Lecture 22: Laplacian Mesh Editing

COMPSCI/MATH 290-04

Chris Tralie, Duke University

4/5/2016

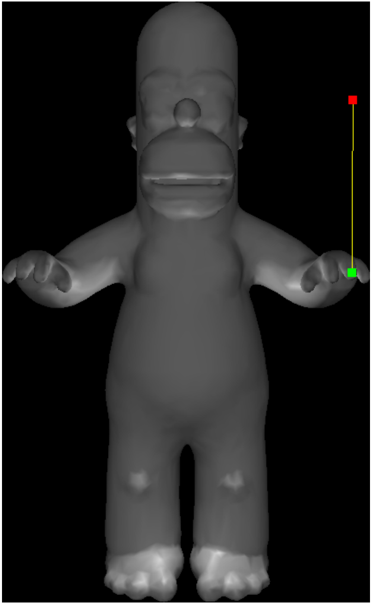
Announcements

- First project milestone Monday 4/11/2016
- First milestone 20%
- Group Assignment 3 Out This Week

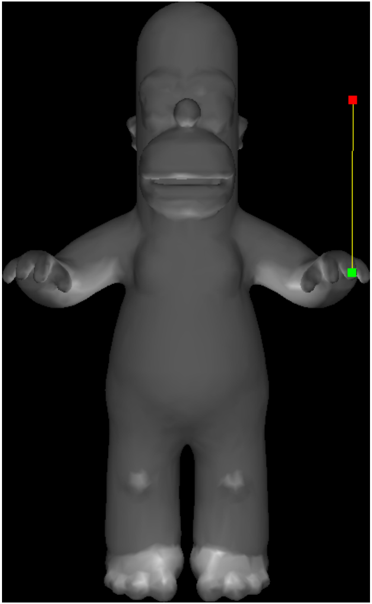
Table of Contents

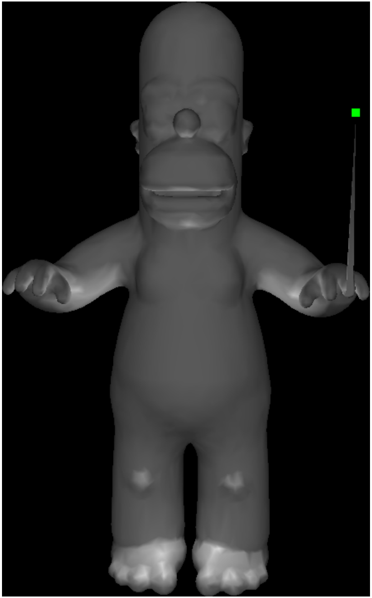
- ▶ Mesh Editing Overview / Discrete Curvature
- ▷ Laplacian Mesh Editing

Mesh Editing Overview



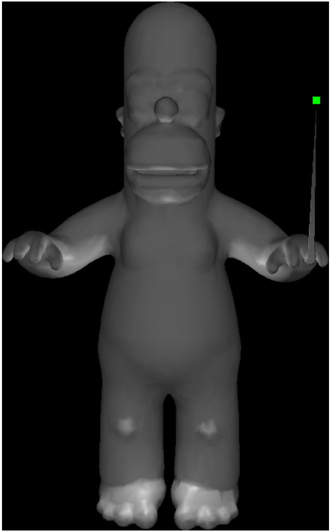
Mesh Editing Overview

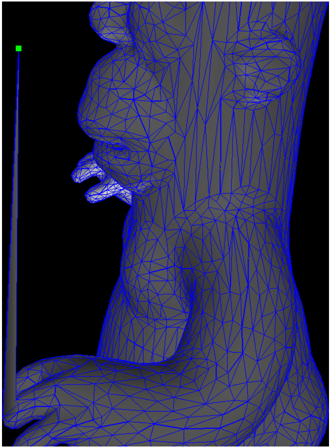




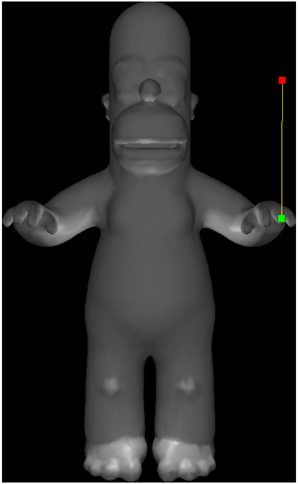
Mesh Editing Overview

This is not what we want!

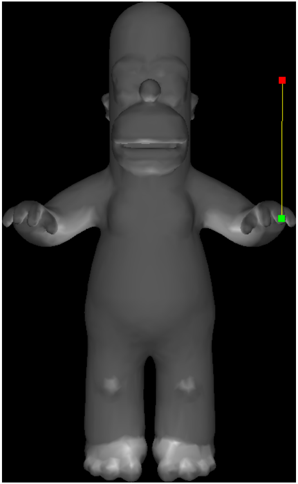


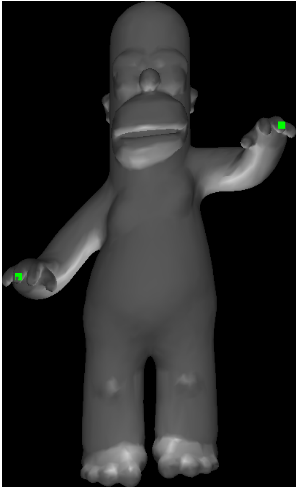


Mesh Editing Overview



Mesh Editing Overview

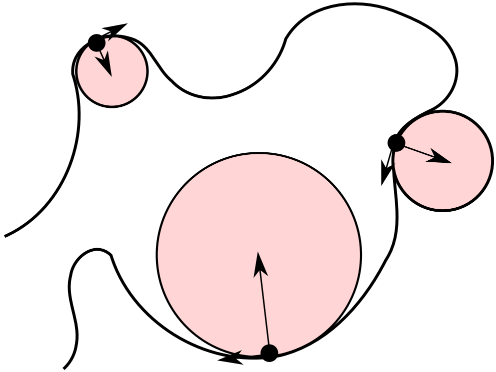




This is much better! Preserve *relative information* about points to neighbors

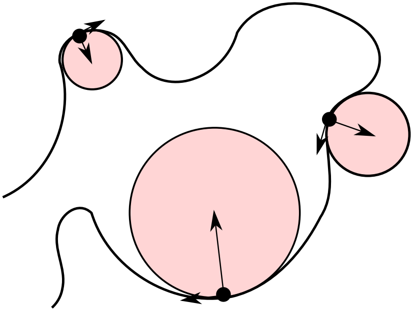
Parameterized Curves / Curvature

Curvature κ is $\frac{1}{r}$, where r is radius of *osculating circle*



Parameterized Curves / Curvature

Curvature κ is $\frac{1}{r}$, where *r* is radius of *osculating circle*



Curvature can also be considered as a vector $\kappa\vec{n}$

Parameterized Curves / Curvature

$$\gamma(t) = (x(t), y(t))$$

Velocity

$$\gamma'(t) = \left(\frac{dx(t)}{dt}, \frac{dy(t)}{dt}\right)$$

Parameterized Curves / Curvature

$$\gamma(t) = (x(t), y(t))$$

Velocity

$$\gamma'(t) = \left(\frac{dx(t)}{dt}, \frac{dy(t)}{dt}\right)$$

Assume *parameterized by arc length*; that is, curve moving at a unit speed. In other words

$$\gamma'(t) \cdot \gamma'(t) = 1$$

Parameterized Curves / Curvature

$$\gamma(t) = (x(t), y(t))$$

Velocity

$$\gamma'(t) = \left(\frac{dx(t)}{dt}, \frac{dy(t)}{dt}\right)$$

Assume *parameterized by arc length*; that is, curve moving at a unit speed. In other words

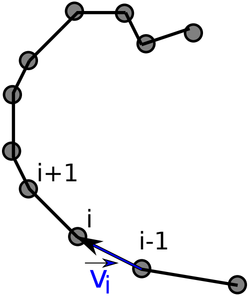
$$\gamma'(t) \cdot \gamma'(t) = 1$$

Differentiate both sides with respect to t, use product rule, end up with

$$2\gamma''(t) \cdot \gamma'(t) = 0 \implies \gamma''(t) \perp \gamma'(t)$$

$\gamma''(t) = \kappa \vec{n(t)}$ is the *curvature vector*

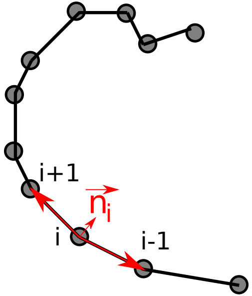
Discrete Curvature



Derivative at point *i* (velocity vector) is approximately

$$\vec{v_i} = \vec{x_i} - \vec{x_{i-1}}$$

Discrete Curvature



Derivative at point *i* (velocity vector) is approximately

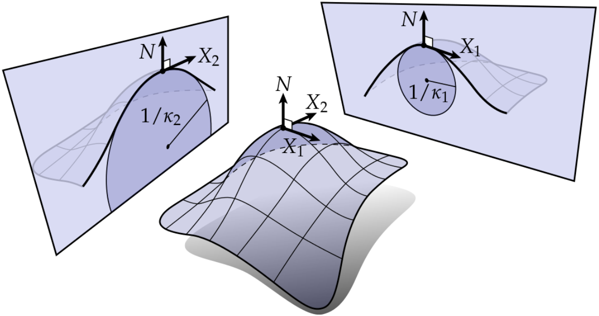
$$\vec{v_i} = \vec{x_i} - \vec{x_{i-1}}$$

Second derivative (curvature vector) is approximately

$$\vec{n_i} = \vec{v_{i+1}} - \vec{v_i} = \vec{x_{i+1}} - \vec{x_i} - (\vec{x_i} - \vec{x_{i-1}}) = -2\vec{x_i} + \vec{x_{i-1}} + \vec{x_{i+1}}$$

Curvature of Surfaces

Cut surface with plane, look at curvature of curve going through point

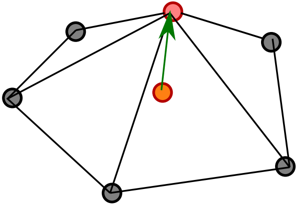


Courtesy of Keenan Crane, *Discrete Differential Geometry: An Applied Introduction*

Discrete Mean Curvature Approximation

Still just a difference of a point with its neighbors! Convention is negative the curvature vector: $-H(v_i)\vec{n}_i$, where $H(v_i)$ is the *mean curvature*

Example with curvature



Example with flat

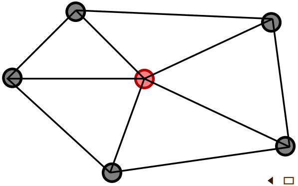
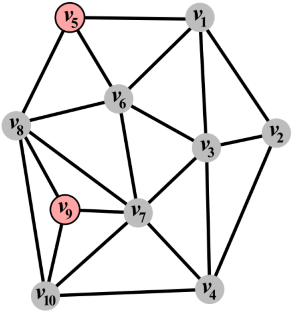


Table of Contents

- ▷ Mesh Editing Overview / Discrete Curvature
- ▶ Laplacian Mesh Editing

Laplacian Mesh Matrix



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

Sorkine05

$$
L_{ij} = \left\{ \begin{array}{cc} -1 & \text{edge connecting i and j} \\ \text{degree}(i) & i = j \\ 0 & \text{otherwise} \end{array} \right\}
$$

Graph Laplacian

More generally, $L = D - A$

- A: "Adjacency matrix"

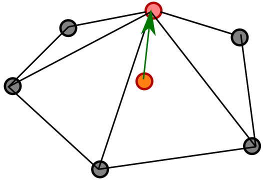
$$A_{ij} \left\{ \begin{array}{cc} 1 & \text{edge connecting i and j} \\ 0 & \text{otherwise} \end{array} \right\}$$

- D: "Degree matrix"

$$D_{ij} = \left\{ \begin{array}{cc} \text{degree}(i) = \sum_{j=1}^{N} A_{ij} & i = j \\ 0 & \text{otherwise} \end{array} \right\}$$

L is *symmetric* and *sparse*. Number of nonzero entries is $O(N)$ for meshes of constant genus

Laplacian Mesh Editing

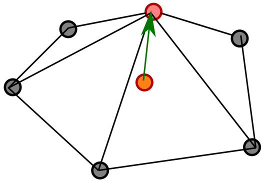


$$\delta_i = \sum_{j \in N(i)} (v_i - v_j) = d_i v_i - \sum_{j \in N(i)} v_j$$

Can be written as $Lv = \delta$. Each vector is *along a row* now

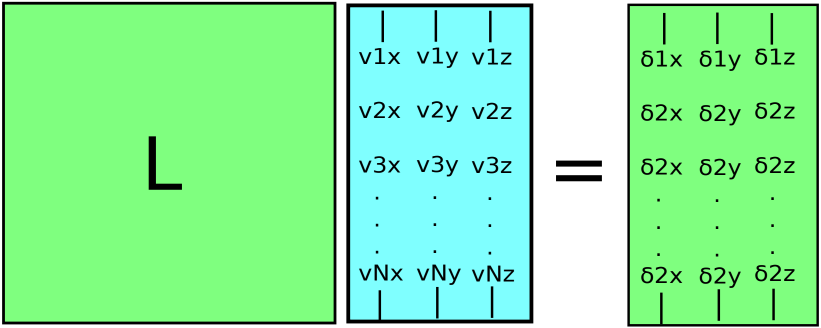
$$L \begin{bmatrix} | & | & | \\ v1x & v1y & v1z \\ v2x & v2y & v2z \\ v3x & v3y & v3z \\ \vdots & \vdots & \vdots \\ vNx & vNy & vNz \\ | & | & | \end{bmatrix} = \begin{bmatrix} | & | & | \\ \delta 1x & \delta 1y & \delta 1z \\ \delta 2x & \delta 2y & \delta 2z \\ \delta 2x & \delta 2y & \delta 2z \\ \vdots & \vdots & \vdots \\ \delta 2x & \delta 2y & \delta 2z \\ | & | & | \end{bmatrix}$$

Laplacian Mesh Editing

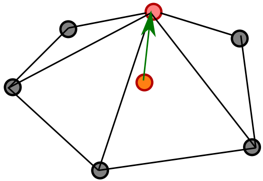


$$\delta_i = \sum_{j \in N(i)} (v_i - v_j)$$

Can we reconstruct v from δ?

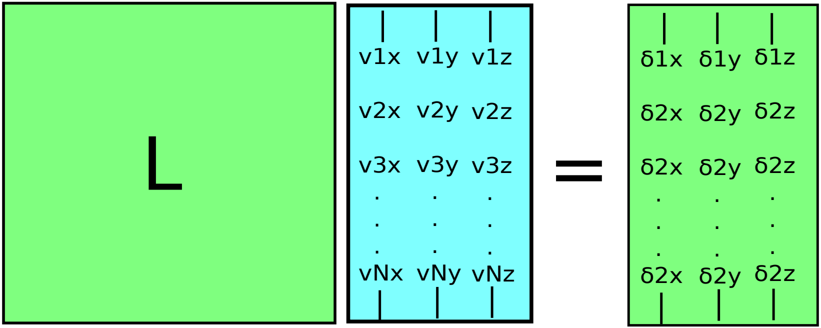


Laplacian Mesh Editing

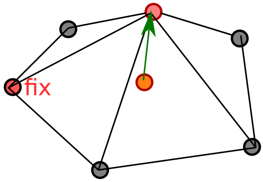


$$\delta_i = \sum_{j \in N(i)} (v_i - v_j)$$

Can we reconstruct v from δ? *No: L is rank $N-1$ for a connected mesh*

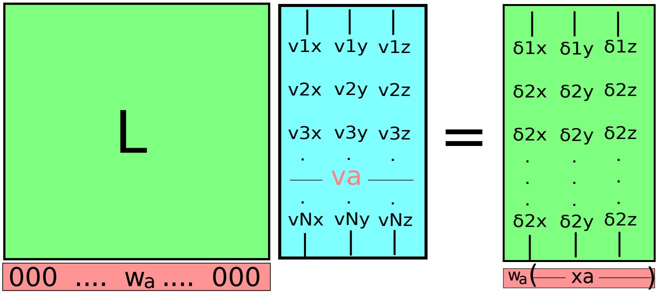


Laplacian Mesh Editing: Anchors

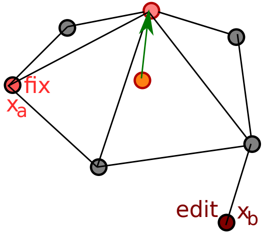


$$\delta_i = \sum_{j \in N(i)} (v_i - v_j)$$

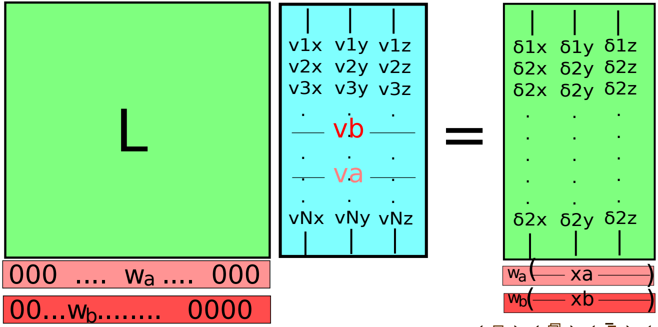
Delta coordinates define geometry *up to a translation*. Fix a point v_a, fix translation



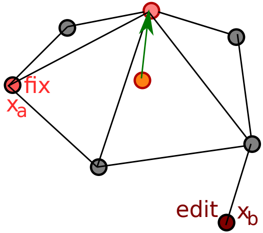
Laplacian Mesh Editing: Anchors



Can add more anchors, but may not be a solution

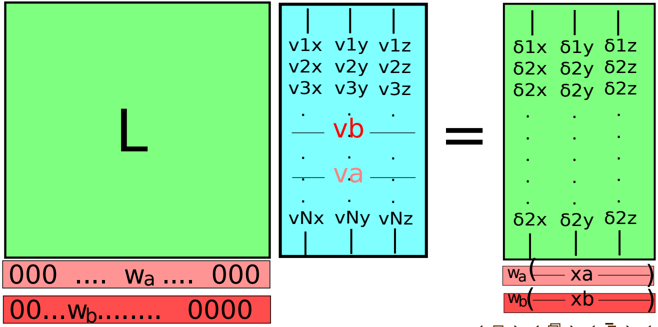


Laplacian Mesh Editing: Anchors



Can add more anchors, but may not be a solution Solve in the *least squares sense*

$$\tilde{v} = \text{argmin}_v ||Lv - \delta||_2^2 + \sum_{s=1}^{k} w_s ||v_s - x_s||_2^2$$



Laplacian Mesh Editing: Anchors: Another Example

Laplacian Mesh Editing: Anchors

$$\widetilde{v} = \text{argmin}_v ||Lx - \delta||_2^2 + \sum_{s=1}^{k} w_s ||x_s - v_s||_2^2$$

Laplacian Mesh Editing: Anchors

$$\tilde{v} = \text{argmin}_v ||Lx - \delta||_2^2 + \sum_{s=1}^{k} w_s ||x_s - v_s||_2^2$$

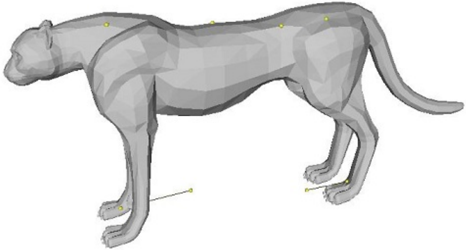
- Let $\overline{L}$ be L augmented with the anchor rows
- Let $\overline{\delta}$ be δ augmented with the weighted anchor coordinates

Can all be written in matrix form

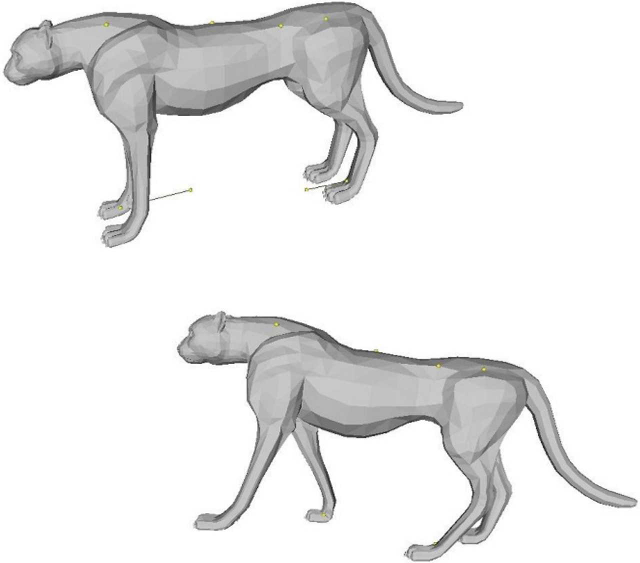
$$\text{Squared Error: } \epsilon(v) = ||\overline{L}v - \overline{\delta}||_2^2$$

$$\text{Least Squares Solution: } v^* = (\overline{L}^T \overline{L})^{-1} \overline{L}^T \overline{\delta}$$

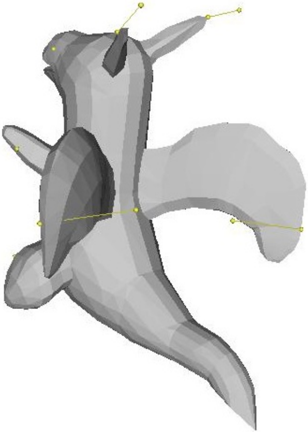
Laplacian Mesh Editing: Examples



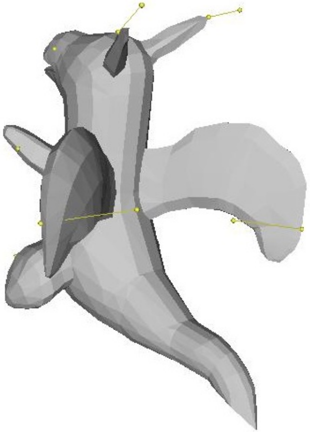
Laplacian Mesh Editing: Examples

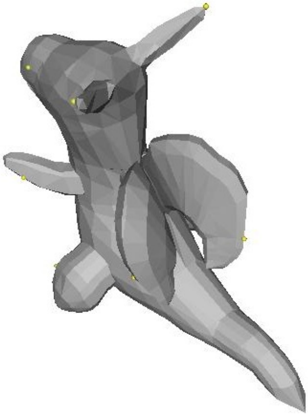


Laplacian Mesh Editing: Examples

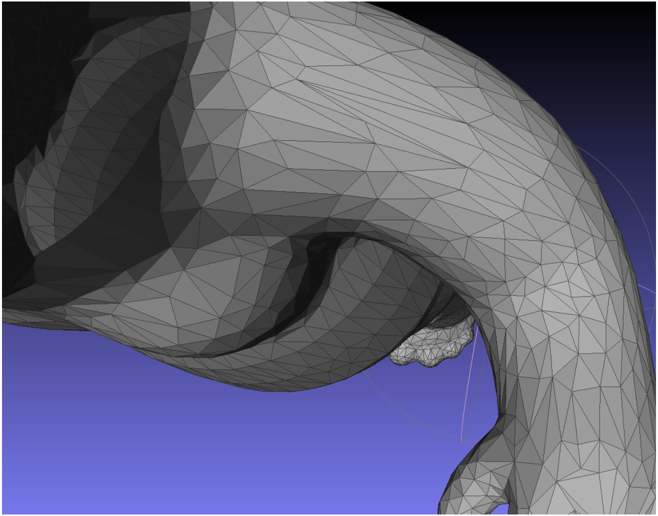


Laplacian Mesh Editing: Examples



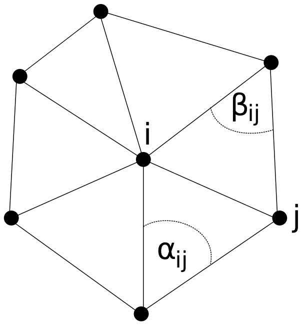


What About Irregular Meshes?



Homer's upper arm

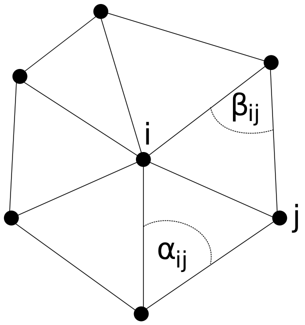
Cotangent Weights



$$\delta_i = \sum_{j \in N(i)} w_{ij}(v_i - v_j)$$

$$w_{ij} = \frac{1}{2}(\cot(\beta_{ij}) + \cot(\alpha_{ij}))$$

Cotangent Weights



$$\delta_i = \sum_{j \in N(i)} w_{ij}(v_i - v_j)$$

$$w_{ij} = \frac{1}{2}(\cot(\beta_{ij}) + \cot(\alpha_{ij}))$$

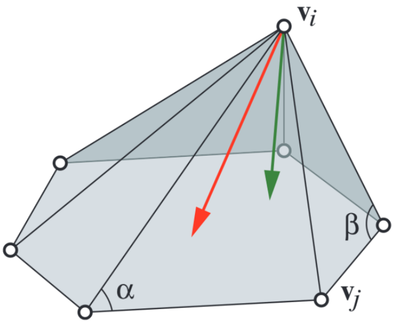
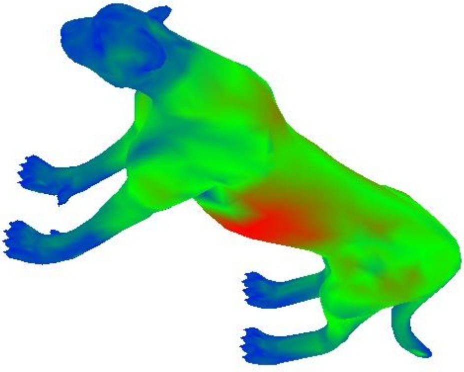


Figure: Nealen2006, umbrella vs cotangent

Cotangent Weights: Mean Curvature

Mean curvature is approximated by

$$\frac{1}{2}\sum_{j\in N(i)}(\cot(\beta_{ij})+\cot(\alpha_{ij}))||\vec{v}_i-\vec{v}_j||_2$$



Laplacian Mesh Editing: Umbrella vs Cotangent

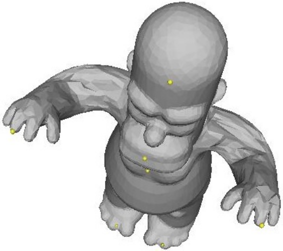


Figure: umbrella

Laplacian Mesh Editing: Umbrella vs Cotangent

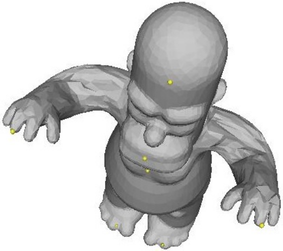


Figure: umbrella

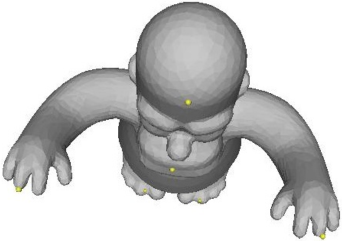


Figure: cotangent

Laplacian Mesh Editing: Umbrella vs Cotangent

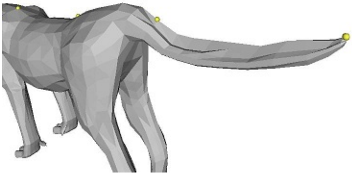


Figure: umbrella

Laplacian Mesh Editing: Umbrella vs Cotangent

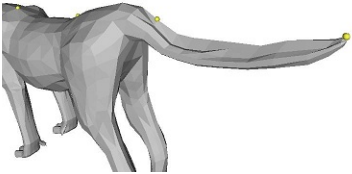


Figure: umbrella

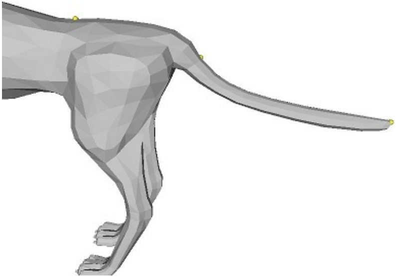
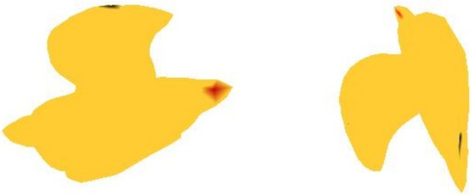
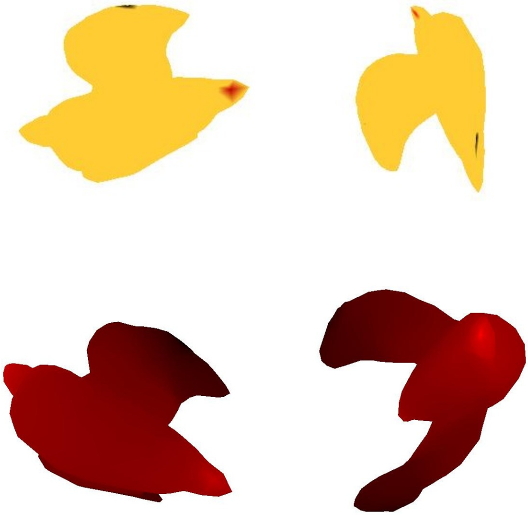


Figure: cotangent

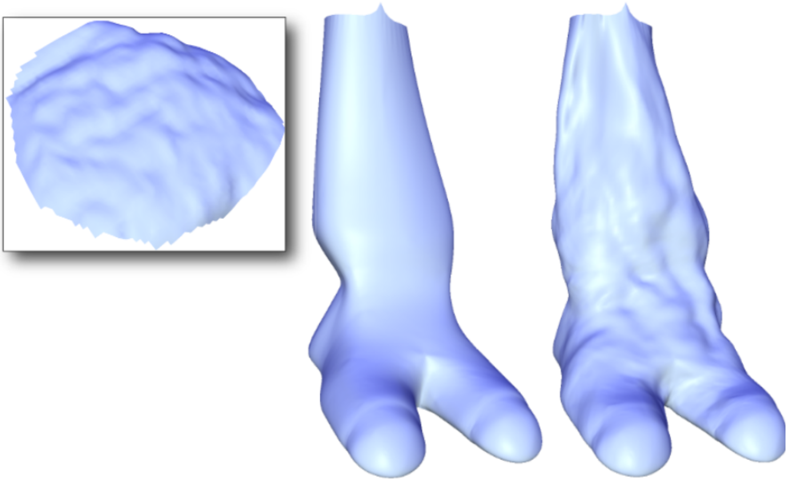
Applications: Function Interpolation



Applications: Function Interpolation

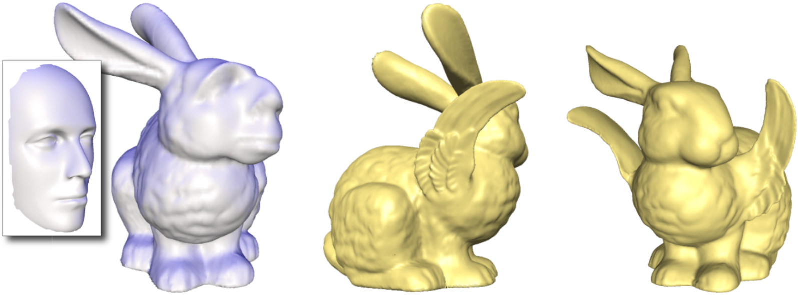


Applications: Detail Transfer / Mesh Mixing



Sorkine 05

Applications: Detail Transfer / Mesh Mixing



Sorkine 05

Applications

More surprises in group assignment 3 and the following lectures!

A Note About Rotation Invariance

$$Lx = \delta$$

- δ is a *vector*. $||\delta|| \propto \kappa$, but it has a direction

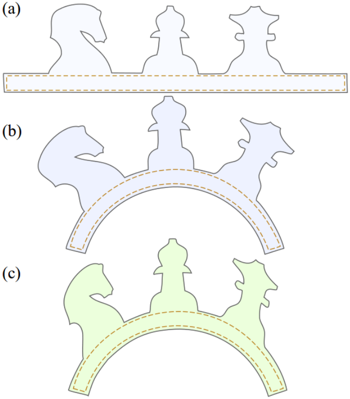
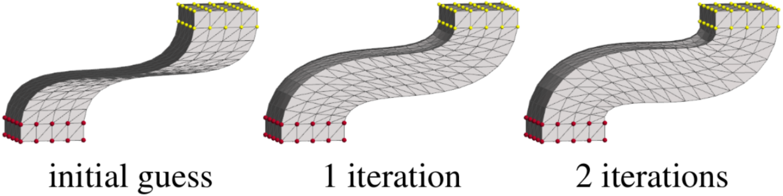


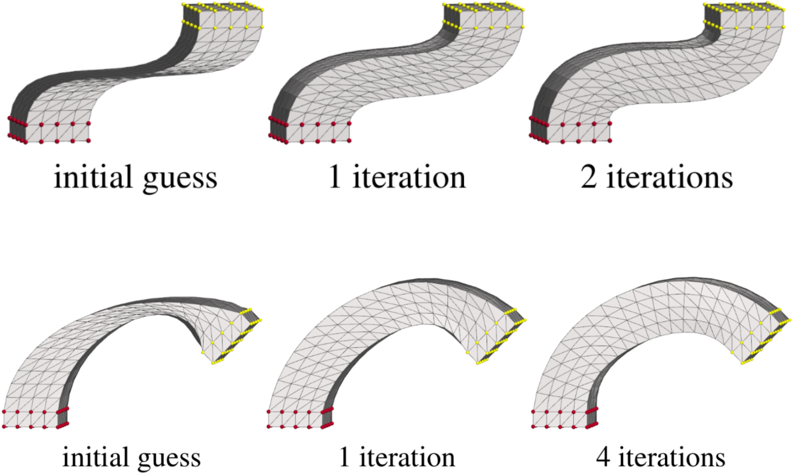
Figure: Sorkine05

As Rigid As Possible Surface Editing



Sorkine2007

As Rigid As Possible Surface Editing



Sorkine2007

A Note On Sparse Matrices

Sparse matrices in numpy