Laplacian Meshes

COS 526 – Fall 2014

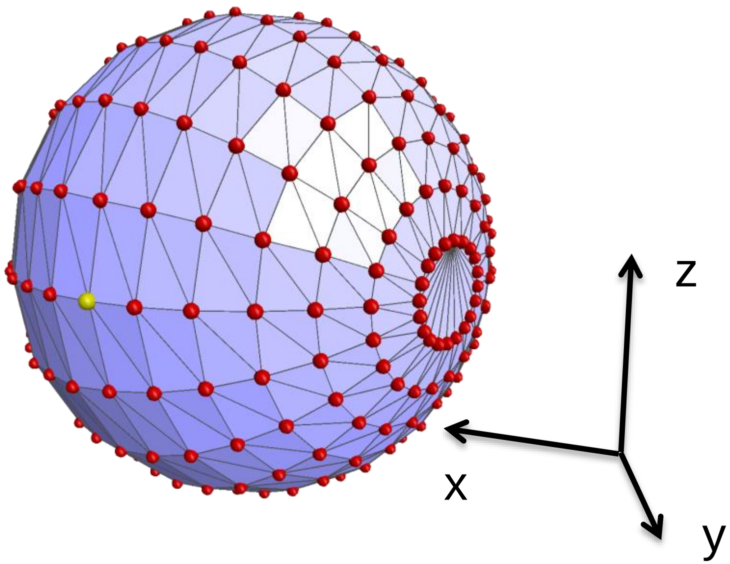
Slides from Olga Sorkine and Yaron Lipman

Outline

- Differential surface representation
- Ideas and applications
  - Compact shape representation
  - Mesh editing and manipulation
  - Membrane and flattening
  - Generalizing Fourier basis for surfaces

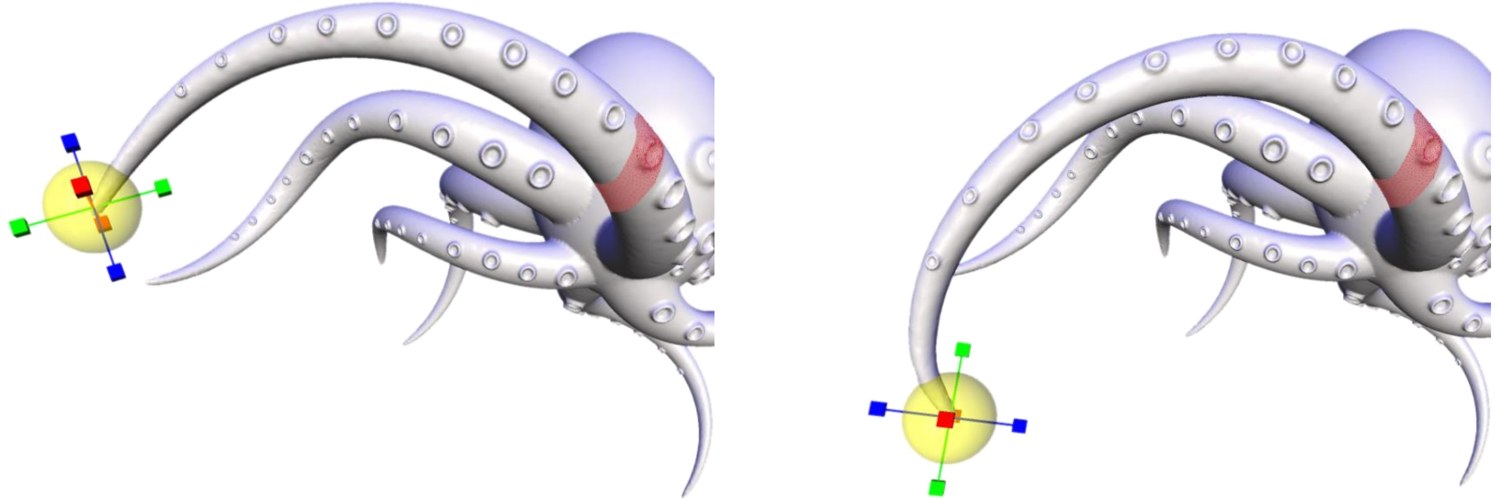
Motivation

- Meshes are great, but:
  - Geometry is represented in a *global* coordinate system
    - Single Cartesian coordinate of a vertex doesn't say much



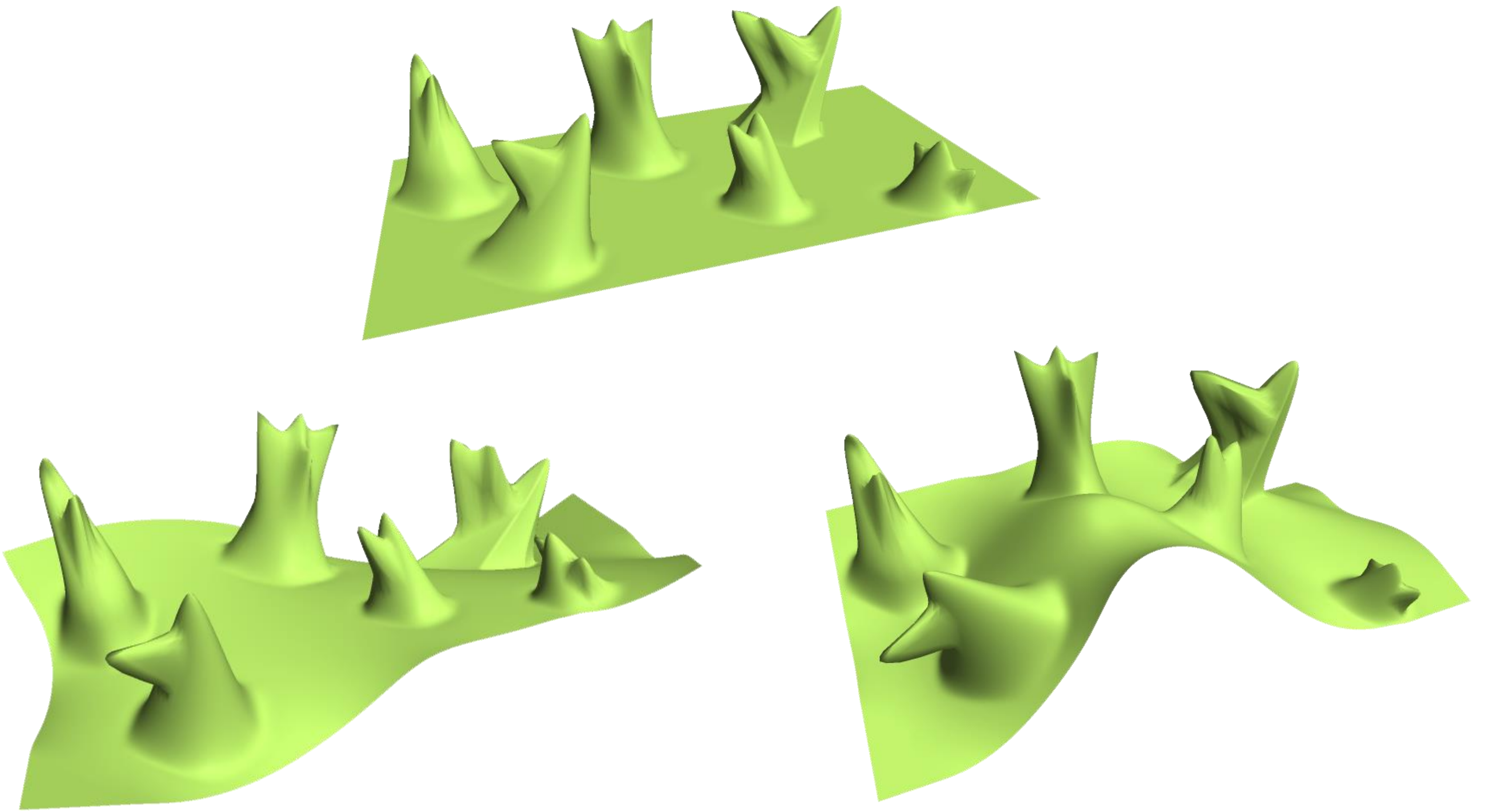
Laplacian Mesh Editing

- Meshes are difficult to edit



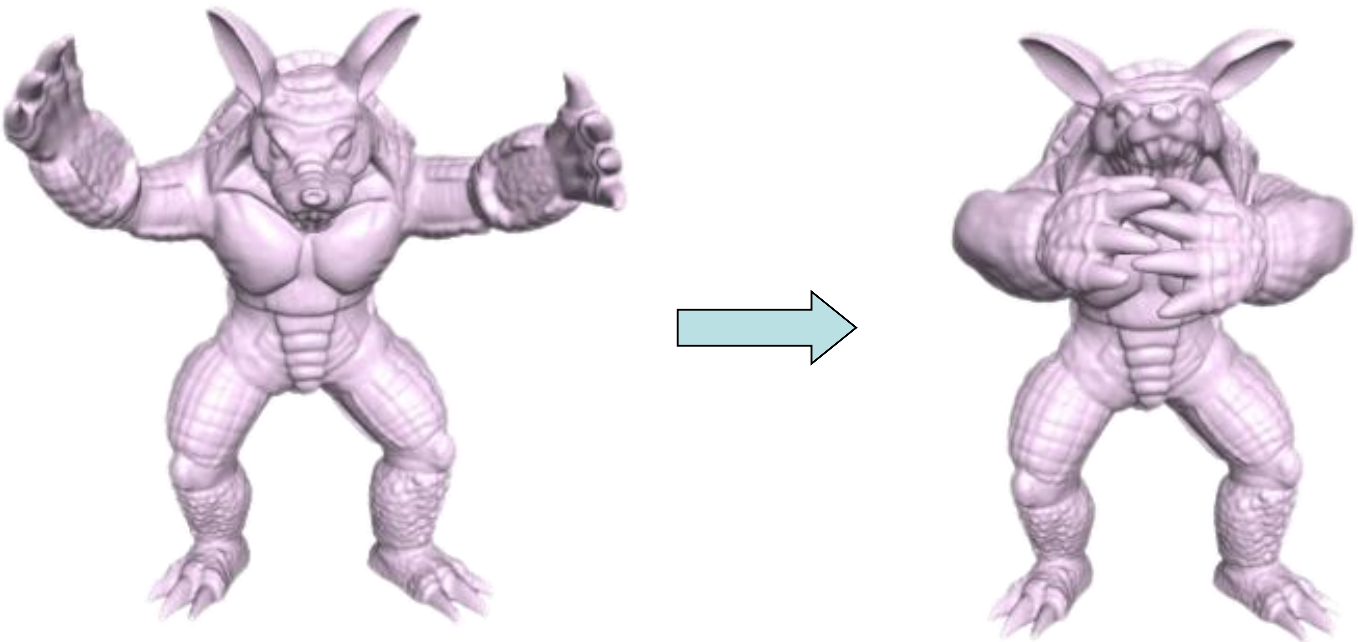
Motivation

- Meshes are difficult to edit



Motivation

- Meshes are difficult to edit



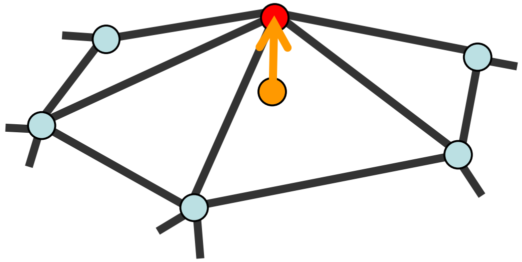
Differential coordinates

- Represent a point ***relative*** to it's neighbors.
- Represent ***local detail*** at each surface point
  - better describe the shape
- Linear transition from global to differential
- Useful for operations on surfaces where surface details are important

Differential coordinates

"Local control for mesh morphing", Alexa 01

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

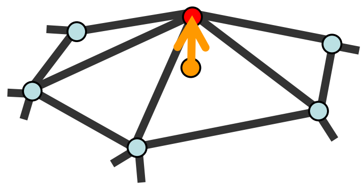


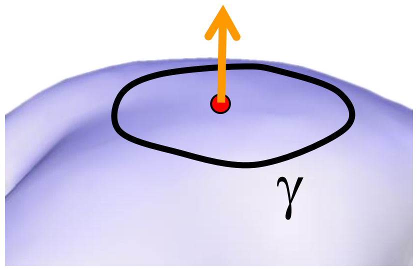
$$\boldsymbol{\delta}_i = \mathbf{v}_i - \frac{1}{d_i} \sum_{j \in N(i)} \mathbf{v}_j$$

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

Connection to the smooth case

- The direction of δ_i approximates the normal
- The size approximates the mean curvature



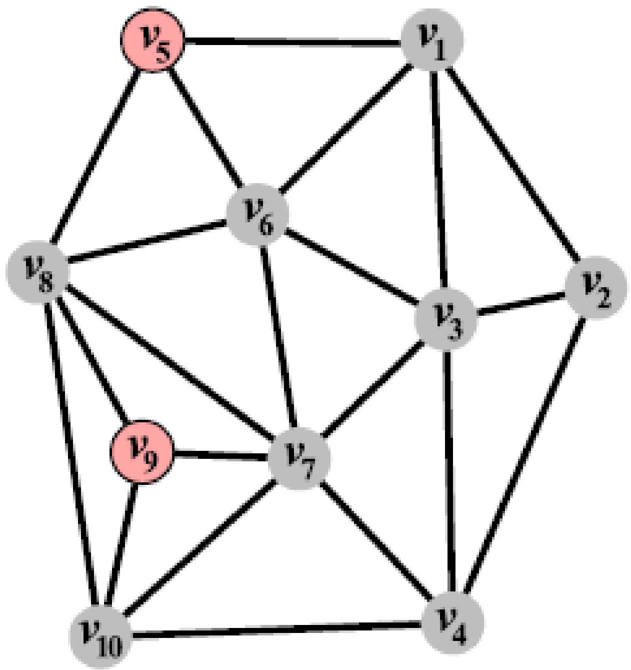


$$\boldsymbol{\delta}_{\mathbf{i}} = \frac{1}{d_i} \sum_{\mathbf{v} \in N(i)} \left(\mathbf{v}_{\mathbf{i}} - \mathbf{v} \right)$$

$$\frac{1}{len(\gamma)} \int_{\mathbf{v} \in \gamma} \left(\mathbf{v}_{\mathbf{i}} - \mathbf{v} \right) ds$$

$$\lim_{len(\gamma) \to 0} \frac{1}{len(\gamma)} \int_{\mathbf{v} \in \gamma} \left(\mathbf{v}_{\mathbf{i}} - \mathbf{v} \right) ds = H(\mathbf{v}_{\mathbf{i}}) \mathbf{n}_{\mathbf{i}}$$

Laplacian matrix



The mesh

$$\begin{bmatrix}
4 & -1 & -1 & & -1 & -1 & & & & \\
-1 & 3 & -1 & -1 & & & & & & \\
-1 & -1 & 5 & -1 & & -1 & -1 & & & \\
 & -1 & -1 & 4 & & & -1 & & & -1 \\
-1 & & & & 3 & -1 & & -1 & & \\
-1 & & -1 & & & 4 & -1 & -1 & & \\
 & & -1 & -1 & & -1 & 6 & -1 & -1 & -1 \\
 & & & & -1 & -1 & -1 & 6 & -1 & -1 \\
 & & & & & & -1 & -1 & 3 & -1 \\
 & & & -1 & & & -1 & -1 & -1 & 4
\end{bmatrix}$$

The symmetric Laplacian L_s

Weighting schemes

$$\delta_i = \frac{\sum_{j\in N(i)} w_{ij}\left(\mathbf{v}_i - \mathbf{v}_j\right)}{\sum_{j\in N(i)} w_{ij}}$$

- Ignore geometry

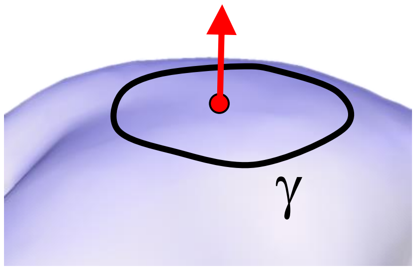
  $\delta_{\textbf{umbrella}}$: $w_{ij} = 1$

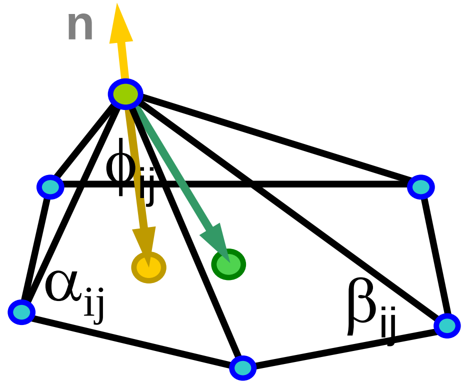
- Integrate over circle around vertex

  $\delta_{\textbf{mean value}}$: $w_{ij} = \tan \phi_{ij}/2 + \tan \phi_{ij+1}/2$

- Integrate over Voronoi region of vertex

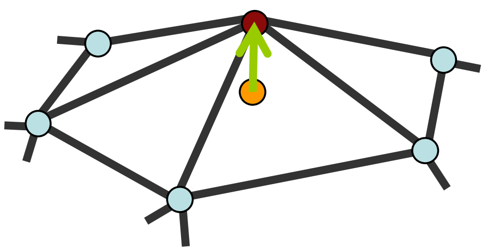
  $\delta_{\textbf{cotangent}}$: $w_{ij} = \cot \alpha_{ij} + \cot \beta_{ij}$





Laplacian mesh

- Vertex positions are represented by Laplacian coordinates (δ_x δ_y δ_z)



$$\boldsymbol{\delta}_i = \sum_{j \in N(i)} w_{ij} \left(\mathbf{v}_i - \mathbf{v}_j \right)$$

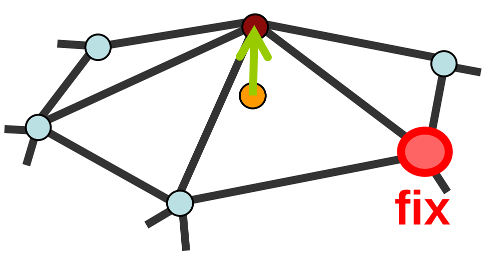
$$\mathbf{L} \, \mathbf{v}_\mathbf{x} = \delta_\mathbf{x}$$

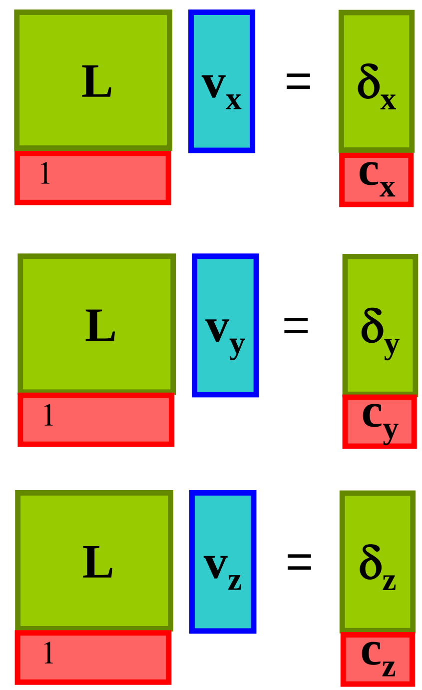
$$\mathbf{L} \, \mathbf{v}_\mathbf{y} = \delta_\mathbf{y}$$

$$\mathbf{L} \, \mathbf{v}_\mathbf{z} = \delta_\mathbf{z}$$

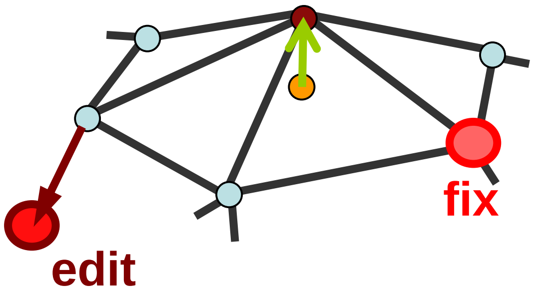
Basic properties

- $\mathrm{rank}(L) = n - c$ ($n - 1$ for connected meshes)
- We can reconstruct the xyz geometry from δ up to translation





Reconstruction

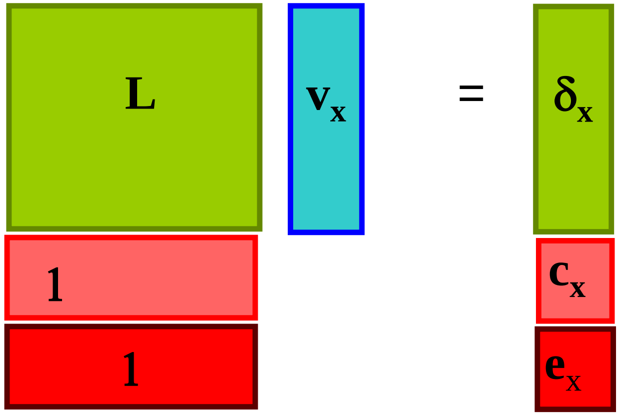


$$\begin{bmatrix} L \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} v_x \end{bmatrix} = \begin{bmatrix} \delta_x \\ c_x \\ e_x \end{bmatrix}$$

$$\begin{bmatrix} L \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} v_y \end{bmatrix} = \begin{bmatrix} \delta_y \\ c_y \\ e_y \end{bmatrix}$$

$$\begin{bmatrix} L \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} v_z \end{bmatrix} = \begin{bmatrix} \delta_z \\ c_z \\ e_z \end{bmatrix}$$

Reconstruction



$$\boxed{\tilde{\mathbf{x}} = \arg\min_{\mathbf{x}} \left(\left\| L\mathbf{x} - \boldsymbol{\delta}_x \right\|^2 + \sum_{s=1}^{k} \left| x_k - c_k \right|^2 \right)}$$

Reconstruction

$$\begin{bmatrix} L \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} \mathbf{v}_x \end{bmatrix} = \begin{bmatrix} \delta_x \\ \mathbf{c}_x \\ \mathbf{e}_x \end{bmatrix}$$

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

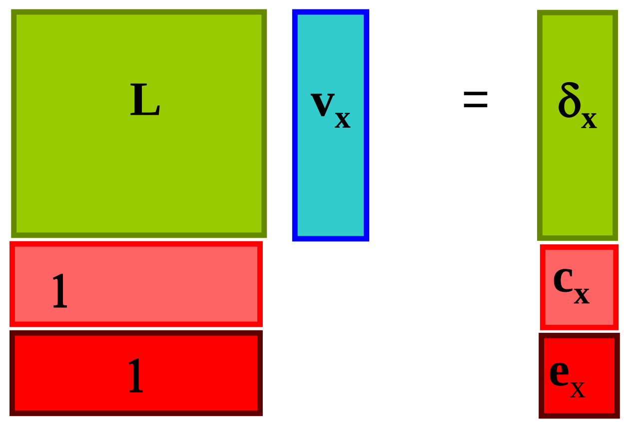
Normal Equations:

$$\mathbf{A}^T\mathbf{A} \ \mathbf{x} = \mathbf{A}^T \mathbf{b}$$

$$\mathbf{x} = \underbrace{(\mathbf{A}^T\mathbf{A})^{-1}}_{\text{compute once}} \ \mathbf{A}^T \mathbf{b}$$

Cool underlying idea

- Mesh vertex positions are defined by minimizer of an objective function



$$\tilde{\mathbf{x}} = \arg\min_{\mathbf{x}} \left(\left\| L\mathbf{x} - \boldsymbol{\delta}_x \right\|^2 + \sum_{s=1}^{k} \left| x_k - c_k \right|^2 \right)$$

What we have so far

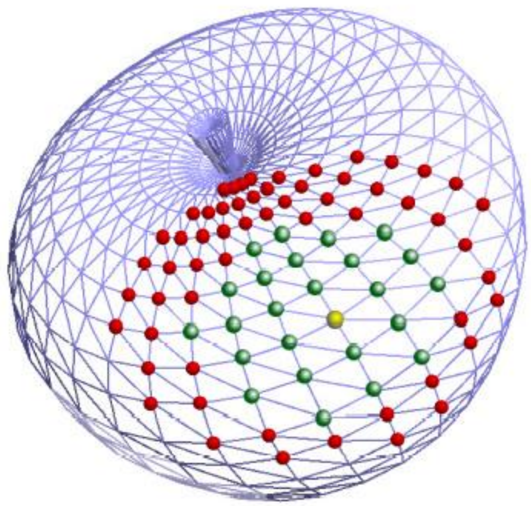
- Laplacian coordinates δ
  - Local representation
  - Translation-invariant
- Linear transition from δ to xyz
  - can constrain more that 1 vertex
  - least-squares solution

Editing using differential coordinates

- The editing process from the user's point of view:

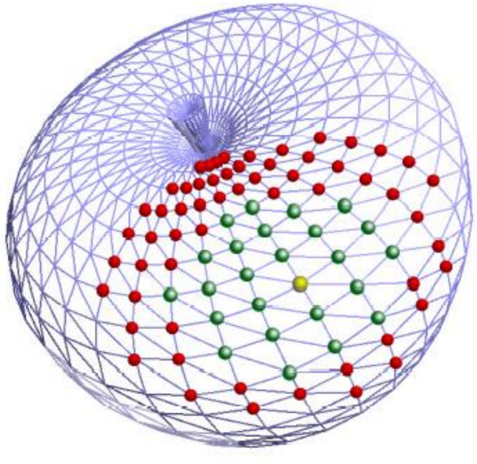
1) First, a ROI , anchors and a handle vertex should be set.

2) Then the edit is Performed By moving this vertex.

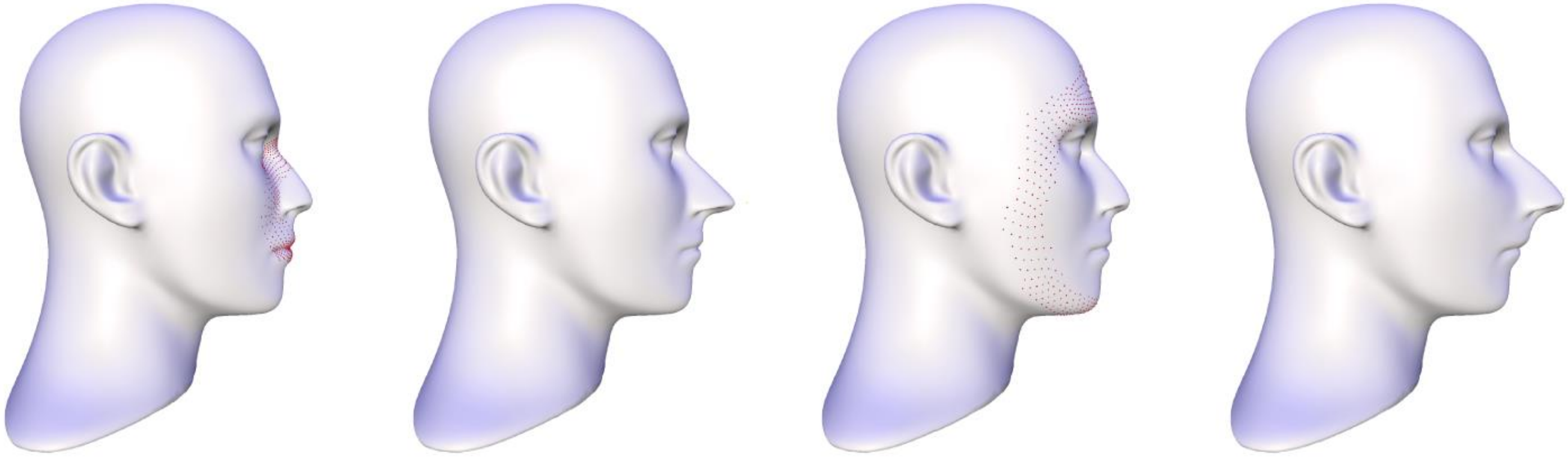


Editing using differential coordinates

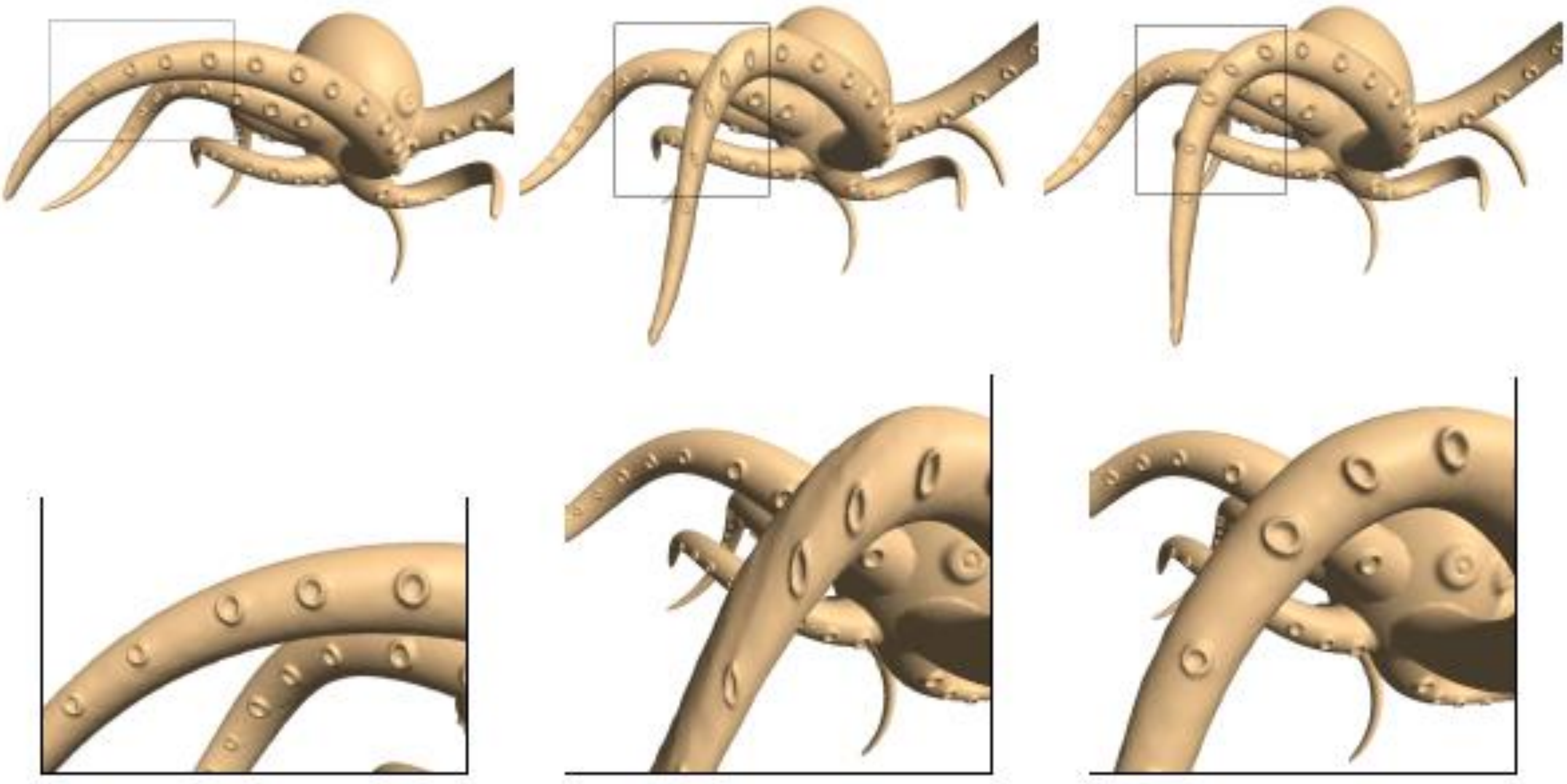
- The user moves the handle and interactively the surface changes.
- The stationary anchors are responsible for smooth transition of the edited part to the rest of the mesh.
- This is done using increasing weight with geodesic distance in the soft spatial equations.



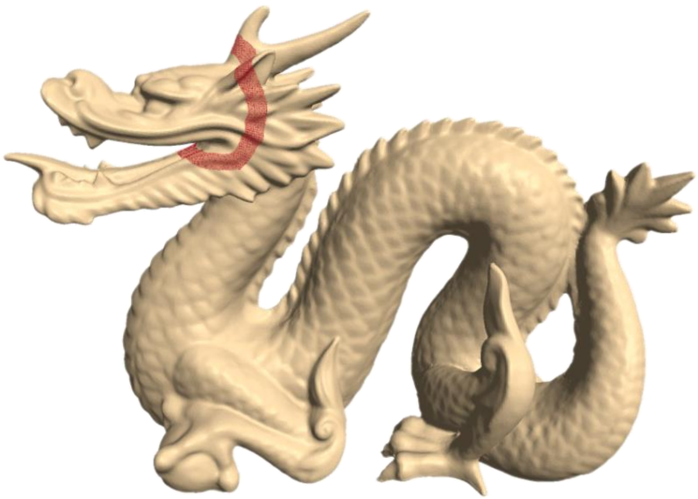
Mesh Editing Example

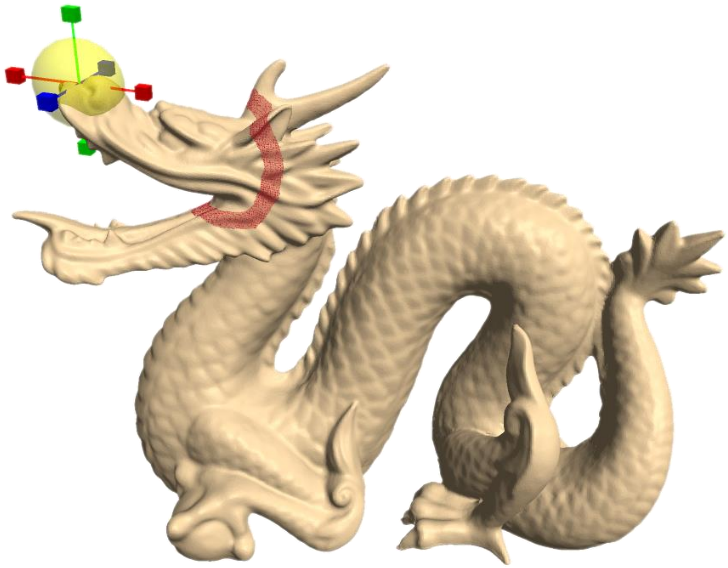


Mesh Editing Example

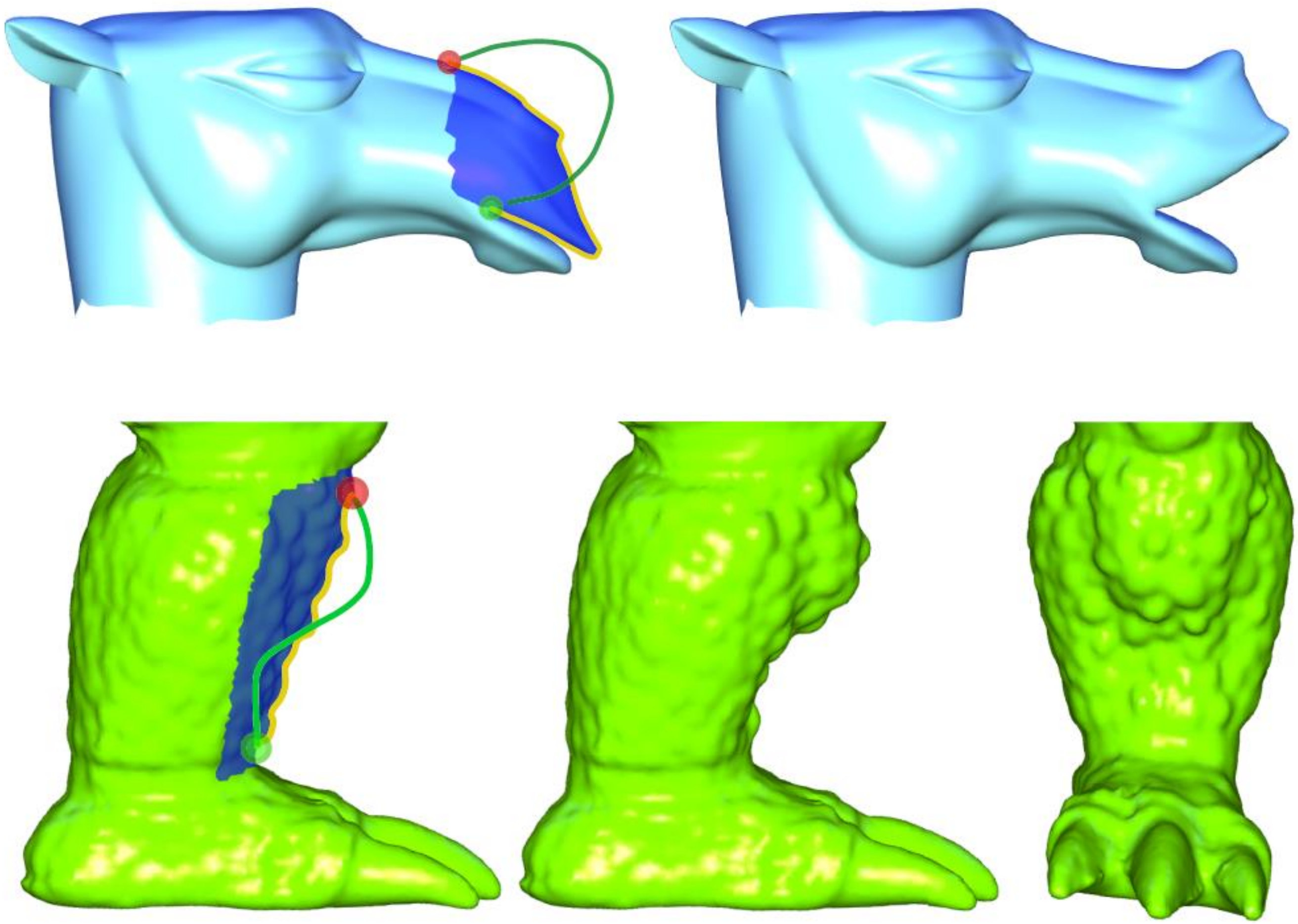


Mesh Editing Example





Mesh Editing Example

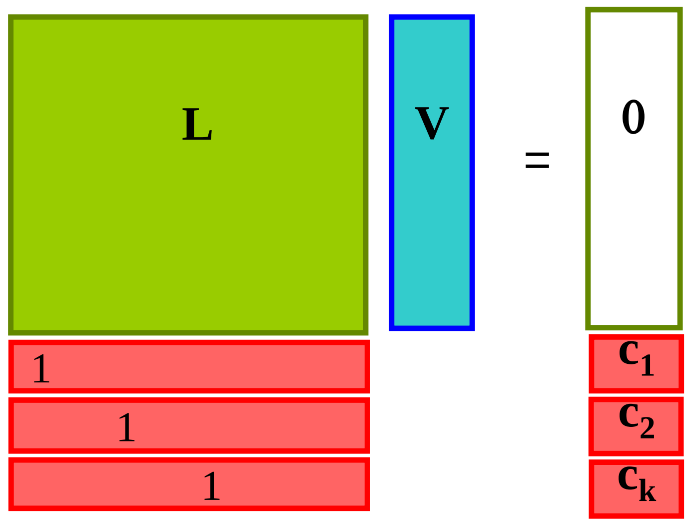


What else can we do with it?

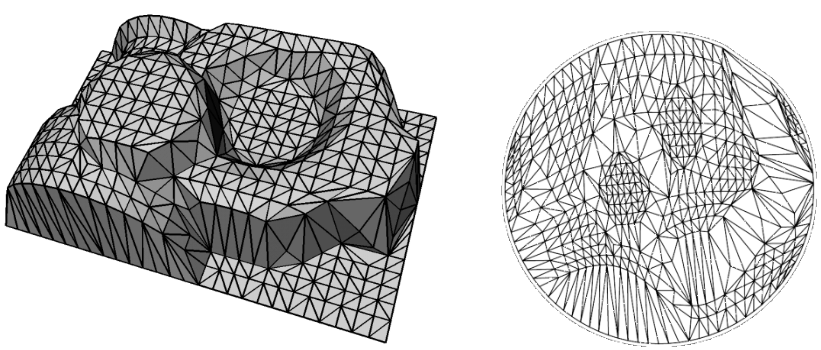
▪

Parameterization

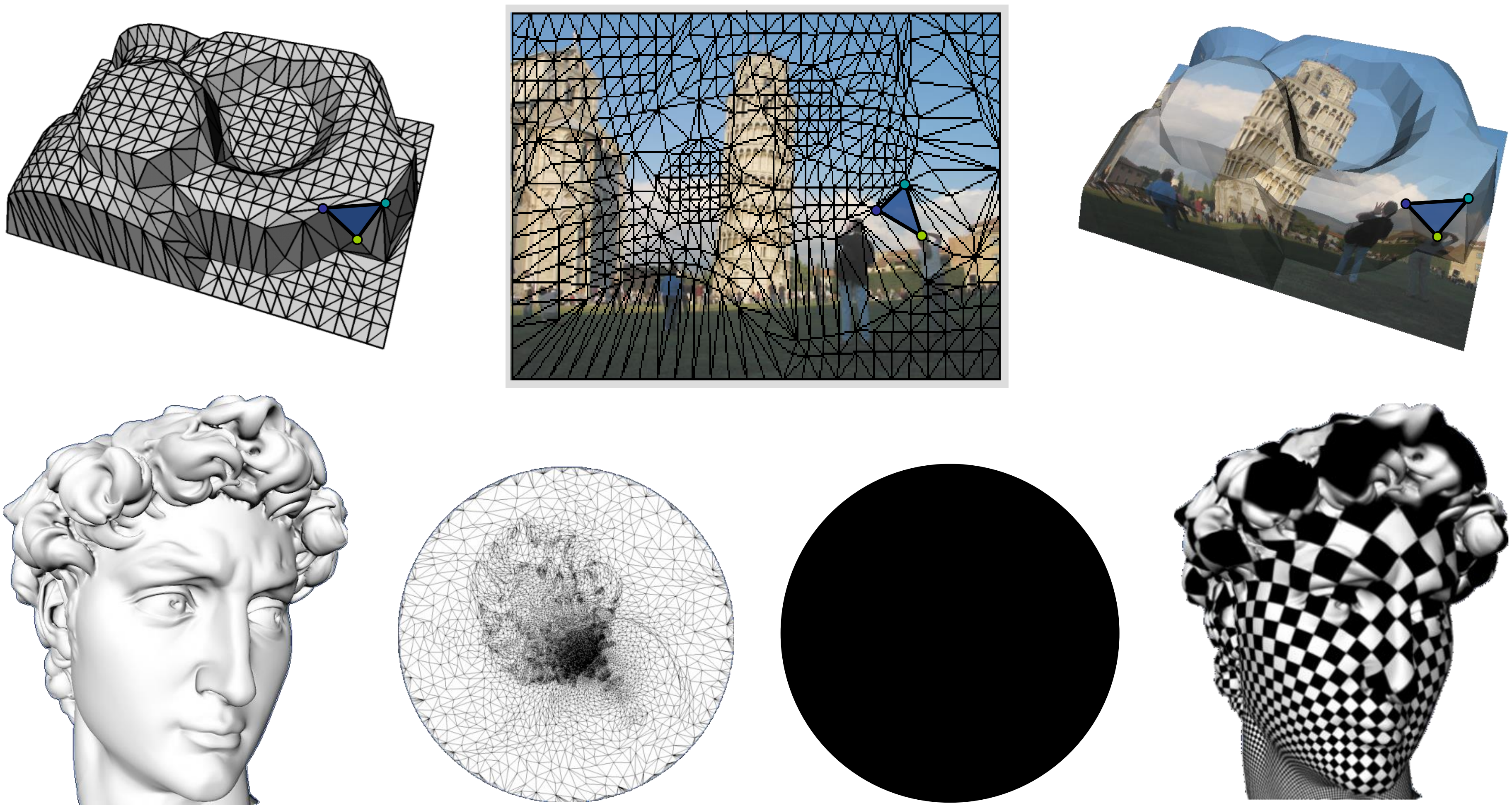
- Use zero Laplacians.



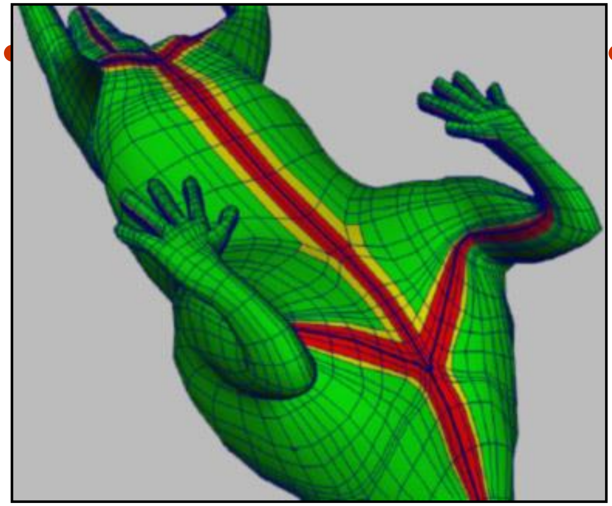
In 2D:

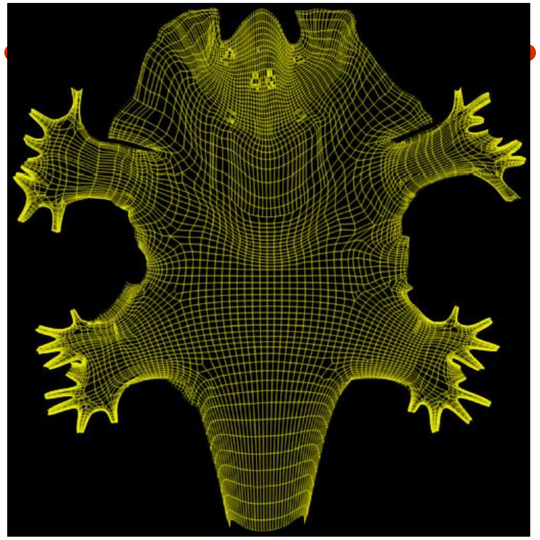


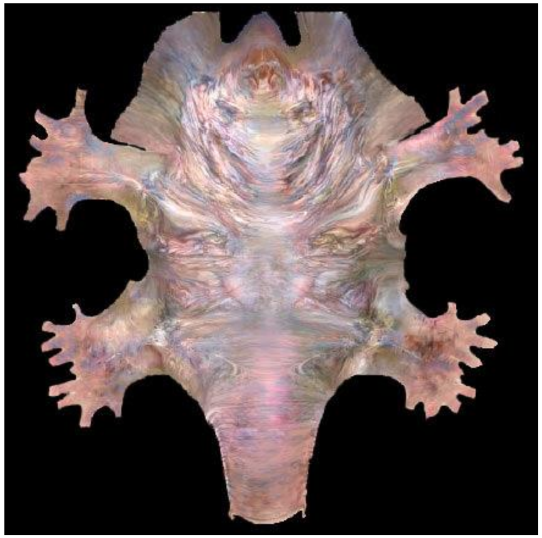
Texture Mapping



Texture Mapping



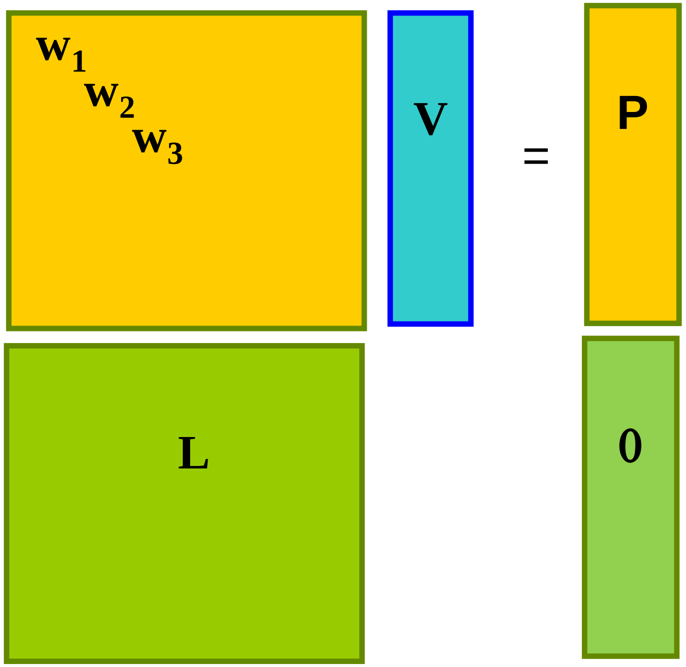




[Piponi2000]

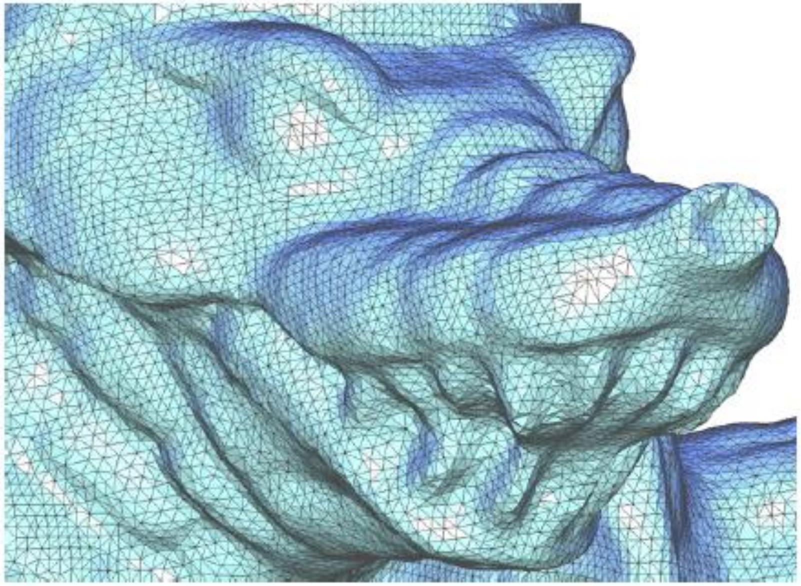
Feature Preserving Smoothing

- Weighted positional and smoothing constraints

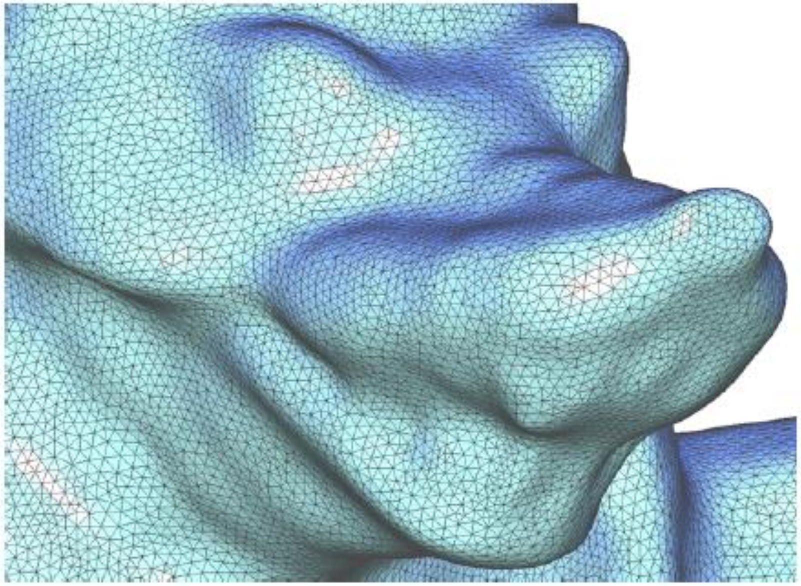


Feature Preserving Smoothing

- Weighted positional and smoothing constraints



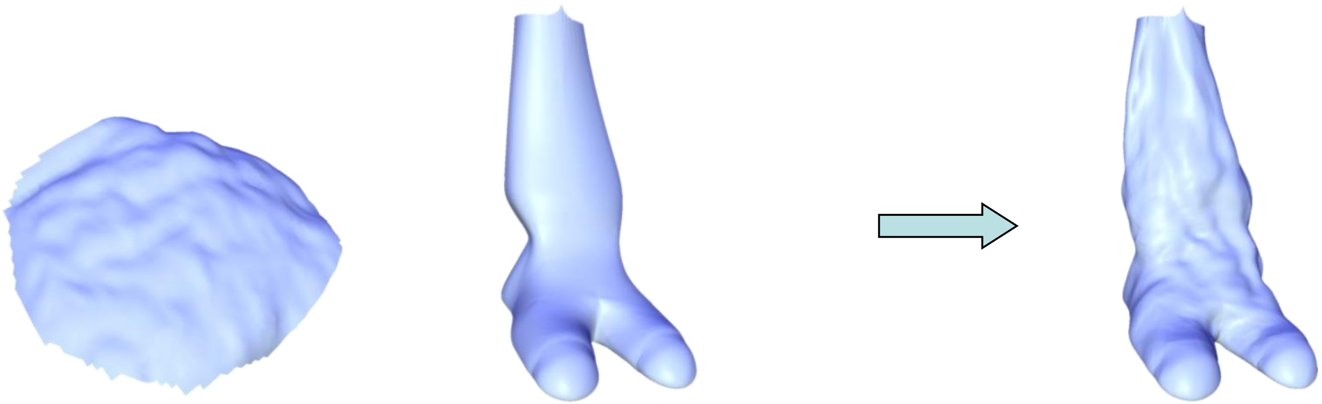
Original



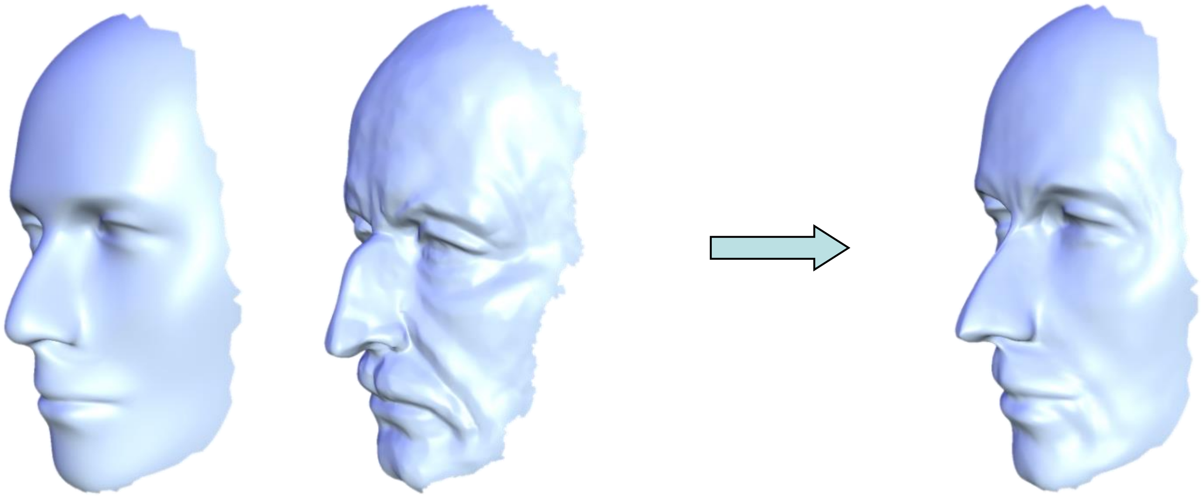
Smoothed

Detail transfer

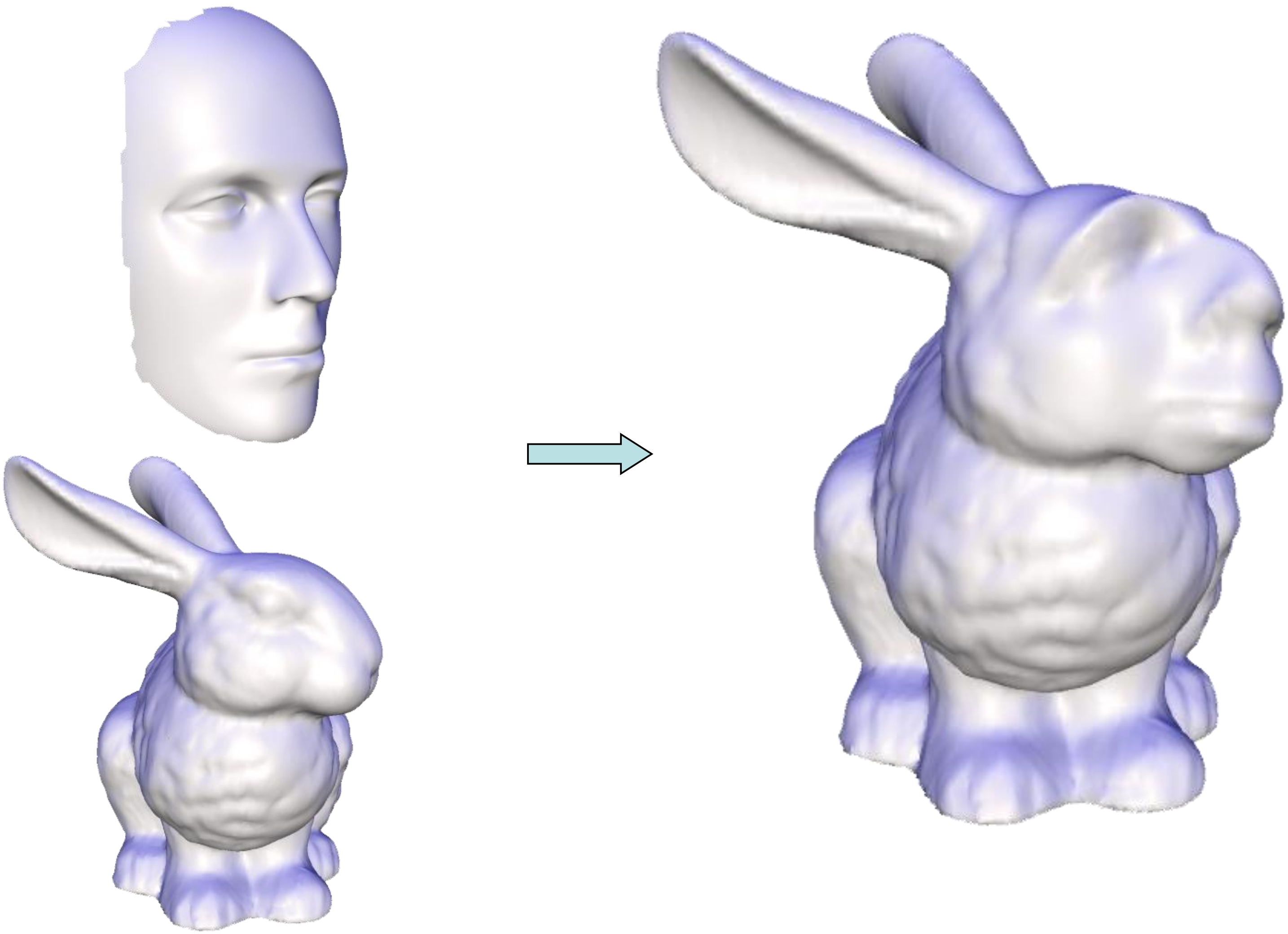
- “Peel“ the coating of one surface and transfer to another



Detail transfer

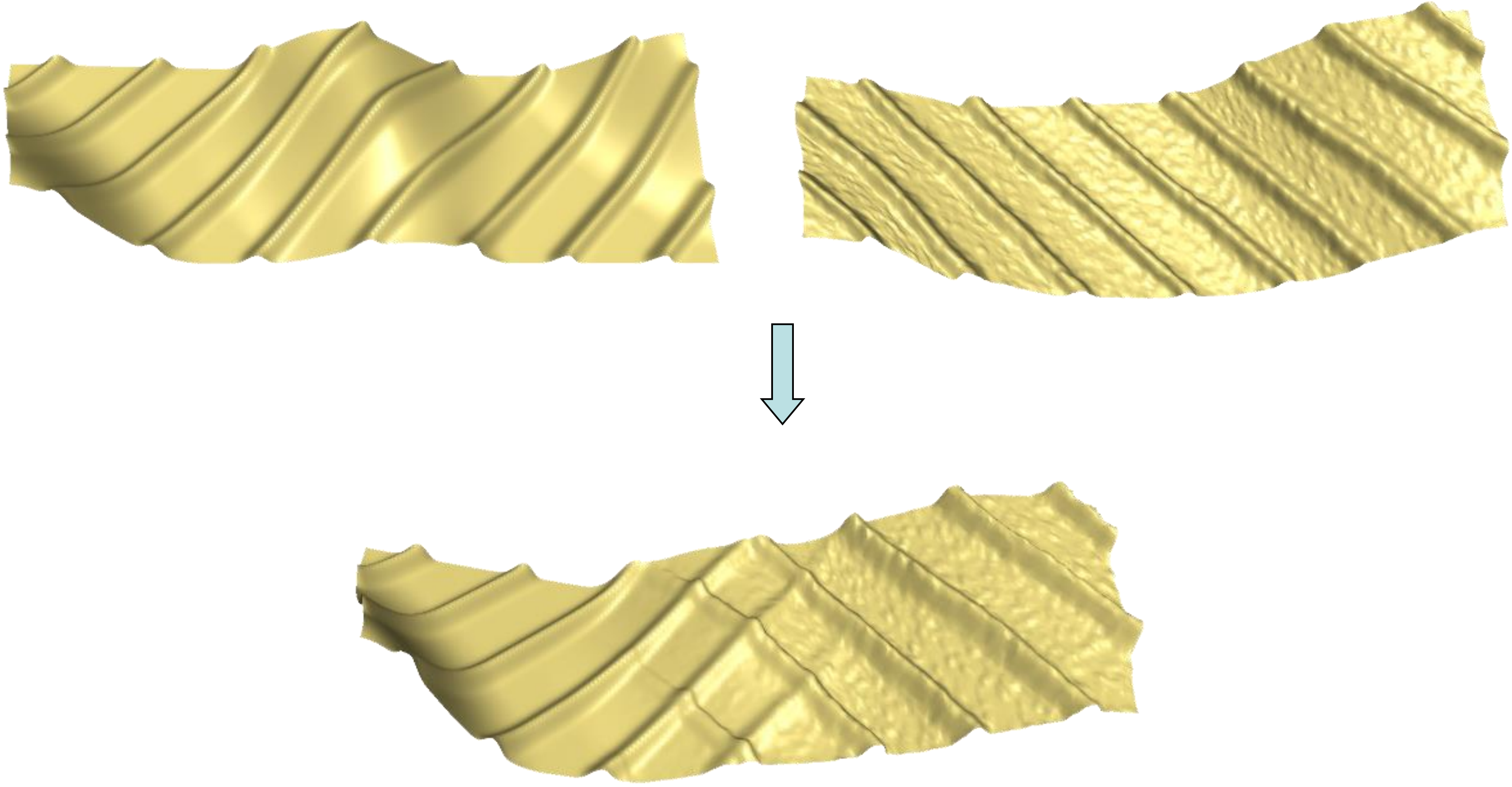


Detail transfer



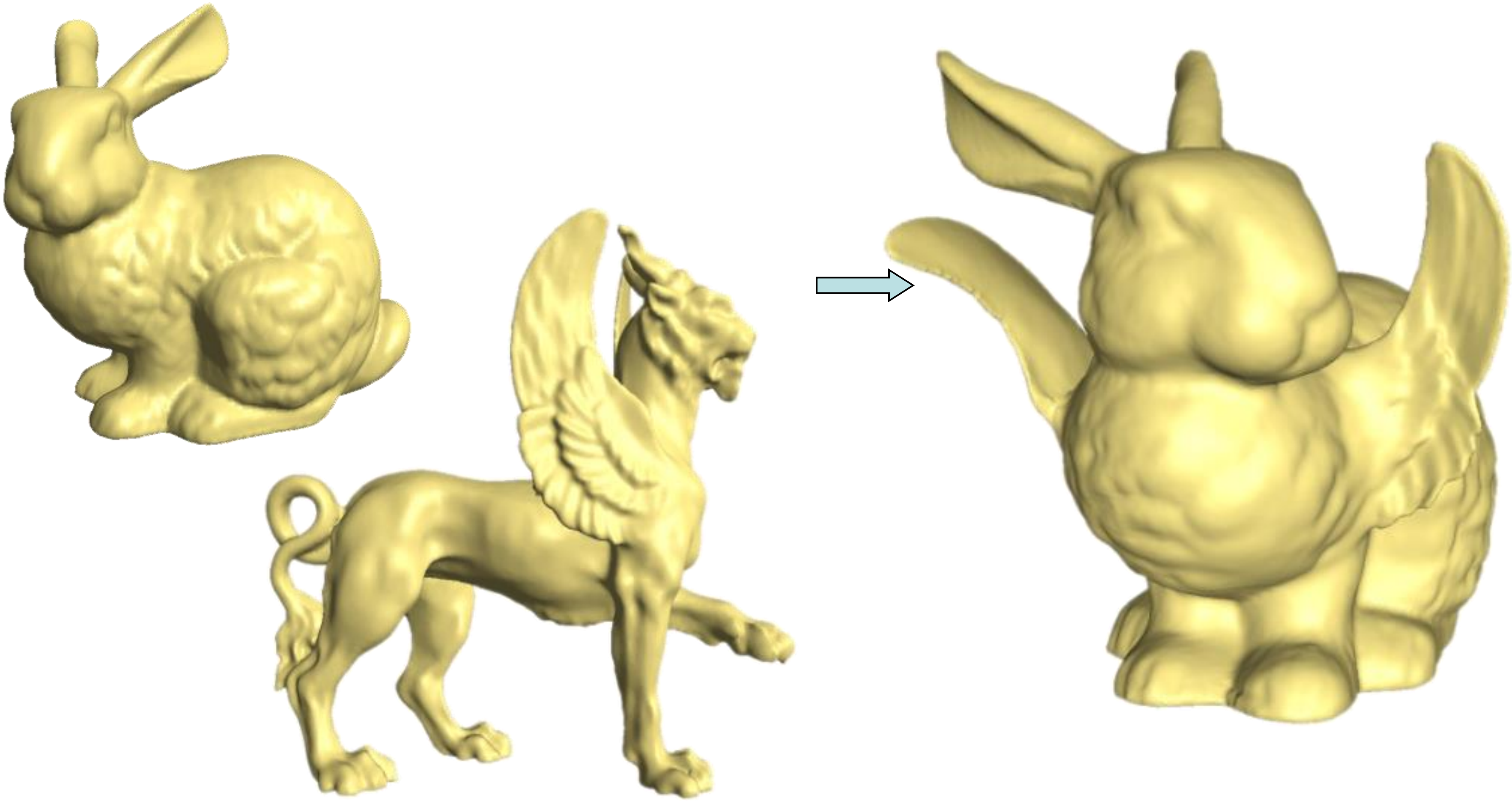
Mixing Laplacians

- Taking weighted average of δ_i and δ'_i



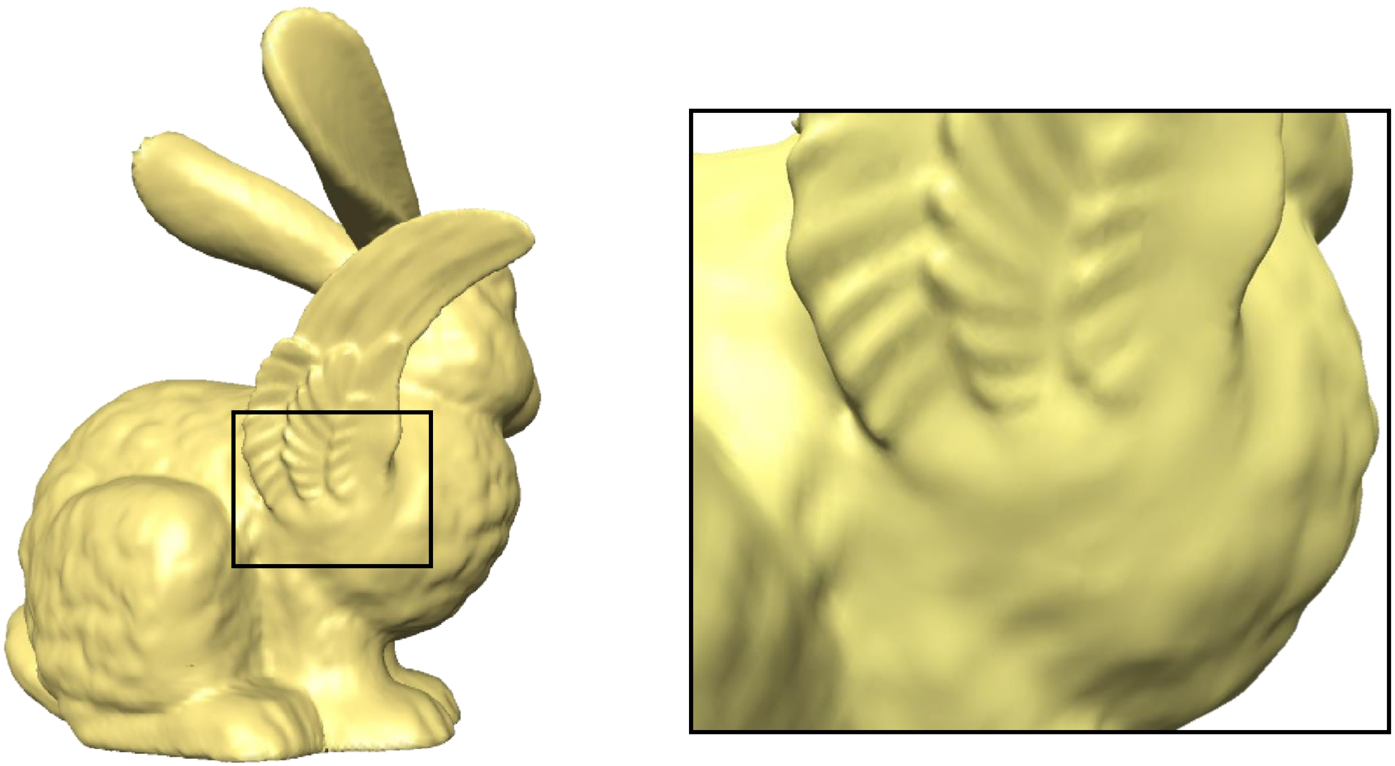
Mesh transplanting

- Geometrical stitching via Laplacian mixing



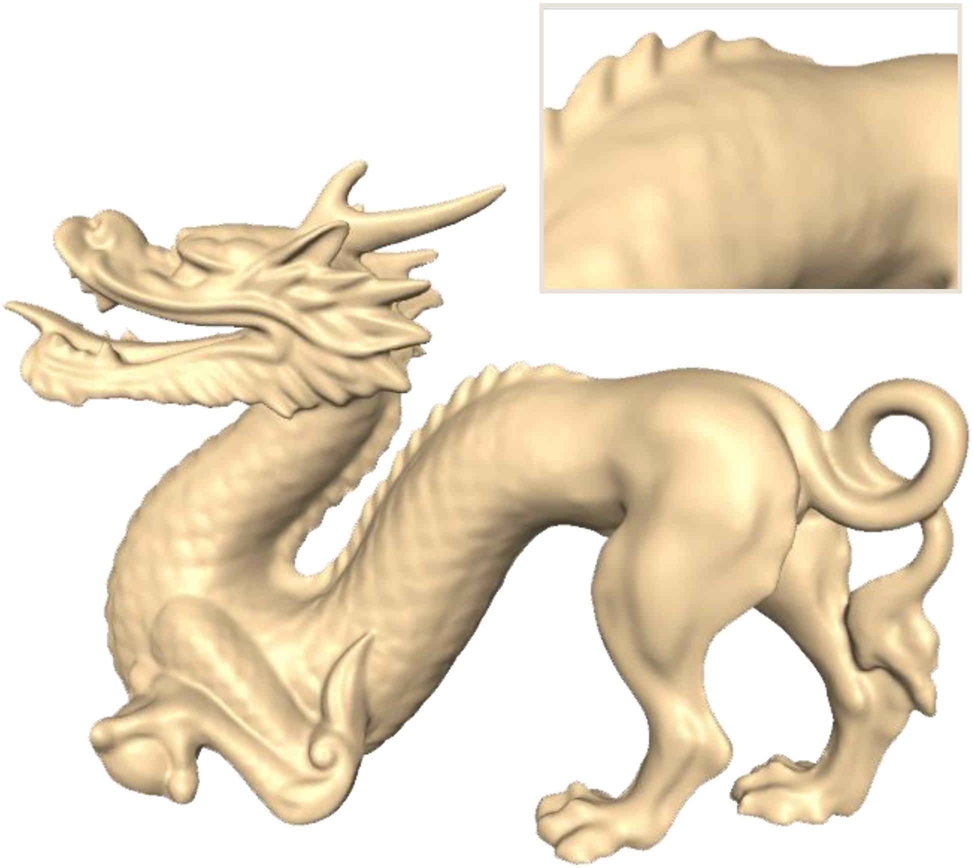
Mesh transplanting

- Details gradually change in the transition area



Mesh transplanting

- Details gradually change in the transition area



The End