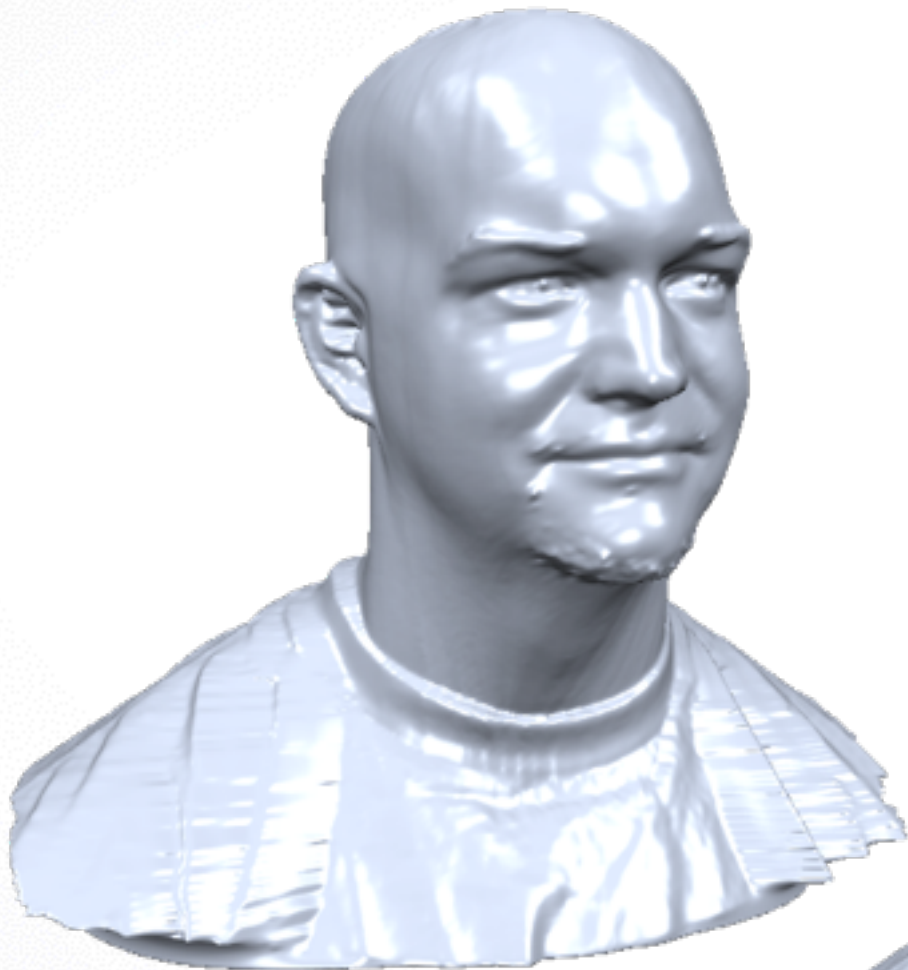


Global deformation
with intuitive
detail preservation



Mesh Deformation

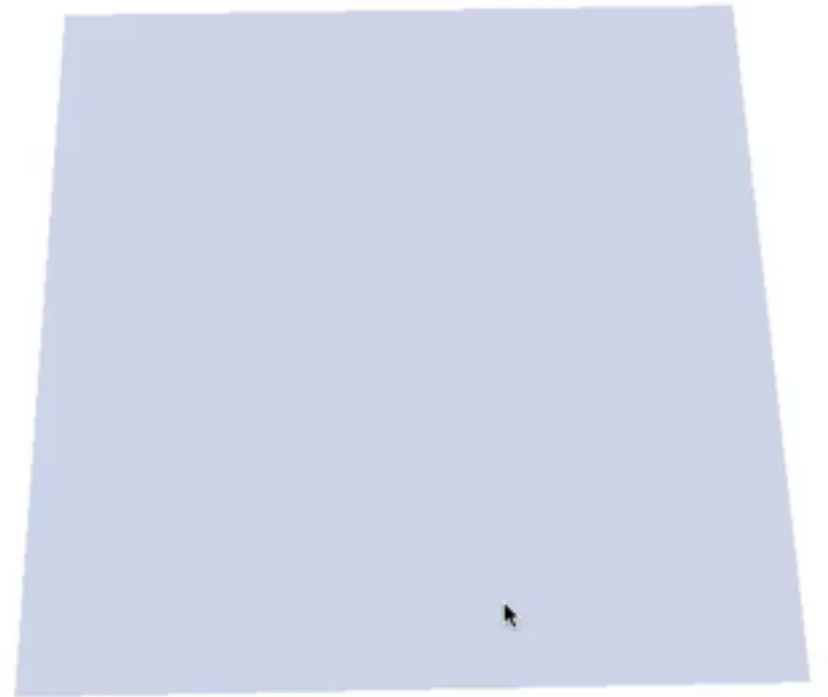
Local & global
deformations



Linear Surface-Based Deformation

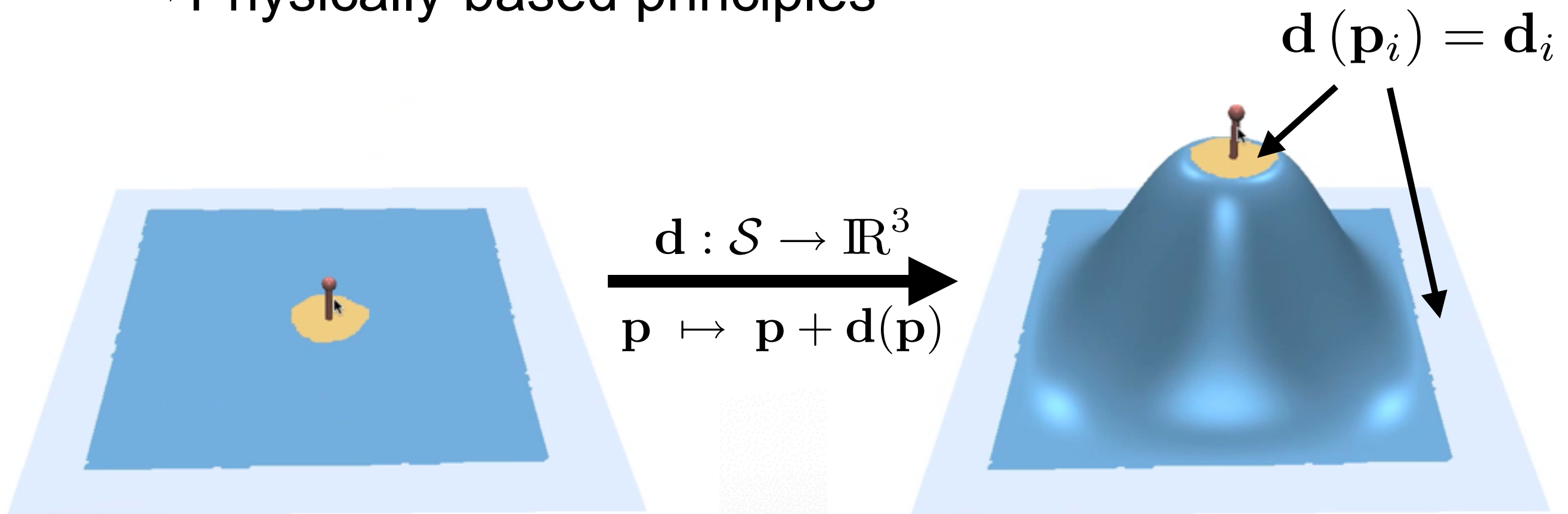
- **Shell-Based Deformation**
- Multiresolution Deformation
- Differential Coordinates

Modeling Metaphor



Modeling Metaphor

- Mesh deformation by displacement function $\mathbf{d}$
 - Interpolate prescribed constraints
 - Smooth, intuitive deformation
 - ➔ Physically-based principles



Shell Deformation Energy

- **Stretching**

- Change of local distances
- Captured by 1st fundamental form

$$\int_{\Omega} k_s \|\mathbf{I} - \bar{\mathbf{I}}\|^2$$

$$\mathbf{I} = \begin{bmatrix} \mathbf{x}_u^T \mathbf{x}_u & \mathbf{x}_u^T \mathbf{x}_v \\ \mathbf{x}_v^T \mathbf{x}_u & \mathbf{x}_v^T \mathbf{x}_v \end{bmatrix}$$

- **Bending**

- Change of local curvature
- Captured by 2nd fundamental form

$$\int_{\Omega} k_b \|\mathbf{\Pi} - \bar{\mathbf{\Pi}}\|^2$$

$$\mathbf{\Pi} = \begin{bmatrix} \mathbf{x}_{uu}^T \mathbf{n} & \mathbf{x}_{uv}^T \mathbf{n} \\ \mathbf{x}_{vu}^T \mathbf{n} & \mathbf{x}_{vv}^T \mathbf{n} \end{bmatrix}$$

- **Stretching & bending is sufficient**

- Differential geometry: “1st and 2nd fundamental forms determine a surface up to rigid motion.”

Physically-Based Deformation

- Nonlinear stretching & bending energies

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

stretching bending

- Linearize terms $\rightarrow$ Quadratic energy

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

stretching bending

Physically-Based Deformation

- Minimize linearized bending energy

$$E(\mathbf{d}) = \int_{\mathcal{S}} \|\mathbf{d}_{uu}\|^2 + 2 \|\mathbf{d}_{uv}\|^2 + \|\mathbf{d}_{vv}\|^2 \, dudv \rightarrow \min$$

$f(x) \rightarrow \min$

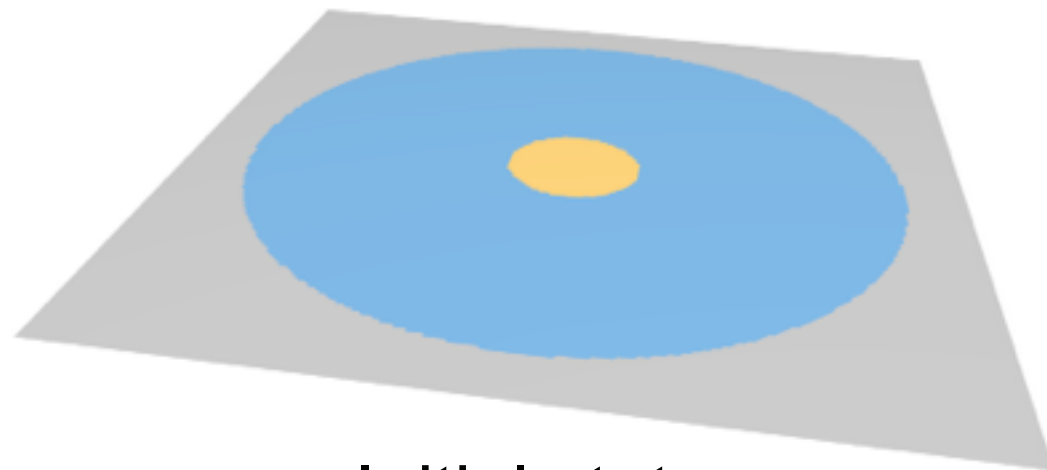
- Variational calculus $\rightarrow$ Euler-Lagrange PDE

$$\Delta^2 \mathbf{d} := \mathbf{d}_{uuuu} + 2\mathbf{d}_{uuvv} + \mathbf{d}_{vvvv} = 0$$

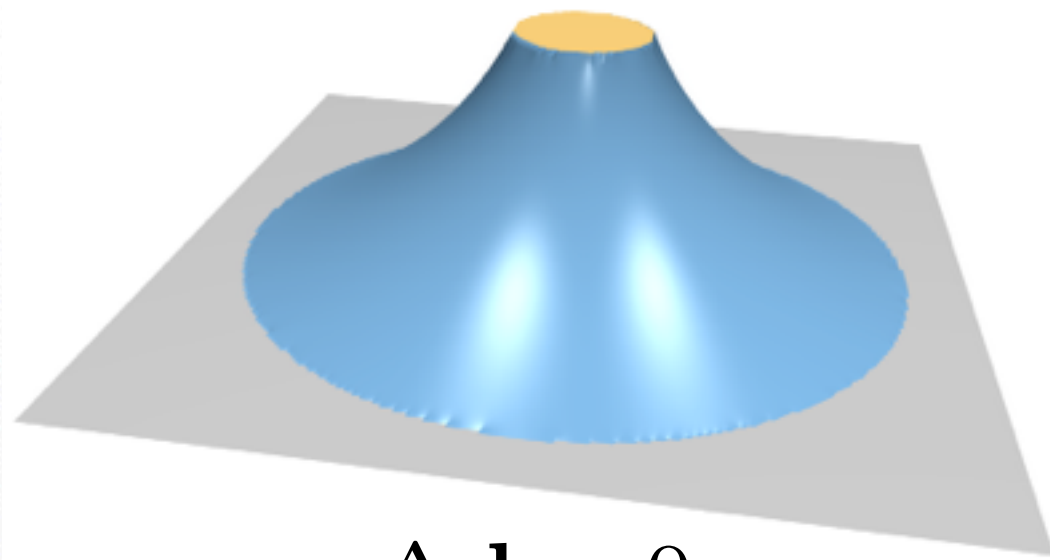
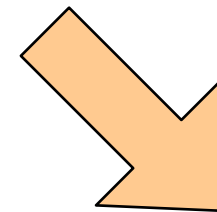
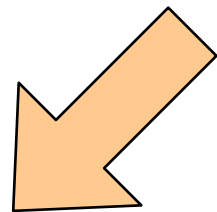
$f'(x) = 0$

$\Rightarrow$ “Best” deformation that satisfies constraints

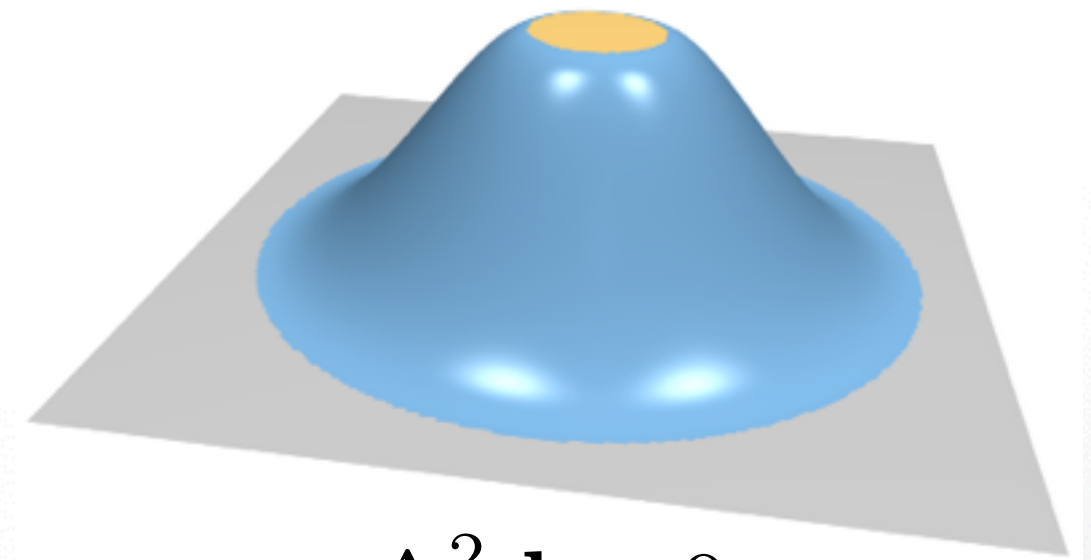
Deformation Energies



Initial state



$\Delta d = 0$
(Membrane)



$\Delta^2 d = 0$
(Thin plate)

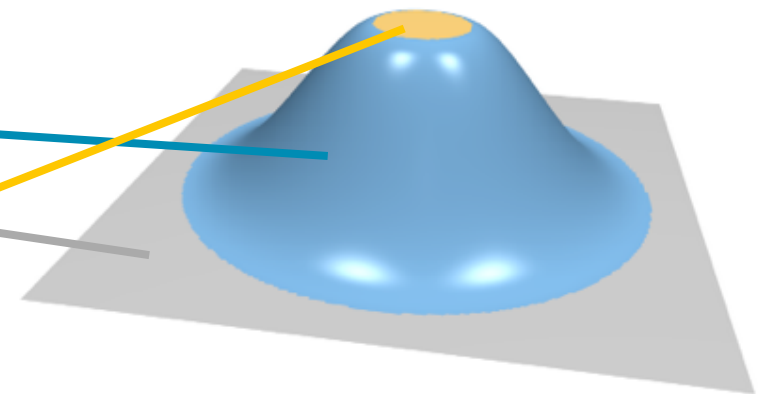
PDE Discretization

- Euler-Lagrange PDE

$$\Delta^2 \mathbf{d} = 0$$

$$\mathbf{d} = 0$$

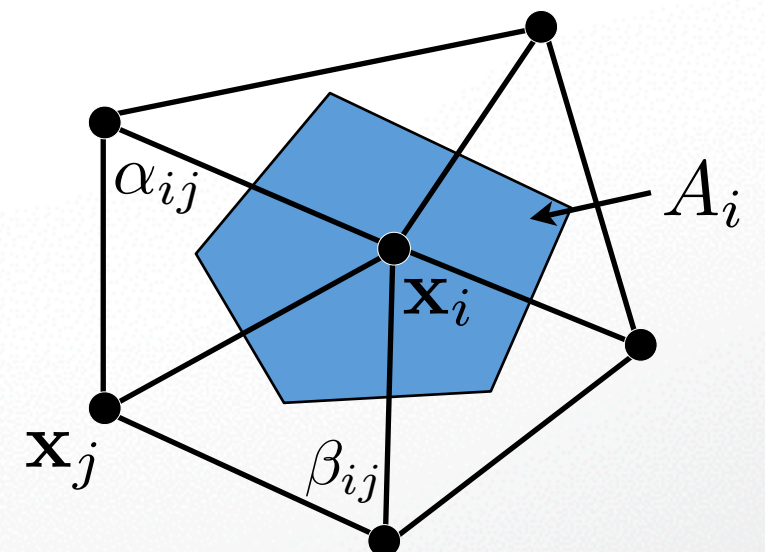
$$\mathbf{d} = \delta \mathbf{h}$$



- Laplace discretization

$$\Delta \mathbf{d}_i = \frac{1}{2A_i} \sum_{j \in \mathcal{N}_i} (\cot \alpha_{ij} + \cot \beta_{ij})(\mathbf{d}_j - \mathbf{d}_i)$$

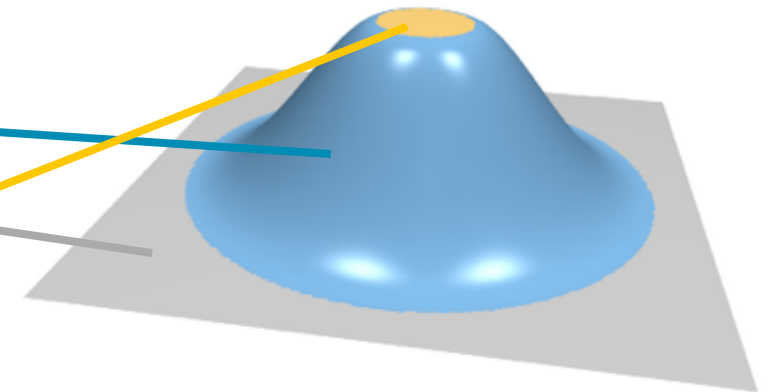
$$\Delta^2 \mathbf{d}_i = \Delta(\Delta \mathbf{d}_i)$$



Linear System

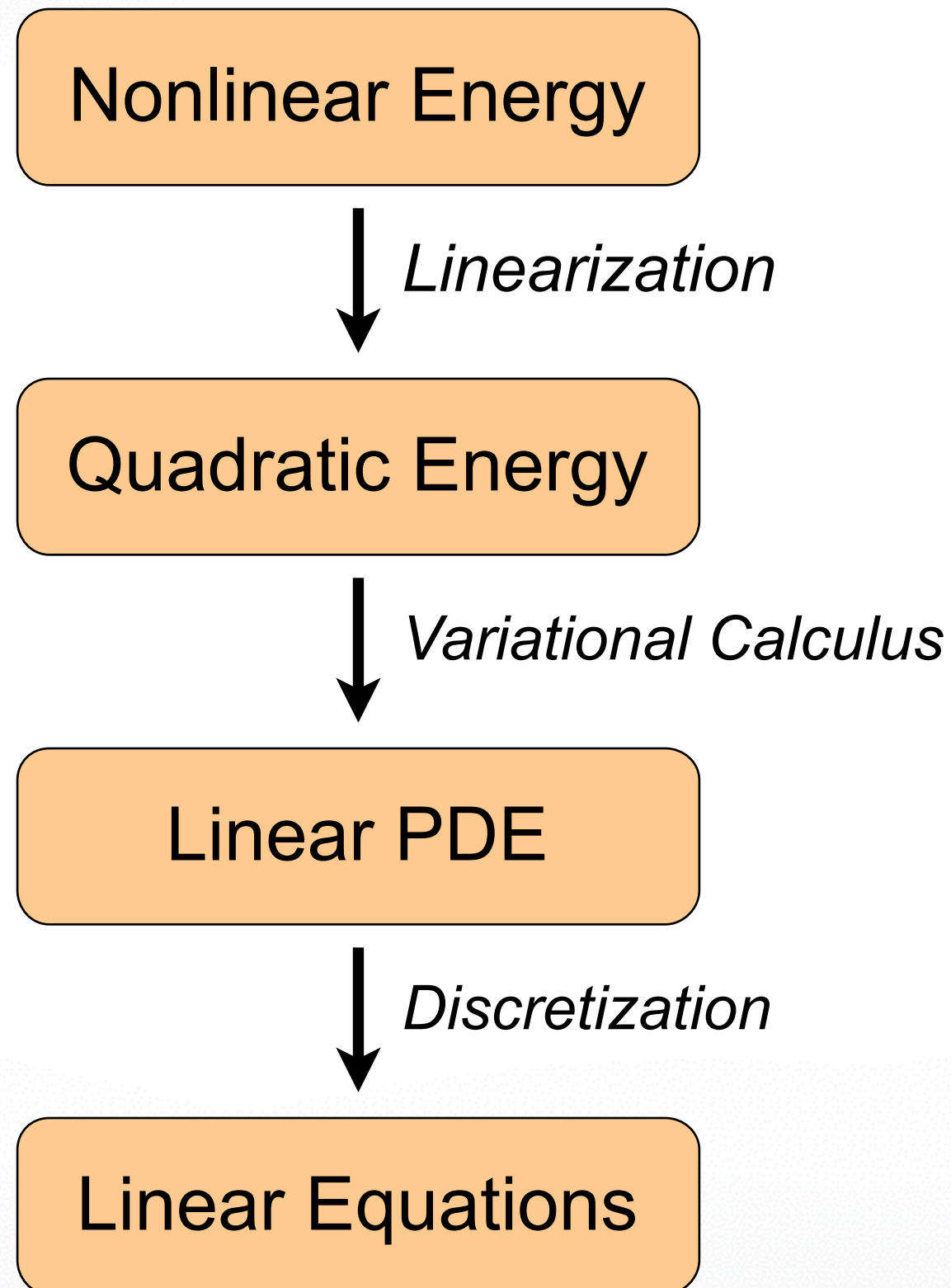
- Sparse linear system (19 nz/row)

$$\begin{pmatrix} & \Delta^2 & \\ 0 & \mathbf{I} & 0 \\ 0 & 0 & \mathbf{I} \end{pmatrix} \begin{pmatrix} \vdots \\ \mathbf{d}_i \\ \vdots \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \delta \mathbf{h}_i \end{pmatrix}$$

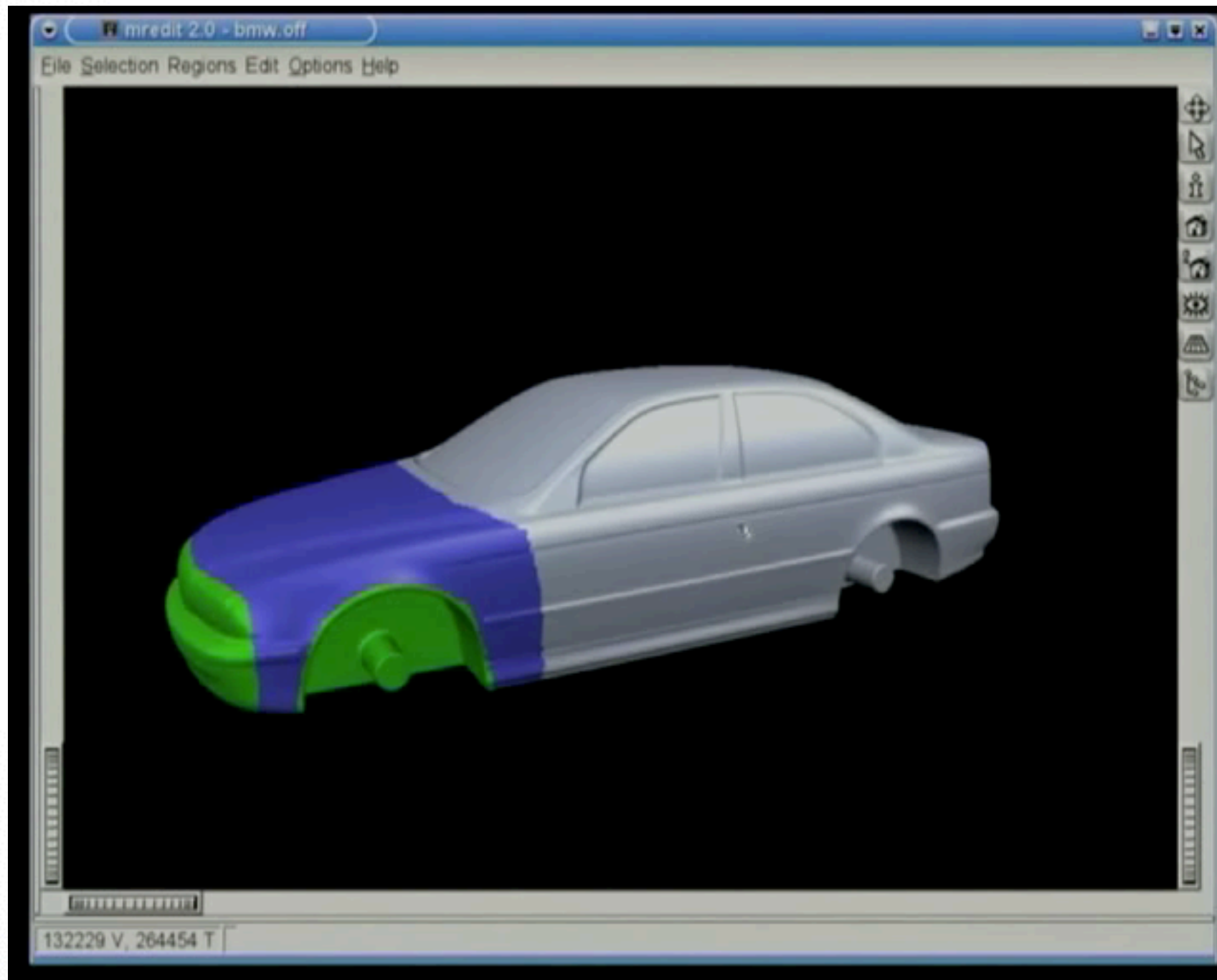


- Turn into symmetric positive definite system
- Solve this system *each frame*
 - Use efficient linear solvers !!!
 - Sparse Cholesky factorization
 - **See book for details**

Derivation Steps



CAD-Like Deformation



[Botsch & Kobbelt, SIGGRAPH 04]

Facial Animation

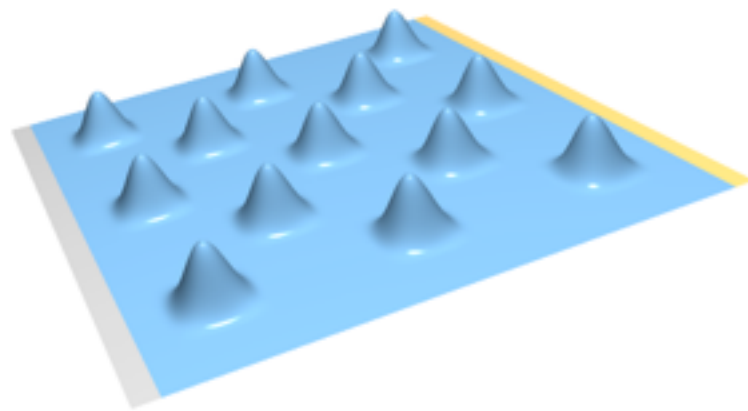


Linear Surface-Based Deformation

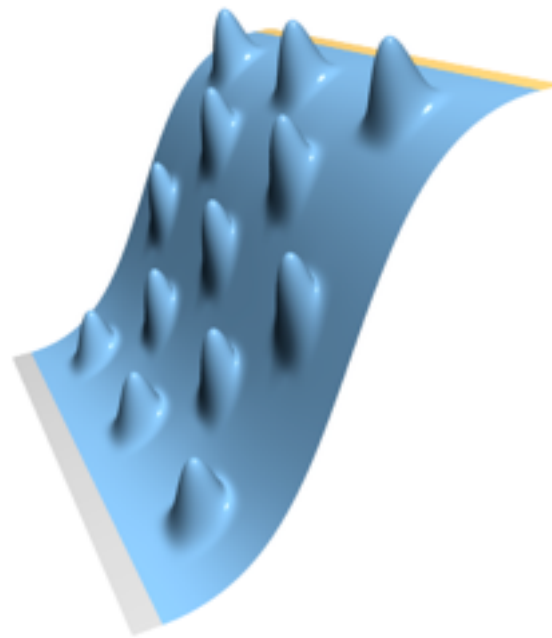
- Shell-Based Deformation
- **Multiresolution Deformation**
- Differential Coordinates

Multiresolution Modeling

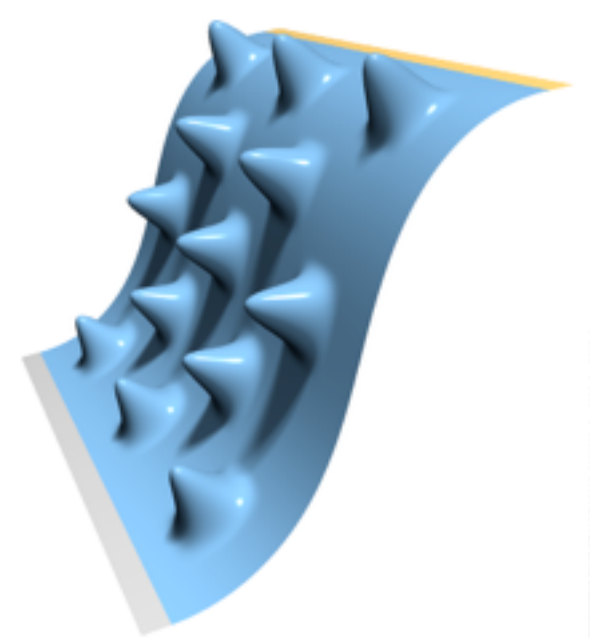
- Even pure translations induce local rotations!
 - ➔ Inherently non-linear coupling
- Alternative approach
 - Linear deformation + multi-scale decomposition...



Original

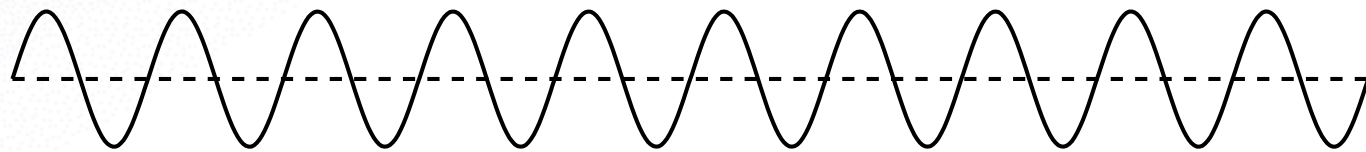


Linear



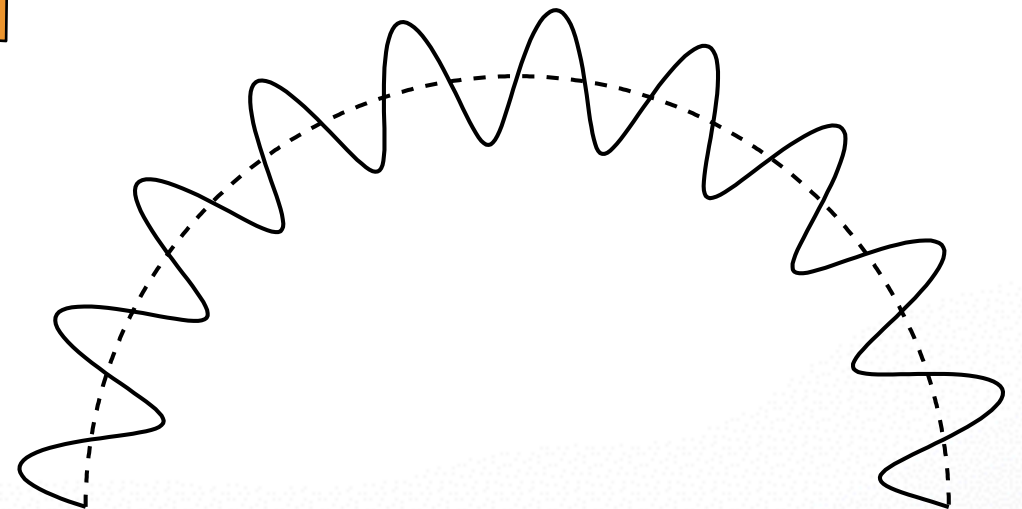
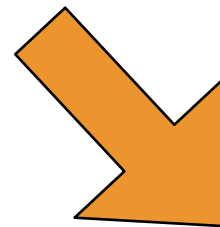
Nonlinear

Multiresolution Editing



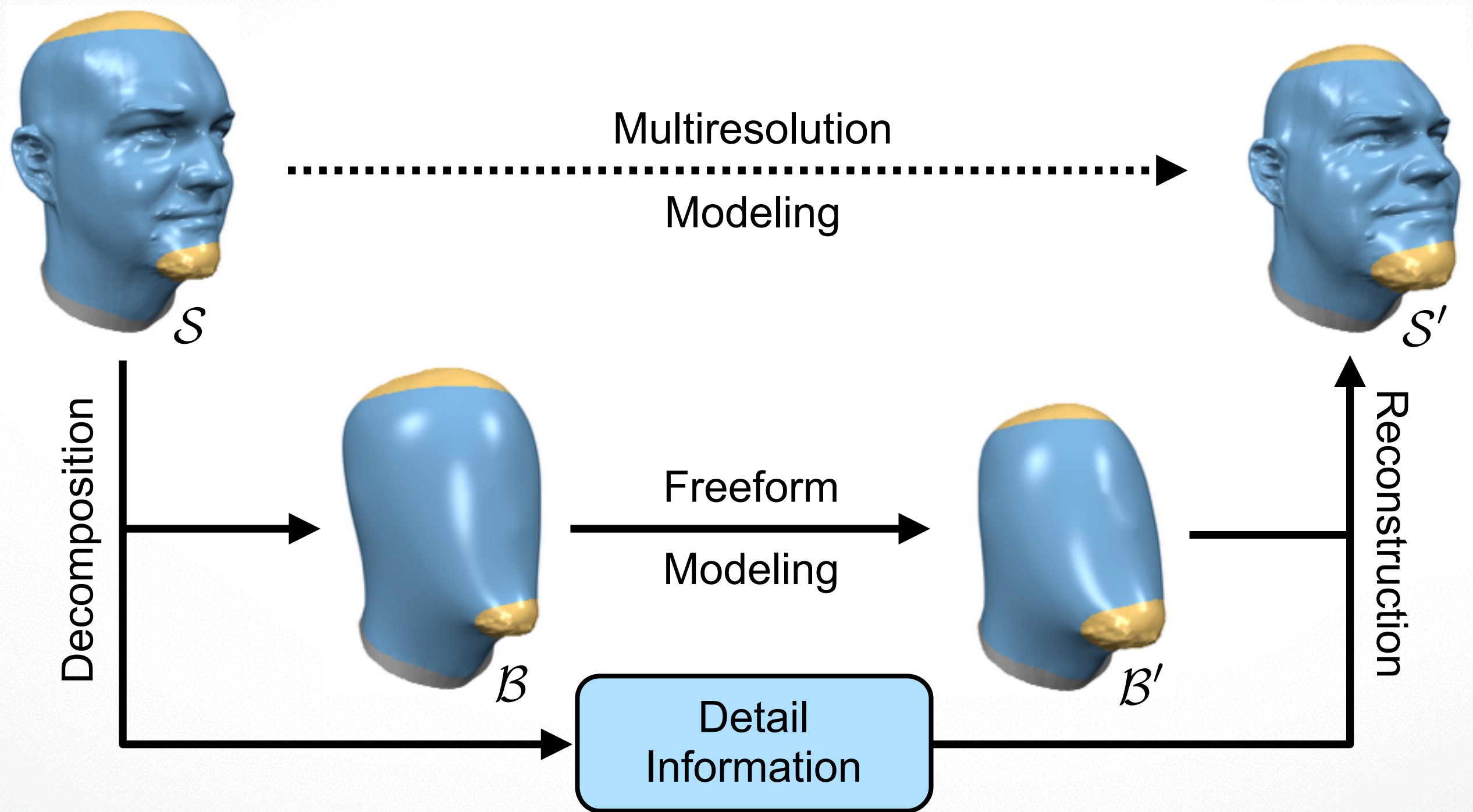
Frequency decomposition

Change low
frequencies

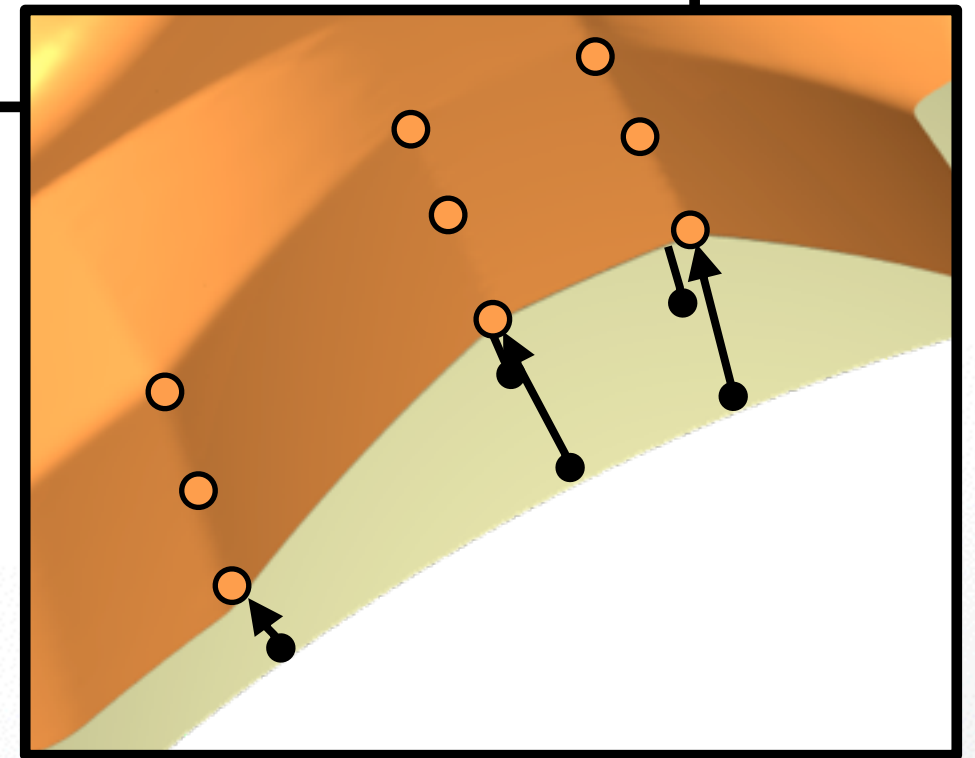
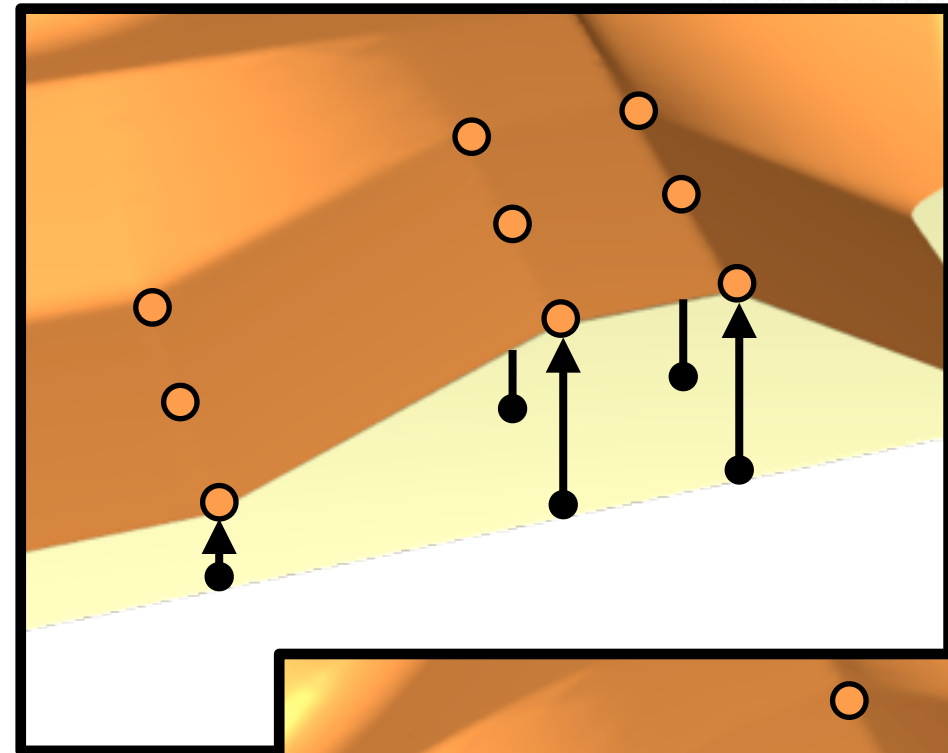
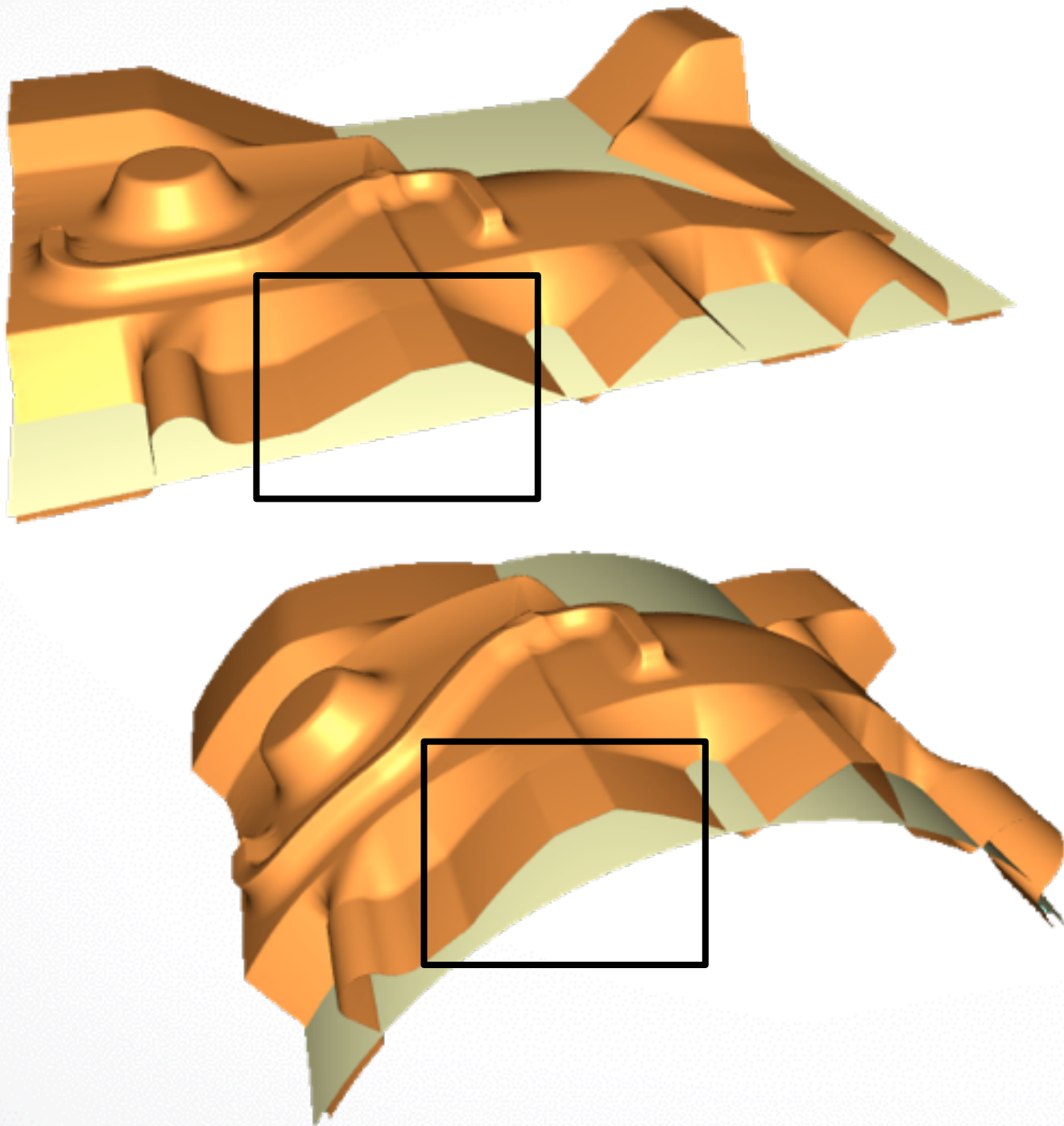


Add high frequency details,
stored in local frames

Multiresolution Editing

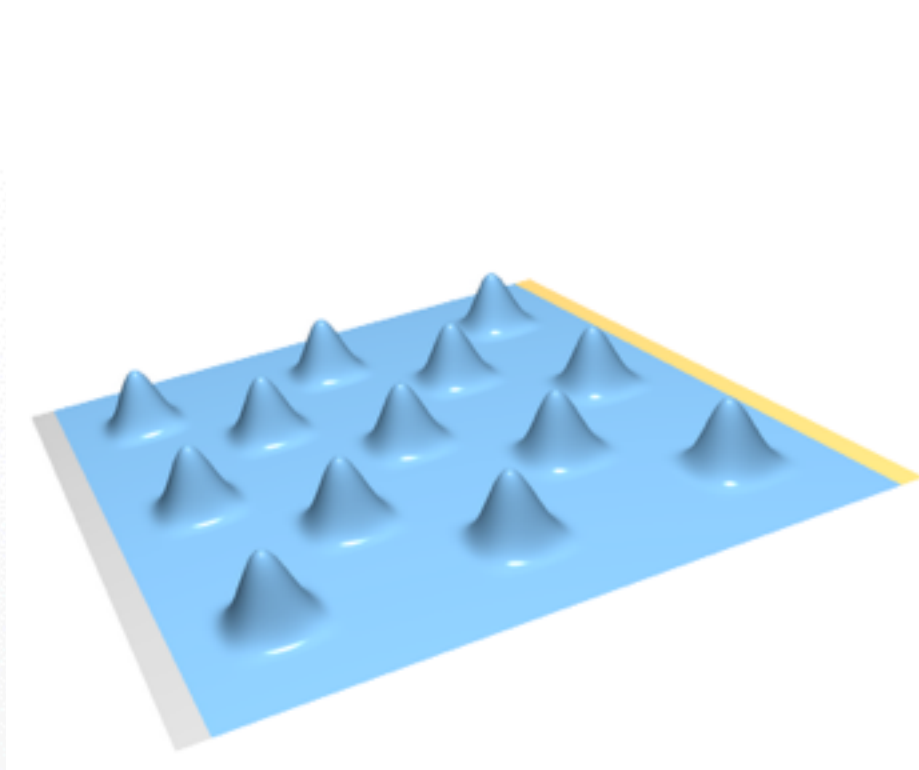


Normal Displacements

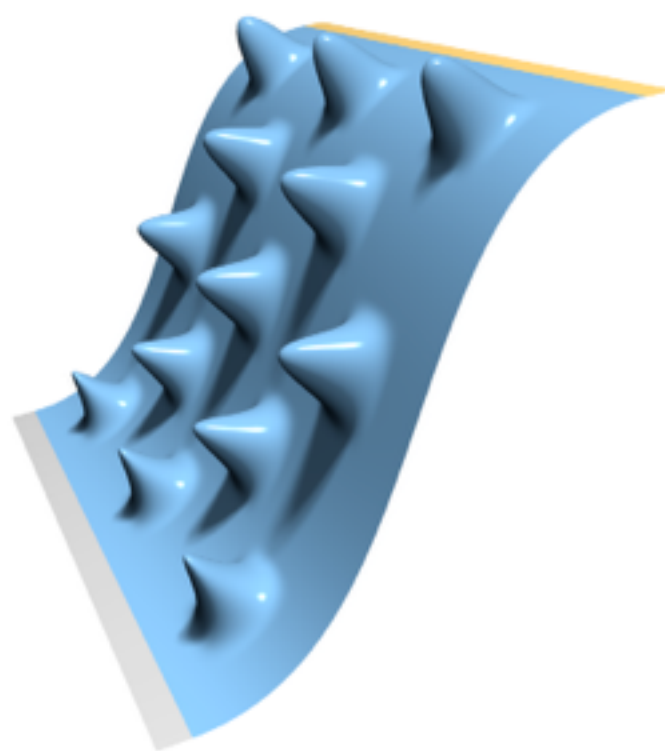


Limitations

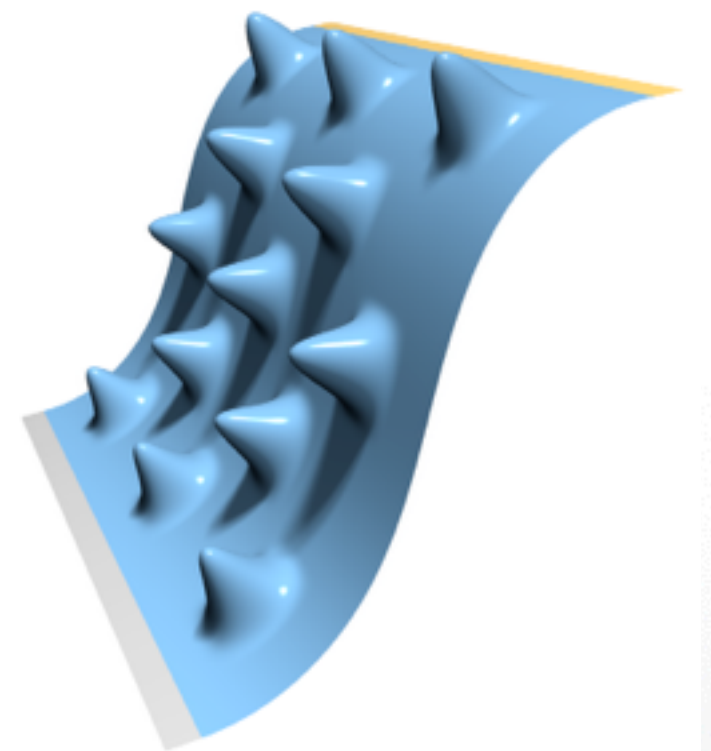
- Neighboring displacements are not coupled
 - Surface bending changes their angle
 - Leads to volume changes or self-intersections



Original



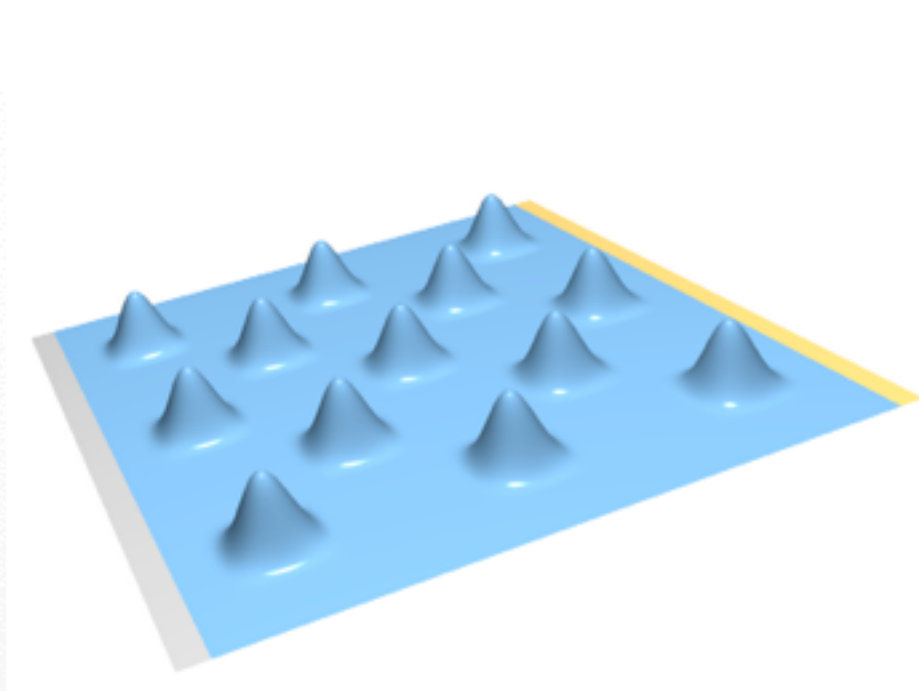
Normal Displ.



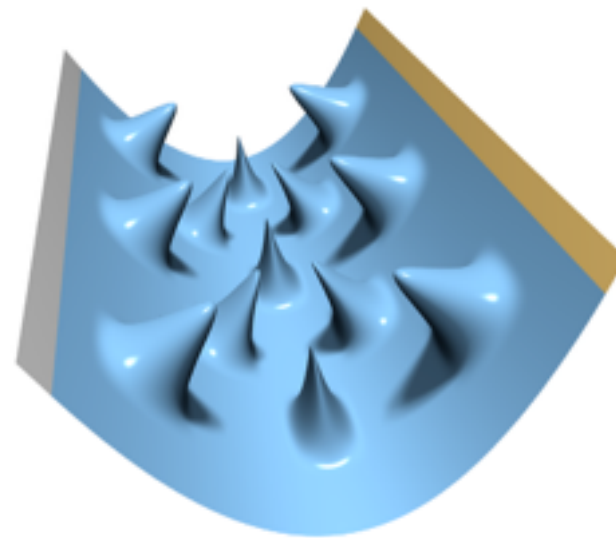
Nonlinear

Limitations

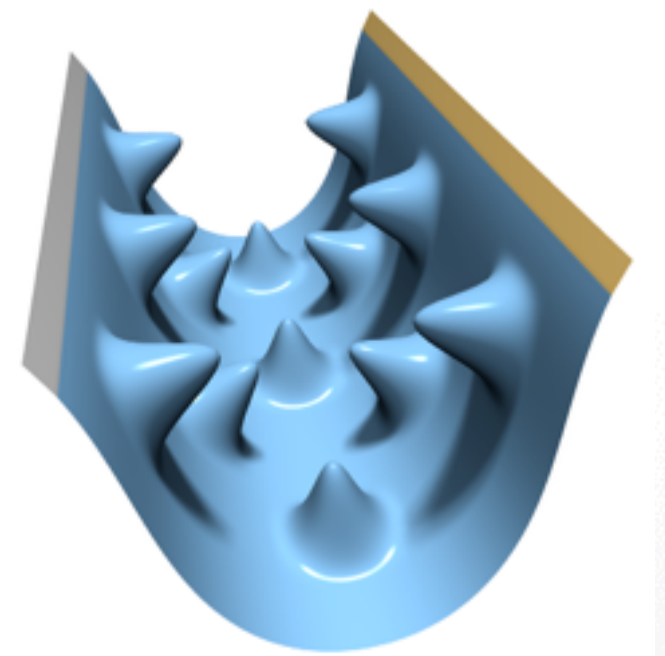
- Neighboring displacements are not coupled
 - Surface bending changes their angle
 - Leads to volume changes or self-intersections



Original



Normal Displ.



Nonlinear

Limitations

- Neighboring displacements are not coupled
 - Surface bending changes their angle
 - Leads to volume changes or self-intersections
- Multiresolution hierarchy difficult to compute
 - Complex topology
 - Complex geometry
 - Might require more hierarchy levels

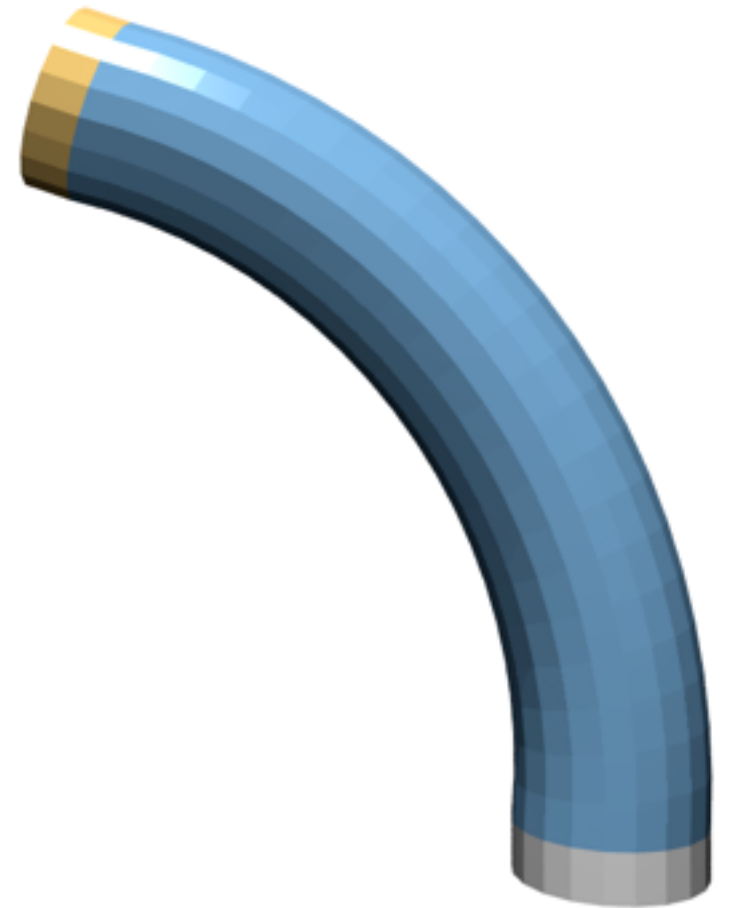
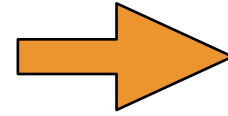
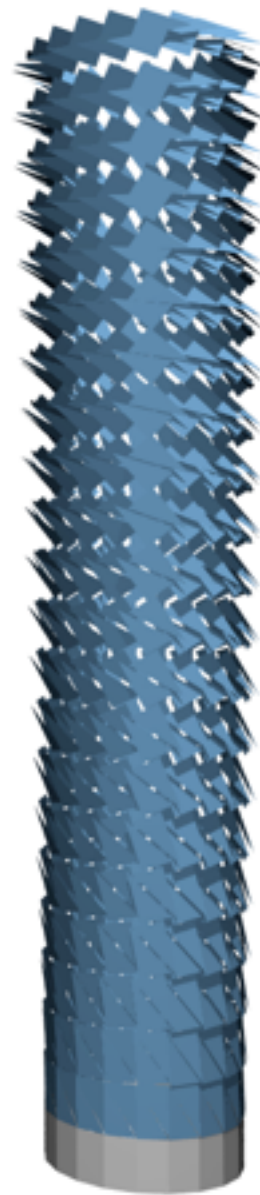
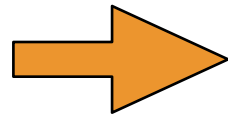
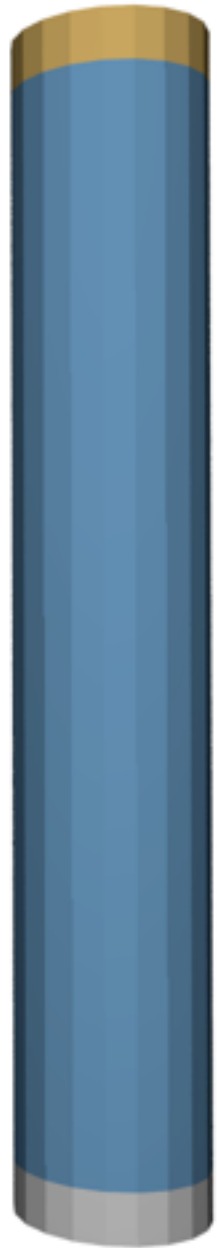
Linear Surface-Based Deformation

- Shell-Based Deformation
- Multiresolution Deformation
- **Differential Coordinates**

Differential Coordinates

1. Manipulate differential coordinates instead of *spatial* coordinates
 - Gradients, Laplacians, local frames
 - Intuition: Close connection to surface normal
2. Find mesh with desired differential coords
 - Cannot be solved exactly
 - Formulate as energy minimization

Differential Coordinates



Original

Rotated Diff-Coords

Reconstructed Mesh

Differential Coordinates

- **Which differential coordinate δ_i ?**
 - Gradients
 - Laplacians
 - ...
- **How to get local transformations $T_i(\delta_i)$?**
 - Smooth propagation
 - Implicit optimization
 - ...

Gradient-Based Editing

- Manipulate gradient of a function (e.g. a surface)

$$\mathbf{g} = \nabla \mathbf{f} \quad \mathbf{g} \mapsto \mathbf{T}(\mathbf{g})$$

- Find function $\mathbf{f}'$ whose gradient is (close to) $\mathbf{g}' = \mathbf{T}(\mathbf{g})$

$$\mathbf{f}' = \operatorname{argmin}_{\mathbf{f}} \int_{\Omega} \|\nabla \mathbf{f} - \mathbf{T}(\mathbf{g})\|^2 \, du \, dv$$

- Variational calculus $\rightarrow$ Euler-Lagrange PDE

$$\Delta \mathbf{f}' = \operatorname{div} \mathbf{T}(\mathbf{g})$$

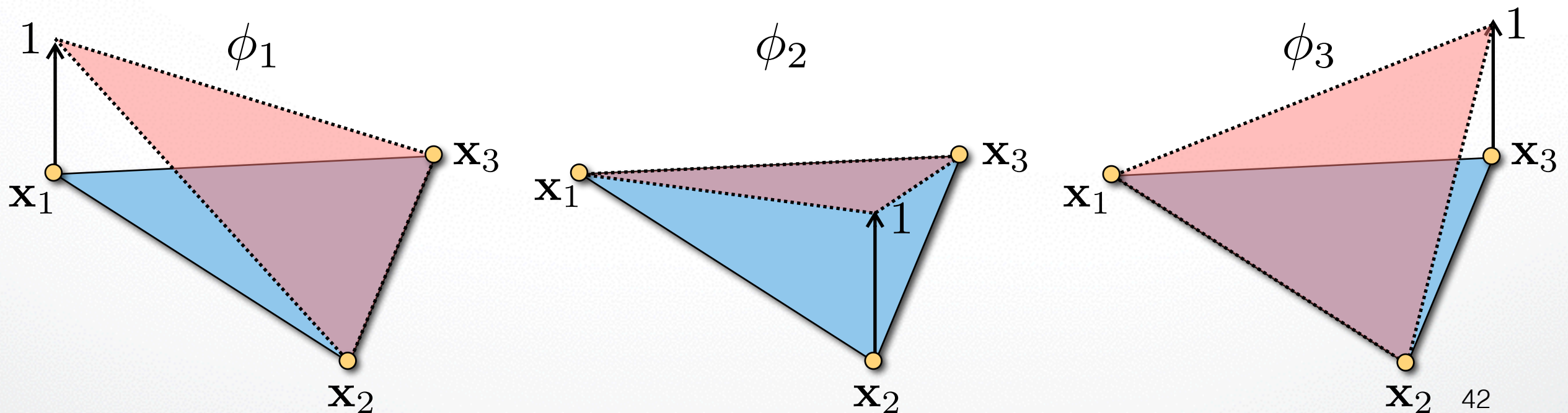
Gradient-Based Editing

- Consider piecewise linear **coordinate function**

$$\mathbf{p}(u, v) = \sum_{v_i} \mathbf{p}_i \cdot \phi_i(u, v)$$

- Its gradient is

$$\nabla \mathbf{p}(u, v) = \sum_{v_i} \mathbf{p}_i \cdot \nabla \phi_i(u, v)$$



Gradient-Based Editing

- Consider piecewise linear ***coordinate function***

$$\mathbf{p}(u, v) = \sum_{v_i} \mathbf{p}_i \cdot \phi_i(u, v)$$

- Its gradient is

$$\nabla \mathbf{p}(u, v) = \sum_{v_i} \mathbf{p}_i \cdot \nabla \phi_i(u, v)$$

- It is constant per triangle

$$\nabla \mathbf{p}|_{f_j} =: \mathbf{g}_j \in \mathbb{R}^{3 \times 3}$$

Gradient-Based Editing

- Gradient of coordinate function $\mathbf{p}$

$$\begin{pmatrix} \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_F \end{pmatrix} = \underbrace{\mathbf{G}}_{(3F \times V)} \begin{pmatrix} \mathbf{p}_1^T \\ \vdots \\ \mathbf{p}_V^T \end{pmatrix}$$

- Manipulate per-face gradients

$$\mathbf{g}_j \mapsto \mathbf{T}_j(\mathbf{g}_j)$$

Gradient-Based Editing

- Reconstruct mesh from new gradients
 - Overdetermined ($3F \times V$) system
 - Weighted least squares system
- ➡ Linear Poisson system $\Delta \mathbf{p}' = \text{div } \mathbf{T}(\mathbf{g})$

$$\text{div } \nabla = \Delta \quad \mathbf{G} \cdot \begin{pmatrix} \mathbf{p}'_1{}^T \\ \vdots \\ \mathbf{p}'_V{}^T \end{pmatrix} = \text{div} \begin{pmatrix} \mathbf{T}_1(\mathbf{g}_1) \\ \vdots \\ \mathbf{T}_F(\mathbf{g}_F) \end{pmatrix}$$

Laplacian-Based Editing

- Manipulate Laplacians field of a surface

$$\mathbf{l} = \Delta(\mathbf{p}) \quad , \quad \mathbf{l} \mapsto \mathbf{T}(\mathbf{l})$$

- Find surface whose Laplacian is (close to) $\delta' = \mathbf{T}(\mathbf{l})$

$$\mathbf{p}' = \operatorname{argmin}_{\mathbf{p}} \int_{\Omega} \|\Delta \mathbf{p} - \mathbf{T}(\mathbf{l})\|^2 \, du dv$$

- Variational calculus yields Euler-Lagrange PDE

$$\Delta^2 \mathbf{p}' = \Delta \mathbf{T}(\mathbf{l})$$

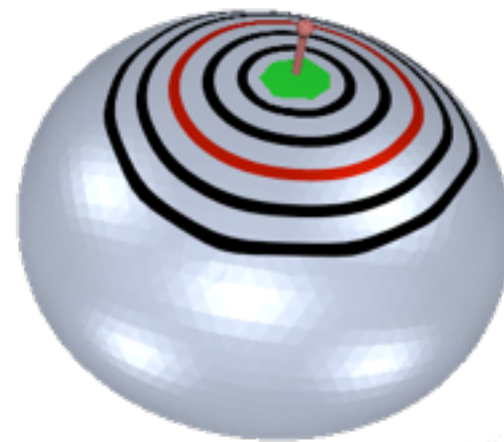
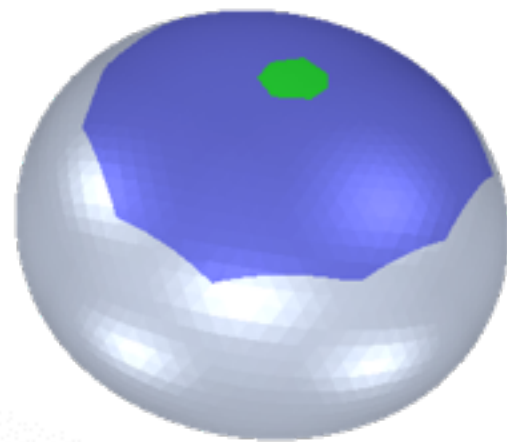
soft constraints

Differential Coordinates

- Which differential coordinate δ_i ?
 - Gradients
 - Laplacians
 - ...
- **How to get local transformations $T_i(\delta_i)$?**
 - Smooth propagation
 - Implicit optimization
 - ...

Smooth Propagation

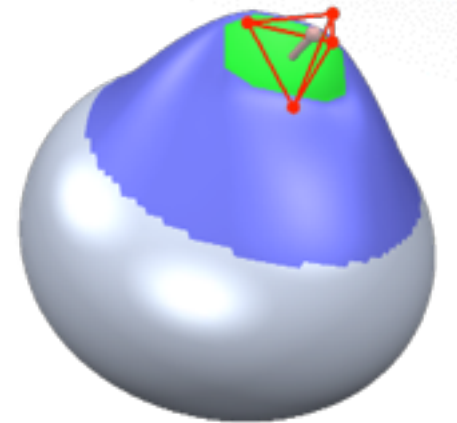
1. Compute handle's deformation gradient
2. Extract rotation and scale/shear components
3. Propagate damped rotations over ROI



Deformation Gradient

- Handle has been transformed affinely

$$\mathbf{T}(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{t}$$



- Deformation gradient is

$$\nabla \mathbf{T}(\mathbf{x}) = \mathbf{A}$$

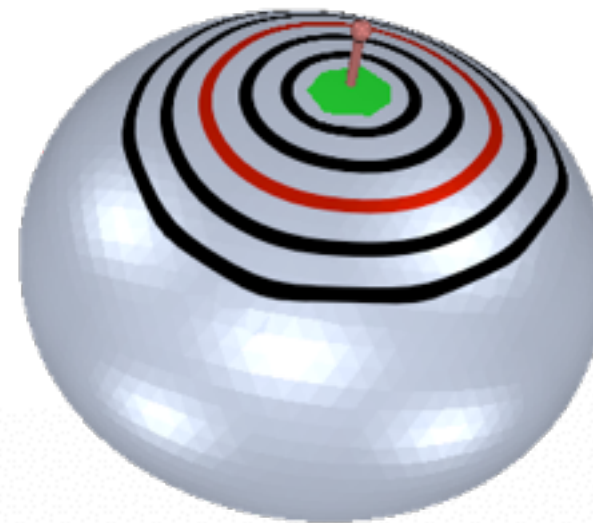
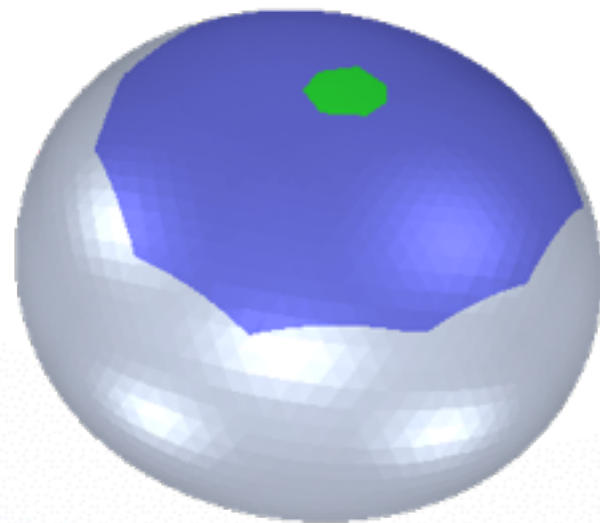
- Extract rotation $\mathbf{R}$ and scale/shear $\mathbf{S}$

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \Rightarrow \mathbf{R} = \mathbf{U}\mathbf{V}^T, \mathbf{S} = \mathbf{V}\mathbf{\Sigma}\mathbf{V}^T$$

SVD

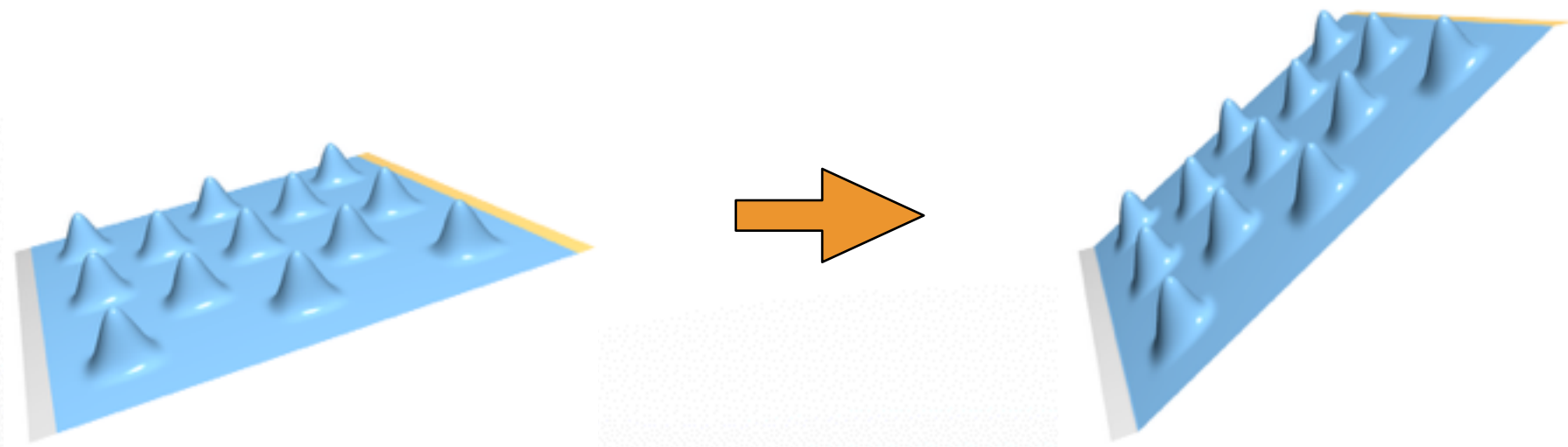
Smooth Propagation

- Construct smooth scalar field $[0,1]$
 - $s(\mathbf{x})=1$: Full deformation (handle)
 - $s(\mathbf{x})=0$: No deformation (fixed part)
 - $s(\mathbf{x})\in(0,1)$: Damp handle transformation (in between)



Limitations

- Differential coordinates work well for **rotations**
 - Represented by deformation gradient
- **Translations** don't change deformation gradient
 - Translations don't change differential coordinates
 - “*Translation insensitivity*”



Implicit Optimization

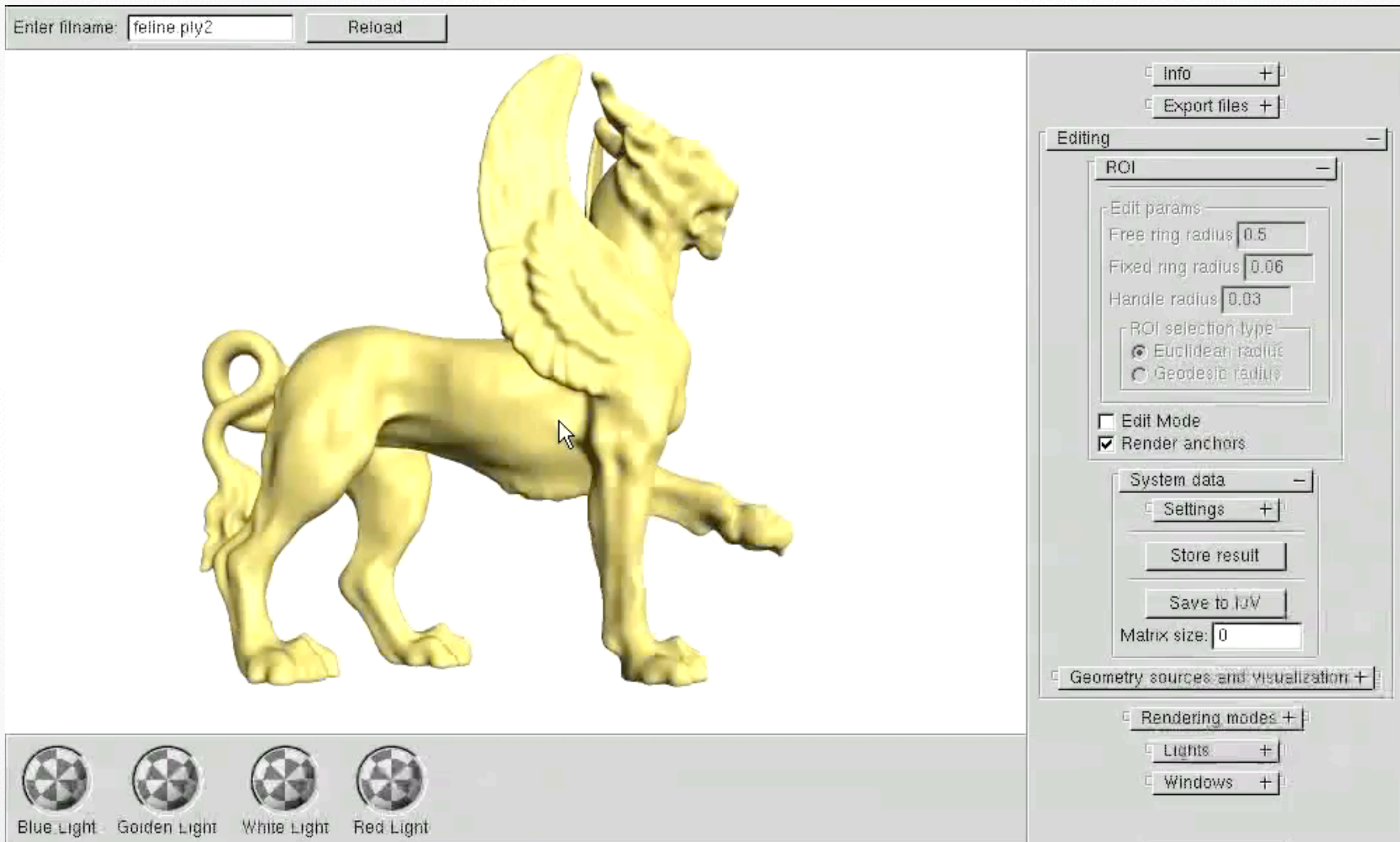
- Optimize for positions $\mathbf{p}_i'$ & transformations $\mathbf{T}_i$

$$\Delta^2 \begin{pmatrix} \vdots \\ \mathbf{p}'_i \\ \vdots \end{pmatrix} = \begin{pmatrix} \vdots \\ \Delta \mathbf{T}_i(\mathbf{l}_i) \\ \vdots \end{pmatrix} \iff \mathbf{T}_i(\mathbf{p}_i - \mathbf{p}_j) = \mathbf{p}'_i - \mathbf{p}'_j$$

- Linearize rotation/scale $\rightarrow$ one linear system

$$\mathbf{R}\mathbf{x} \approx \mathbf{x} \mathbf{T}_i(\mathbf{r} \times \mathbf{x}) \mathbf{r} = \begin{pmatrix} s & -r_3 & r_2 \\ r_3 & -r_1 & -r_2 \\ -r_2 & r_1r_2 & s_1 \end{pmatrix} \mathbf{x}$$

Laplacian Surface Editing



Connection to Shells?

- Neglect local transformations $\mathbf{T}_i$ for a moment...

$$\int \|\Delta \mathbf{p}' - \mathbf{l}\|^2 \rightarrow \min \longrightarrow \Delta^2 \mathbf{p}' = \Delta \mathbf{l}$$

- Basic formulations equivalent!
- Differ in detail preservation
 - Rotation of Laplacians
 - Multi-scale decomposition

$$\begin{array}{c} \mathbf{p}' = \mathbf{p} + \mathbf{d} \\ \mathbf{l} = \Delta \mathbf{p} \\ \downarrow \end{array}$$

$$\Delta^2(\mathbf{p} + \mathbf{d}) = \Delta^2 \mathbf{p}$$

$$\downarrow$$

$$\int \|\mathbf{d}_{uu}\|^2 + 2 \|\mathbf{d}_{uv}\|^2 + \|\mathbf{d}_{vv}\|^2 \rightarrow \min \longleftarrow \Delta^2 \mathbf{d} = 0$$

Linear Surface-Based Deformation

- Shell-Based Deformation
- Multiresolution Deformation
- Differential Coordinates

Projects

Geometry Processing Project

Goal

- Small research project
- 1 week for project proposal, **deadline March 31**
 - **choose between 3 options: A,B, or C**
- 1 month for project, **deadline April 24**
- group, size up to 2
- contributes **30%** to the final grade.
- send to olszewsk@usc.edu

Scope

A) For the disciplined

- Deformation Project, we will provide a framework
- You will implement a surface-based linear deformation algorithm (bending minimizing deformation).

B) For the creative [+10 points]

- Imagine an interesting topic around geometry processing or related to your PhD research or something you always wanted to do, and **write a proposal**.
- If it gets approved, you are good to go.

C) For the bad ass [+10 points]

- Implement a Siggraph, SGP, SCA, or Eurographics Paper.
- Geometry processing related of course ;-)

Project Submission

Deliverables for A)

- Source Code, Binary, Data
- Text files describing the project, how to run it.

Deliverables for B) and C)

- Short Presentation will be held April 22 and 24th (length TBD)
- Video / Figures
- Documentation (pdf, doc, txt file): 2 or more pages, short paper style, be rigorous and organized, must include at least **abstract**, **methodology**, and **results**.

Project Proposal

Structure

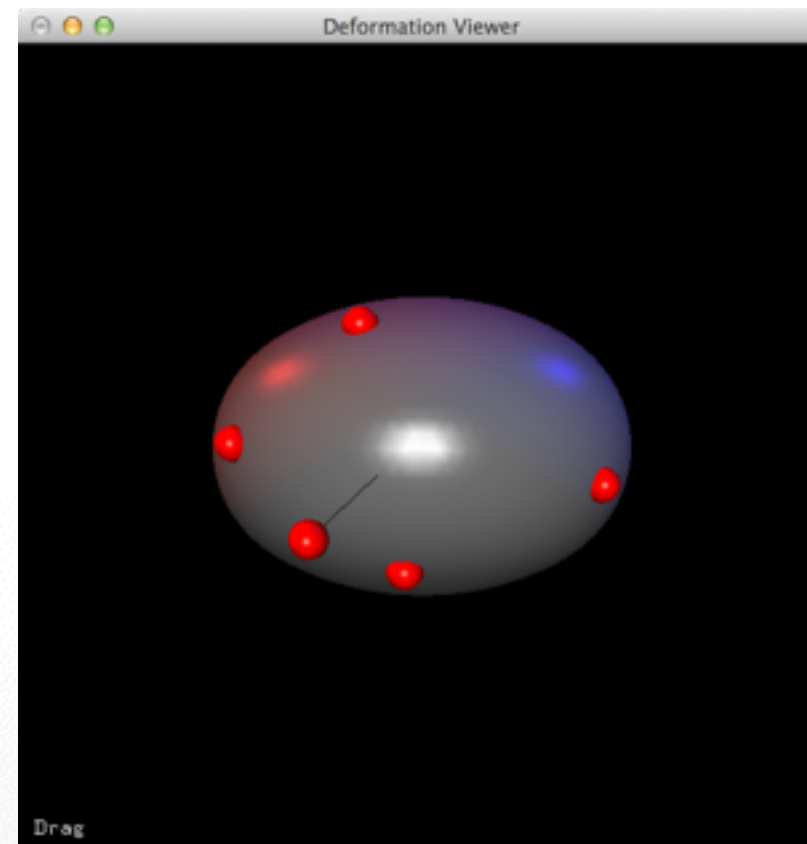
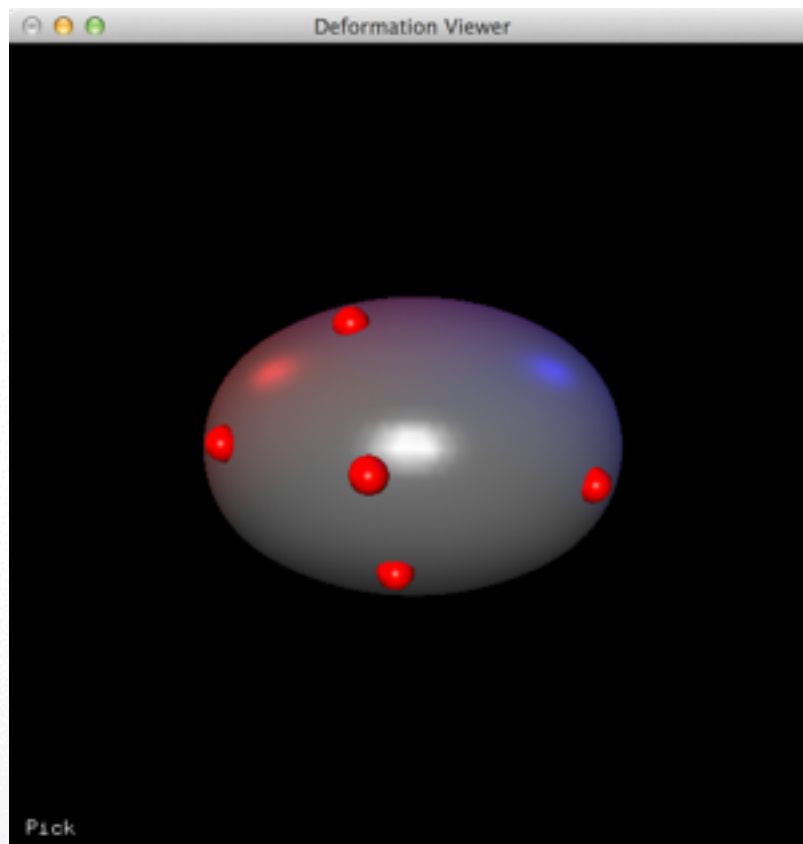
- Title
- Motivation
- Goal
- Proposed Method
- References

Format

- authors' names/student IDs
- 1-2 pages
- .doc, .pdf, .txt
- figures

Deformation Framework for A)

- Inherit from MeshViewer with user interface:
 - `'p'`: pick a handle
 - `'d'`: drag a handle (last one with starting code)
 - `'m'`: move the mesh



Deformation Framework for A)

- add handle picking code to
`DeformationViewer::mouse()`
- add deformation codes to
`DeformationViewer::deform_mesh()`
- add extra classes and files if needed
- **gmm** is provided to solve linear systems

Some ideas for B) or C)

- **registration**: articulated / deformable motions...
- **shape matching**: RANSAC, spin images, spherical harmonics...
- **Smoothing**: implicit surface fairing...
- **parameterization**: harmonic/conformal mapping...
- **remeshing**: anisotropic, quad mesh...
- **deformation**: As-rigid-as-possible, gradient-based...
- ...

Next Time

Non-Linear

Surface Deformations



<http://cs599.hao-li.com>

Thanks!

