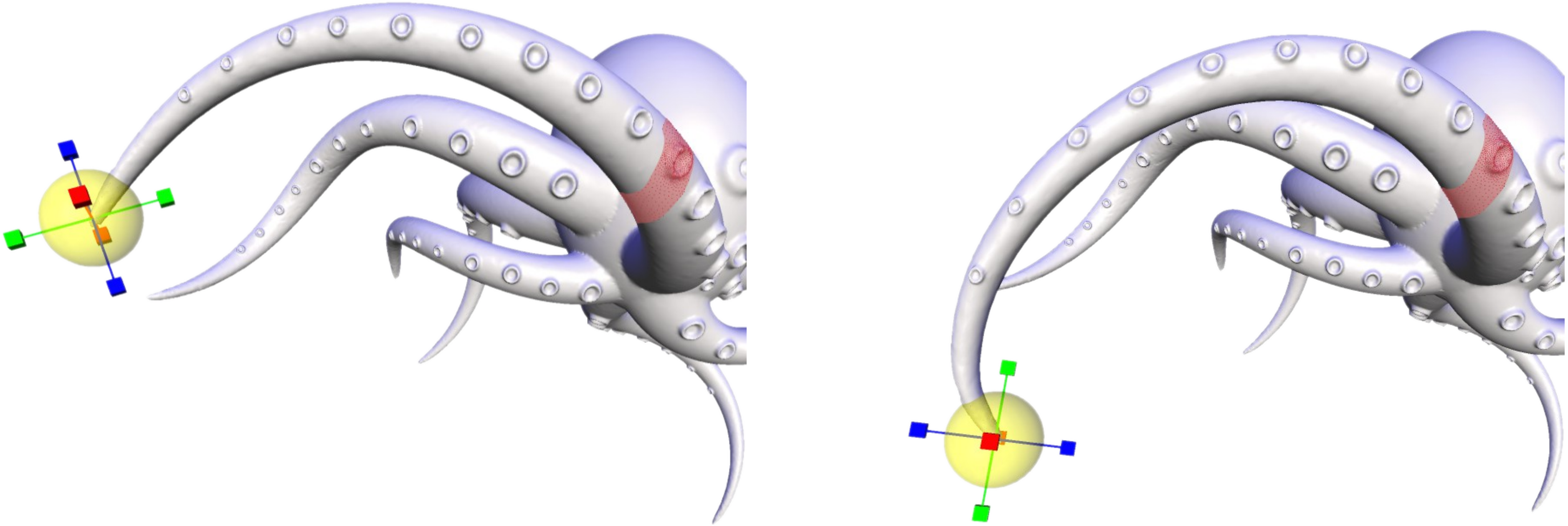
Sorkine et al.



Laplacian Mesh Processing

(includes material from Olga Sorkine, Yaron Lipman, Marc Pauly, Adrien Treuille, Marc Alexa and Daniel Cohen-Or)

Siddhartha Chaudhuri http://www.cse.iitb.ac.in/~cs749

Recap: Laplacian in Euclidean space

- **Laplacian** (of scalar-valued function): $\Delta f = \nabla \cdot \nabla f = \frac{\partial^2 f}{\partial x_1^2} + \frac{\partial^2 f}{\partial x_2^2} + \ldots + \frac{\partial^2 f}{\partial x_n^2}$
  - In operator form:

$$\Delta = \left(\frac{\partial^2}{\partial x_1^2}, \frac{\partial^2}{\partial x_2^2}, \ldots, \frac{\partial^2}{\partial x_n^2} \right)$$

  - Maps scalar field to scalar field

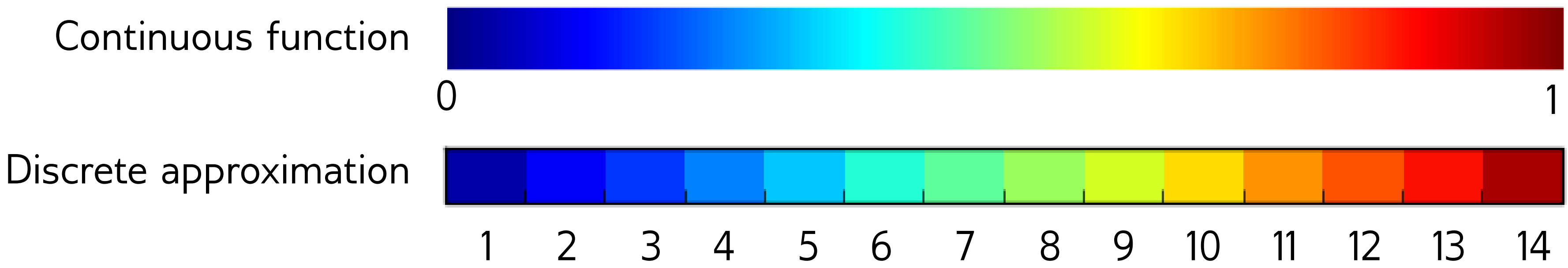
Divergence of gradient of f

Original function

After applying Laplacian

Recap: Laplacian in Euclidean space

- The Laplacian can be discretized



$$
\frac{1}{h}\begin{bmatrix}
-1 & 1 & 0 & \cdots & 0 & 0 \\
0 & -1 & 1 & \cdots & 0 & 0 \\
 & 0 & -1 & \cdots & 0 & 0 \\
\vdots & & & \ddots & & \vdots \\
0 & 0 & & & -1 & 1 \\
0 & 0 & \cdots & & 0 & -1
\end{bmatrix}
$$

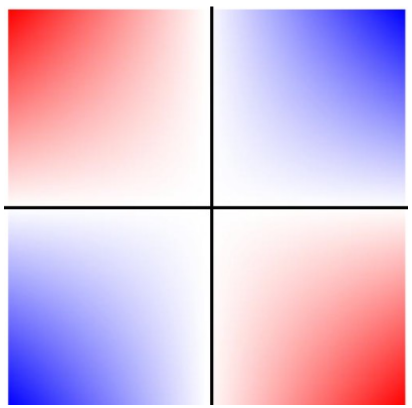
1D discrete gradient

$$
\frac{1}{h^2}\begin{bmatrix}
-2 & 1 & 0 & \cdots & 0 & 0 \\
1 & -2 & 1 & \cdots & 0 & 0 \\
 & 1 & -2 & \cdots & 0 & 0 \\
\vdots & & & \ddots & & \vdots \\
0 & 0 & & & -2 & 1 \\
0 & 0 & \cdots & & 1 & -2
\end{bmatrix}
$$

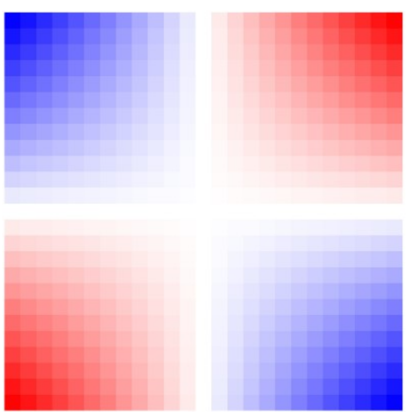
1D discrete Laplace operator

Laplacian in 2D Euclidean space

- The Laplacian can be discretized



Continuous function



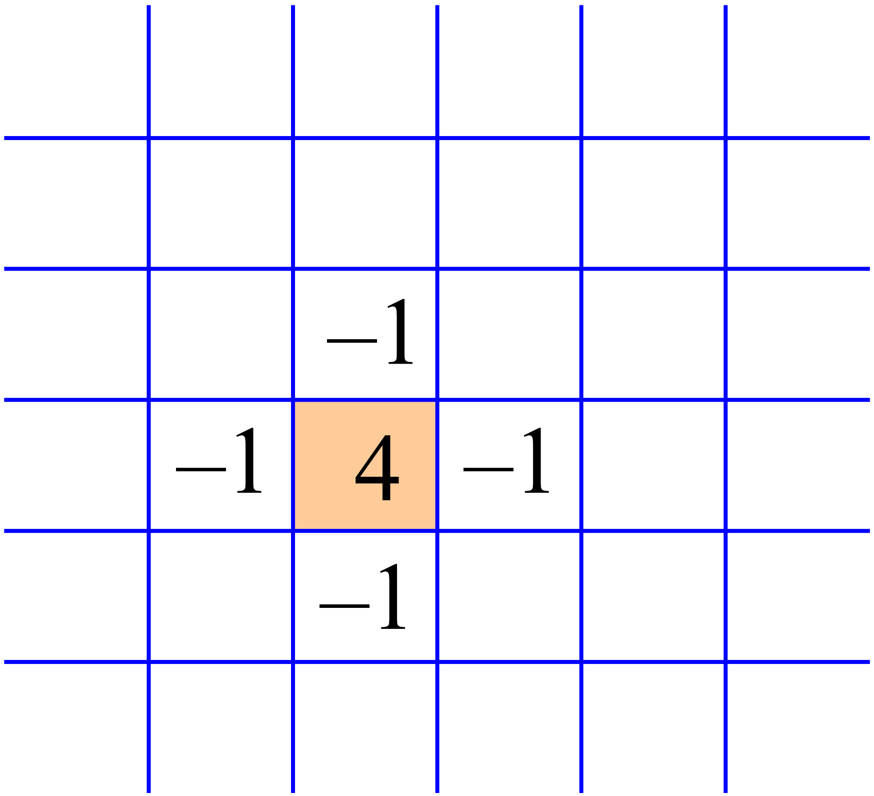
Discrete approximation

$$\begin{bmatrix} -4 & 1 & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & -4 & 1 & \cdot & 1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & 1 & -4 & \cdot & \cdot & 1 & \cdot & \cdot & \cdot \\ 1 & \cdot & \cdot & -4 & 1 & \cdot & 1 & \cdot & \cdot \\ \cdot & 1 & \cdot & 1 & -4 & 1 & \cdot & 1 & \cdot \\ \cdot & \cdot & 1 & \cdot & 1 & -4 & \cdot & \cdot & 1 \\ \cdot & \cdot & \cdot & 1 & \cdot & \cdot & -4 & 1 & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 & \cdot & 1 & -4 & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & 1 & -4 \end{bmatrix}$$

2D discrete Laplace operator

Laplacian in 2D Euclidean space

- The Laplacian is computed via differences of a cell from its neighbors
- We can think of the grid as a graph connecting adjacent cells



Thought for the Day #1

Can you express applying the Laplacian (or other differential operator) as a convolution?

Graph Laplacian

- For a general graph, we can compute a similar Laplace operator
  - The function f is represented by its values at graph vertices
  - The discrete Laplace operator is applied on graph neighborhoods centred at the vertices
  - If the graph is a grid, we should recover the standard Euclidean Laplacian

Graph Laplacian

- Let $G = (V, E)$ be an undirected graph
- Let $N_1(i)$ represent the 1-ring neighborhood of vertex i, that is, $N_1(i) = \{ j \mid (i, j) \in E \}$
- The degree d_i of vertex i is $|N_1(i)|$
- Let D be the diagonal matrix with $D_{ii} = d_i$
- The adjacency matrix A of G is $A_{ij} = \begin{cases} 1 & (i, j) \in E \\ 0 & \text{otherwise} \end{cases}$

Graph Laplacian

- The (topological) **Laplacian matrix** L of the graph is defined as

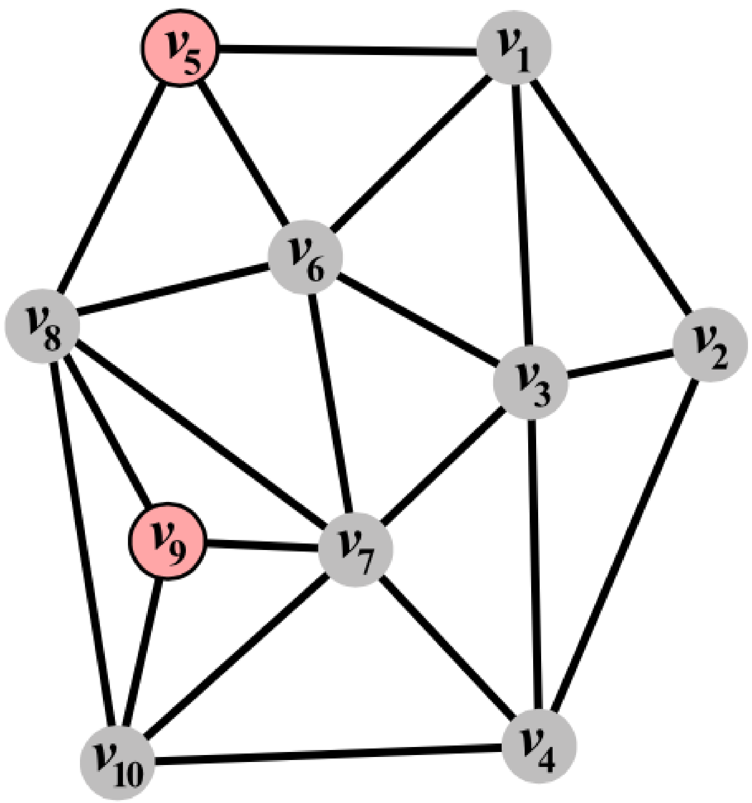
$$L = I - D^{-1}A$$

- Commonly, we can multiply by D and consider the **symmetric Laplacian** $L_s = DL = D - A$ instead

$$(L_s)_{ij} = \begin{cases} d_i & i = j \\ -1 & (i,j) \in E \\ 0 & \text{otherwise} \end{cases}$$

Verify that this gives the correct grid Laplacian

Graph Laplacian



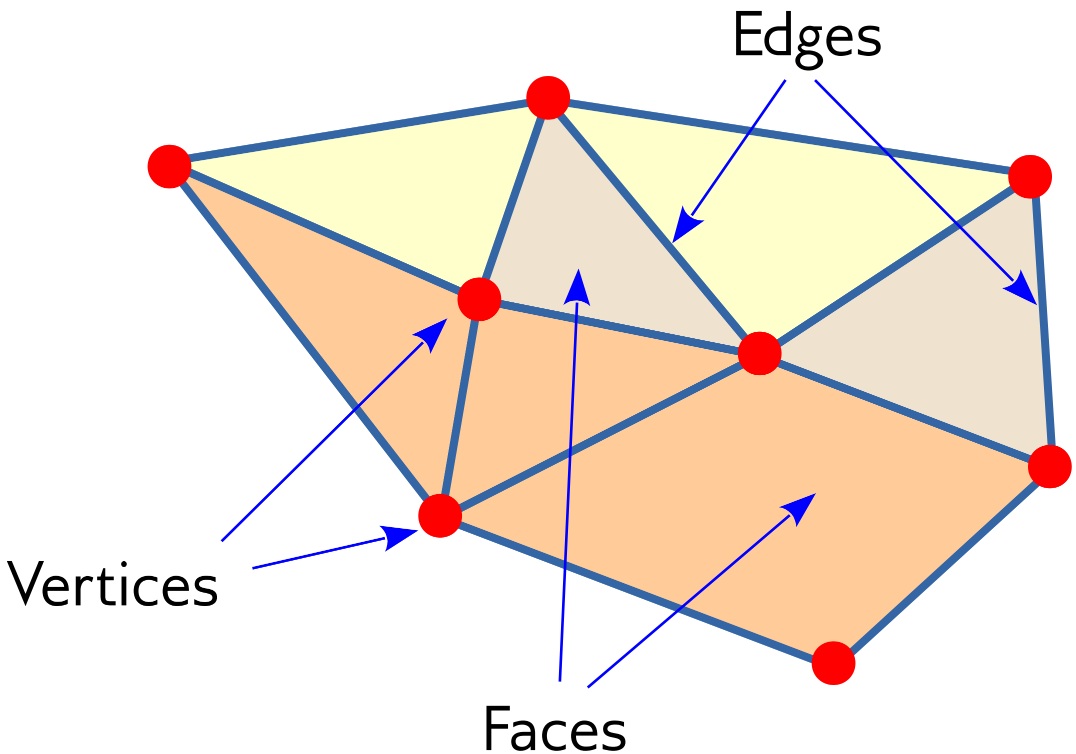
Graph

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

Symmetric Laplacian L_s

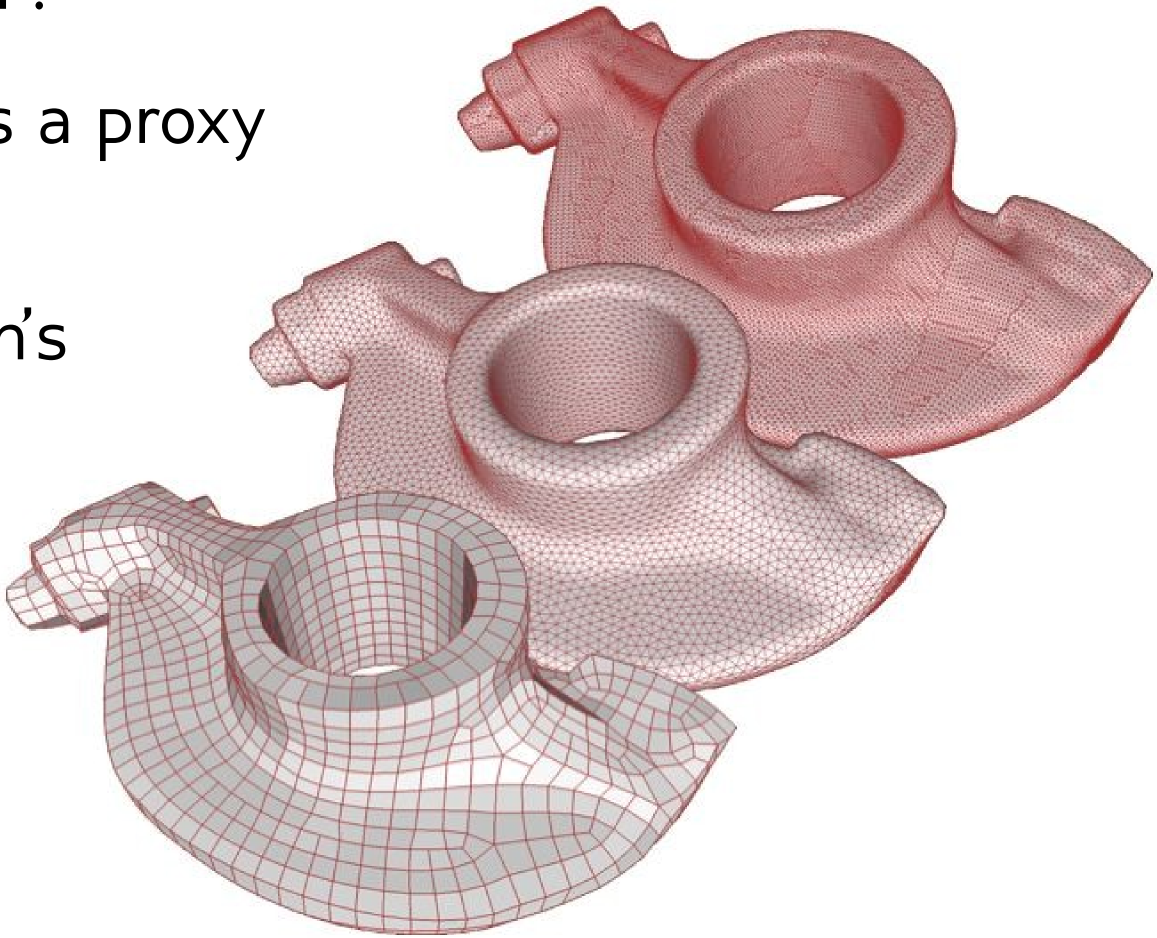
Mesh (Topological) Laplacian

- Recall: a mesh is a graph
- … so we can compute its topological Laplacian



Refining the mesh

- What happens to the Laplacian as we make the mesh finer and finer?
  - The graph serves as a proxy for the surface
  - Eventually, the graph's metric properties should reduce to geometric surface properties



Leif Kobbelt Group

Thought for the Day #2

The *topological* Laplacian does not care about the embedding metric, i.e. the lengths of edges in the graph. How, then, can the Laplacian be sensitive to the *geometry* of the surface?

Once again: the Euclidean Laplacian

Laplace operator

gradient operator

2nd partial derivatives

$$\Delta f = \mathrm{div} \nabla f = \sum_i \frac{\partial^2 f}{\partial x_i^2}$$

function in Euclidean space

divergence operator

Cartesian coordinates

The Laplace-Beltrami operator

- Extension of continuous Laplacian to manifolds ("smooth curved surfaces/spaces")

Laplace-Beltrami

gradient operator

$$\Delta_{\mathcal{S}} f = \mathrm{div}_{\mathcal{S}}\, \nabla_{\mathcal{S}} f$$

function on manifold $\mathcal{S}$

divergence operator

The Laplace-Beltrami operator

- Example: Let $\mathbf{x}$ be the coordinate function: $\mathbf{x}(p)$ = position of p

Laplace-Beltrami

gradient operator

mean curvature

$$\Delta_{\mathcal{S}}\mathbf{x} = \mathrm{div}_{\mathcal{S}}\, \nabla_{\mathcal{S}}\mathbf{x} = -2H\mathbf{n}$$

coordinate function

divergence operator

surface normal

Here, applying the Laplacian to a vector-valued function means applying it to each coordinate individually

The Laplace-Beltrami operator

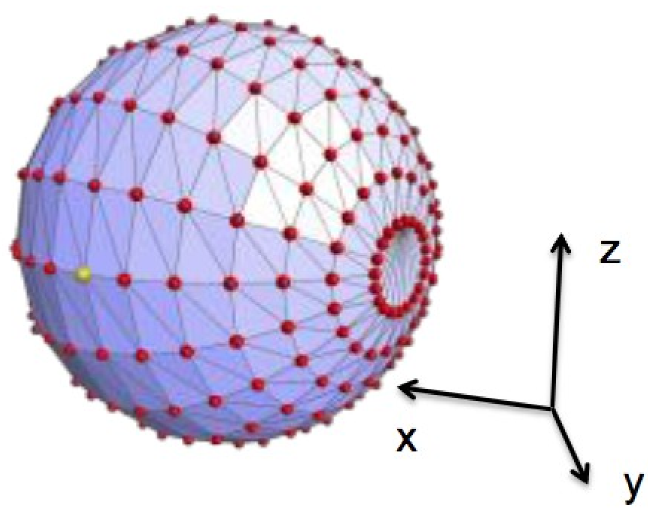
- Just like in 2D Euclidean space, if we know the Laplace-Beltrami operator of the surface, and the function value at a single point, we can solve for the function at all points on the surface
- Why is this important?
  - The LB operator captures global properties of the surface in terms of local properties
  - It expresses a local invariant
    - E.g. for the coordinate function, it expresses the surface in terms of its local curvature and orientation
  - We can modify the properties locally in one small region and recompute the global properties, trying to preserve the local property everywhere else

The Laplace-Beltrami operator

- The topological Laplacian is an approximation to the discrete Laplace-Beltrami operator
- We shall see better approximations soon

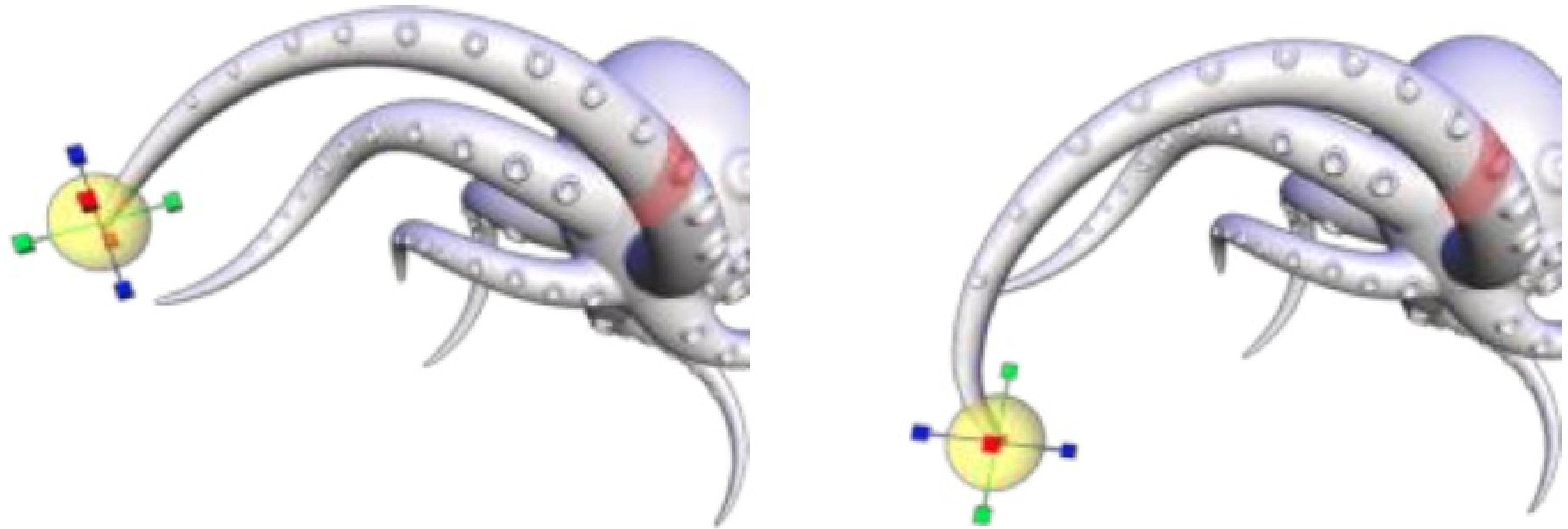
Laplacian Mesh Editing: Motivation

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



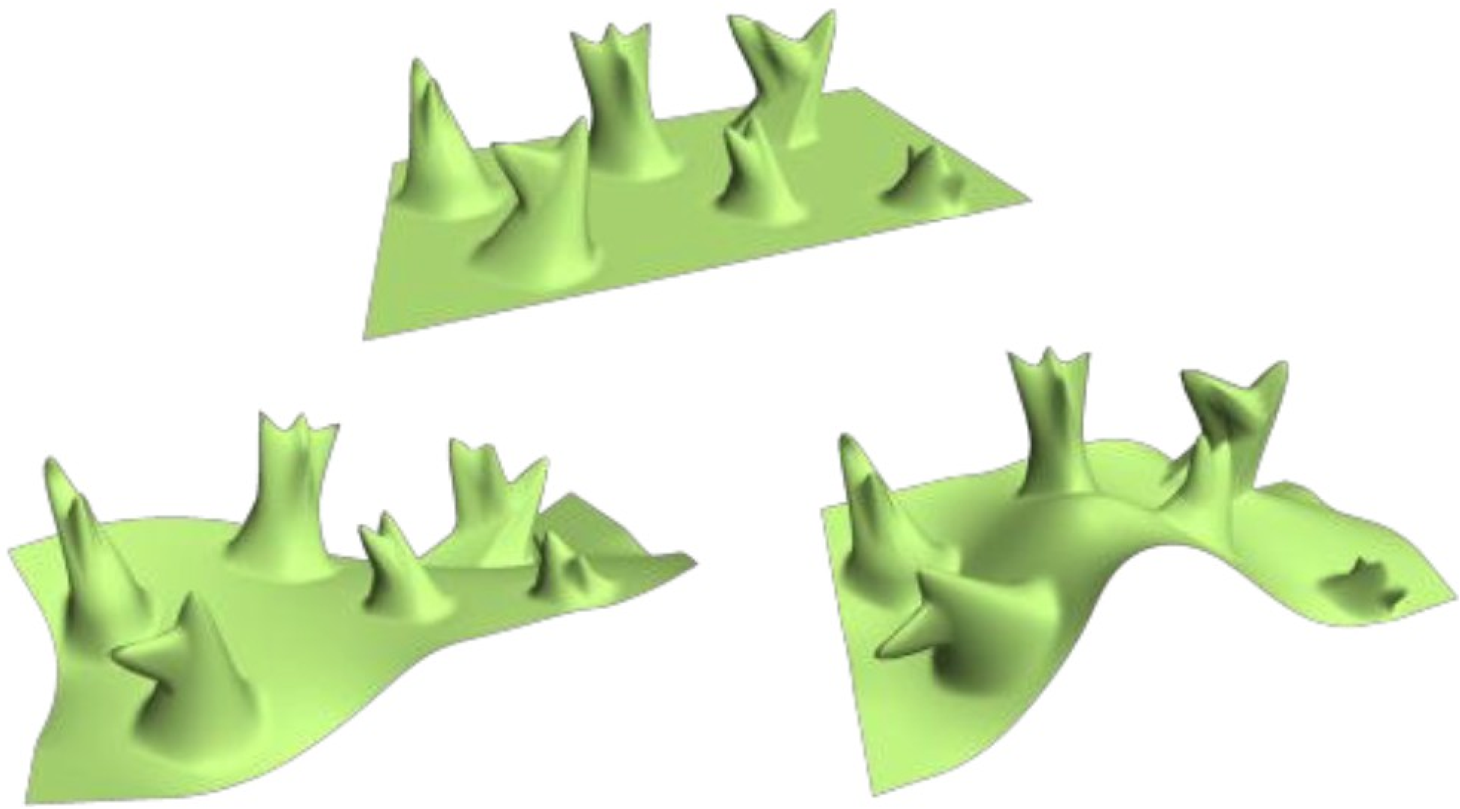
Laplacian Mesh Editing: Motivation

- Meshes are difficult to edit



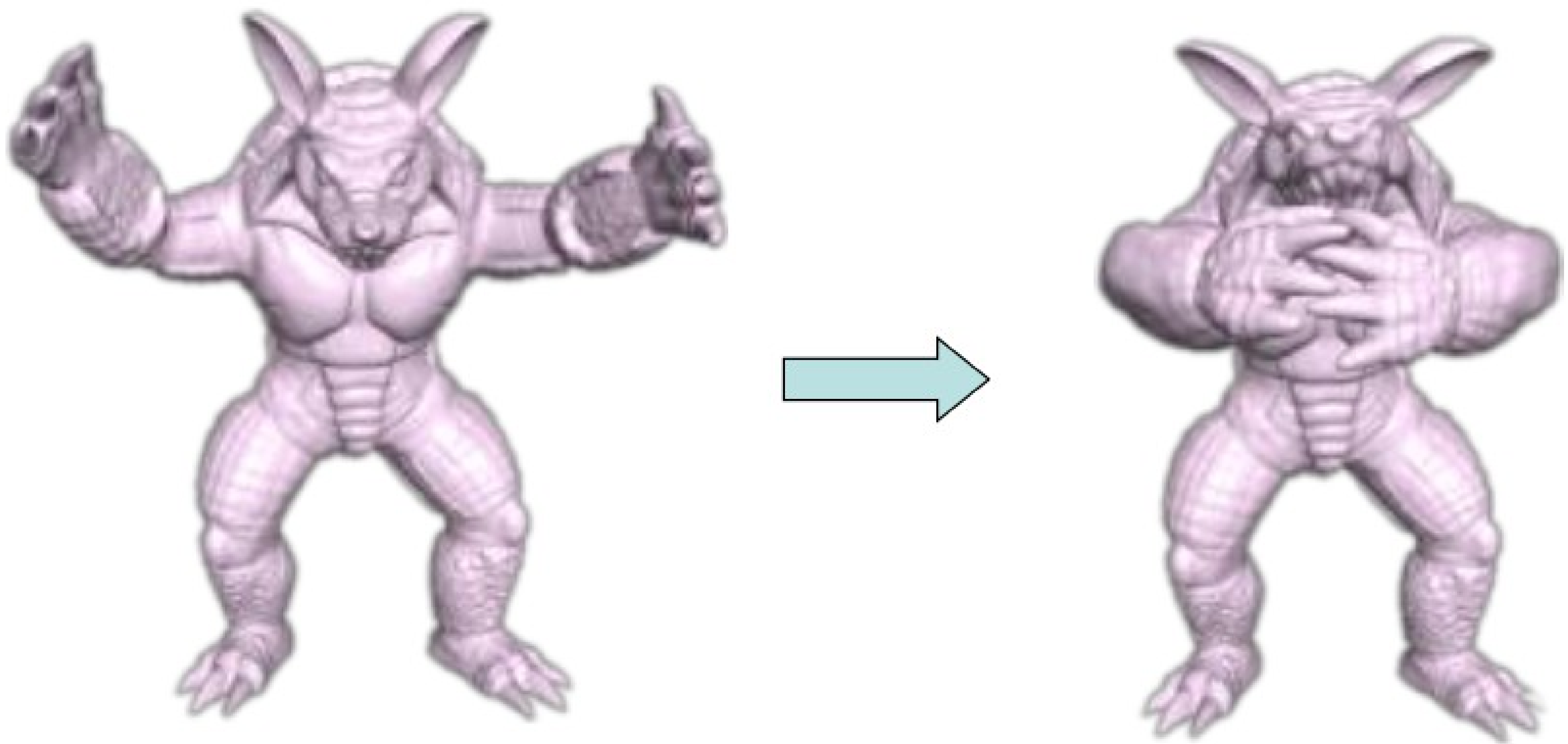
Laplacian Mesh Editing: Motivation

- Meshes are difficult to edit



Laplacian Mesh Editing: Motivation

- Meshes are difficult to edit

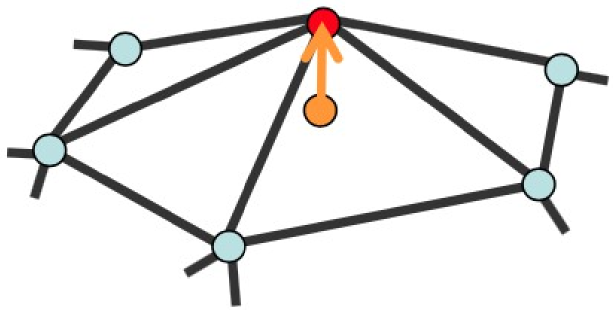


Differential Coordinates

- Represent a point *relative* to its neighbors
- Represent *local detail* at each surface point
- Linear operator takes us from global to differential
- Useful for operations on surfaces where surface details are important

Differential Coordinates

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

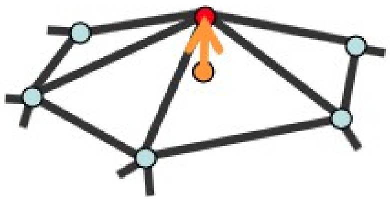


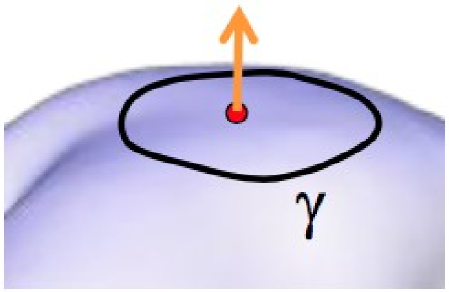
$$\boldsymbol{\delta}_i = \mathbf{v}_i - \frac{1}{d_i} \sum_{j \in N(i)} \mathbf{v}_j$$

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

Connection to the smooth case

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



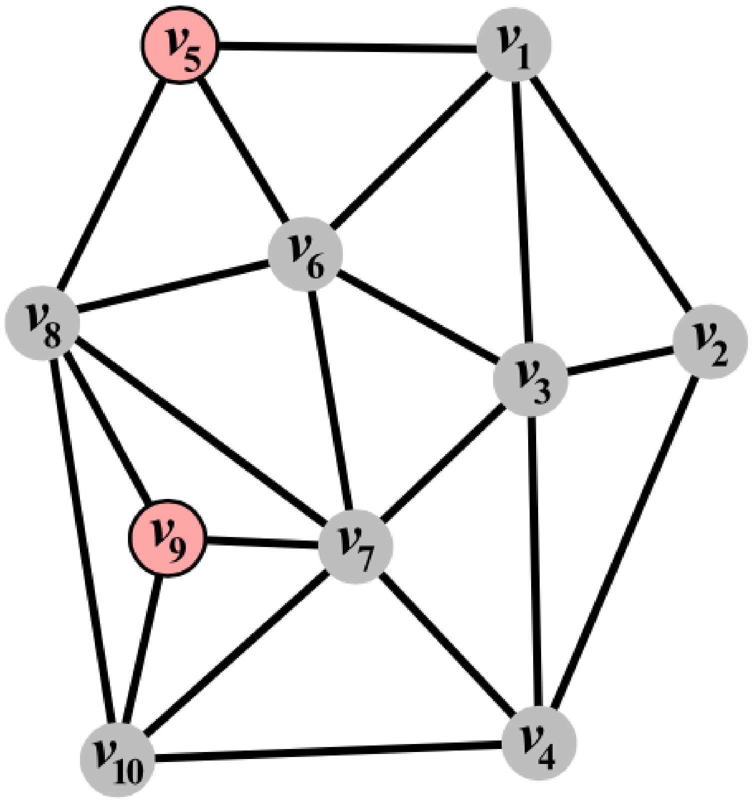


$$\boldsymbol{\delta}_{\mathbf{i}} = \frac{1}{d_i} \sum_{\mathbf{v} \in N(i)} (\mathbf{v}_{\mathbf{i}} - \mathbf{v})$$

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

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

Connection to graph Laplacian



Graph

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

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

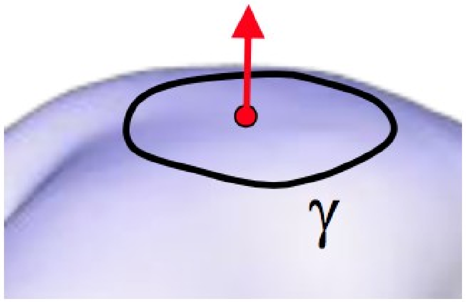
  $\delta_{\mathbf{umbrella}}$: $w_{ij} = 1$

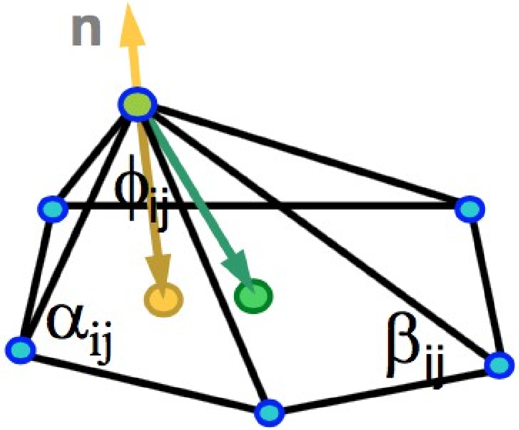
- Integrate over circle around vertex

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

- Integrate over Voronoi region of vertex

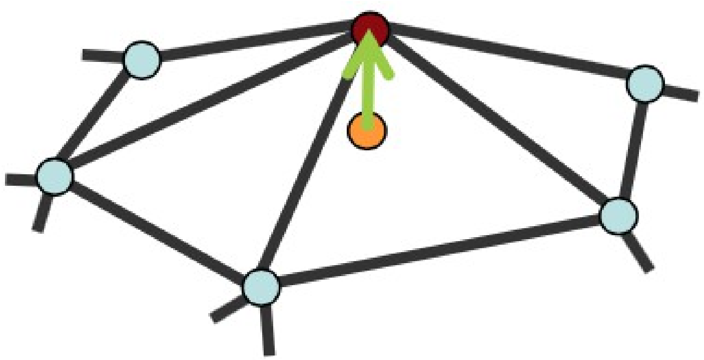
  $\delta_{\mathbf{cotangent}}$: $w_{ij} = \cot \alpha_{ij} + \cot \beta_{ij}$





Laplacian Mesh

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



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

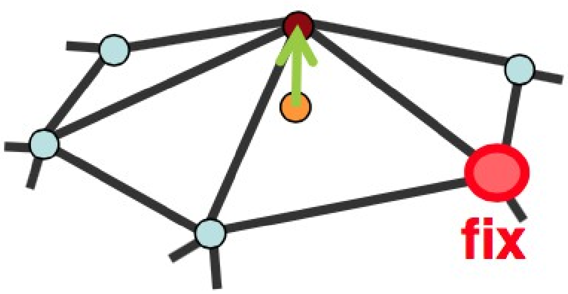
$$\mathbf{L} \; \mathbf{v_x} = \boldsymbol{\delta}_\mathbf{x}$$

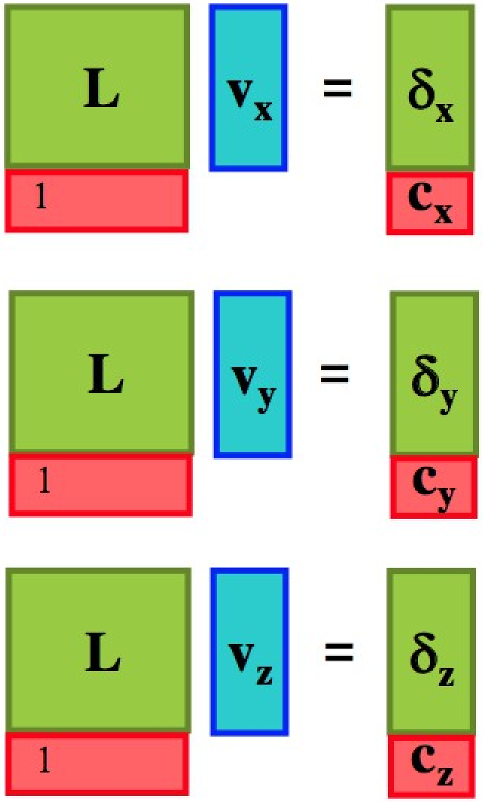
$$\mathbf{L} \; \mathbf{v_y} = \boldsymbol{\delta}_\mathbf{y}$$

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

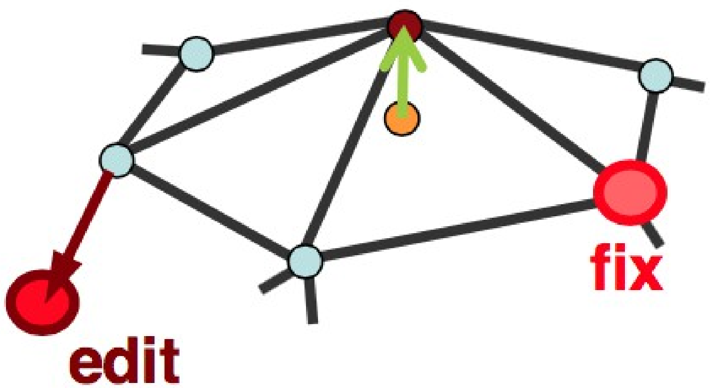
Basic properties

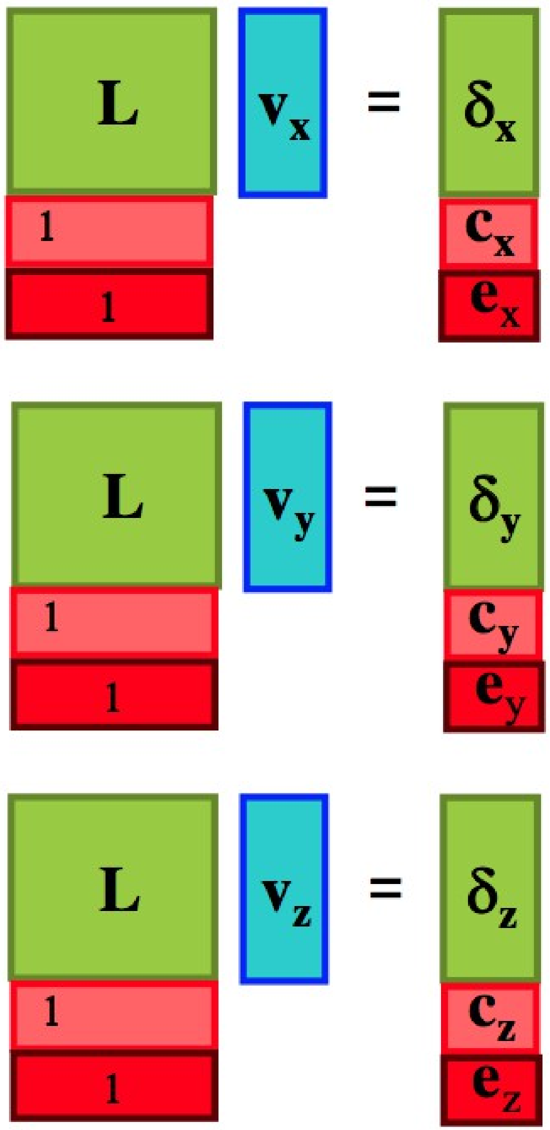
- rank(L) = n – c (n – 1 for connected meshes)
- We can reconstruct the xyz geometry from δ upto translation



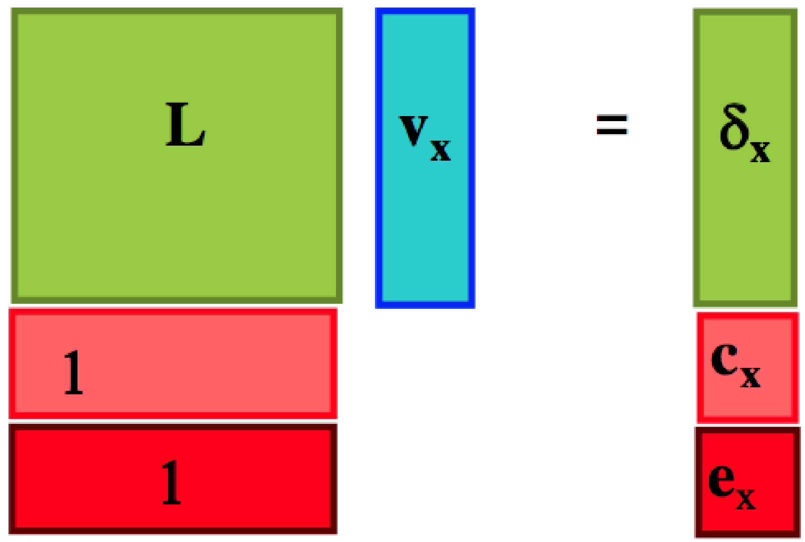


Reconstruction



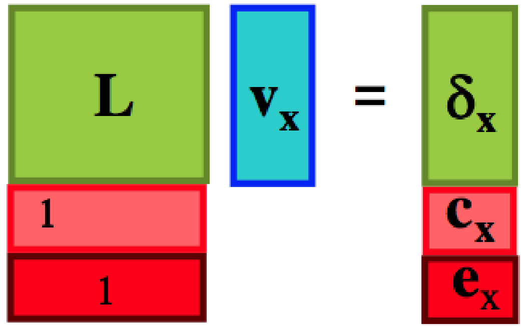


Reconstruction



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

Reconstruction



$$A \quad x \quad = \quad b$$

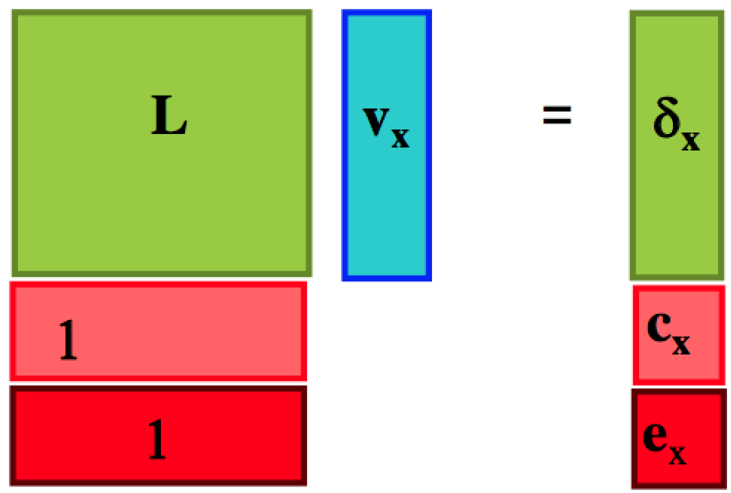
Normal Equations:

$$A^T A \quad x \quad = \quad A^T b$$

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

Cool underlying idea

- Mesh vertices are defined by the minimizer of an objective function



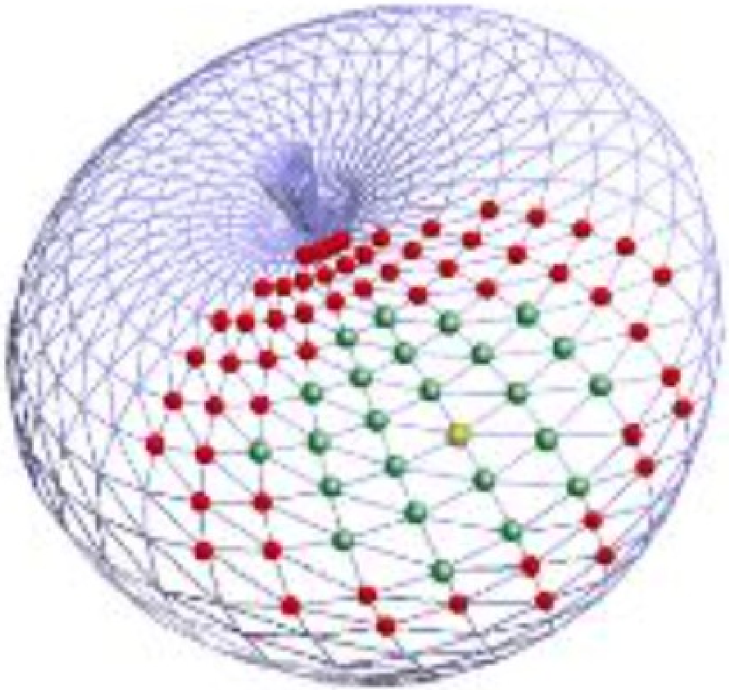
$$\boxed{\tilde{\mathbf{x}} = \arg\min_{\mathbf{x}} \left(\left\| L\mathbf{x} - \boldsymbol{\delta}_x \right\|^2 + \sum_{s=1}^{k} \left| x_k - c_k \right|^2 \right)}$$

What we have so far

- Laplacian coordinates $\boldsymbol{\delta} = L\mathbf{x}$
  - Local representation
  - Translation invariant
- Linear transition from $\boldsymbol{\delta}$ to xyz
  - Can constrain one or more vertices
  - Least squares solution

Editing using differential coordinates

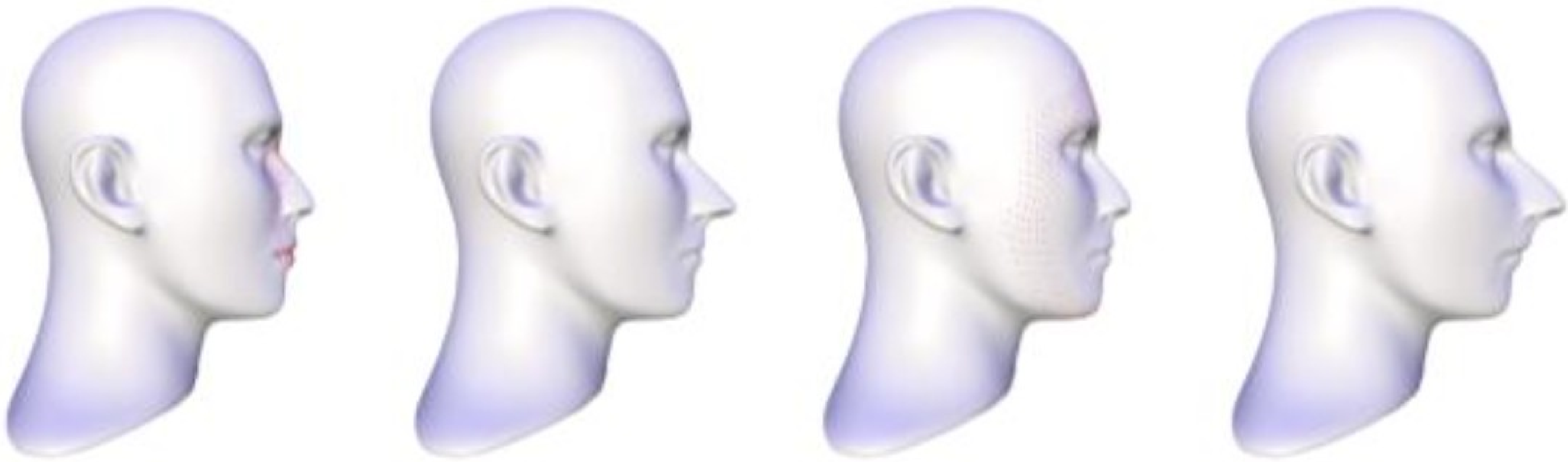
- The editing process from the user's point of view
  - First, set ROI, anchors and handle vertex
  - Move handle vertex to perform edit



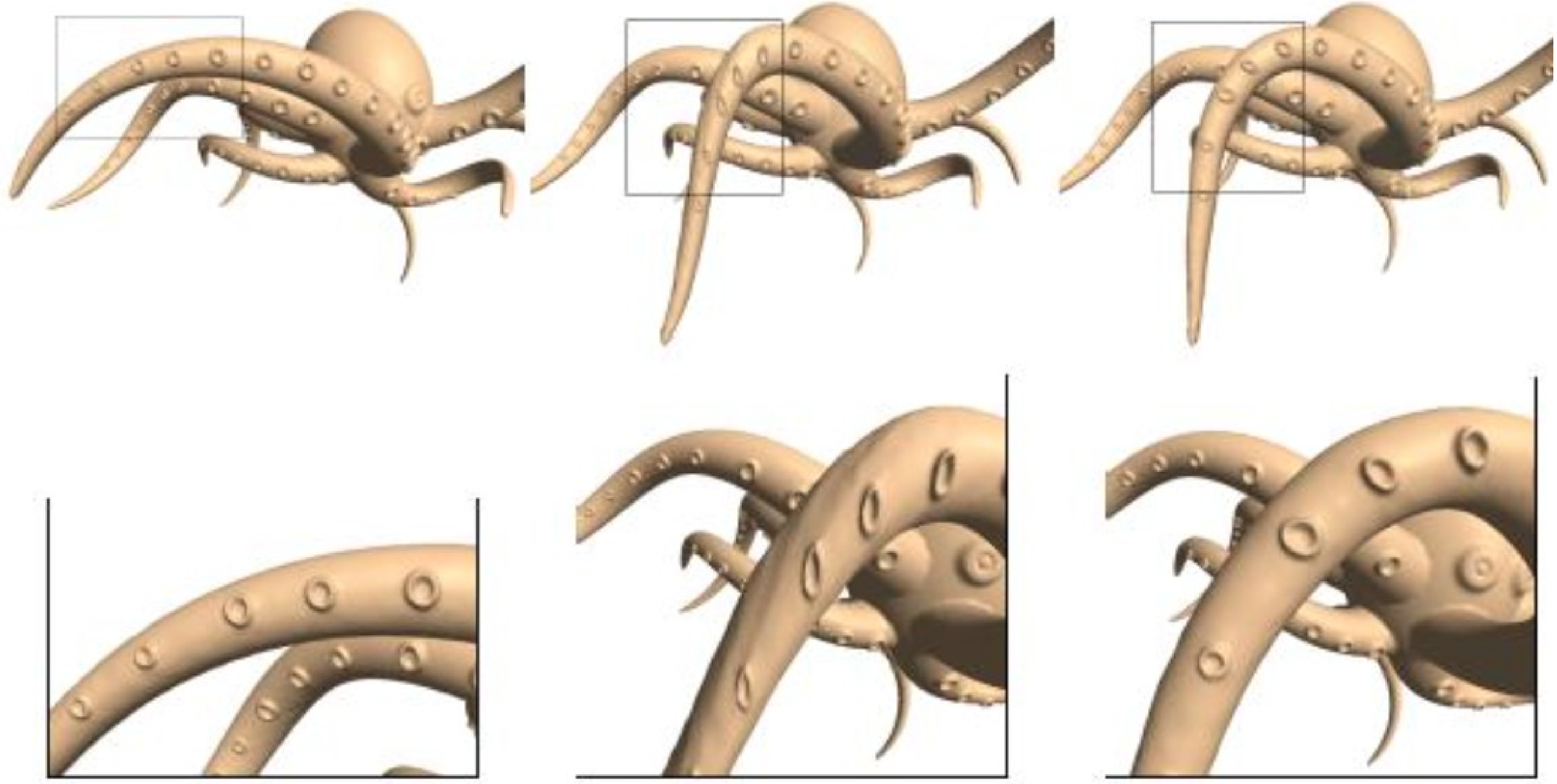
Editing using differential coordinates

- 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

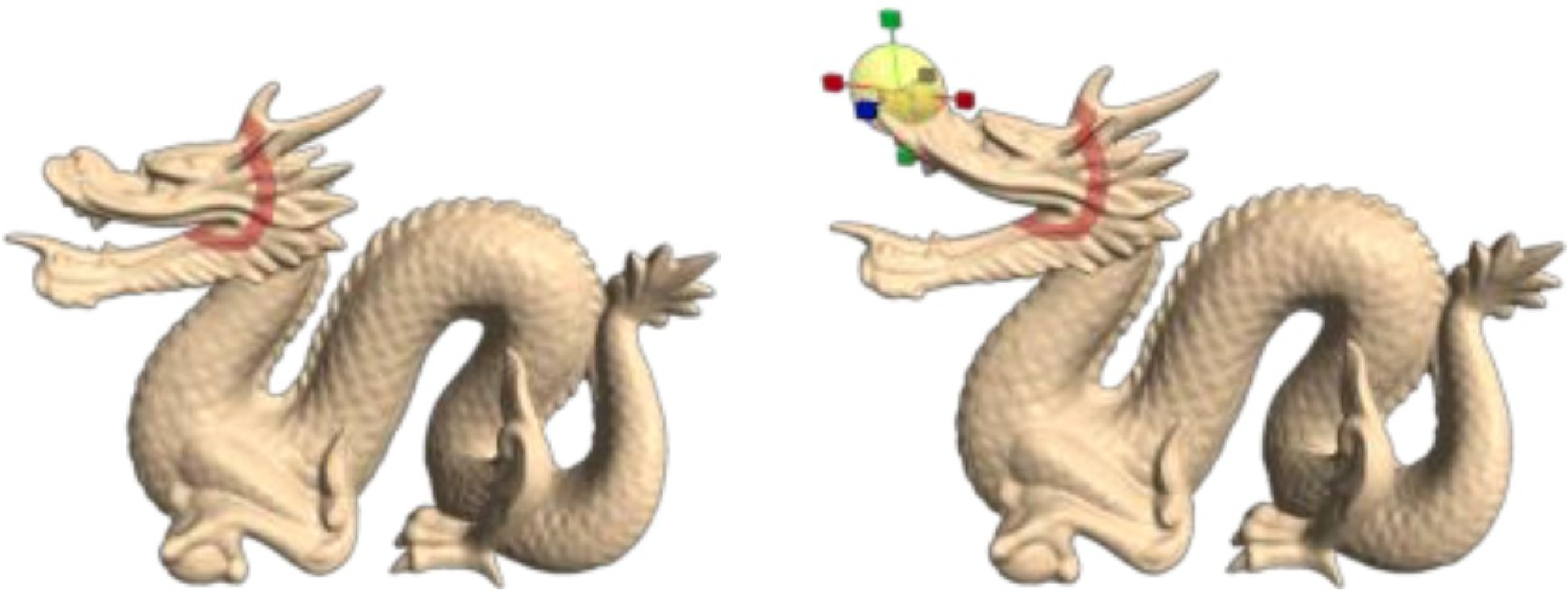
Mesh editing example



Mesh editing example



Mesh editing example



Mesh editing example