

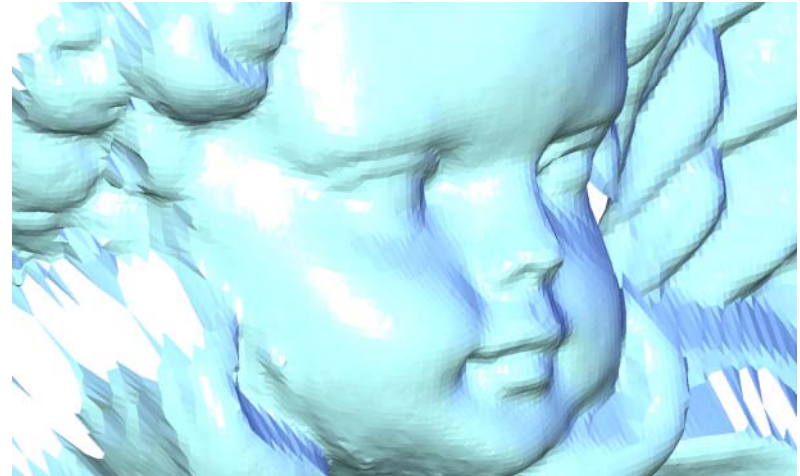
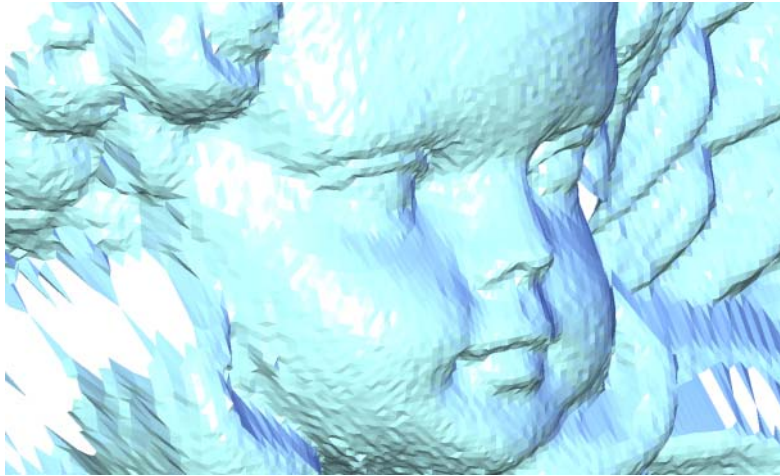
CS 523: Computer Graphics, Spring 2009

Shape Modeling

Laplacian mesh processing

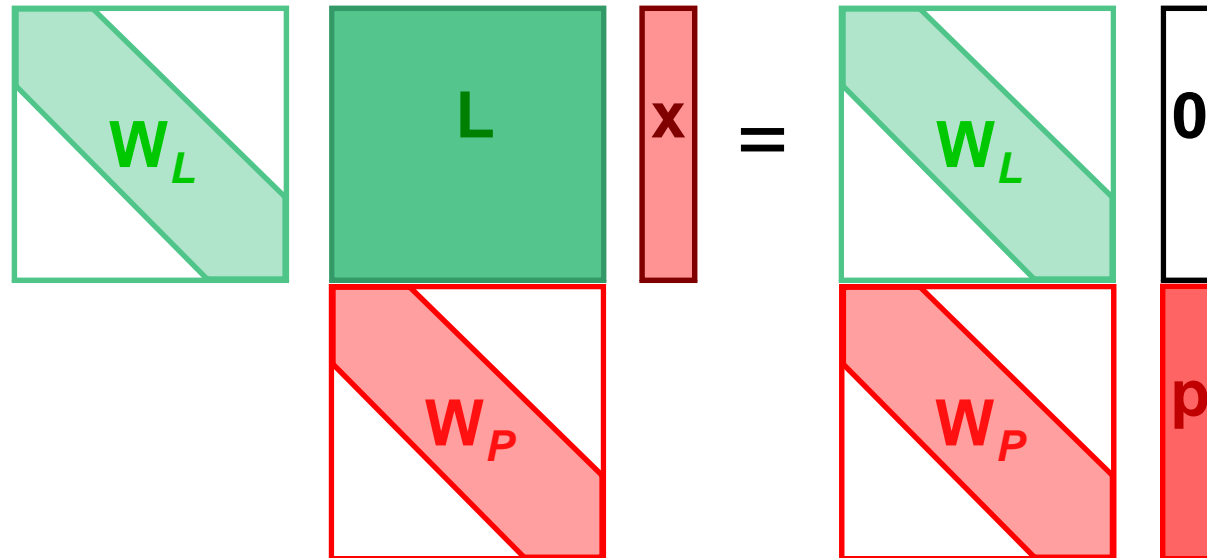
Laplacian mesh optimization

Reminder: mesh smoothing result



Laplacian mesh optimization

Reminder: mesh smoothing setup

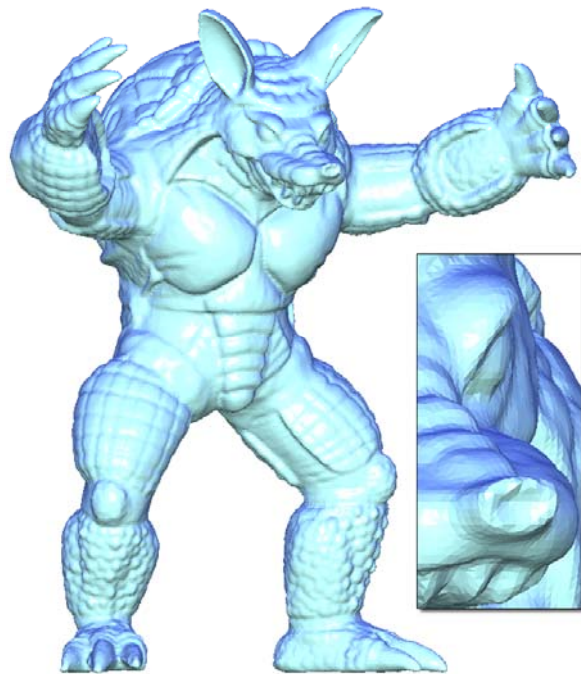


- **Mesh smoothing** $L = L_{cot}$ (outer fairness) or $L = L_{uni}$ (outer and inner fairness)
- Controlled by W_p and W_L (Intensity, Features)
- Least squares solve using normal equations

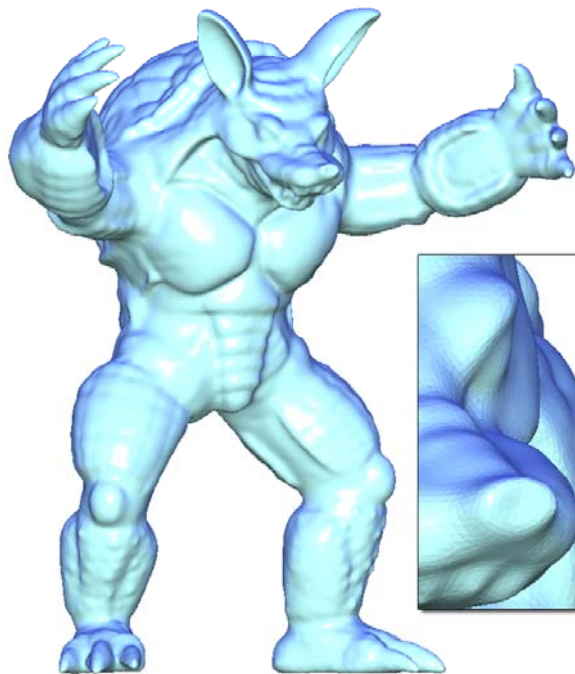
$$\begin{aligned} A^T A \mathbf{x} &= A^T \mathbf{b} \\ \mathbf{x} &= (A^T A)^{-1} A^T \mathbf{b} \end{aligned}$$

$$\begin{bmatrix} W_L & & \\ & L & \\ & & W_P \end{bmatrix} \begin{bmatrix} x \\ \\ p \end{bmatrix} = \begin{bmatrix} W_L & & \\ & 0 & \\ & & W_P \end{bmatrix} \begin{bmatrix} 0 \\ \\ p \end{bmatrix}$$

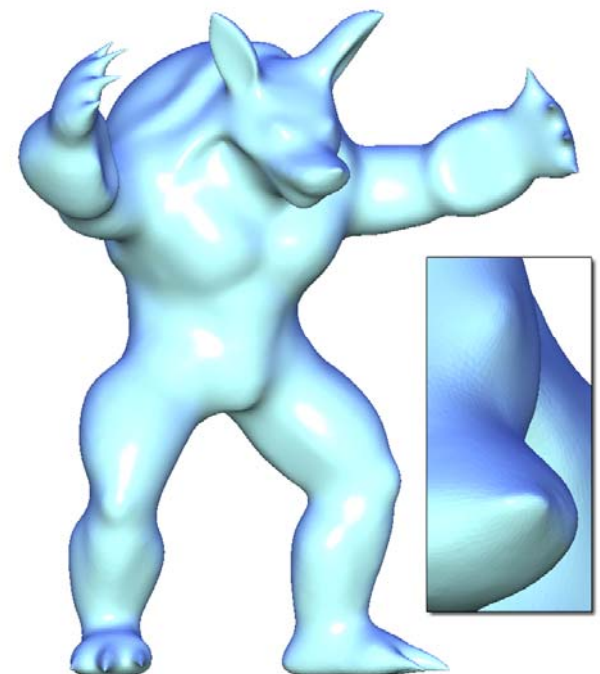
Using W_P



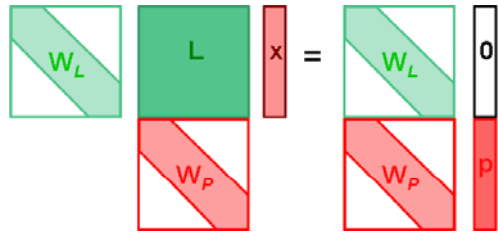
(a) original (173k)



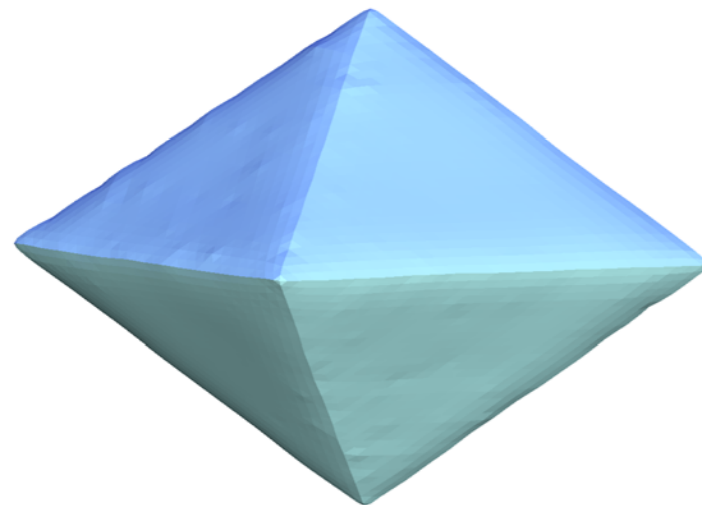
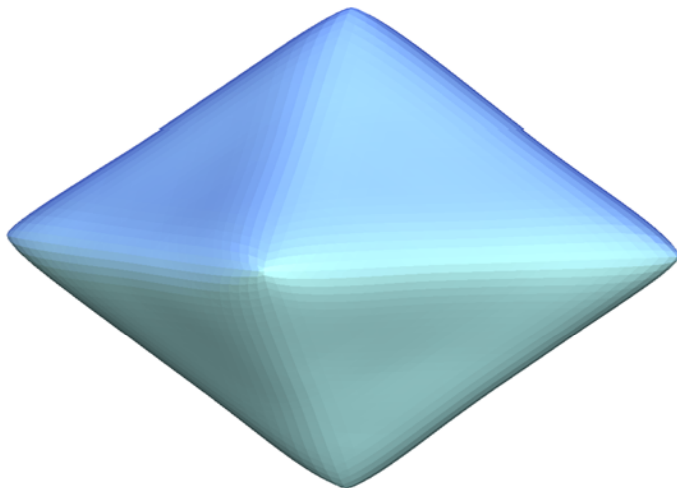
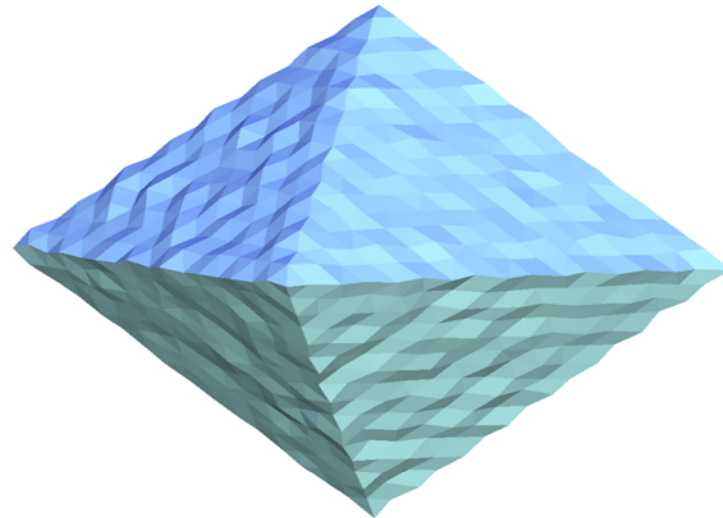
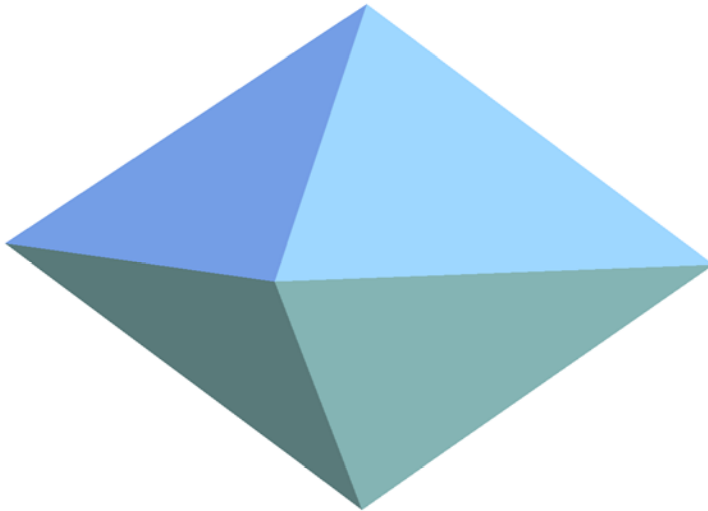
(c) cdf weights ($s = 0.2$)



(f) cdf weights ($s = 0.02$)

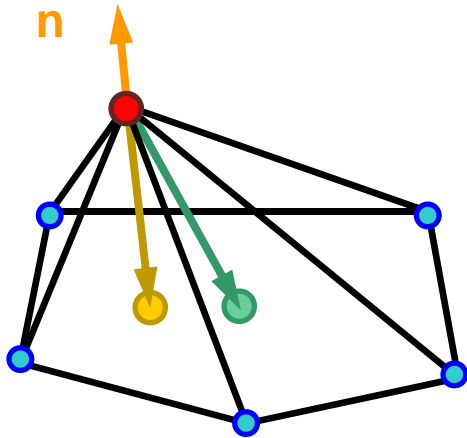


Using W_P and W_L



Laplacian Mesh Processing

- Discrete Laplacians



$$\mathbf{L} \mathbf{x} = \delta$$

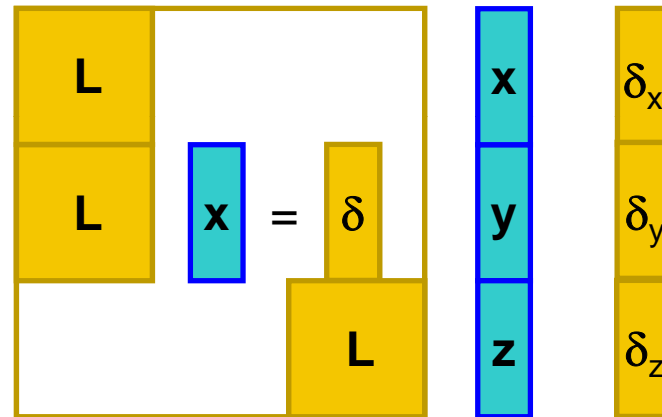
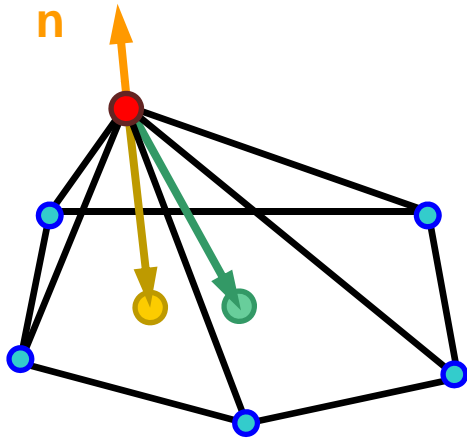
$$\delta_i = \mathbf{x}_i - \frac{1}{\sum_{(i,j) \in E} w_{ij}} \sum_{(i,j) \in E} w_{ij} \mathbf{x}_j$$

$$\delta_{\text{uniform}} : w_{ij} = 1$$

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

Laplacian Mesh Processing

- Surface reconstruction



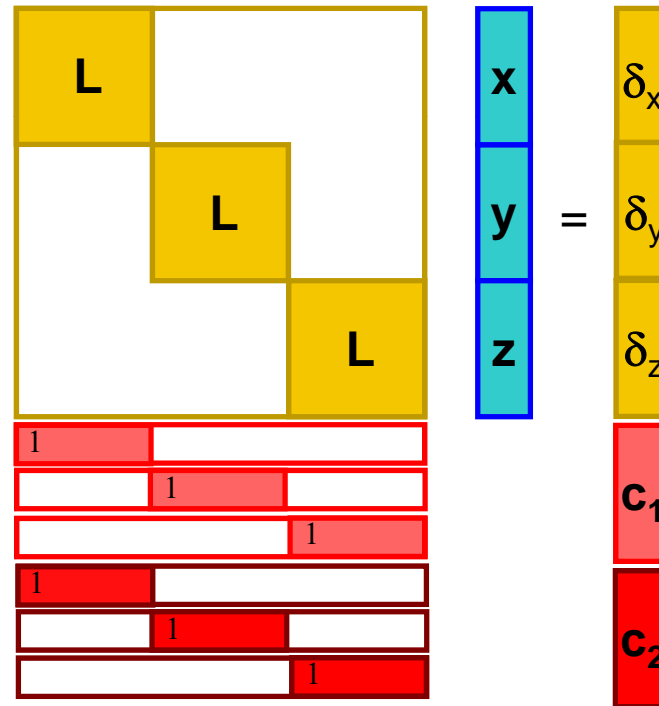
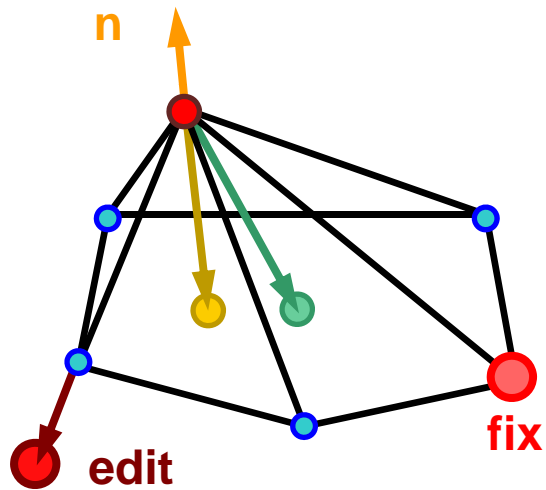
$$\delta_i = \mathbf{x}_i - \frac{1}{\sum_{(i,j) \in E} w_{ij}} \sum_{(i,j) \in E} w_{ij} \mathbf{x}_j$$

$$\delta_{\text{uniform}} : w_{ij} = 1$$

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

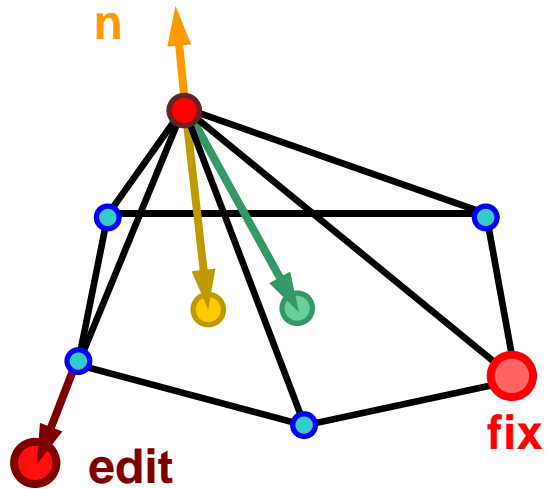
Laplacian Mesh Processing

- Surface reconstruction + editing



Laplacian Mesh Processing

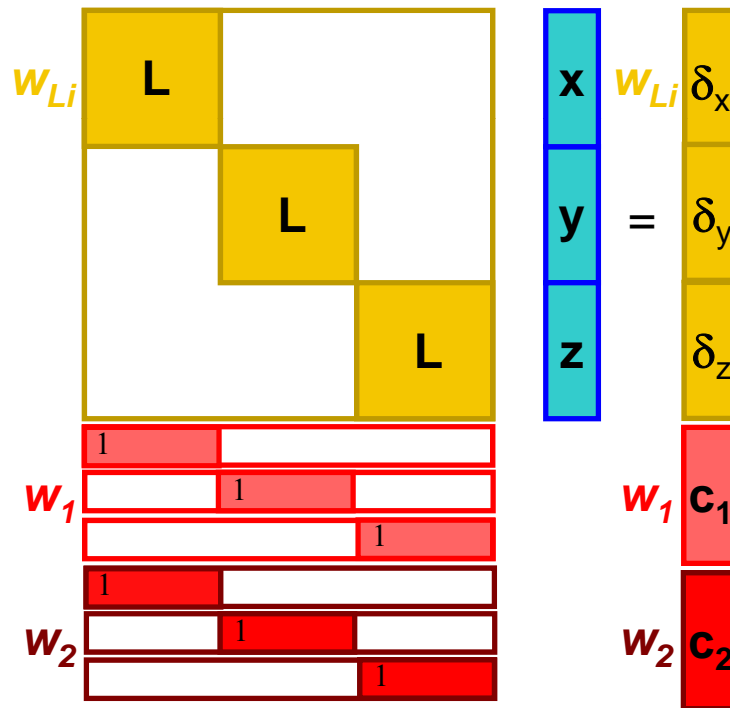
- Least-squares solution



Normal Equations

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

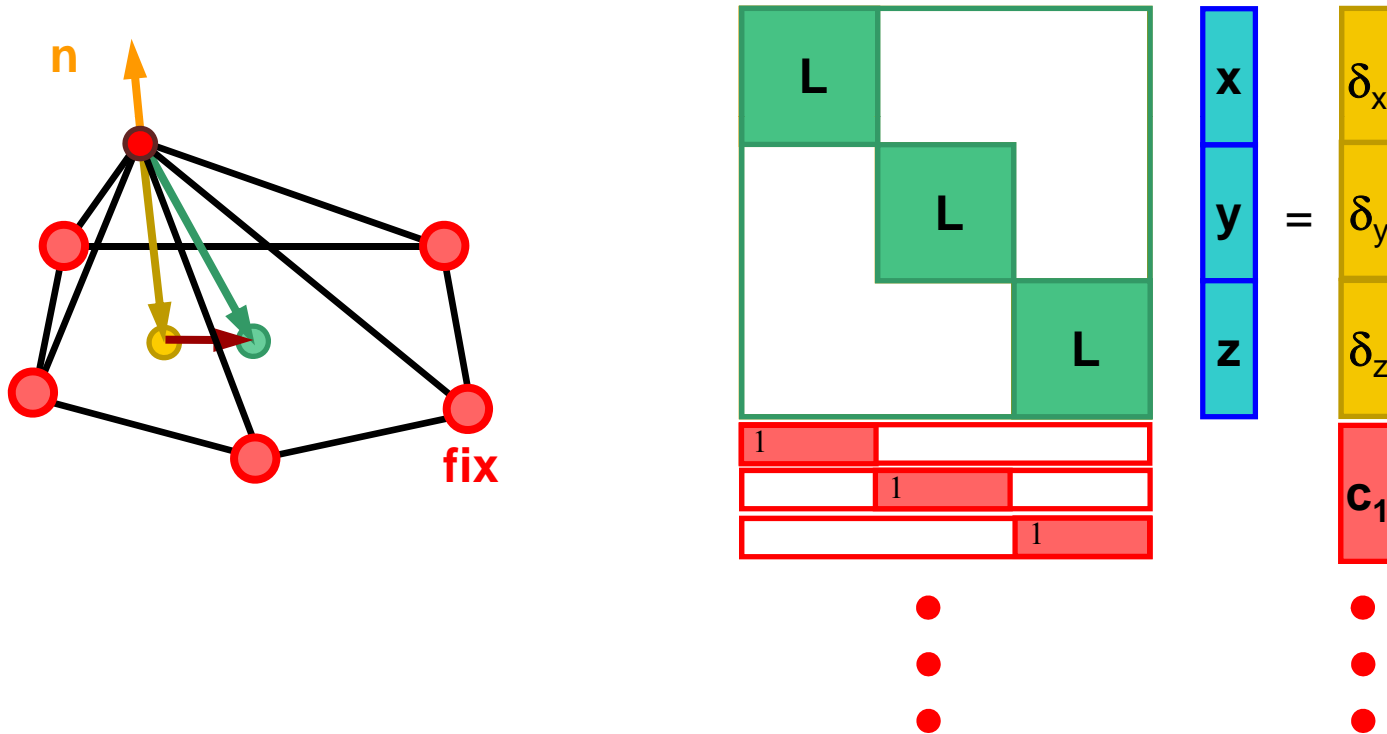
$$\mathbf{x} = (A^T A)^{-1} A^T \mathbf{b}$$



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

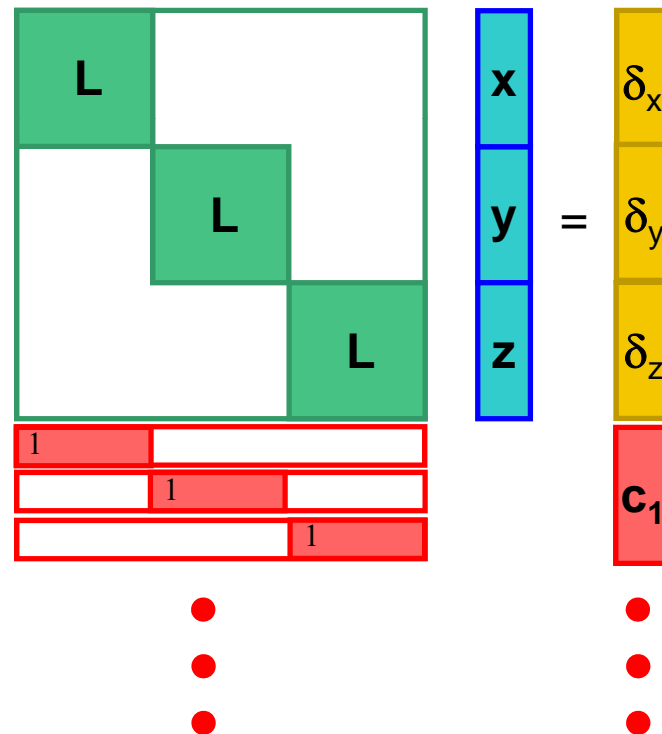
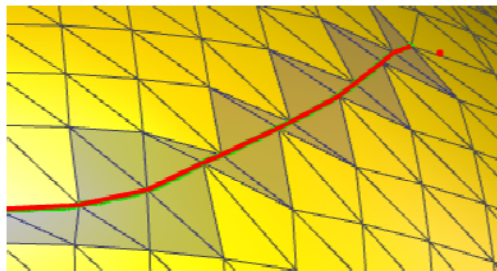
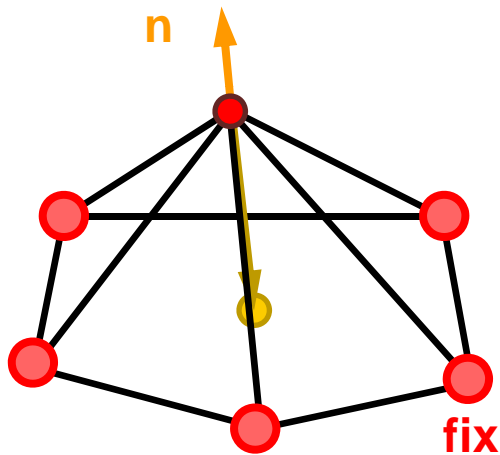
Laplacian Mesh Processing

- Tangential smoothing



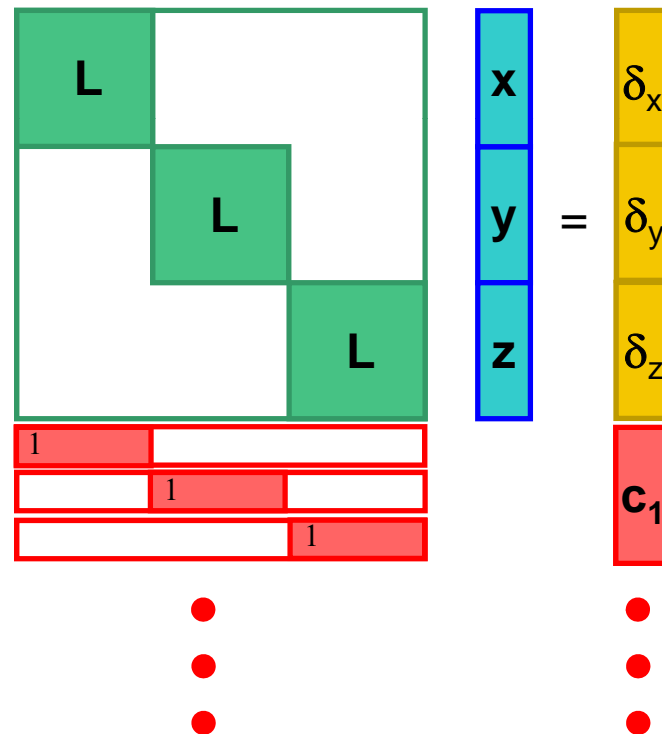
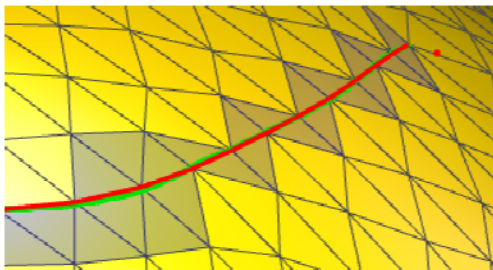
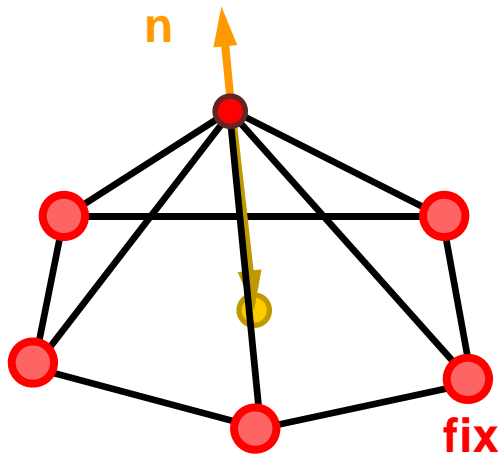
Laplacian Mesh Processing

- Tangential smoothing



Laplacian Mesh Processing

- Tangential smoothing

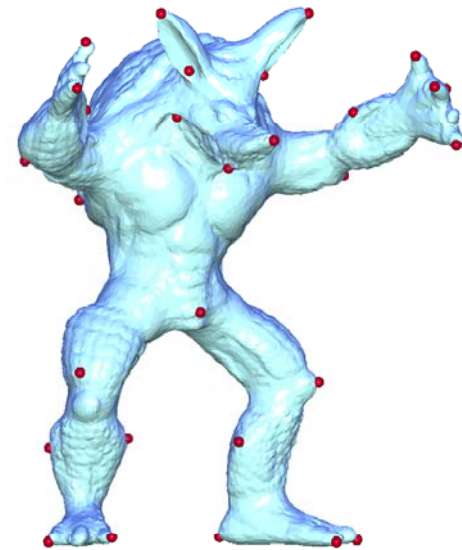


Idea

- Can we use such a system for **global** optimization ?

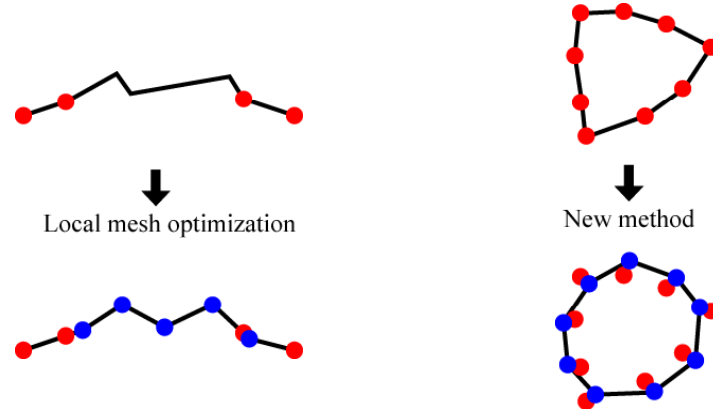


$$\mathbf{L} \quad \mathbf{x} = \delta$$



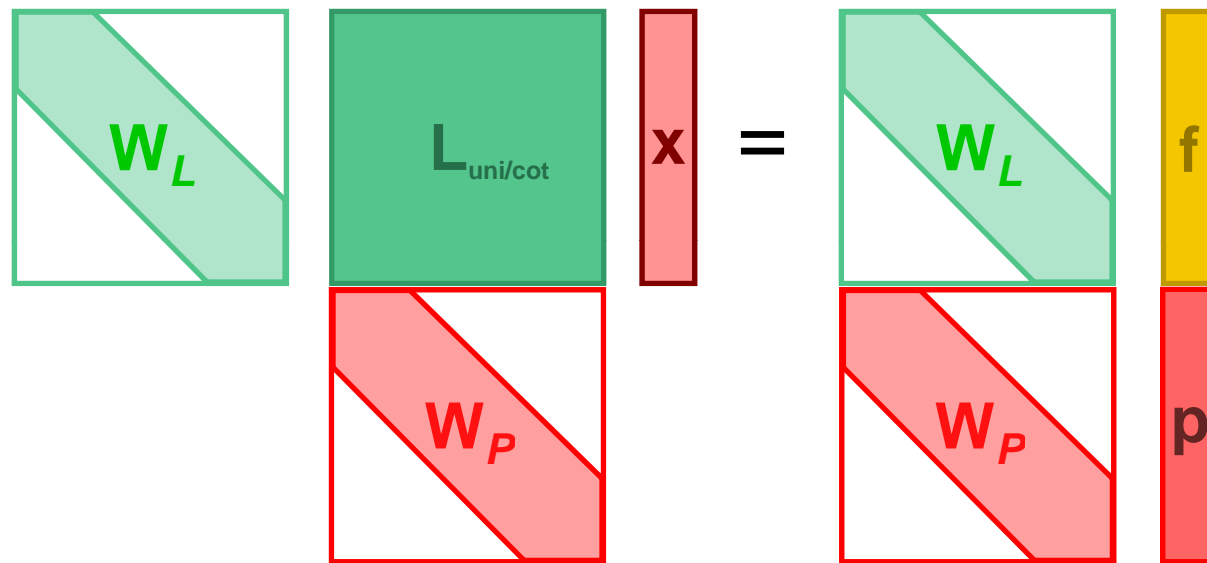
One solution

- All vertices are (weighted) anchors



- Preserves global shape
- Uses existing LS framework
- Anchor + Laplacian weights determine result

Laplacian mesh processing framework



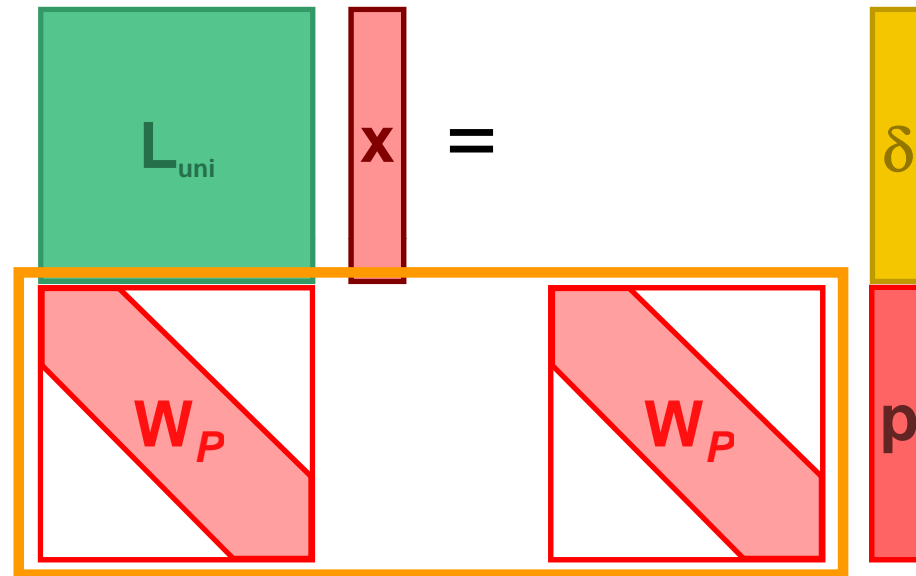
- **Detail preserving tri shape optimization** for $L = L_{uni}$ and $f = \delta_{cot}$ (similar to local optimization)
- **Mesh smoothing** $L = L_{cot}$ (outer fairness) or $L = L_{uni}$ (outer and inner fairness) and $f = 0$

Application: Triangle shape optimization

Global vertex relocation

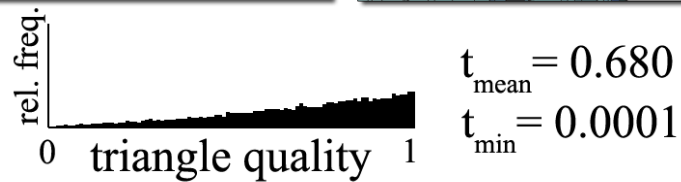
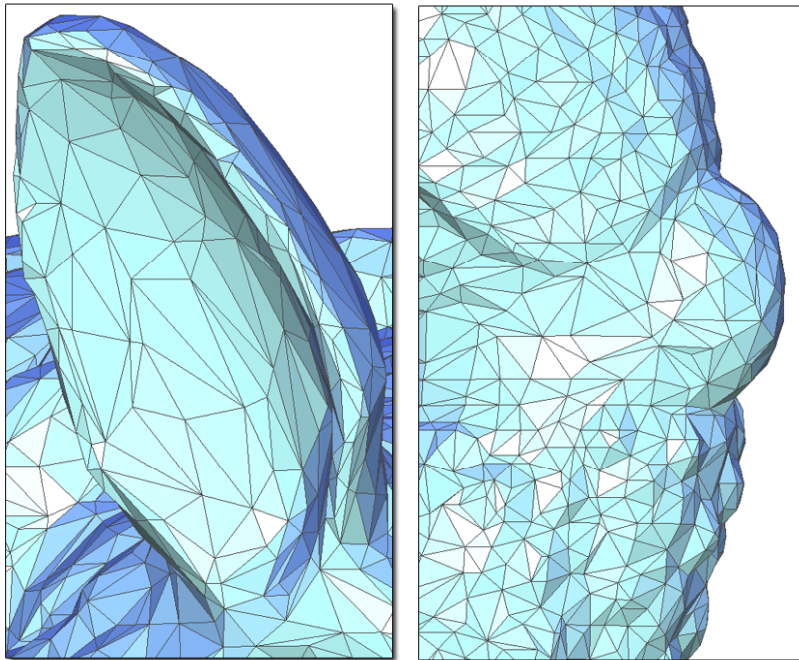
Triangle shape Optimization

By global vertex relocation

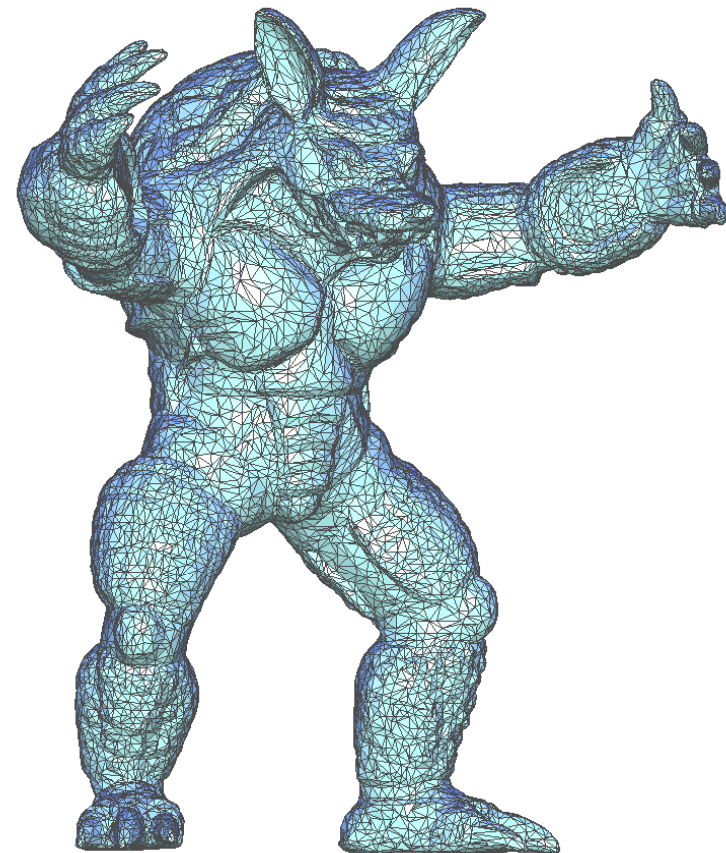


- **Detail preserving tri shape optimization** for $L = L_{uni}$ and $f = \delta_{cot}$ (similar to local optimization)

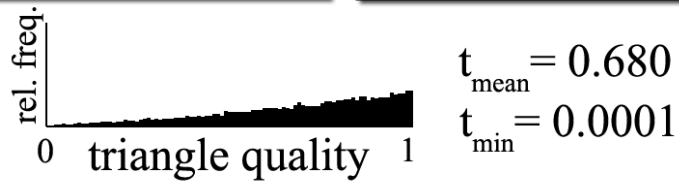
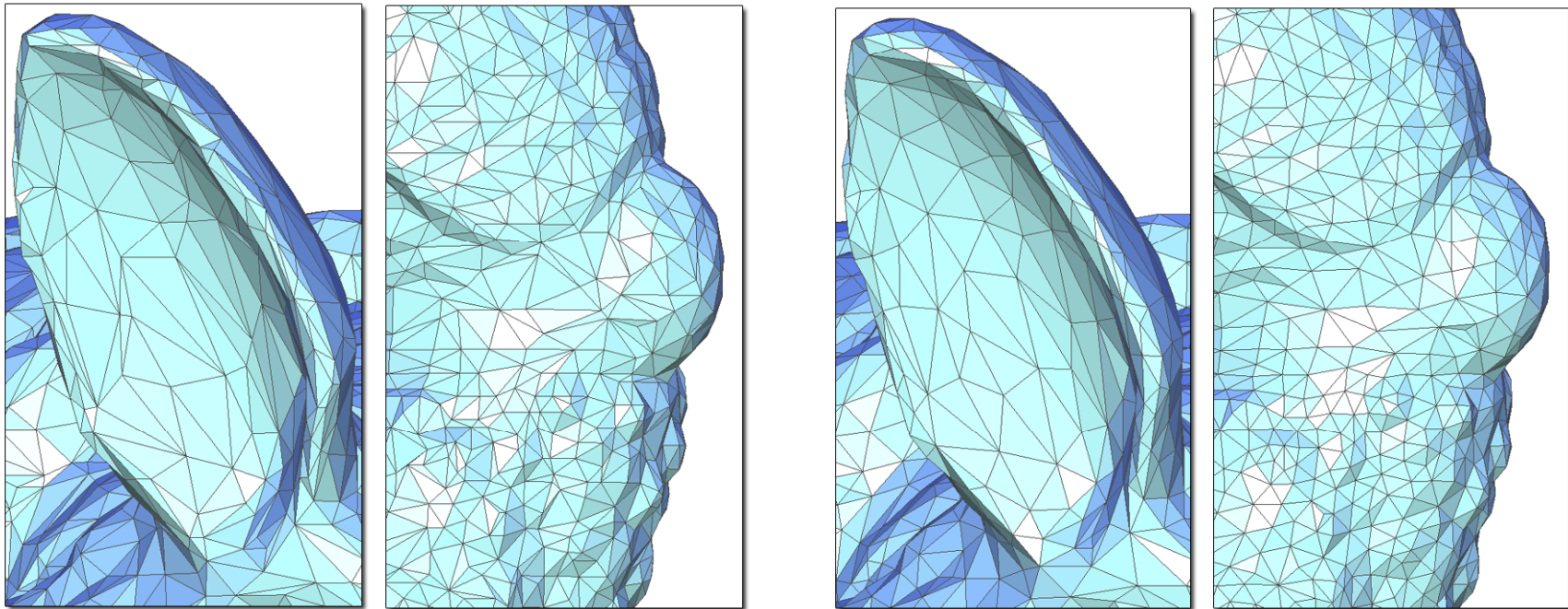
Positional Weights



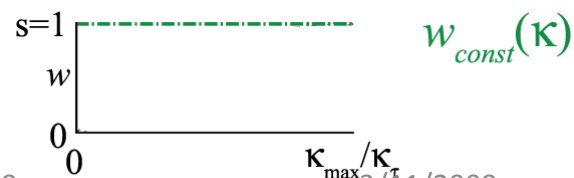
(a) original (17k)



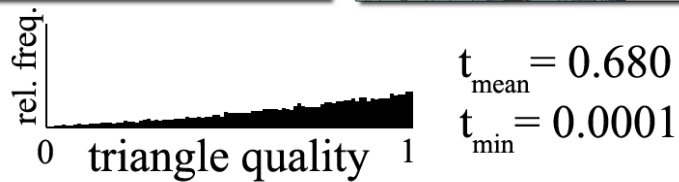
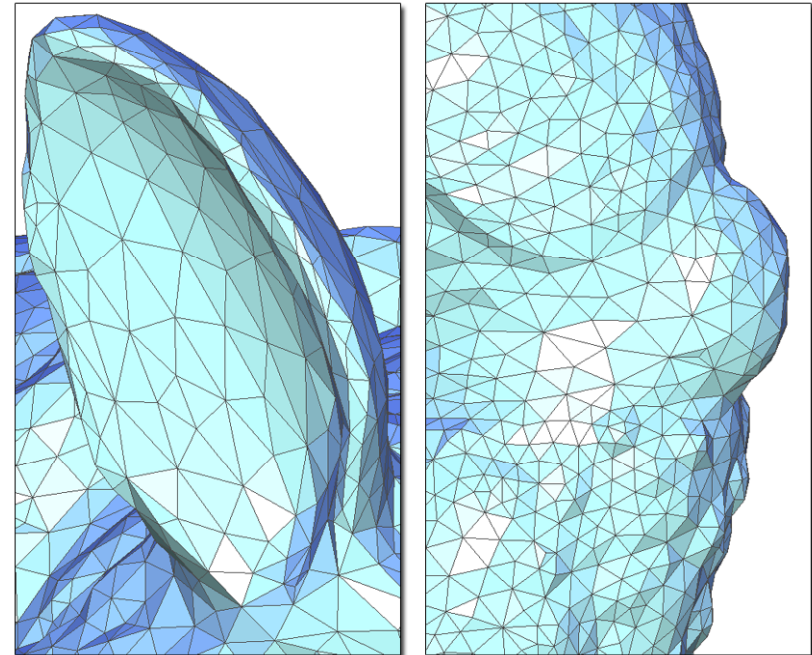
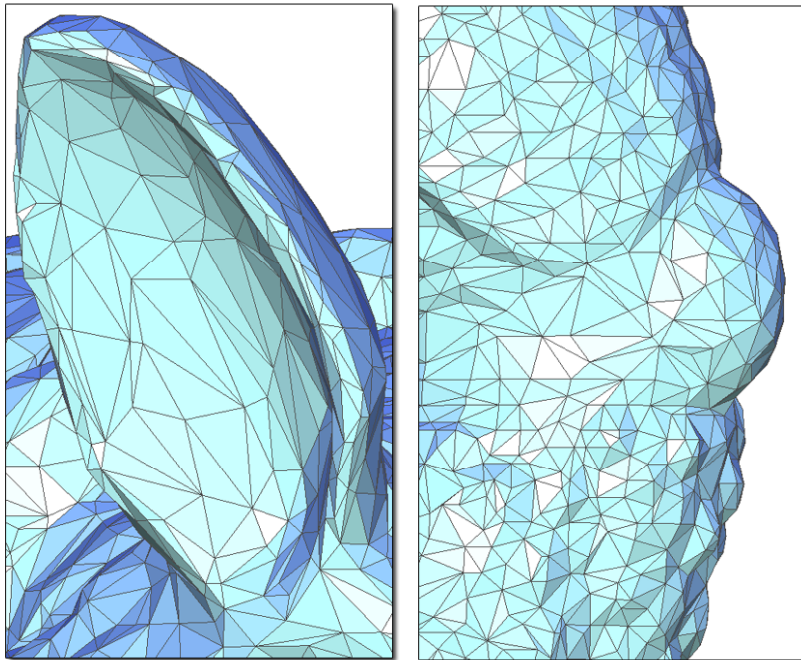
Constant Weights



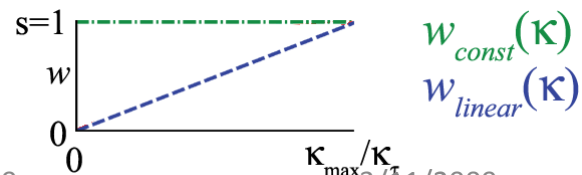
dist = $1.24 \cdot 10^{-3}$



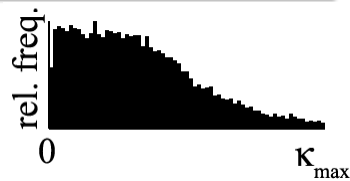
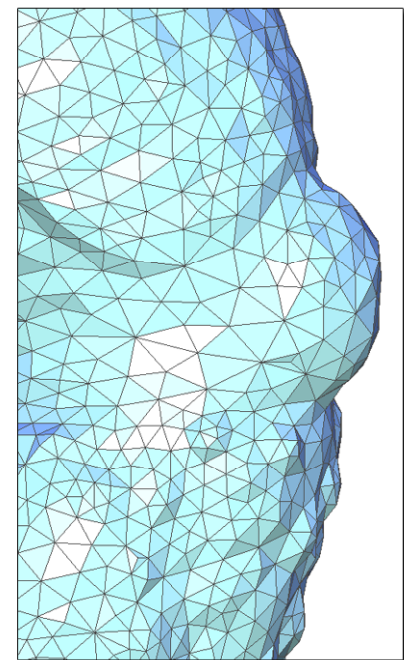
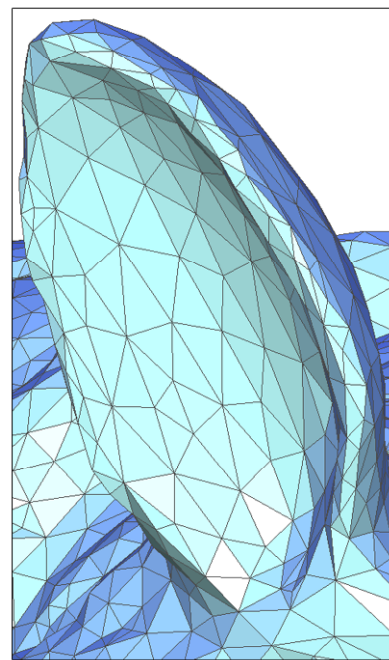
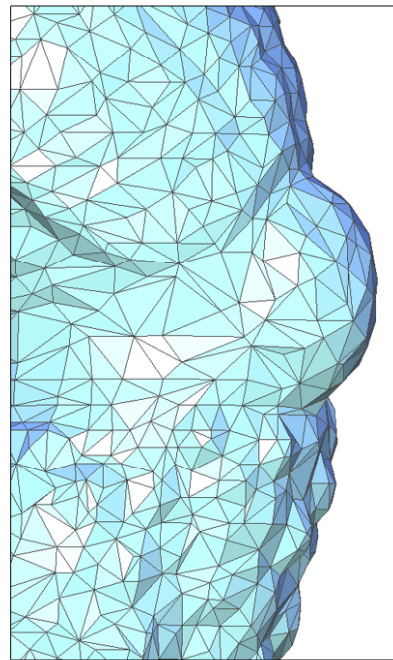
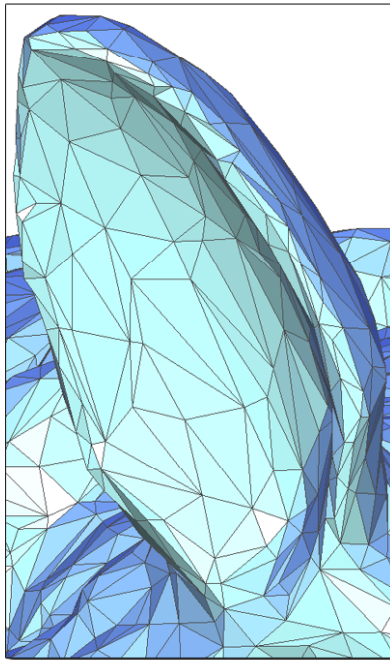
Linear Weights



dist = $2.53 \cdot 10^{-3}$



CDF Weights

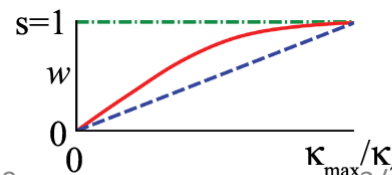


mean curvature
distribution $c(\kappa)$

dist = $2.53 \cdot 10^{-3}$



Andrew Nealen, Rutgers, 2009



$w_{const}(\kappa)$

$w_{linear}(\kappa)$

$w_{cdf}(\kappa)$

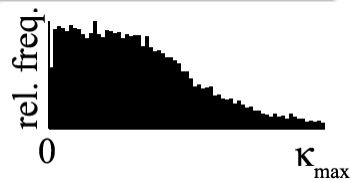
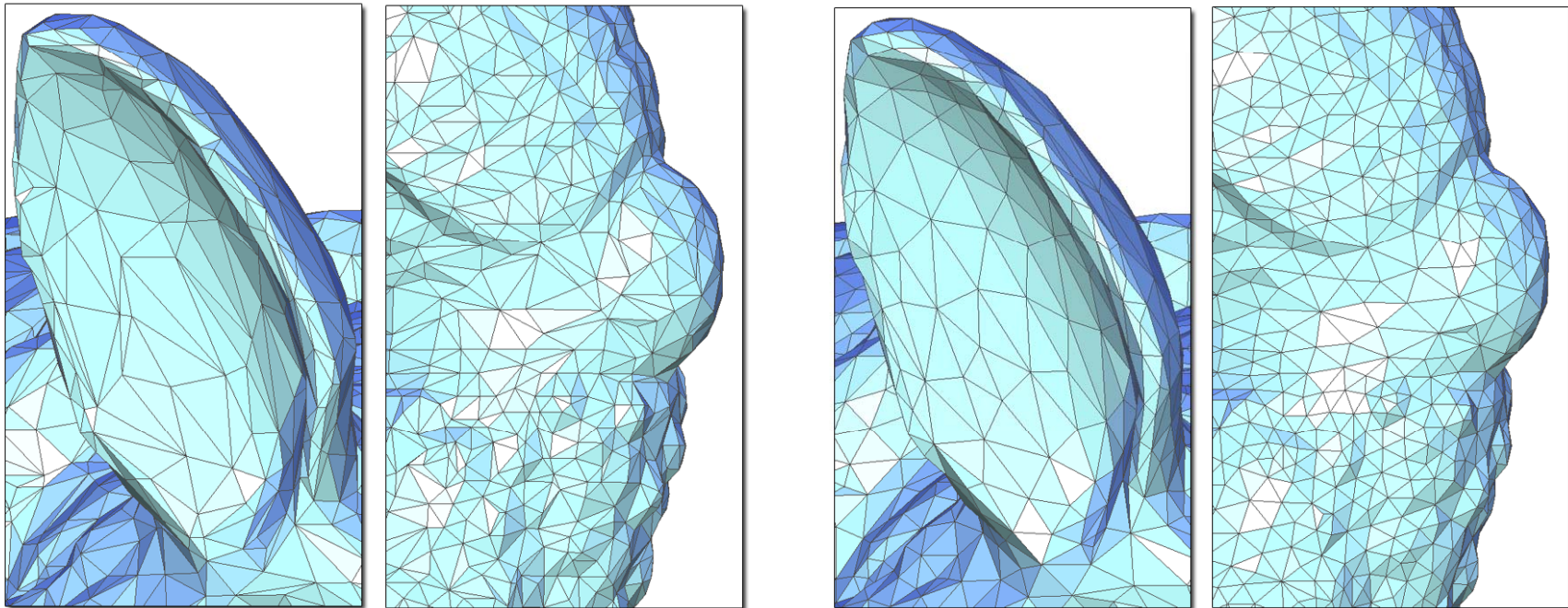


$t_{\text{mean}} = 0.842$

$t_{\text{min}} = 0.040$

3/11/2009

CDF Weights

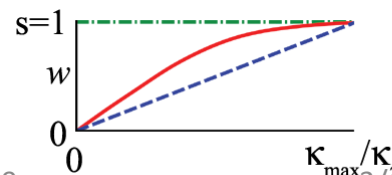


mean curvature distribution $c(\kappa)$

dist = $2.04 \cdot 10^{-3}$



Andrew Nealen, Rutgers, 2009



$w_{const}(\kappa)$
 $w_{linear}(\kappa)$
 $w_{cdf}(\kappa)$

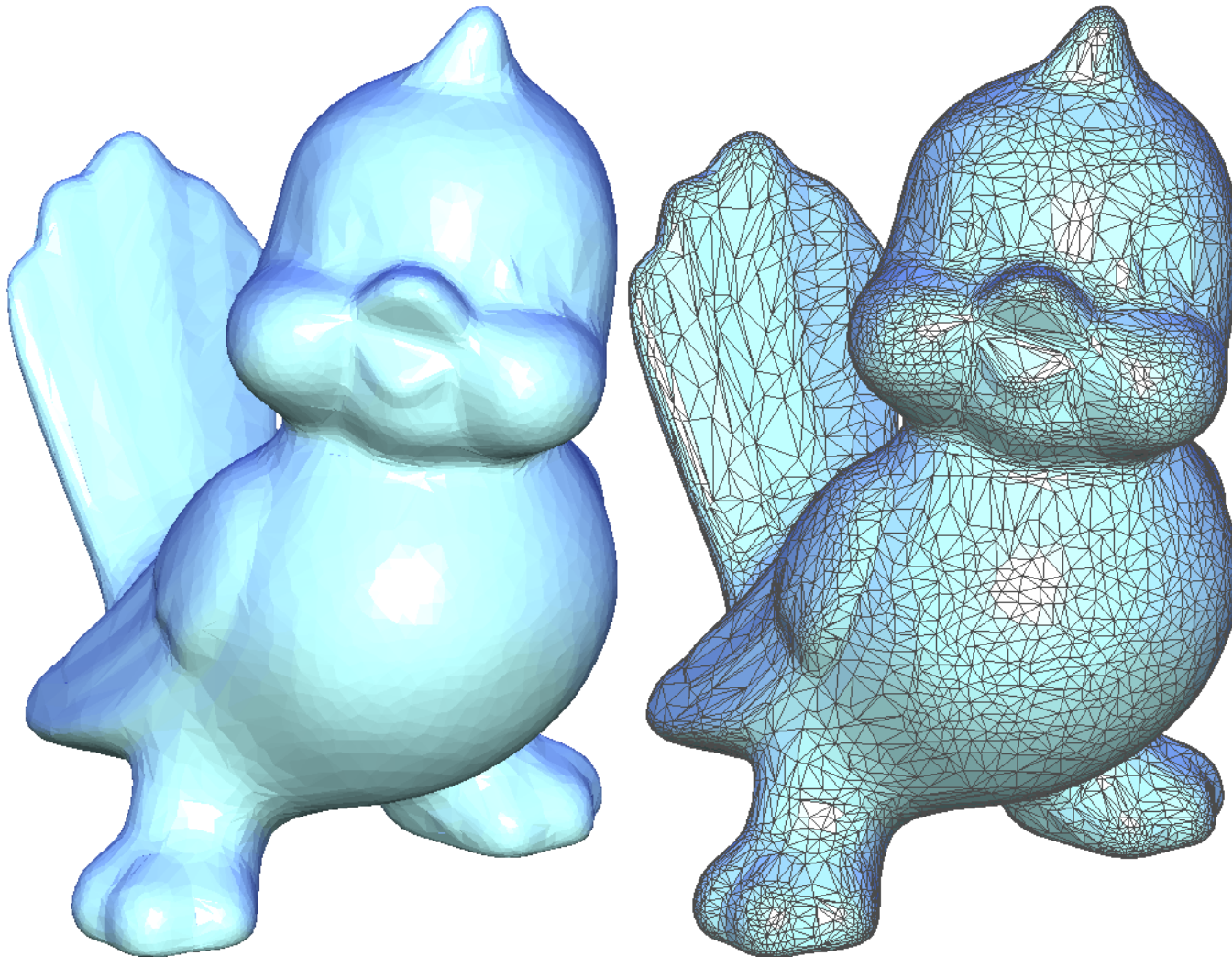


$t_{\text{mean}} = 0.826$

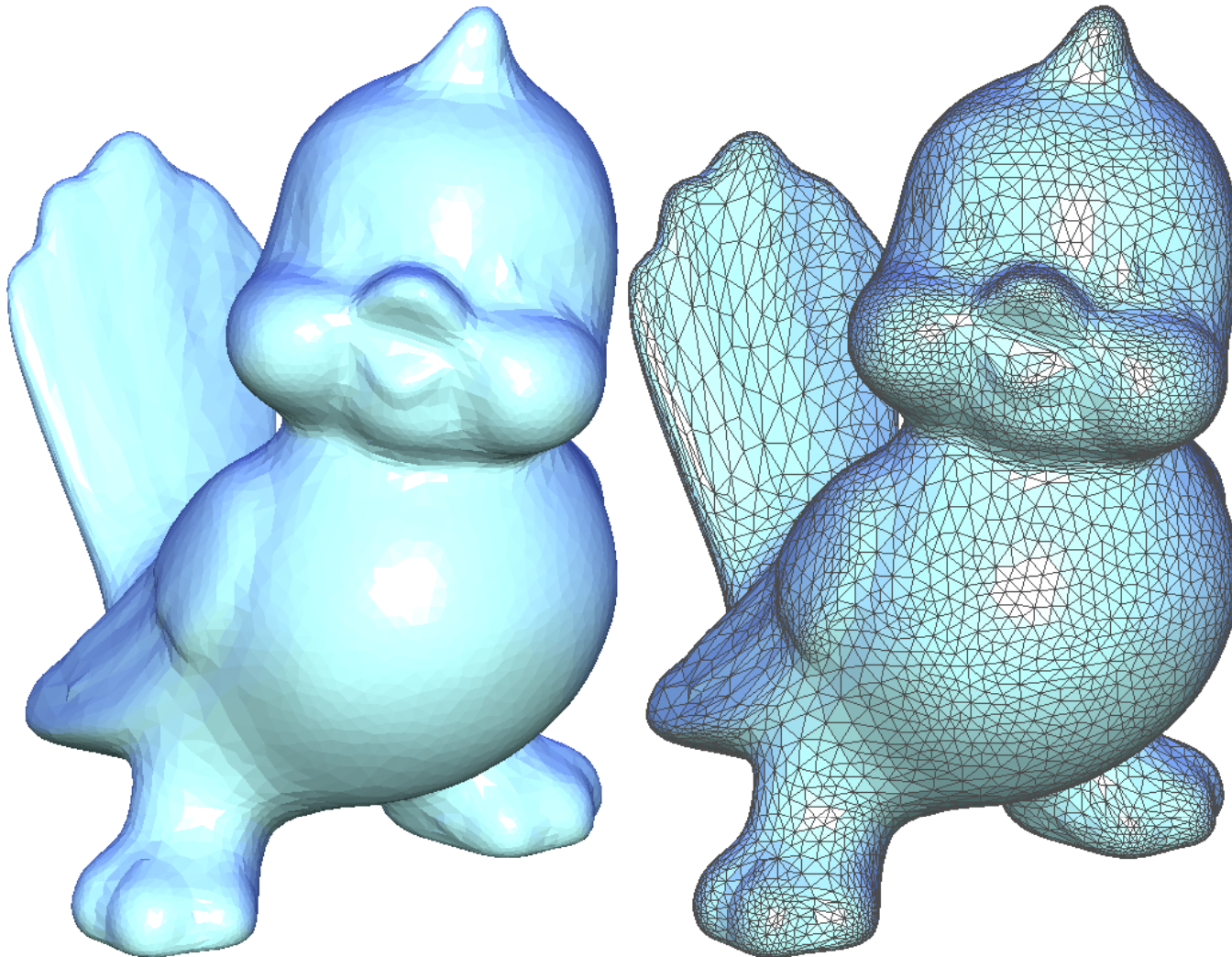
$t_{\text{min}} = 0.034$

3/11/2009

Original



Tri Shape Optimization

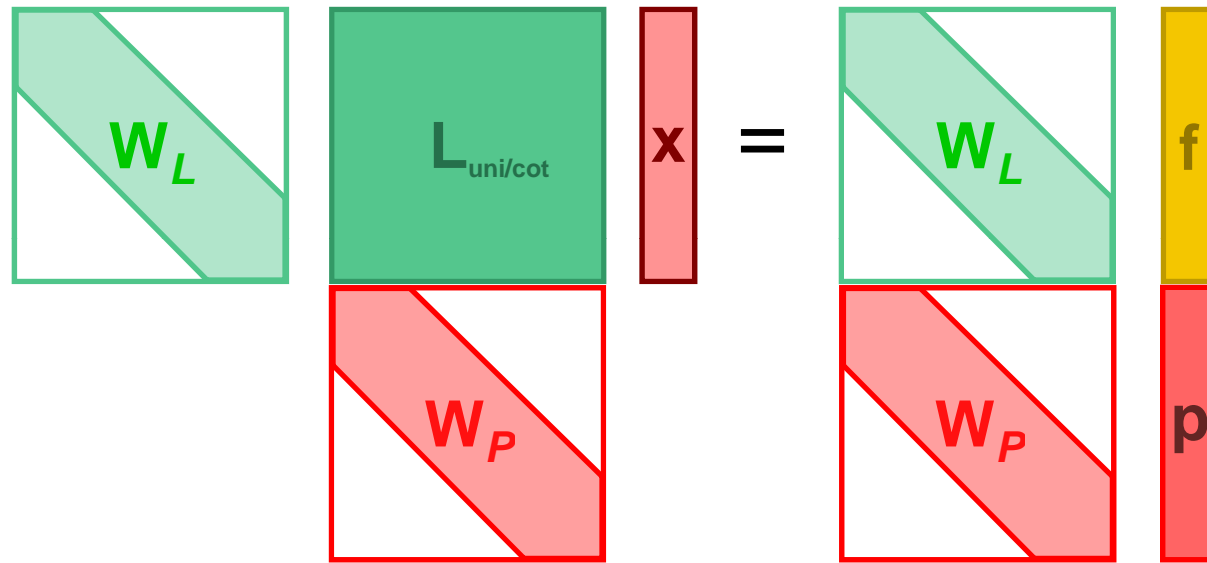


Application:

Detail preserving mesh editing

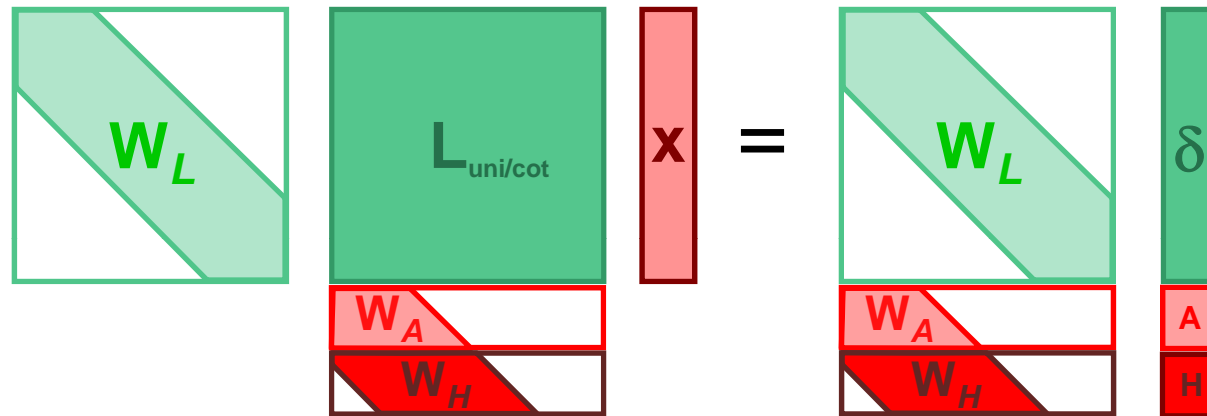
Retain local features
as much as possible

Laplacian mesh processing framework



- **Detail preserving mesh editing** for $L = L_{\text{uni or cot}}$ and $f = \delta_{\text{uni or cot}}$

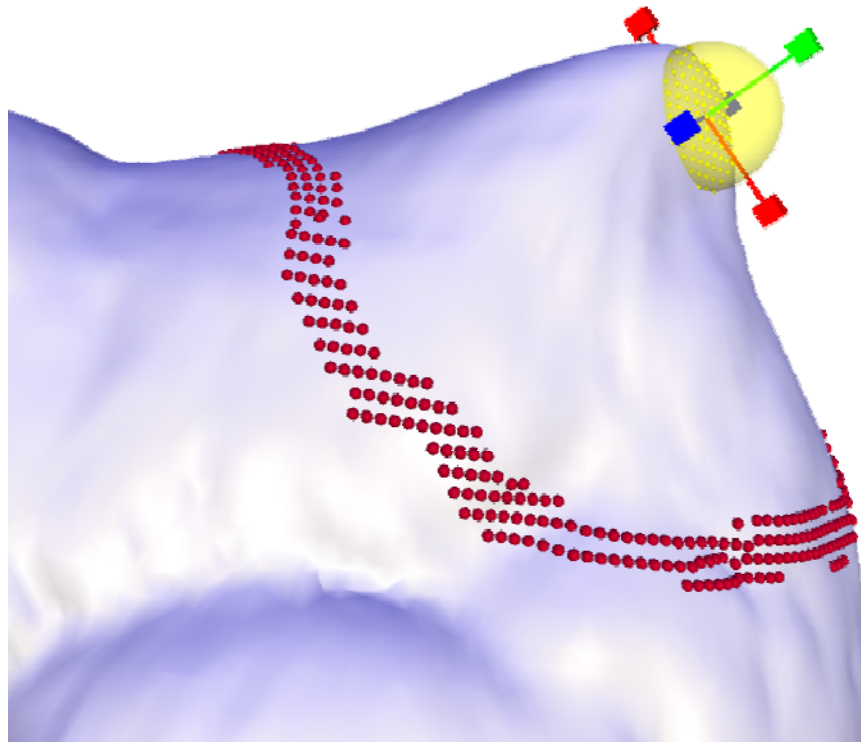
Laplacian surface editing framework



- **Detail preserving mesh editing** for $L = L_{uni \text{ or } cot}$ and $f = \delta_{uni \text{ or } cot}$
- using a **subset** of the mesh, padded by anchor vertices **A** and using vertices **H** as the deformation control handle

Laplacian surface editing framework

- Region of interest (ROI) is bounded by a belt of static anchors
- Manipulation through handle vertices



Why local Laplacian coordinates?

- Local detail representation – enables **detail preservation** through various modeling tasks
- Representation with **sparse** matrices
- Efficient **linear** surface reconstruction

