Laplacian Mesh Representation and Editing

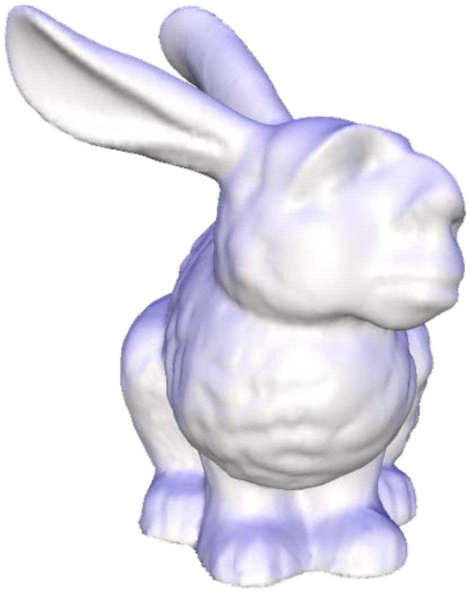
COS 526: Advanced Computer Graphics

PRINCETON UNIVERSITY

Slide credits: Olga Sorkine and Yaron Lipman

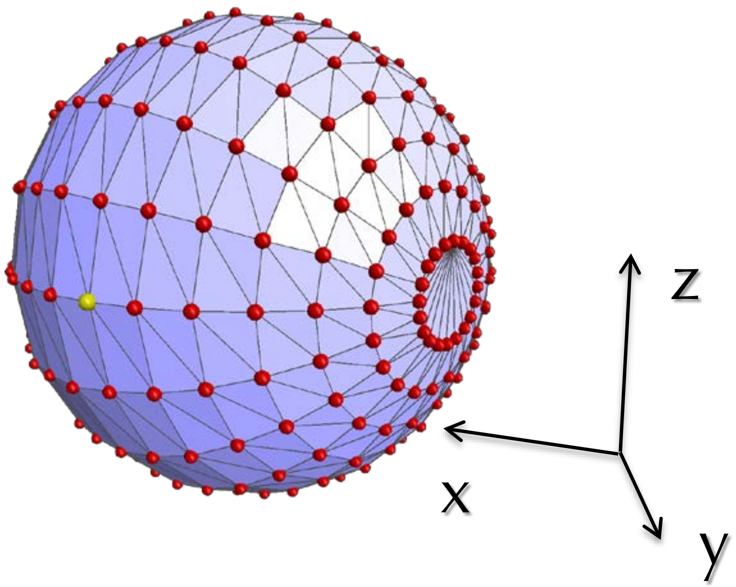
Outline

- Differential surface representation
- Ideas and applications
  - Compact shape representation
  - Mesh editing and manipulation
  - Membrane and flattening



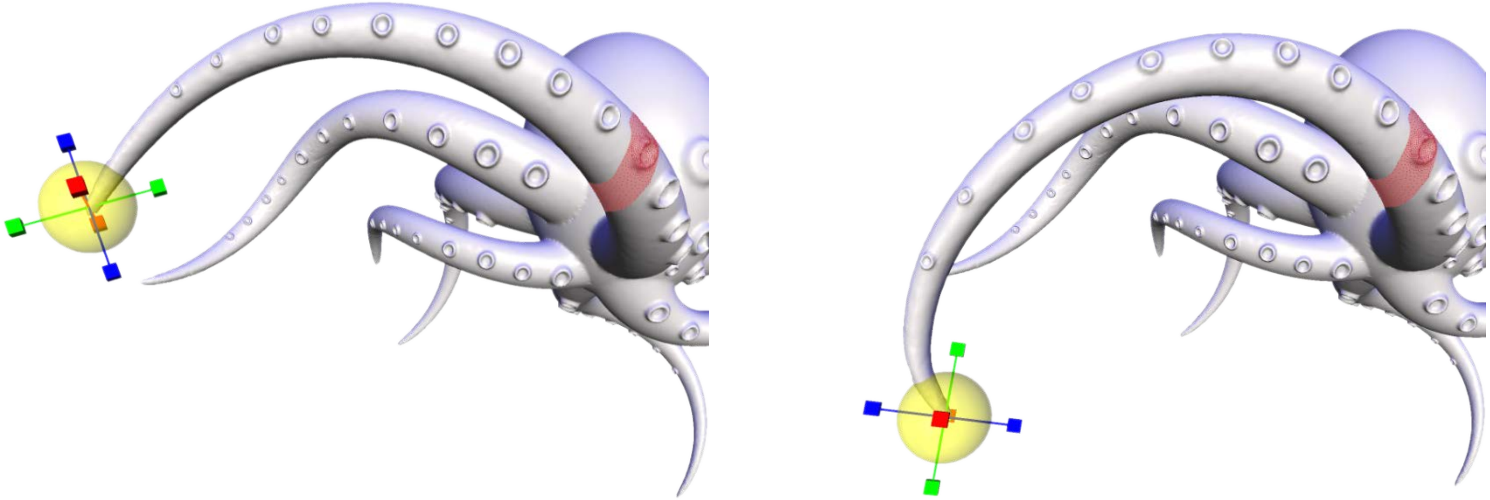
Motivation

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



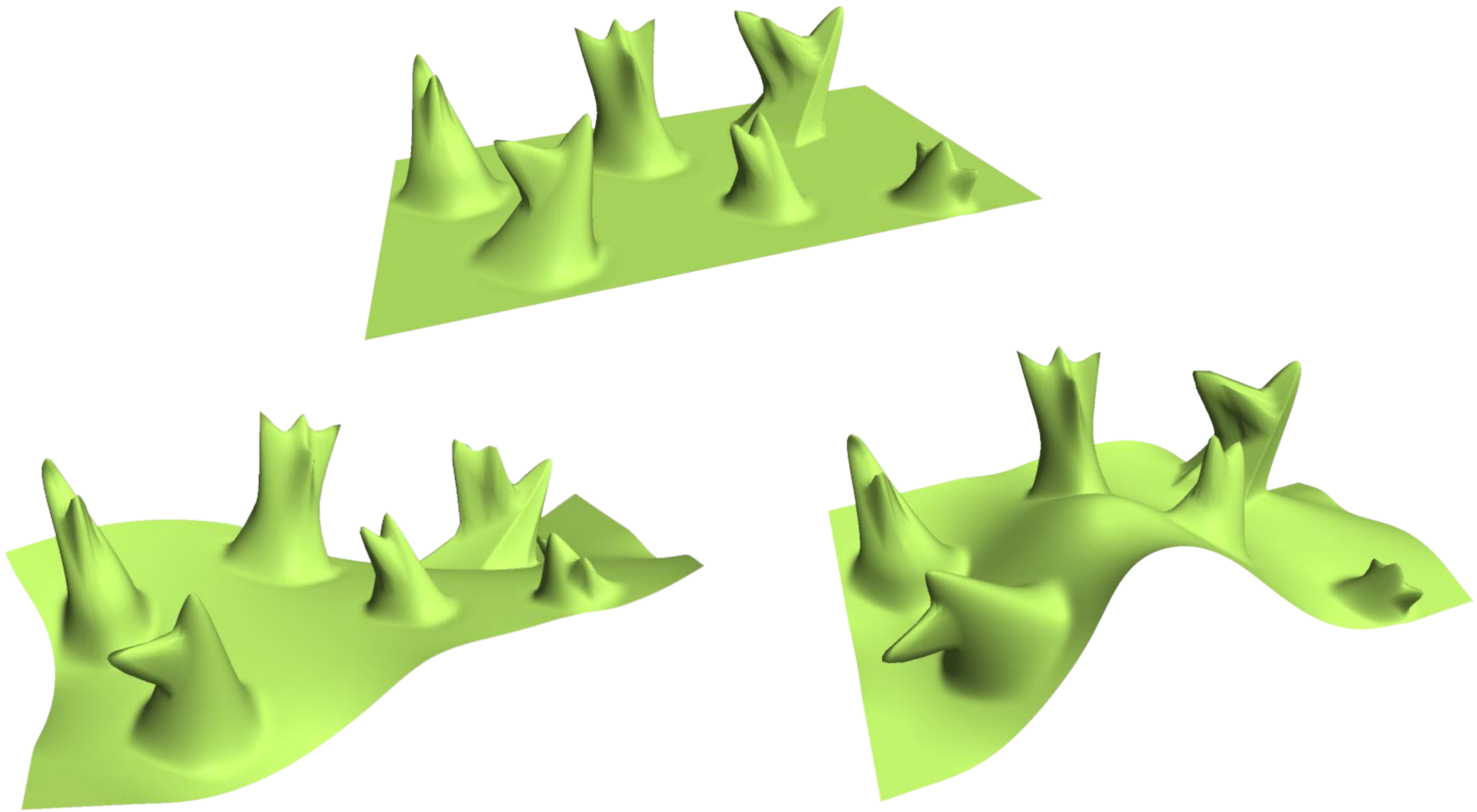
Laplacian Mesh Editing

- Meshes are difficult to edit



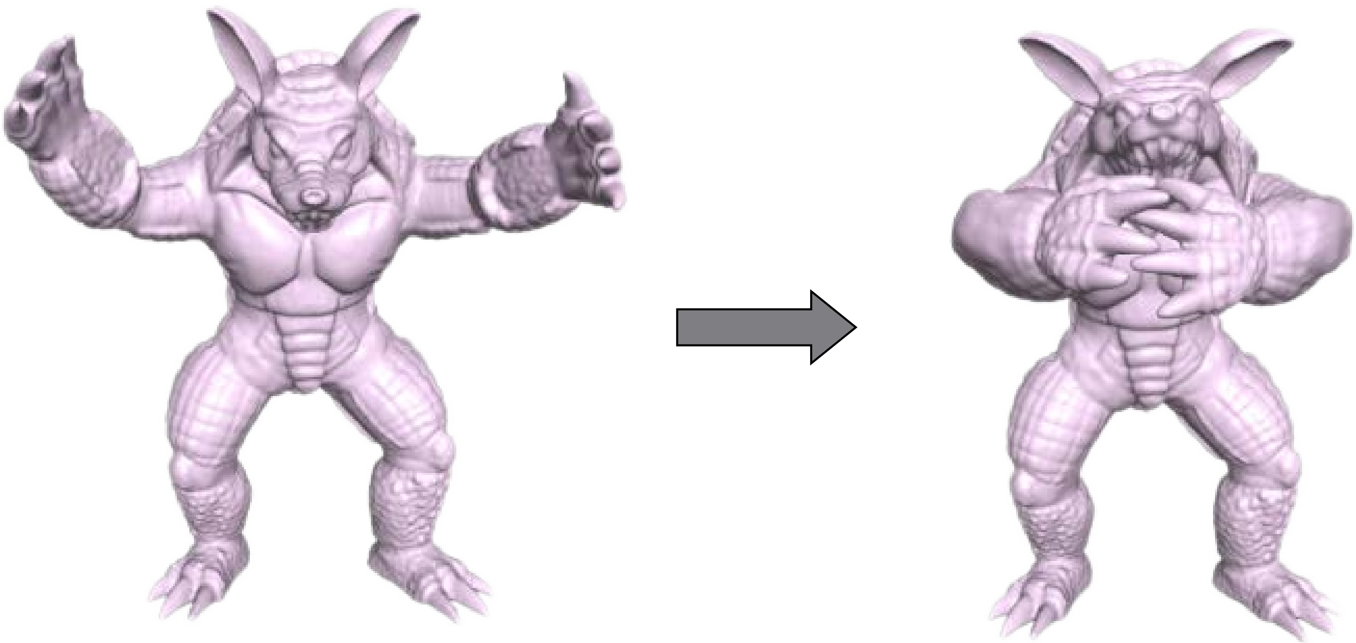
Motivation

- Meshes are difficult to edit



Motivation

- Meshes are difficult to edit

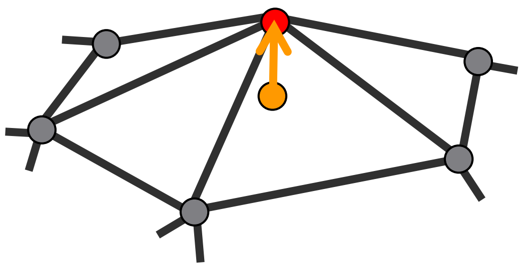


Differential Coordinates

- Represent a point ***relative*** to its 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

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



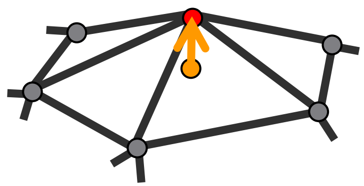
$$\boldsymbol{\delta}_i = \mathbf{v}_i - \frac{1}{d_i} \sum_{j \in N(i)} \mathbf{v}_j$$

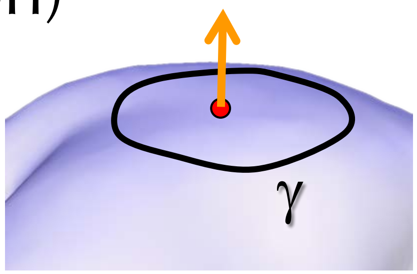
$$\boldsymbol{\delta}_i = \sum_{j \in N(i)} \frac{1}{d_i} \left(\mathbf{v}_i - \mathbf{v}_j \right)$$

"Local control for mesh morphing", Alexa 01

Connection to the Smooth Case

- The direction of $\boldsymbol{\delta}_i$ approximates the normal
- The size approximates the mean curvature H
  - 1 / radius of local best-fit sphere
- Laplace-Beltrami operator on surface (like Laplacian of a 2D function)





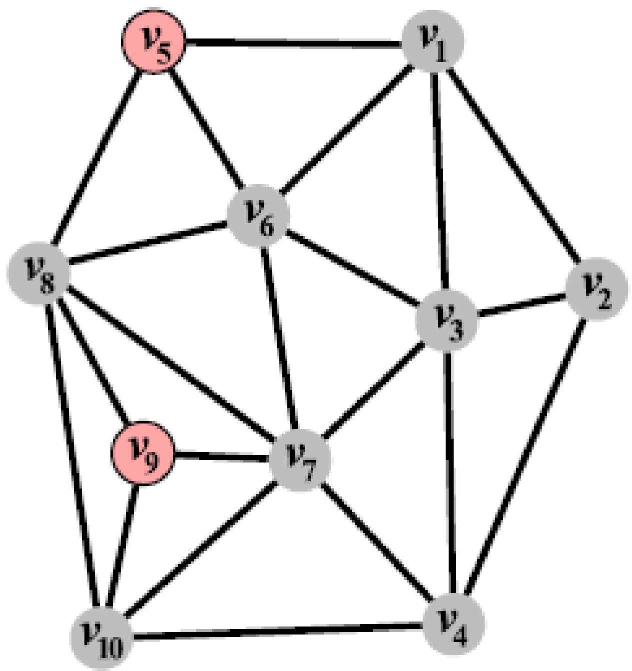
$$\boldsymbol{\delta}_\mathbf{i} = \frac{1}{d_i} \sum_{\mathbf{v} \in N(i)} (\mathbf{v}_\mathbf{i} - \mathbf{v})$$

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

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

Laplacian Matrix

- Coefficient of each vertex in computation of Laplacian at every other vertex



The mesh

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

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

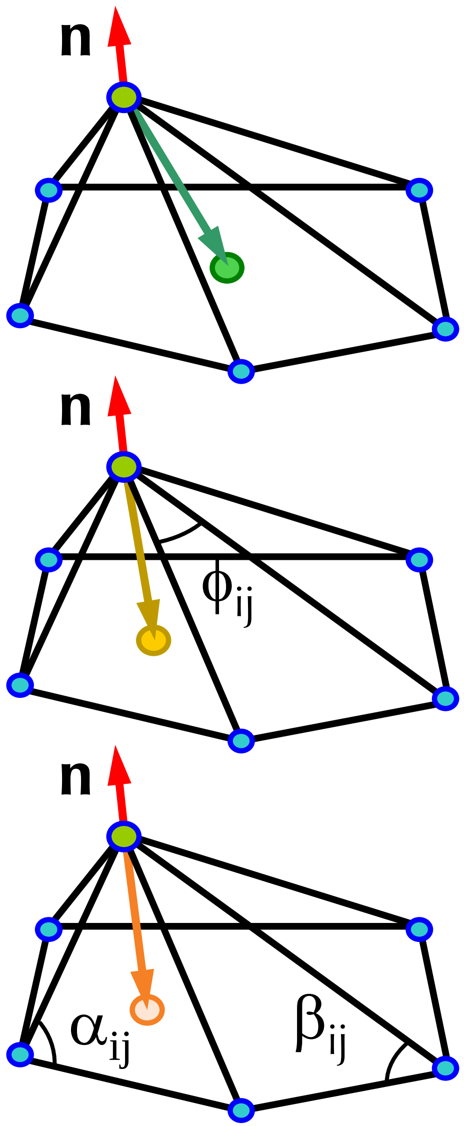
  $\delta_{\mathbf{umbrella}} : w_{ij} = 1$

- Integrate over circle around vertex

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

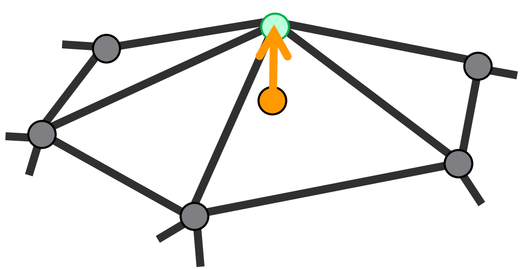
- Integrate over Voronoi region of vertex

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



Laplacian Mesh Representation

- Vertex positions are represented by Laplacian coordinates $(\delta_x, \delta_y, \delta_z)$



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

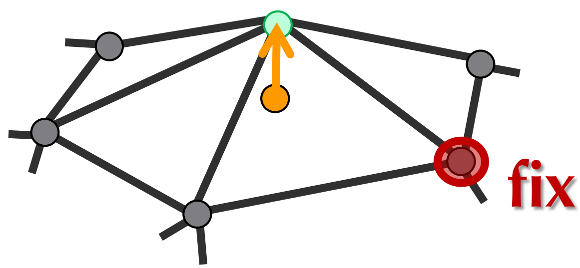
$$\mathbf{L} \, \mathbf{v_x} = \boldsymbol{\delta}_\mathbf{x}$$

$$\mathbf{L} \, \mathbf{v_y} = \boldsymbol{\delta}_\mathbf{y}$$

$$\mathbf{L} \, \mathbf{v_z} = \boldsymbol{\delta}_\mathbf{z}$$

Basic Properties

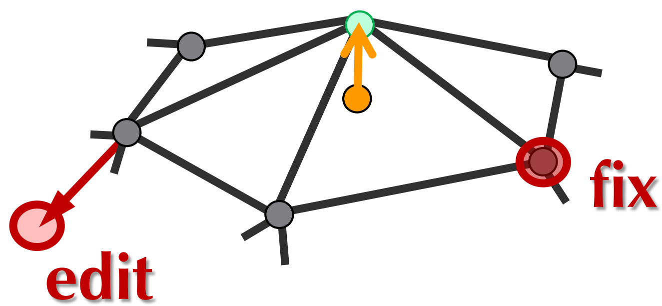
- rank(L) = n – c   (n – 1 for connected meshes)
- Can reconstruct geometry from δ up to translation
  - Add constraint on one vertex for unique solution



Reconstruction

- Constrain additional vertices: overdetermined system

$$\arg\min_{x}\left(\|Lx-\delta_x\|^2+\sum_{k=1}^{n_c}\|x_k-c_k\|^2\right)$$



- Cool underlying idea: shape defined as minimizer of an objective function

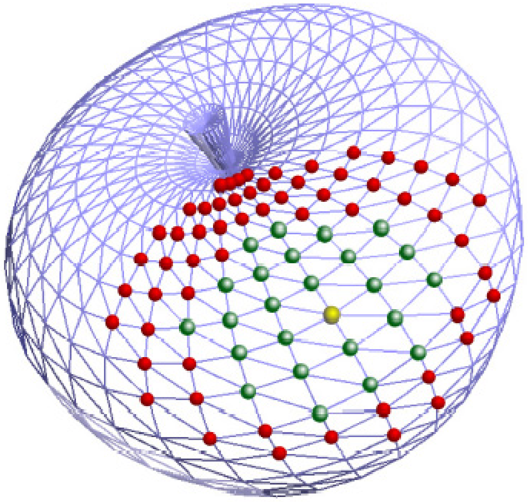
So Far…

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

Editing Using Laplacian Coordinates

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

1. Set ROI, anchors, and a *handle* vertex
2. Move the handle, interactively see effect on mesh



Editing Using Laplacian Coordinates

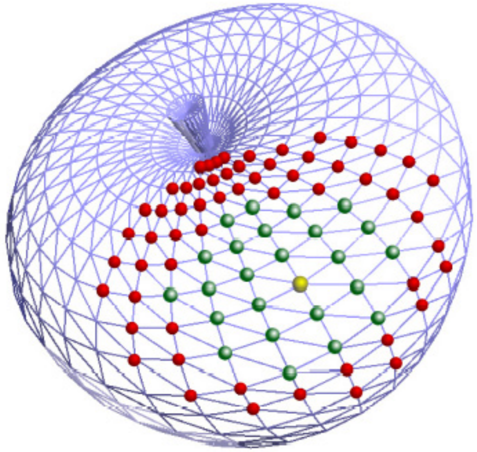
Behind the scenes…

- ROI defines vertices that are included in the solve
- Constraints at anchors: responsible for smooth transition of the edited part to the rest of the mesh
  - Increasing weight with distance away from handle
- Precomputation enables interactivity

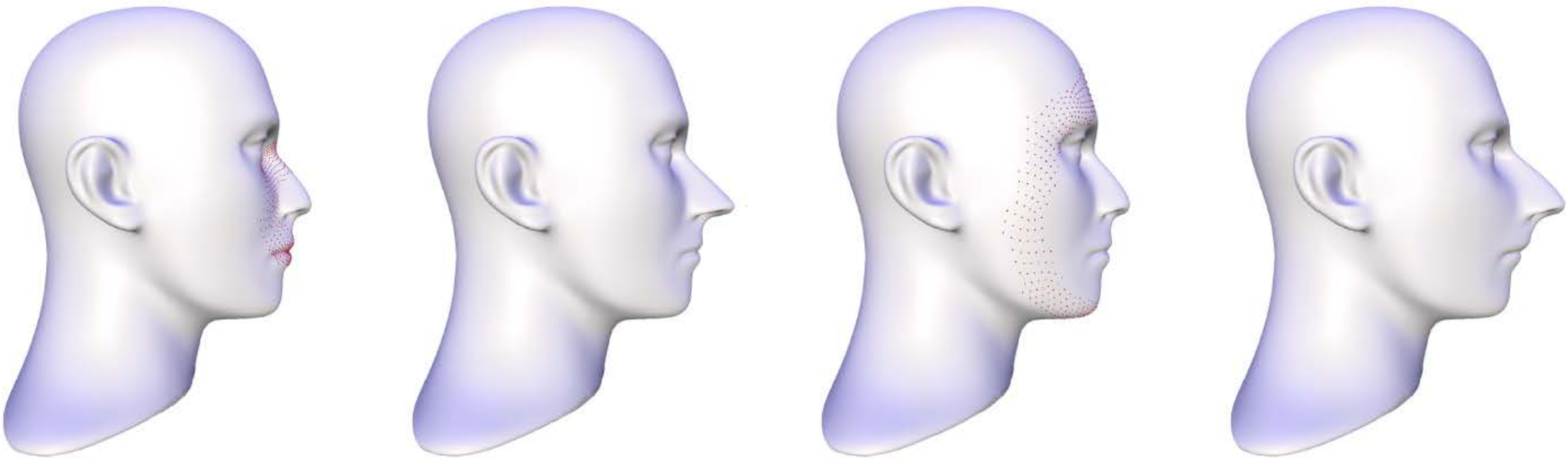
$$\mathbf{A}\,\mathbf{x} = \mathbf{b}$$

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

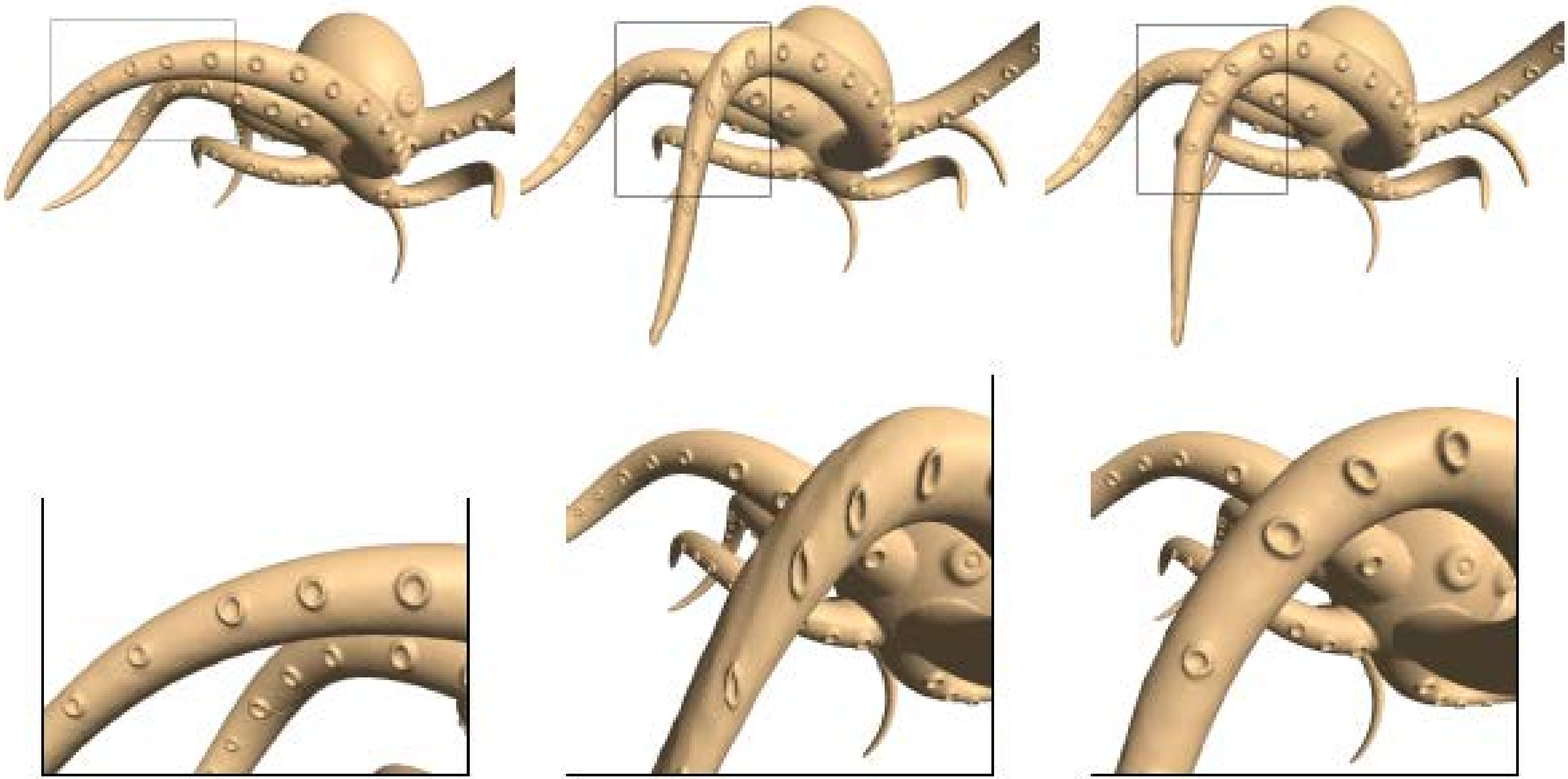
$$\mathbf{x} = \underbrace{(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T}_{\text{compute once}}\mathbf{b}$$



Mesh Editing Example

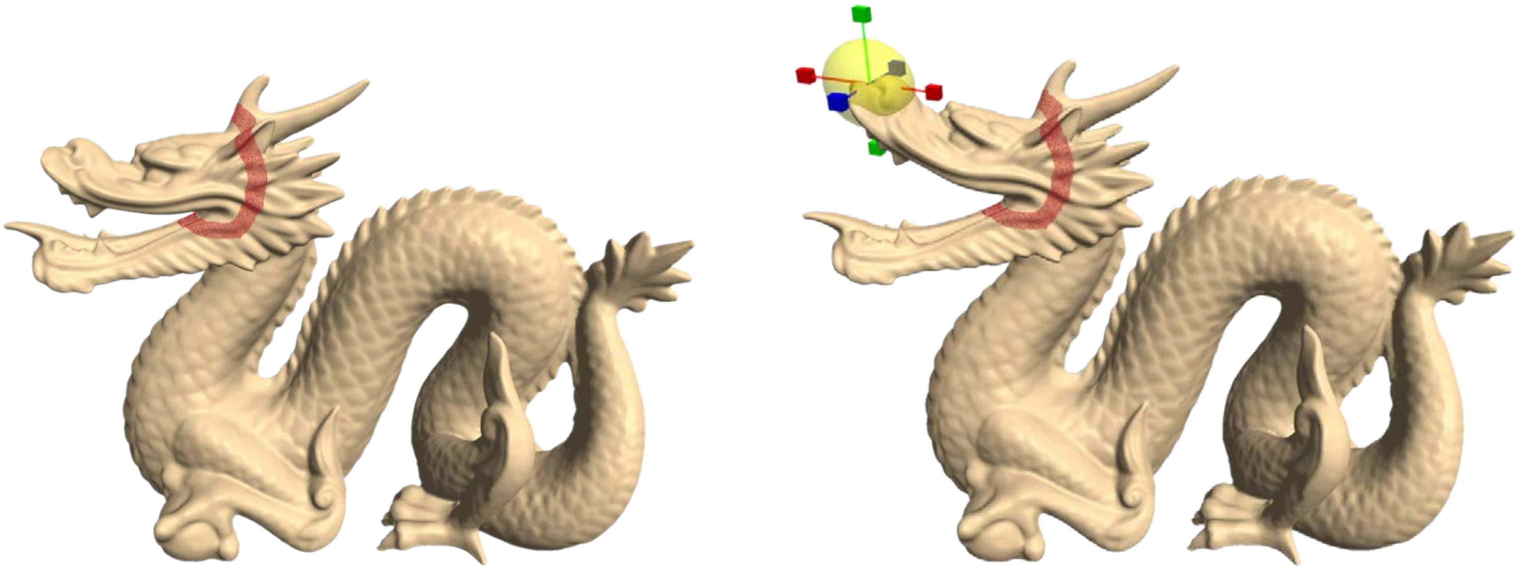


Mesh Editing Example

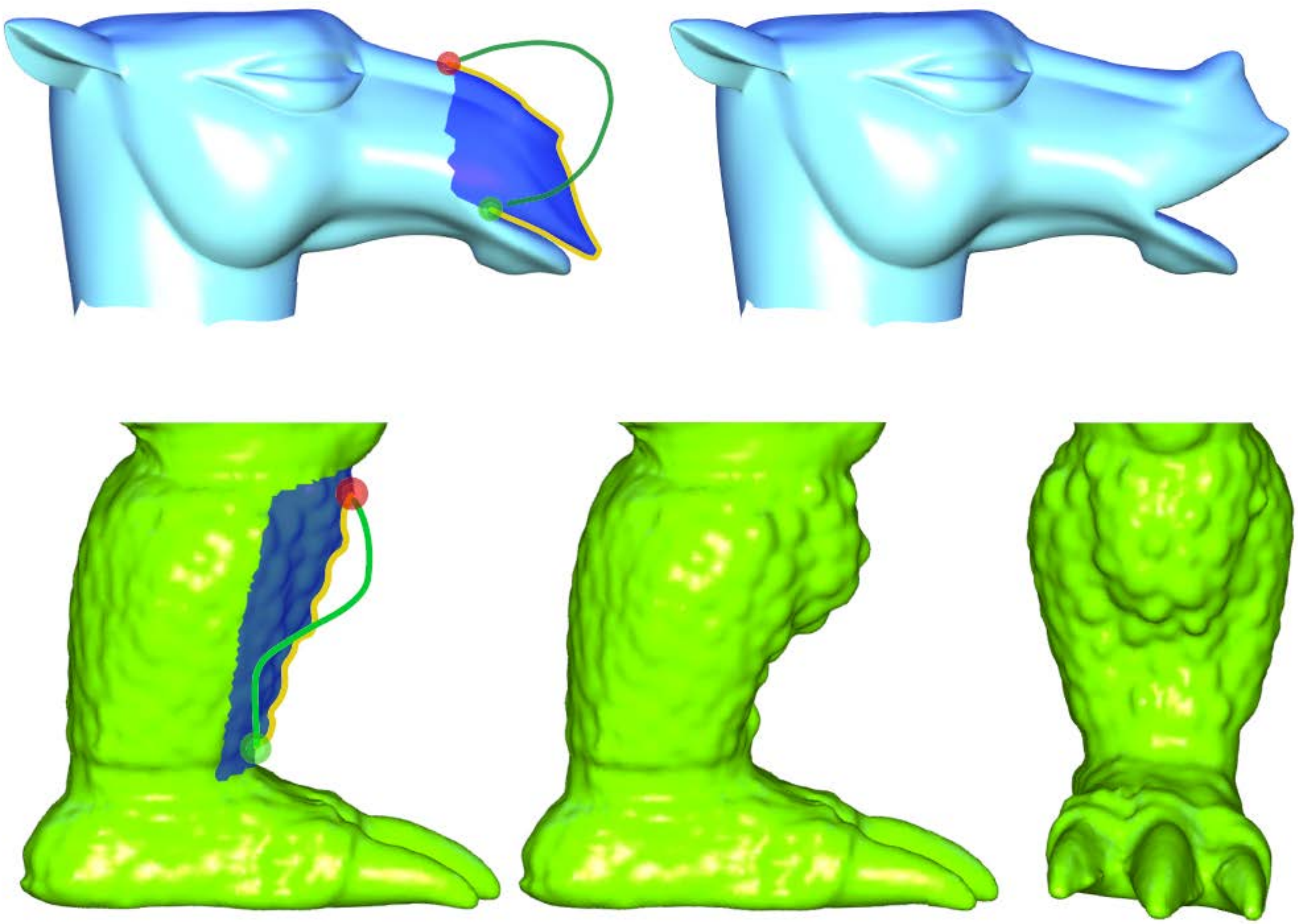


Original Regular Laplacian editing Solve for transformations

Mesh Editing Example



Mesh Editing Example

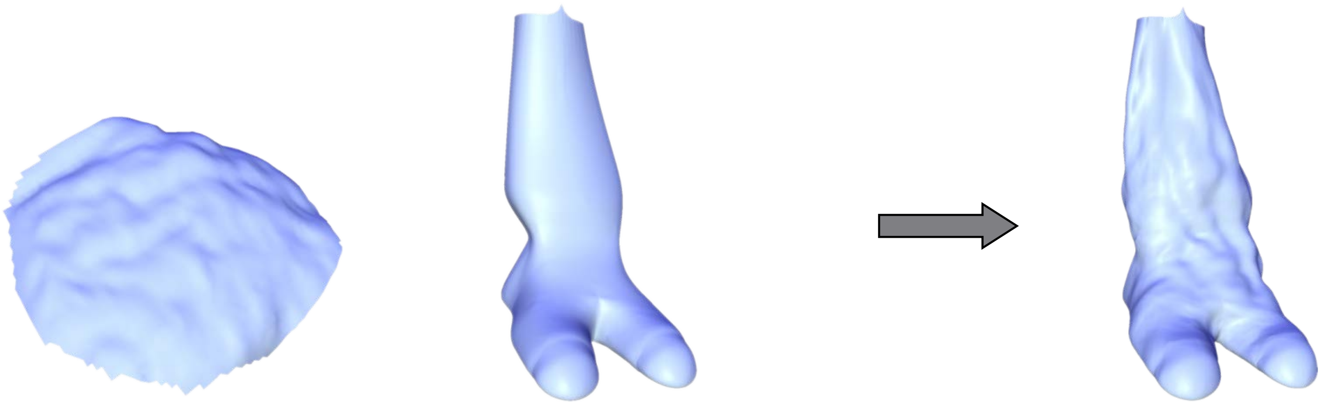


What Else Can We Do with It?

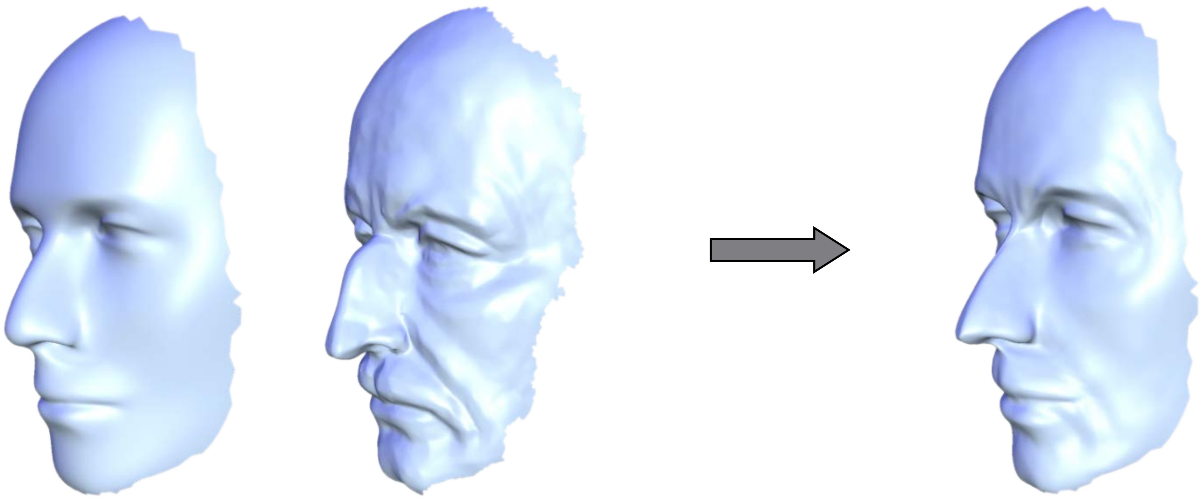
- By modifying Laplacians or positional constraints, can achieve a variety of other effects

Detail Transfer

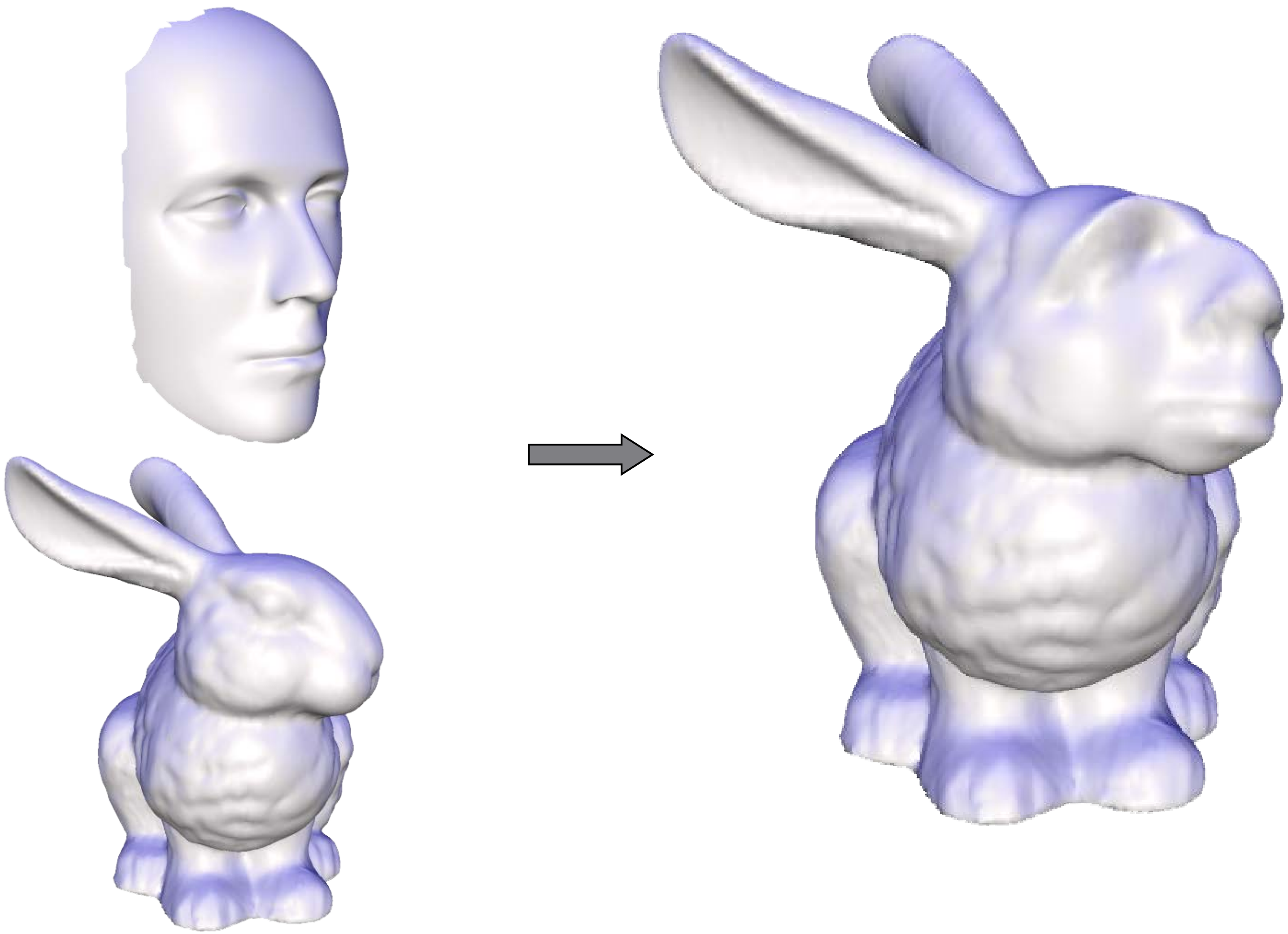
- “Peel“ the detail off one surface and transfer to another



Detail Transfer

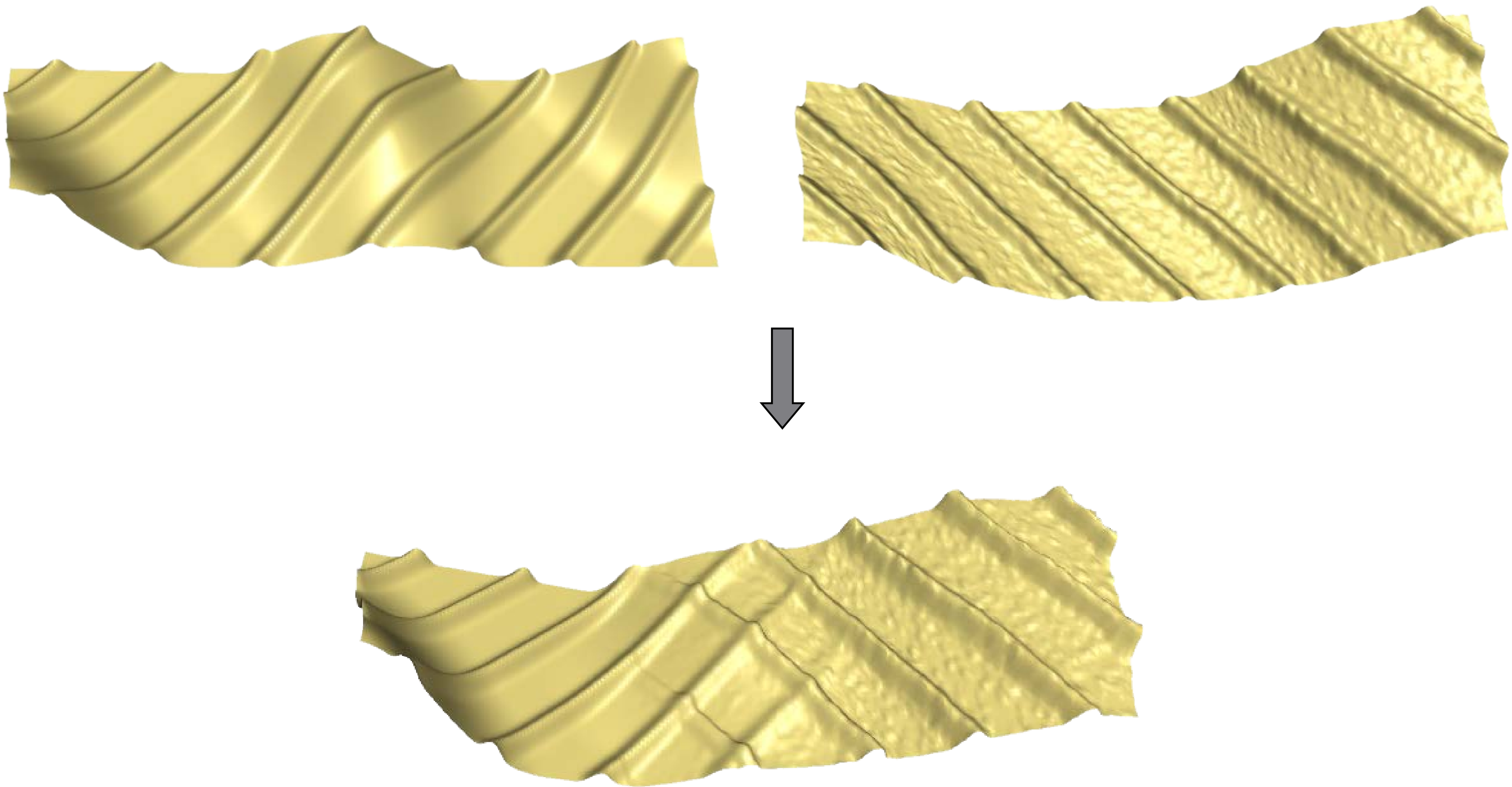


Detail Transfer



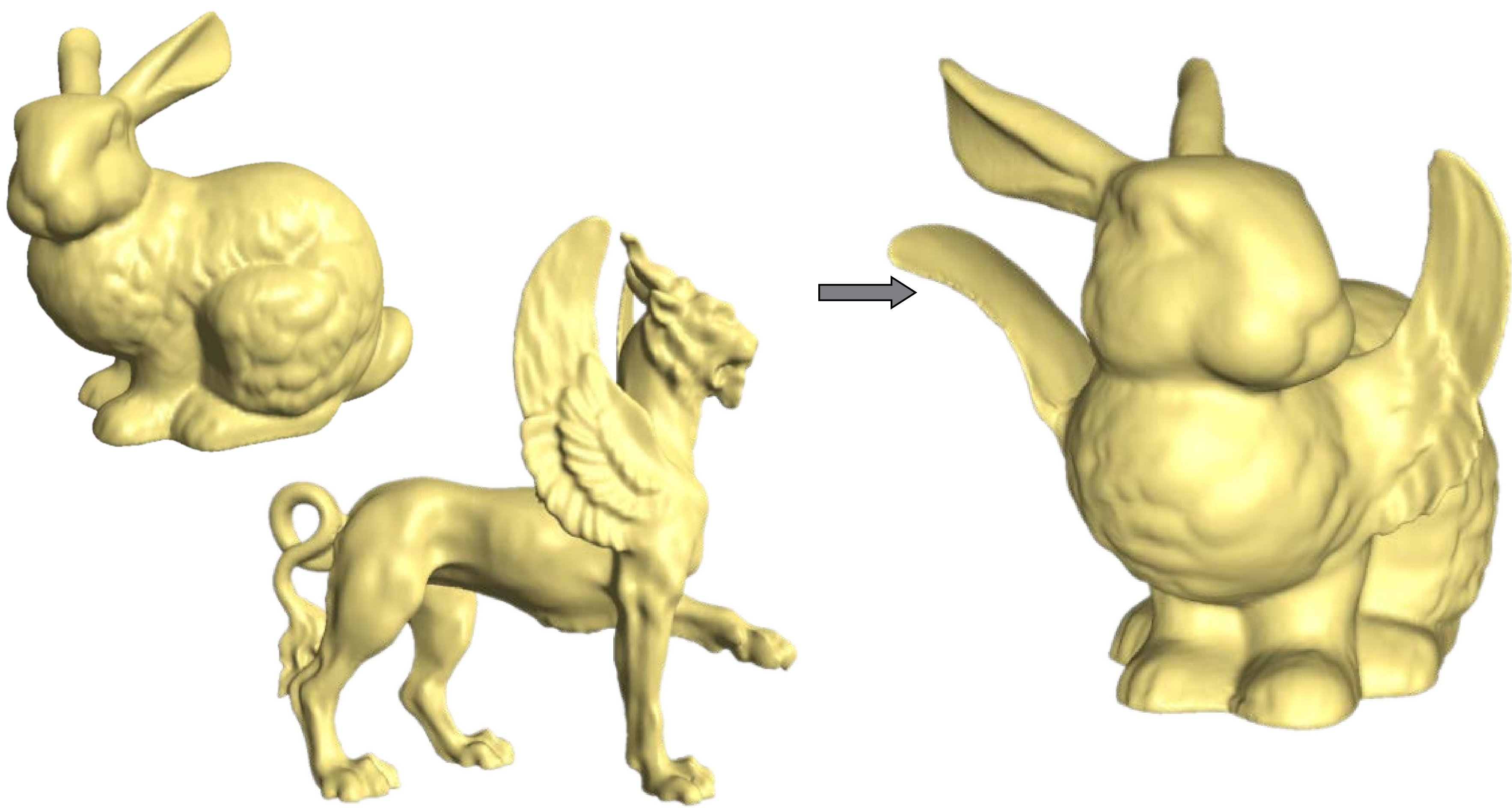
Mixing Laplacians

- Take weighted average of δ_i and δ'_i



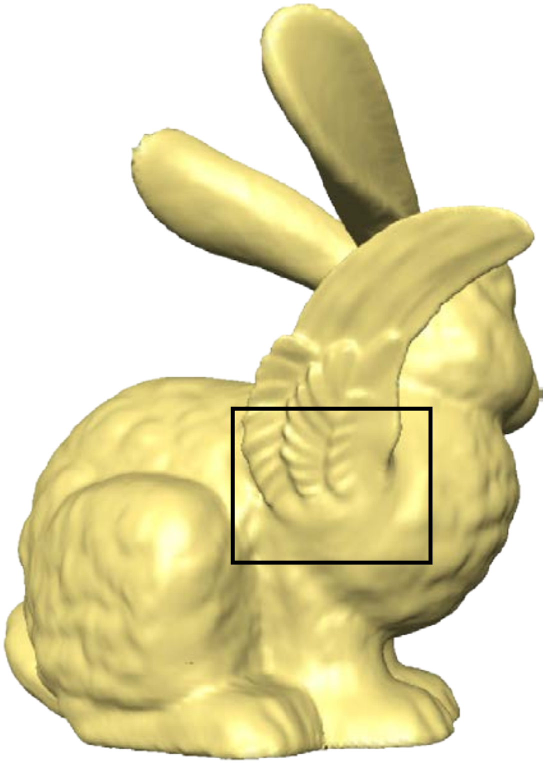
Mesh Transplanting

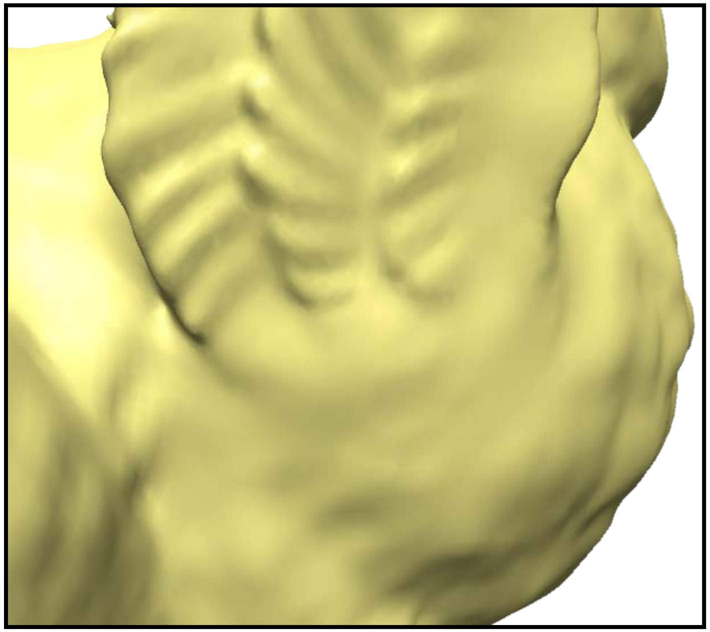
- Geometrical stitching via Laplacian mixing



Mesh Transplanting

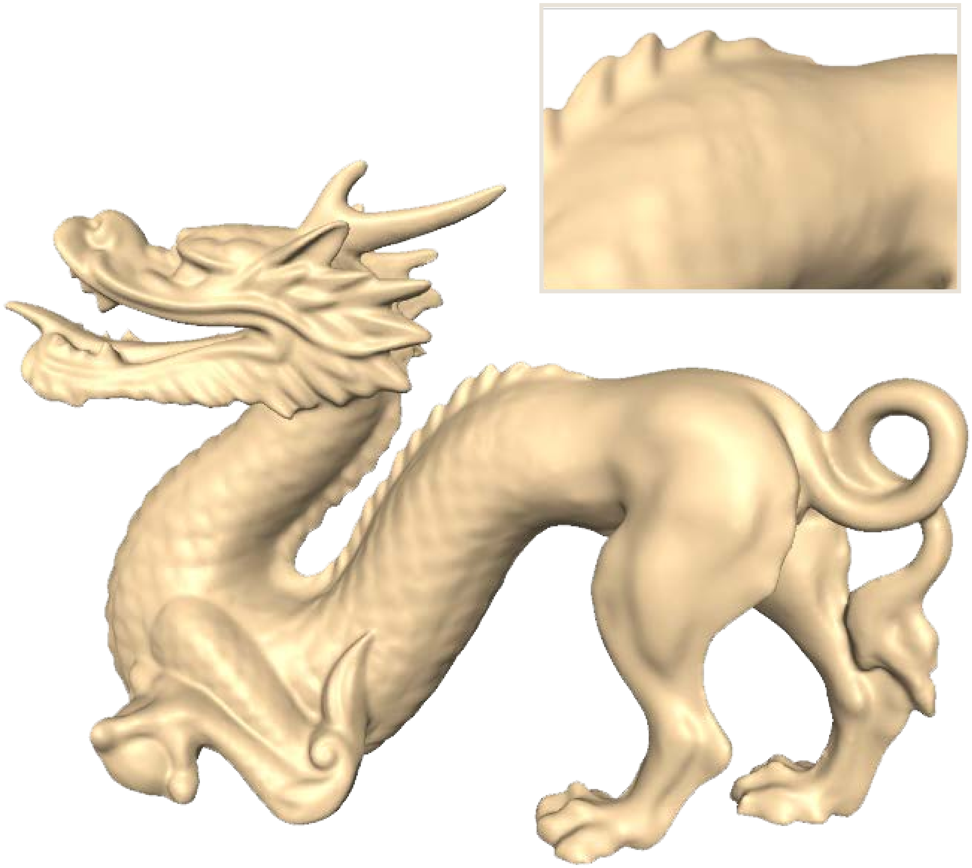
- Details gradually change in the transition area





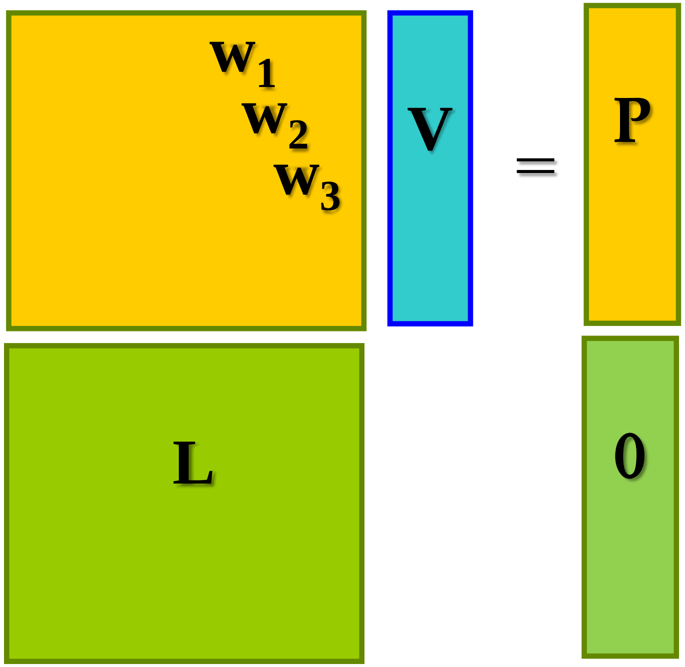
Mesh Transplanting

- Details gradually change in the transition area



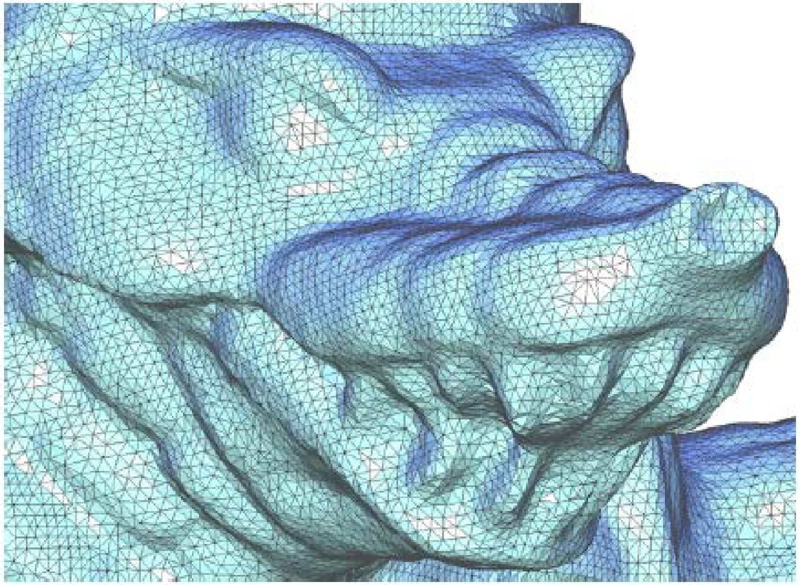
Feature Preserving Smoothing

- Weighted positional and smoothing constraints

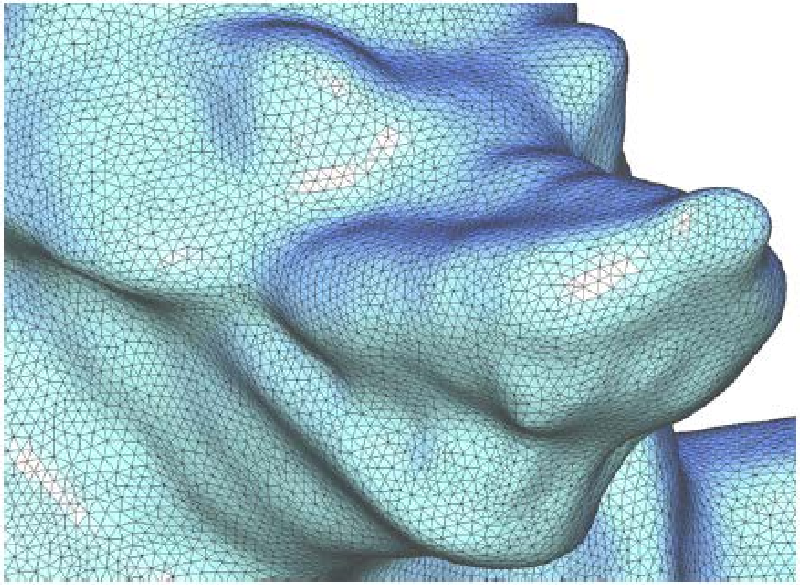


Feature Preserving Smoothing

- Weighted positional and smoothing constraints



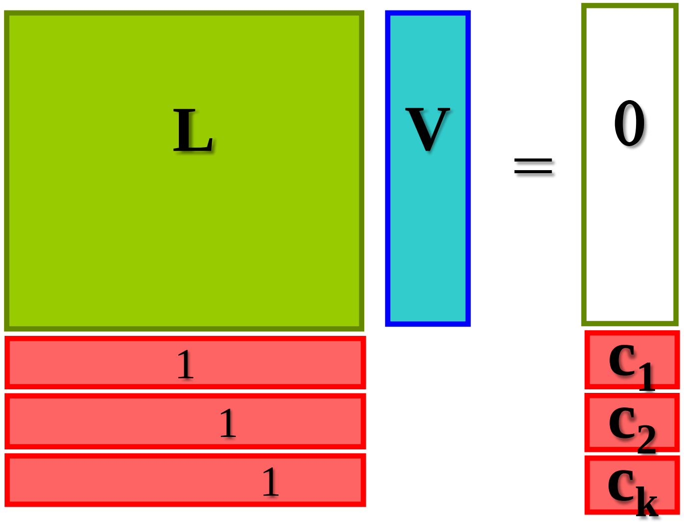
Original



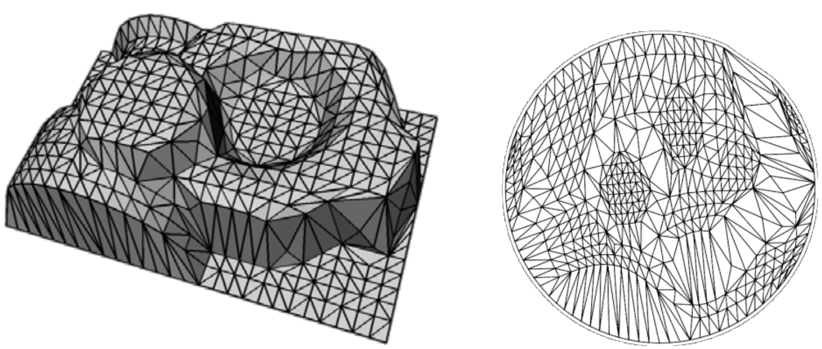
Smoothed

Parameterization

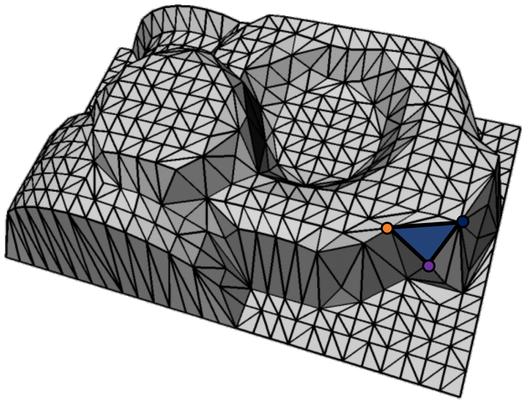
- Use zero Laplacians.

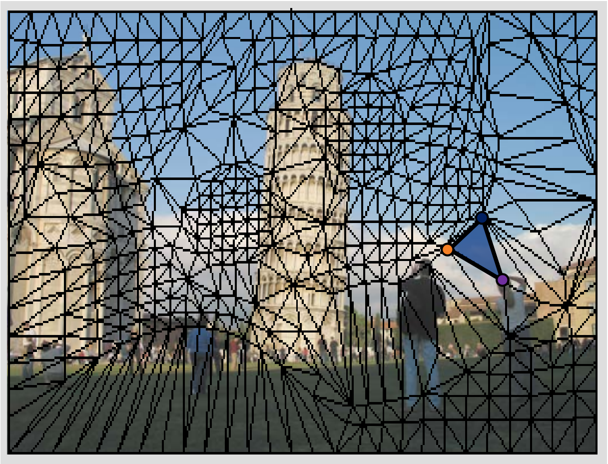


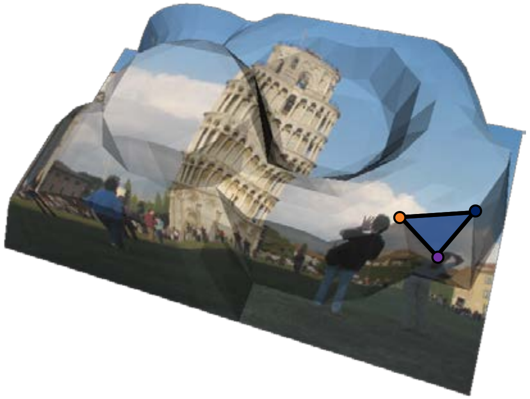
In 2D:

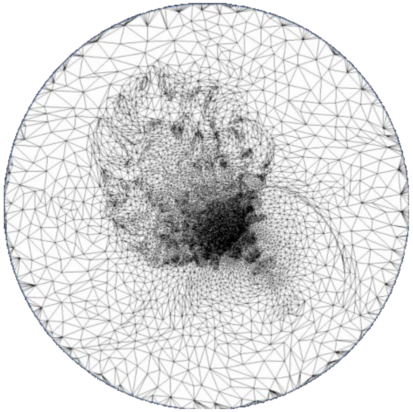


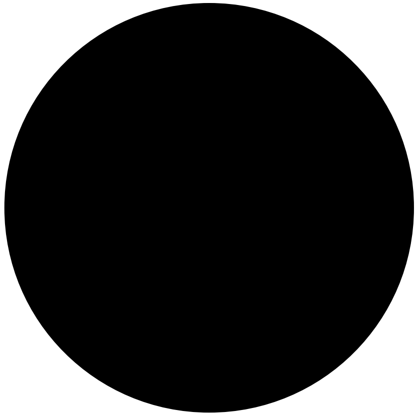
Texture Mapping



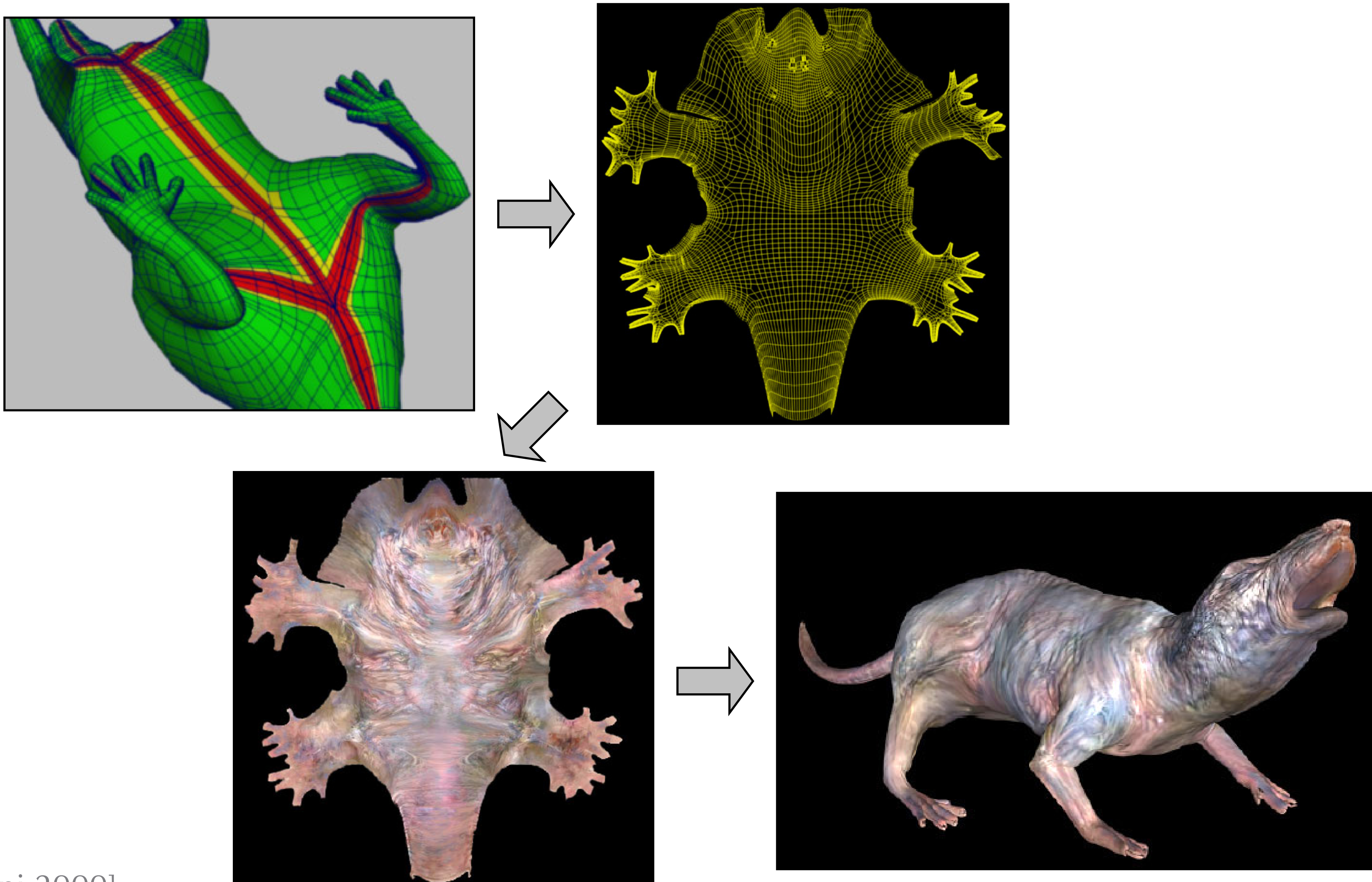








Texture Mapping



[Piponi 2000]