12 - Mesh Deformation

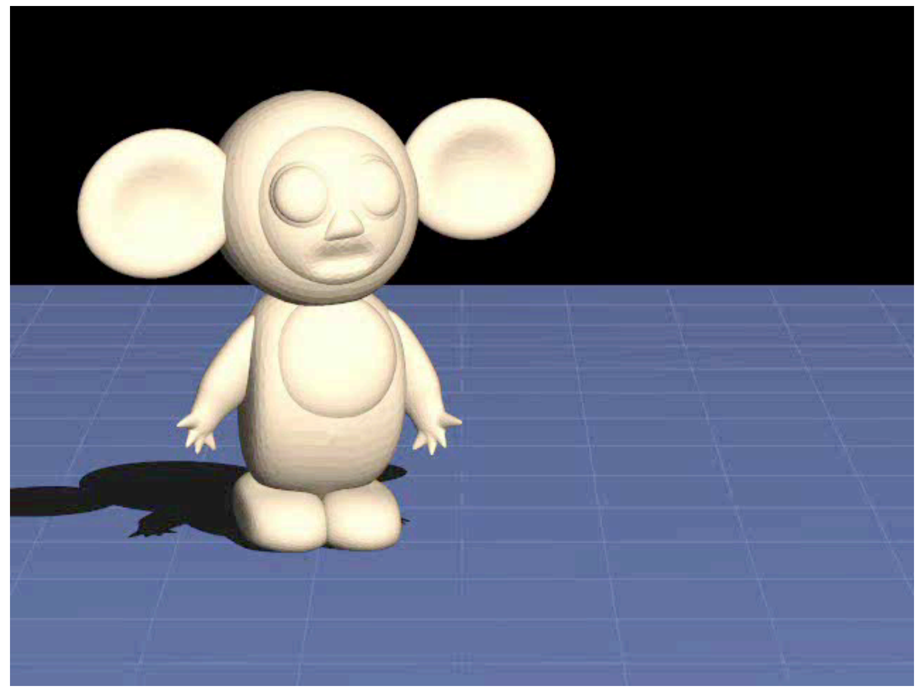
Acknowledgements: Daniele Panozzo, Olga Sorkine-Hornung

In this lecture

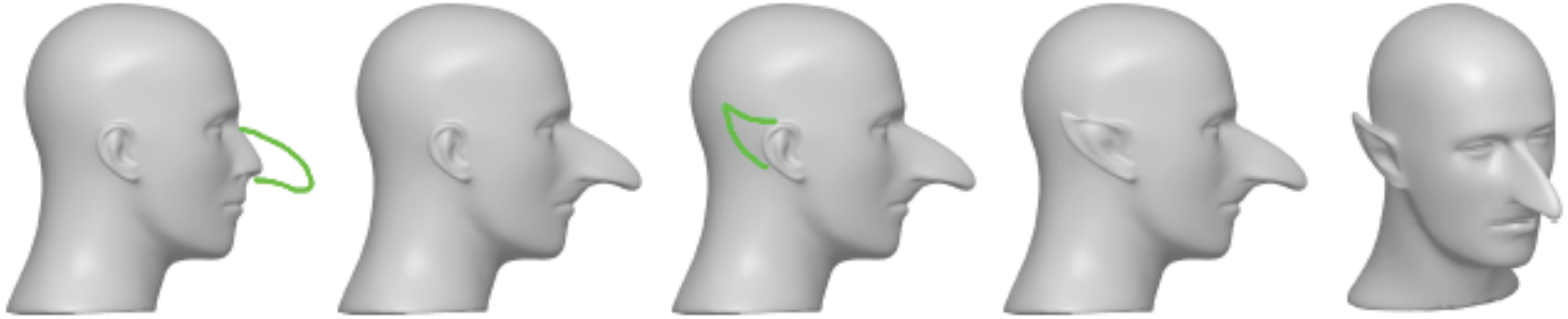
- Paradigms for shape deformation
- Simple linear algorithms

Why Shape Deformation?

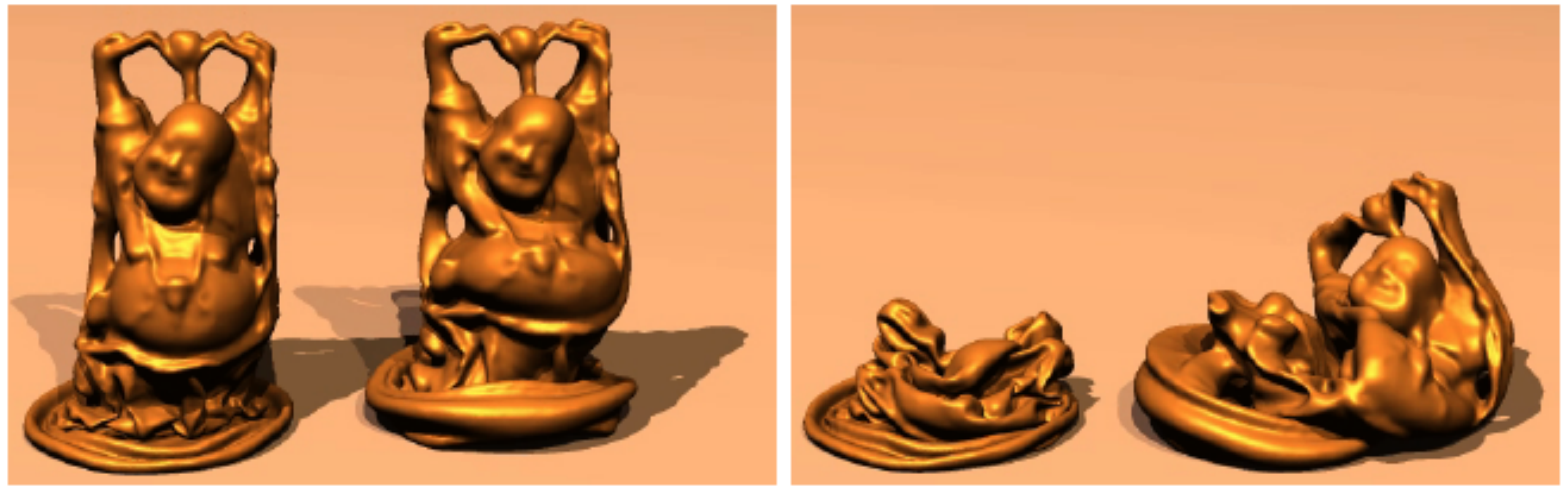
- Animation



- Editing



- Simulation

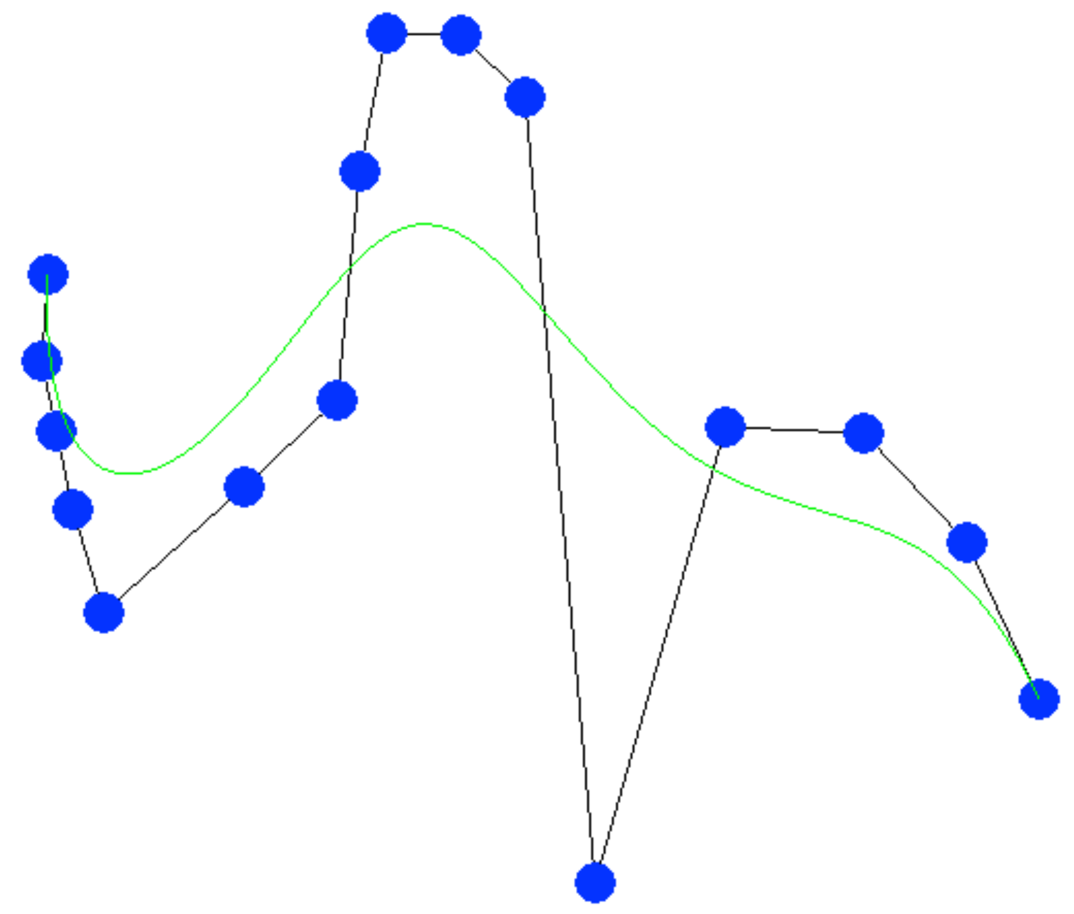


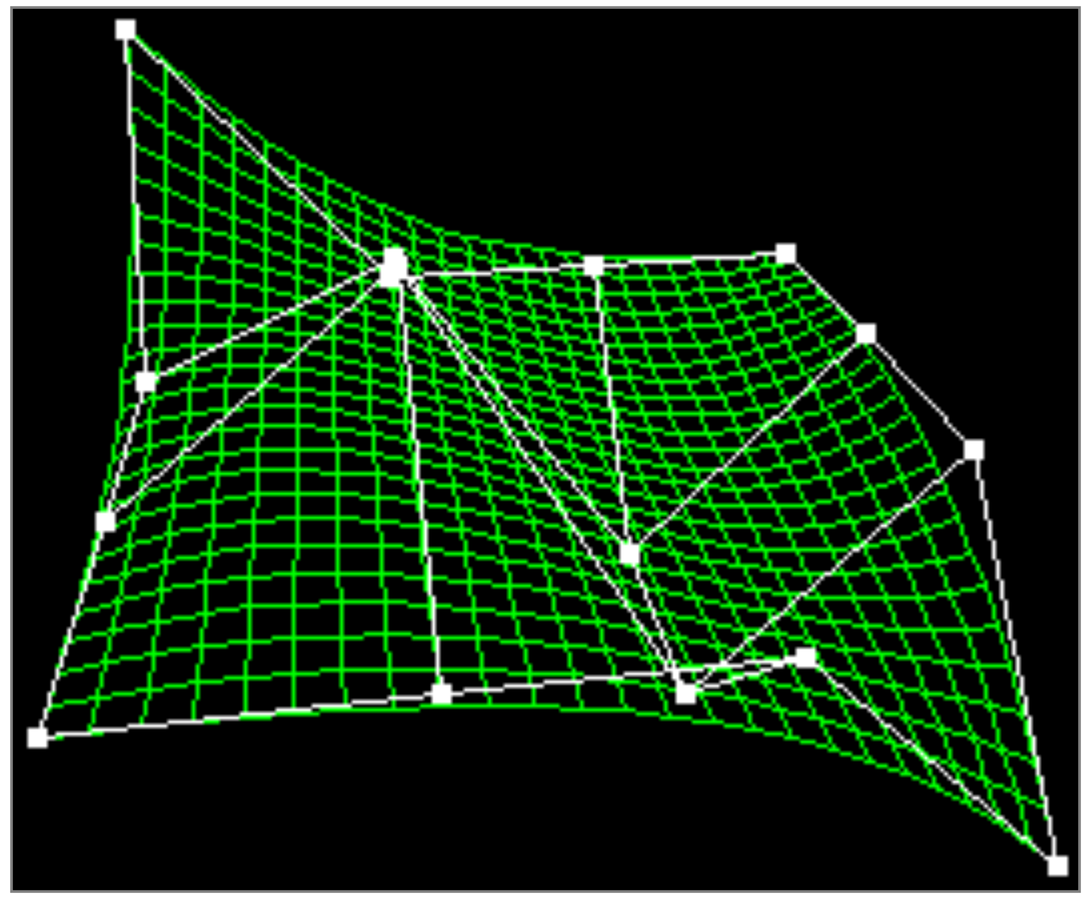
Parametric Curves and Surfaces

- Deformation by control point manipulation

- Built-in deformation mechanism

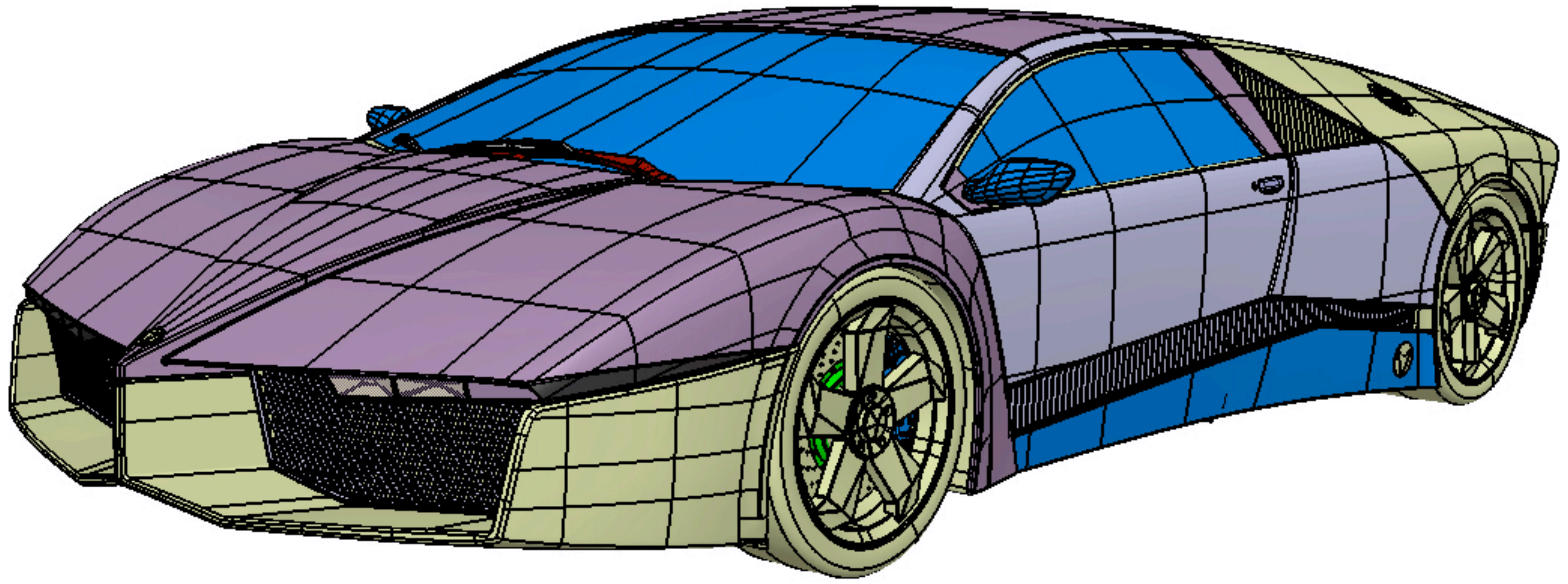
- Control structure is pre-set (can’t pull on arbitrary points)

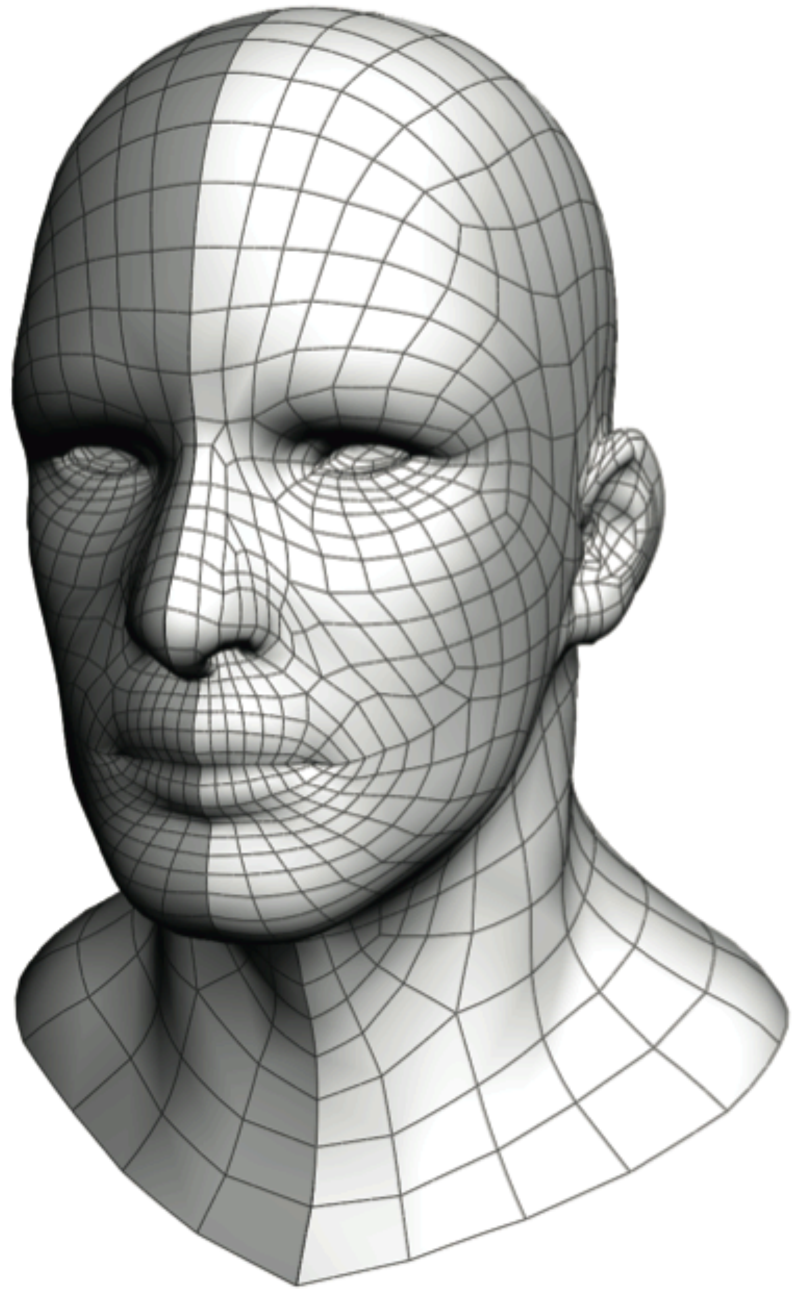




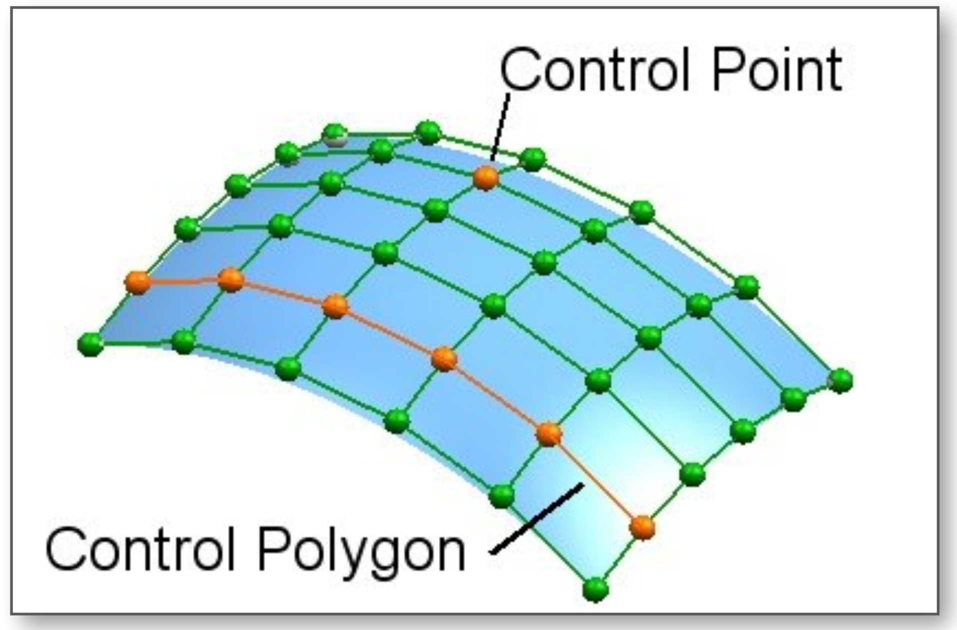
Parametric Curves and Surfaces

- Hard to change / adapt control structure to user needs
- Hard to experiment, need a precise idea of what will be modeled
- See them in the next lectures!

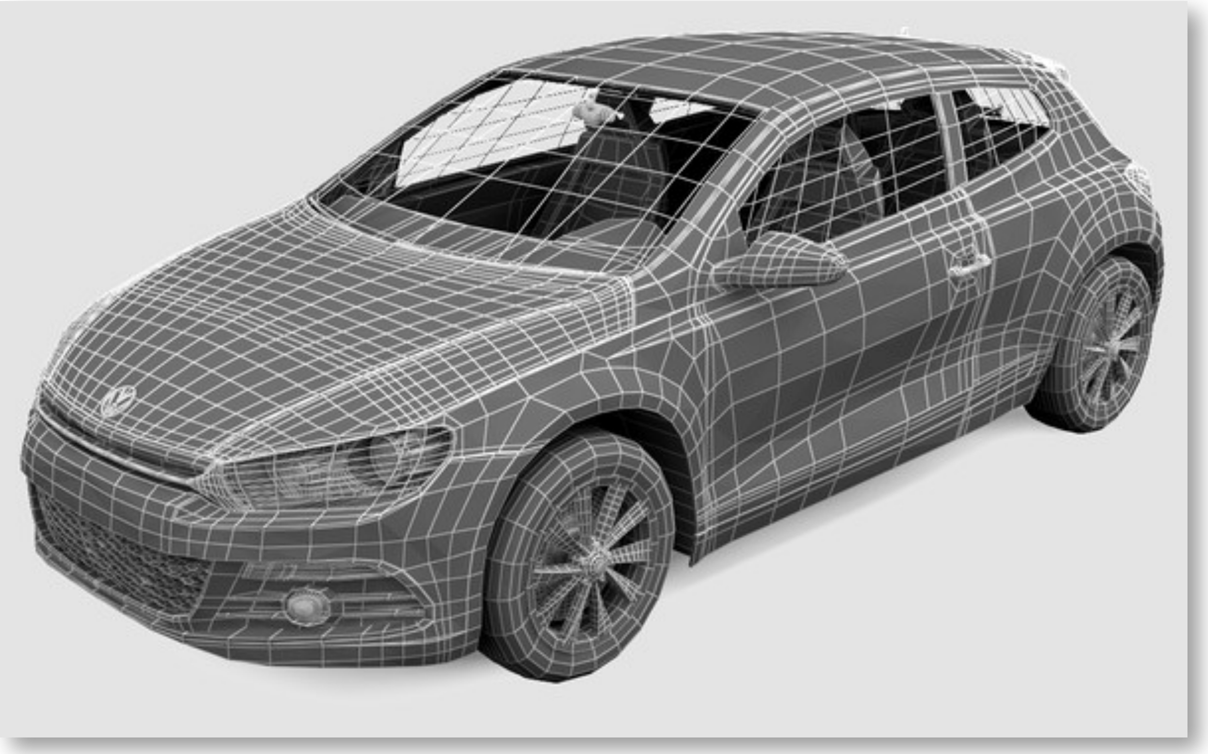




Traditional CAD vs Unstructured Meshes



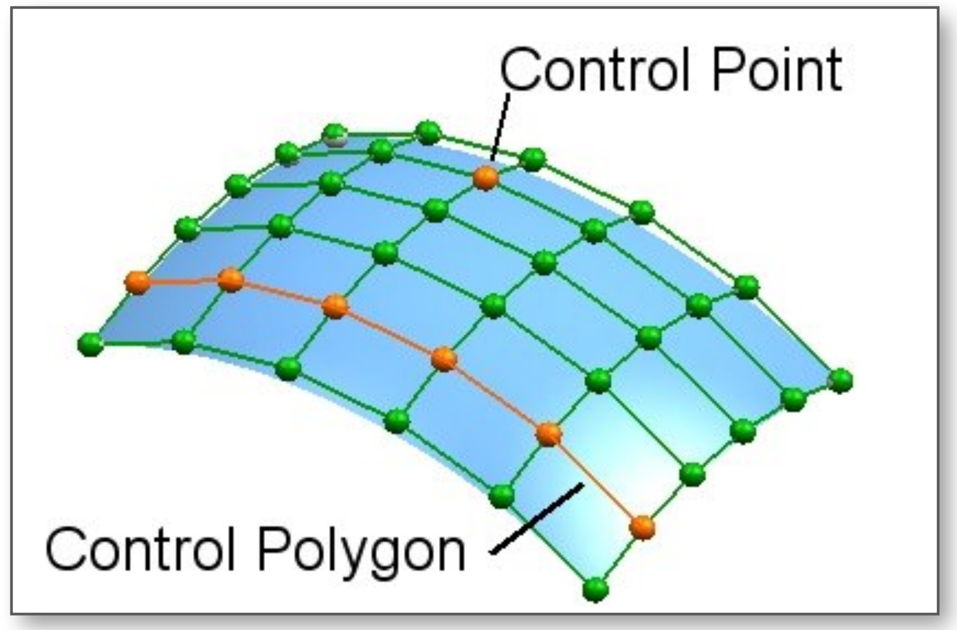
$$s(u,v) = \sum_{i,j} \mathbf{p}_{i,j} B_i(u) B_j(v)$$



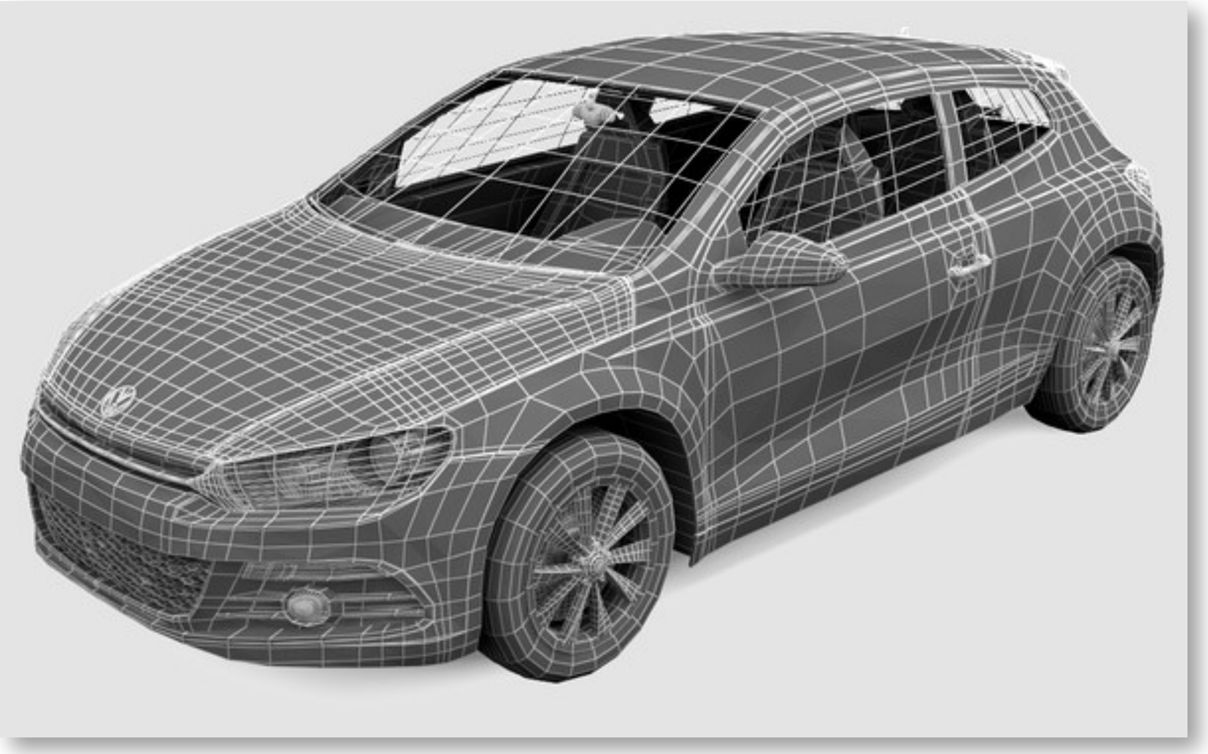
$$\min_{\mathbf{x}'} \int_{\mathcal{S}} \|\Delta_{\mathcal{S}}\mathbf{x}' - \delta_0\|^2$$
$$s.t.\ \mathbf{x}'|_{\mathcal{C}} = \mathbf{x}_{\text{fixed}}$$

images from Jacobson et al., SGP 2010

Traditional CAD vs Unstructured Meshes



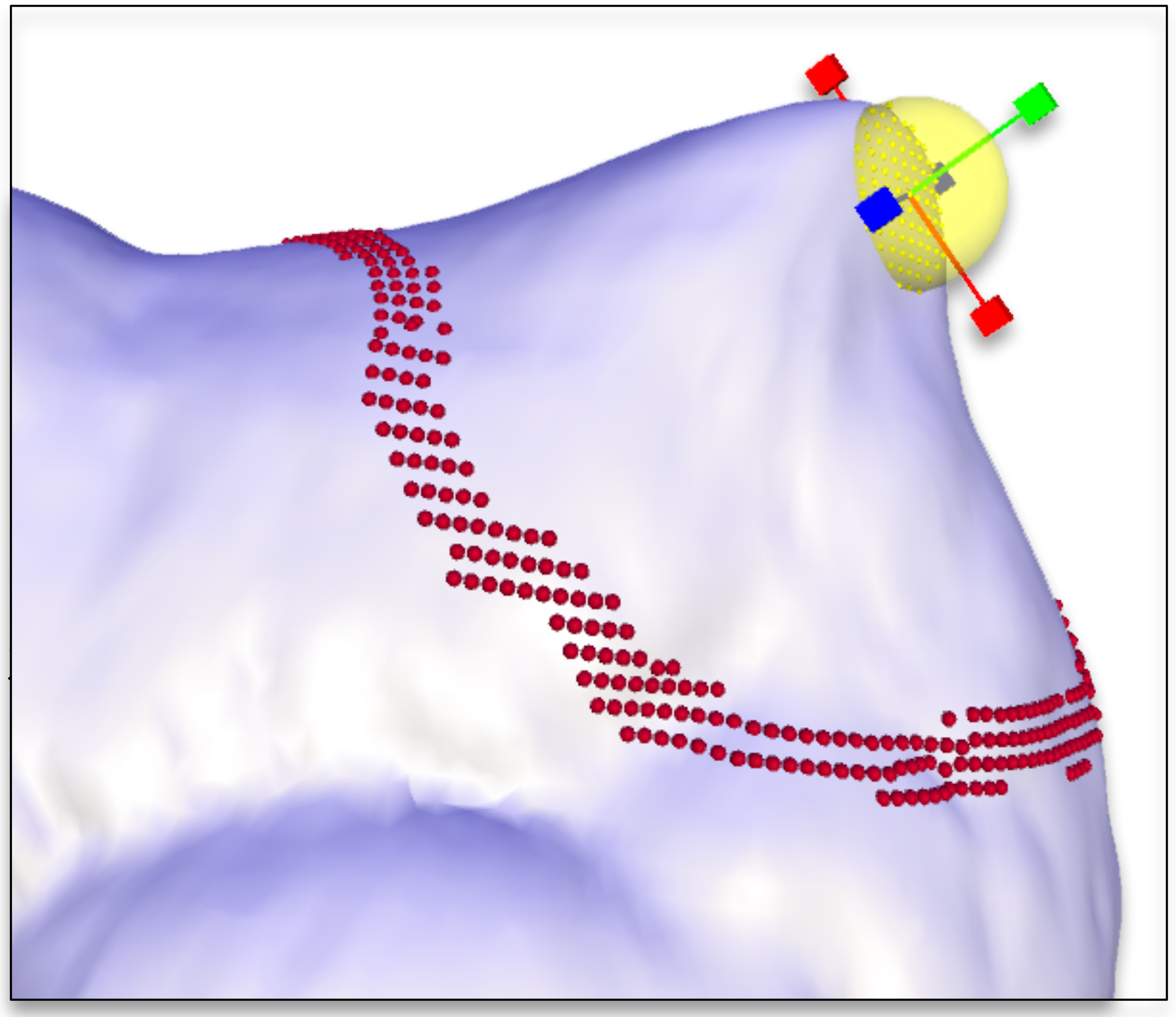
$$s(u,v) = \sum_{i,j} \mathbf{p}_{i,j} B_i(u) B_j(v)$$



$$\min_{\mathbf{x}'} \int_{\mathcal{S}} \left\| \Delta_{\mathcal{S}} \mathbf{x}' - \delta_0 \right\|^2$$
$$s.t.\ \mathbf{x}'|_{\mathcal{C}} = \mathbf{x}_{\text{fixed}}$$

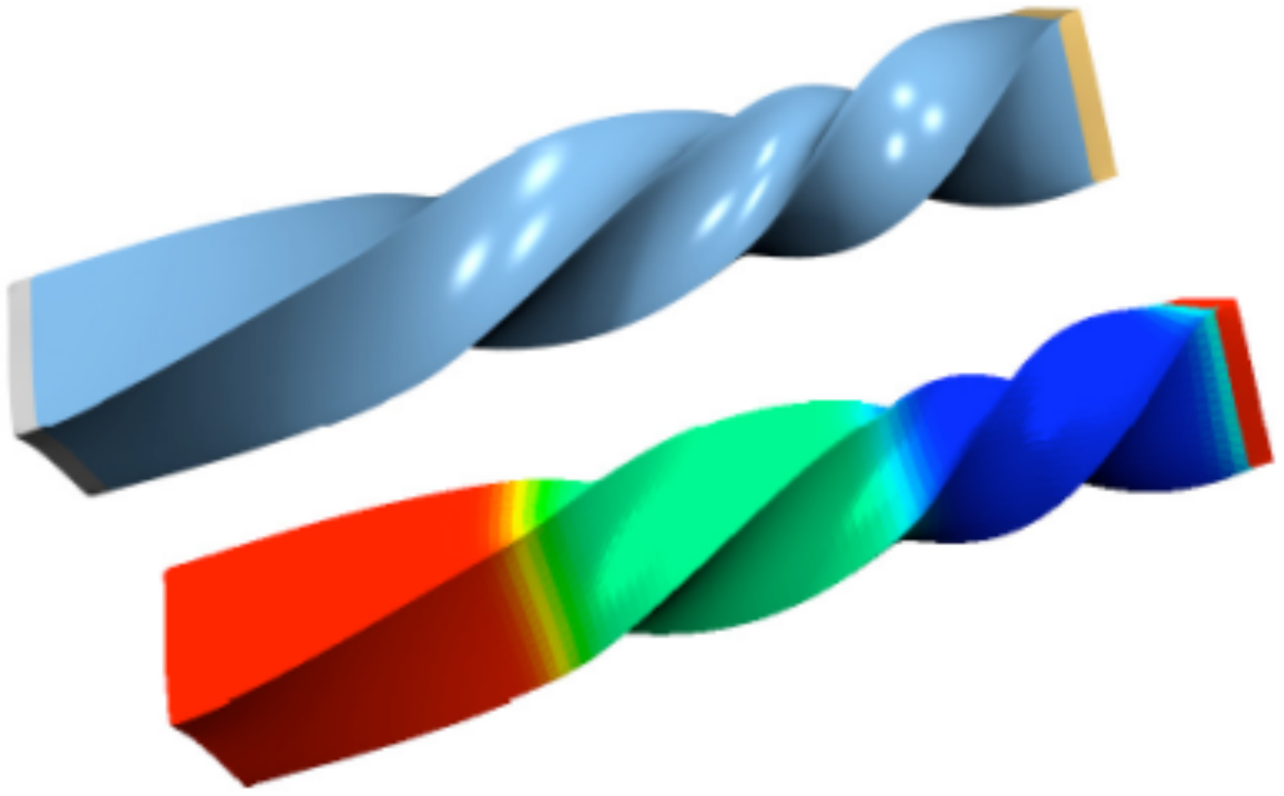
Mesh Deformation

- Naïve method: dragging single vertices
  - Cumbersome
  - Hard to control
- Smarter:
  - Introduce a small set of deformation handles
    - Makes deformation/editing easier
    - Introduces a trade-off between degrees of freedom and deformation task
  - Create a small set of control parameters
    - Affine transformations



Deformation: Common Paradigms

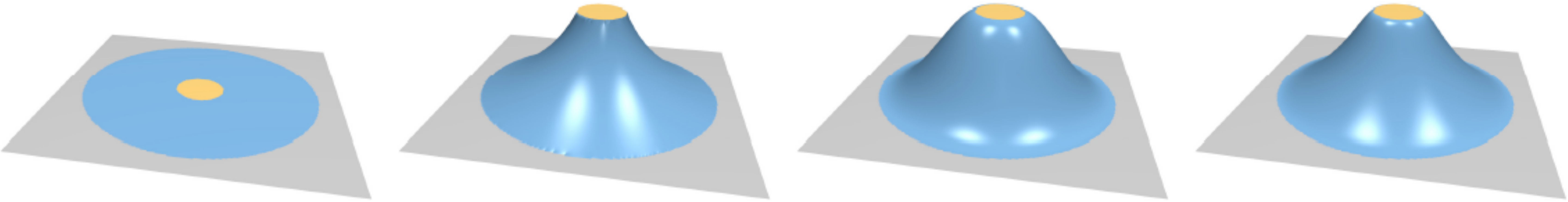
- **Surface based deformation**
  - Optimization on the surface
  - Physically motivated: variants of elastic energy minimization

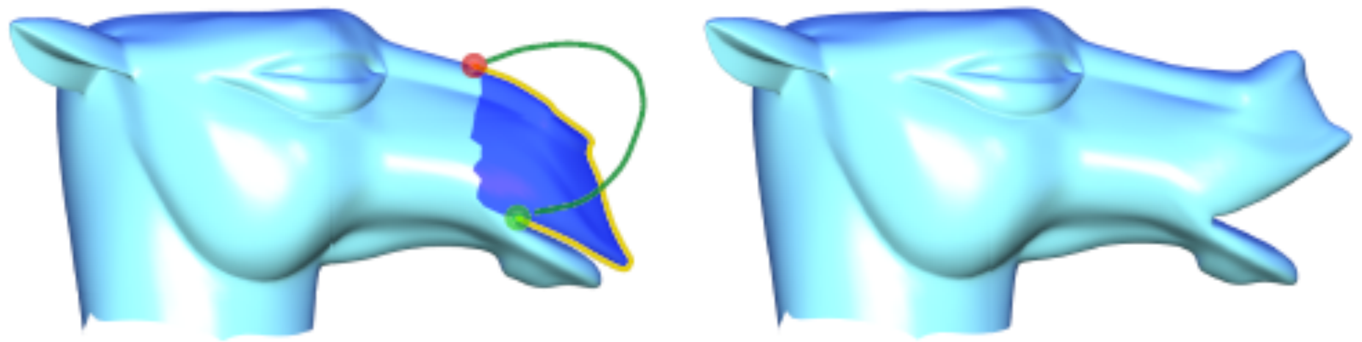


- **Space deformation**
  - Deforms some 2D/3D space using a *cage*
  - Deformation propagation to all points in the space
  - Independent of shape representation

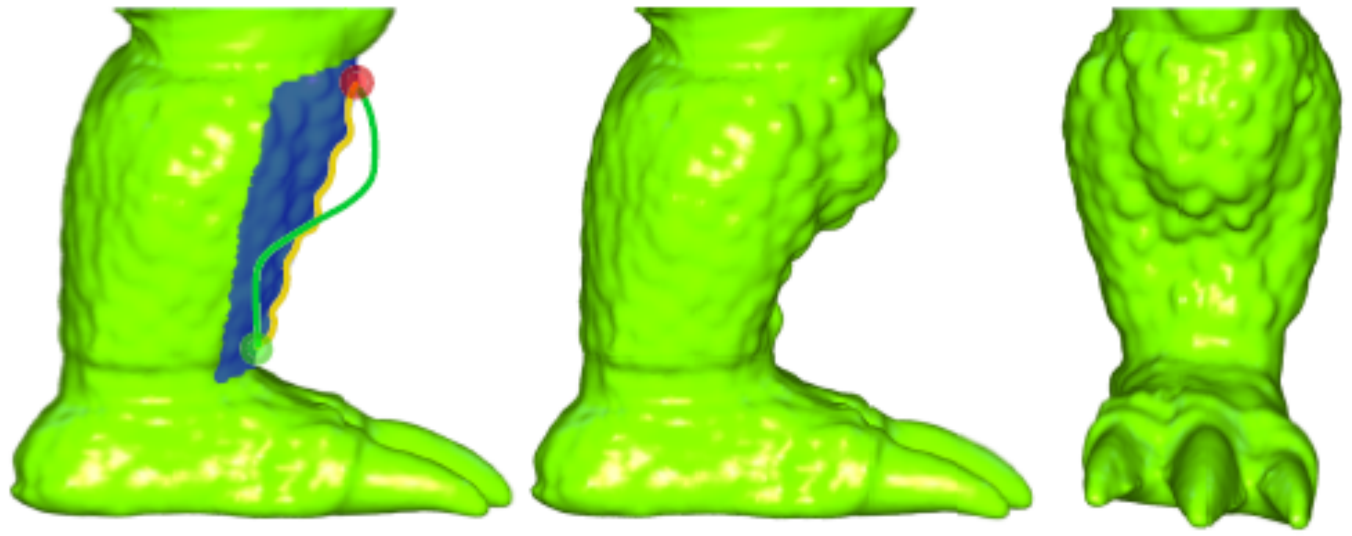
Surface-based Deformations: Examples

- Region of interest (ROI) + affine deformation handle with variable boundary continuity



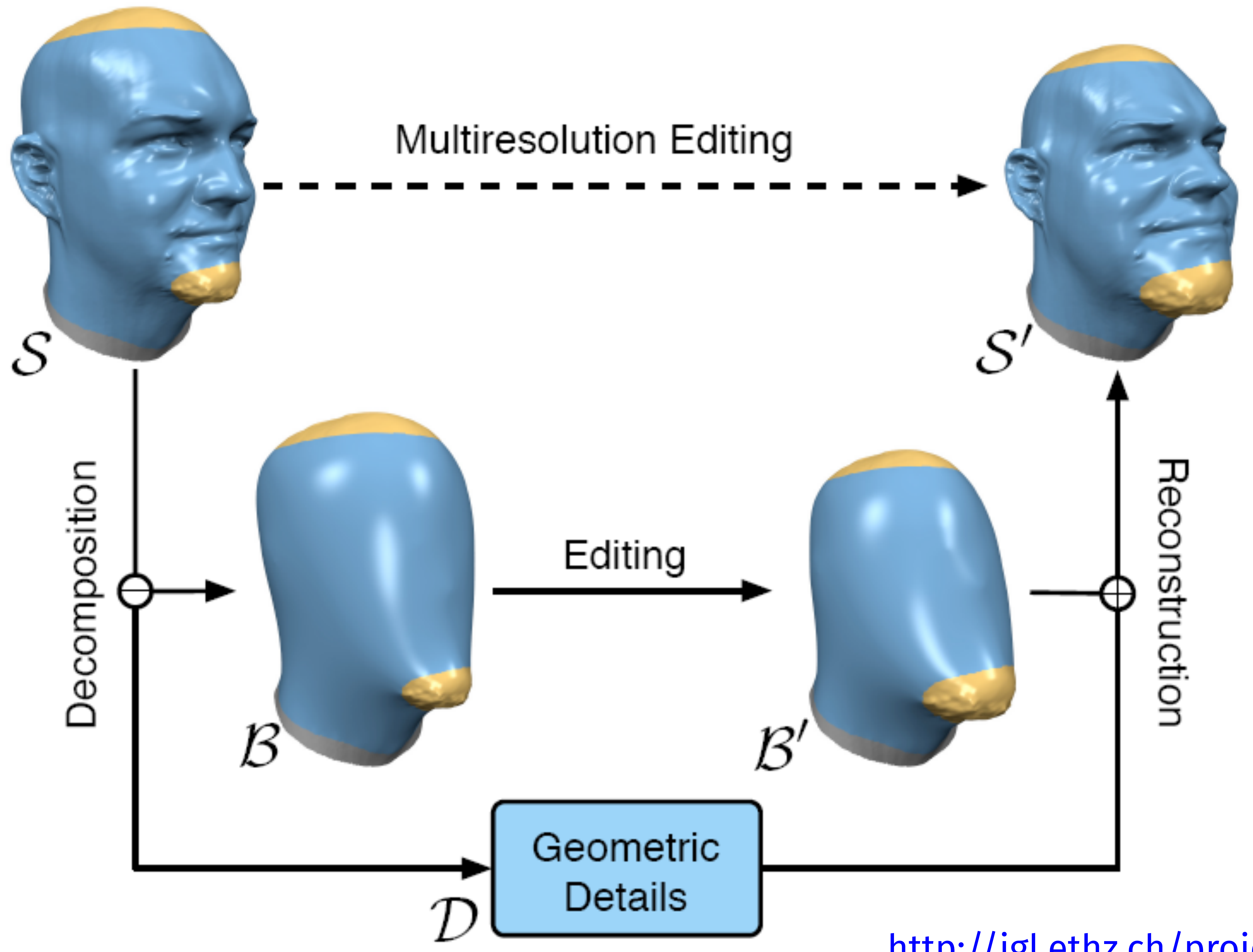


- Intuitive sketch-based deformation interfaces



Surface-based Deformations: Examples

- Multi-resolution mesh editing



http://igl.ethz.ch/projects/deformation-survey/

Surface-based Deformations: General Framework

- Find a mesh that optimizes some objective functional and satisfies modeling constraints

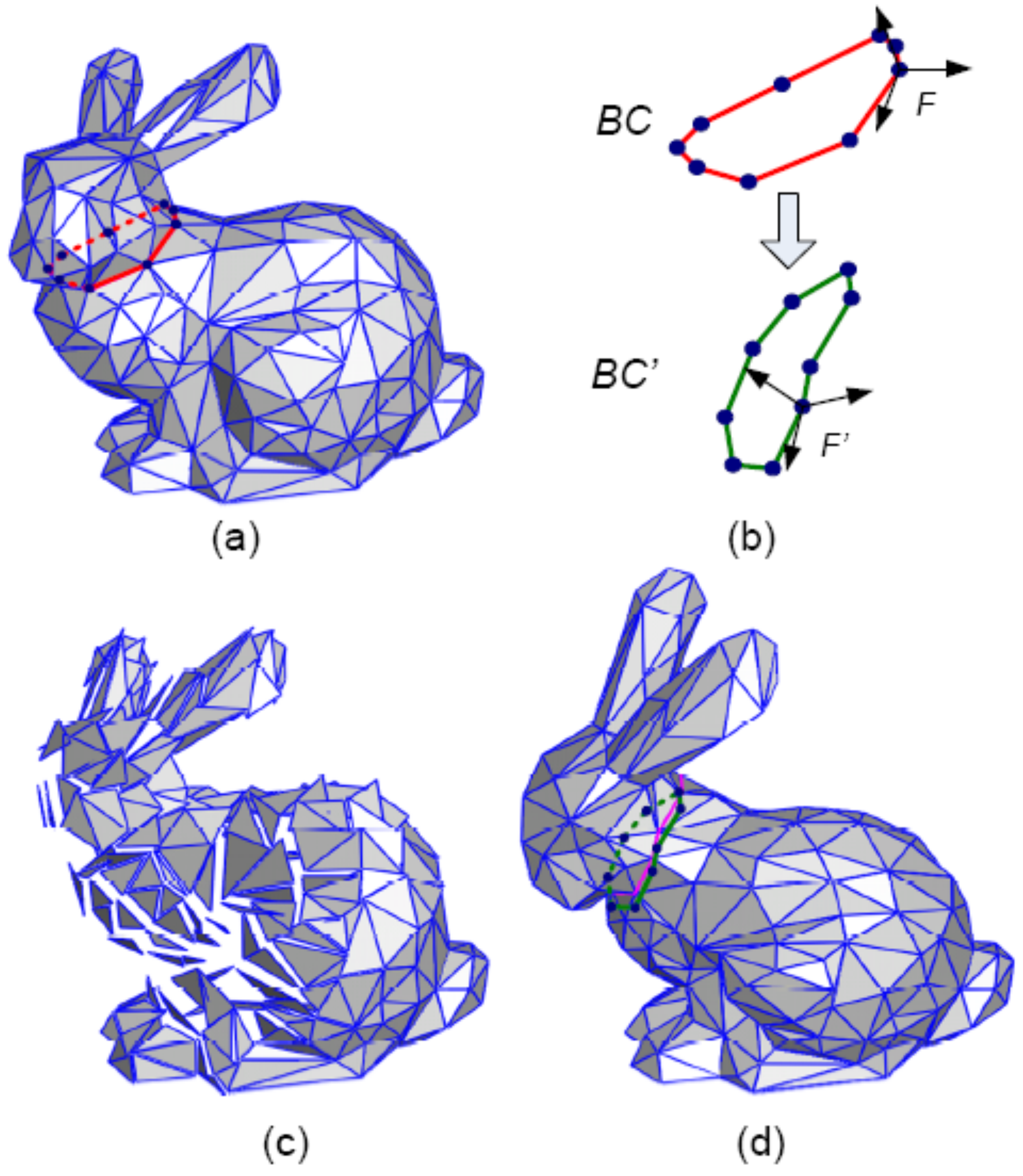
$$\mathbf{x}_{\mathrm{def}} = \operatorname*{argmin}_{\mathbf{x}'} E(\mathbf{x}') \quad s.t. \quad \mathbf{x}'_i = \mathbf{c}_i$$

Surface-based Deformations: Linear Methods

- Triangle gradient methods (2004-2005)

  http://dl.acm.org/citation.cfm?id=1015774

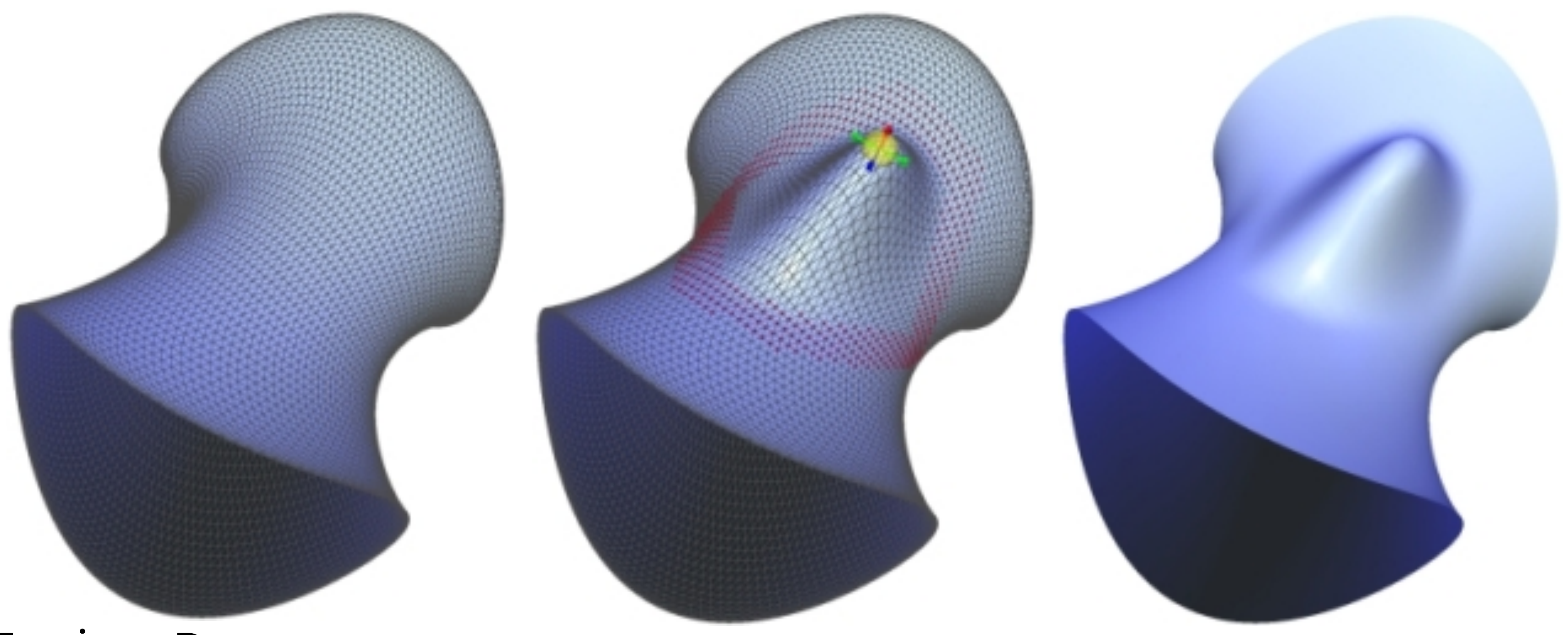
  http://www-ui.is.s.u-tokyo.ac.jp/~takeo/research/rigid/



- Laplacian surface editing (2004-2005)

  http://dl.acm.org/citation.cfm?id=1015772

  http://igl.ethz.ch/projects/Laplacian-mesh-processing/Laplacian-mesh-editing/

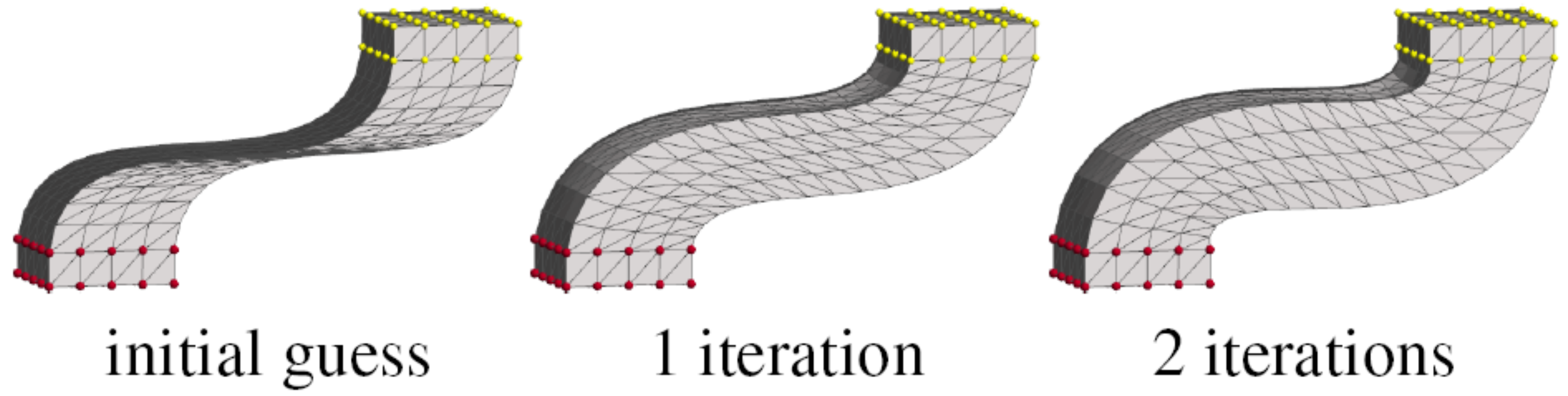


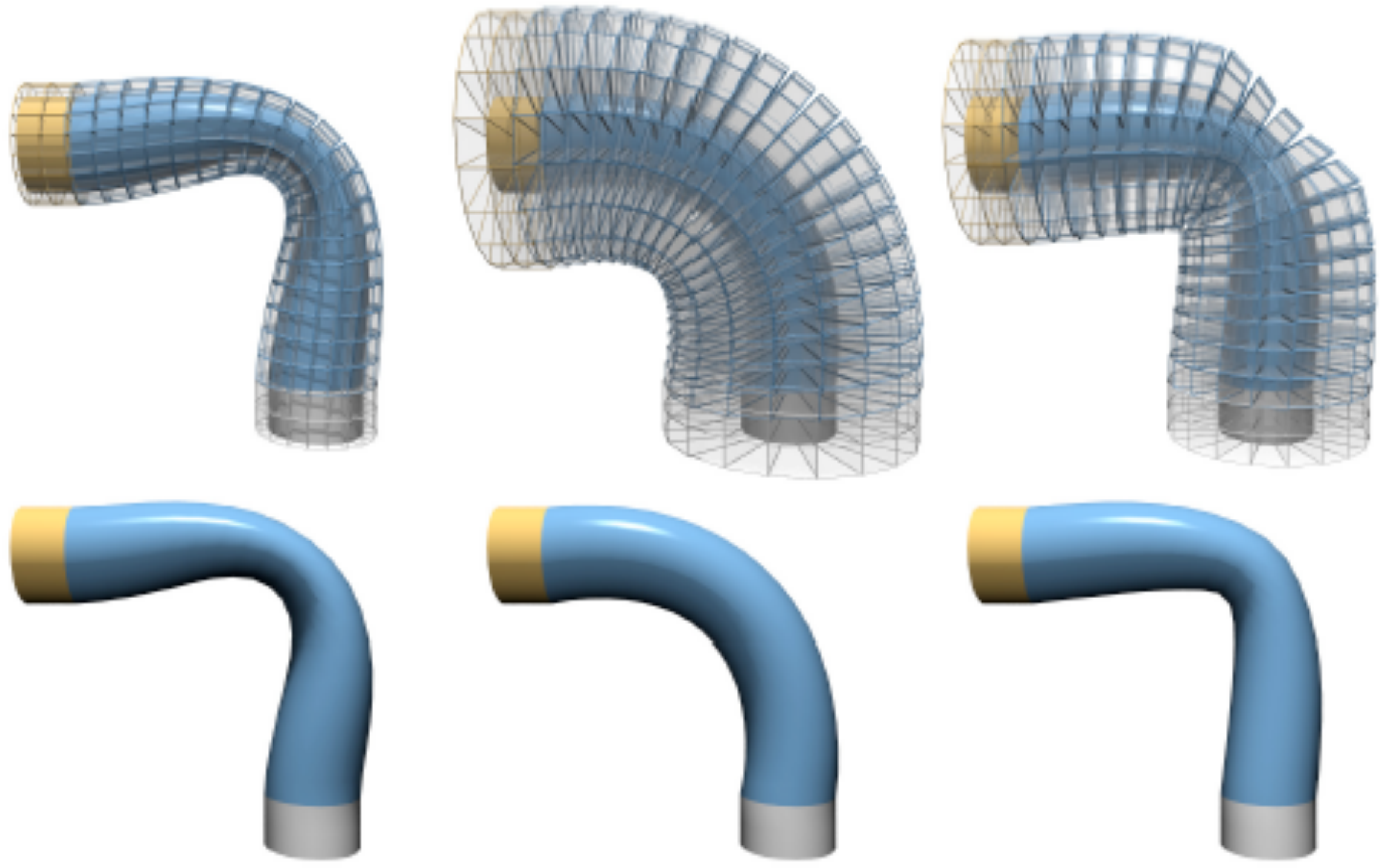
Surface-based Deformations: Nonlinear Methods

- As Rigid as Possible surface modeling

  http://igl.ethz.ch/projects/ARAP/

  http://dl.acm.org/citation.cfm?id=1778775
  http://igl.ethz.ch/projects/fast/
- PriMo
  http://dl.acm.org/citation.cfm?id=1281959
- Mesh Puppetry
  http://dl.acm.org/citation.cfm?id=1275808.1276479





Surface-based Deformations: Summary So Far

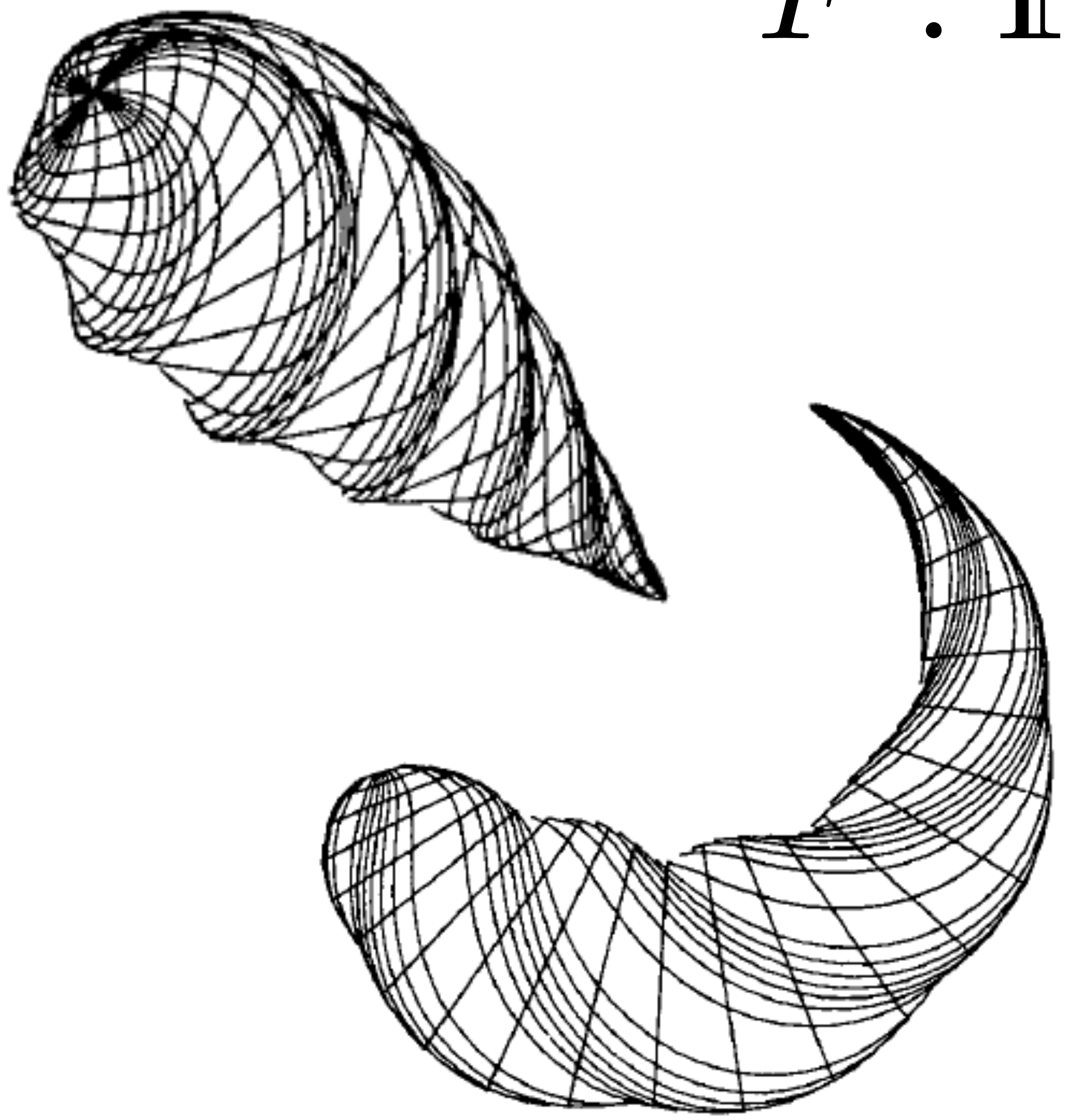
- Objective functional expressed in the mesh elements (vertices)
- Complexity depends on the mesh resolution
- Linear methods:
  - Solve a global linear system on the mesh
  - Usually suffer from some artifacts
- Nonlinear methods
  - Fewer artifacts but slower, and harder to implement

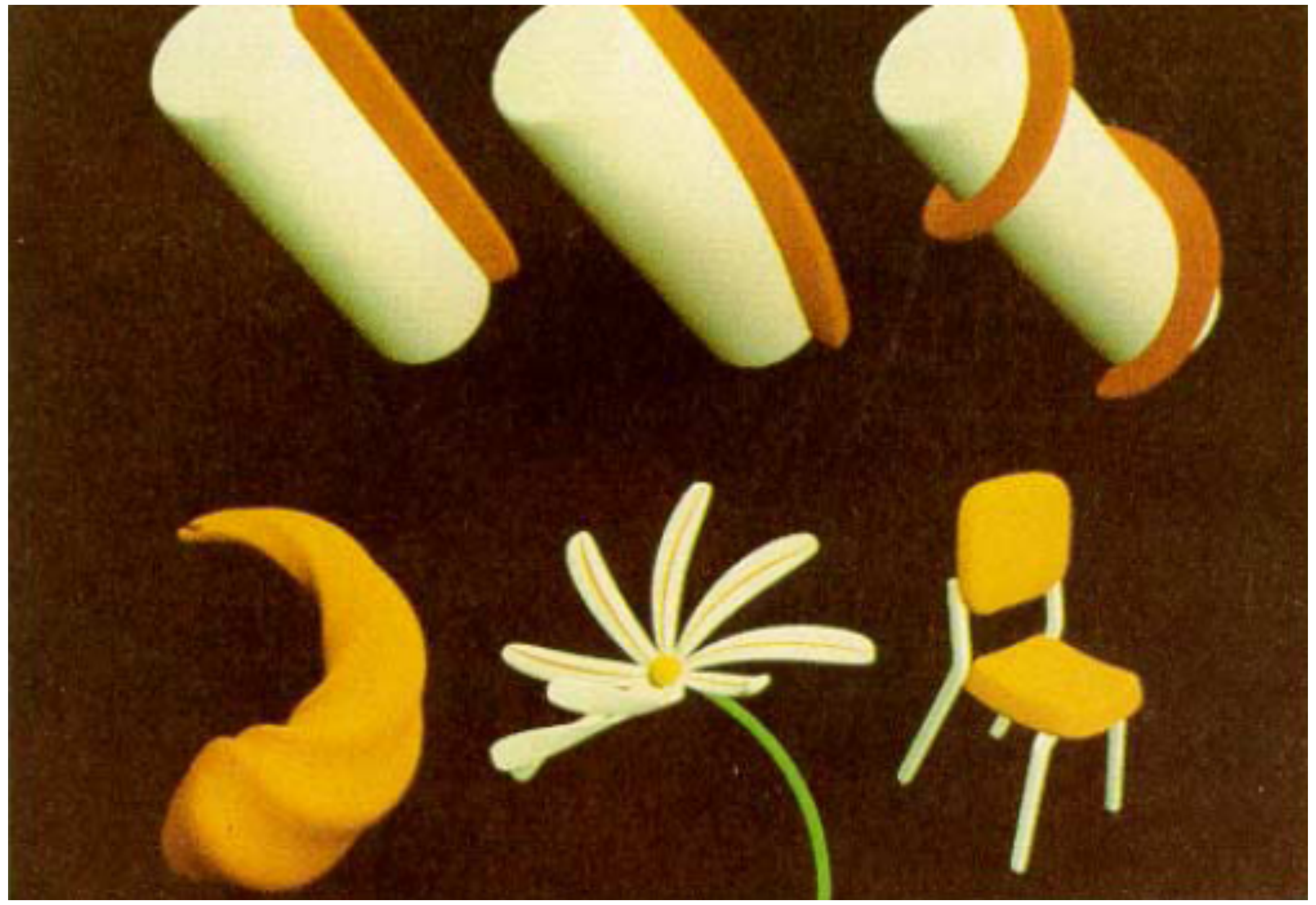
Space Deformations

Early seminal work in computer graphics

- Global and local deformation of solids [Barr 1984] http://dl.acm.org/citation.cfm?id=808573

$$F : \mathbb{R}^3 \to \mathbb{R}^3$$



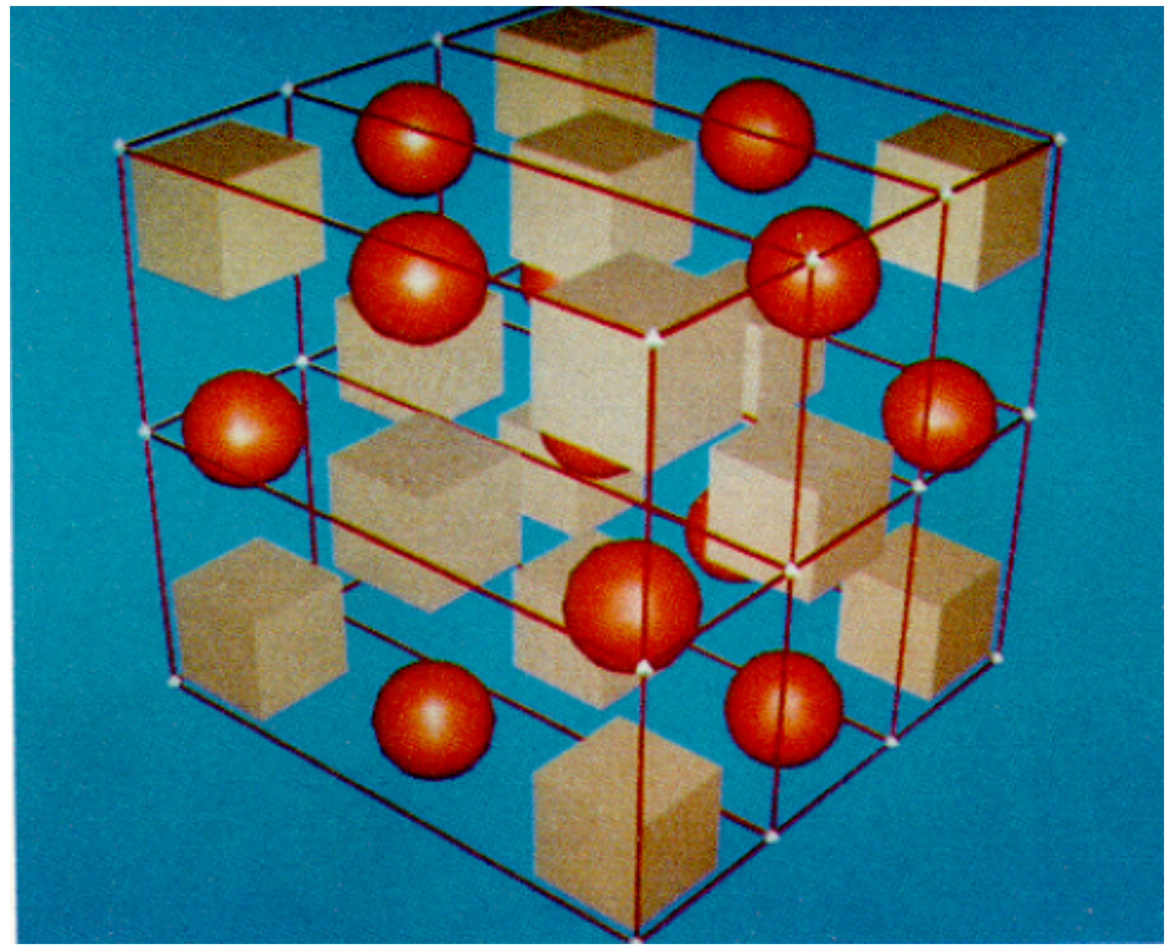


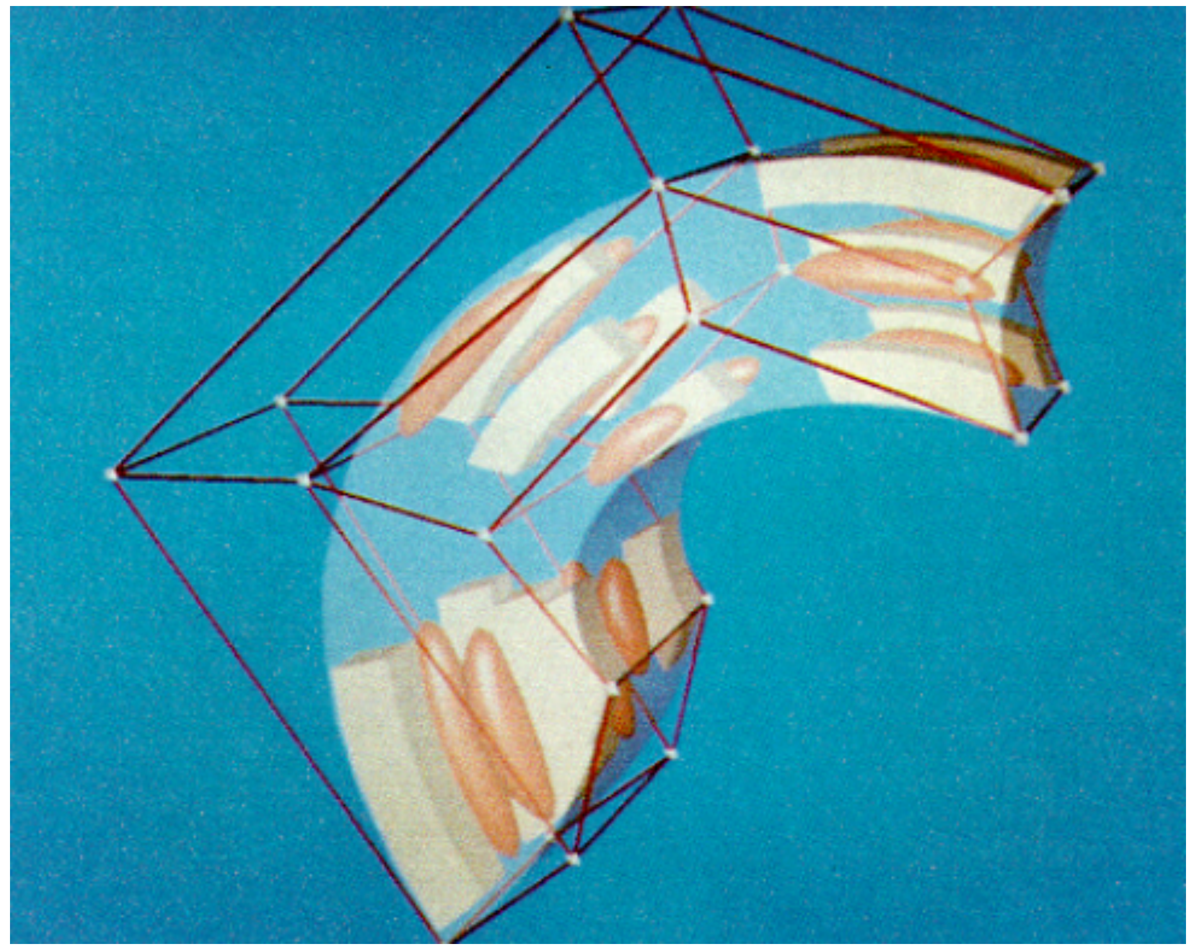
Space Deformations

Early seminal work in computer graphics

- Free form deformations
  [Sederberg and Parry 1986] http://dl.acm.org/citation.cfm?id=15903
  - Uses trivariate tensor product polynomial basis

$$f : \mathbb{R}^3 \to \mathbb{R}^3$$



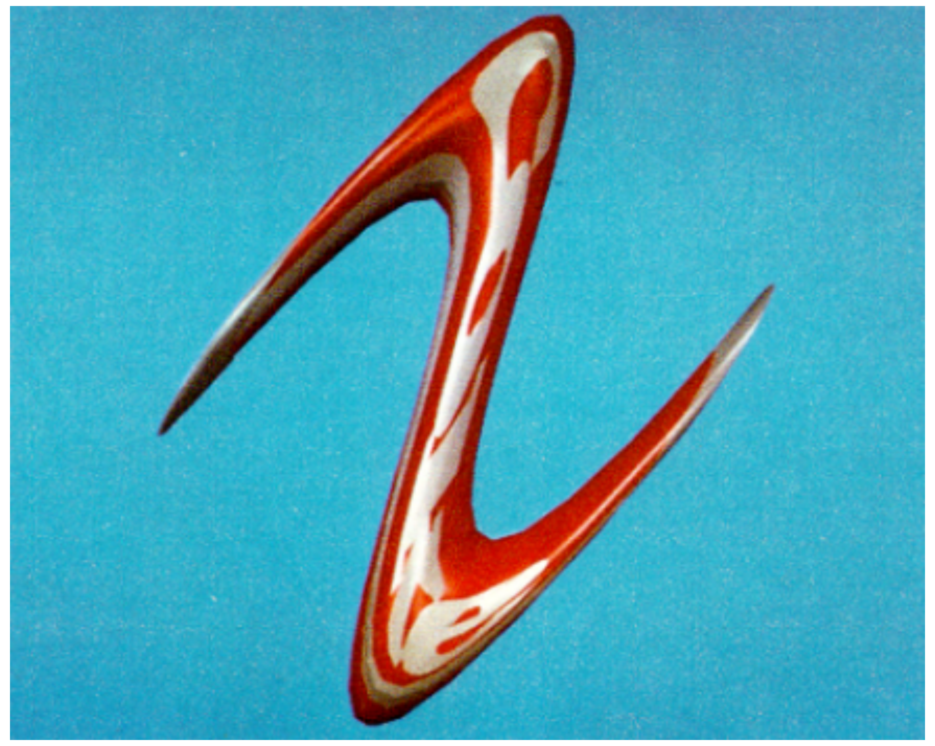


Space Deformations

Early seminal work in computer graphics

- Can be designed to be volume preserving

$$f : \mathbb{R}^3 \rightarrow \mathbb{R}^3$$



$$\mathbf{F}(x,y,z) = (F(x,y,z), G(x,y,z), H(x,y,z))$$

then the Jacobian is the determinant

$$Jac(\mathbf{F}) = \begin{vmatrix} \frac{\partial F}{\partial x} & \frac{\partial F}{\partial y} & \frac{\partial F}{\partial z} \\ \frac{\partial G}{\partial x} & \frac{\partial G}{\partial y} & \frac{\partial G}{\partial z} \\ \frac{\partial H}{\partial x} & \frac{\partial H}{\partial y} & \frac{\partial H}{\partial z} \end{vmatrix}$$

Space Deformations: Basic Idea

- Design a set of coordinates for all points in R^d w.r.t. the “cage” vertices
  - Each point $\mathbf{x}$ can be represented as a weighted sum of cage points $\mathbf{p}_i$

$$\mathbf{x} = \sum_{i=1}^{k} w_i(\mathbf{x})\,\mathbf{p}_i$$

  - When the cage changes, the coords stay the same, substitute the new cage geometry:

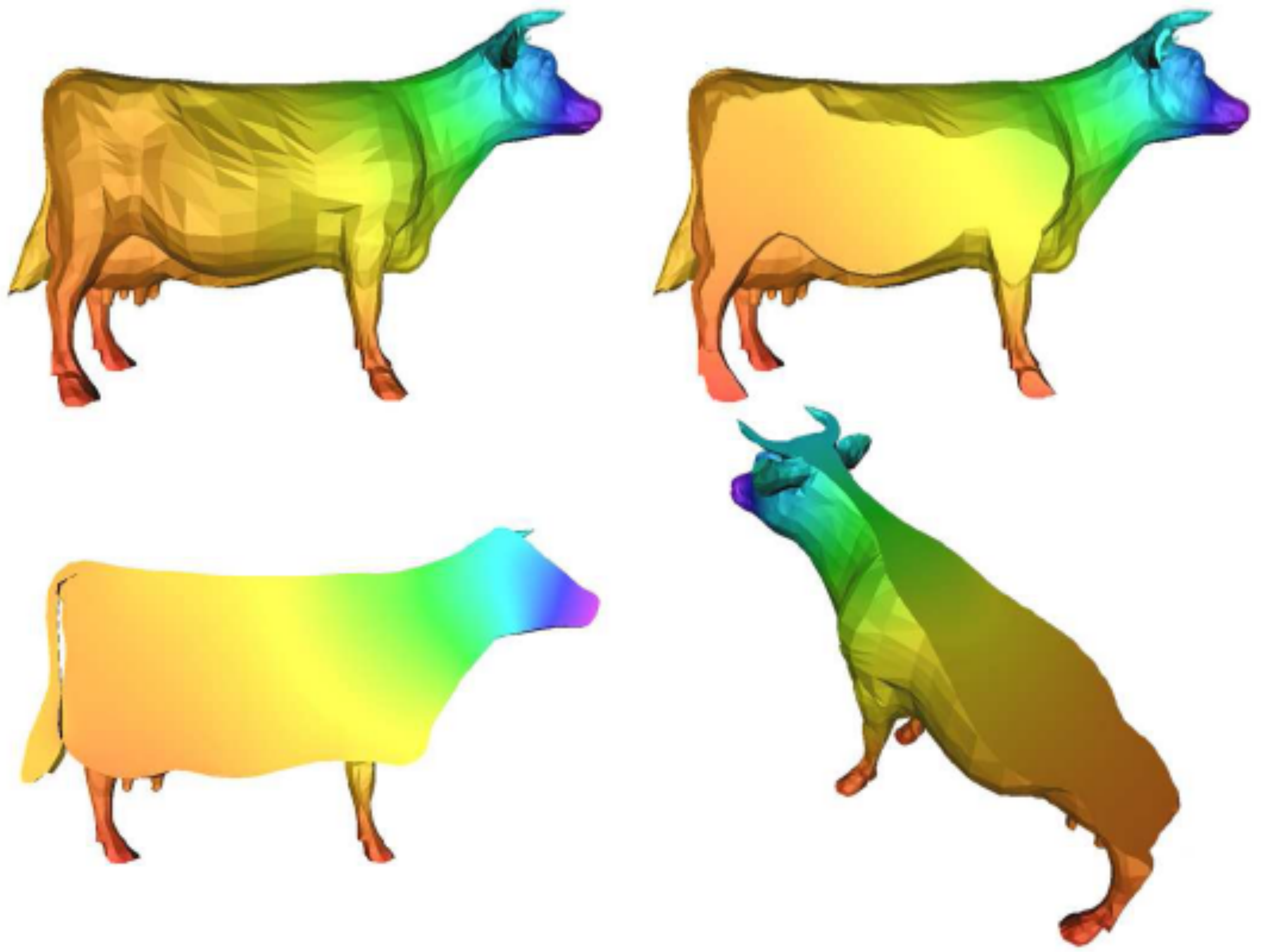
$$\mathbf{x}' = \sum_{i=1}^{k} w_i(\mathbf{x})\,\mathbf{p}'_i$$

Space Deformations: Basic Idea

- Design a set of coordinates for all points in R^d w.r.t. the "cage" vertices
  - Each point $\mathbf{x}$ can be represented as a weighted sum of cage points $\mathbf{p}_i$

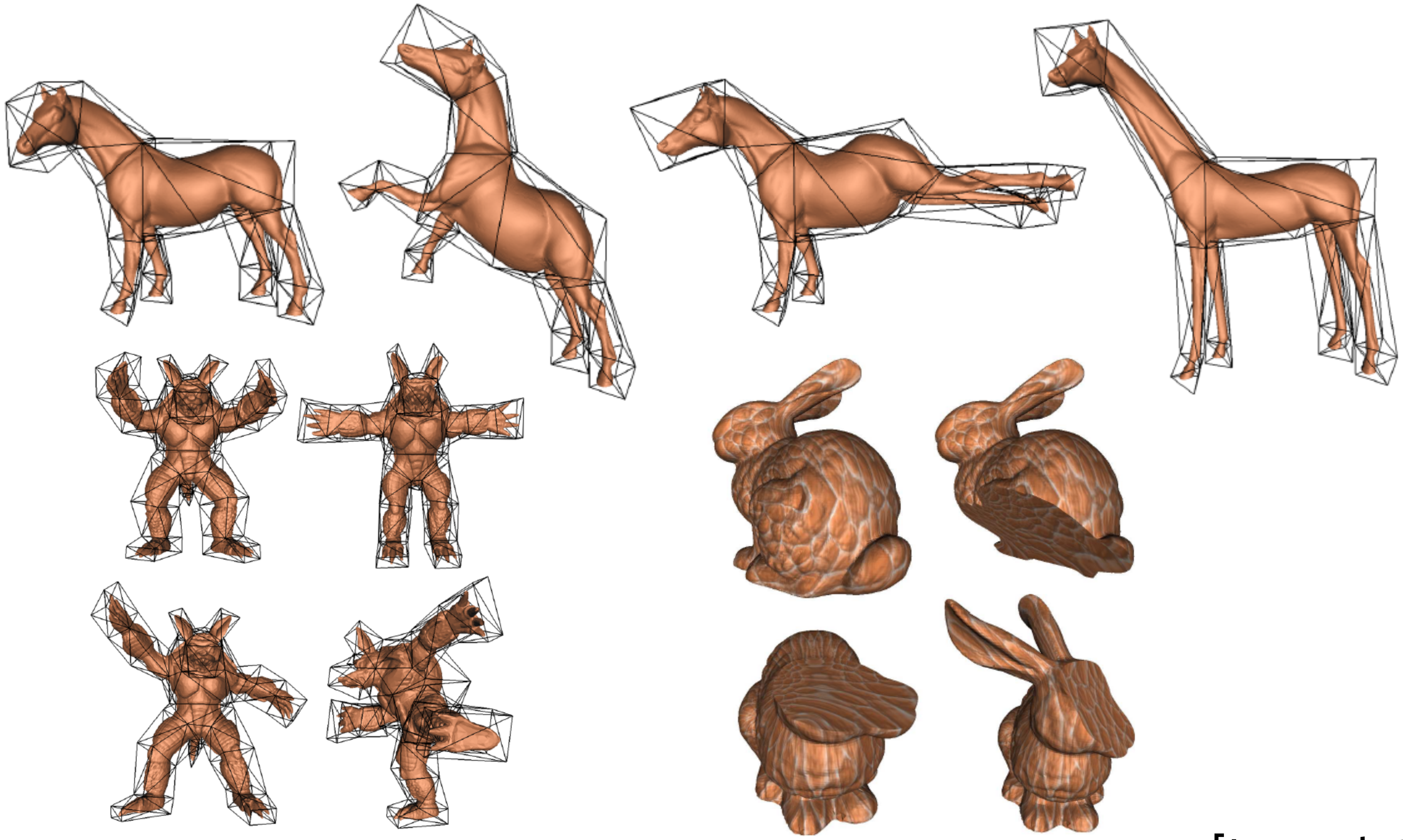
$$\mathbf{x} = \sum_{i=1}^{k} w_i(\mathbf{x})\,\mathbf{p}_i$$

  - The coordinates are smoothly varying and guarantee continuity inside the volume



Space Deformations: Examples

- Mean value coordinates for closed tri meshes

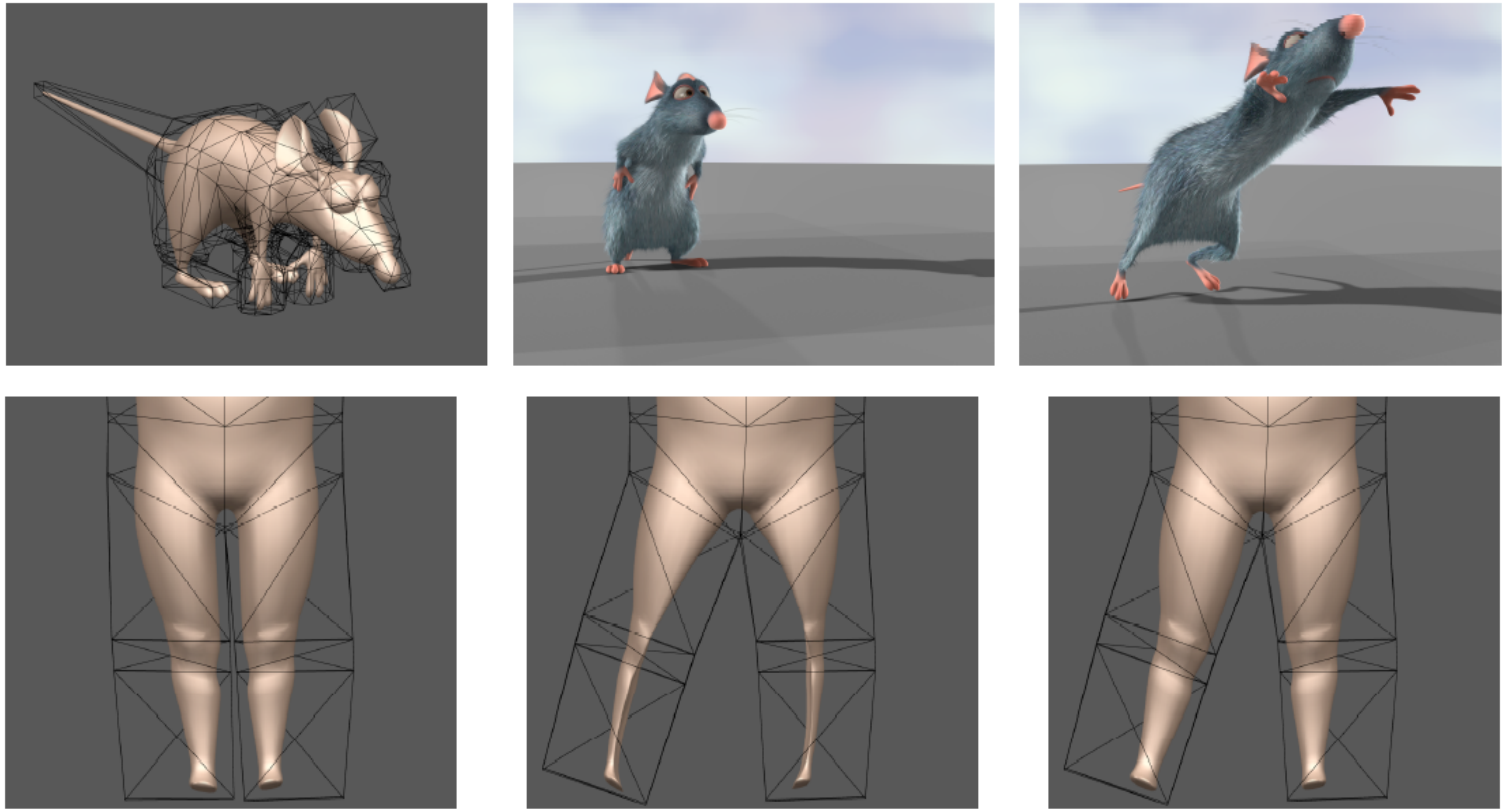


[Ju et al. 2005]

http://dl.acm.org/citation.cfm?id=1186822.1073229

Space Deformations: Examples

- Harmonic coordinates [Joshi et al. 2007]

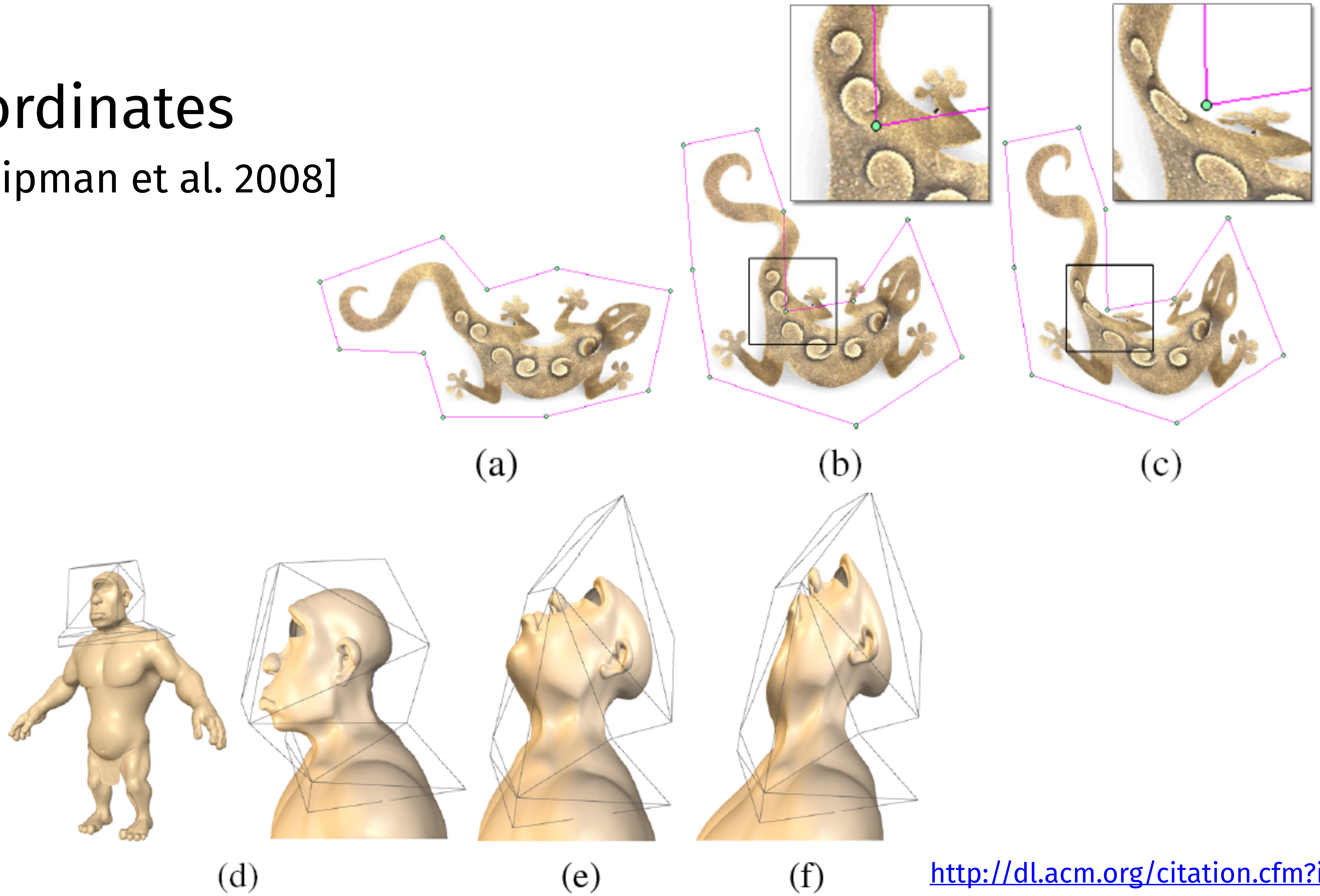


http://dl.acm.org/citation.cfm?id=1276466

Space Deformations: Examples

- Green coordinates

  [Lipman et al. 2008]



http://dl.acm.org/citation.cfm?id=1360677

Space Deformations: Summary So Far

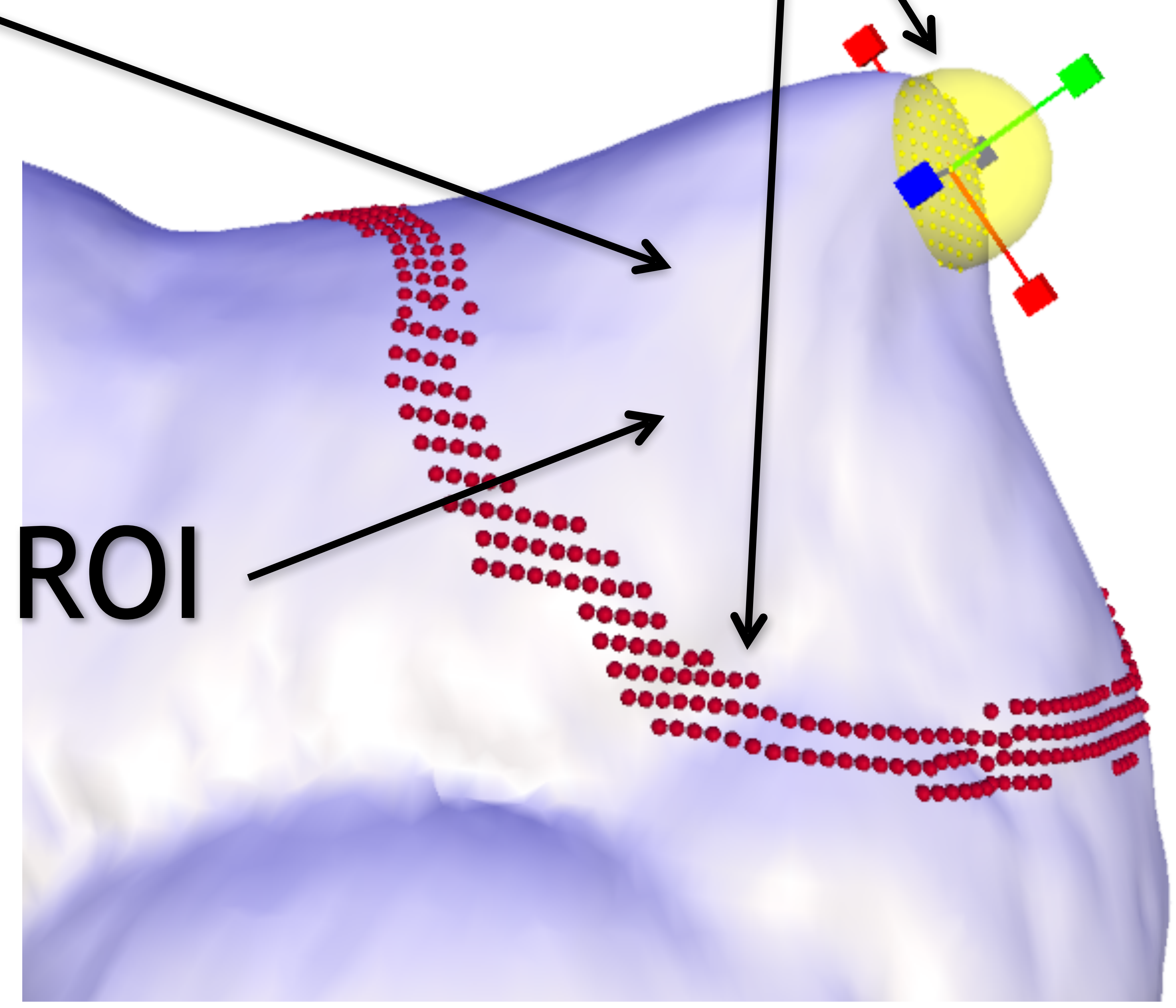
- Complexity depends mainly on the cage; linear in the number of mesh elements
  - Parallel execution, GPU!
- Can handle disconnected components or even just point sets
- Harder to control the surface properties since the whole space is being warped

Surface-based Differential Deformations

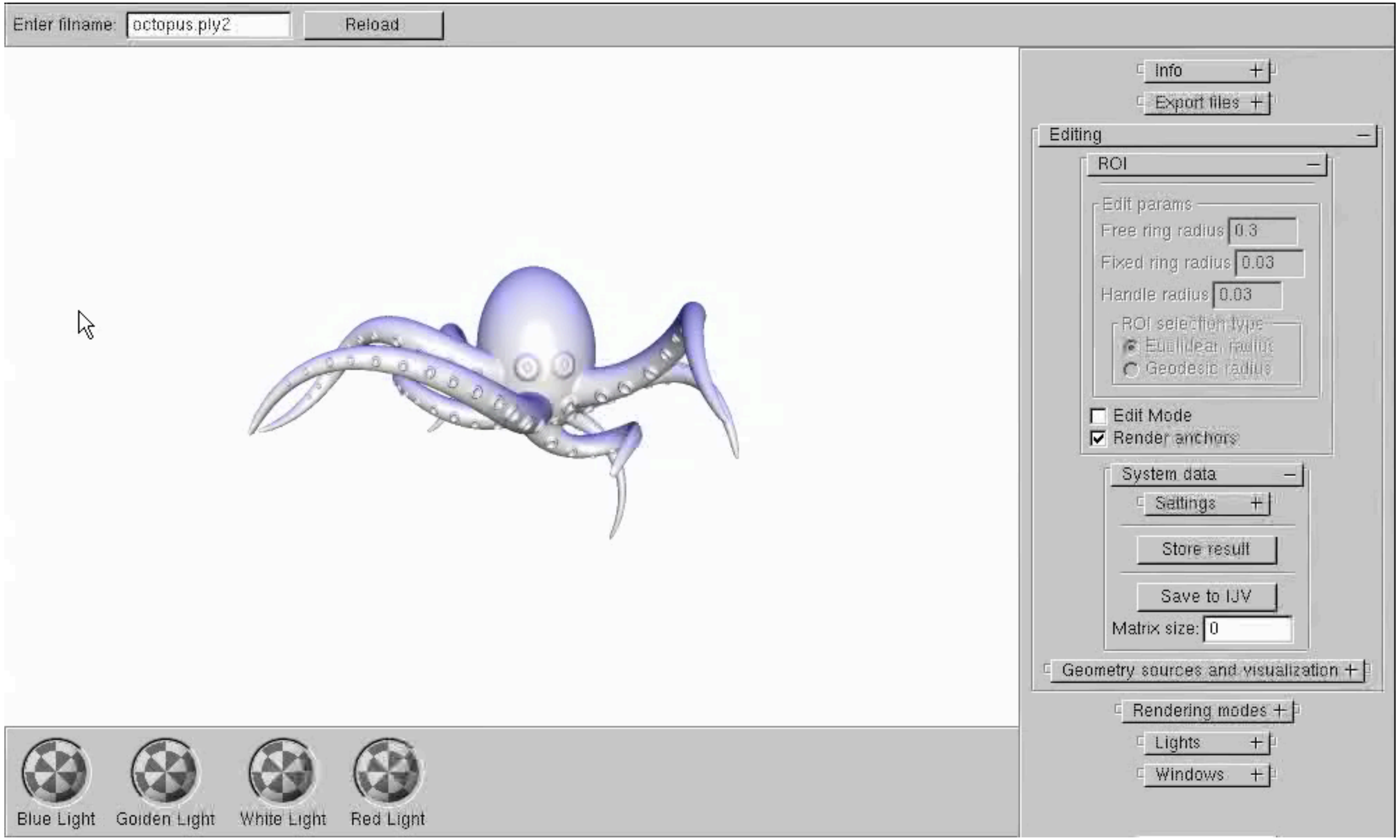
Surface-based Deformation:
ROI-Handle Editing Metaphor

$$\mathbf{x}_{\text{def}} = \operatorname{argmin}_{\mathbf{x}'} E(\mathbf{x}') \quad s.t. \quad \mathbf{x}'_i = \mathbf{c}_i$$

- ROI is bounded by a belt (static anchors)
- Manipulation through handle(s) – affine transformations

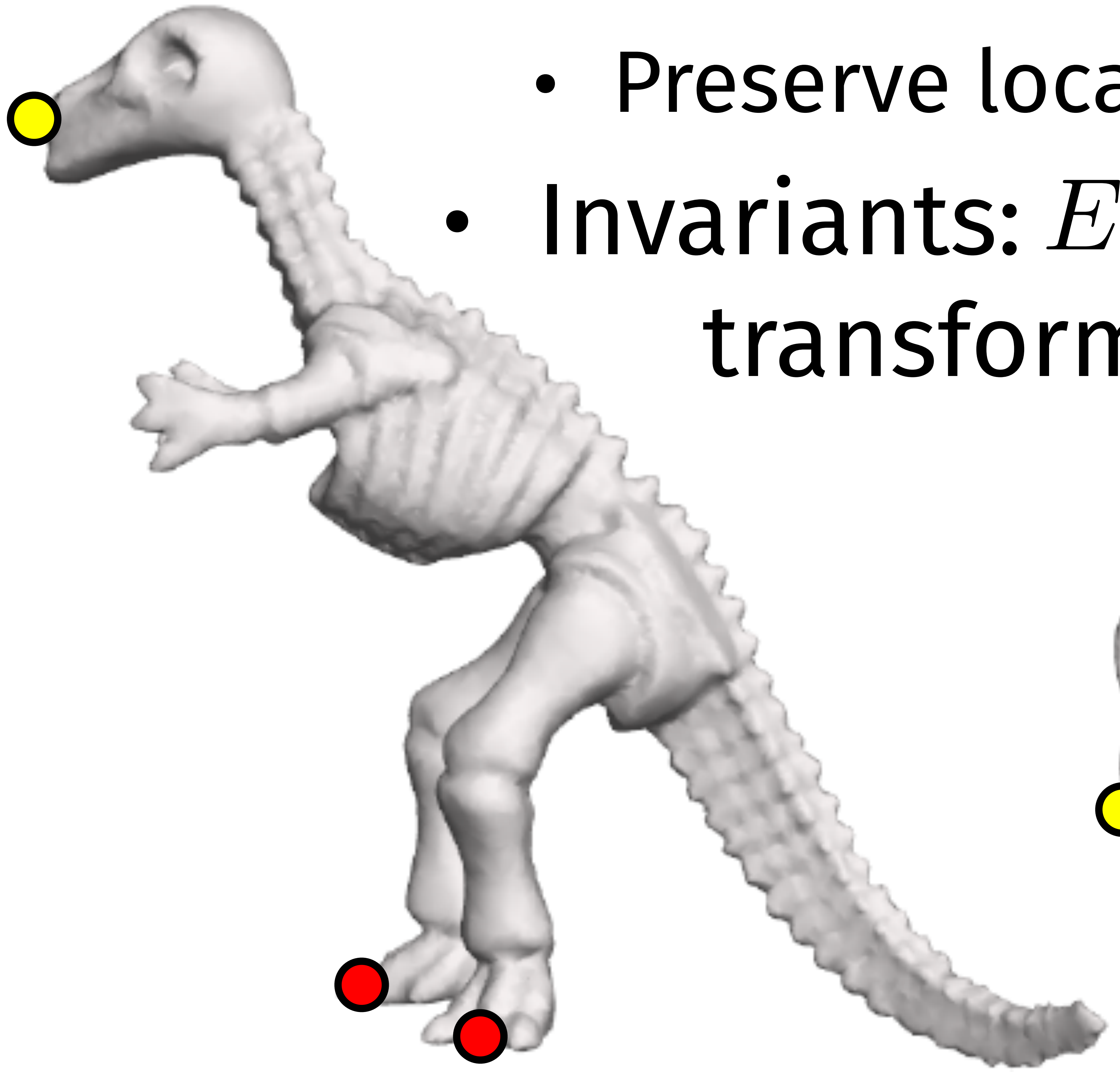


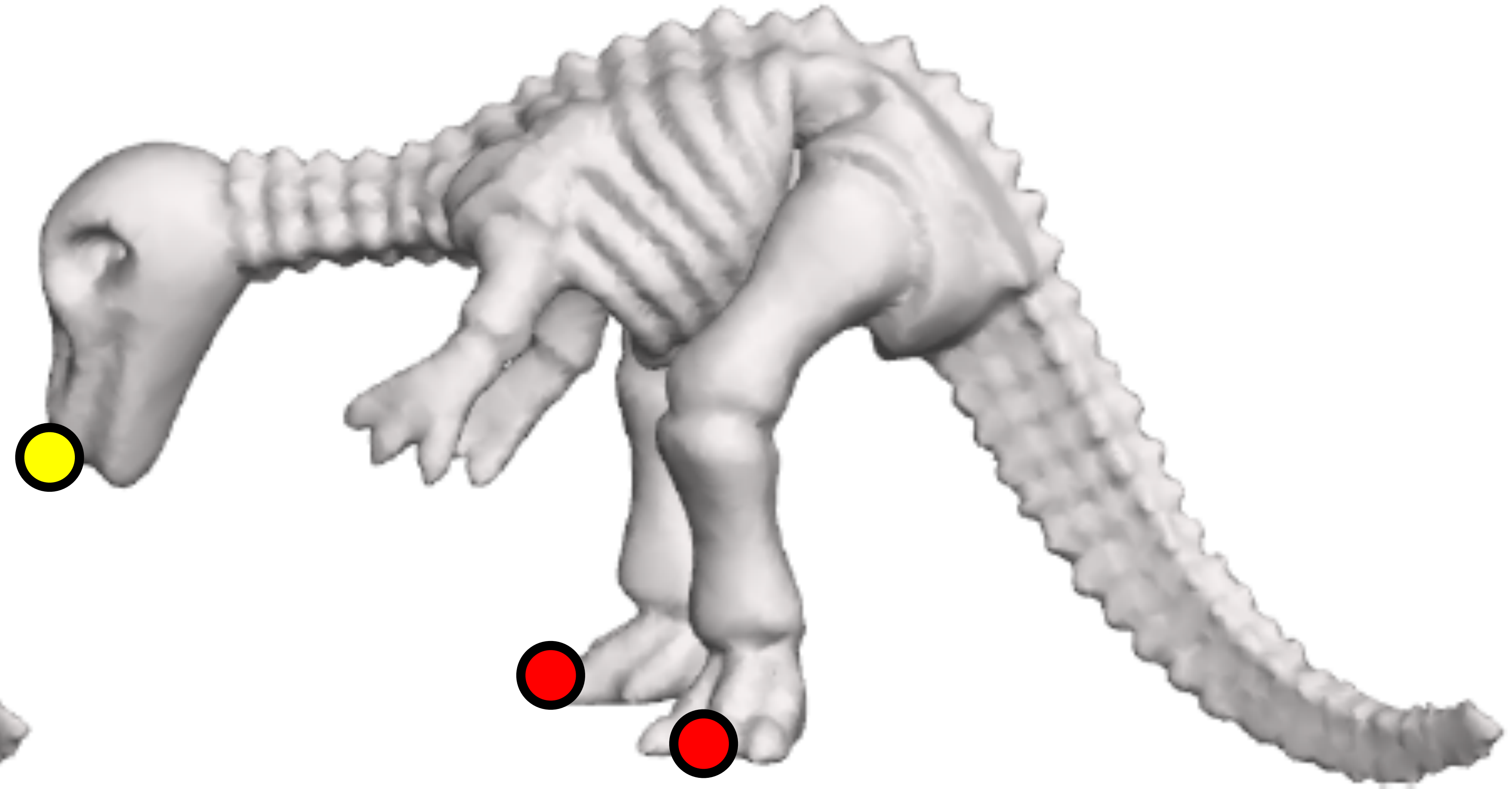
Surface-based Deformation:
ROI-Handle Editing Metaphor



How to Define $E(\mathbf{x}')$?

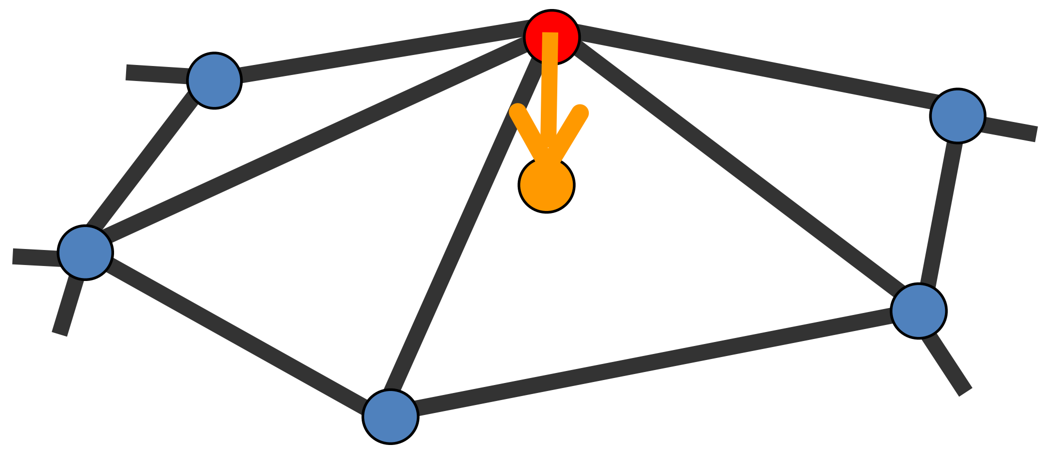
- Intuitive deformations:
  - Smooth deformation on the global scale
  - Preserve local details (curvatures)
- Invariants: $E(\mathbf{x}')$ should be zero if $\mathbf{x}'$ is a rigid transformation of original geometry $\mathbf{x}$





Recap: Differential Coordinates

- Detail = *smooth*(surface) – surface
- Smoothing = averaging



$$\delta_i = \frac{1}{W_i} \sum_{j \in \mathcal{N}(i)} w_{ij}(\mathbf{x}_j - \mathbf{x}_i) \approx -2H_i\mathbf{n}_i$$

Recap: Differential Coordinates

- Represent ***local detail*** at each surface point
  - More descriptive of the shape than just xyz
- Linear transition from xyz to δ
- Useful for operations on surfaces where surface details are important

Simple Laplacian Editing

- Preserve mean curvature normal [≈differential coordinates] at every point in the ROI [≈ every vertex of the ROI]

continuous: $$E(\mathcal{S}') = \int_{\mathcal{S}'} \|\Delta \mathbf{x}' - \delta\|^2 d\mathbf{x}'$$

discrete: $$E(\mathbf{x}') = \sum_{i=1}^{n} A_i \|\Delta(\mathbf{x}'_i) - \delta_i\|^2$$

Simplifying the Laplacian Energy

$$E(\mathbf{x}') = \sum_{i=1}^{n} A_i \|\Delta(\mathbf{x}'_i) - \delta_i\|^2 = \sum_{i=1}^{n} A_i \left(\Delta(\mathbf{x}'_i)^T \Delta(\mathbf{x}'_i) - 2\Delta(\mathbf{x}'_i)^T \delta_i + \delta_i^T \delta_i\right) =$$

$$= \ \mathbf{x}'^T \underline{L^T M L} \mathbf{x}' - 2\mathbf{x}'^T \underline{L^T M} \delta + \text{const}$$

Simplifying the Laplacian Energy

$$E(\mathbf{x}') = \sum_{i=1}^{n} A_i \|\Delta(\mathbf{x}'_i) - \delta_i\|^2 = \sum_{i=1}^{n} A_i \left(\Delta(\mathbf{x}'_i)^T \Delta(\mathbf{x}'_i) - 2\Delta(\mathbf{x}'_i)^T \delta_i + \delta_i^T \delta_i\right) =$$

$$= \ \mathbf{x}'^T \underline{L^T M L} \mathbf{x}' - 2\mathbf{x}'^T \underline{L^T M} \delta + \text{const}$$

$$L^T M L = (M^{-1} L_w)^T M (M^{-1} L_w) = L_w M^{-1} M M^{-1} L_w =$$

$$= L_w M^{-1} L_w$$ Symmetric sparse matrix!

Minimizing the Laplacian Energy

- To find the minimum, gradient = 0 and substitute the modeling constraints

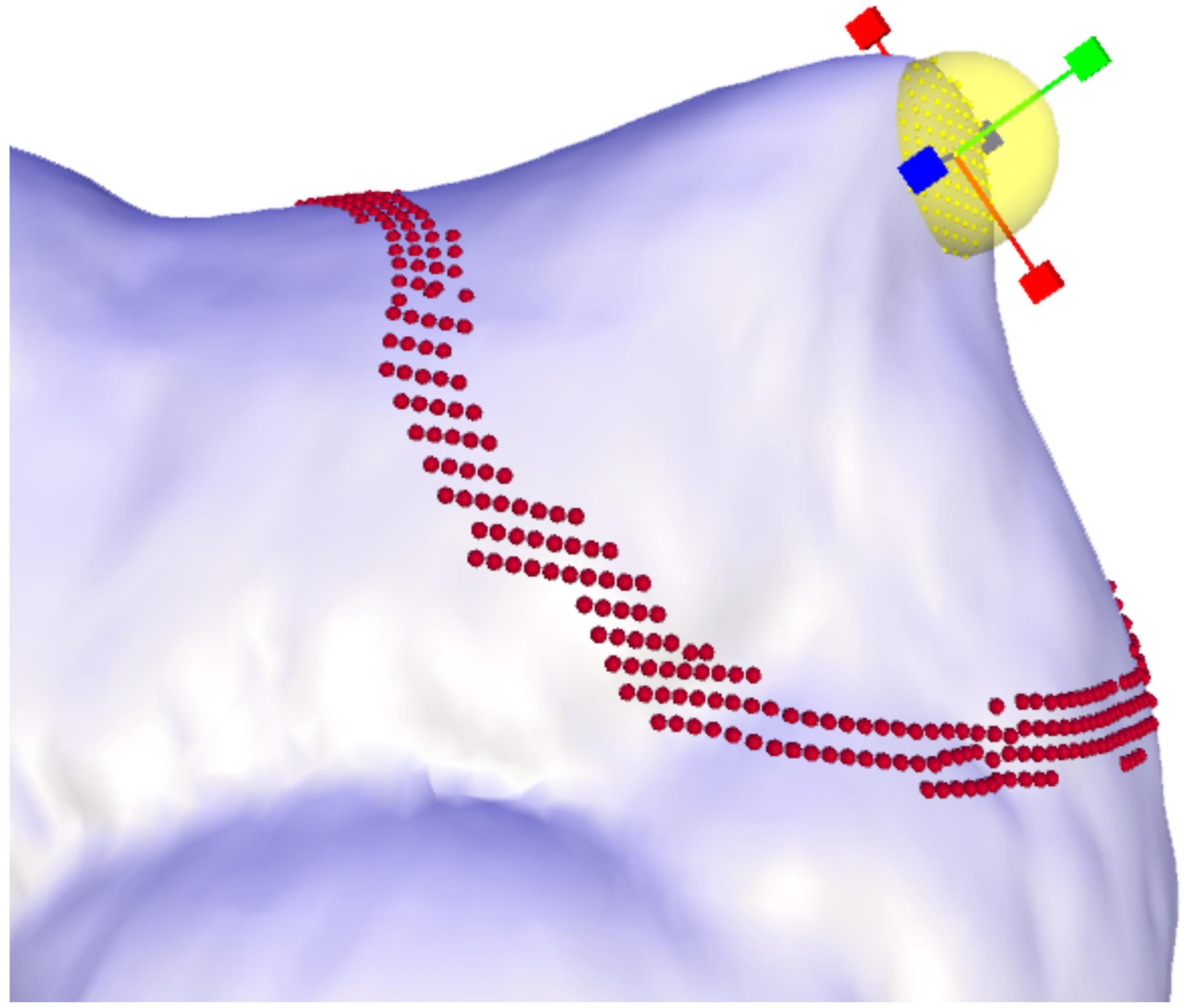
$$E(\mathbf{x}') = \ \mathbf{x}'^T L_w M^{-1} L_w \mathbf{x}' - 2\mathbf{x}'^T L_w \delta + \text{const}$$

$$\frac{\partial}{\partial \mathbf{x}'} E(\mathbf{x}') = 2 L_w M^{-1} L_w \mathbf{x}' - 2 L_w \delta$$

$$\mathbf{x}'_i = \mathbf{c}_i \,, \ \ i \in \mathcal{C}$$

Minimizing the Laplacian Energy

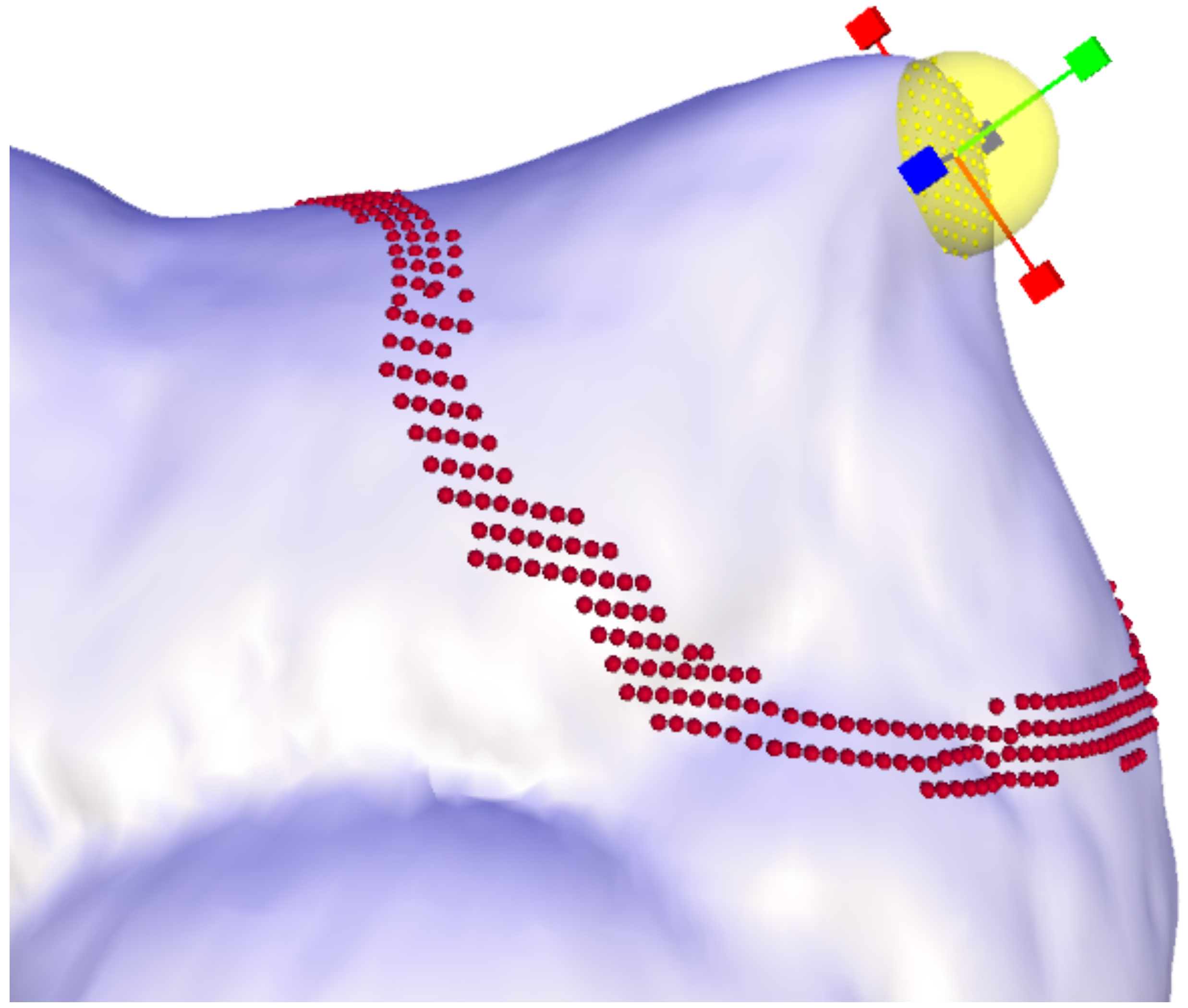
$$A\mathbf{x}' = \mathbf{b}$$



Matrix depends on the initial mesh and the indices of the constraints only. Matrix is fixed!

Right-hand side contains the coordinates of the constraints (handles)

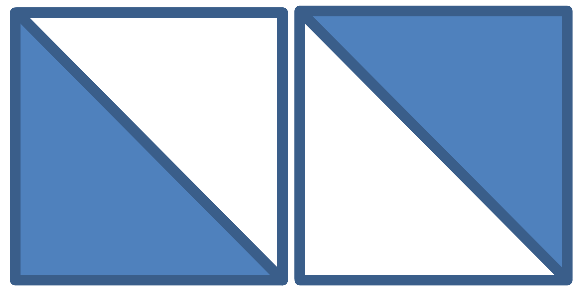
Minimizing the Laplacian Energy



$$A\mathbf{x}' = \mathbf{b}$$

Sparse Cholesky decomposition:

$$A = L_{\text{chol}} L_{\text{chol}}^T$$



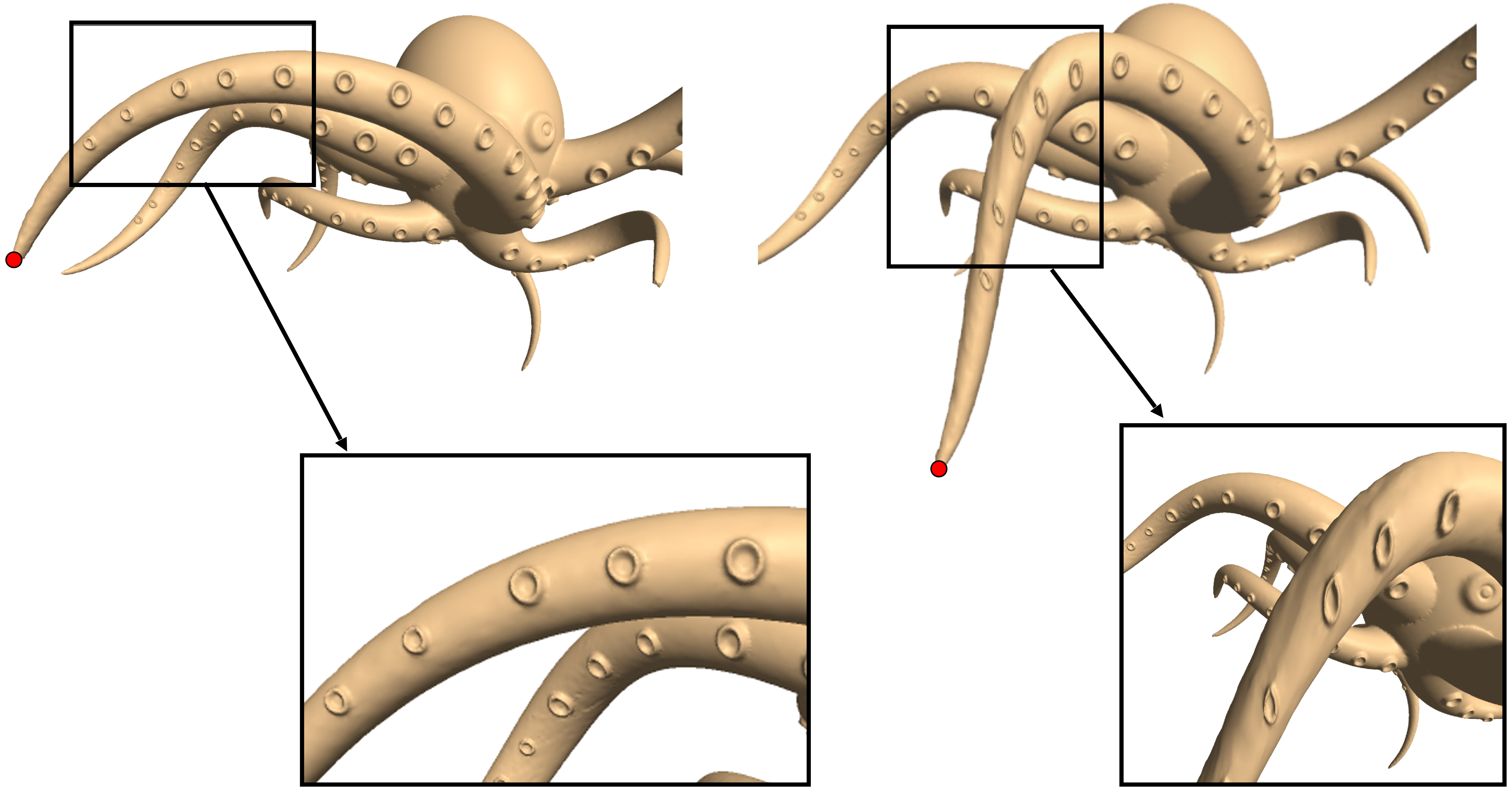
At run-time: just back-substitution!

$$L_{\text{chol}}\mathbf{y} = \mathbf{b}$$

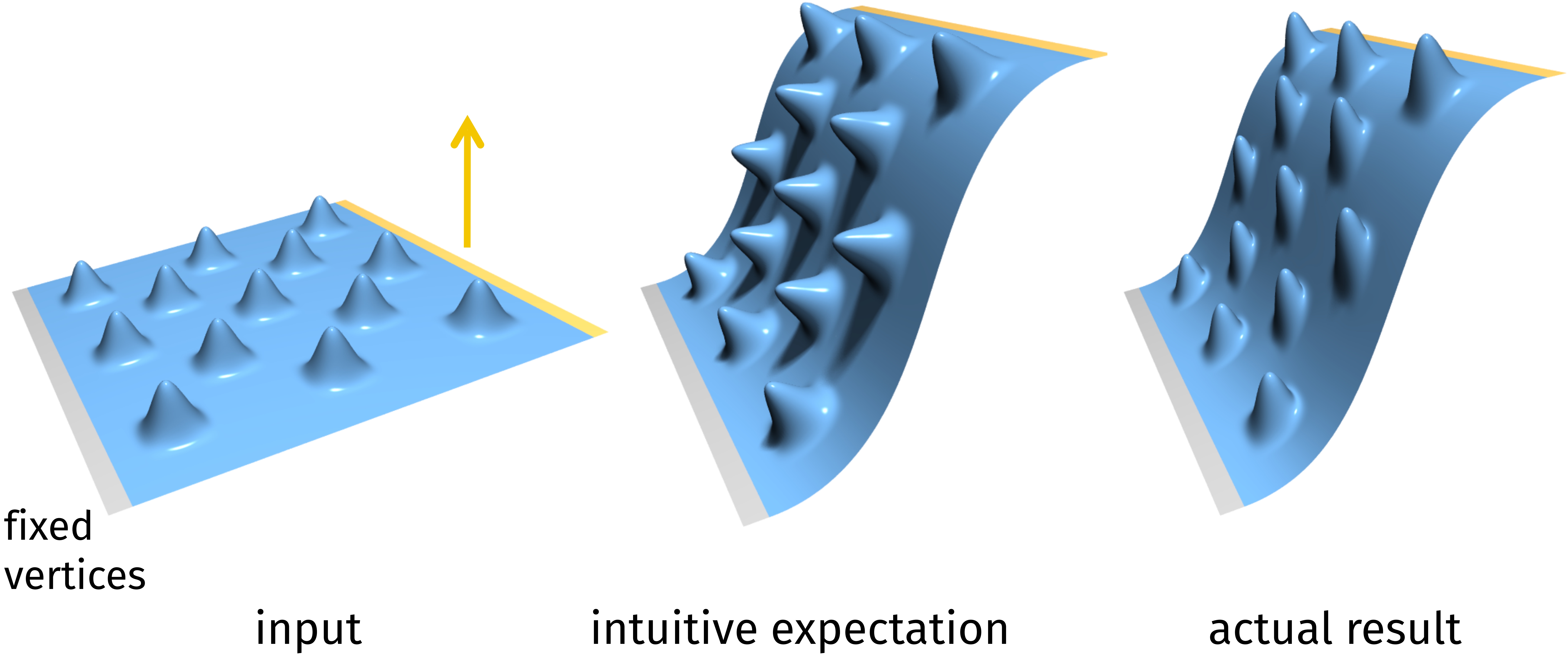
$$L_{\text{chol}}^T\mathbf{x}' = \mathbf{y}$$

Fundamental Problem:
Invariance to Transformations

- The basic Laplacian operator is ***translation***-invariant, but not **rotation-**invariant
- $E(\mathbf{x}')$ attempts to preserve the **original global** orientation of the details (the normal directions)

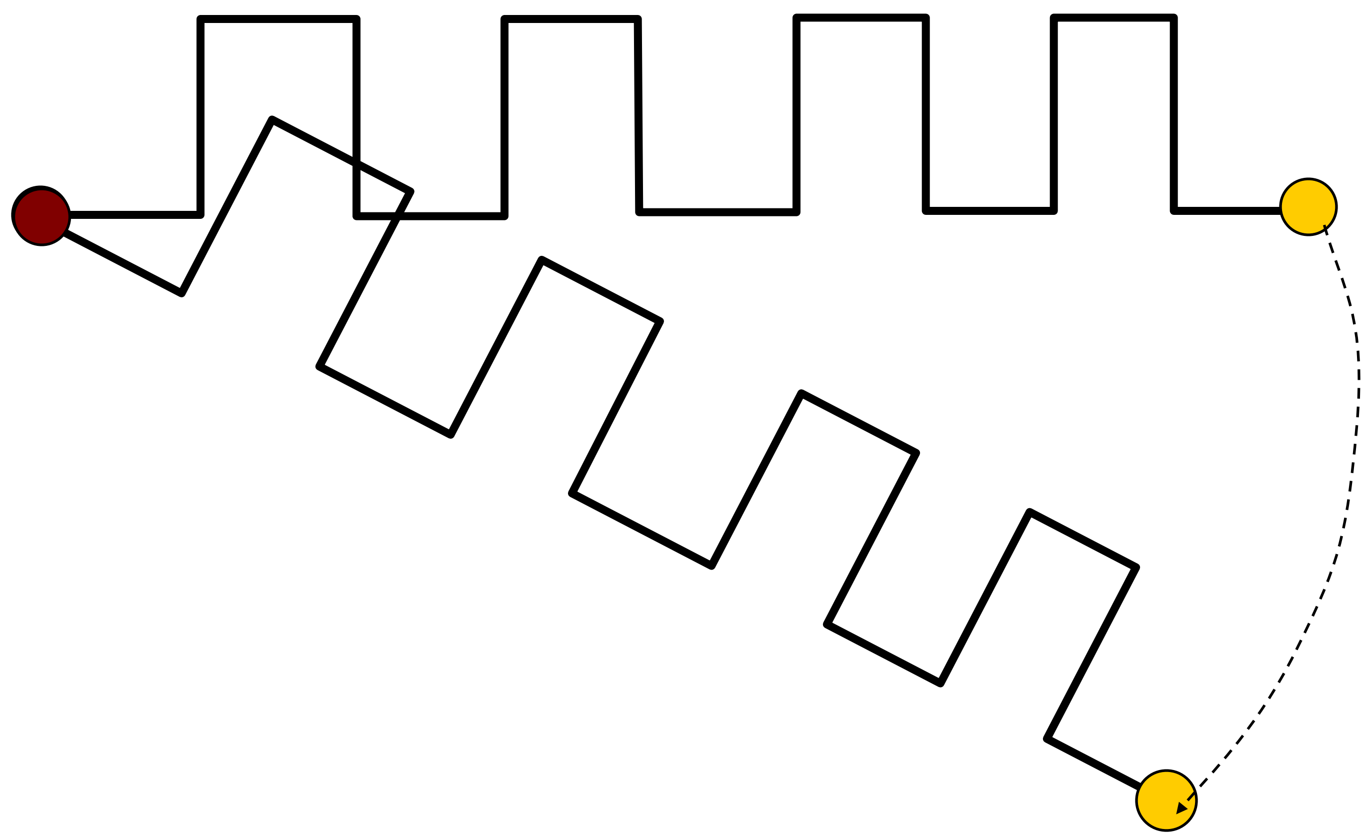


Fundamental Problem: Invariance to Transformations



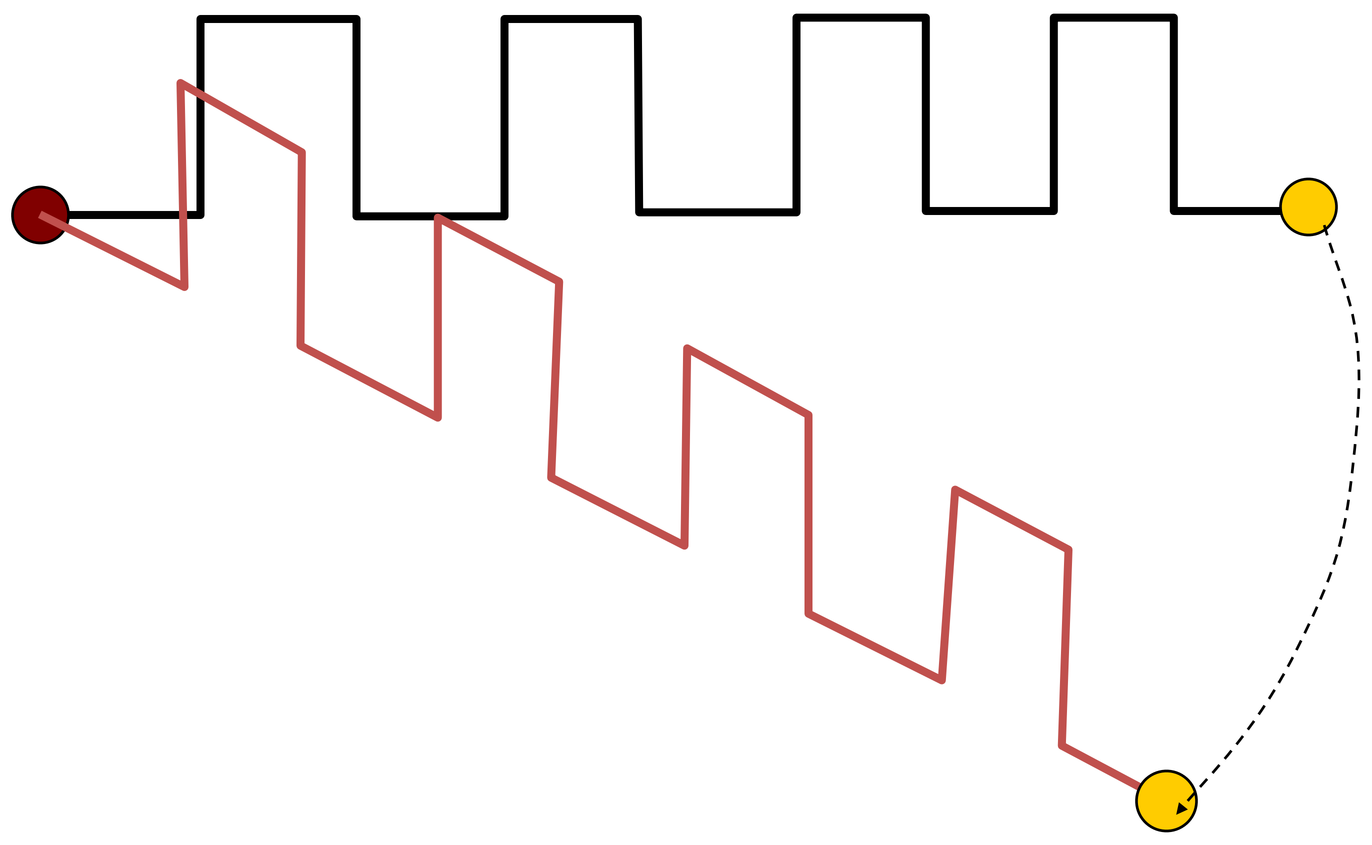
Fundamental Problem:
Invariance to Transformations

- The basic Laplacian operator is ***translation***-invariant, but not **rotation-**invariant
- $E(\mathbf{x}')$ attempts to preserve the **original global** orientation of the details (the normal directions)



Fundamental Problem: Invariance to Transformations

- The basic Laplacian operator is ***translation***-invariant, but not **rotation-**invariant
- $E(\mathbf{x}')$ attempts to preserve the **original global** orientation of the details (the normal directions)



Energy Functional

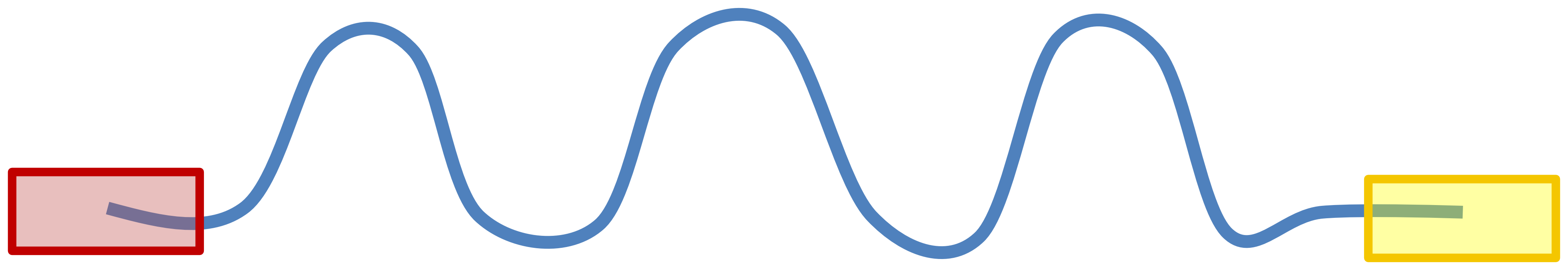
- We need a rigid-invariant energy...

$$E(\mathbf{x}') = \sum_{i=1}^{n} A_i \|\Delta(\mathbf{x}'_i) - \delta_i\|^2$$

Need to locally
rotate the *target*
m.c. normals

Fixing Local Rotations: Multiresolution Approach

input



$\mathcal{S}$

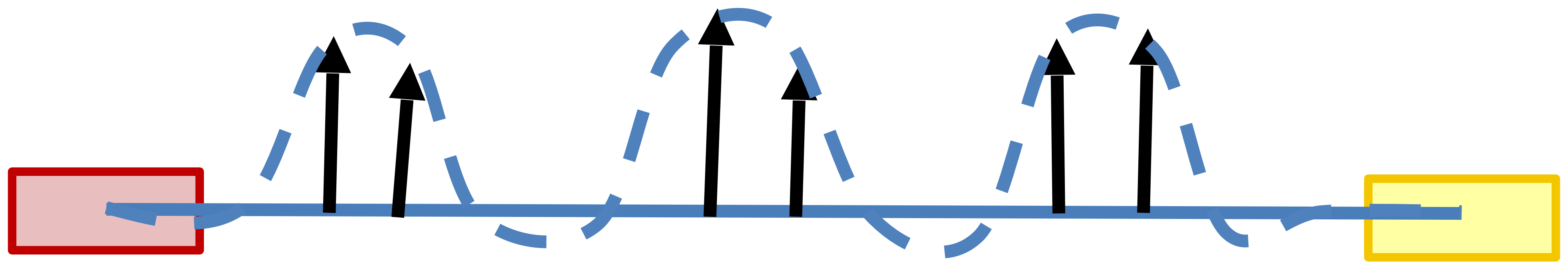
Fixing Local Rotations: Multiresolution Approach

Smooth base surface

$\mathcal{B}$

Fixing Local Rotations: Multiresolution Approach

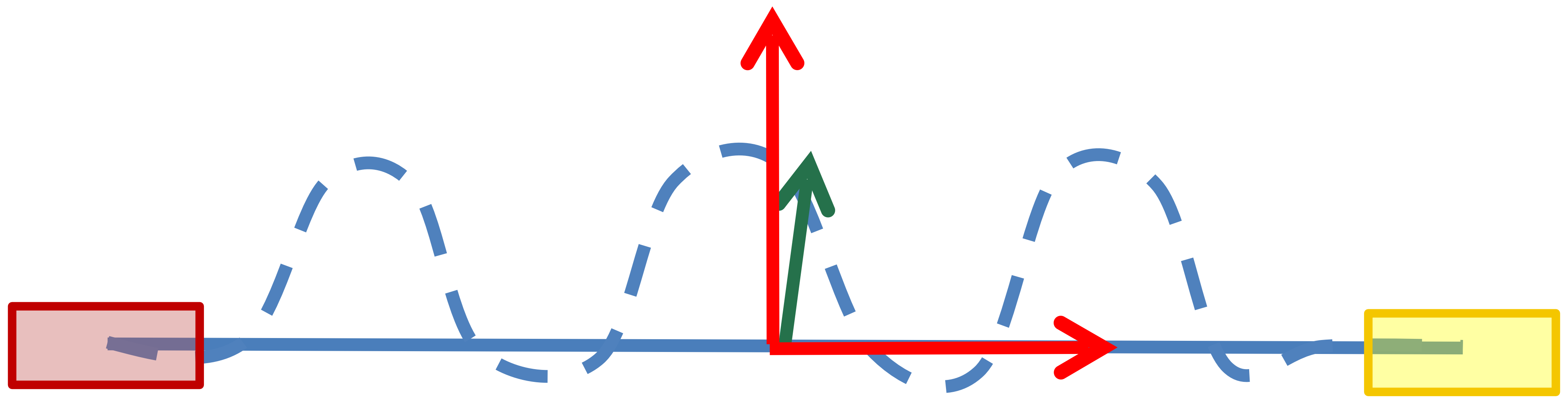
Details – displacement vectors



$$\mathcal{S} - \mathcal{B}$$

Fixing Local Rotations: Multiresolution Approach

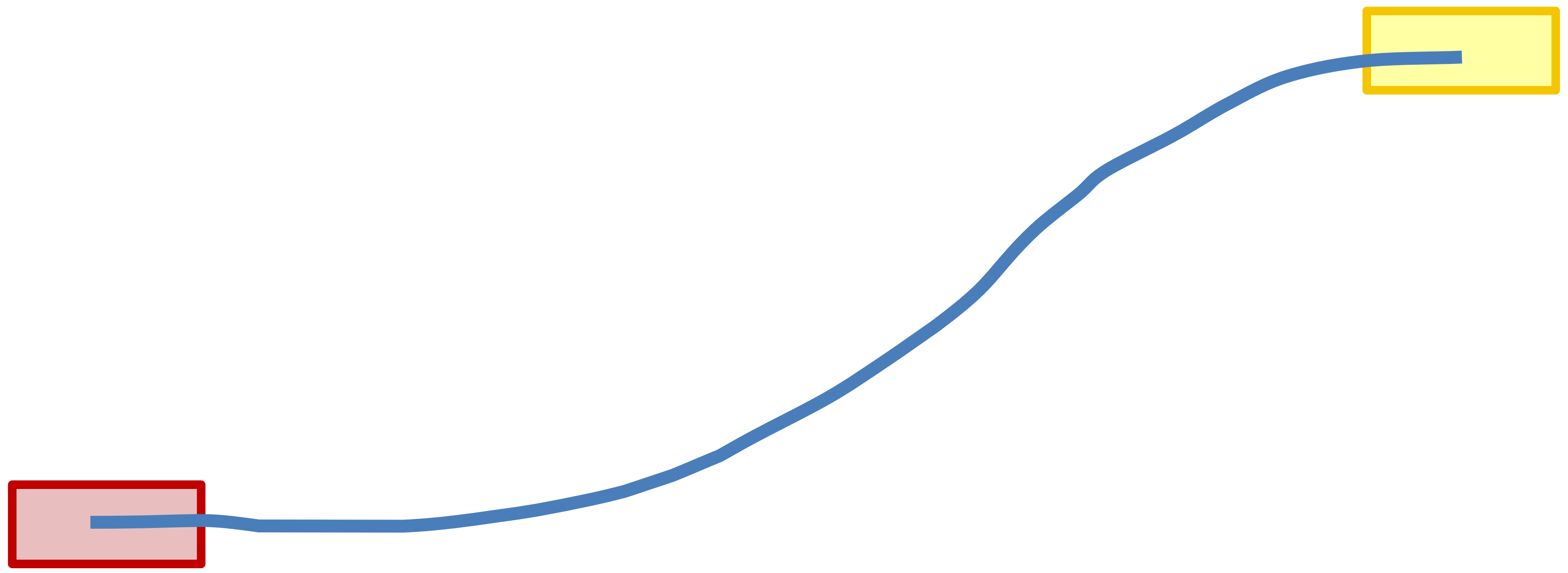
Encode details in the local frame of B



$$\mathbf{d}_i = a_1 \mathbf{t}_i + a_2 \mathbf{n}_i$$

Fixing Local Rotations: Multiresolution Approach

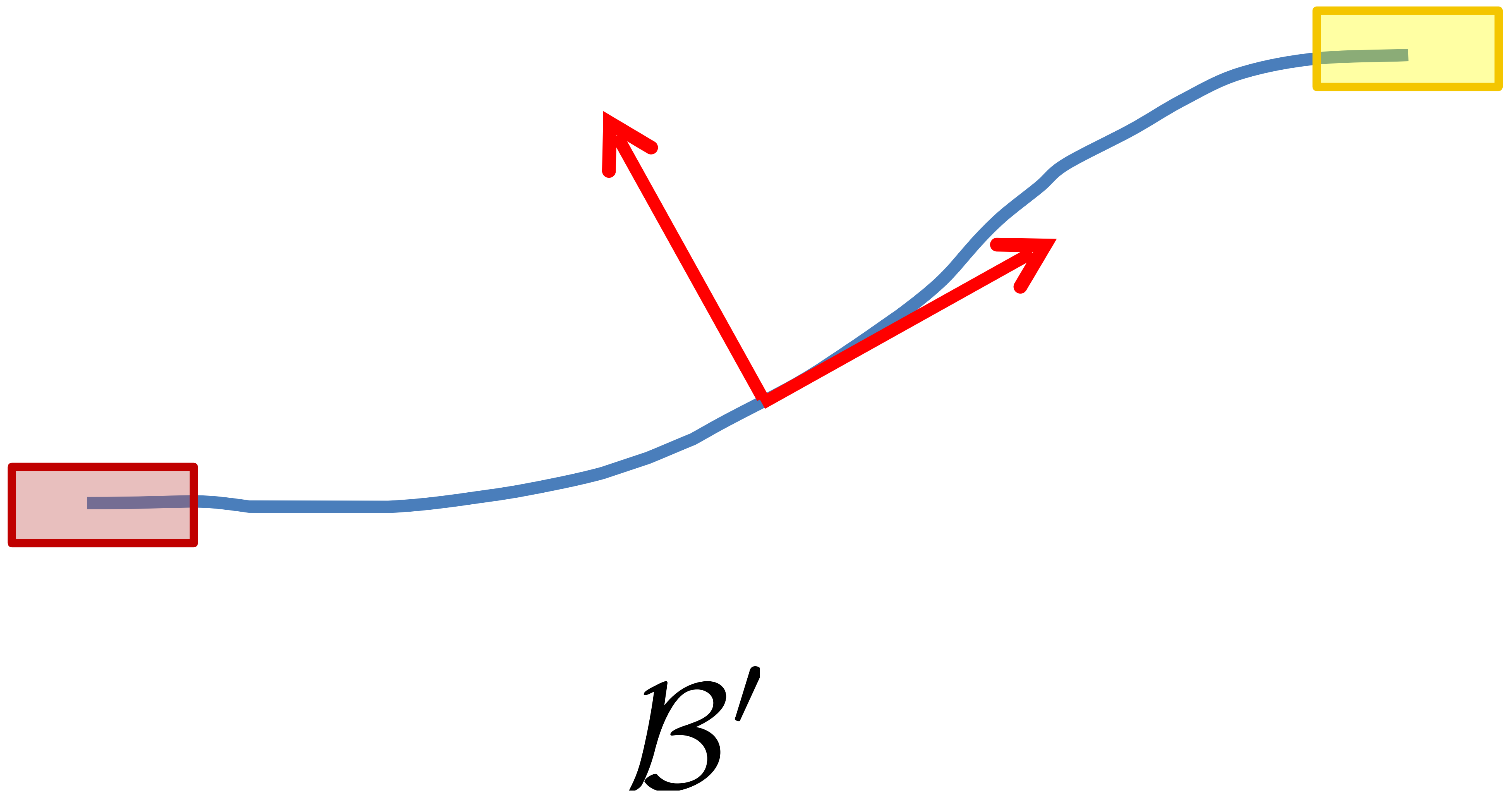
Deform smooth base surface



$\mathcal{B}'$

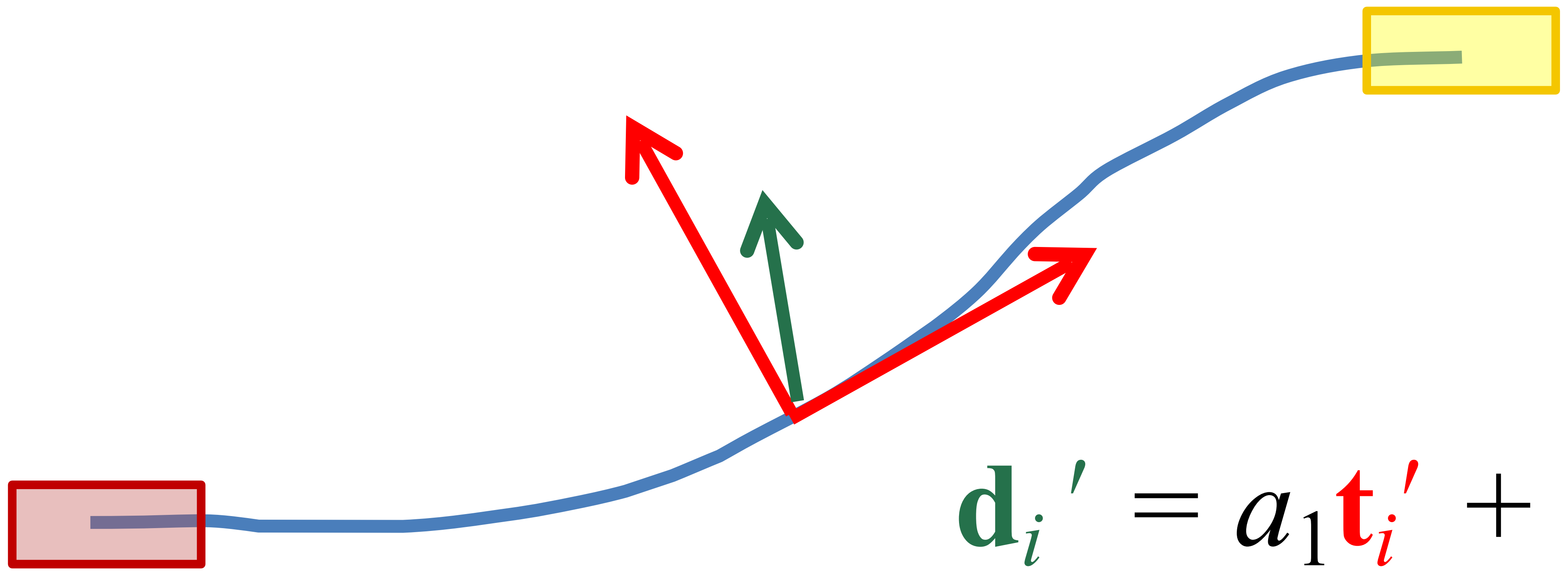
Fixing Local Rotations: Multiresolution Approach

Local frames on B'



Fixing Local Rotations: Multiresolution Approach

Add details back – in local frame!

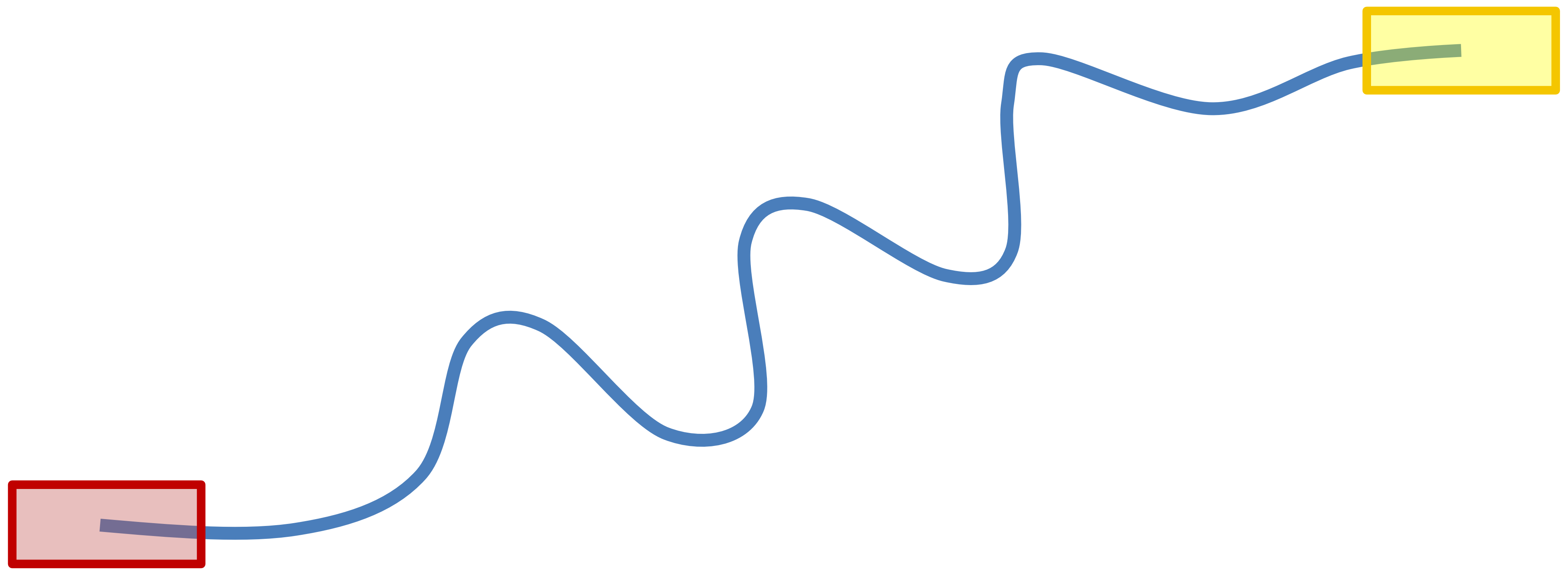


$$\mathbf{d}_i{}' = a_1\mathbf{t}_i' + a_2\mathbf{n}_i'$$

$$\mathcal{B}'$$

Fixing Local Rotations: Multiresolution Approach

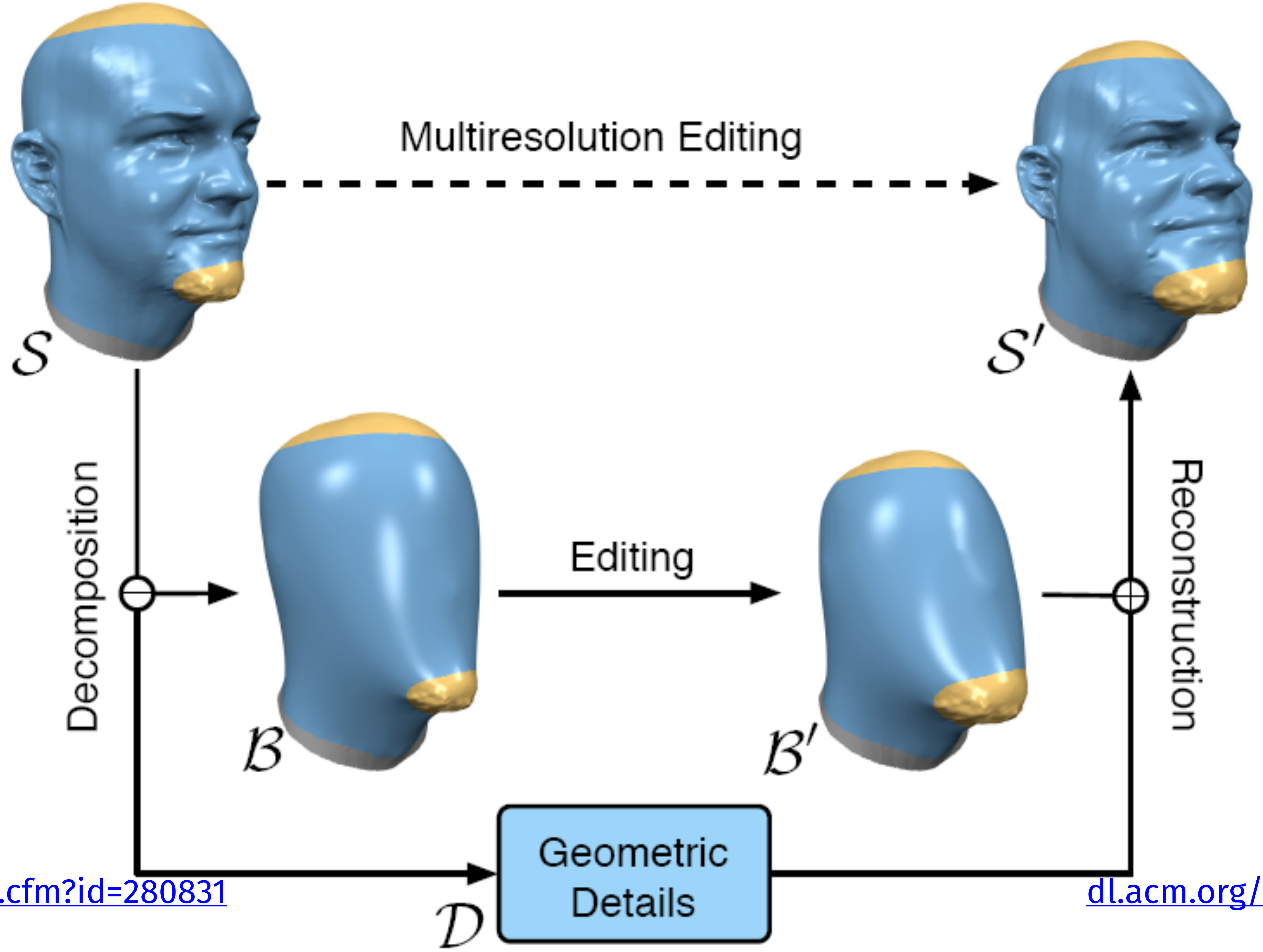
Displace the vertices to get the result



$\mathcal{S}'$

Fixing Local Rotations: Multiresolution Approach

- Kobbelt et al. SIGGRAPH 98, Botsch and Kobbelt SIGGRAPH 2004

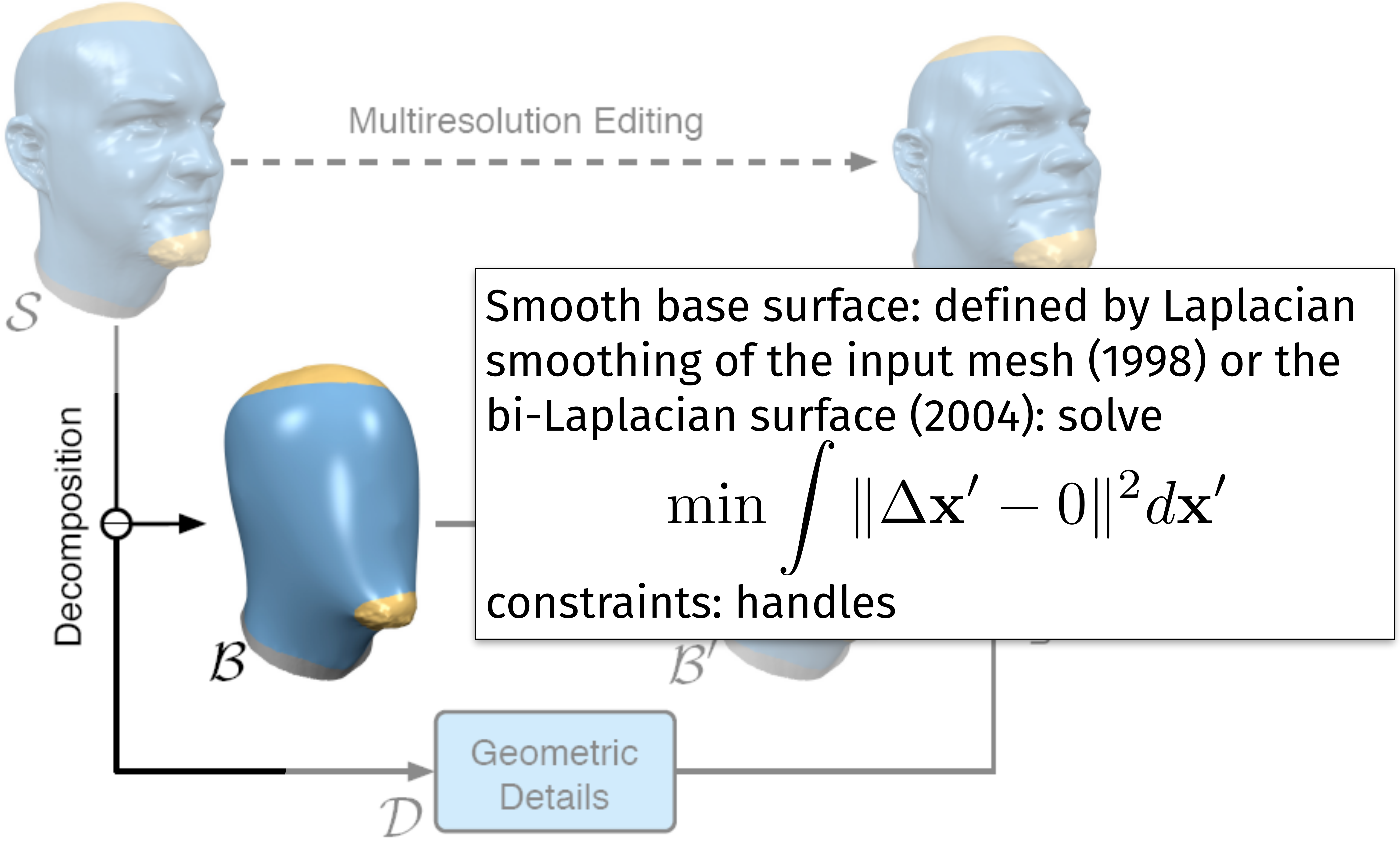


http://dl.acm.org/citation.cfm?id=280831

dl.acm.org/citation.cfm?id=1015772

Fixing Local Rotations: Multiresolution Approach

- Kobbelt et al. SIGGRAPH 98, Botsch and Kobbelt SIGGRAPH 2004



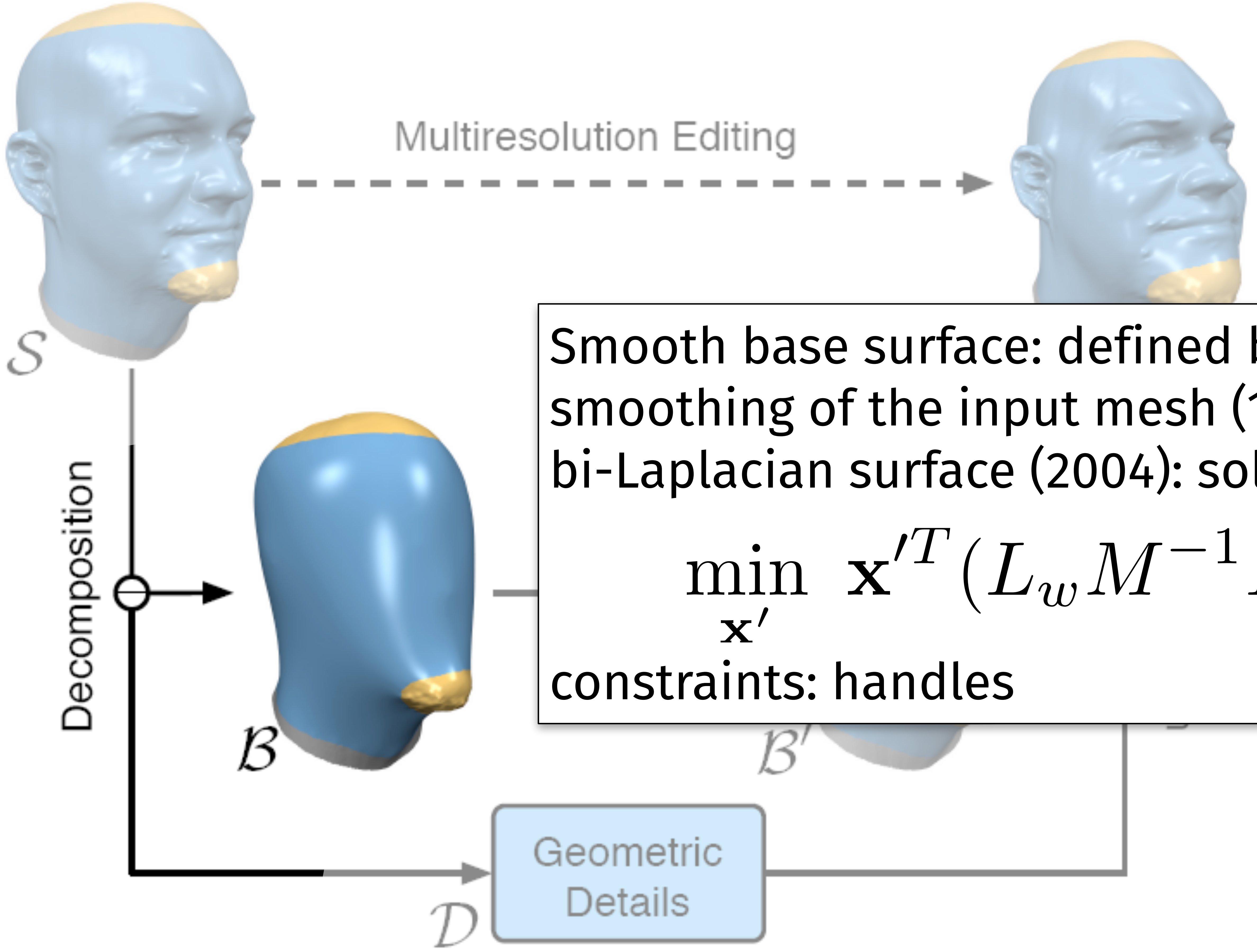
Smooth base surface: defined by Laplacian smoothing of the input mesh (1998) or the bi-Laplacian surface (2004): solve

$$\min \int \|\Delta \mathbf{x}' - 0\|^2 d\mathbf{x}'$$

constraints: handles

Fixing Local Rotations: Multiresolution Approach

- Kobbelt et al. SIGGRAPH 98, Botsch and Kobbelt SIGGRAPH 2004



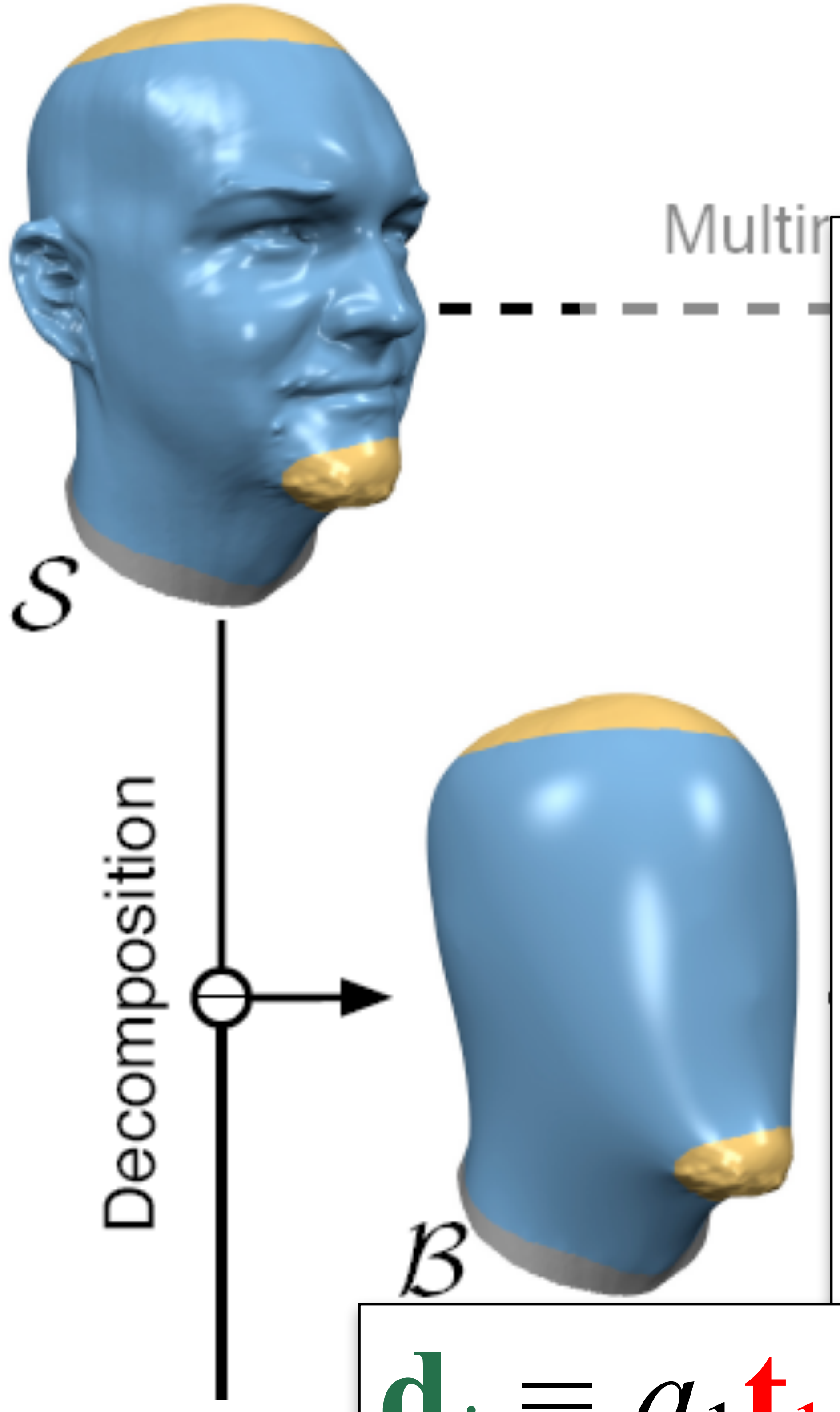
Smooth base surface: defined by Laplacian smoothing of the input mesh (1998) or the bi-Laplacian surface (2004): solve

$$\min_{\mathbf{x}'} \; \mathbf{x}'^T \left(L_w M^{-1} L_w\right) \mathbf{x}'$$

constraints: handles

Fixing Local Rotations: Multiresolution Approach

- Kobbelt et al. SIGGRAPH 98, Botsch and Kobbelt SIGGRAPH 2004

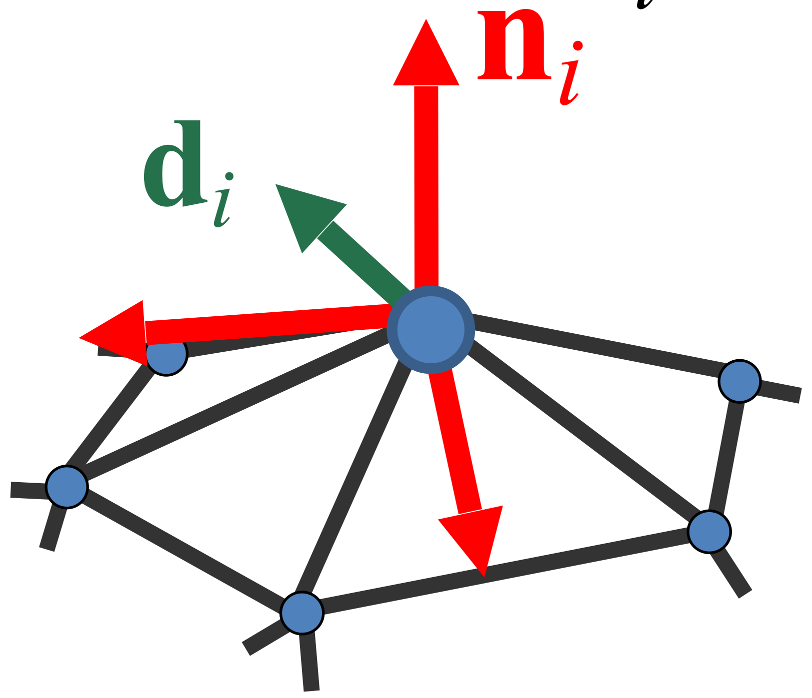


Geometric details: simply the difference vectors between base surface *B* and input *S*

For each vertex, compute

$$\mathbf{d}_i = \mathbf{v}_i - \mathbf{b}_i$$

Represent $\mathbf{d}_i$ in the **local frame** of $\mathbf{b}_i$



$$\mathbf{d}_i = a_1\mathbf{t}_1 + a_2\mathbf{t}_2 + a_3\mathbf{n}_i$$

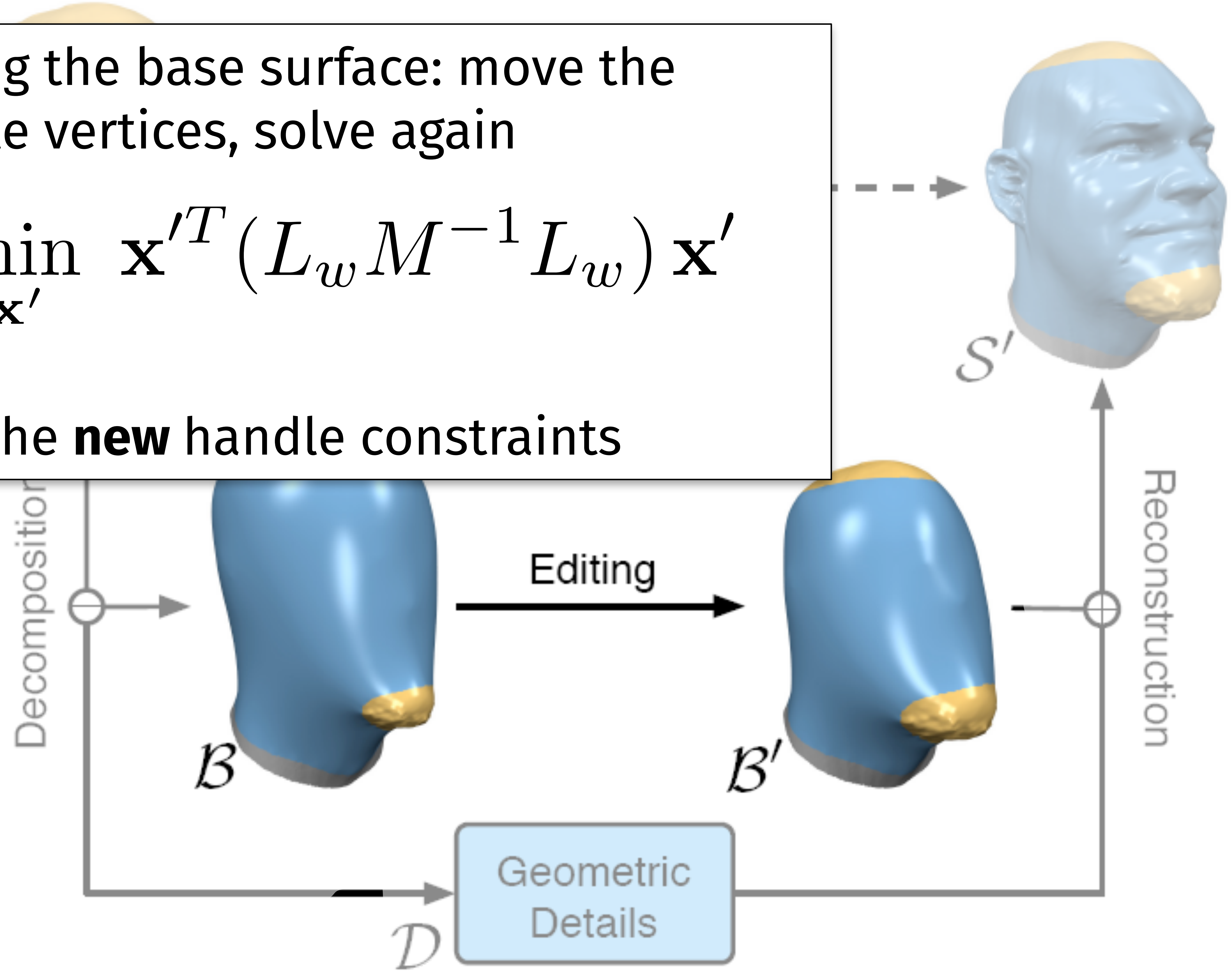
Fixing Local Rotations: Multiresolution Approach

- Kobbelt et al. SIGGRAPH 98, Botsch and Kobbelt SIGGRAPH 2004

Editing the base surface: move the handle vertices, solve again

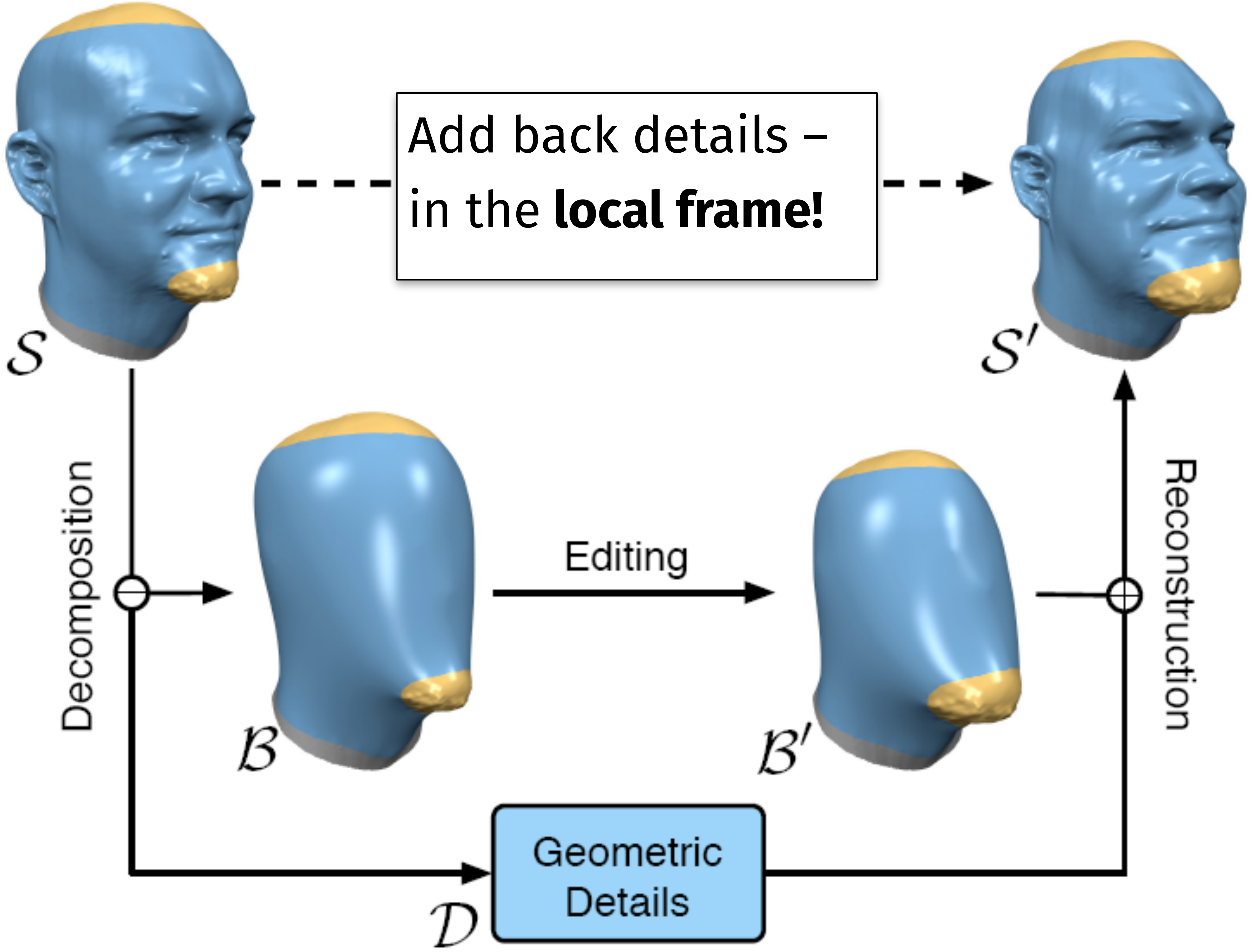
$$\min_{\mathbf{x}'} \; \mathbf{x}'^T \left(L_w M^{-1} L_w \right) \mathbf{x}'$$

with the **new** handle constraints



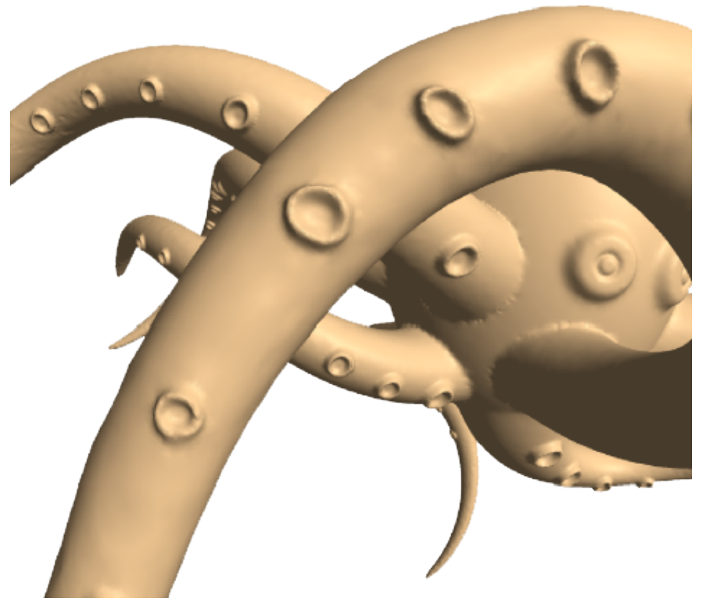
Fixing Local Rotations: Multiresolution Approach

- Kobbelt et al. SIGGRAPH 98, Botsch and Kobbelt SIGGRAPH 2004

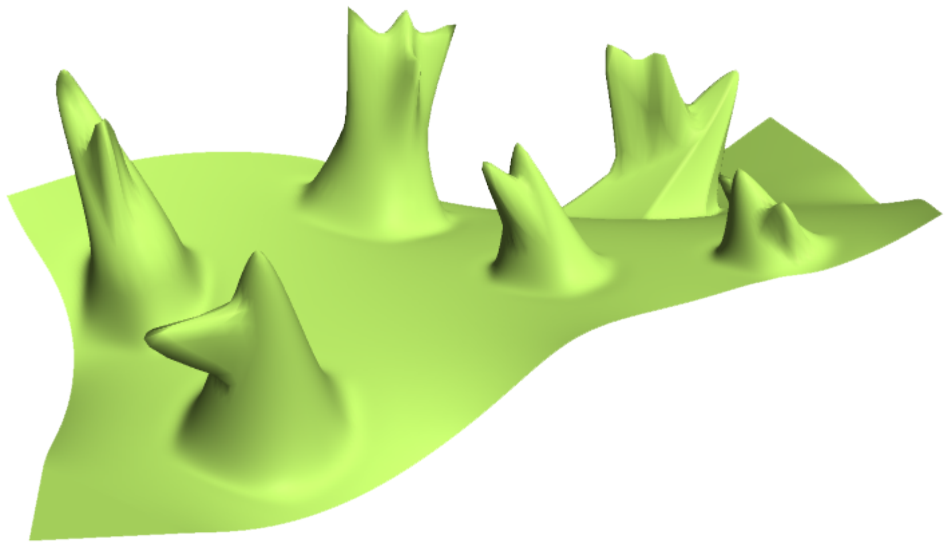


Multiresolution Framework: Discussion

- Advantages:
  - Fast! Linear solve for the base surface deformation, and then add back displacements
  - Intuitive, easy to implement
- Problem: works only for small height fields (when details vectors are small)



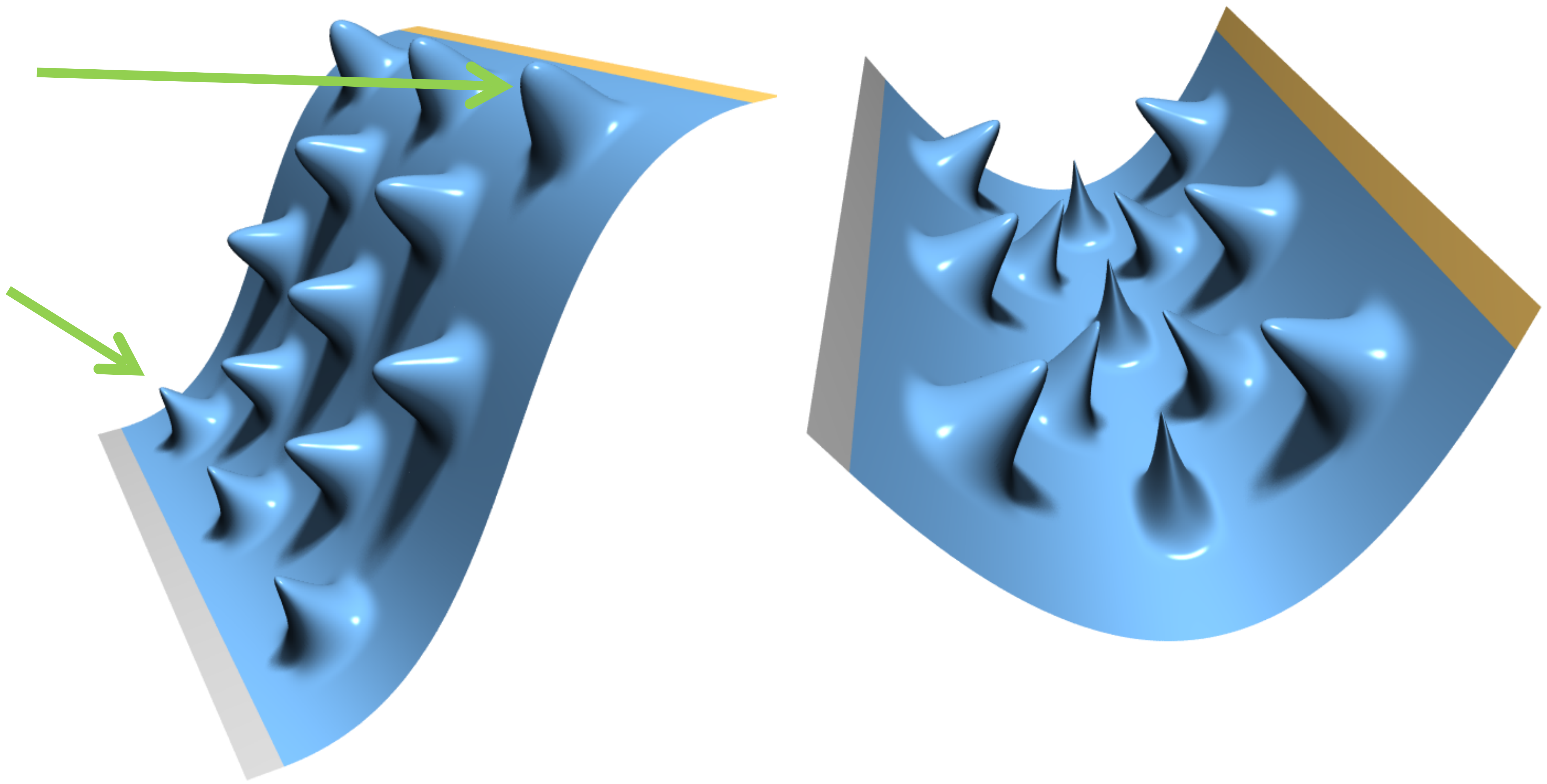
almost a height field



not a height field

Multiresolution Framework: Discussion

- Problem: If detail vectors are too big we get overshooting and **self-intersections,**
  especially in concave cases



Thank you