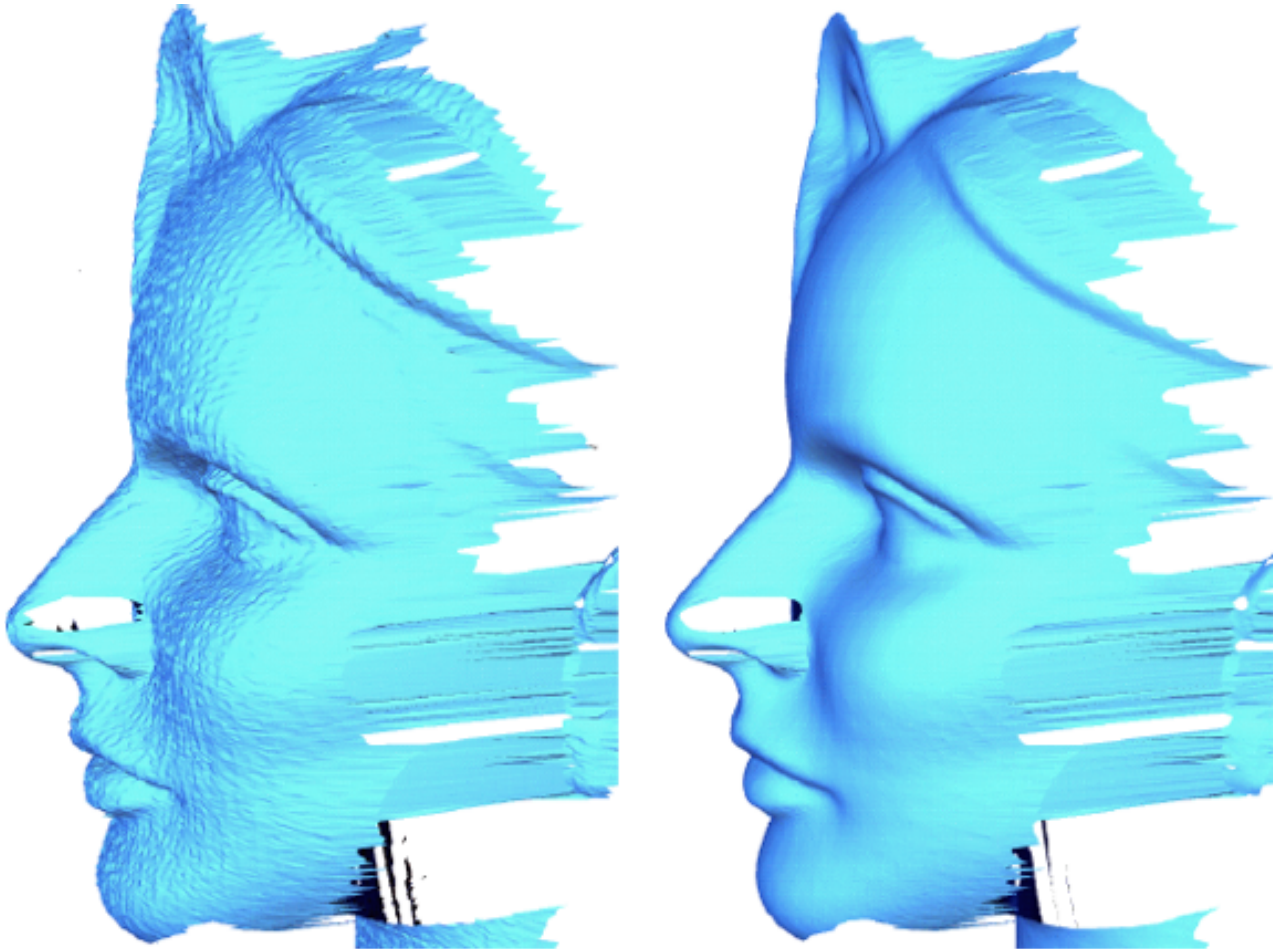
7–Smoothing

Part I

Acknowledgements: Daniele Panozzo

Surface Smoothing – Motivation

- Scanned surfaces can be noisy

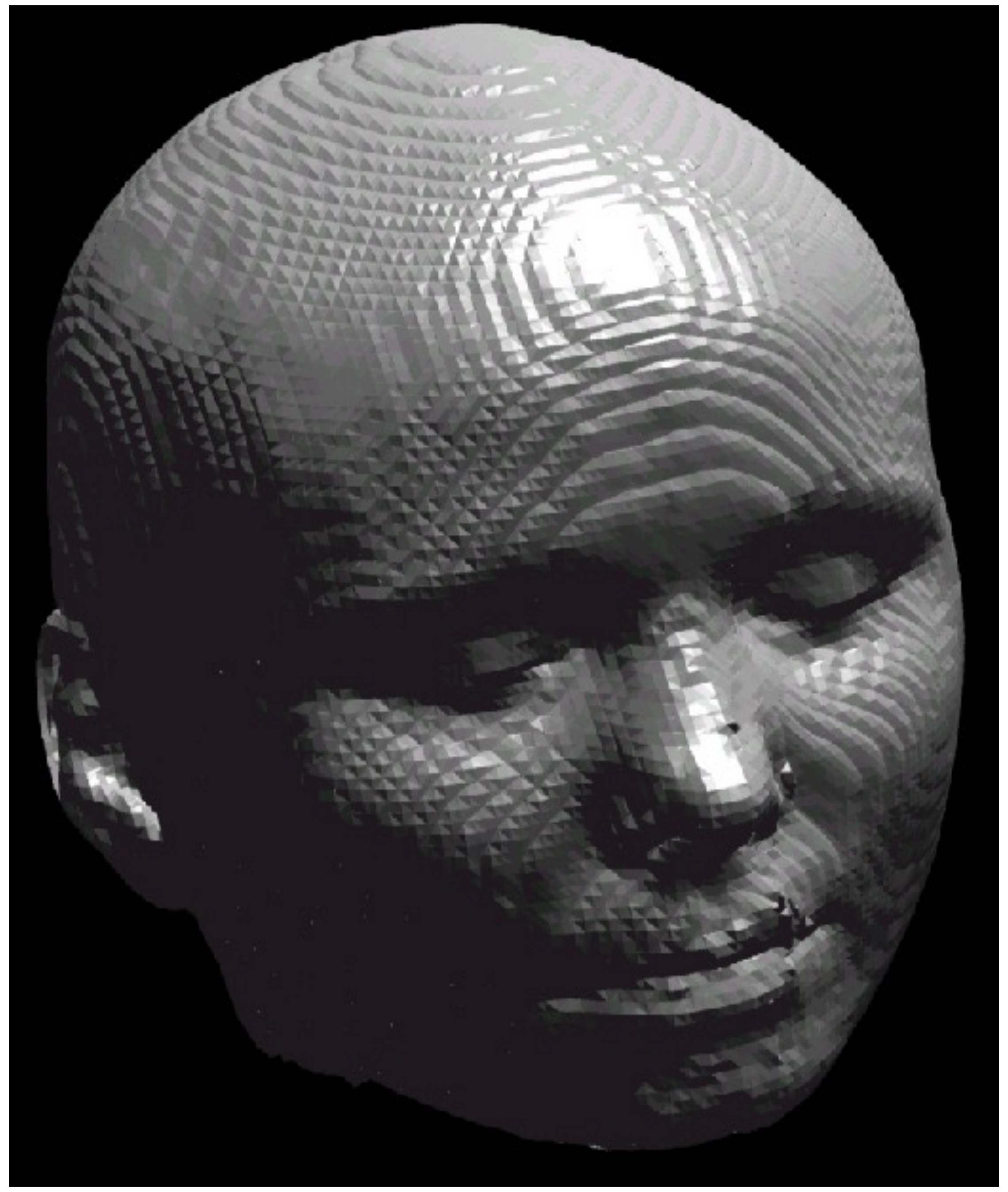


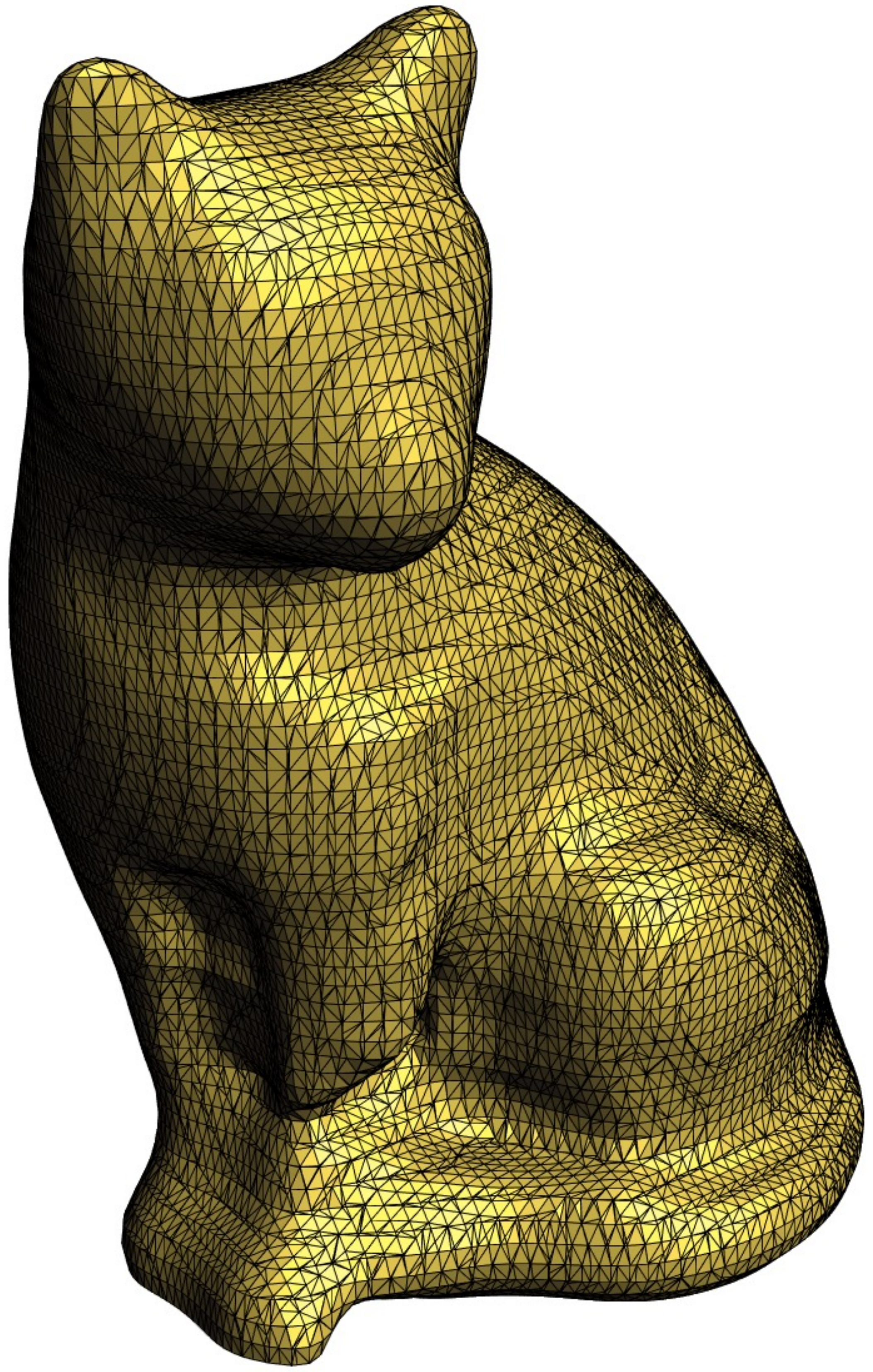
Surface Smoothing – Motivation

- Scanned surfaces can be noisy

Surface Fairing – Motivation

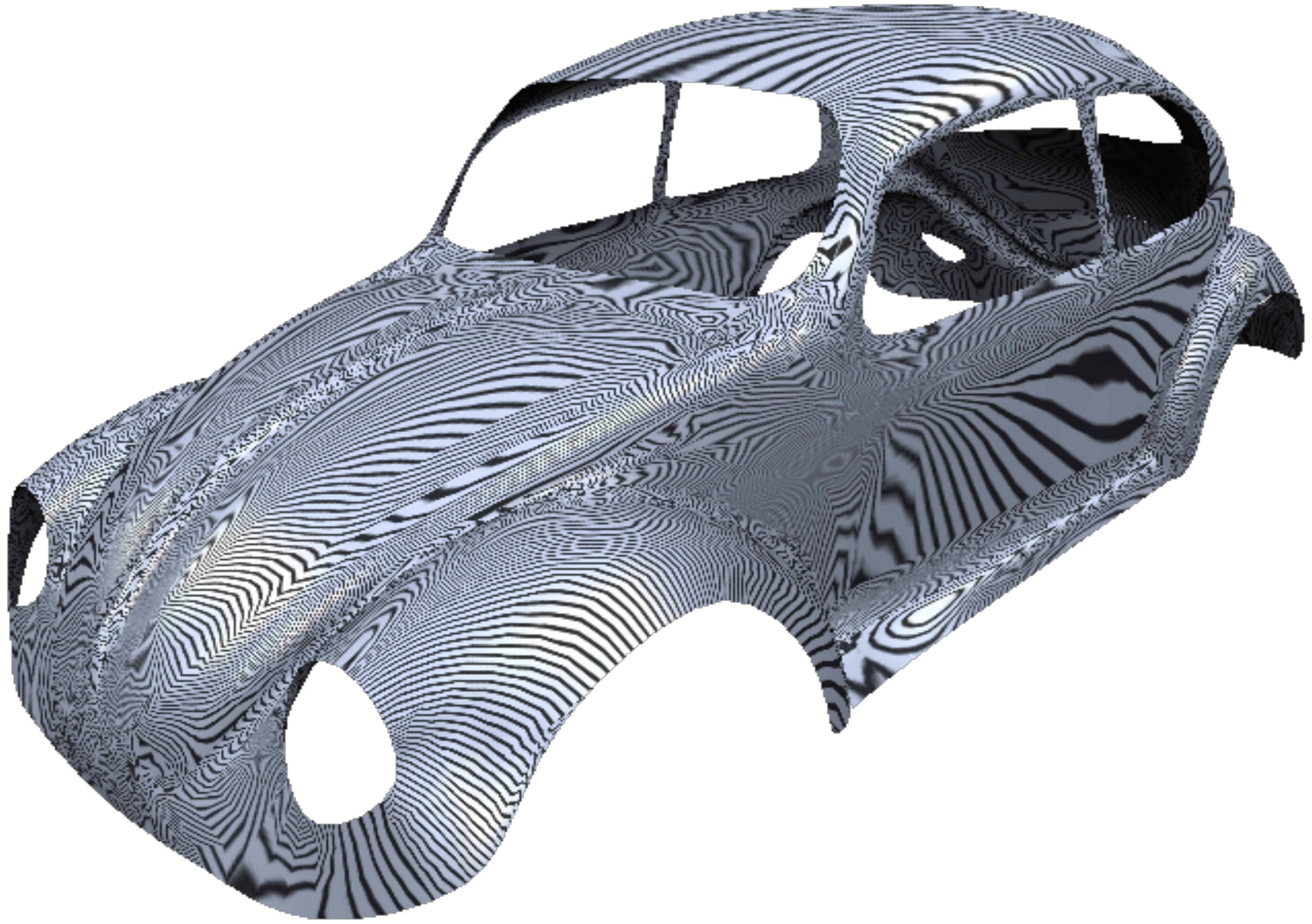
- Marching Cubes meshes can be ugly





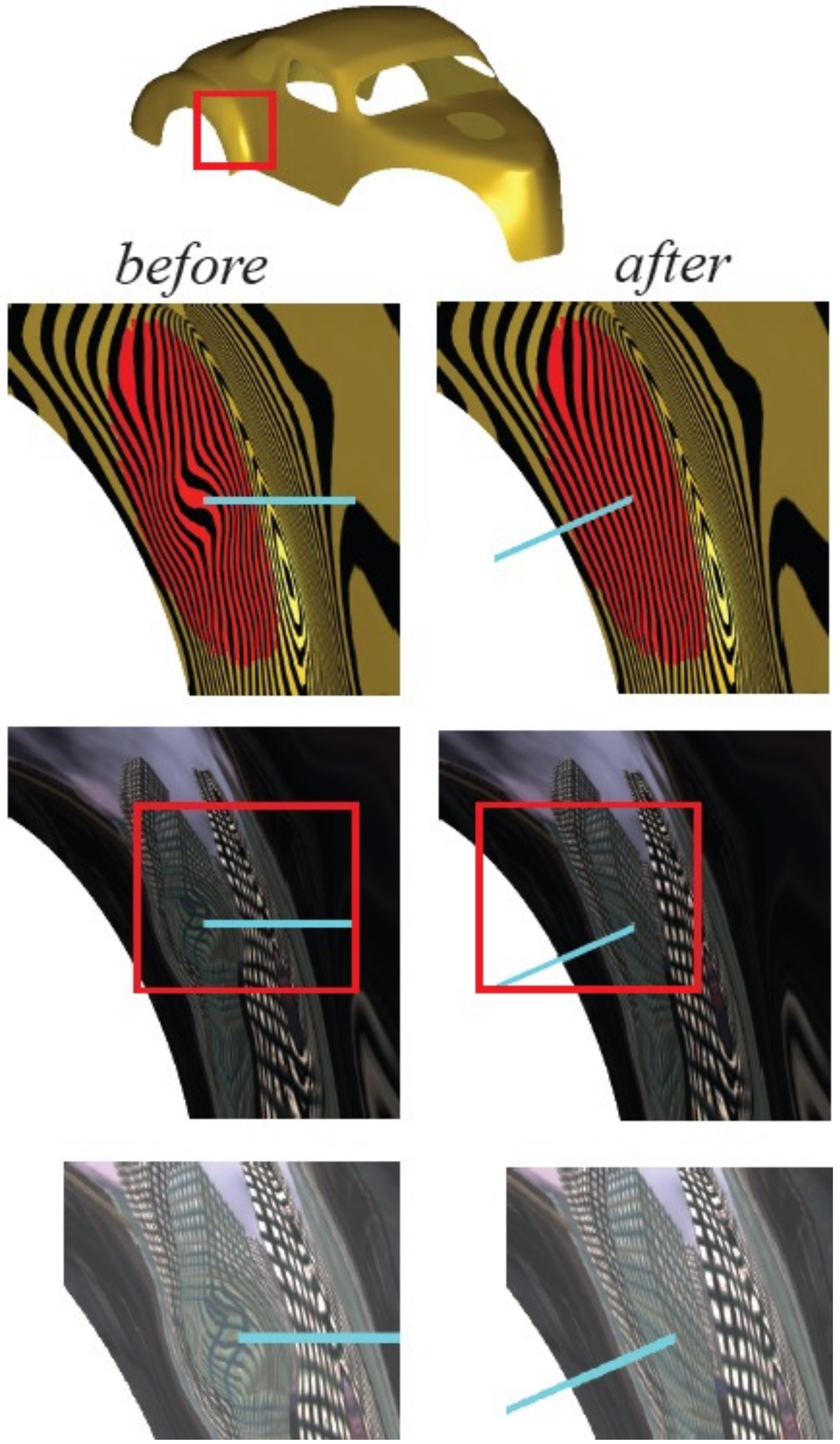
- Why is the left mesh ugly?
- Why is the right mesh ugly?
  - What is the problem with such triangles?

Visual Assessment of Surface Smoothness



Reflection Lines as an Inspection Tool

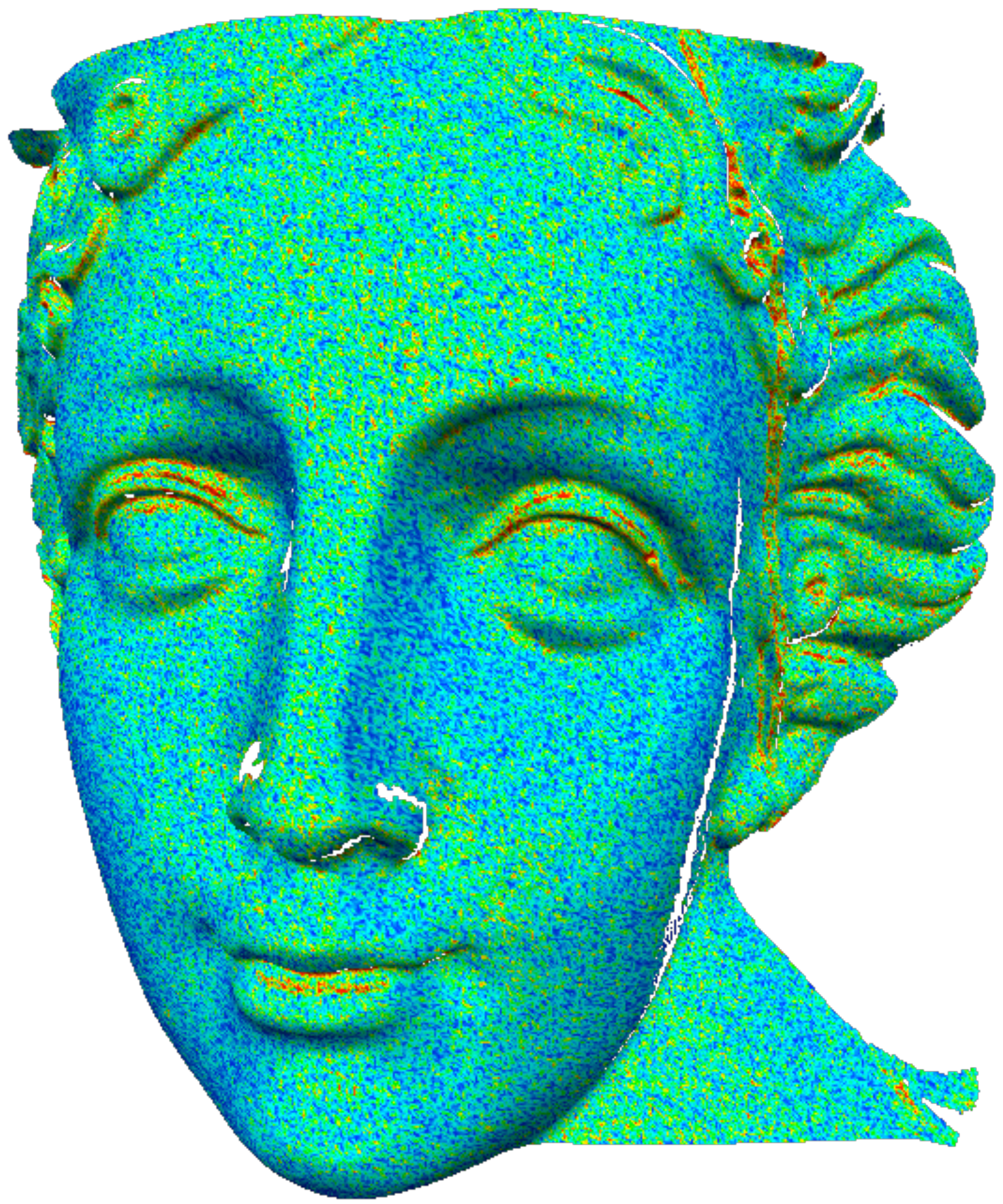
- Shape optimization using reflection lines
  E. Tosun, Y. I. Gingold, J. Reisman, D. Zorin
  Symposium on Geometry Processing 2007



How to measure smoothness?

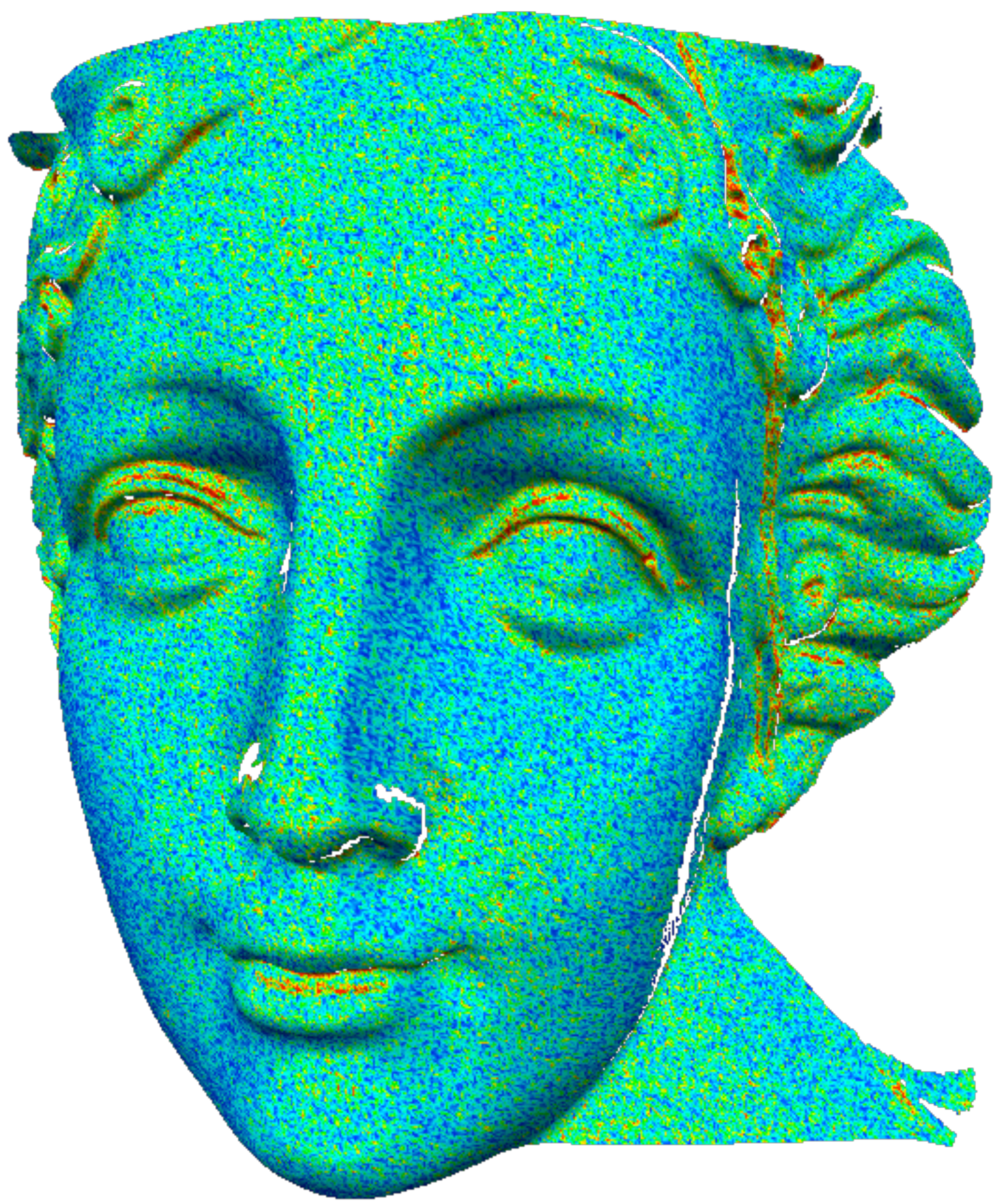
Curvature and Smoothness

Curvature and Smoothness

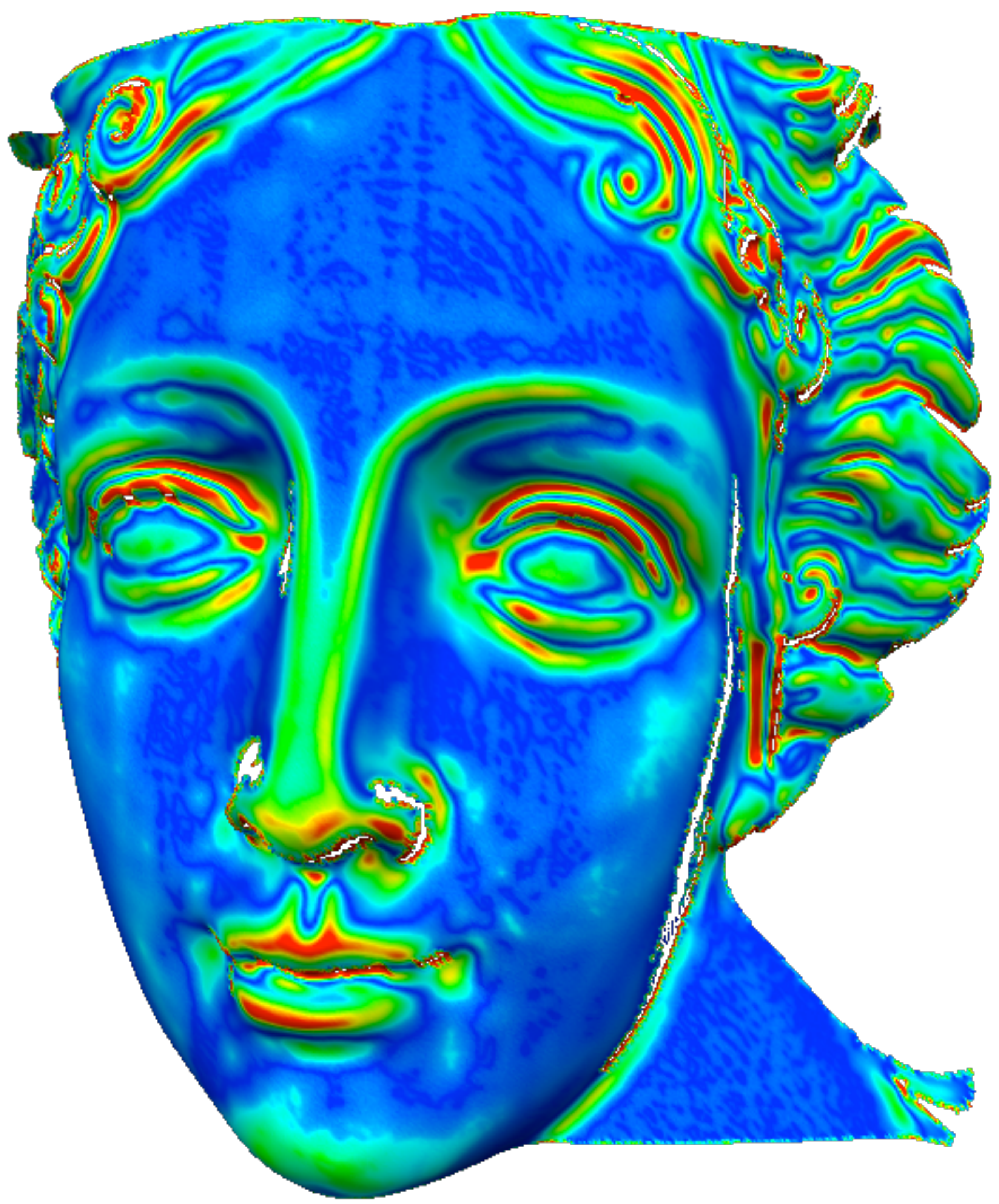


mean curvature plot

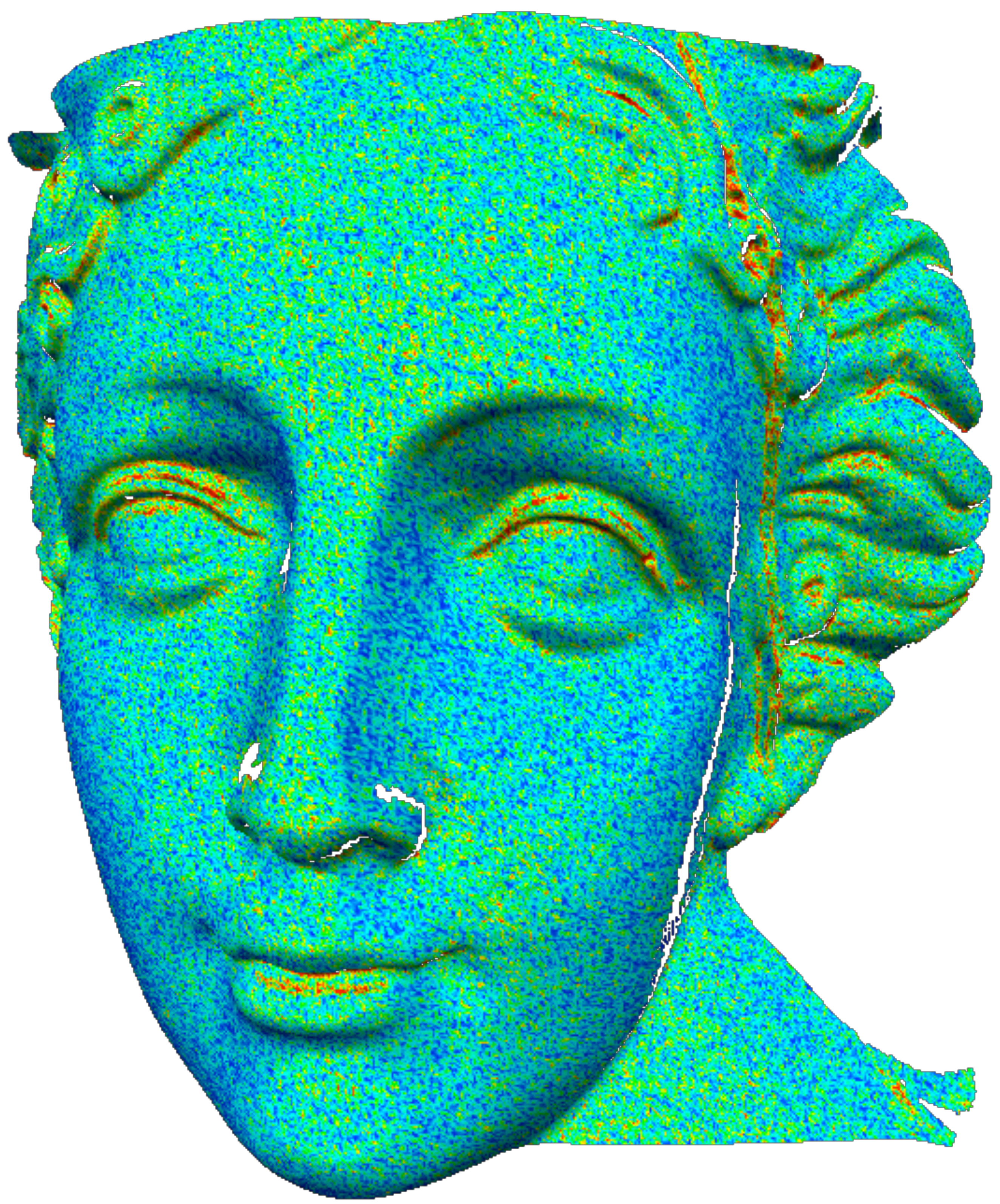
Curvature and Smoothness



mean curvature plot



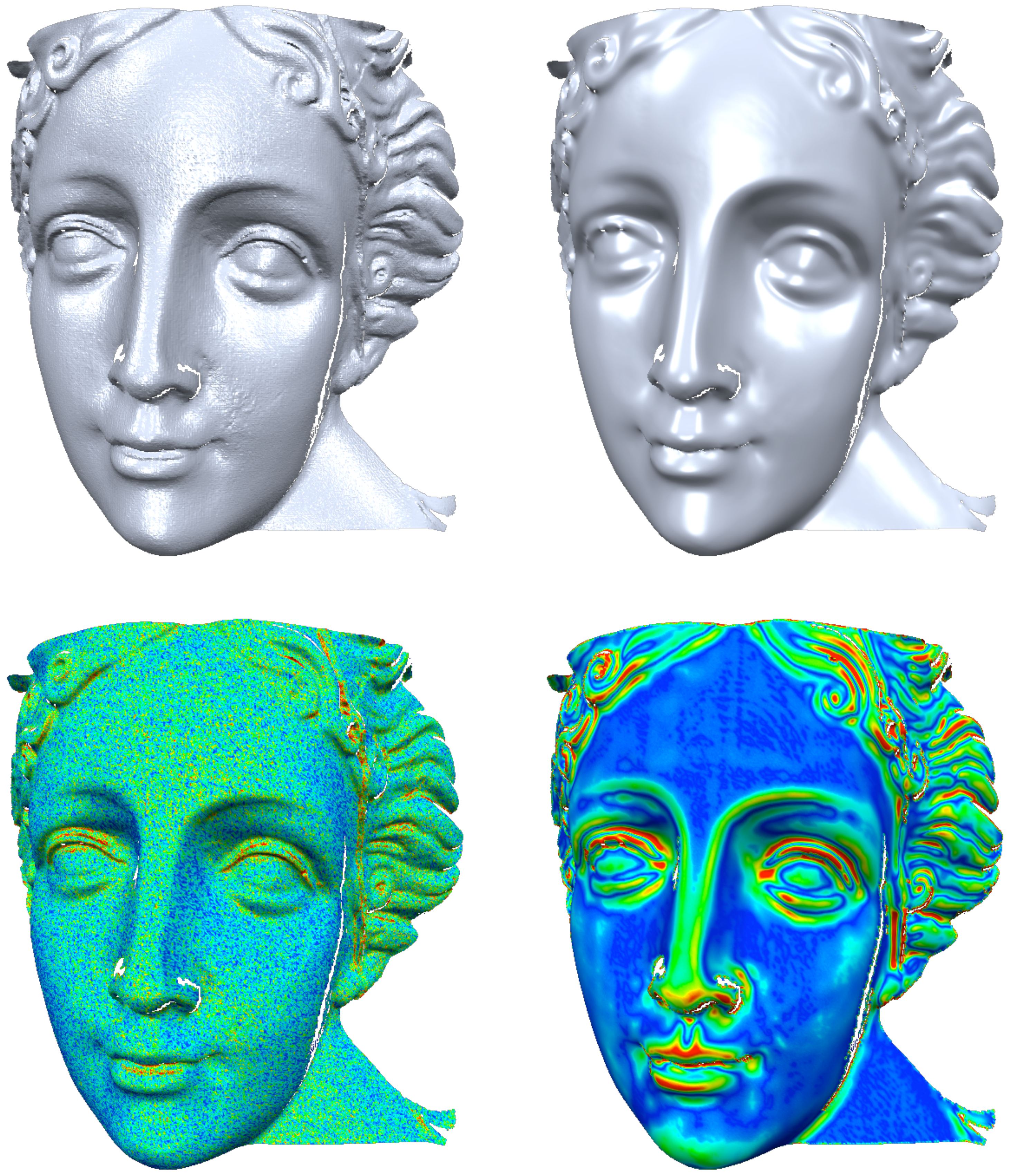
Curvature and Smoothness



mean curvature plot

Curvature and Smoothness

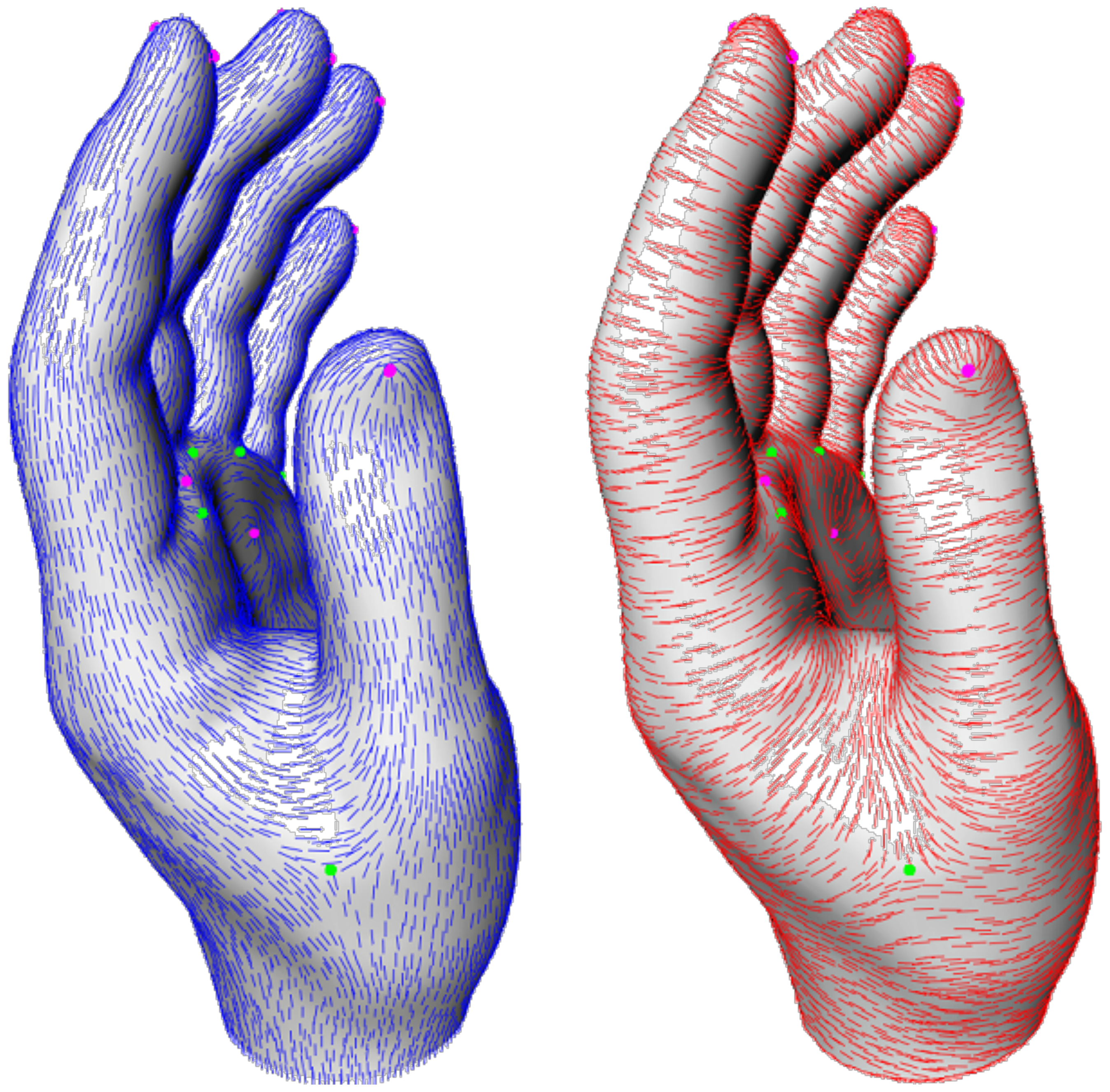
- Smoothing =
  reducing curvature?
- Smoothing =
  make curvature vary less?



Which curvature?

- Principal curvatures
  - Nonlinear and “discontinuous” operator in the definition (min, max)

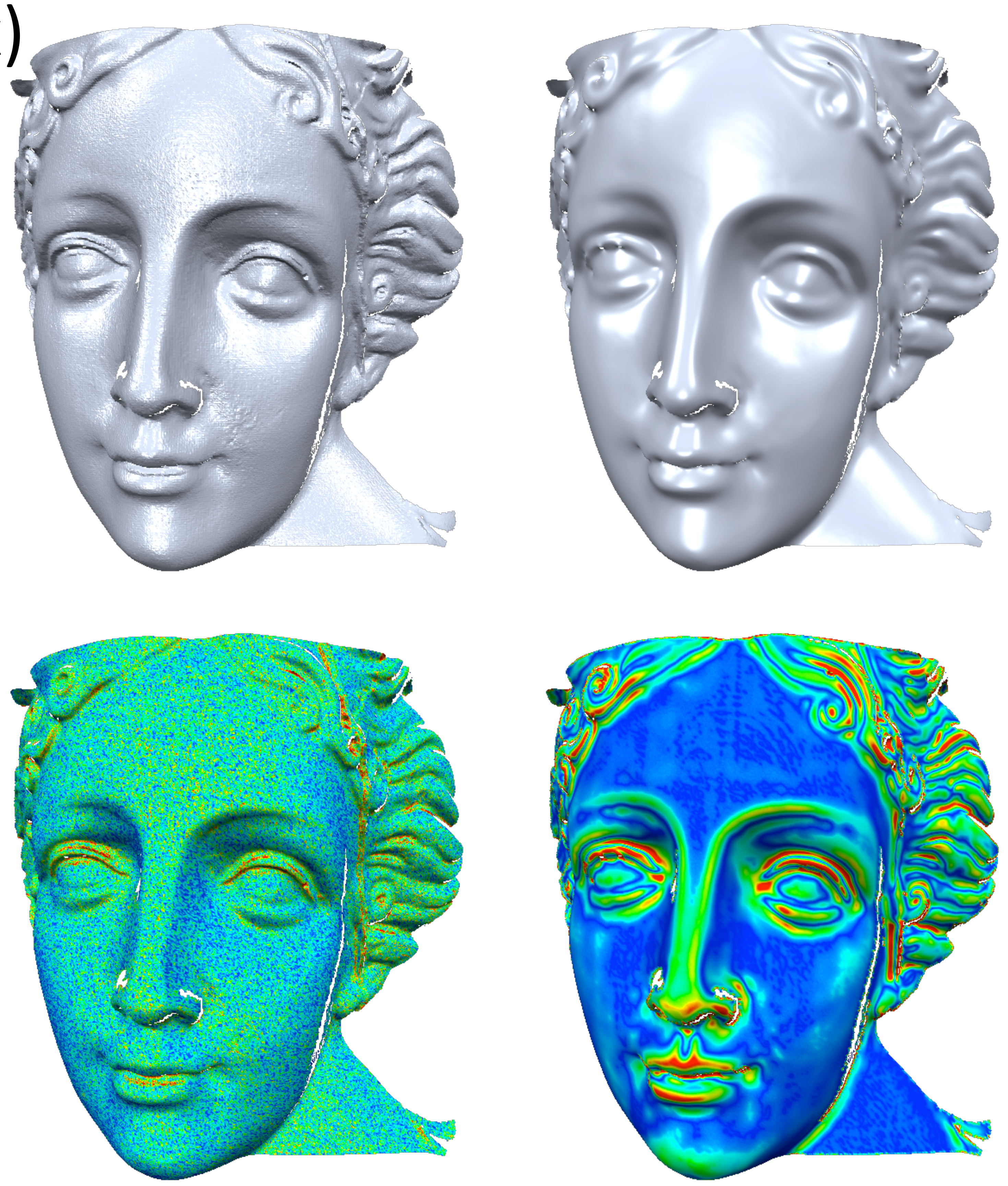
$$\kappa_{\min}, \kappa_{\max}$$



principal directions

Which curvature?

- Principal curvatures
  - Nonlinear and “discontinuous” operator in the definition (min, max)
- Mean curvature H
  - Relatively simple to extract on meshes via Laplace-Beltrami:



$$\Delta_{\mathcal{M}} \mathbf{p} = -2H\mathbf{n}$$

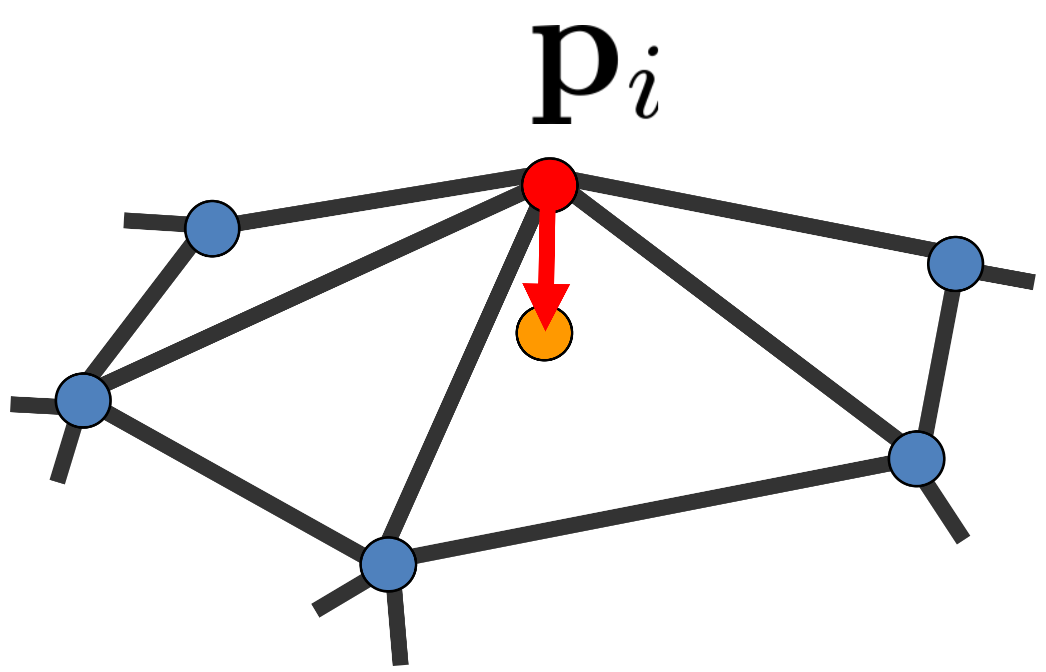
goal: $H = 0$ or $H = \text{const}$

Laplace as linear operator

$$\boxed{\Delta_{\mathcal{M}}\mathbf{p} = -2H\mathbf{n}}$$

Recap: Laplace-Beltrami

- High-pass filter: extracts local surface detail
  - Detail = *smooth*(surface) – surface
  - Assumption: smoothing = averaging

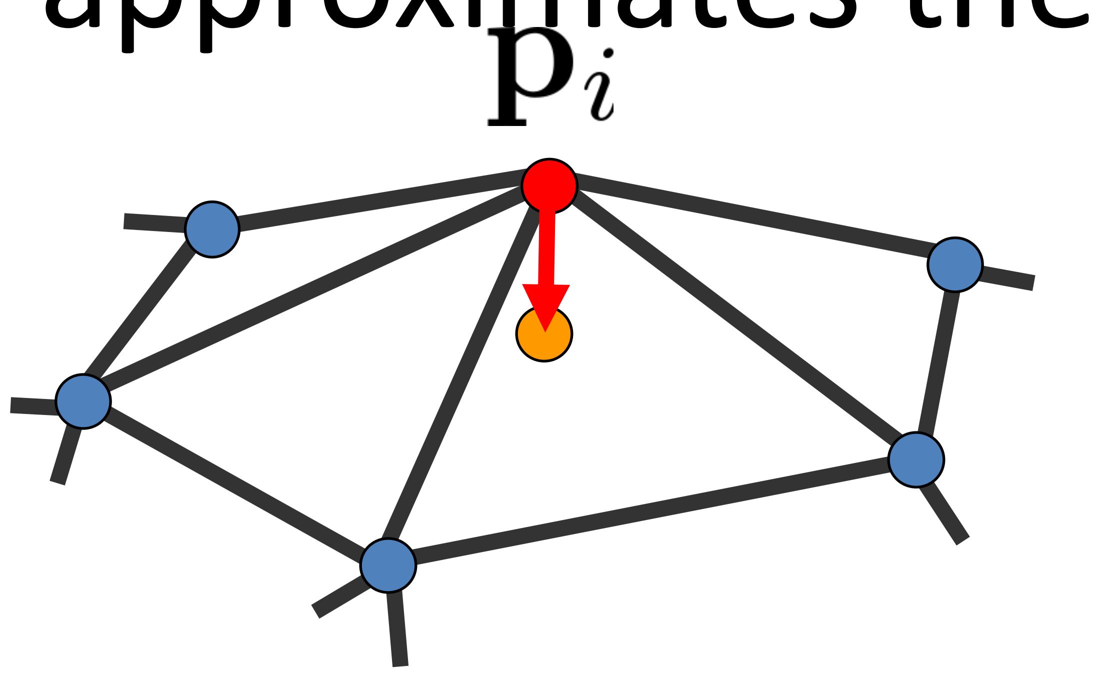


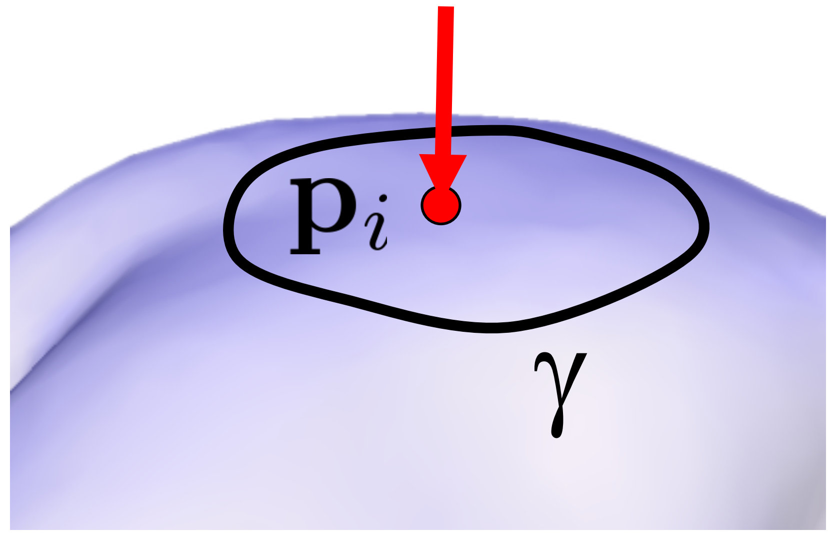
$$\Delta_{\mathcal{M}}(\mathbf{p}_i) = \delta_i = \frac{1}{W_i} \sum_{j \in \mathcal{N}(i)} w_{ij}(\mathbf{p}_j - \mathbf{p}_i)$$

$$\boxed{\Delta_{\mathcal{M}}\mathbf{p} = -2H\mathbf{n}}$$

Recap: Laplace-Beltrami

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





$$\delta_i = \frac{1}{W_i} \sum_{j \in \mathcal{N}(i)} w_{ij} (\mathbf{p}_j - \mathbf{p}_i) \qquad \delta_i = \frac{1}{\mathrm{len}(\gamma)} \int_\gamma (\gamma(s) - \mathbf{p}_i)\, ds$$

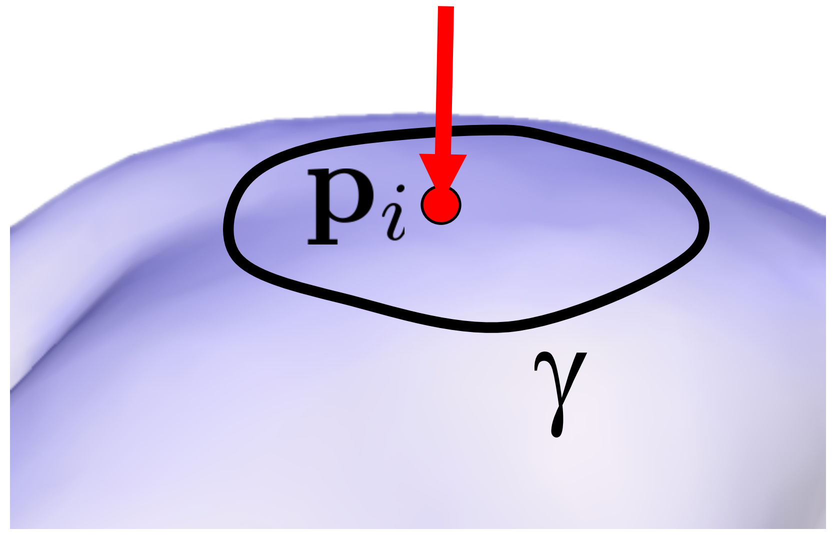
$$\lim_{\mathrm{len}(\gamma) \to 0} \frac{1}{\mathrm{len}(\gamma)} \int_\gamma (\gamma(s) - \mathbf{p}_i)\, ds = -2H(\mathbf{p}_i)\, \mathbf{n}_i$$

L-B: Weighting Schemes

$$\delta_i = \frac{1}{W_i} \sum_{j \in \mathcal{N}(i)} w_{ij}(\mathbf{p}_j - \mathbf{p}_i)$$

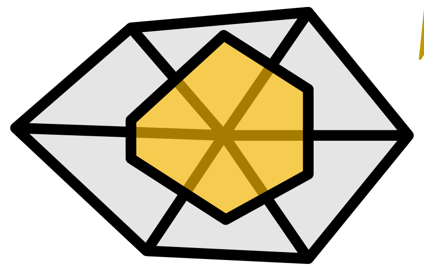
- Ignore geometry

- Integrate over Voronoi region of the vertex

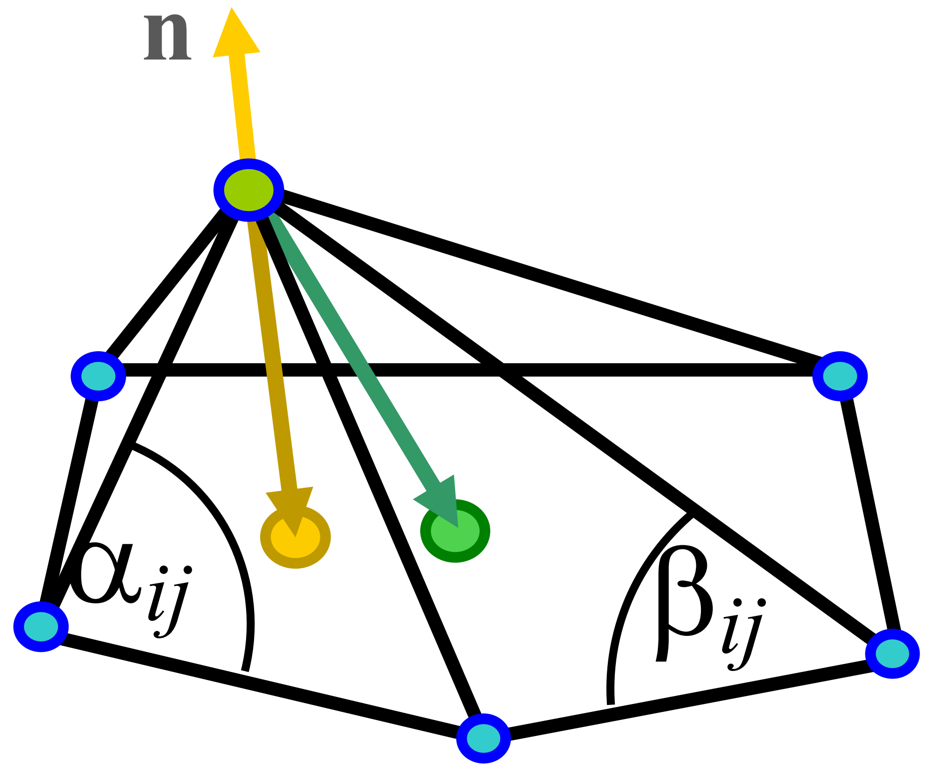


δ_{uniform} : $W_i = 1,\ w_{ij} = 1/|N(i)|$

δ_{cotan} : $w_{ij} = 0.5(\cot \alpha_{ij} + \cot \beta_{ij})$

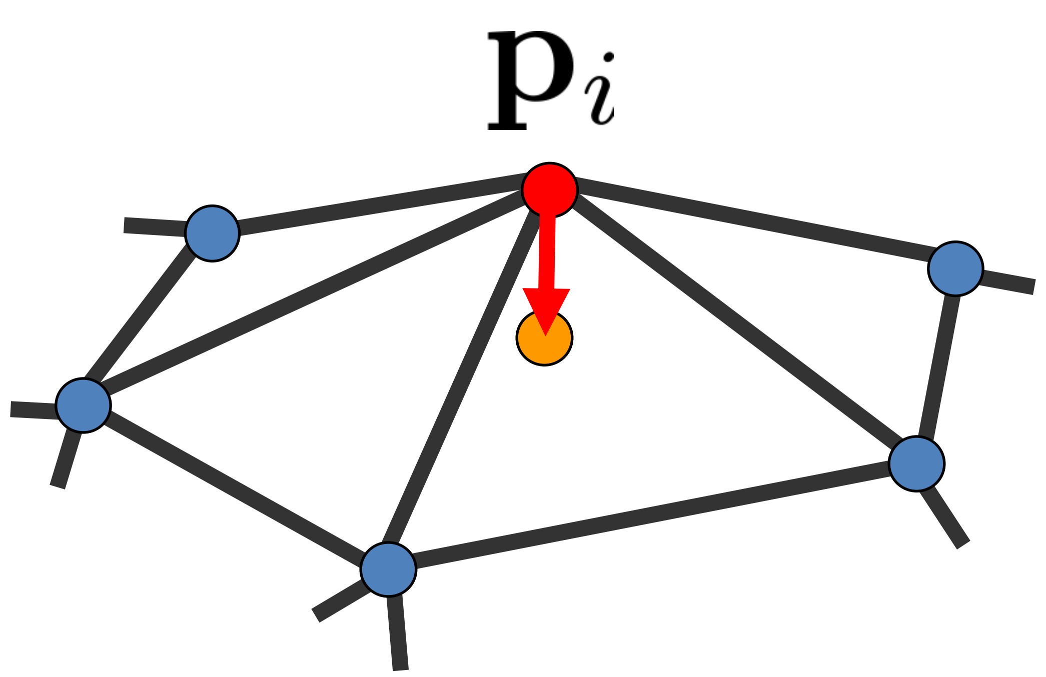


$W_i = A_i$



Laplacian Matrix

- The transition between *xyz* and δ is linear:



$$\delta_i = \frac{1}{W_i} \sum_{j \in \mathcal{N}(i)} w_{ij} (\mathbf{p}_j - \mathbf{p}_i)$$

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

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

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

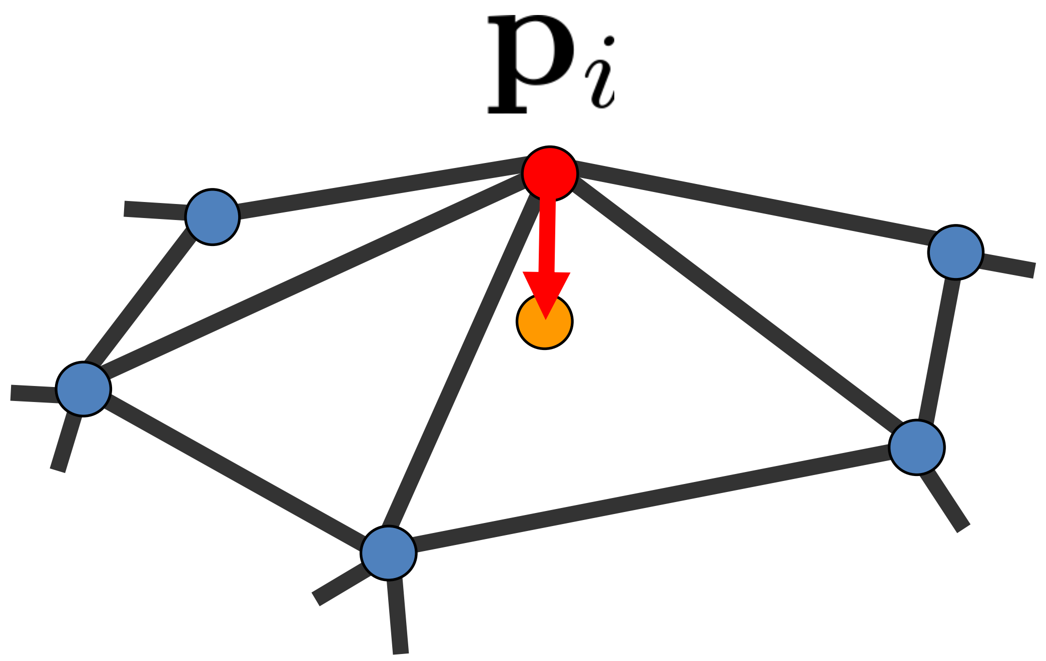
$$\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{R}^n$$

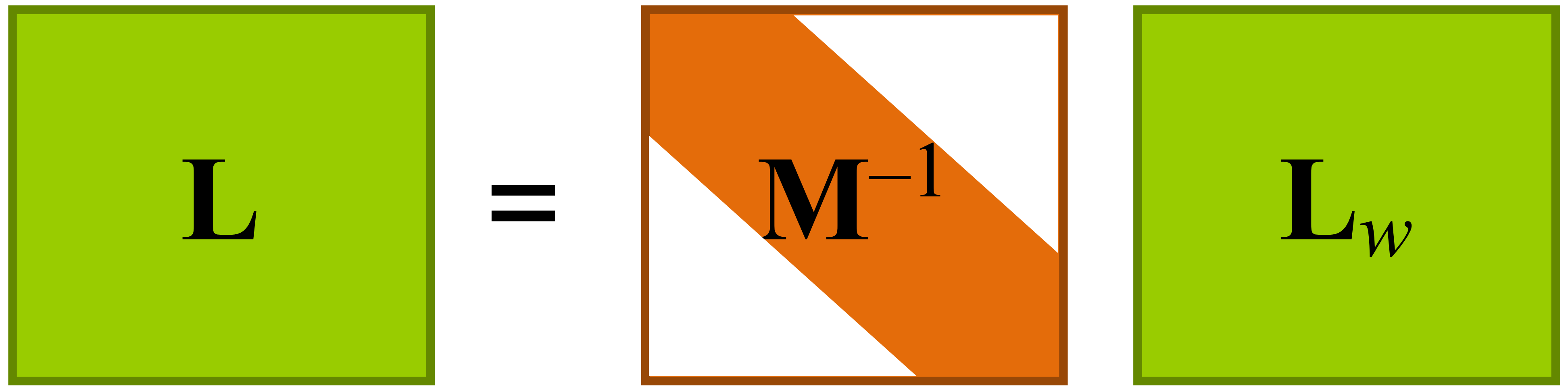
$$\mathbf{L} \in \mathbb{R}^{n \times n}$$

$$\delta_{\mathbf{x}}, \delta_{\mathbf{y}}, \delta_{\mathbf{z}} \in \mathbb{R}^n$$

Laplacian Matrix

- Breaking down the Laplace matrix:
- $\mathbf{M}$ = mass matrix; $\mathbf{L}_w$ = stiffness matrix





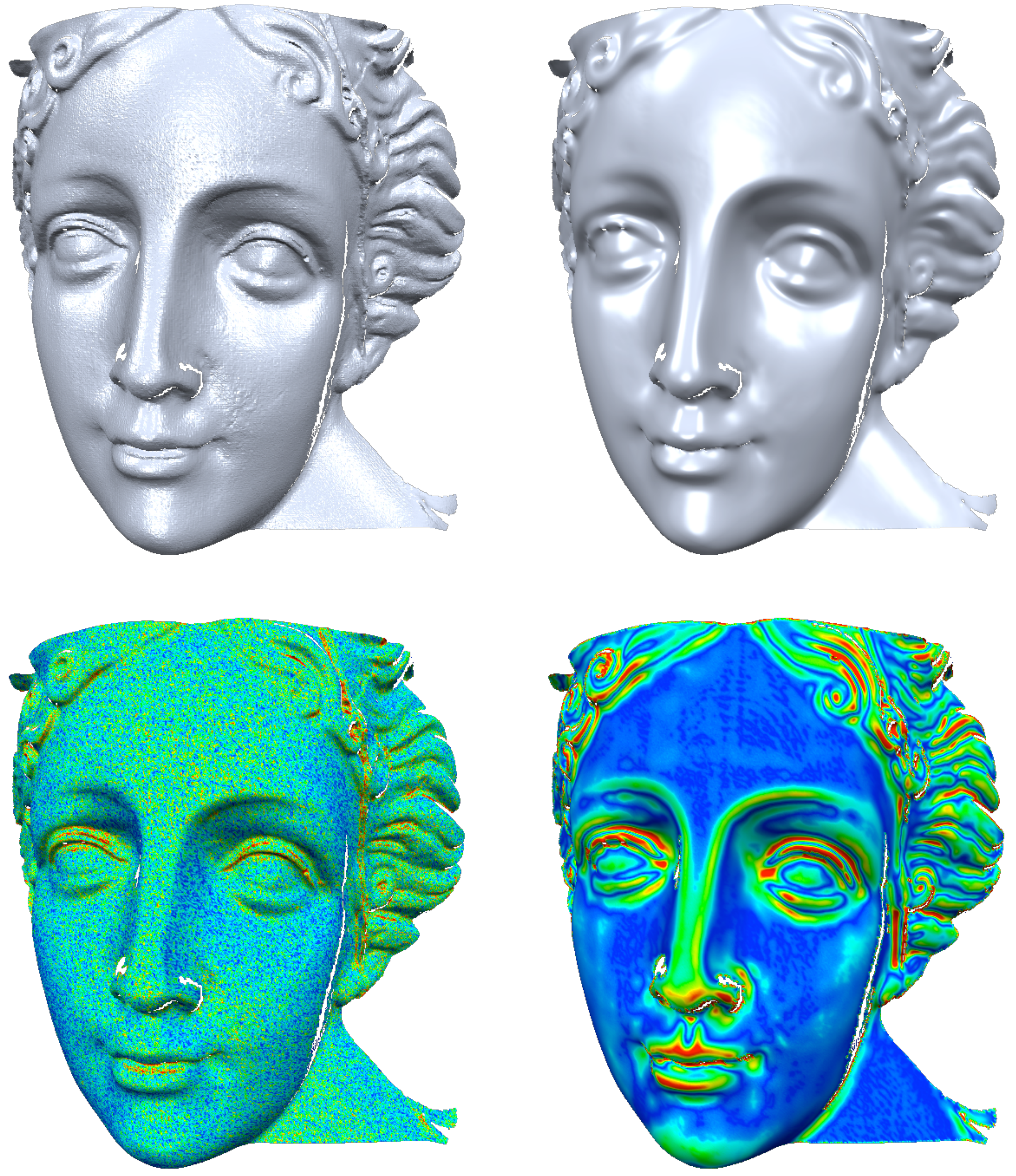
$$\delta_i = \frac{1}{W_i} \sum_{j \in \mathcal{N}(i)} w_{ij} (\mathbf{p}_j - \mathbf{p}_i)$$

How to do the smoothing?

$$\Delta_{\mathcal{M}} \mathbf{p} = -2H\mathbf{n}$$

goal: $H = 0$ or $H = \text{const}$

- Smooth H, obtain $\tilde{H}$
- Find a surface that has $\tilde{H}$ as mean curvature
  - H doesn't define the surface
  - $\mathbf{n}$ nonlinear in $\mathbf{p}$
- Another idea:

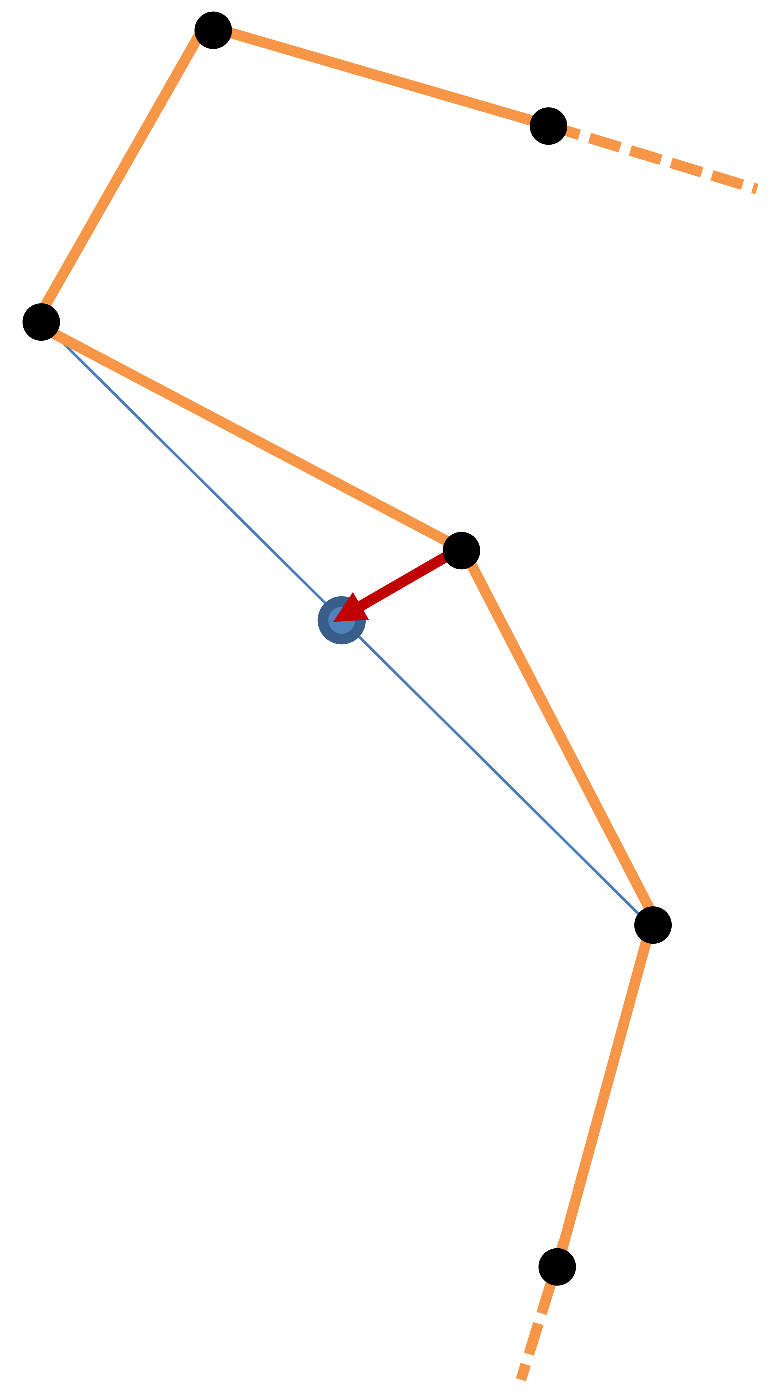


Smoothing by flowing

Example – smoothing curves

- Laplace in 1D = second derivative:

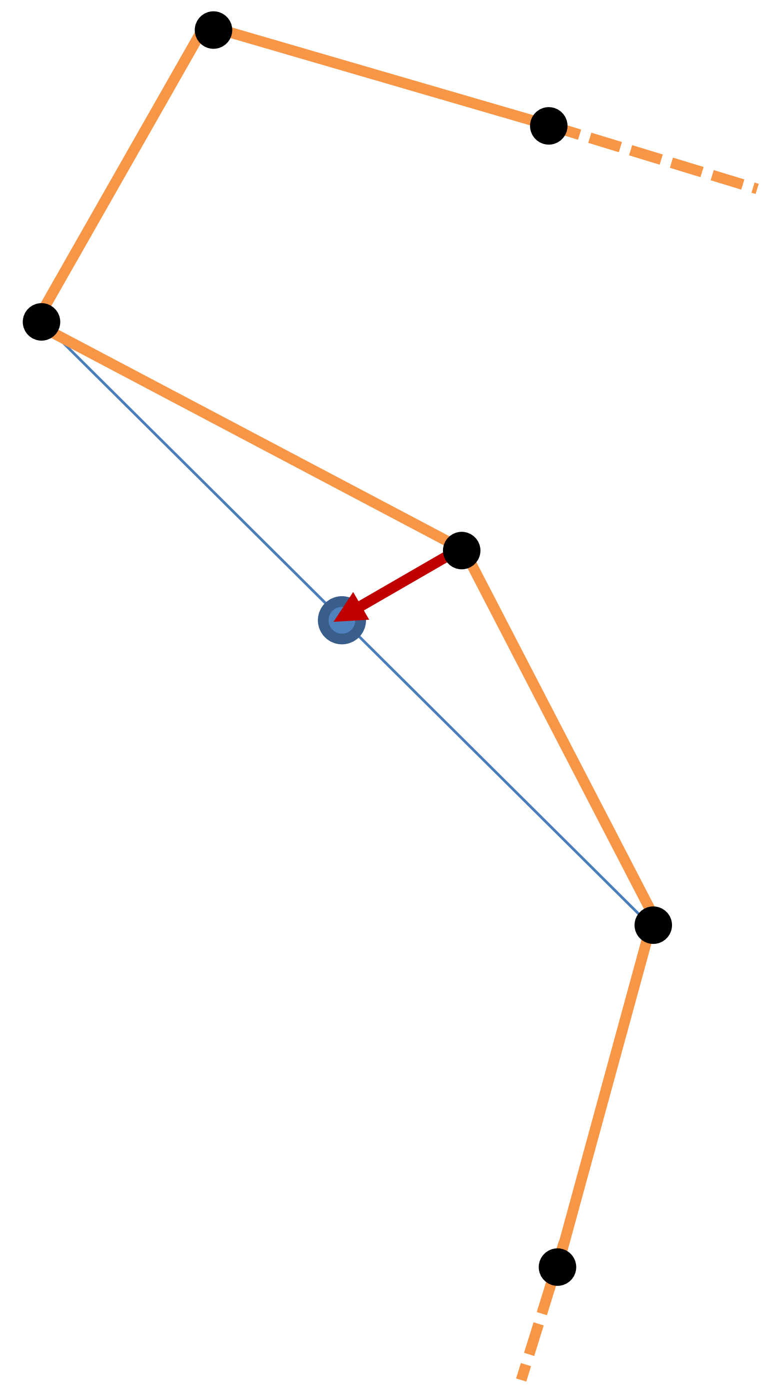
$$L(\mathbf{p}_i) = \frac{1}{2}(\mathbf{p}_{i-1} - \mathbf{p}_i) + \frac{1}{2}(\mathbf{p}_{i+1} - \mathbf{p}_i)$$



Example – smoothing curves

- Laplace in 1D = second derivative:

$$L(\mathbf{p}_i) = \frac{1}{2}(\mathbf{p}_{i-1} - \mathbf{p}_i) + \frac{1}{2}(\mathbf{p}_{i+1} - \mathbf{p}_i)$$



- In matrix-vector form for the whole curve

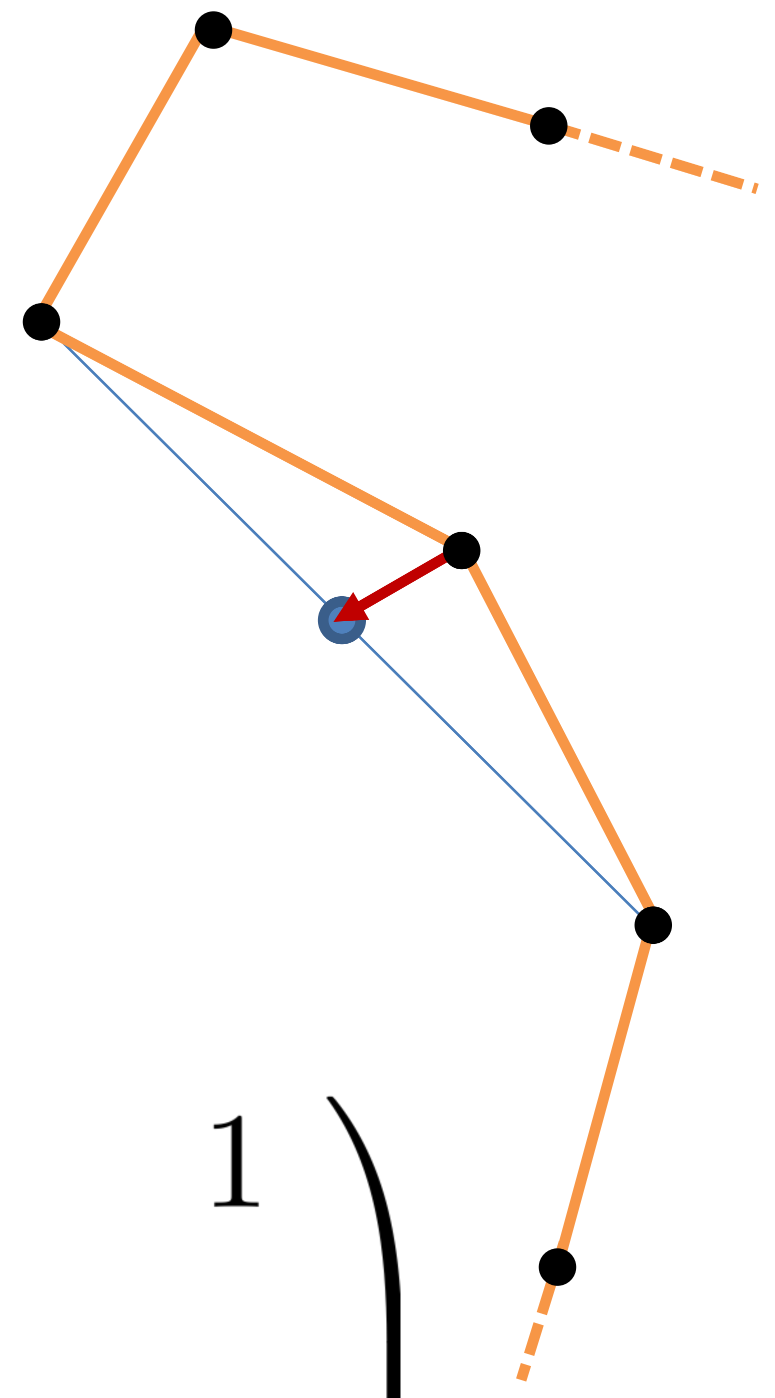
$$L\mathbf{p}$$

$$\mathbf{p} = [\mathbf{x}\ \mathbf{y}] \in \mathbb{R}^{n\times 2}$$

Example – smoothing curves

- Laplace in 1D = second derivative:

$$L(\mathbf{p}_i) = \frac{1}{2}(\mathbf{p}_{i-1} - \mathbf{p}_i) + \frac{1}{2}(\mathbf{p}_{i+1} - \mathbf{p}_i)$$

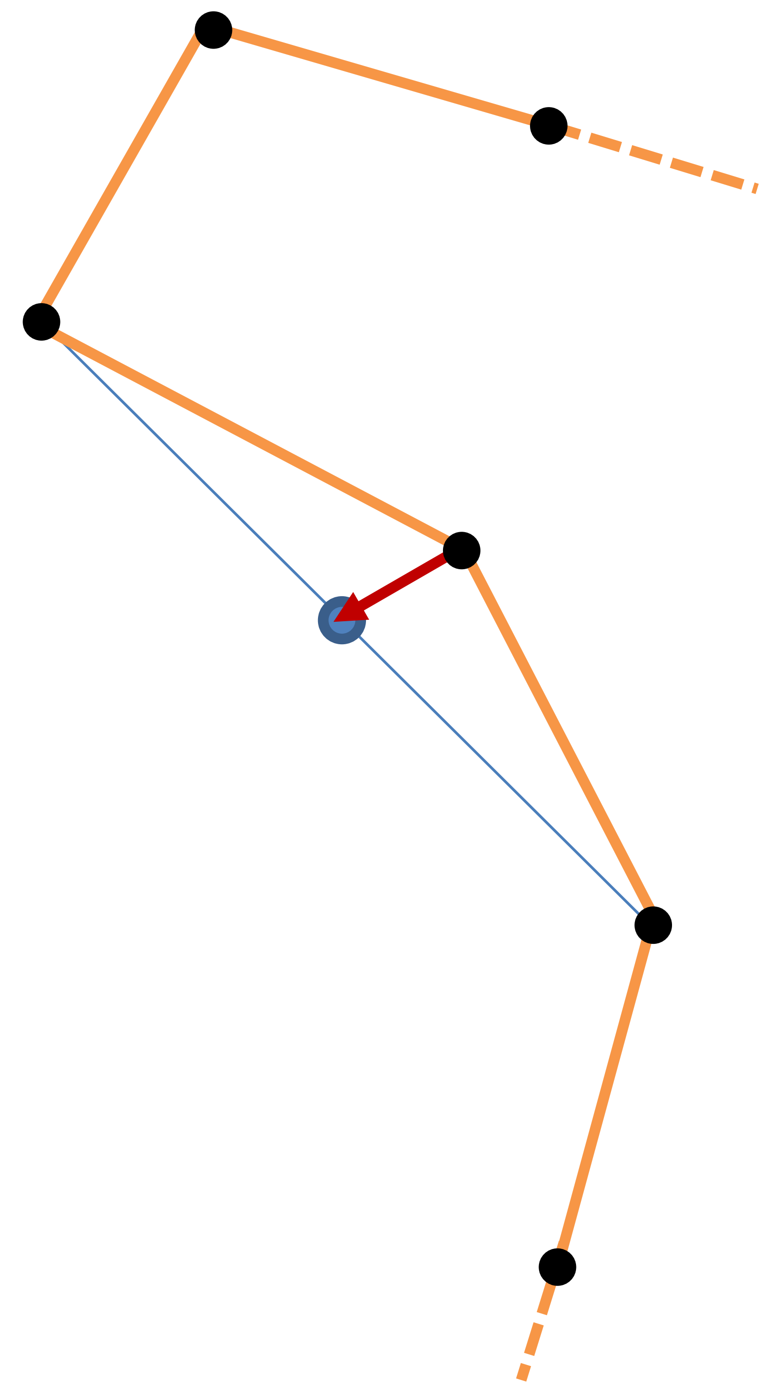


- In matrix-vector form for the whole curve

$$L\mathbf{p}$$

$$\mathbf{p} = [\mathbf{x}\ \mathbf{y}] \in \mathbb{R}^{n\times 2} \quad L = \frac{1}{2}\begin{pmatrix} -2 & 1 & & & 1 \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ 1 & & & 1 & -2 \end{pmatrix}$$

Example – smoothing curves



- Flow to reduce curvature:

$$\tilde{\mathbf{p}}_i = \mathbf{p}_i + \lambda \frac{d^2}{ds^2}(\mathbf{p}_i)$$

- Scale factor $0 < \lambda < 1$
- Matrix-vector form:

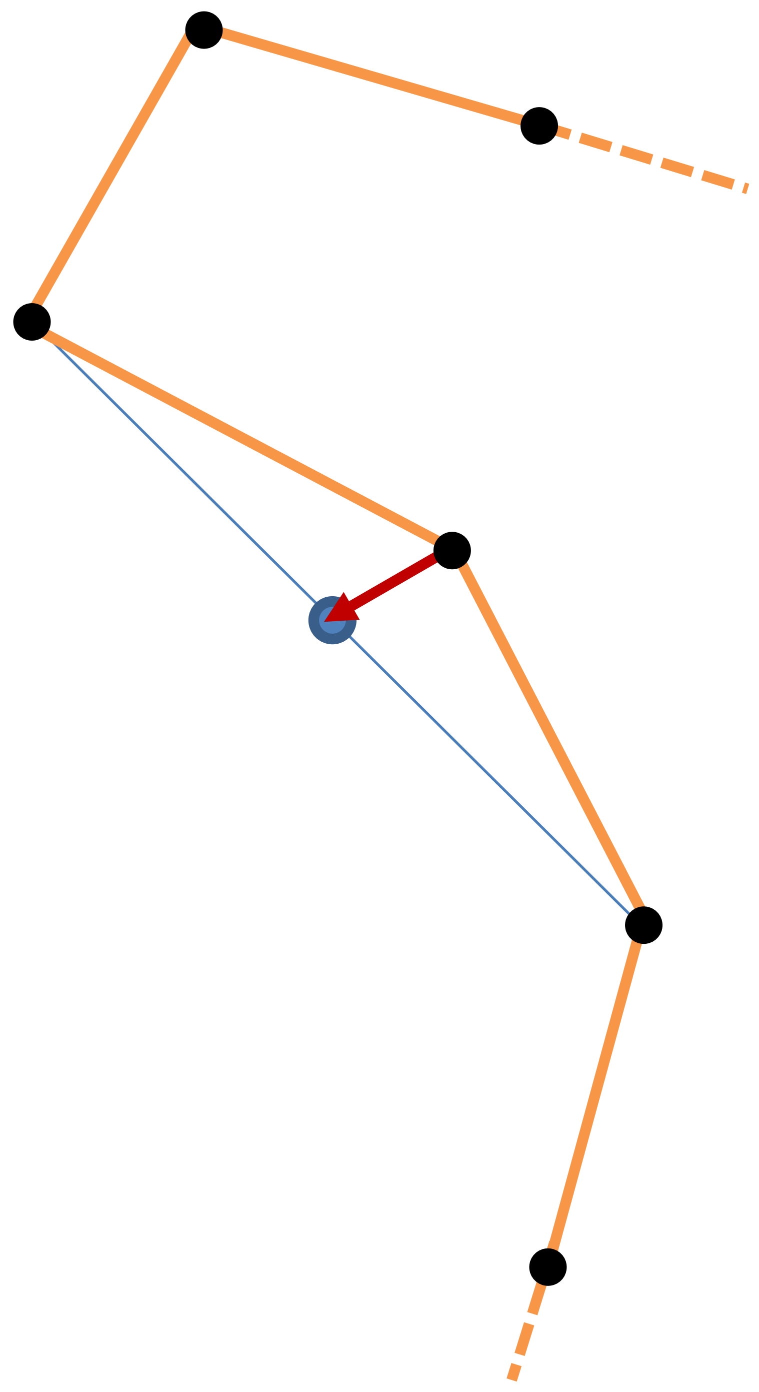
$$\tilde{\mathbf{p}} = \mathbf{p} + \lambda L \mathbf{p}, \quad \mathbf{p} \in \mathbb{R}^{n \times 2}$$

- Drawbacks?

Example – smoothing curves

- Flow to reduce curvature:

$$\tilde{\mathbf{p}}_i = \mathbf{p}_i + \lambda \frac{d^2}{ds^2}(\mathbf{p}_i)$$

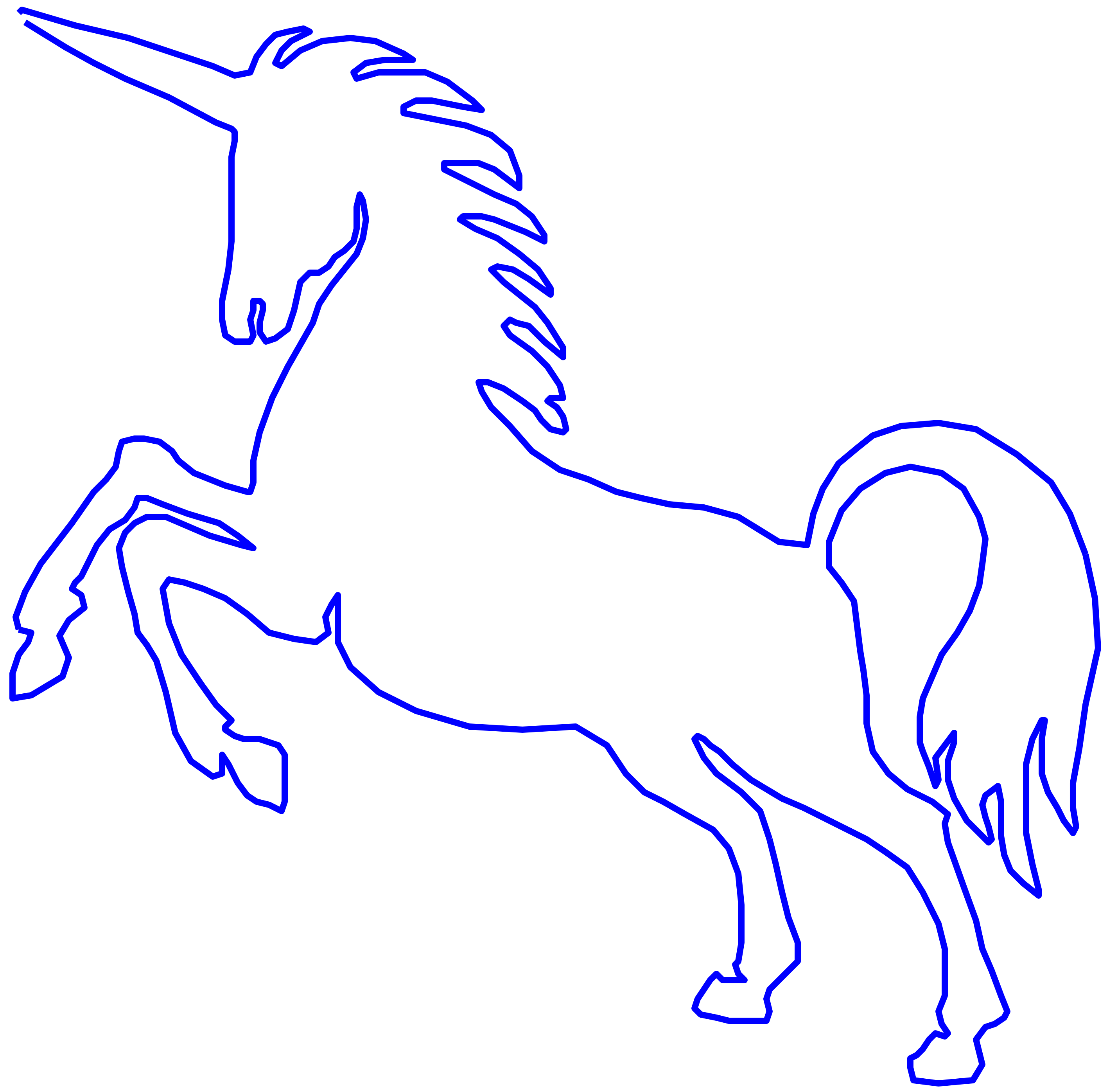


- Scale factor $0 < \lambda < 1$
- Matrix-vector form:

$$\tilde{\mathbf{p}} = \mathbf{p} + \lambda L \mathbf{p}, \quad \mathbf{p} \in \mathbb{R}^{n \times 2}$$

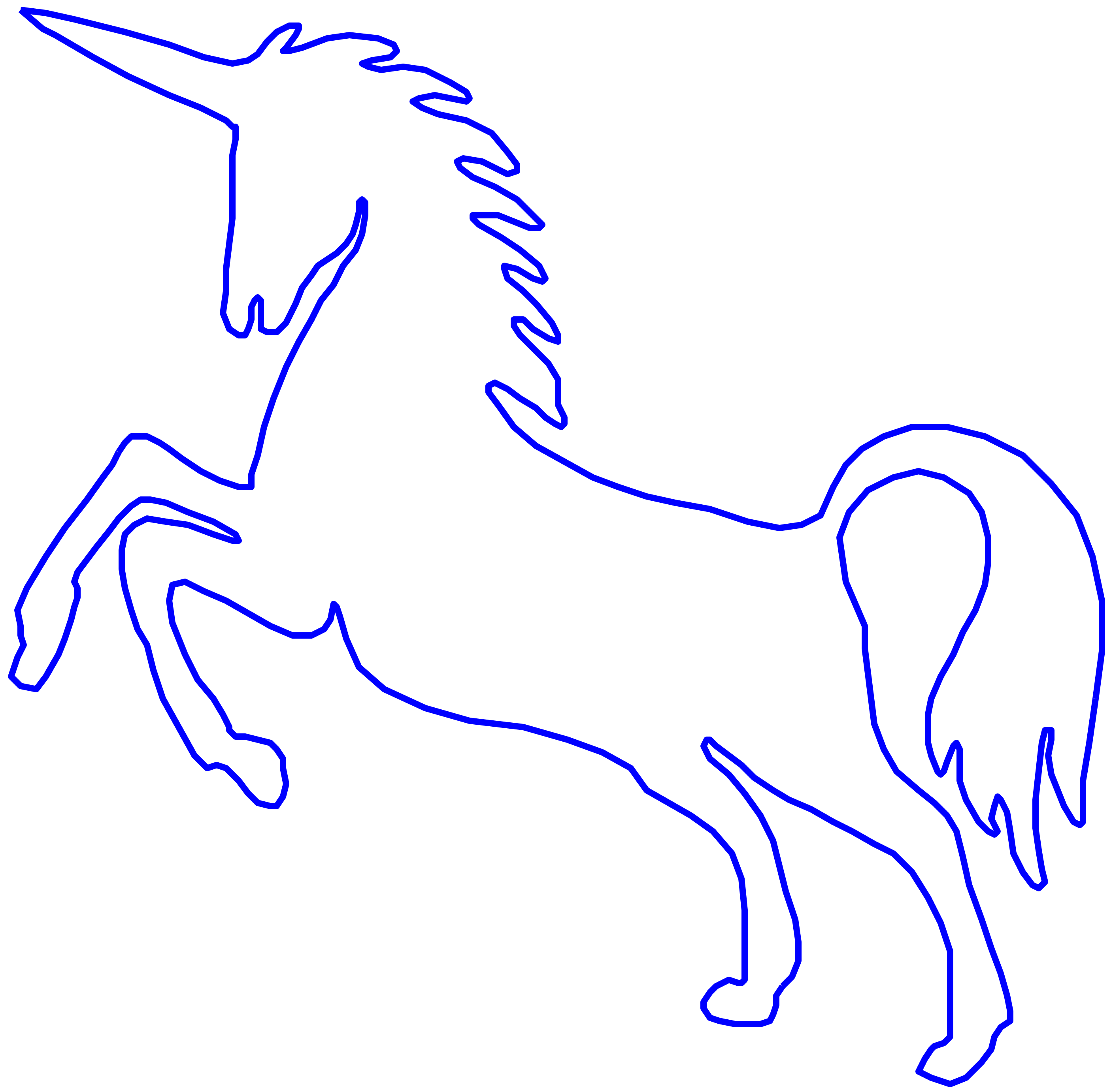
- May shrink the shape; can be slow

Filtering Curves



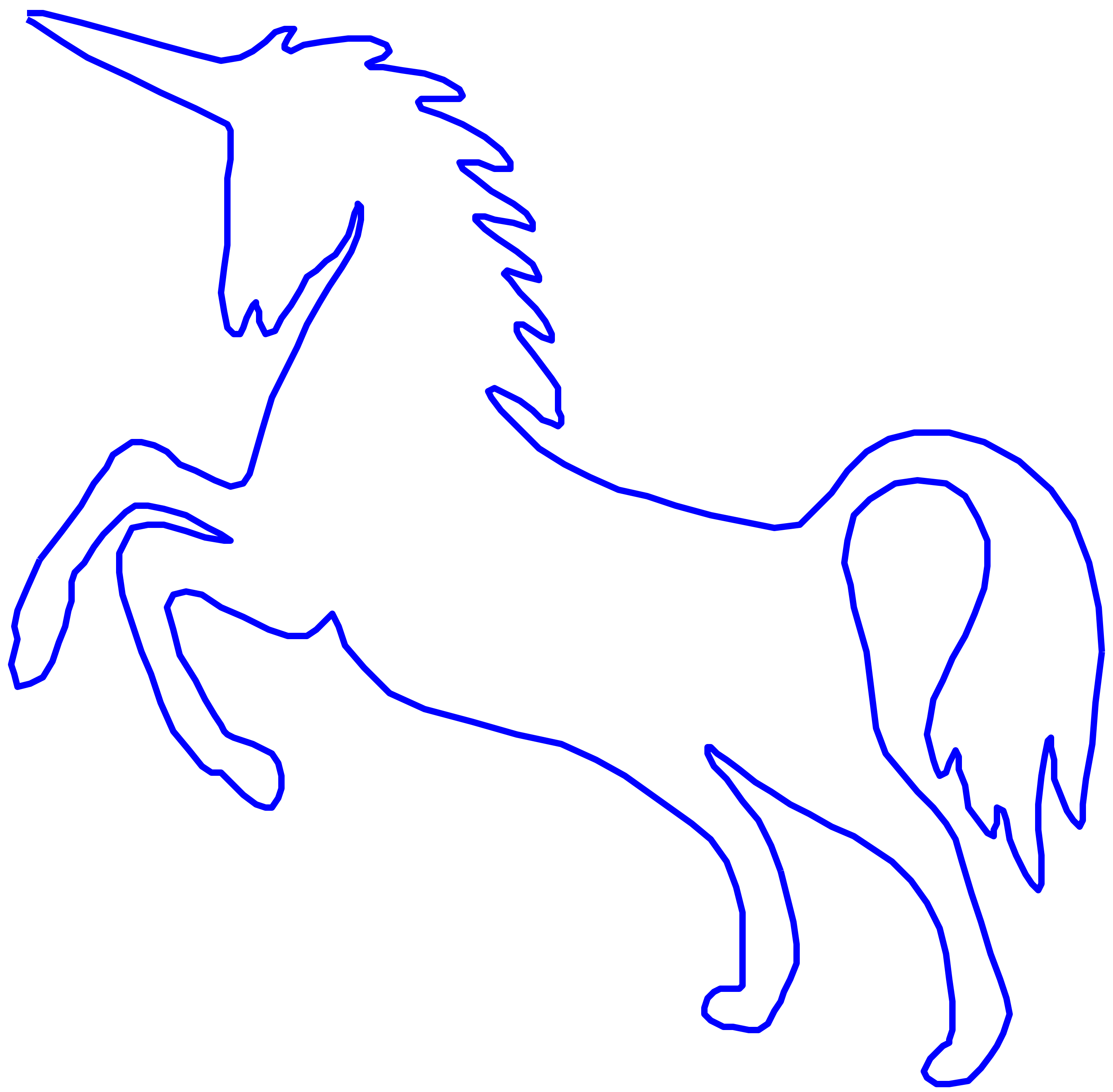
Original curve

Filtering Curves



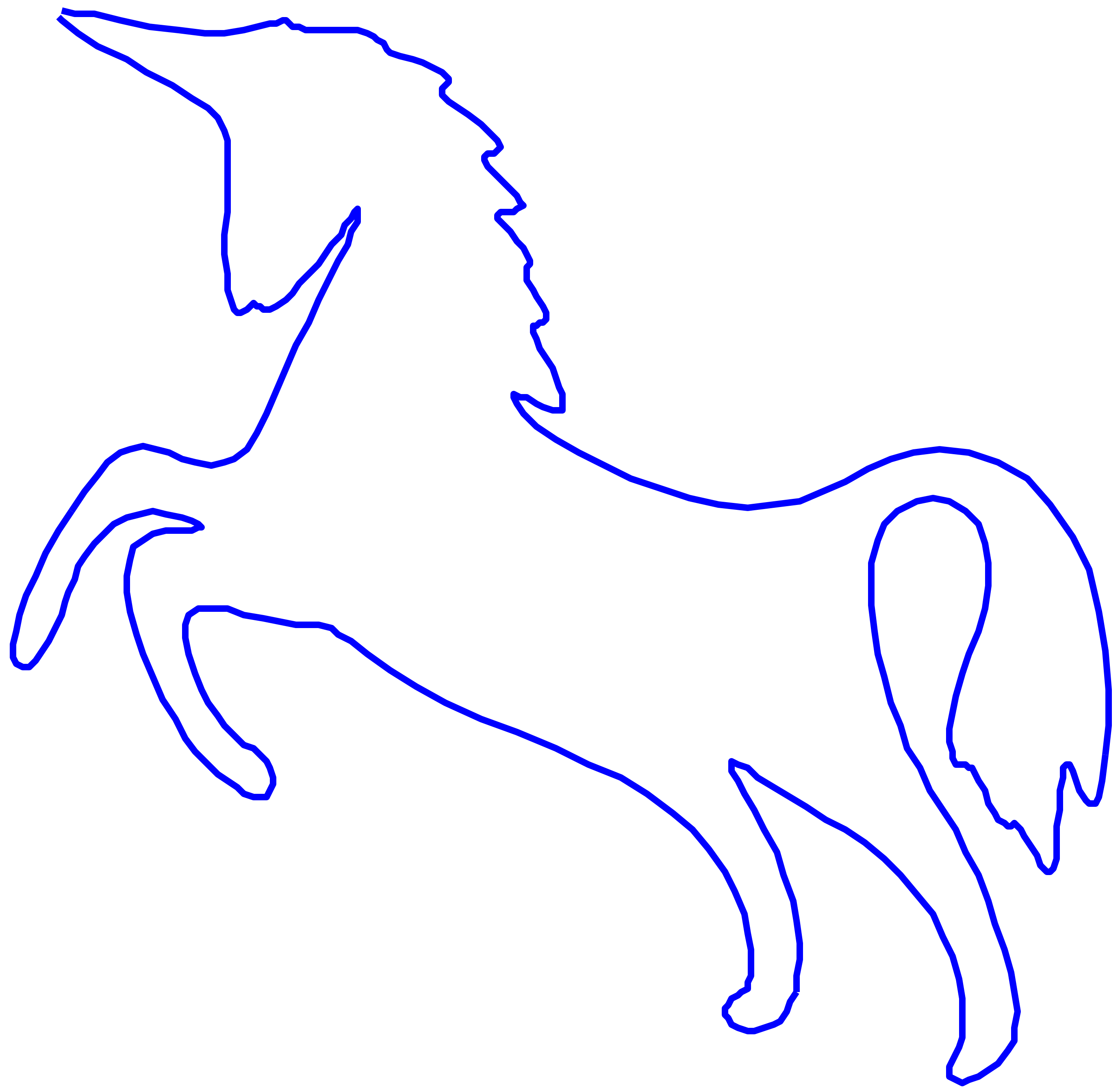
1st iteration; λ=0.5

Filtering Curves



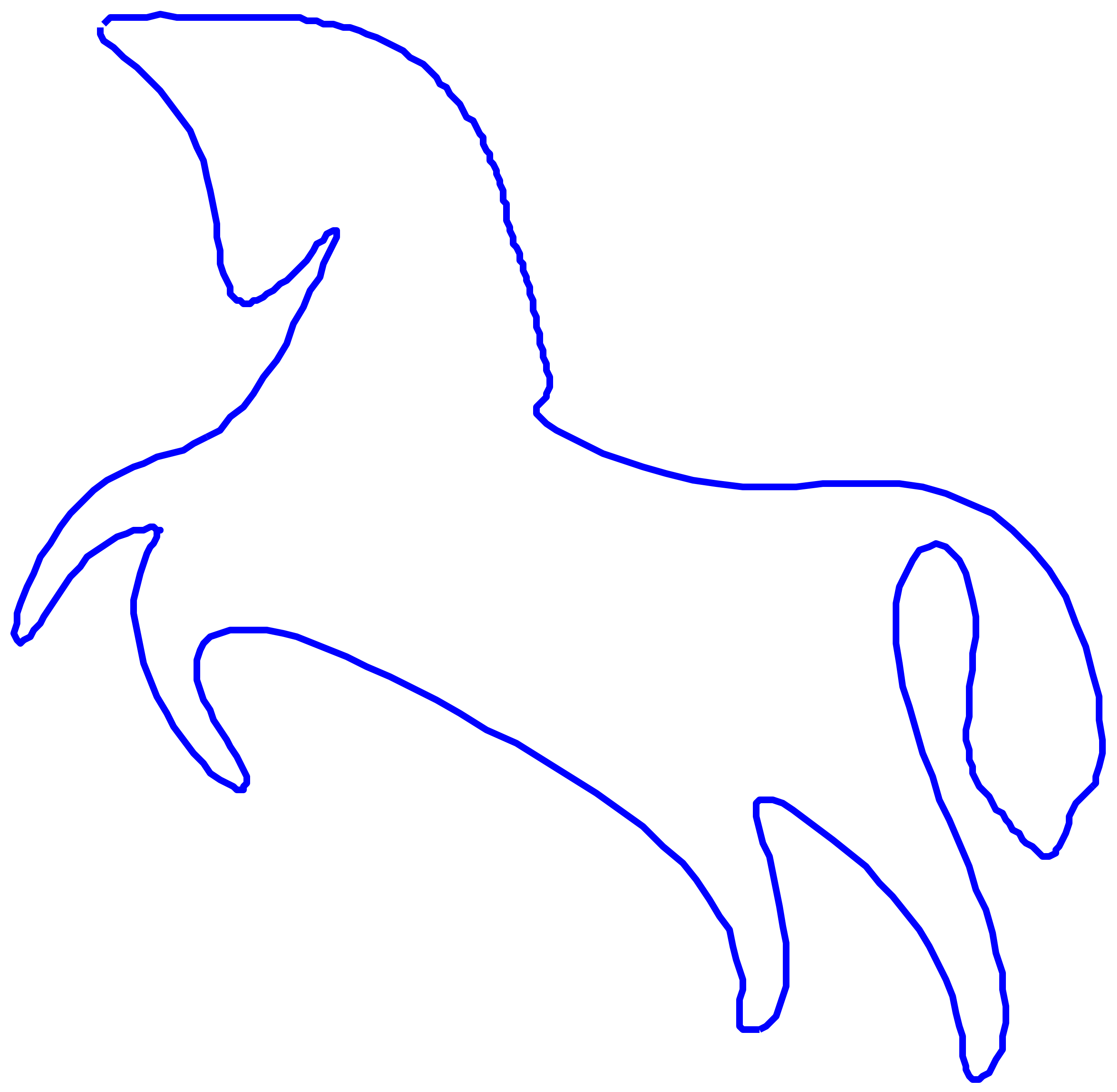
2nd iteration; $\lambda=0.5$

Filtering Curves



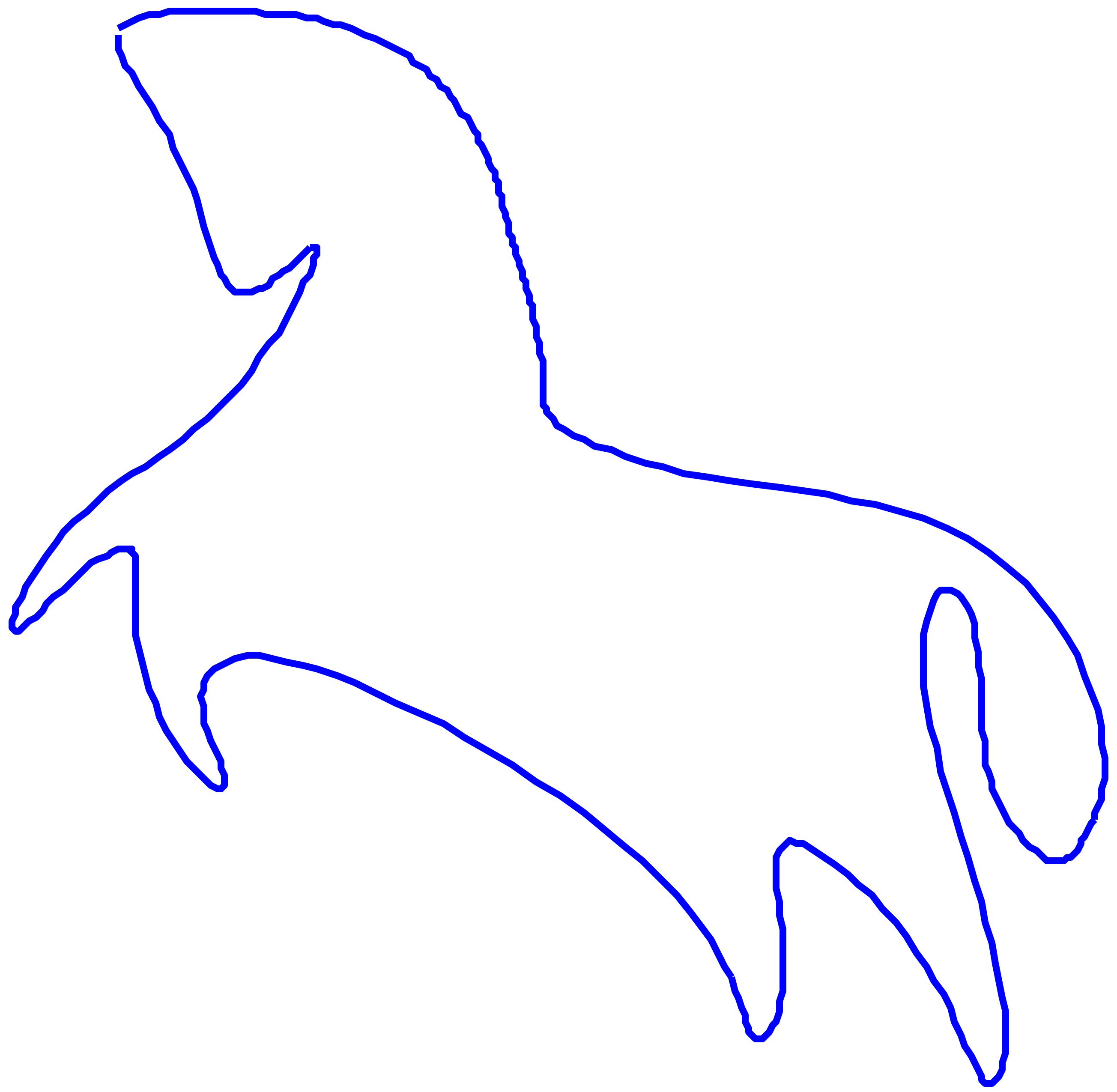
8th iteration; λ=0.5

Filtering Curves



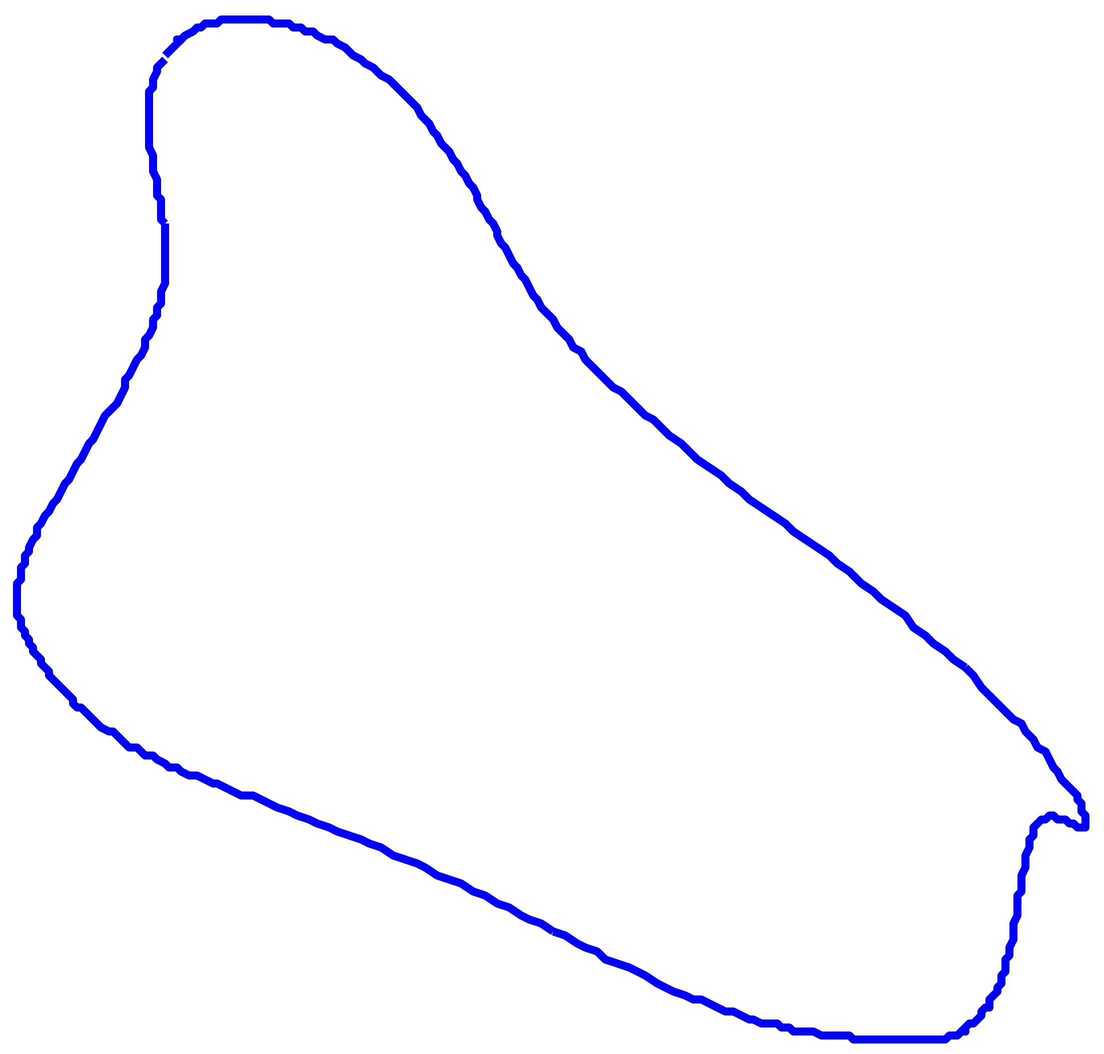
27th iteration; λ=0.5

Filtering Curves



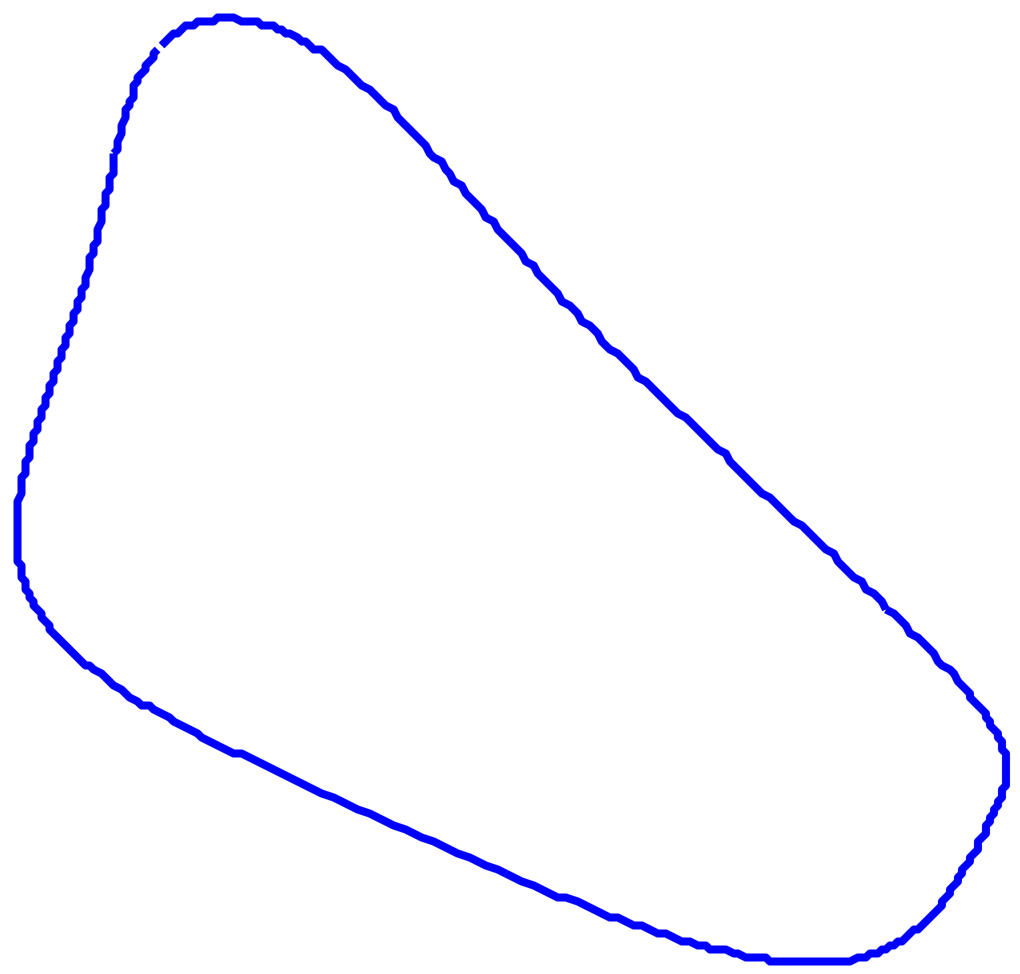
50th iteration; $\lambda=0.5$

Filtering Curves



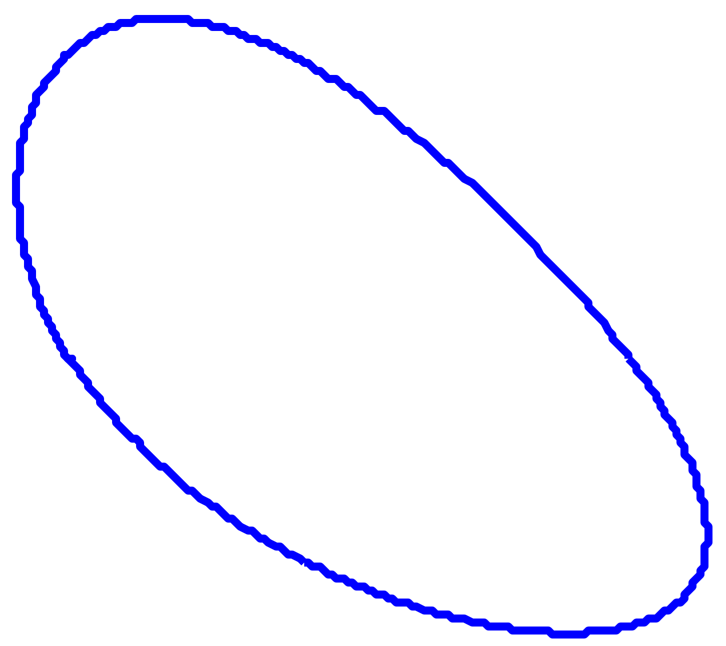
500th iteration; λ=0.5

Filtering Curves



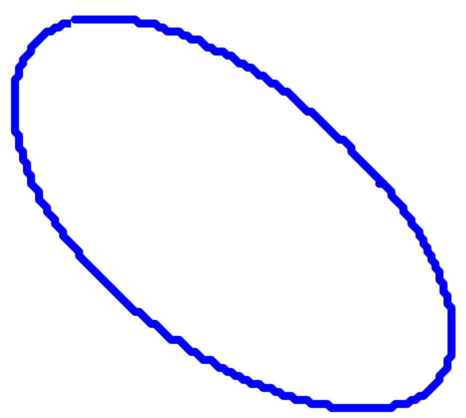
1000th iteration; $\lambda=0.5$

Filtering Curves



5000th iteration; $\lambda=0.5$

Filtering Curves



10000th iteration; λ=0.5

Filtering Curves

50000th iteration; λ=0.5

7-Smoothing

Part II

Acknowledgements: Daniele Panozzo

On meshes: smoothing as mean curvature flow

- Model smoothing as a diffusion process

$$\boxed{\frac{\partial \mathbf{p}}{\partial t} = \lambda \Delta \mathbf{p}} = -2\lambda H \mathbf{n}$$

On meshes: smoothing as mean curvature flow

- Model smoothing as a diffusion process

$$\boxed{\frac{\partial \mathbf{p}}{\partial t} = \lambda \Delta \mathbf{p}} = -2\lambda H \mathbf{n}$$

- Discretize in time, forward differences:

$$\frac{\mathbf{p}^{(n+1)} - \mathbf{p}^{(n)}}{dt} = \lambda L \mathbf{p}^{(n)}$$

On meshes: smoothing as mean curvature flow

- Model smoothing as a diffusion process

$$\boxed{\frac{\partial \mathbf{p}}{\partial t} = \lambda \Delta \mathbf{p}} = -2\lambda H \mathbf{n}$$

- Discretize in time, forward differences:

$$\frac{\mathbf{p}^{(n+1)} - \mathbf{p}^{(n)}}{dt} = \lambda L \mathbf{p}^{(n)}$$

$$\mathbf{p}^{(n+1)} - \mathbf{p}^{(n)} = dt\, \lambda L \mathbf{p}^{(n)}$$

On meshes: smoothing as mean curvature flow

- Model smoothing as a diffusion process

$$\boxed{\frac{\partial \mathbf{p}}{\partial t} = \lambda \Delta \mathbf{p}} = -2\lambda H \mathbf{n}$$

- Discretize in time, forward differences:

$$\frac{\mathbf{p}^{(n+1)} - \mathbf{p}^{(n)}}{dt} = \lambda L \mathbf{p}^{(n)}$$

$$\mathbf{p}^{(n+1)} - \mathbf{p}^{(n)} = dt\, \lambda L \mathbf{p}^{(n)}$$

$$\boxed{\mathbf{p}^{(n+1)} = (I + dt\, \lambda L)\mathbf{p}^{(n)}}$$

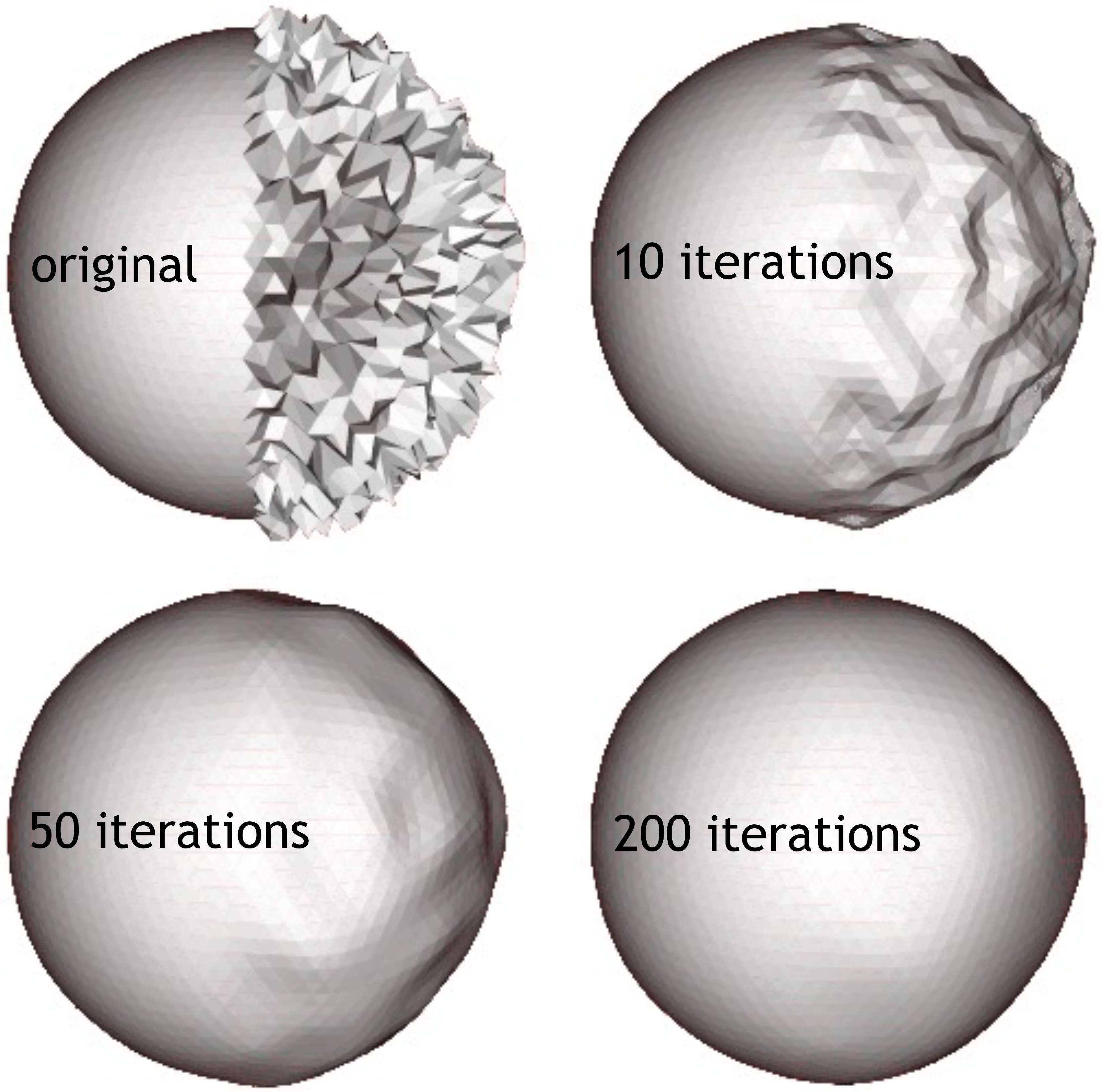
Explicit integration! Unstable unless time step dt is small

Taubin Smoothing: Explicit Steps

- Iterate:

$$\tilde{\mathbf{p}} = \mathbf{p} + \lambda L \mathbf{p} = (I + \lambda L)\mathbf{p}$$

- $\lambda > 0$ to smooth
- $\lambda < 0$ to inflate
- Originally proposed with uniform Laplacian weights



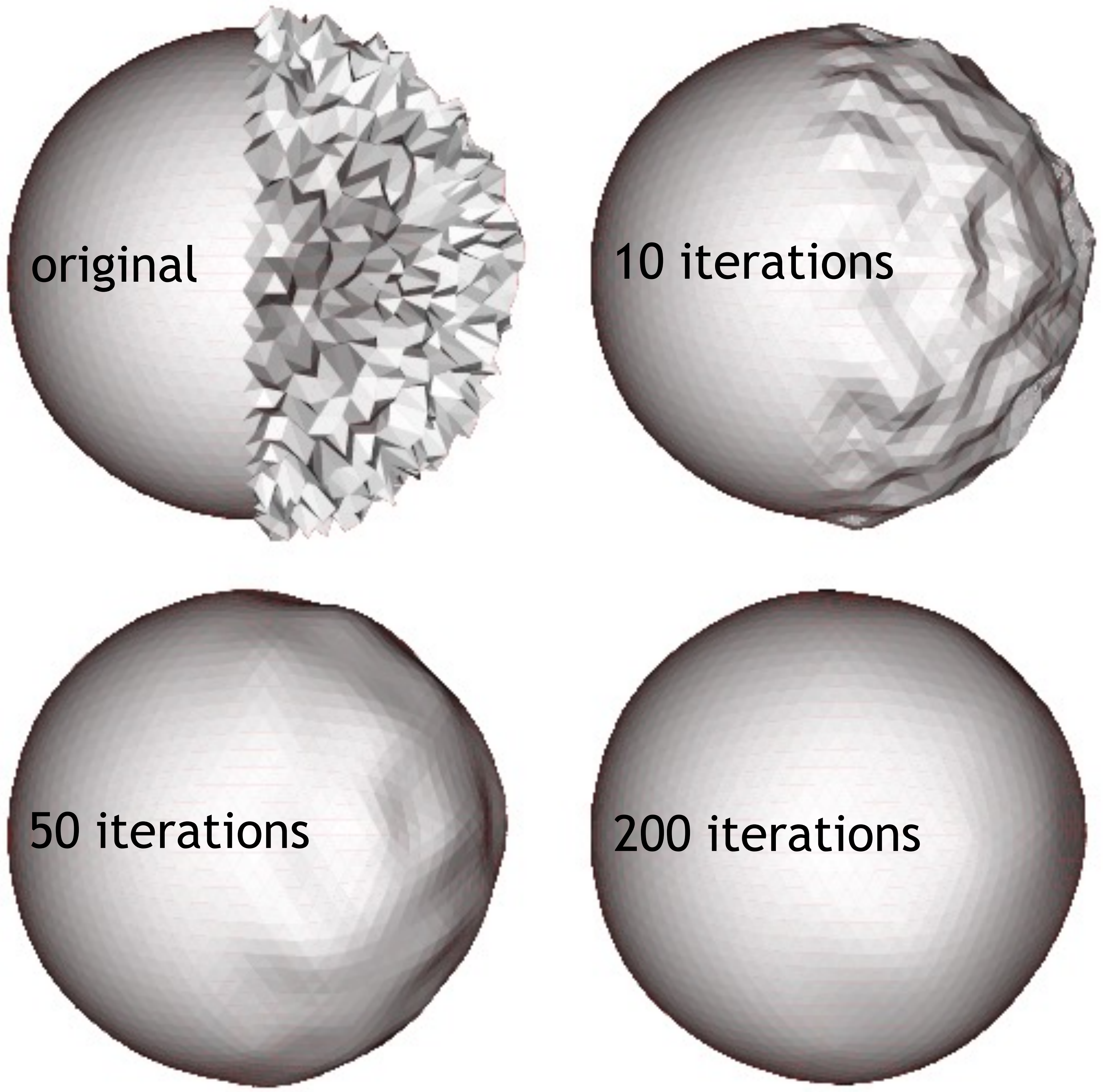
A Signal Processing Approach to Fair Surface Design
Gabriel Taubin, **ACM SIGGRAPH 95**

Taubin Smoothing: Explicit Steps

- Iterate:

$$\tilde{\mathbf{p}} = \mathbf{p} + \lambda L \mathbf{p} = (I + \lambda L)\mathbf{p}$$
$$\tilde{\mathbf{p}} = \mathbf{p} + \mu L \mathbf{p} = (I + \mu L)\mathbf{p}$$

- λ > 0 to smooth
- u < 0 to inflate
- Originally proposed with uniform Laplacian weights



A Signal Processing Approach to Fair Surface Design
Gabriel Taubin, **ACM SIGGRAPH 95**

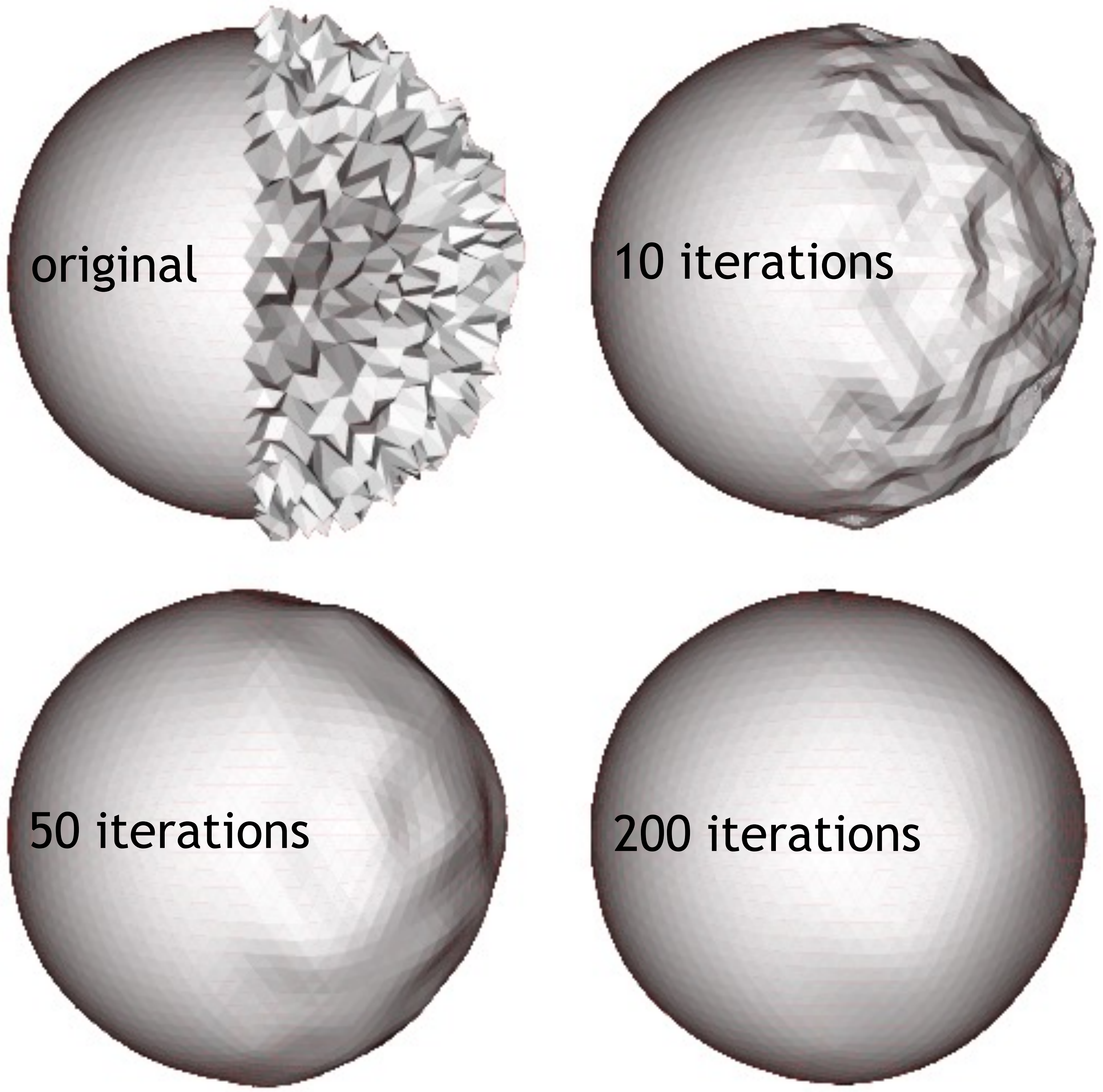
Taubin Smoothing: Explicit Steps

- Per-vertex iterations

$$\tilde{\mathbf{p}}_i = \mathbf{p}_i + \lambda L(\mathbf{p}_i)$$

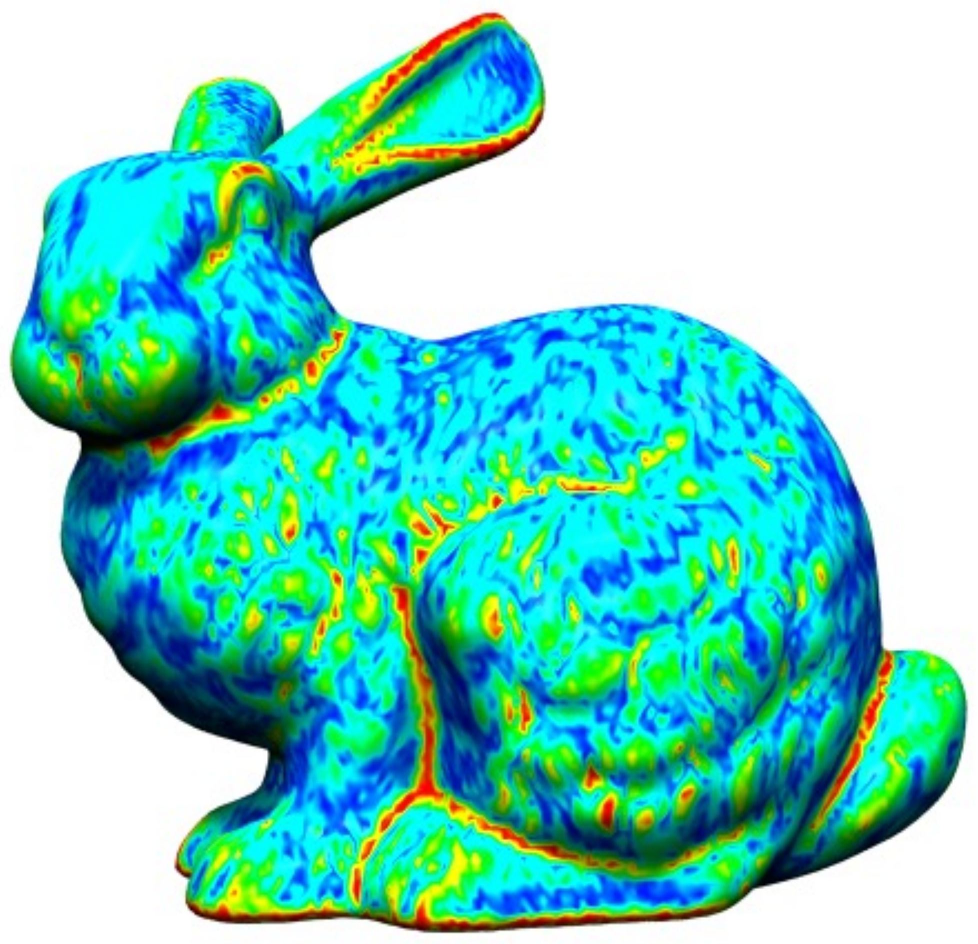
$$\tilde{\mathbf{p}}_i = \mathbf{p}_i + \mu L(\mathbf{p}_i)$$

- Simple to implement
- Requires many iterations
- Need to tweak λ, u

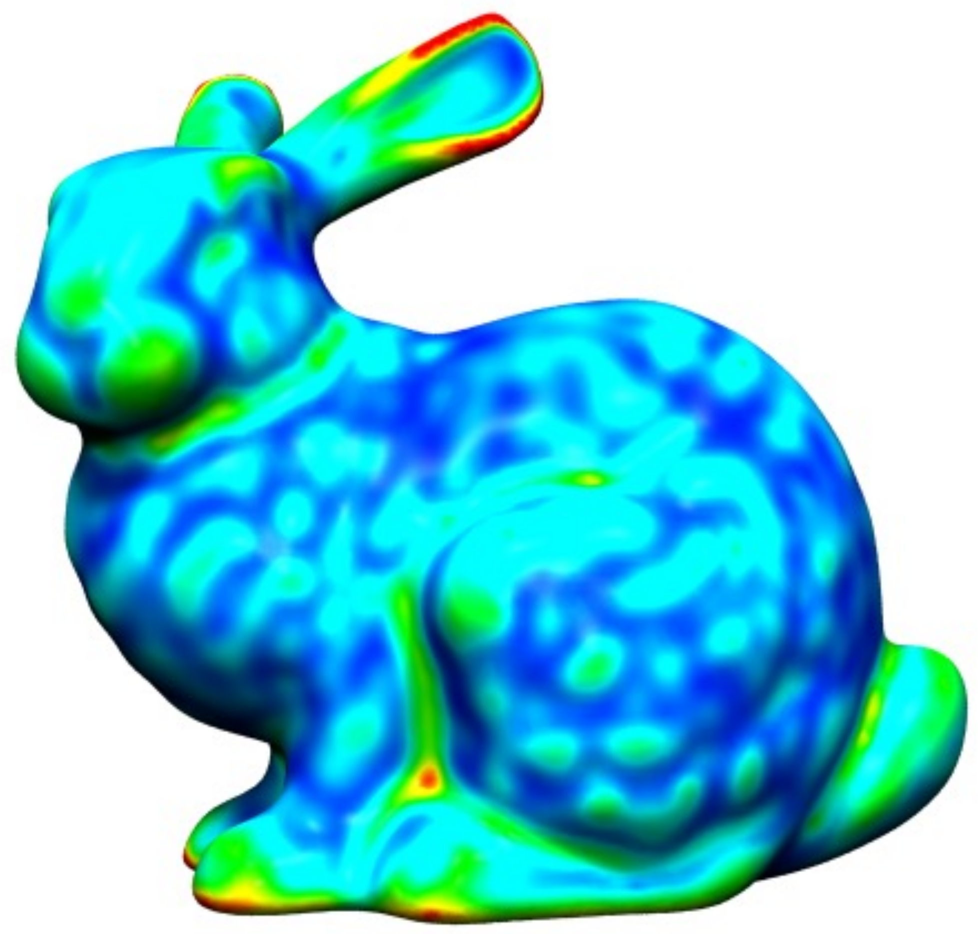


A Signal Processing Approach to Fair Surface Design
Gabriel Taubin, **ACM SIGGRAPH 95**

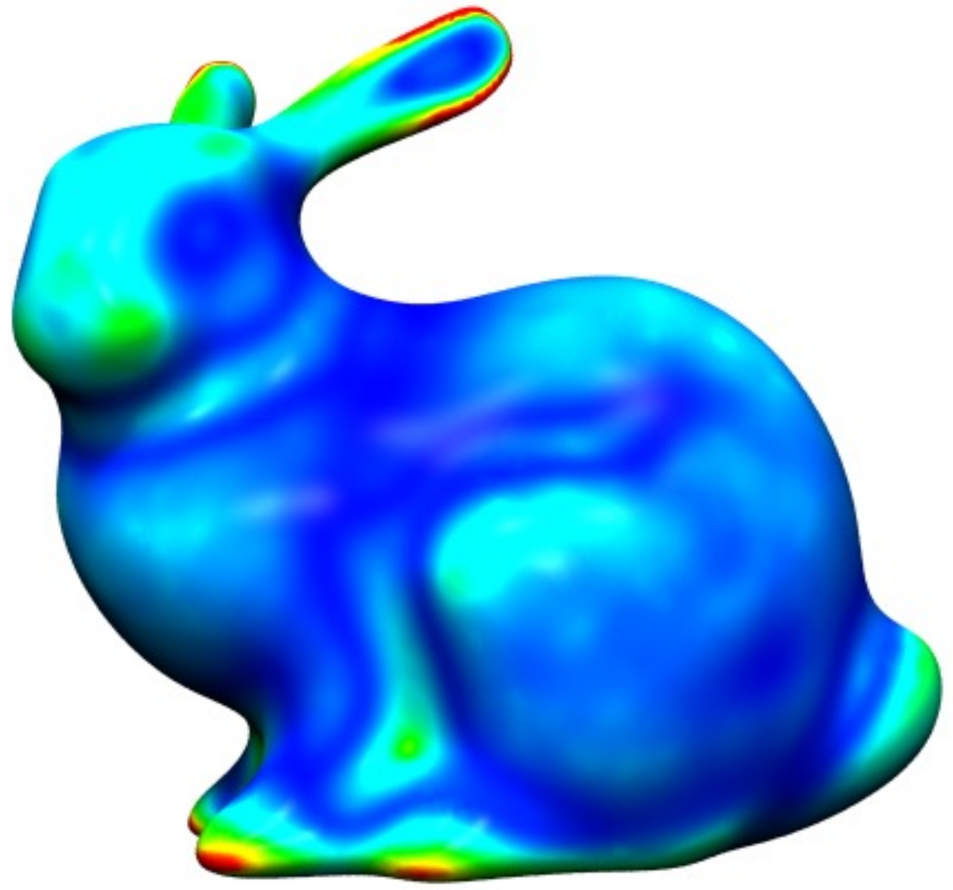
Example



0 iterations



10 iterations



100 iterations

Smoothing as (mean curvature) Flow

- Model smoothing as a diffusion process

$$\boxed{\frac{\partial \mathbf{p}}{\partial t} = \lambda \Delta \mathbf{p}} = -2\lambda H \mathbf{n}$$

- Backward Euler for unconditional stability

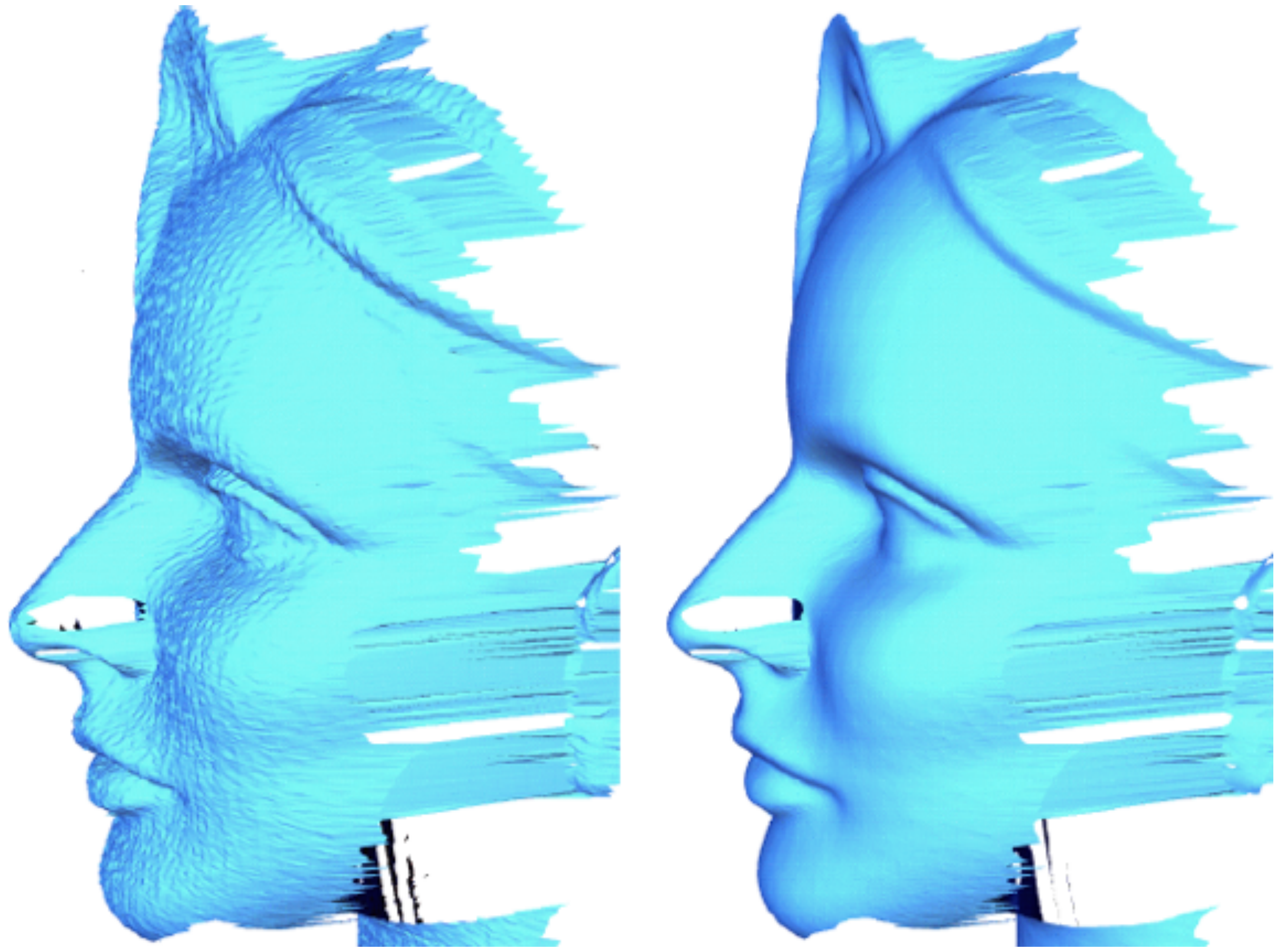
$$\frac{\mathbf{p}^{(n+1)} - \mathbf{p}^{(n)}}{dt} = \lambda L \mathbf{p}^{\boxed{(n+1)}}$$

$$\mathbf{p}^{(n+1)} - \mathbf{p}^{(n)} = dt\, \lambda L \mathbf{p}^{(n+1)}$$

$$\boxed{(I - dt\, \lambda L)\mathbf{p}^{(n+1)} = \mathbf{p}^{(n)}}$$

Implicit Fairing: Implicit Euler Steps

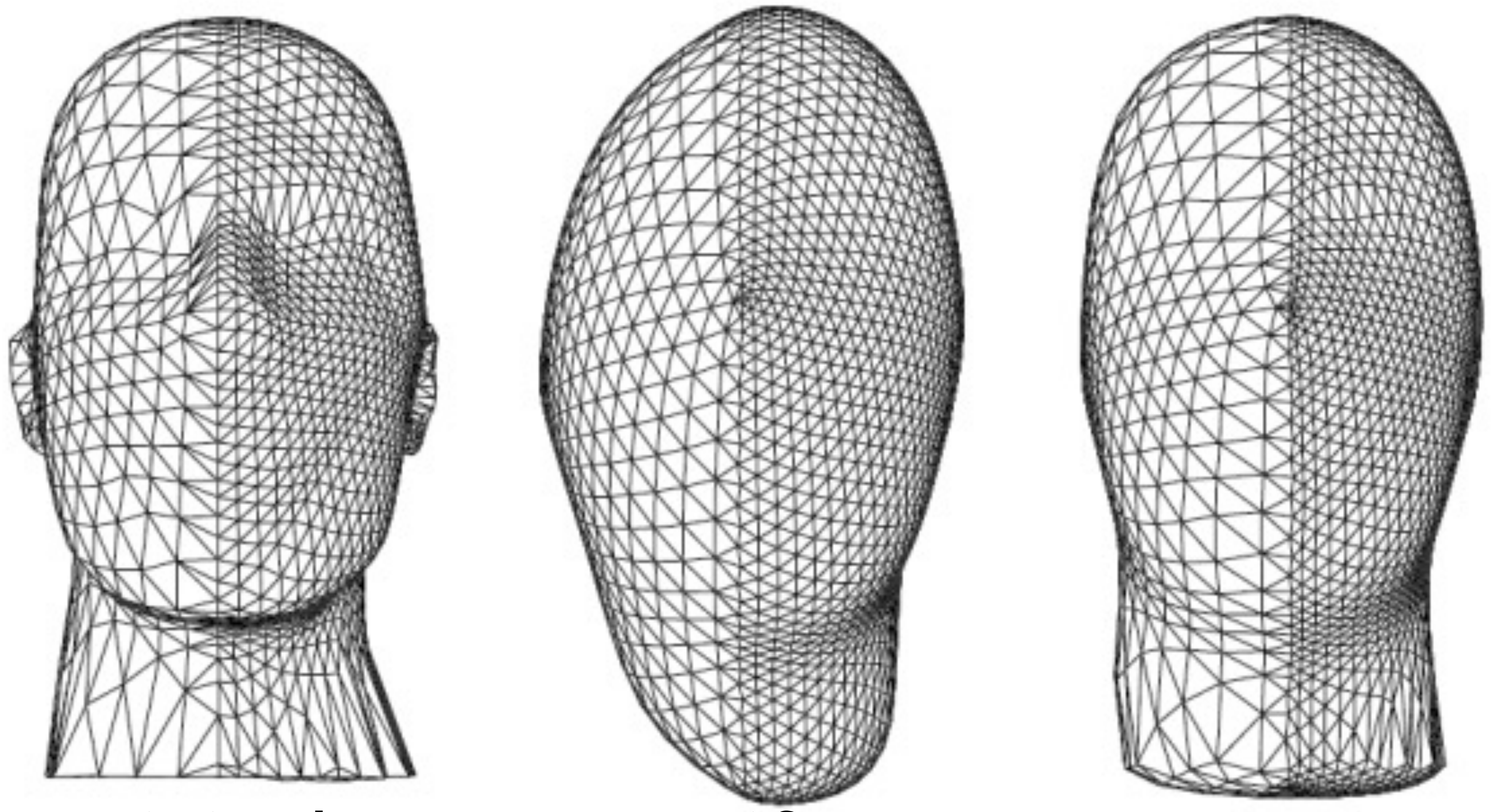
- In each iteration, solve for the smoothed $\tilde{\mathbf{p}}$: $(I - \lambda L)\tilde{\mathbf{p}} = \mathbf{p}$



Implicit fairing of irregular meshes using diffusion and curvature flow
M. Desbrun, M. Meyer, P. Schroeder, A. Barr, **ACM SIGGRAPH 99**

Mesh Independence

- Result of smoothing with uniform Laplacian depends on triangle density and shape
  - Why?
- Asymmetric results although underlying geometry is symmetric

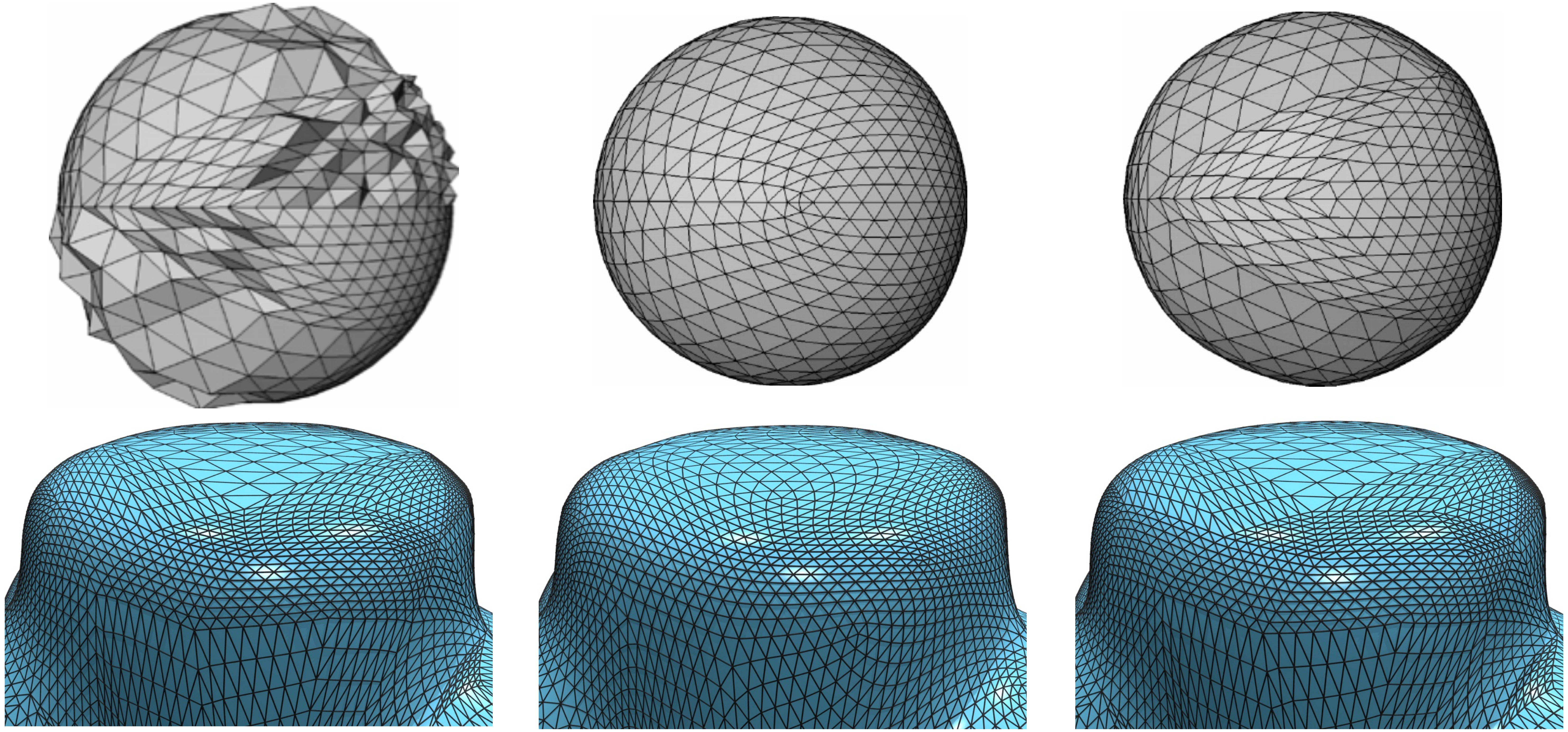


original uniform cotan

Comparison of the weights

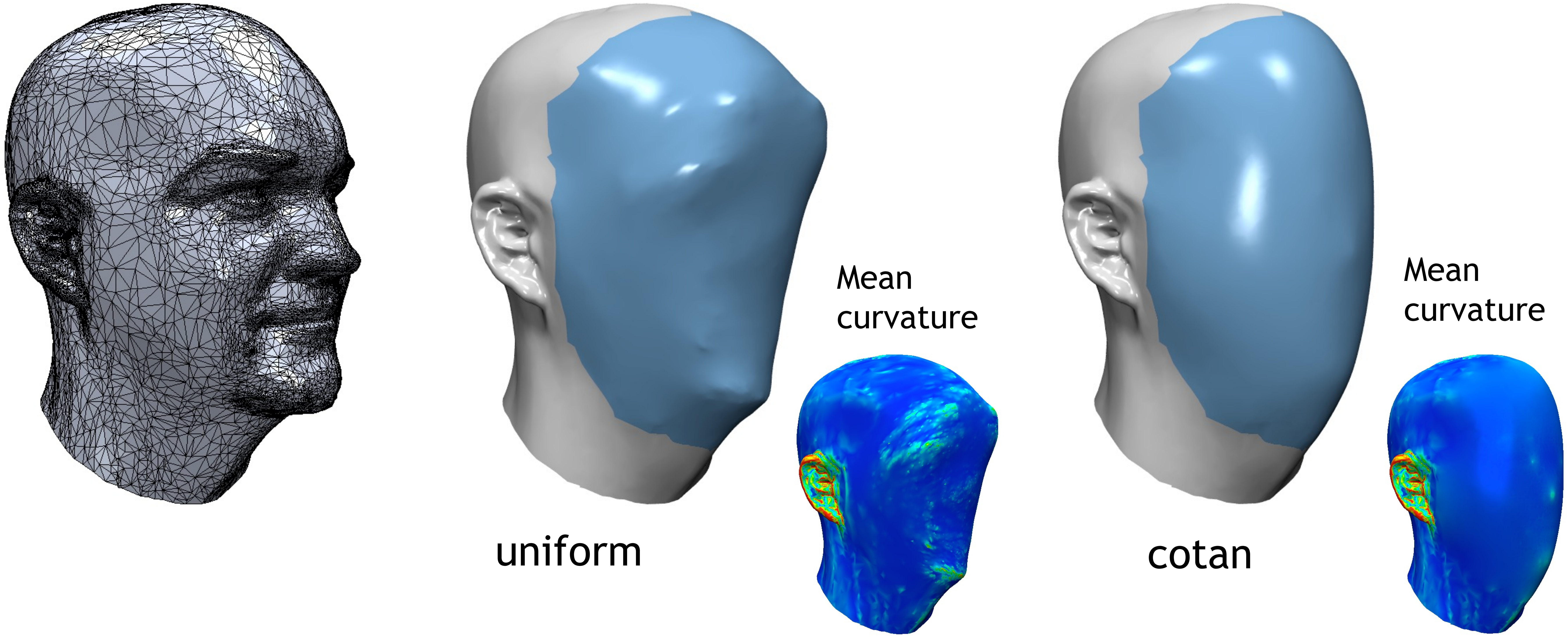
- Explicit flow with different weights:

original uniform cotan



Implicit Fairing

- The importance of using the right weights



Smoothing as optimization

Minimizing a smoothness energy

$$\Delta_{\mathcal{M}}\mathbf{p} = -2H\mathbf{n}$$

- Let's go for $H = 0$ goal: $H = 0$ or $H = \mathrm{const}$
- only trivial solution, no connection to initial surface p

$$\Delta_{\mathcal{M}}\tilde{\mathbf{p}} = 0$$

Minimizing a smoothness energy

$$\Delta_{\mathcal{M}} \mathbf{p} = -2H\mathbf{n}$$

- Let's go for $H = 0$ goal: $H = 0$ or $H = \text{const}$
- only trivial solution, no connection to initial surface p
- Let's regularize!

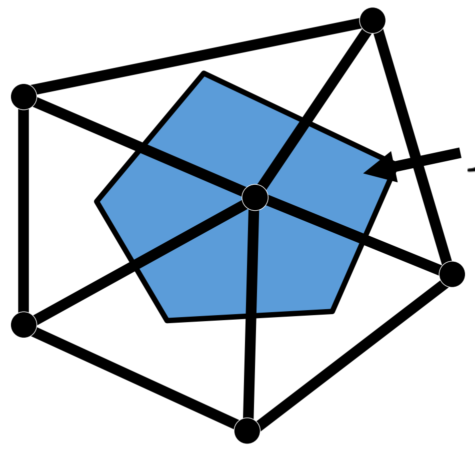
$$\Delta_{\mathcal{M}} \tilde{\mathbf{p}} = 0$$

weighting factor (like $1/\lambda$)

$$\min_{\tilde{\mathbf{p}}} \int_{\mathcal{M}} \underbrace{\|\Delta_{\mathcal{M}} \tilde{\mathbf{p}}\|^2}_{\text{small } H} + w \underbrace{\|\tilde{\mathbf{p}} - \mathbf{p}\|^2}_{\text{stay close to original surface}}$$

Minimizing a smoothness energy

- Discretize: $$\min_{\tilde{\mathbf{p}}} \int_{\mathcal{M}} \|\Delta_{\mathcal{M}} \tilde{\mathbf{p}}\|^2 + w\|\tilde{\mathbf{p}} - \mathbf{p}\|^2$$



- Minimize! $$\min_{\tilde{\mathbf{p}}} \sum_{i=1}^{n} A_i \left(\|L\tilde{\mathbf{p}}_i\|^2 + w\|\tilde{\mathbf{p}}_i - \mathbf{p}_i\|^2\right)$$

$$\frac{\partial}{\partial \tilde{\mathbf{p}}} = 0$$

remember: $$\frac{\partial}{\partial \mathbf{x}}(\mathbf{x}^T A \mathbf{x}) = (A + A^T)\mathbf{x}$$

tip: google “The Matrix Cookbook”

Minimizing a smoothness energy

$$\min_{\tilde{\mathbf{p}}} \sum_{i=1}^{n} A_i \left(\|L\tilde{\mathbf{p}}_i\|^2 + w\|\tilde{\mathbf{p}}_i - \mathbf{p}_i\|^2 \right)$$

$$\tilde{\mathbf{p}} = [\tilde{\mathbf{x}}\ \tilde{\mathbf{y}}\ \tilde{\mathbf{z}}] = \begin{pmatrix} \tilde{p}_{1x} & \tilde{p}_{1y} & \tilde{p}_{1z} \\ \tilde{p}_{2x} & \tilde{p}_{2y} & \tilde{p}_{2z} \\ \vdots & \vdots & \vdots \\ \tilde{p}_{nx} & \tilde{p}_{ny} & \tilde{p}_{nz} \end{pmatrix} \in \mathbb{R}^{n\times 3}$$

$$E(\tilde{\mathbf{p}}) = (L\tilde{\mathbf{p}})^T M (L\tilde{\mathbf{p}}) + w(\tilde{\mathbf{p}} - \mathbf{p})^T M (\tilde{\mathbf{p}} - \mathbf{p})$$

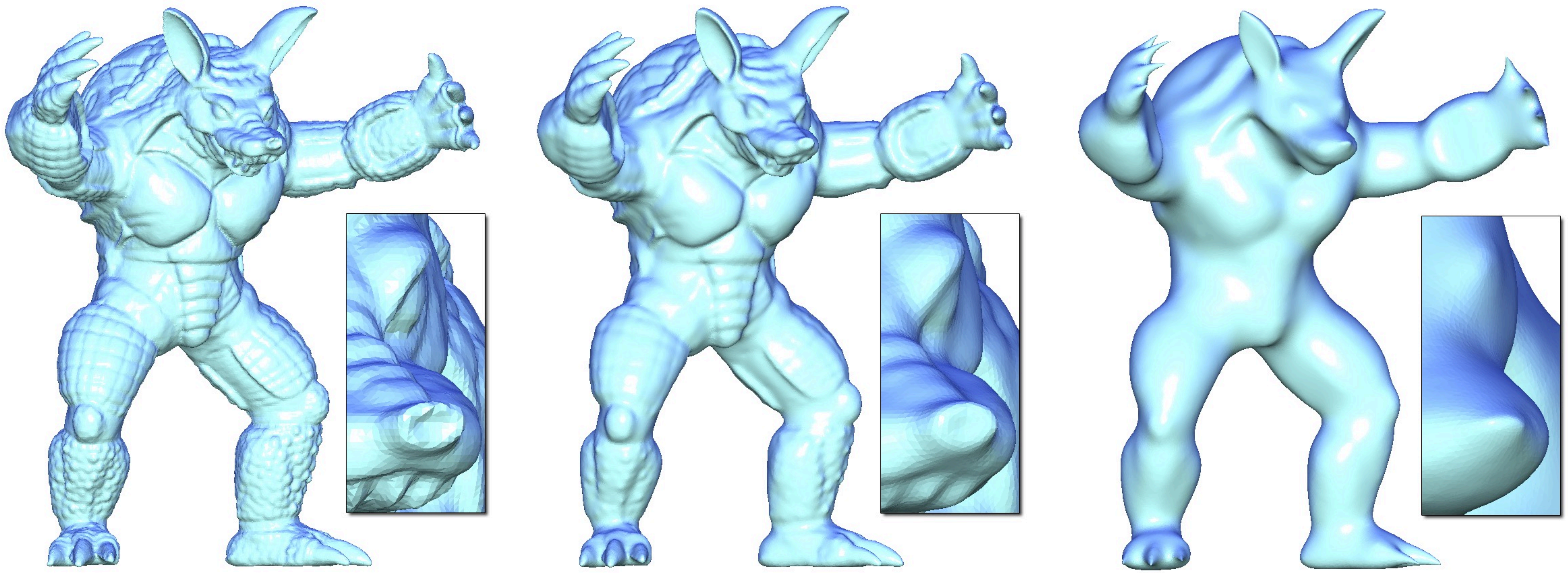
$$\frac{\partial E}{\partial \tilde{\mathbf{p}}} = 2L^T M L\tilde{\mathbf{p}} + 2wM(\tilde{\mathbf{p}} - \mathbf{p}) \stackrel{!}{=} 0$$

$$\Rightarrow \underline{\left(L^T M L + wM\right) \tilde{\mathbf{p}} = wM\mathbf{p}}$$ compare with implicit Euler!

$$NB:$$

$$L = M^{-1} L_w \ \Rightarrow\ L^T M L = L_w M^{-1} L_w$$

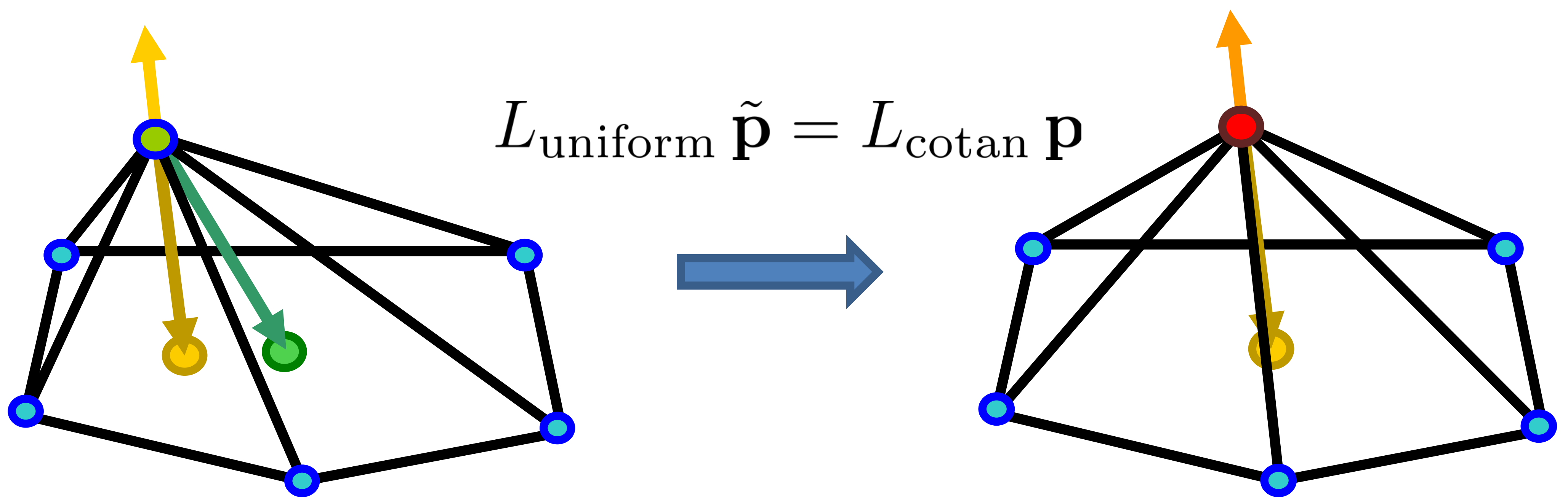
Results



original $\qquad w = 0.2 \qquad w = 0.02$

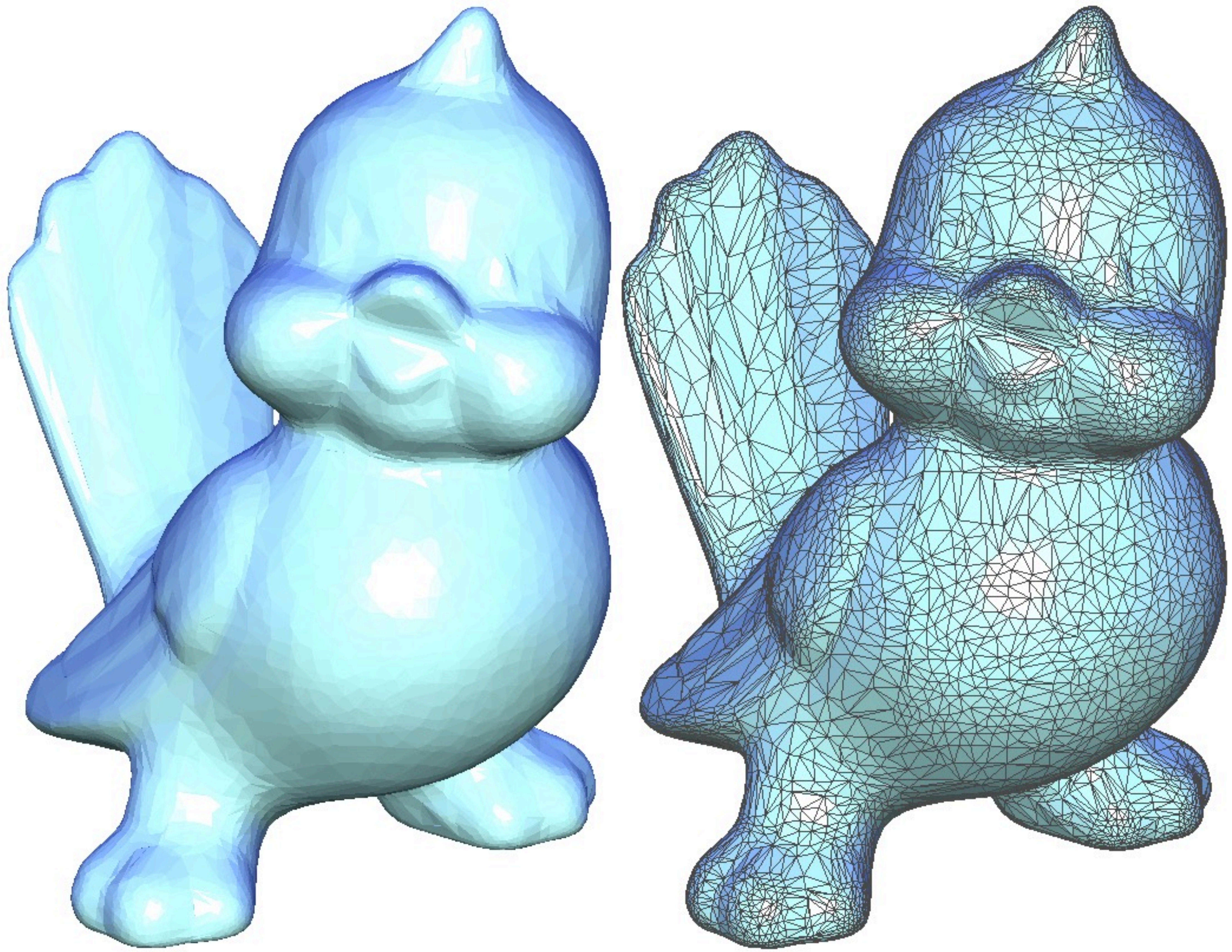
Customize the energy functional

- Can do *tangential* smoothing!
  - Improve the shapes of mesh triangles without changing the surface shape

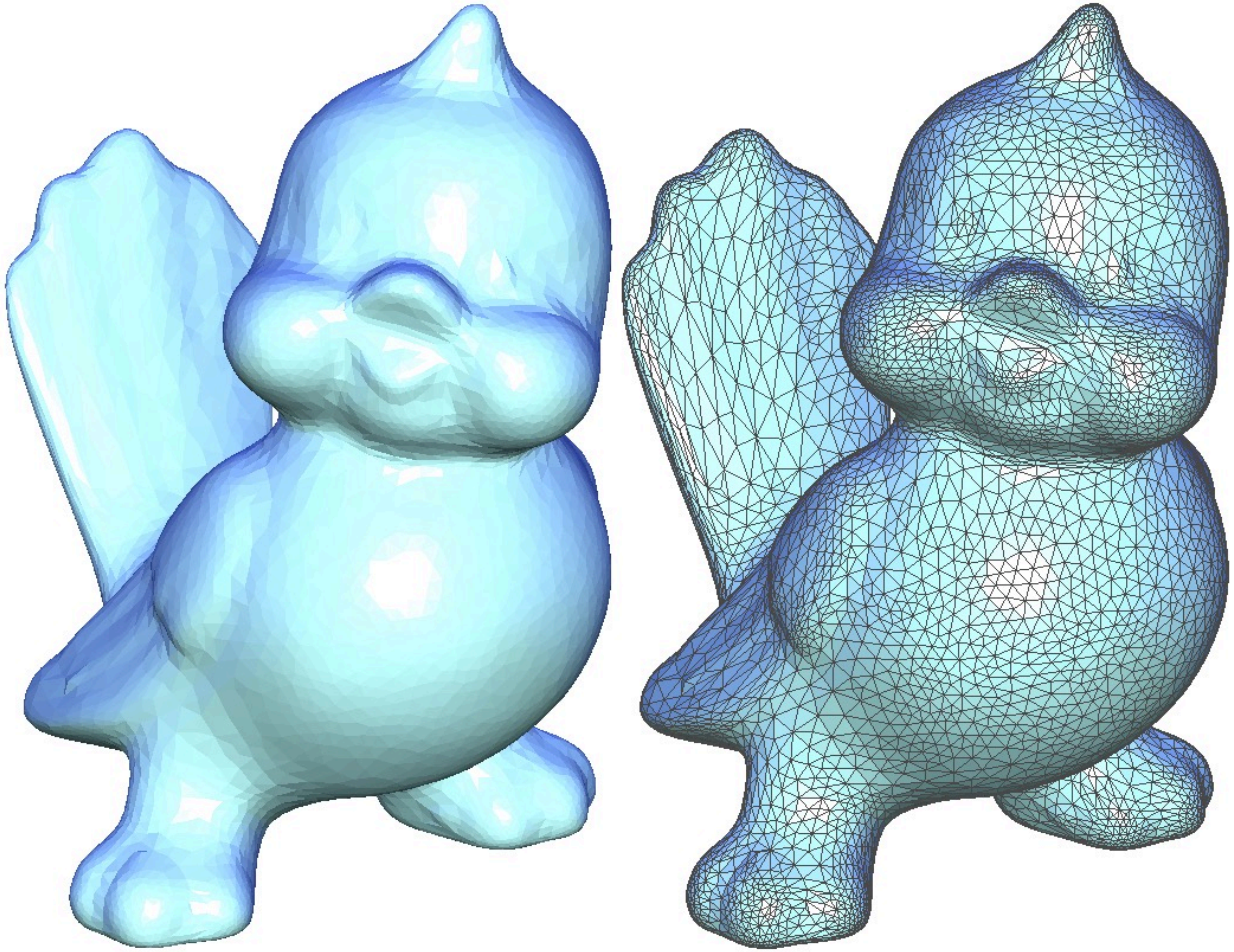


$$\min_{\tilde{\mathbf{p}}} \sum_{i=1}^{n} A_i \left(\|L_{\text{uniform}}\,\tilde{\mathbf{p}}_i - L_{\text{cotan}}\,\mathbf{p}_i\|^2 + w\|\tilde{\mathbf{p}}_i - \mathbf{p}_i\|^2 \right)$$

Original



Triangle Shape Optimization



Thank you