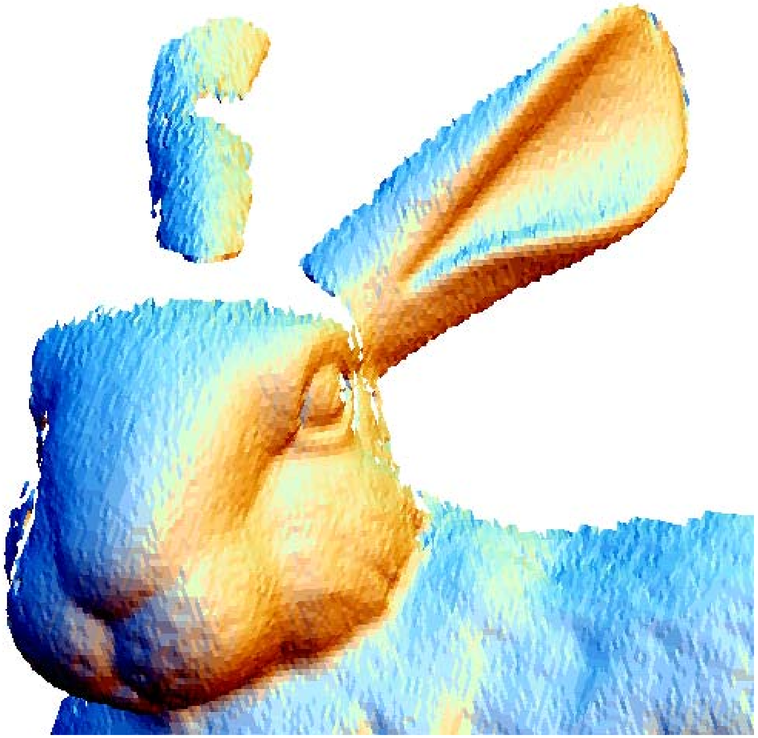
Graphics&Geometric Computing Lab
@USTC

Mesh Smoothing

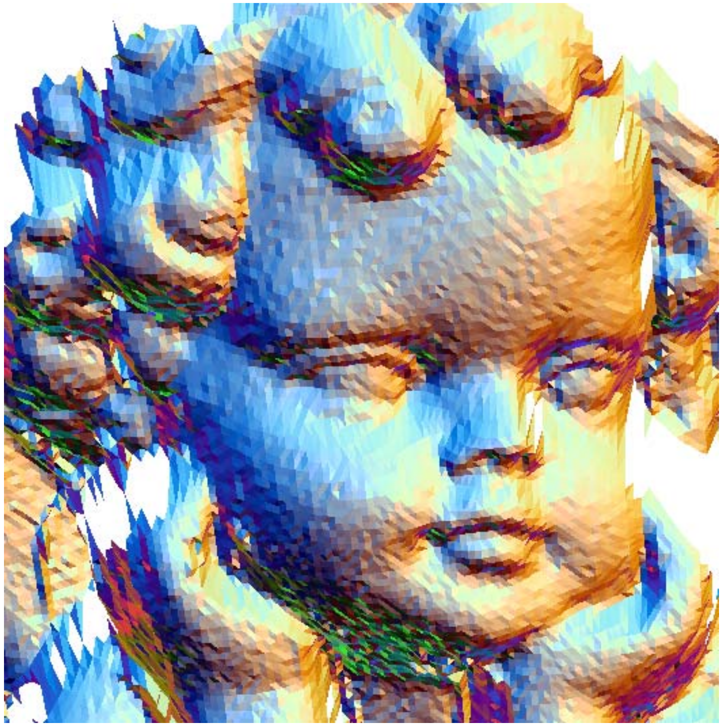
Ligang Liu
Graphics&Geometric Computing Lab
USTC
http://staff.ustc.edu.cn/~lgliu

Noise on Meshes

Meshes obtained from real world objects are often noisy.

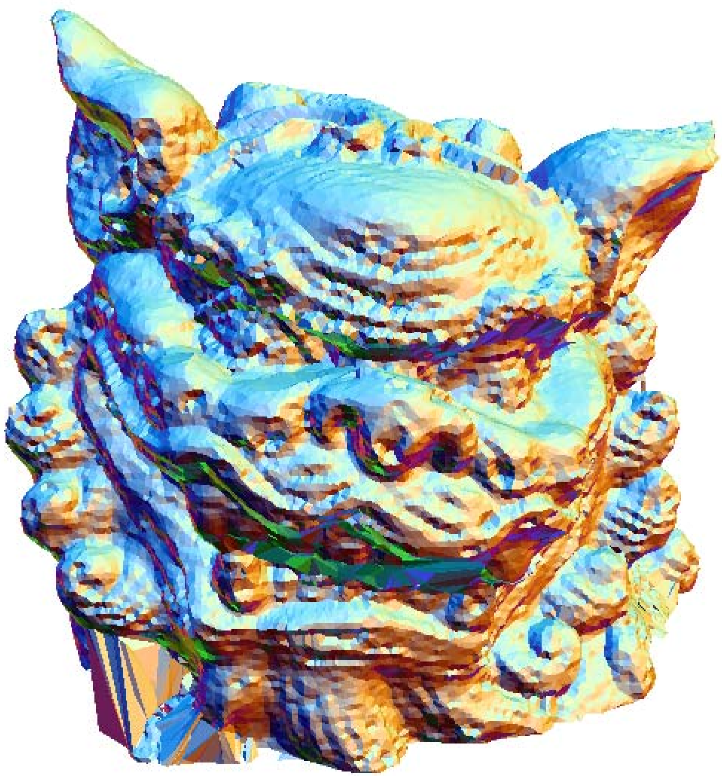


From range image of Stanford Bunny



Angel model from shadow scanning

Mesh Smoothing is Required



Mesh Smoothing

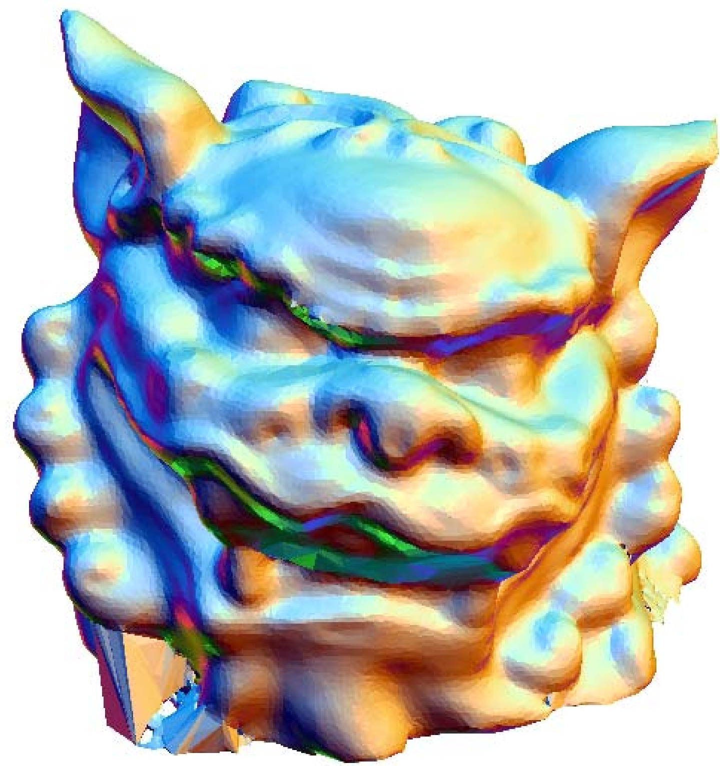


Image denoising

- Wavelet denoising [*Donoho '95*]
- Anisotropic diffusion [*Perona & Malik '90*]
- Bilateral filter [*Smith & Brady '97*], [*Tomasi & Manduchi '98*]

- [*Black et al. '98*]
  - Anisotropic diffusion
  - Robust statistics
- [*Elad '01*], [*Durand & Dorsey '02*] relate
  - Anisotropic diffusion
  - Robust statistics
  - Bilateral filter

Image Examples

Original and noisy (σ^2=900) images

TV filtering:

10 iterations (MSE=146.3339)

50 iterations (MSE=131.5013)

Wavelet Denoising (soft)

Using DB3 (MSE=144.7436) | Using DB5 (MSE=150.7006)

Filtering via the Bilateral

2 iterations with 11×11 (MSE=89.2516)

Sub-gradient based 5×5 (MSE=93.4024)

In the Literature

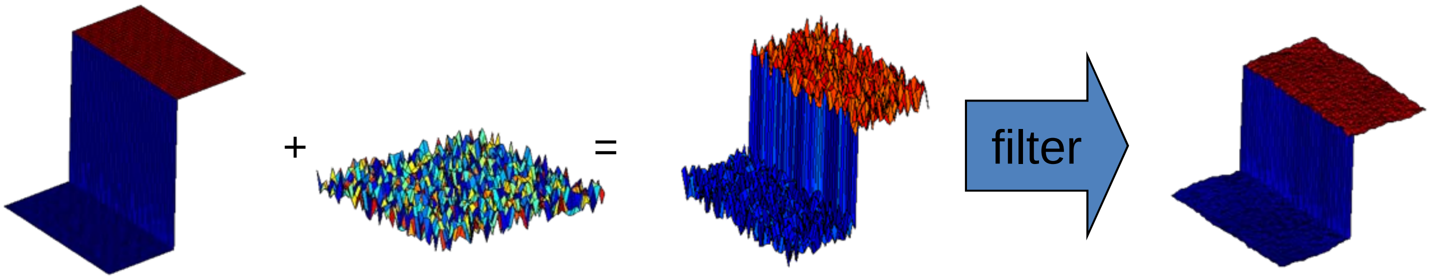
- Fast Mesh Smoothing
  - Taubin 1995
- Feature Preserving
  - Clarenz et al. 2000; Desbrun et al. 2000; Meyer et al. 2002; Zhang and Fiume 2002; Bajaj and Xu 2003
- Diffusion on Normal Field
  - Taubin 2001; Belyaev and Ohtake 2001; Ohtake et al. 2002; Tasdizen et al. 2002
- Wiener Filtering of Meshes
  - Peng et al. 2001; Alexa 2002; Pauly and Gross 2001 (points)
- Bilateral filtering
  - Choudhury and Tumblin 2003, Jones et al. 2003, Fleishman et al. 2005
- Global Smoothing
  - Desbrun et al. 1999, Ji et al. 2005

But…

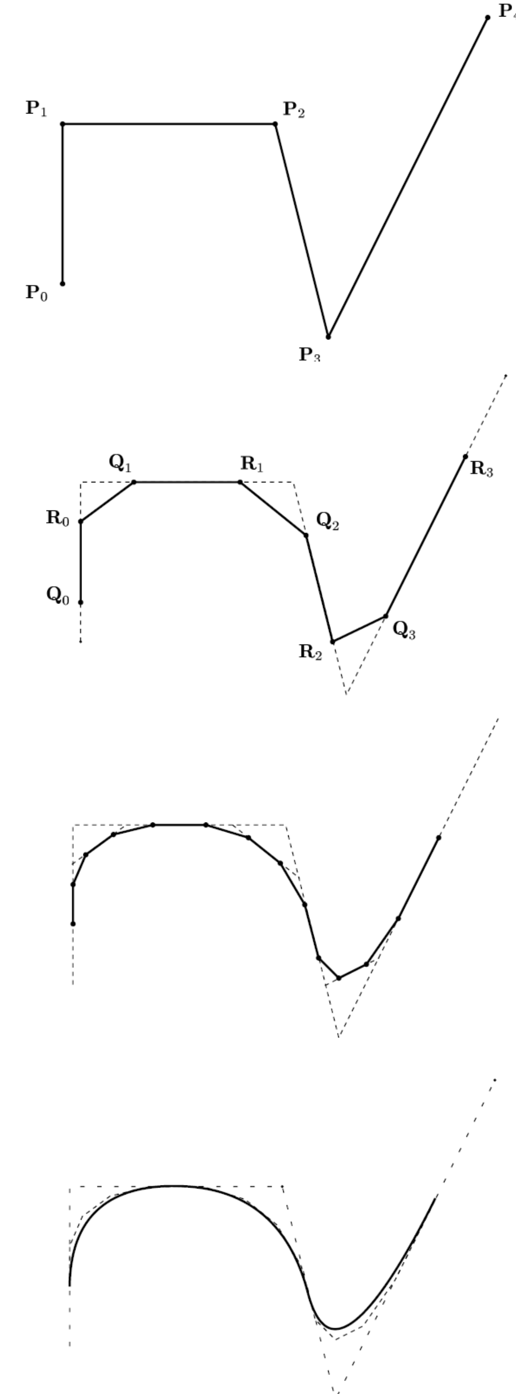
- What is noise on a surface?
  - Small bumps on the surface
  - High-frequent tiny parts on the surface
  - …
- NO accurate math definition!
- How to detect the noise?
  - High curvature parts
  - High fairing energy parts
  - …

How to denoise?

- Eliminate high frequency
- Preserve global features



Smoothing Everywher



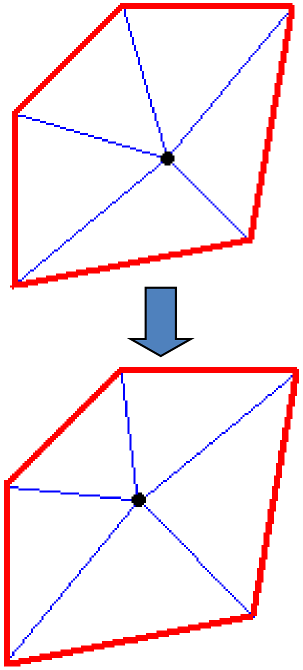
- Real life applications
  - Sculpture
  - Decoration
- Methods
  - Corner cutting
- Geometric modeling
  - Chaikin's scheme
  - Bézier: de Castljau algorithm
  - B-spline: knot insertion
  - Subdivision surface

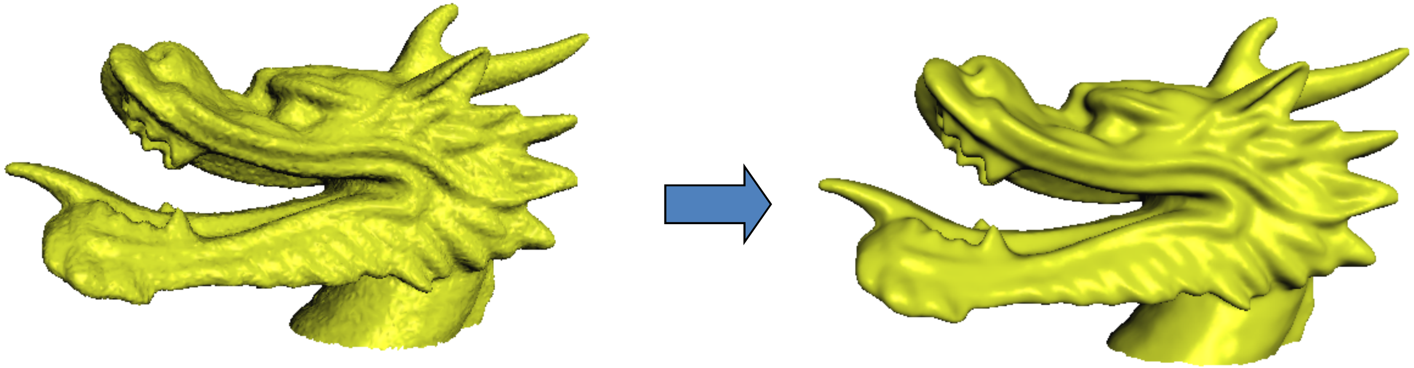
Other Terminologies

- Mesh denoising
- Mesh filtering
- Surface fairing
- Mesh improvement
  - Quality metrics

Mesh Smoothing

- Moving mesh vertices
  - Without changing connectivity
  - reduce curvature variation
- Used to
  - Reduce noise
  - Improve mesh triangle shape





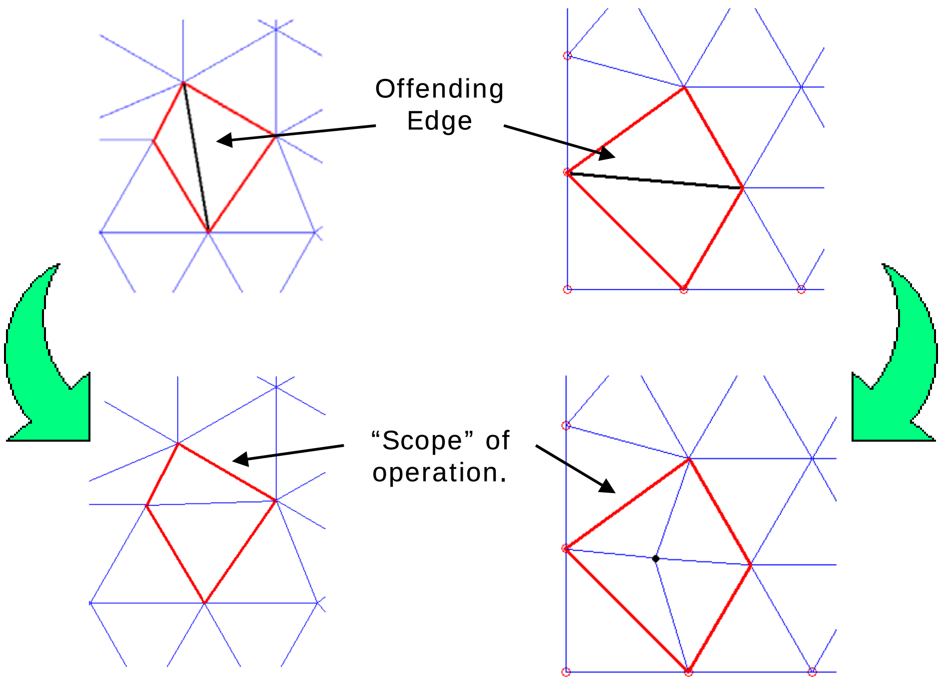
Mesh Improvement

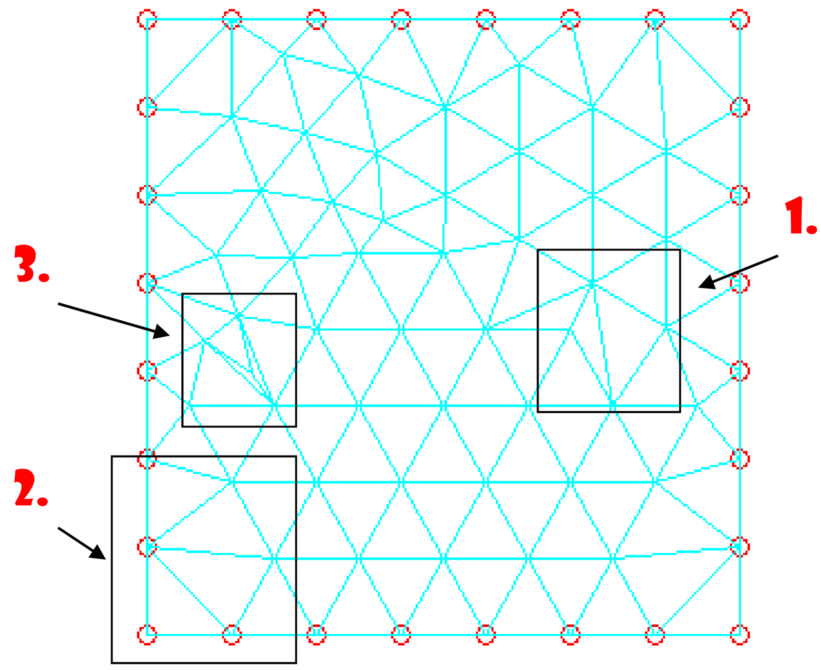
Topology changes

Local correction strategies

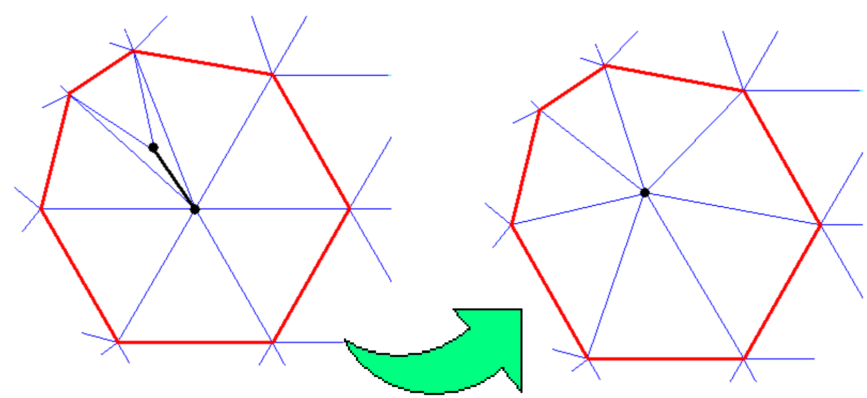
Algorithms

1. **Flip** an edge.
2. **Split** an edge.



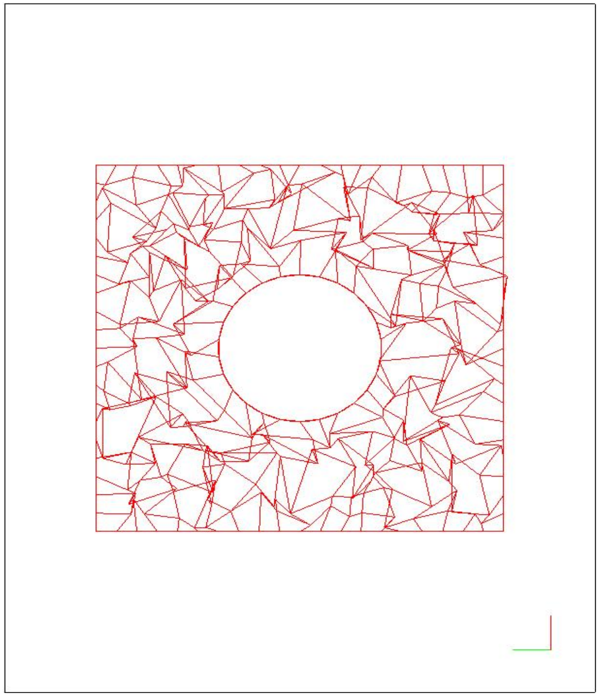
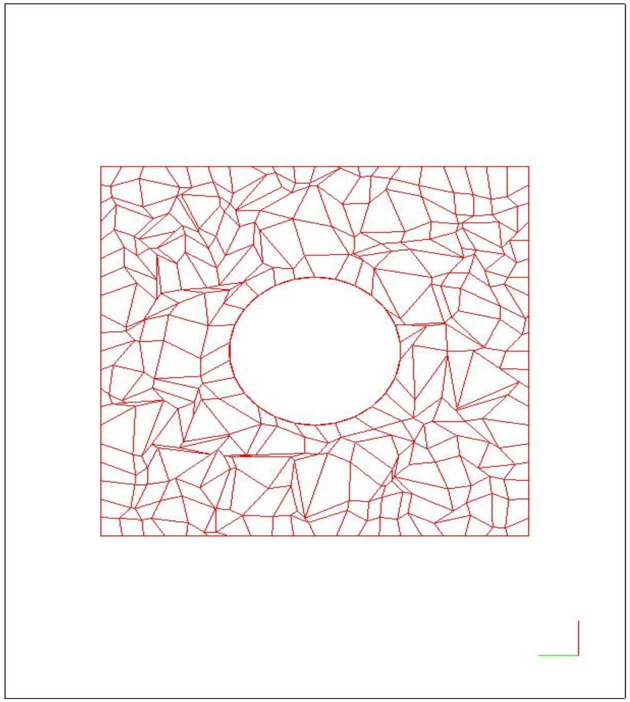
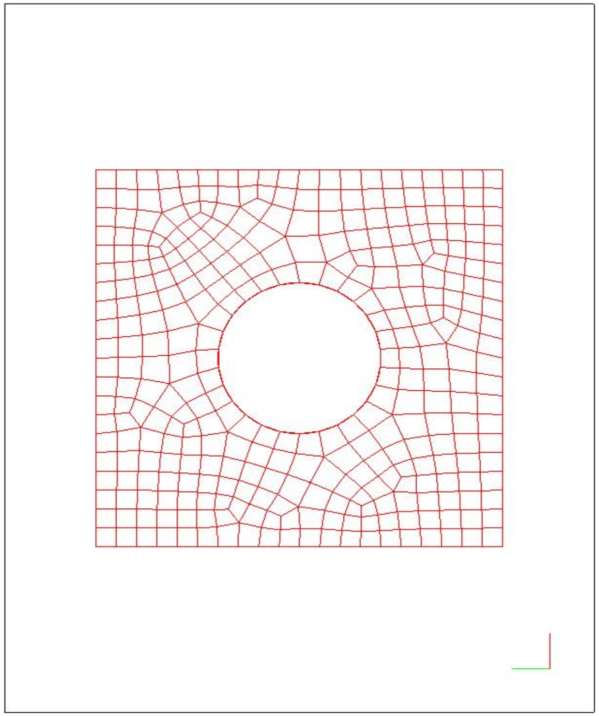


3. **Collapse** an edge.



Mesh Improvement

- Example

Classifications

- Surface
  - Triangular mesh
  - Volume
- Approaches
  - Laplacian
  - Normal smoothing
  - Bilateral filtering
  - Global approach
  - …

Smoothing

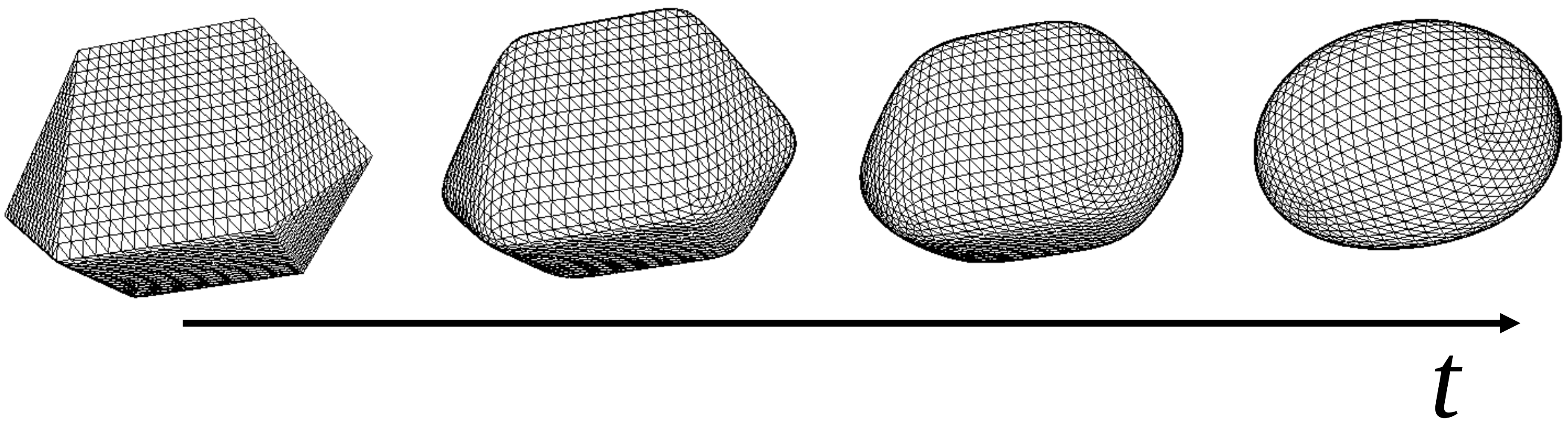
- Can apply not only to positions but also to any property assigned to vertices
  - Curvature, normals, physical properties (color, texture), ...
- Small number of iterations
- Shrinkage problem

General Smoothing Procedure

Shape evolution

$$\frac{\partial P}{\partial t} = \mathbf{F}(P)$$

$$P_{new} \leftarrow P_{old} + \Delta t\, \mathbf{F}(P_{old})$$



Explicit and Implicit Mesh Evolutions

Shape evolution $$\frac{\partial P}{\partial t} = \mathbf{F}(P)$$

$$M_{n+1} = M_n + \lambda \mathsf{L}(M_n) \quad \text{explicit scheme}$$

$$M_{n+1} = M_n + \lambda \mathsf{L}(M_{n+1}) \quad \text{implicit scheme}$$

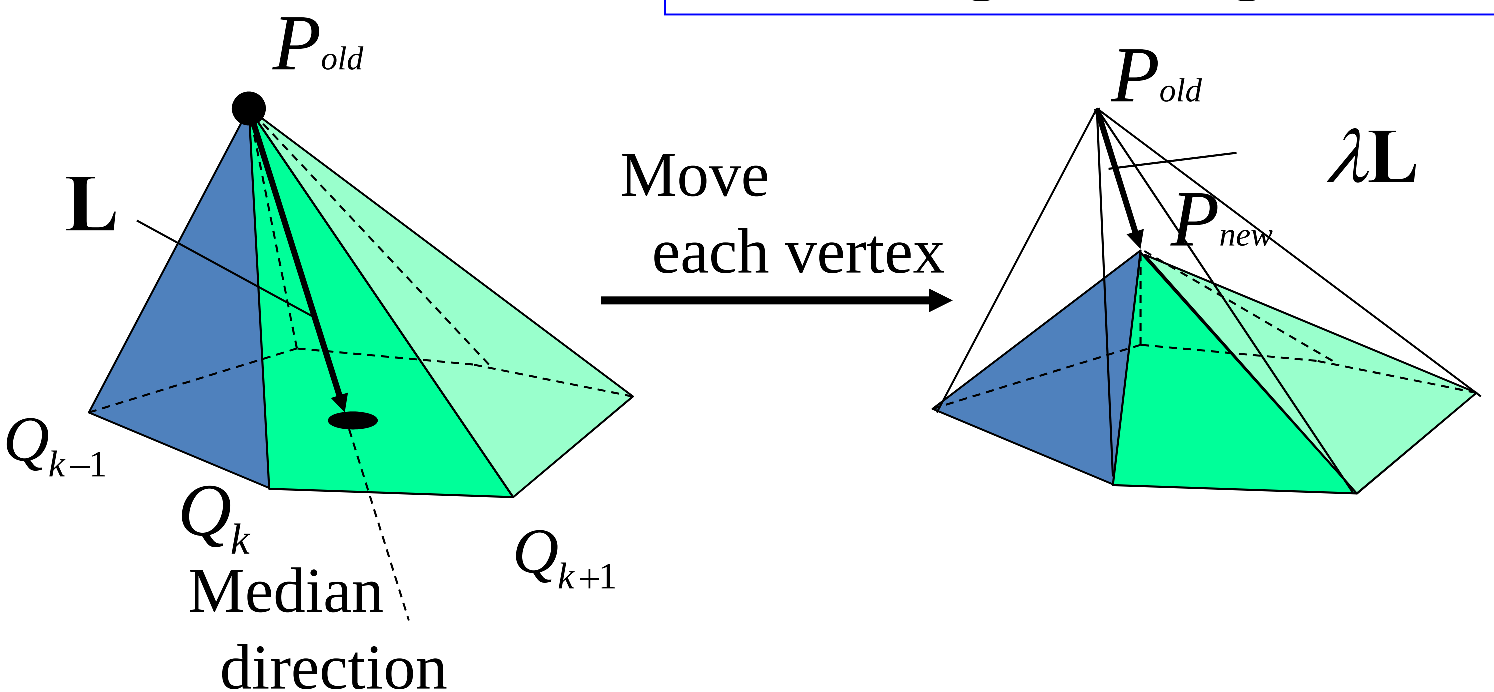
$$\Rightarrow \quad (I - \lambda \mathsf{L})\, M_{n+1} = M_n$$

Laplace Smoothing

Laplacian Smoothing Flow

$$P_{new} \leftarrow P_{old} + \lambda \mathbf{L}(P_{old})$$

Average of the vectors
to neighboring vertices



Umbrella Operator

- Umbrella operator

$$L(P) = \frac{1}{n}\sum_{i=1}^{n}\overrightarrow{PQ_i} = \frac{1}{n}\sum_{i=1}^{n}Q_i - P$$

- Weighted umbrella operator

$$L_w(P) = \frac{1}{\sum w_i}\sum_{i=1}^{n}w_i\overrightarrow{PQ_i} = \frac{1}{\sum w_i}\sum_{i=1}^{n}w_iQ_i - P$$

- Squared umbrella operator

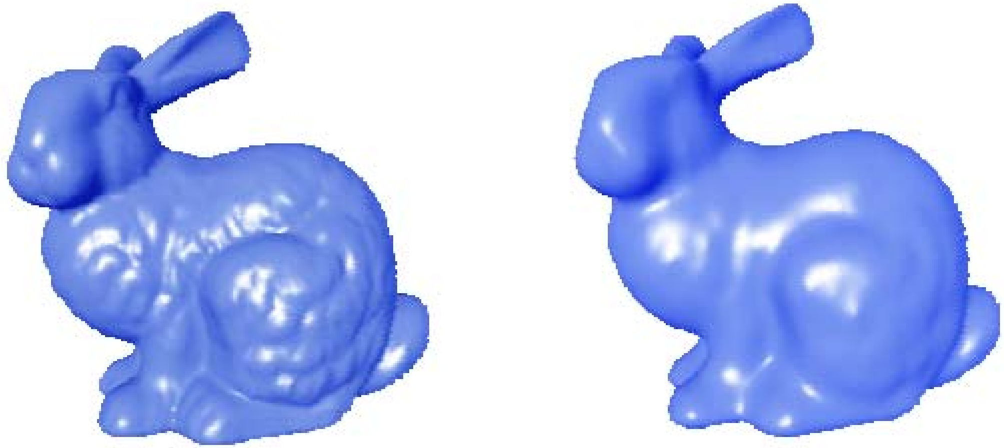
$$L_w^2(P)\frac{1}{\sum w_i}\sum_{i=1}^{n}w_iL(Q_i) - L(P)$$

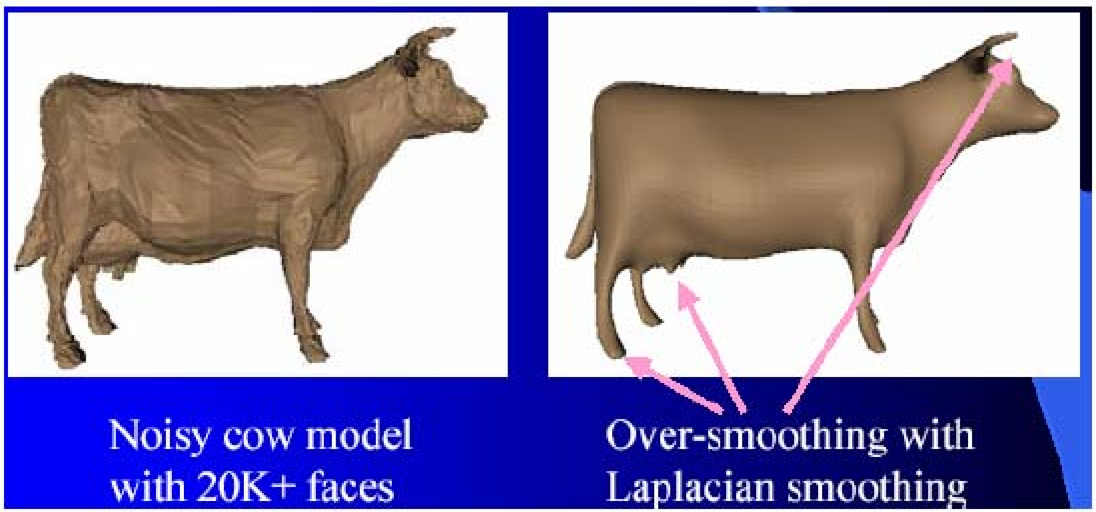
Laplacian Smoothing

$$P^{new} = P^{old} + \lambda L\left(P^{old}\right)$$

- Equivalent to box filter in signal processing
- Apply to all vertices on mesh
- Typically repeat several times
- Can describe as energy minimization
  - Energy = sum of squared edge lengths in mesh
  - Parameter $\lambda > 0$ controls convergence "speed"

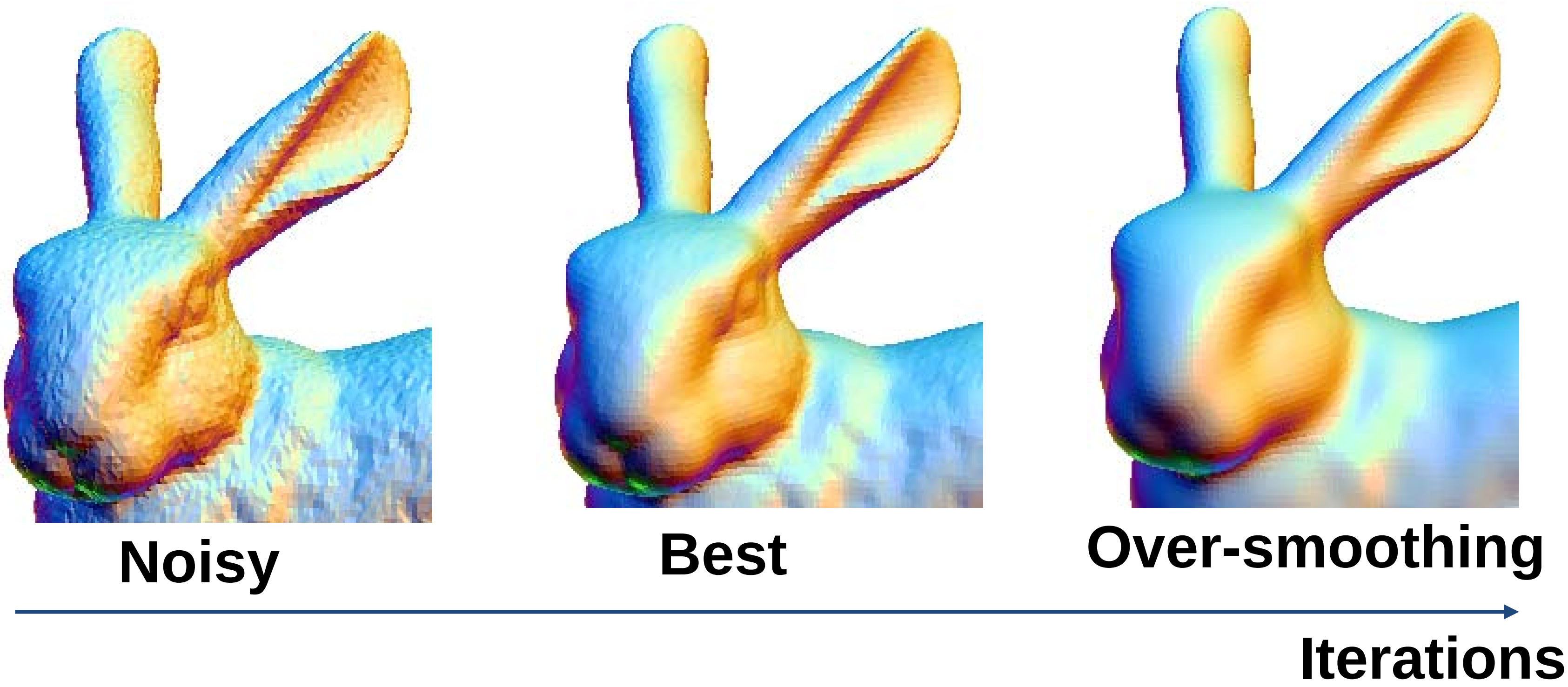
Laplacian Smoothing – Example





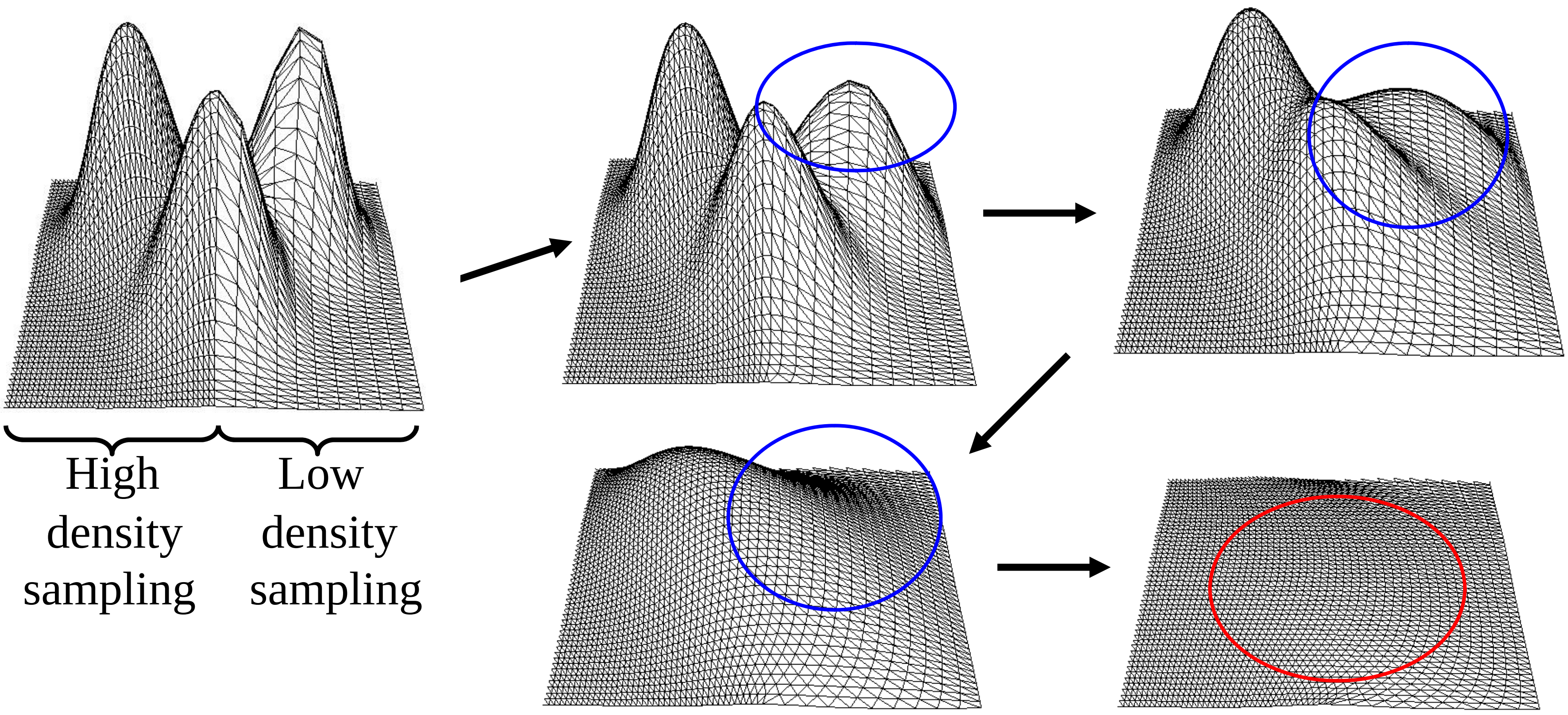
Problem of Over-smoothing

How to find appropriate λ and number of iterations



Properties of Laplacian Flow

- Increases mesh regularity
- Develops unnatural deformations



Shrinkage & Over-Smoothing

- Solutions
  - Projection back to surface
    - Keep original mesh – project each vertex to it
    - Project to approximating surface (e.g.quadric)
  - Volume preservation (scale model)
  - Add expansion term to filter
- Other extensions – add weights (reflecting mesh shape)

$$\Delta v_i = \frac{1}{\sum_{(i,j)} w_{ij}} \sum_{(i,j)} w_{ij} (v_j - v_i)$$

Improved Laplacian

- Laplacian

$$P^{new} = P^{old} + \lambda L\left(P^{old}\right)$$

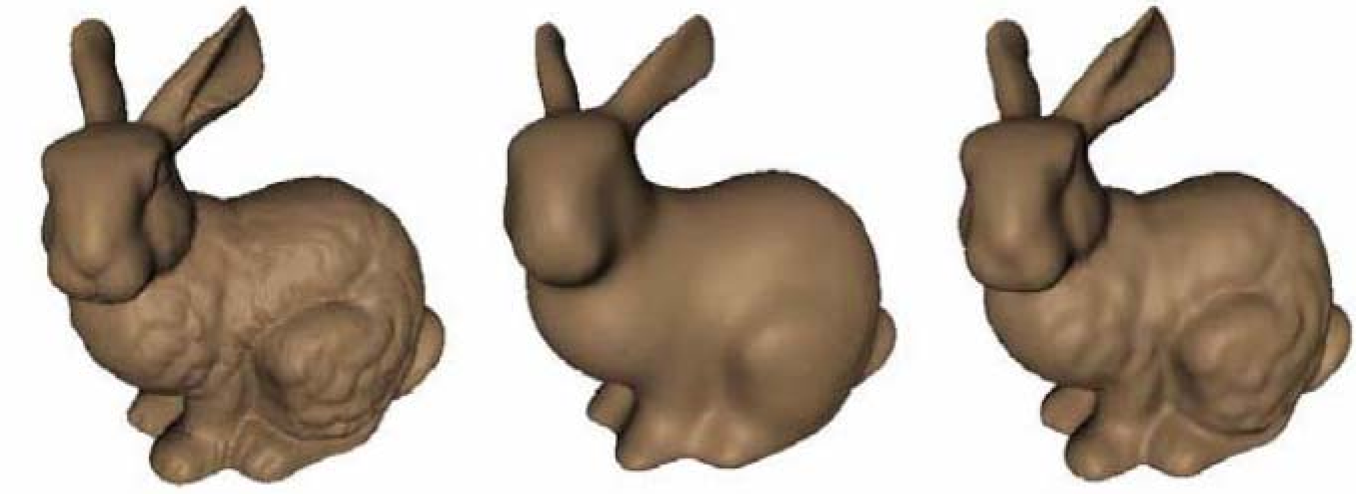
- Taubin'95
  - Laplacian + Expansion

$$P^{new} = P^{old} - (\mu - \lambda) L\left(P^{old}\right) - \mu\lambda L^2\left(P^{old}\right), \mu > \lambda > 0$$

- Bilaplacian
  - Special case of Taubin's

$$P^{new} = P^{old} + \lambda L^2\left(P^{old}\right)$$

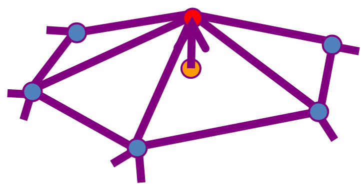
Comparison

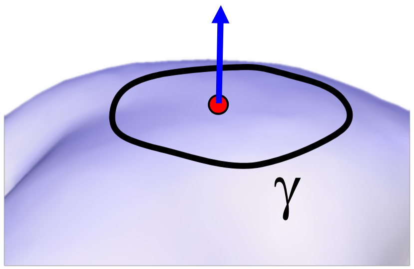


- Drawbacks
  - Slow
  - No stopping criteria

Discrete Mean Curvature Flow

Mean Curvature Flow





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

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

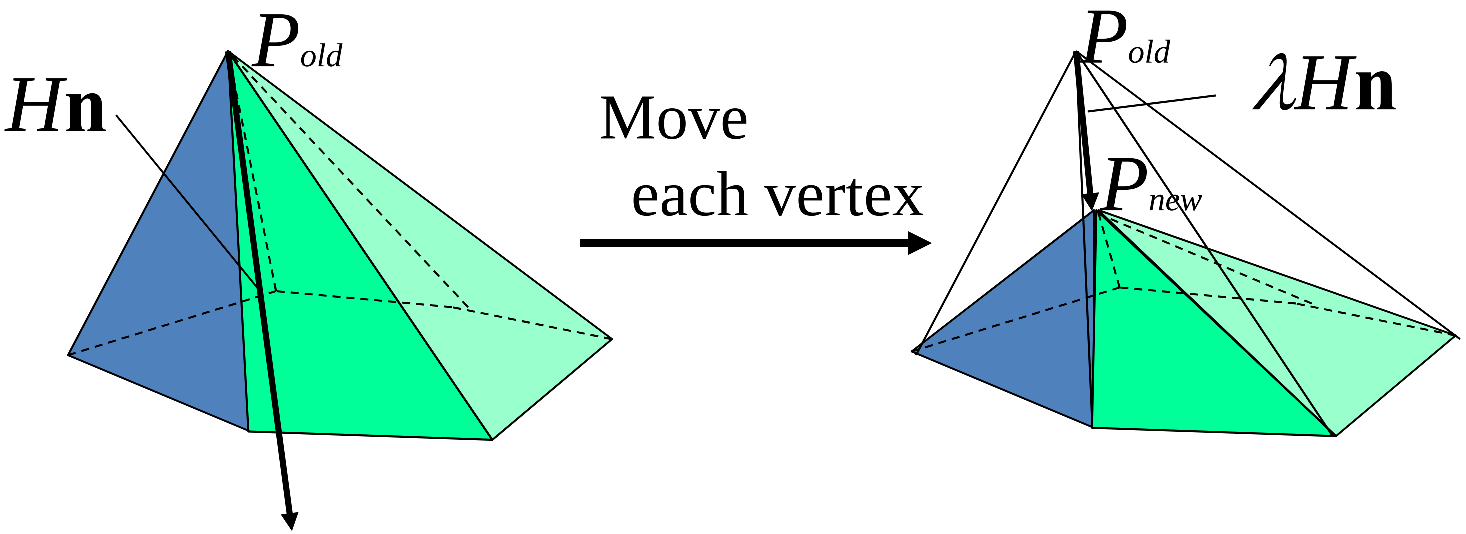
$$\lim_{len(\gamma) \to 0} \frac{1}{len(\gamma)} \int_{\mathbf{v} \in \gamma} \left(\mathbf{v}_\mathbf{i} - \mathbf{v} \right) ds = H(\mathbf{v}_\mathbf{i}) \mathbf{n}_\mathbf{i}$$

Discrete Mean Curvature Flow

$$P_{new} \leftarrow P_{old} + \lambda \boxed{H(P_{old})} \boxed{\mathbf{n}(P_{old})}$$

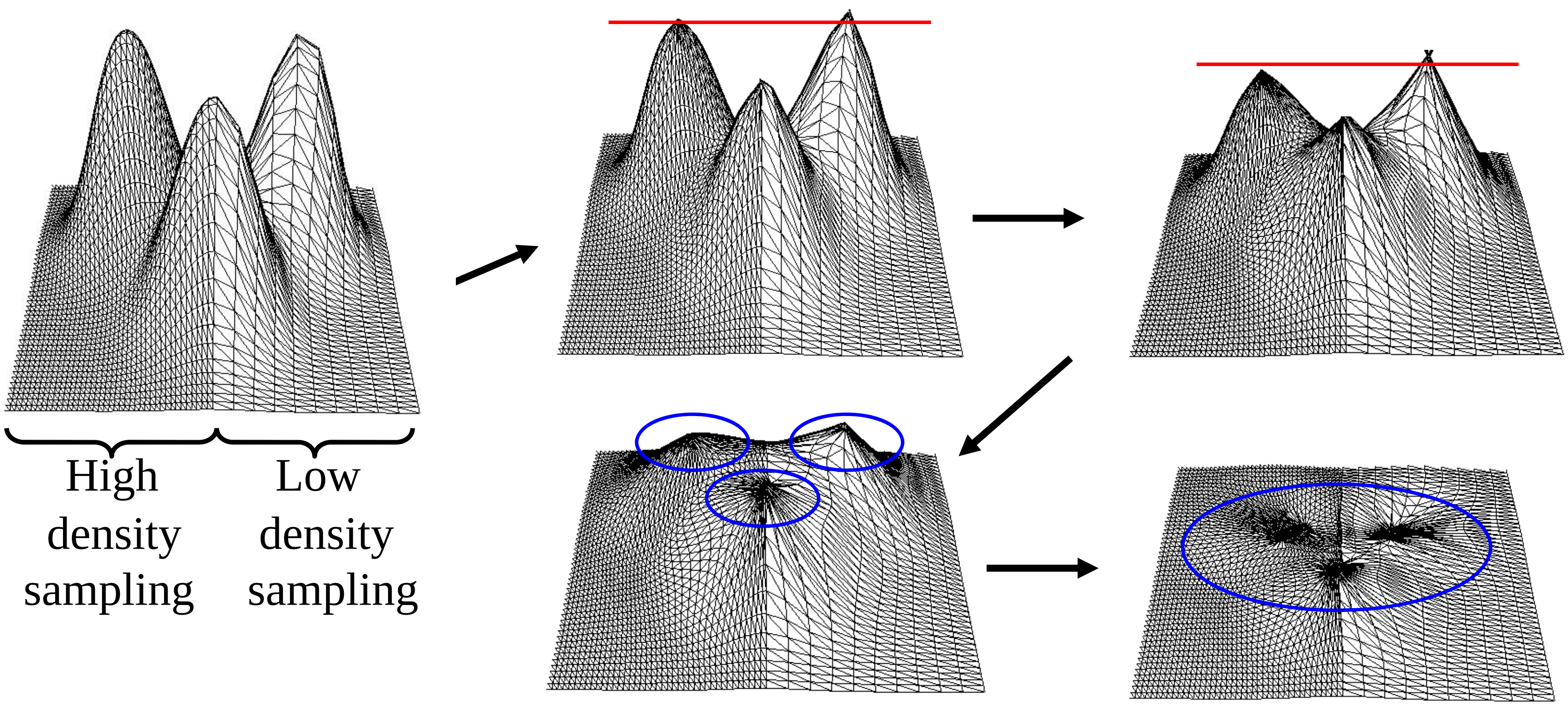
Speed = discrete mean curvature

Direction = normal



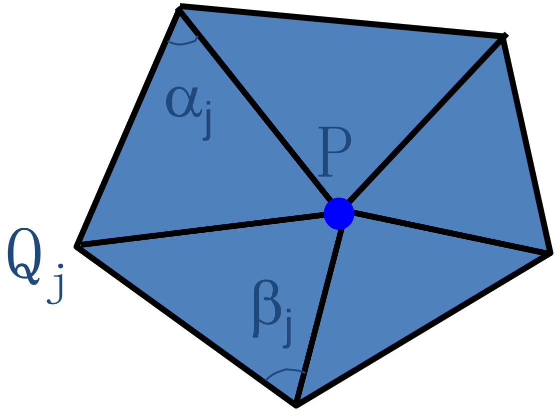
Properties of Mean Curvature Flow

- Increases mesh irregularity.
- Doesn't develop unnatural deformations



Discrete Mean Curvature

$$H\mathbf{n} = \frac{\nabla_P \mathrm{A}}{2\mathrm{A}}$$



$$H\mathbf{n} = \frac{1}{4\mathrm{A}} \sum_j (\cot \alpha_j + \cot \beta_j)(\mathbf{P} - \mathbf{Q}_j)$$

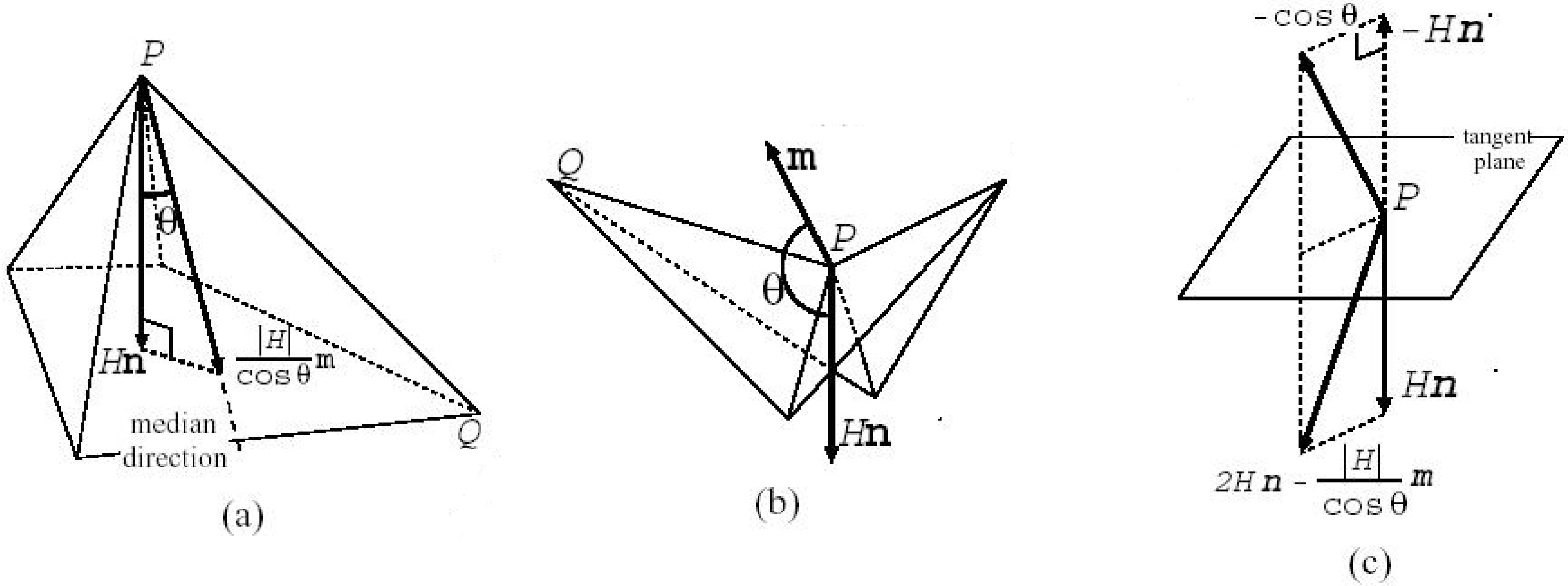
Modified Mean Curvature Flow

$$\mathbf{P}_{new} = \mathbf{P}_{old} + \lambda \mathbf{F}(\mathbf{P}_{old})$$

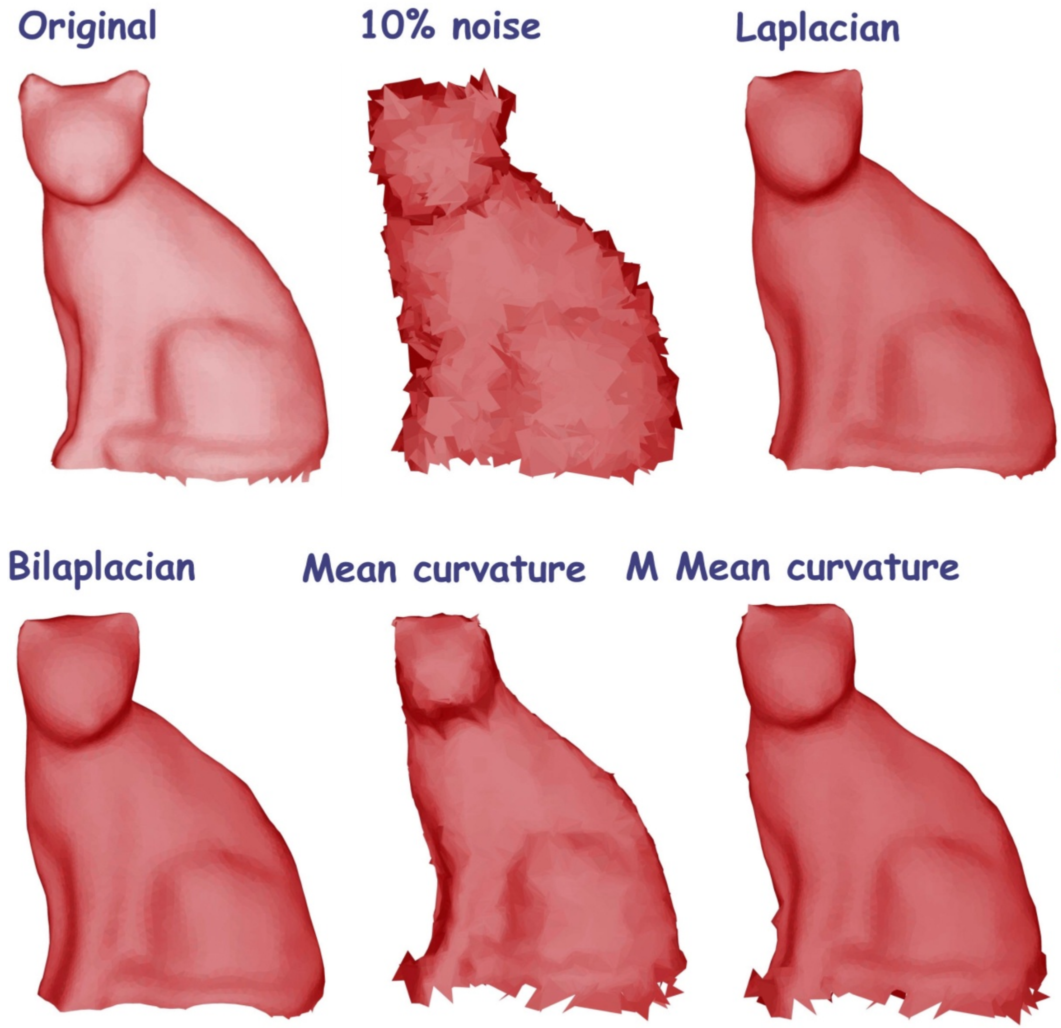
where

$$\mathbf{F}(\mathbf{P}) = \begin{cases} \frac{|H|\mathbf{m}}{cos\theta} & if cos\theta > e \\ 2\mathbf{Hn} - \frac{|H|\mathbf{m}}{cos\theta} & if cos\theta < -e \\ 0 & if |cos\theta| < e \end{cases}$$

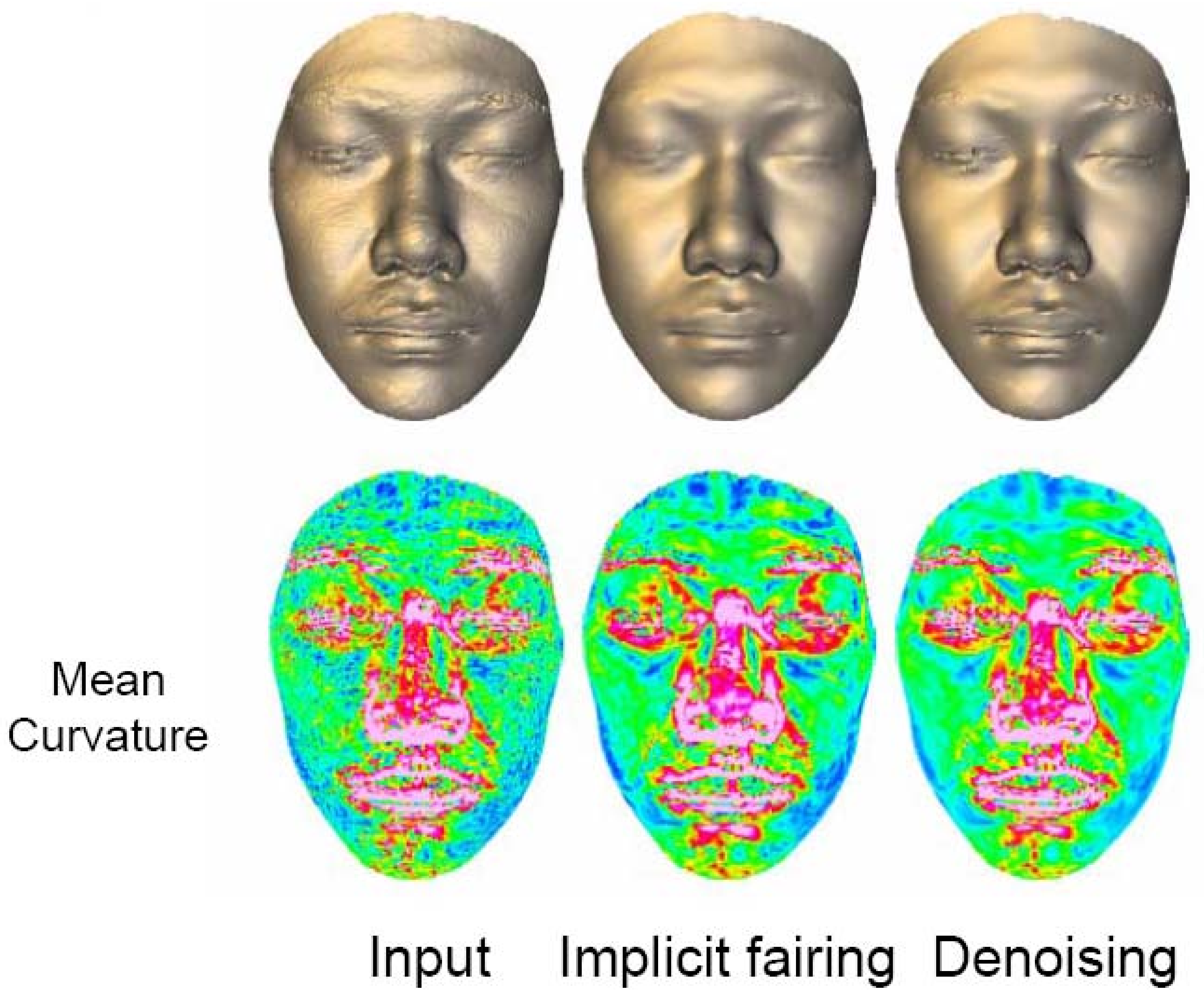
where $\mathbf{m} = \frac{\mathbf{U(P)}}{||\mathbf{U(P)}||}$ and $cos\theta = \frac{\mathbf{Hn.m}}{|H|}$



Comparisons



Smoothing Results

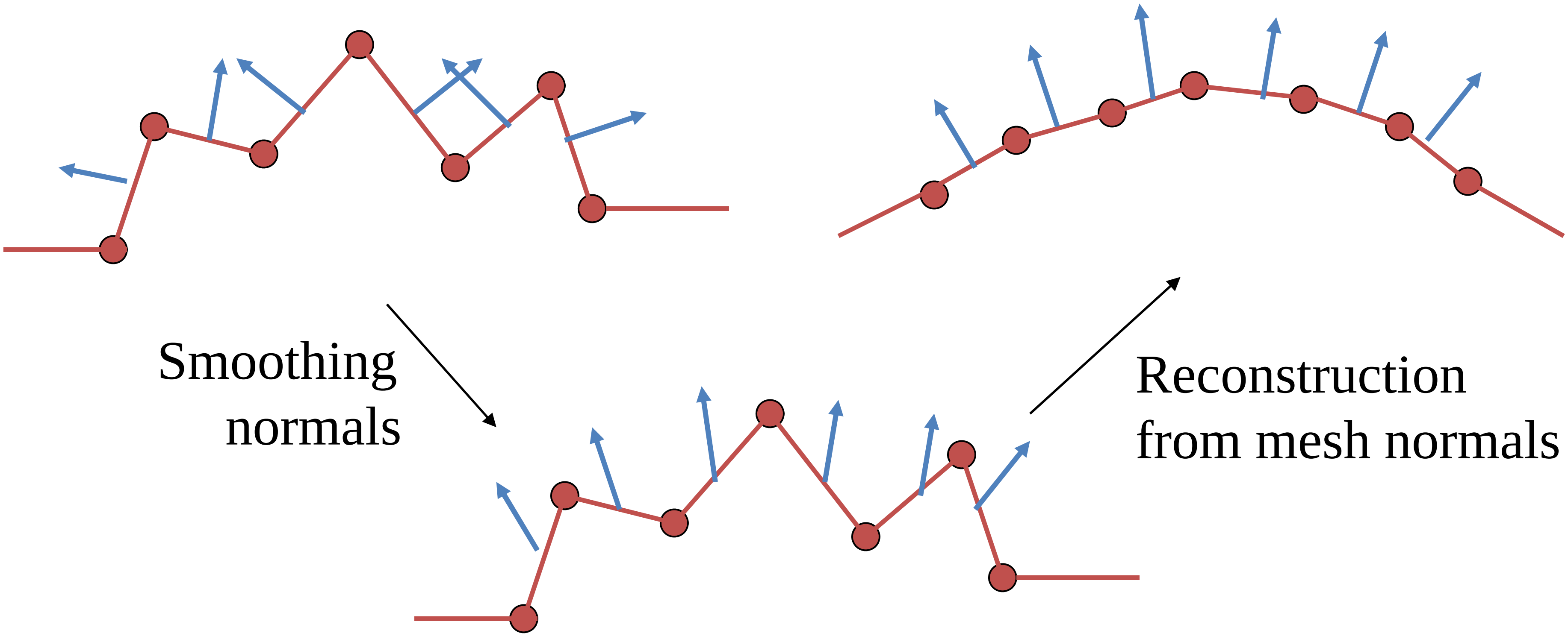


Normal based Smoothing

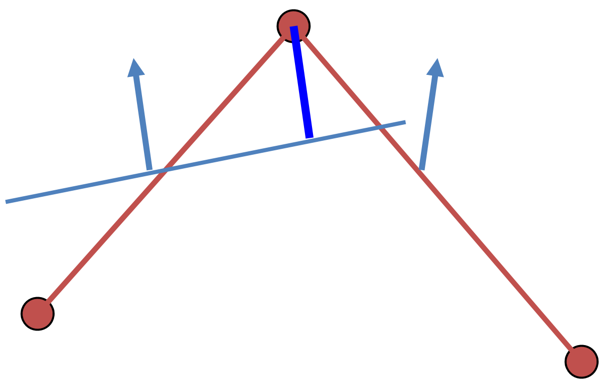
An Image Processing Approach

An Image Processing Approach

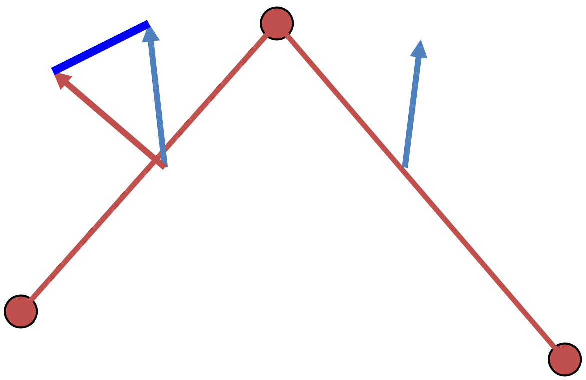
- Smooth mesh normals
- Reconstruct smoothed mesh from new normals



An Image Processing Approach



Distance error

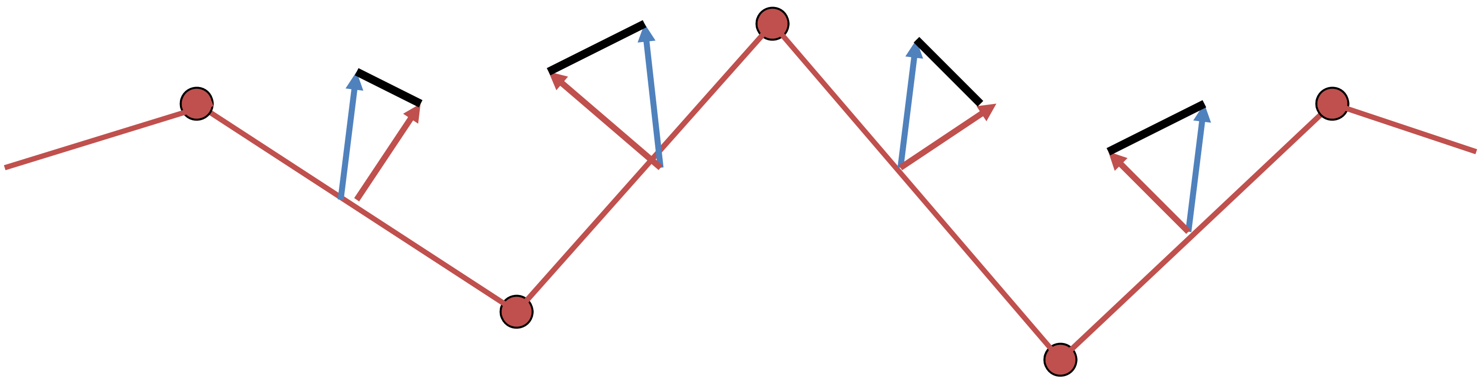


Derivative error

An Image Processing Approach

Reconstruction from smoothed normals

$$E_{\text{fit}}(M) = \sum_{\text{all triangles } T} \text{area}(T) \left| \mathbf{n}(T) - \mathbf{m}(T) \right|^2$$



$$E_{\text{fit}}(M) \to \min$$

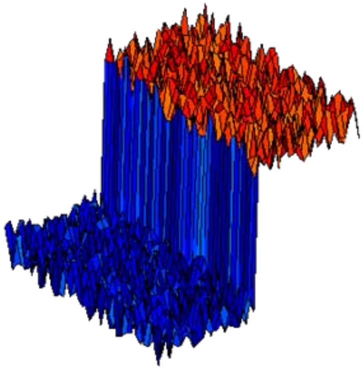
Bilateral Mesh Denoising

Siggraph 2003

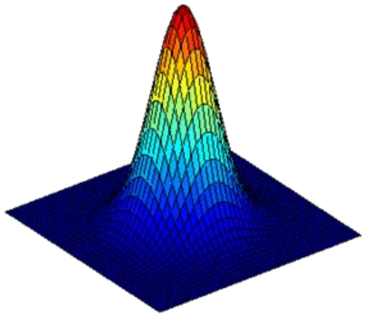
Gaussian Filtering

- Gaussian filter

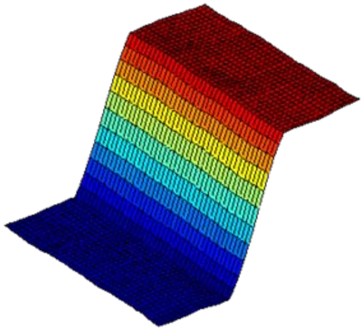
$$I'(u) = \sum_{p \in N(u)} e^{-\frac{\|u-p\|^2}{2\sigma^2}} I(p)$$



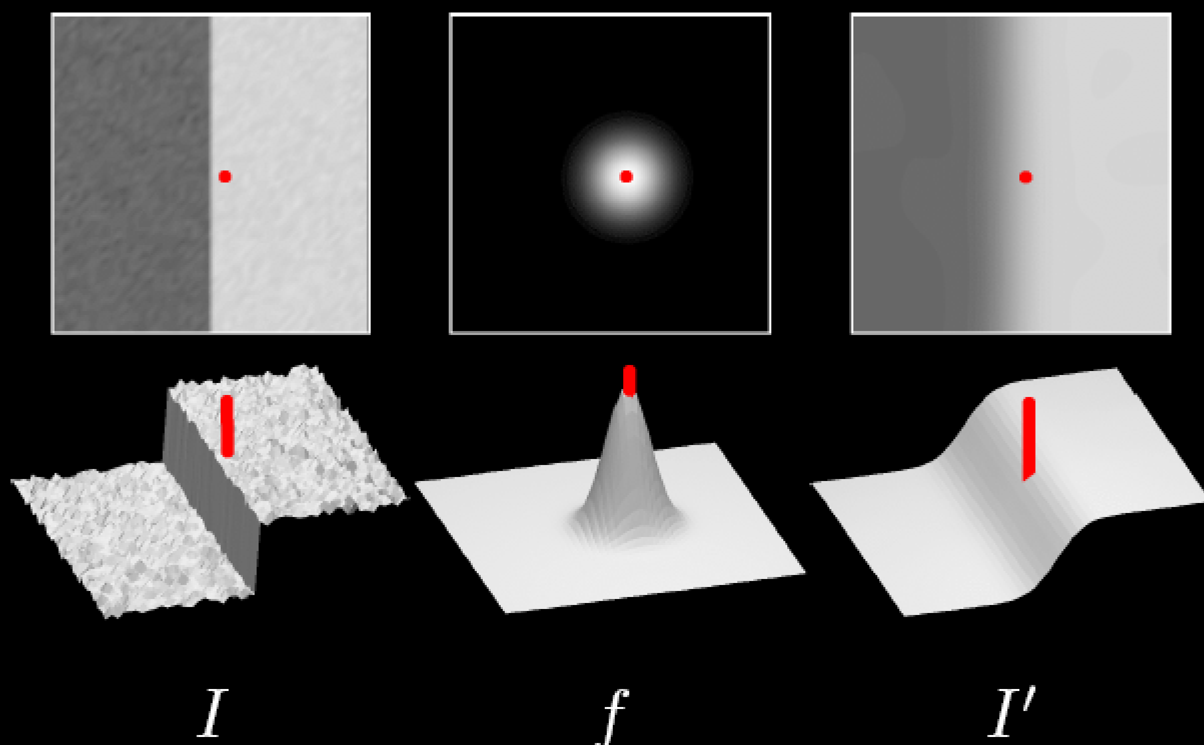
*



=



$$I'_s = \sum_p \overbrace{I(p)}^{\text{image}} \; \overbrace{f(s-p)}^{\text{spatial}}$$



Bilateral Filtering

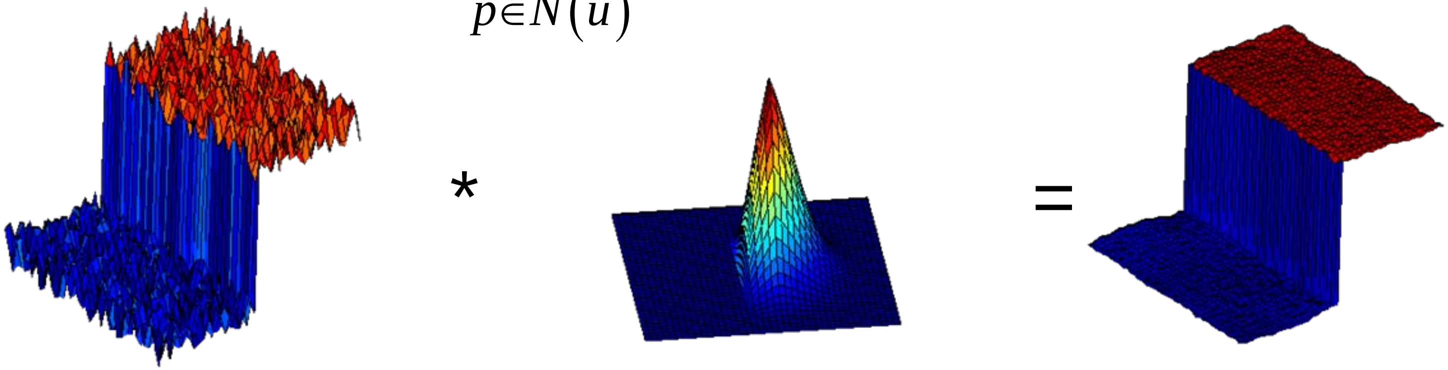
- Bilateral filter

$$I'(u) = \frac{\sum_{p \in N(u)} e^{-\frac{\|u-p\|^2}{2\sigma_c^2}} e^{-\frac{|I(u)-I(p)|}{2\sigma_s^2}} I(p)}{\sum_{p \in N(u)} e^{-\frac{\|u-p\|^2}{2\sigma_c^2}} e^{-\frac{|I(u)-I(p)|}{2\sigma_s^2}}}$$

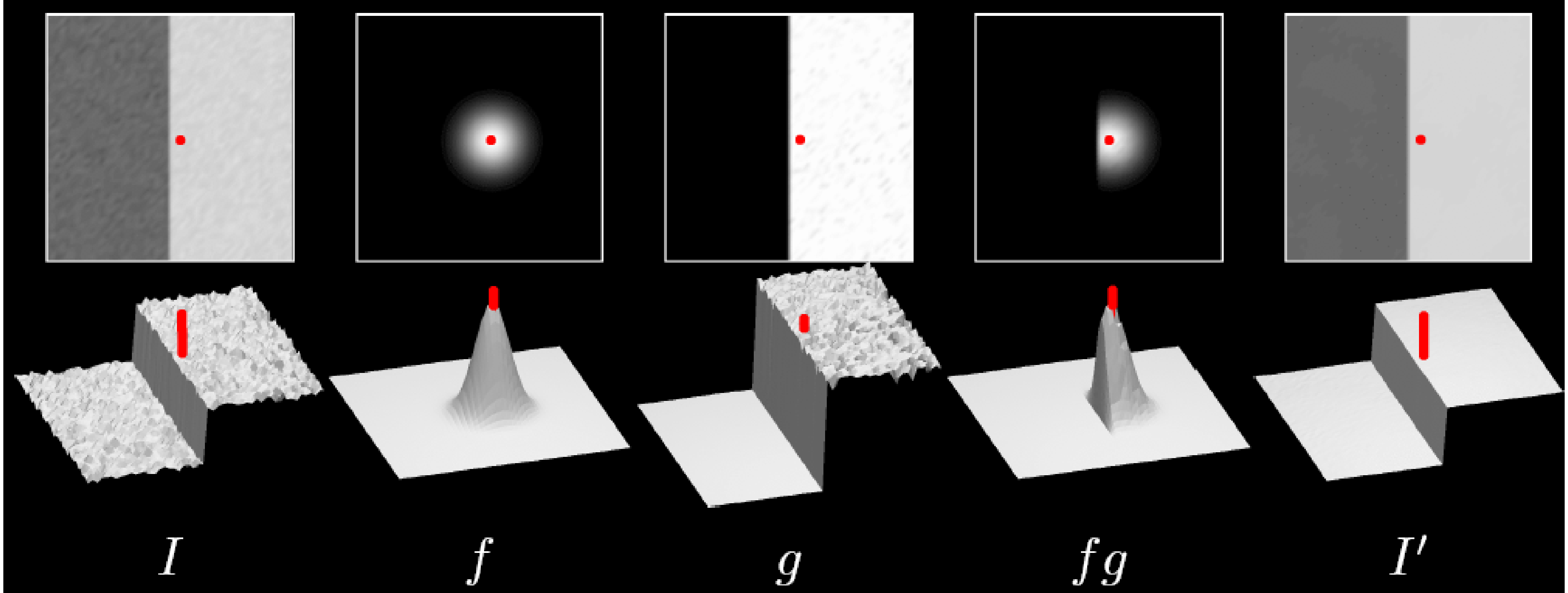
Denoise

Feature preserving

Normalization

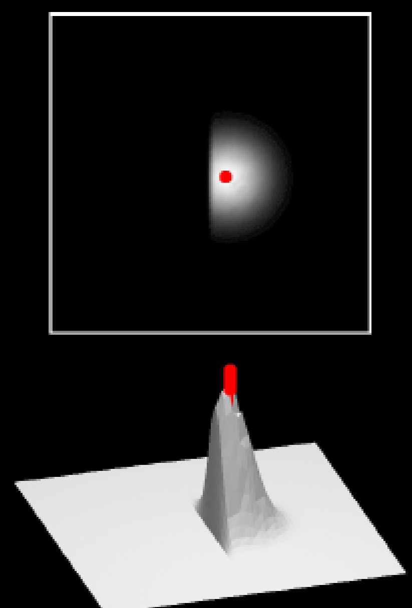


$$I'_s = \frac{1}{k_s} \sum_p \overbrace{I(p)}^{\text{image}} \overbrace{f(s-p)}^{\text{spatial}} \overbrace{g(I_s - I_p)}^{\text{influence}}$$



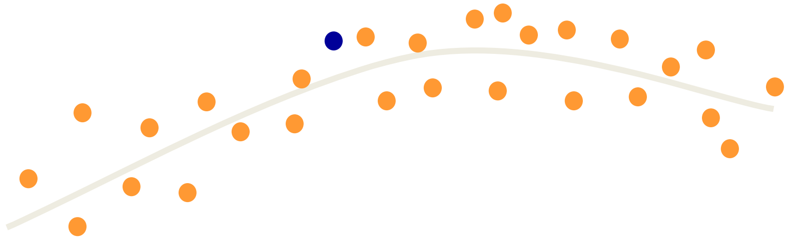
$$I'_s = \frac{1}{k_s} \sum_p \overbrace{I(p)}^{\text{image}} \; \overbrace{f(s-p)}^{\text{spatial}} \; \overbrace{g(I_s - I_p)}^{\text{influence}}$$

$$k_s = \sum_p f(s-p)\, g(I_s - I_p)$$



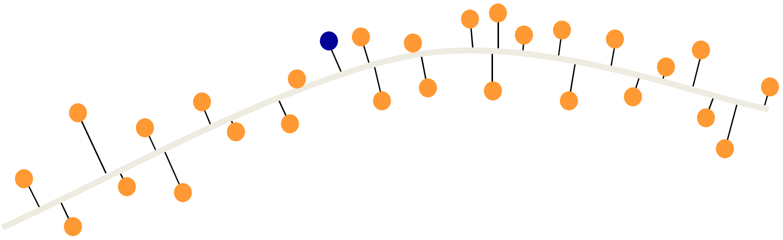
Bilateral filtering of meshes

Bilateral filtering of meshes



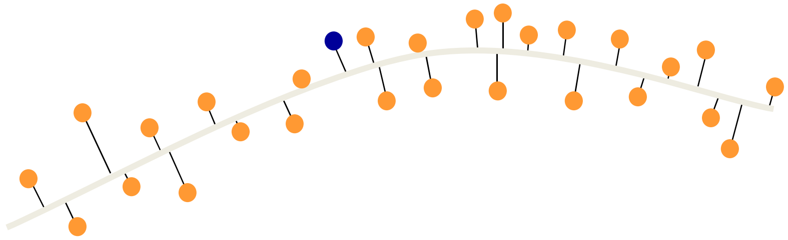
Bilateral filtering of meshes

- Height above surface is equivalent to the gray level values in images



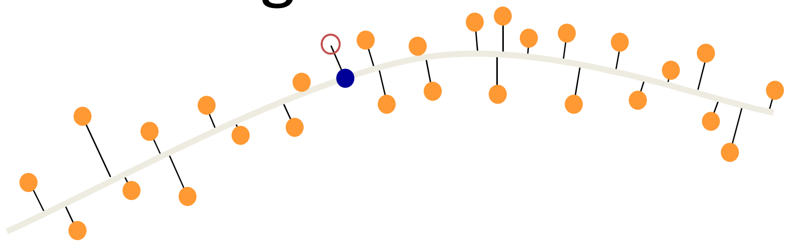
Bilateral filtering of meshes

- Height above surface is equivalent to the gray level values in images
- Apply the bilateral filter to heights



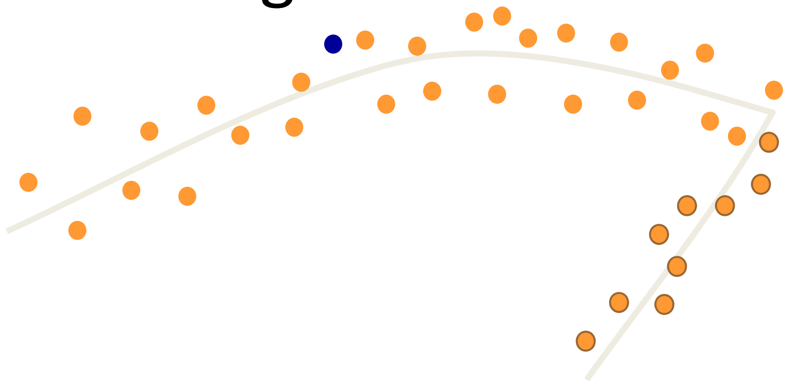
Bilateral filtering of meshes

- Height above surface is equivalent to the gray level values in images
- Apply the bilateral filter to heights
- Move the vertex to its new height



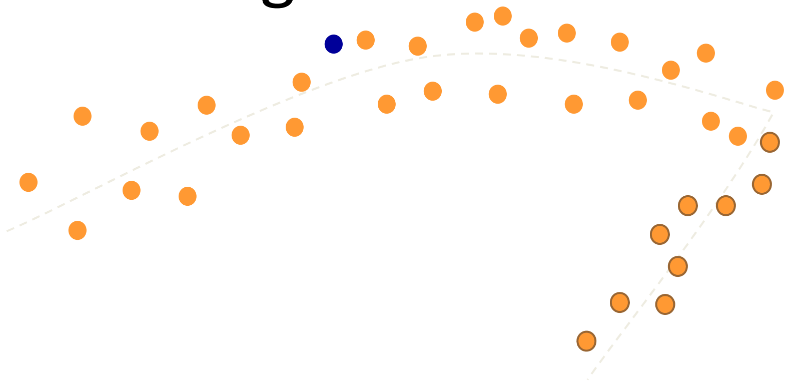
Bilateral filtering of meshes

- Height above surface is equivalent to the gray level values in images
- Apply the bilateral filter to heights
- Move the vertex to its new height
- In practice:
  - Sharp features



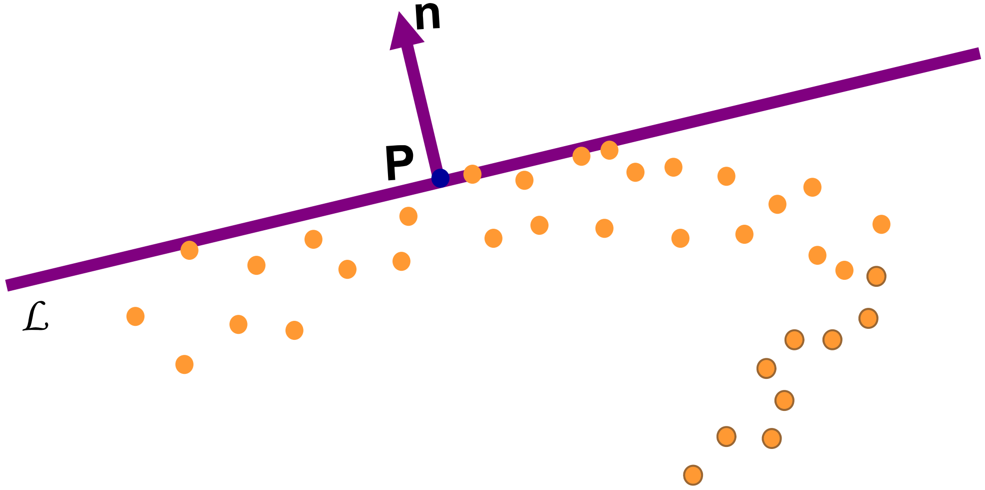
Bilateral filtering of meshes

- Height above surface is equivalent to the gray level values in images
- Apply the bilateral filter to heights
- Move the vertex to its new height
- In practice:
  - Sharp features
  - The noise-free surface is unknown



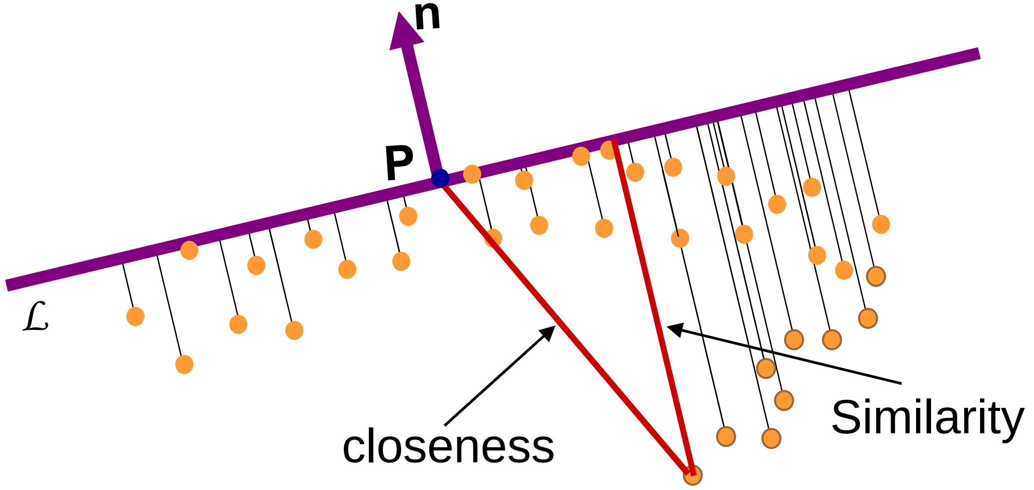
Solution

- A plane that passes through the point is the estimator to the smooth surface
- Plane $\mathcal{L}$=(**p**,**n**)



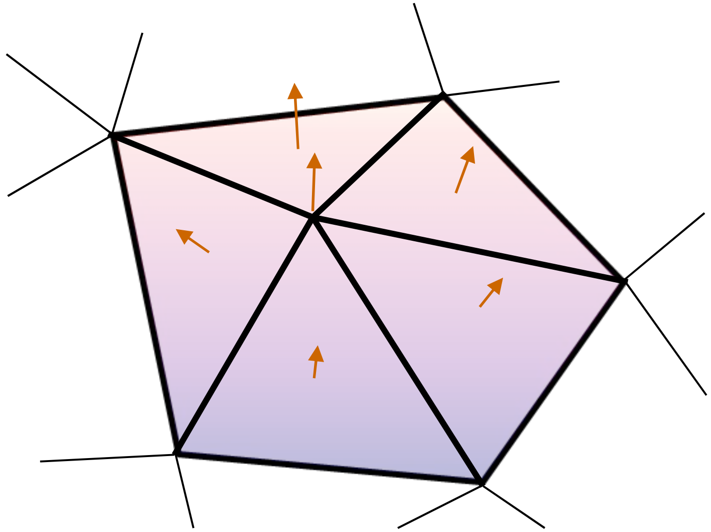
Solution

- A plane that passes through the point is the estimator to the smooth surface
- Plane $\mathcal{L}$=(**p**,**n**)



Computing the plane

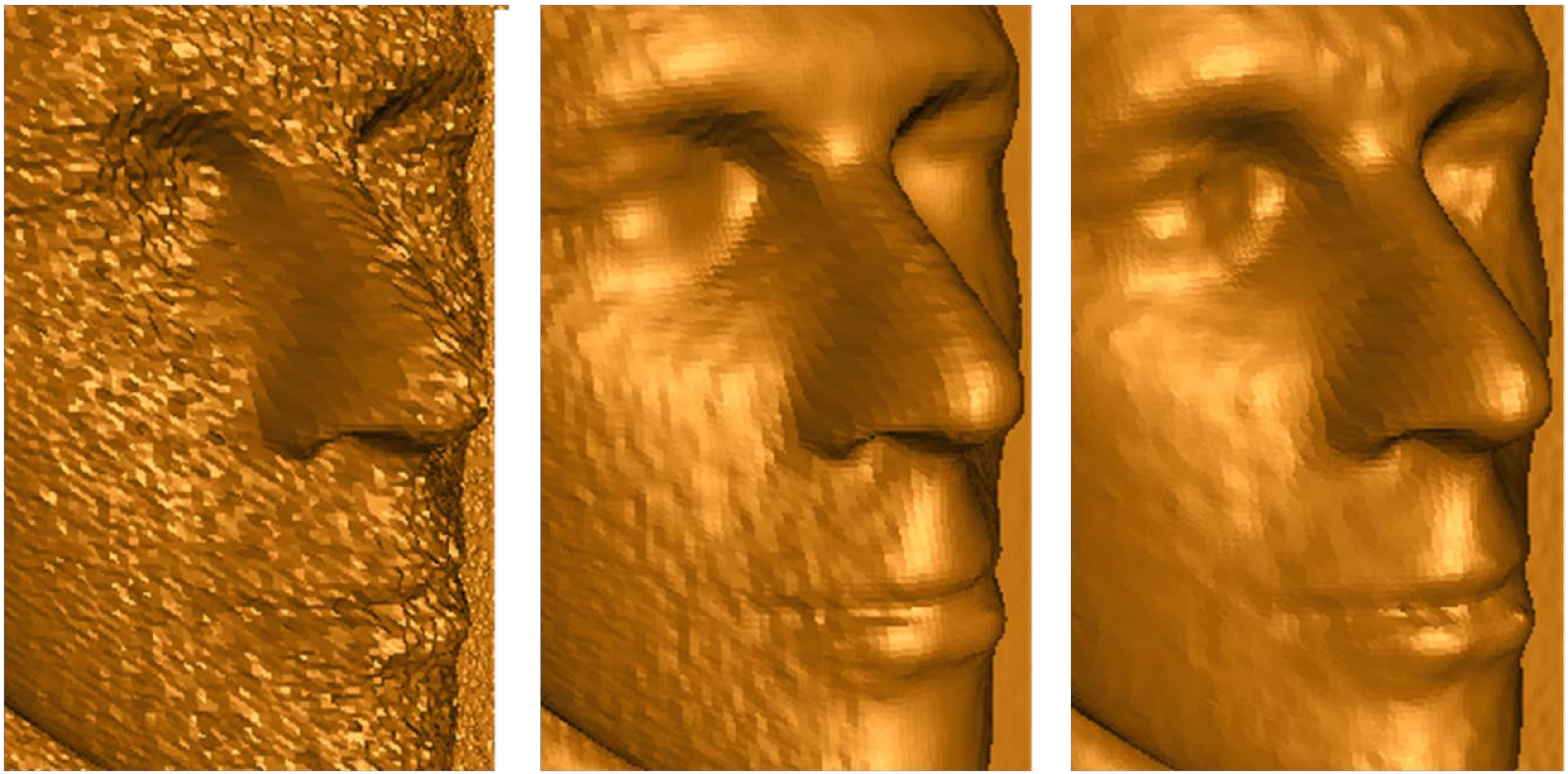
- The approximating plane should be:
  - A good approximation to the surface
  - Preserve features
- Average of the normal to faces in the 1-ring neighborhood



Parame

- The two parameters to the weight function: σ_c, σ_s
  - Interactively select a point ***p*** and the neighborhood radius ρ
  - $\sigma_c = {}^1/_2\,\rho$
  - σ_s = stdv(Nbhd($\boldsymbol{p}$, ρ))
- Number of Iterations

Results



Source

Anisotropic denoising of height fields - Desburn ’00

Bilateral mesh denoising

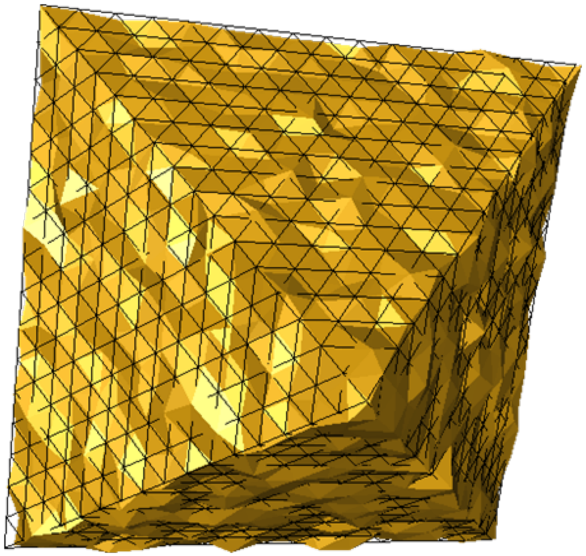
Results

Source

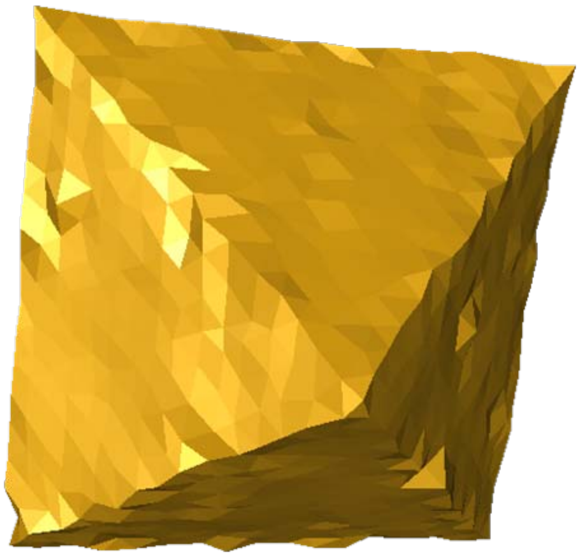
Anisotropic Geometric Diffusion in Surface Processing - Clarenz '00

Bilateral mesh denoising

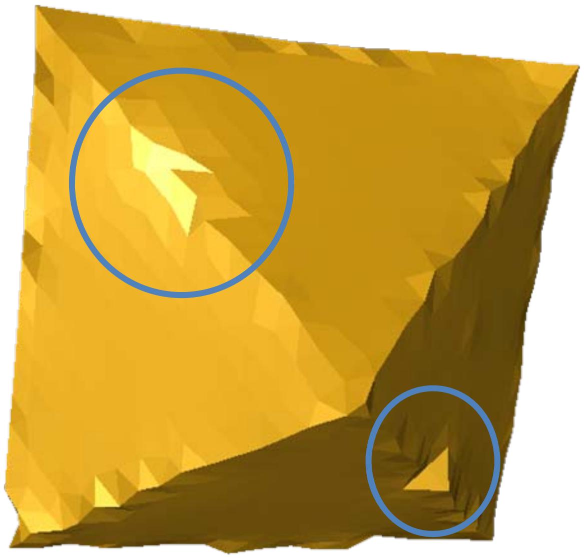
Results



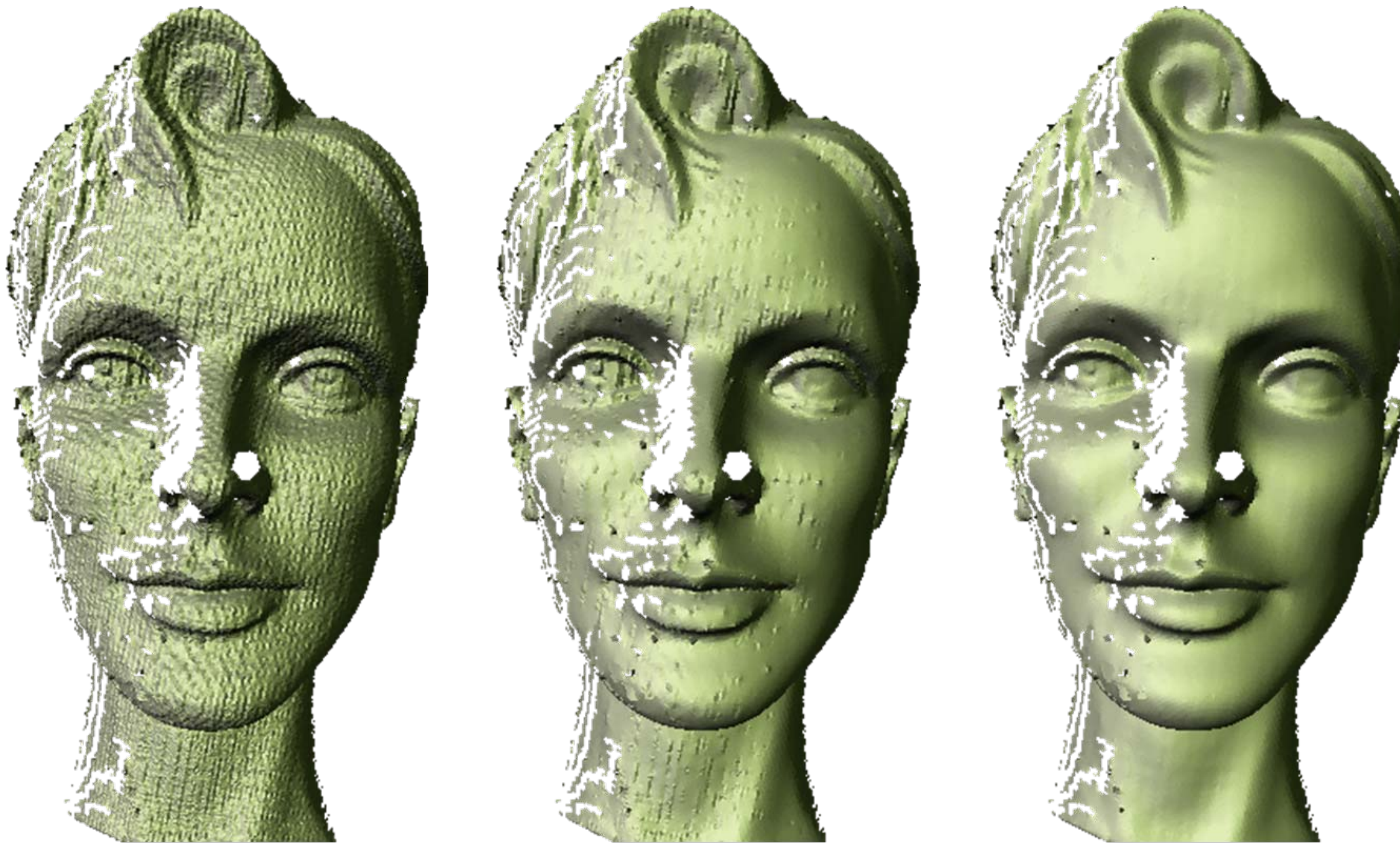
Source



Two iterations



Five iterations

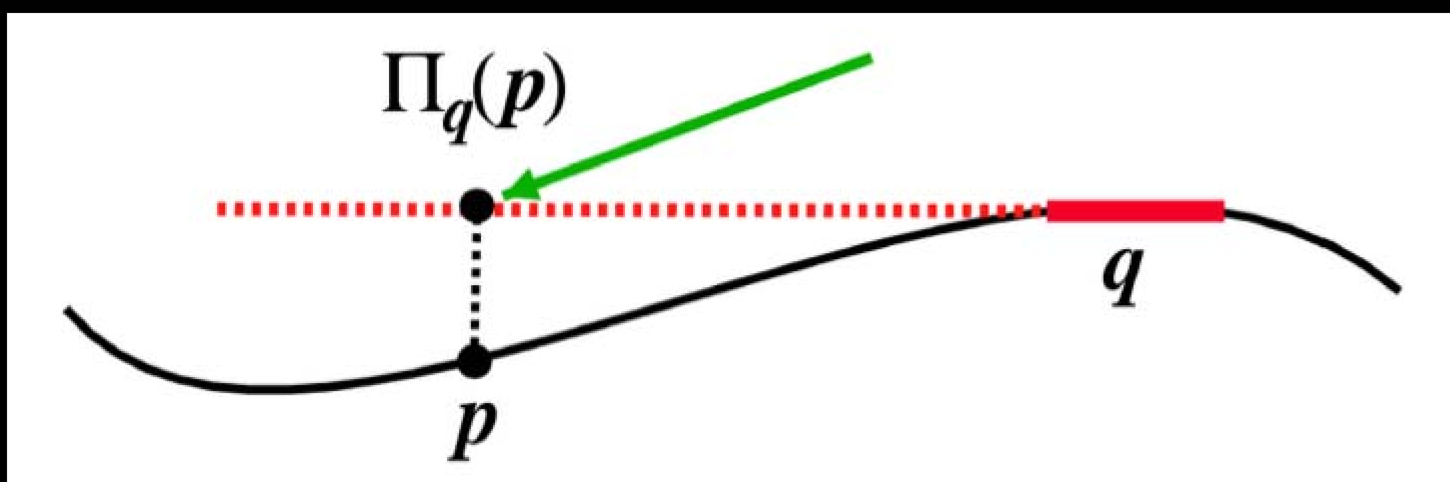


Non-Iterative, Feature Preserving Mesh Smoothing

Siggraph 2003

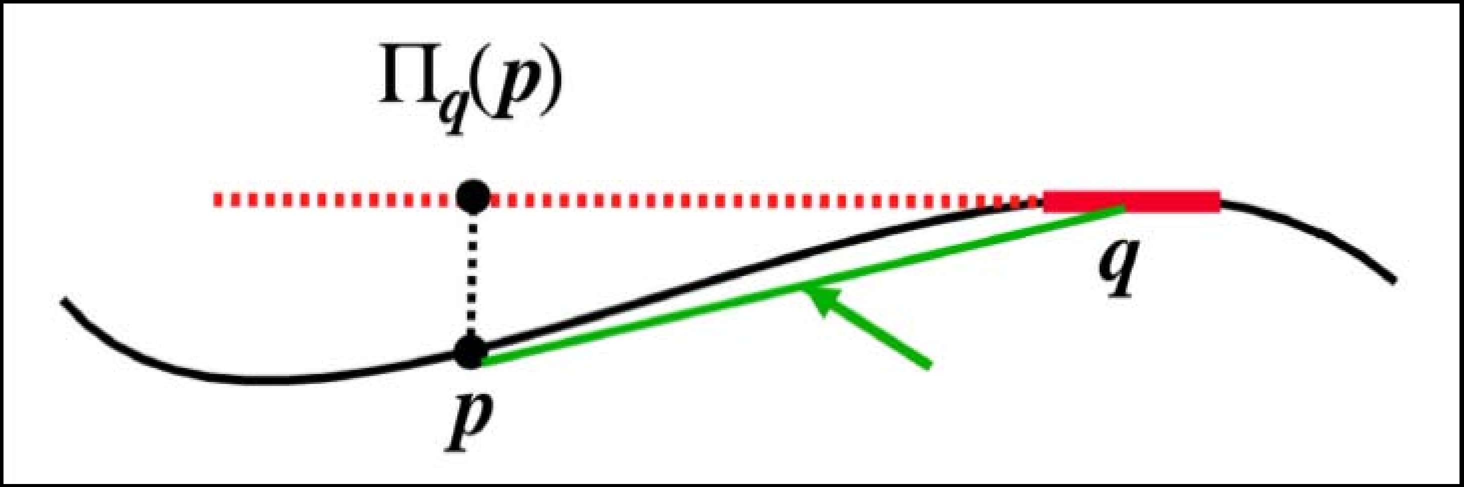
Prediction

$$p' = \frac{1}{k(p)} \sum_{q \in S} \overbrace{\Pi_q(p)}^{\text{prediction}} \; \overbrace{f(||c_q - p||)}^{\text{spatial}} \; \overbrace{g(||\Pi_q(p) - p||)}^{\text{influence}} \; \overbrace{a_q}^{\text{area}}$$



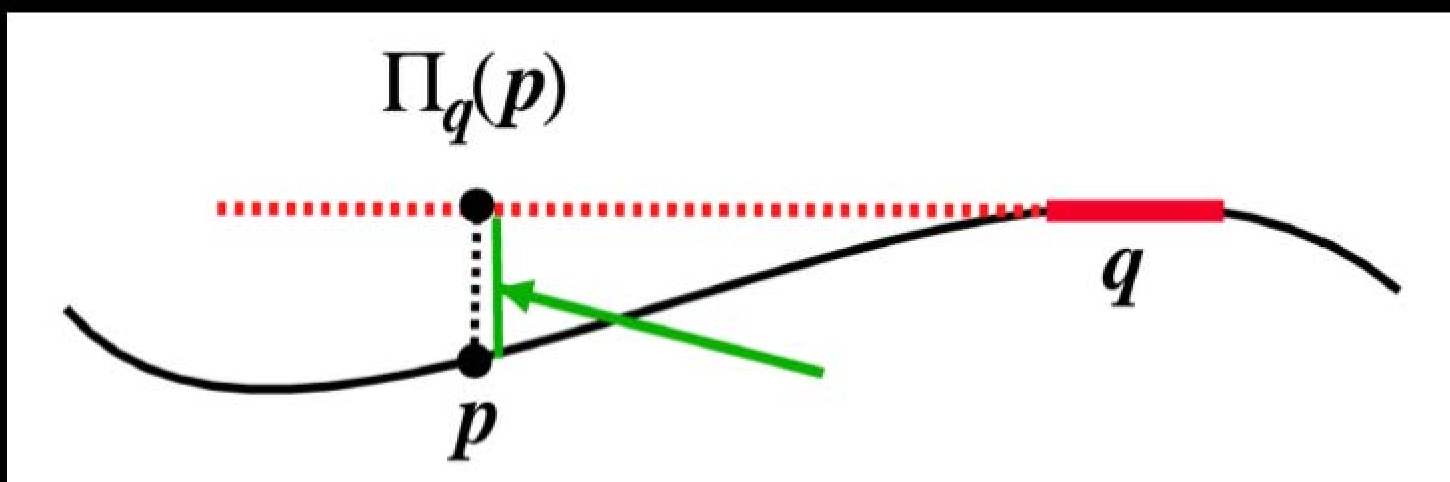
Spatial

$$p' = \frac{1}{k(p)} \sum_{q \in S} \overbrace{\Pi_q(p)}^{\text{prediction}} \; \overbrace{f(||c_q - p||)}^{\text{spatial}} \; \overbrace{g(||\Pi_q(p) - p||)}^{\text{influence}} \; \overbrace{a_q}^{\text{area}}$$



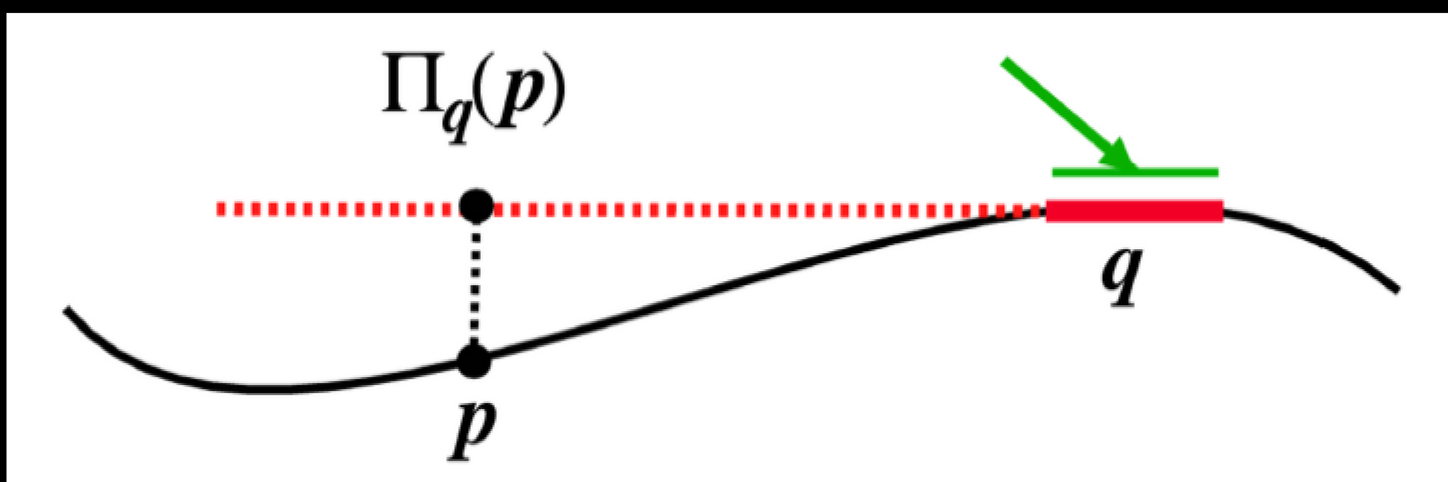
Influence

$$p' = \frac{1}{k(p)} \sum_{q\in S} \overbrace{\Pi_q(p)}^{\text{prediction}} \; \overbrace{f(||c_q - p||)}^{\text{spatial}} \; \overbrace{g(||\Pi_q(p) - p||)}^{\text{influence}} \; \overbrace{a_q}^{\text{area}}$$



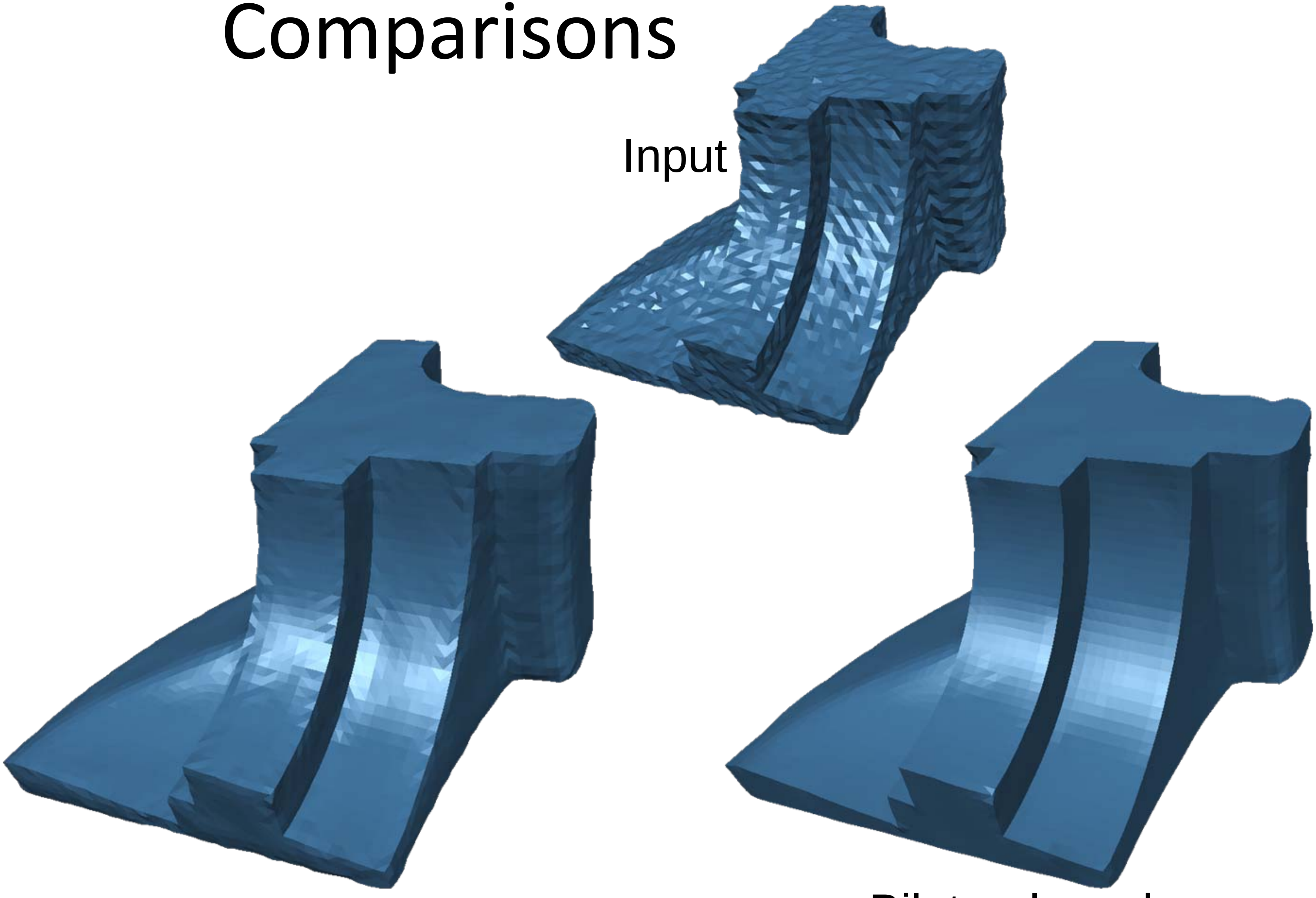
Area

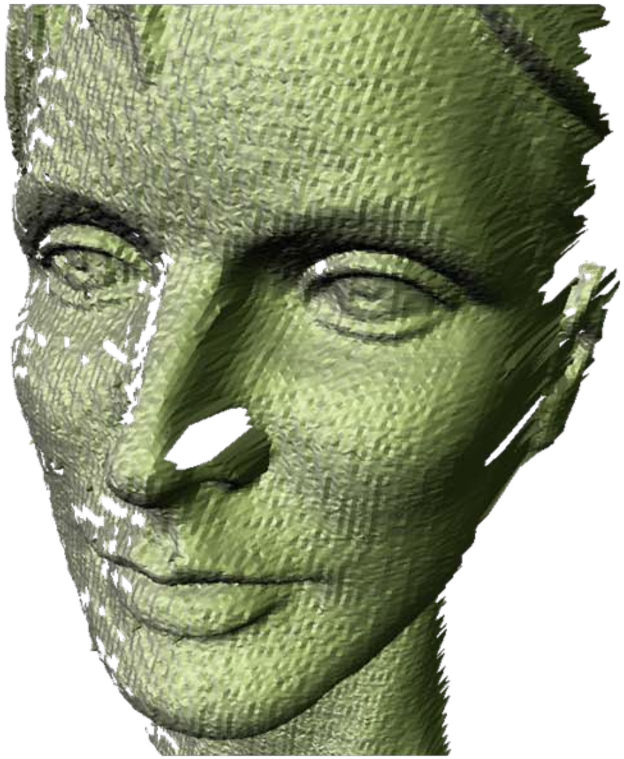
$$p' = \frac{1}{k(p)} \sum_{q \in S} \overbrace{\Pi_q(p)}^{\text{prediction}} \; \overbrace{f(||c_q - p||)}^{\text{spatial}} \; \overbrace{g(||\Pi_q(p) - p||)}^{\text{influence}} \; \overbrace{a_q}^{\text{area}}$$



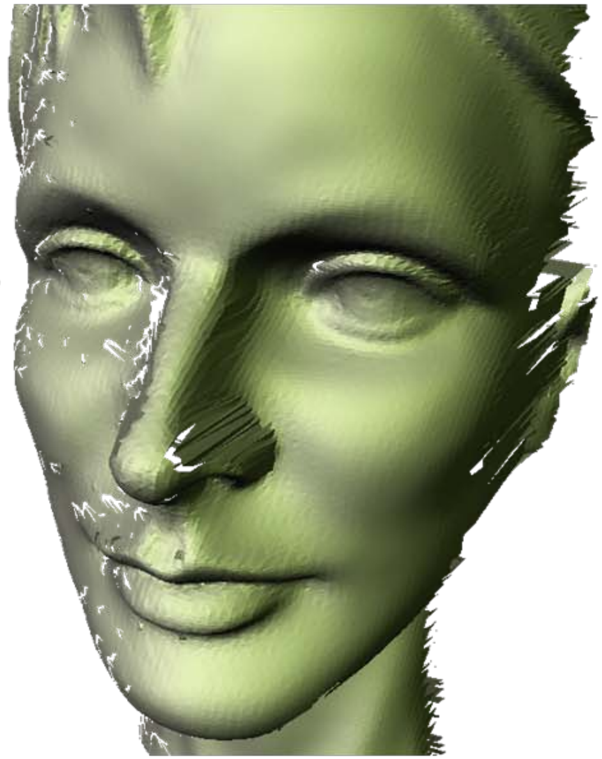
Original Desbrun 1999 Our result

Comparisons

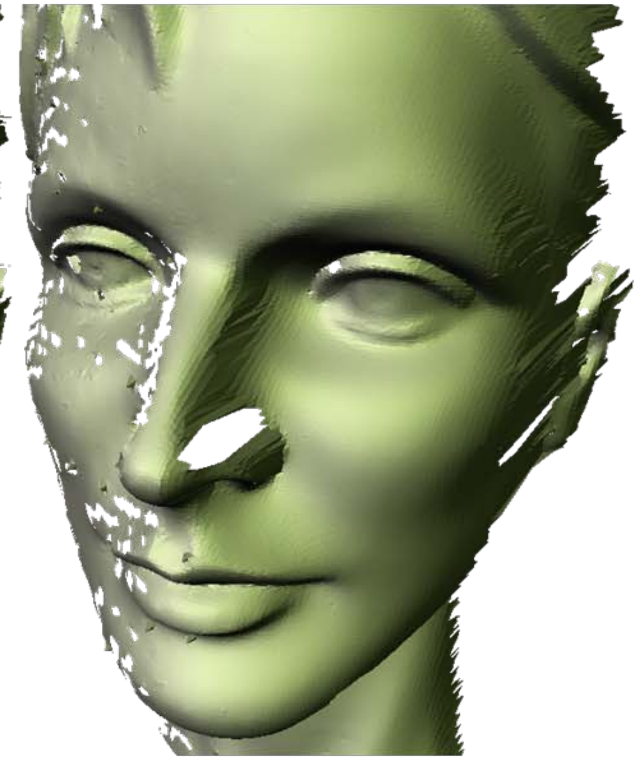




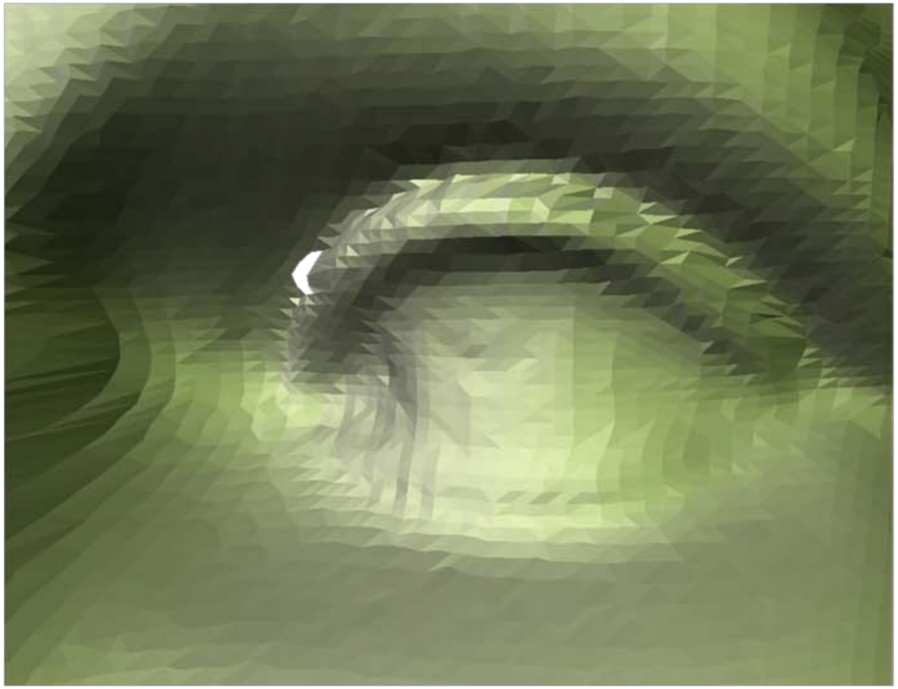
Source



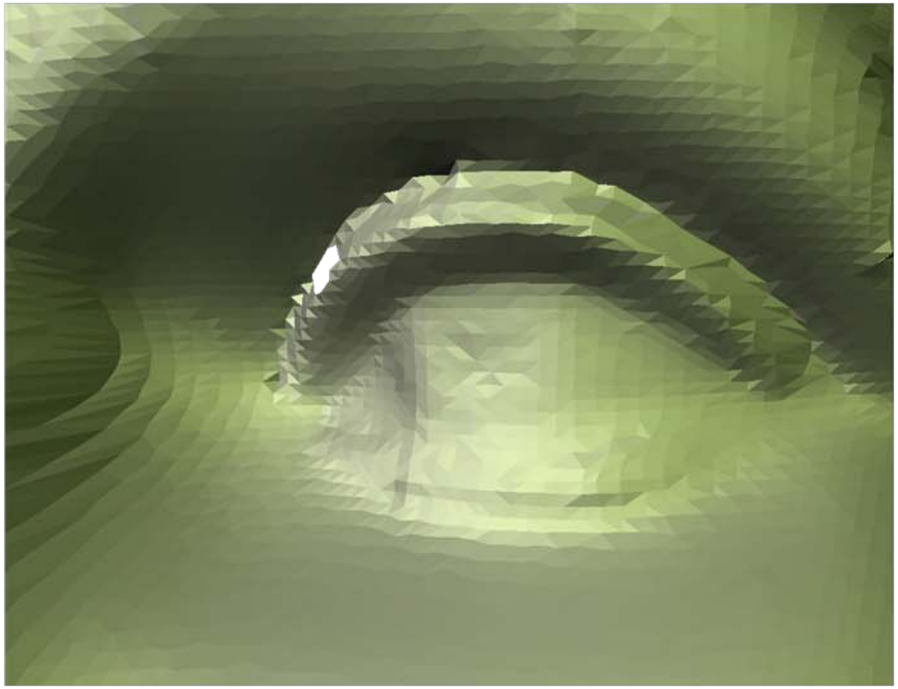
Non-iterative, Feature Preserving Mesh smoothing



Bilateral mesh denoising



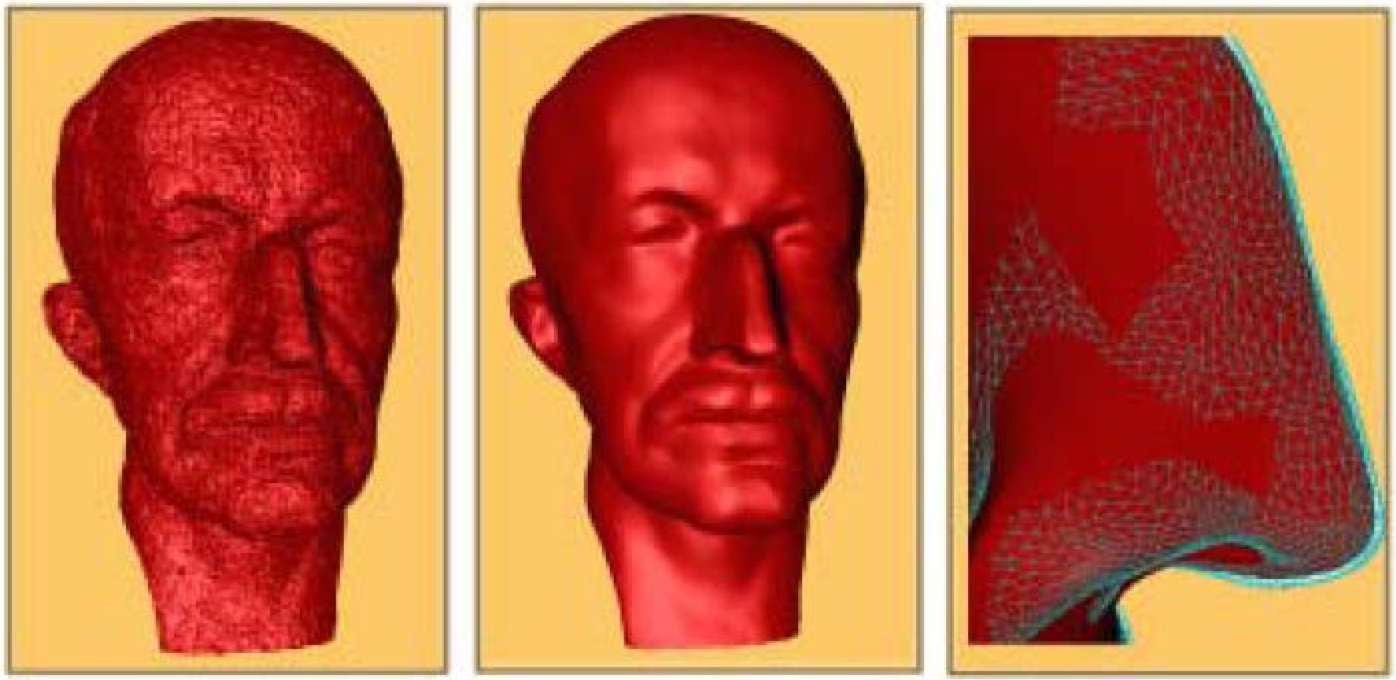
Non-iterative, Feature Preserving Mesh smoothing



Bilateral mesh denoising

Local Volume Preserving

Shrinkage

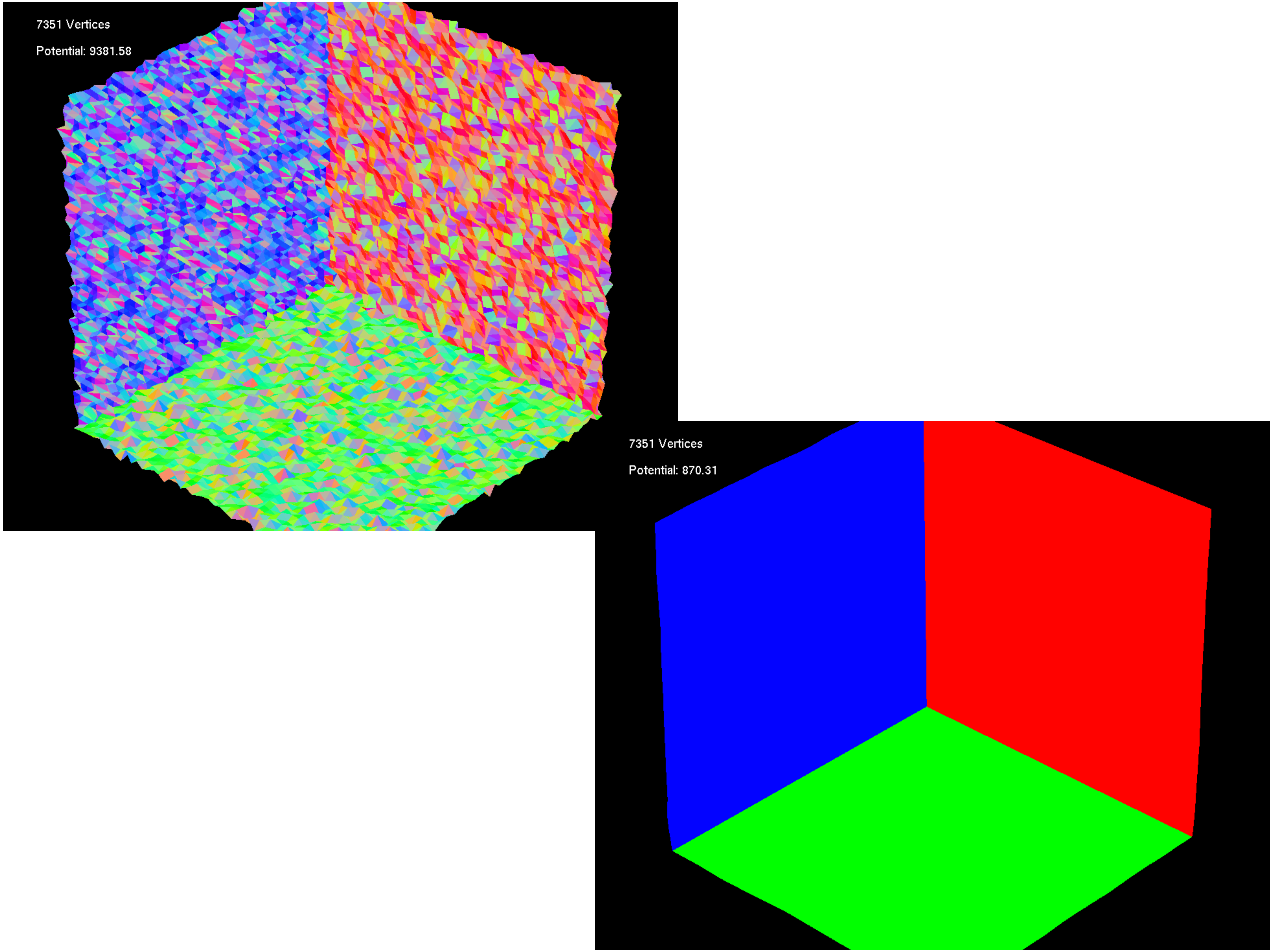


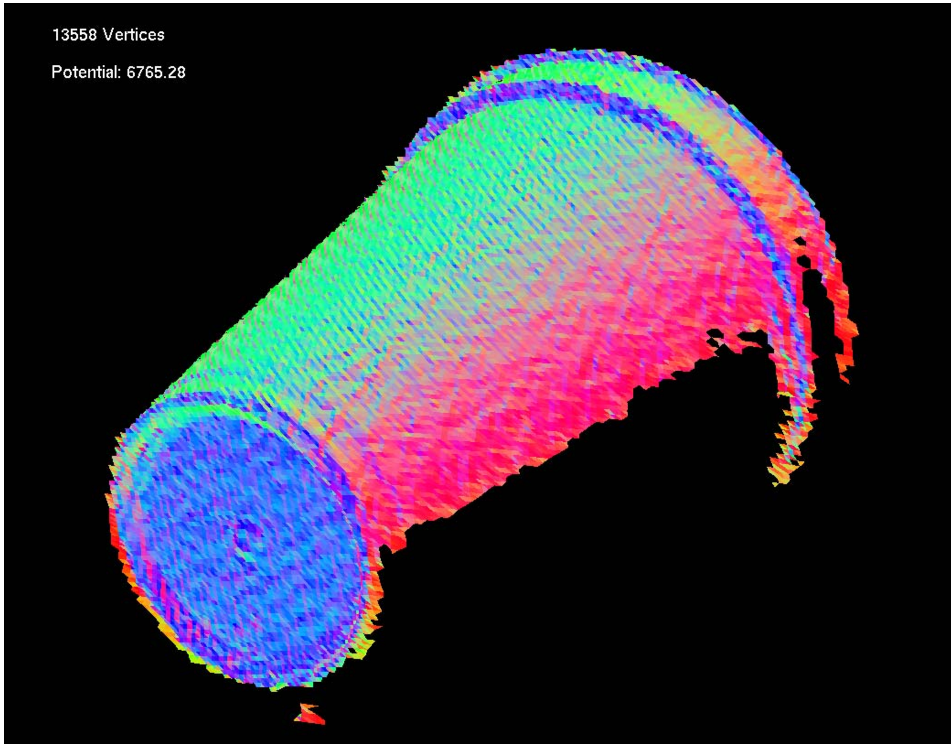
- Special treatment – volume preserving scaling

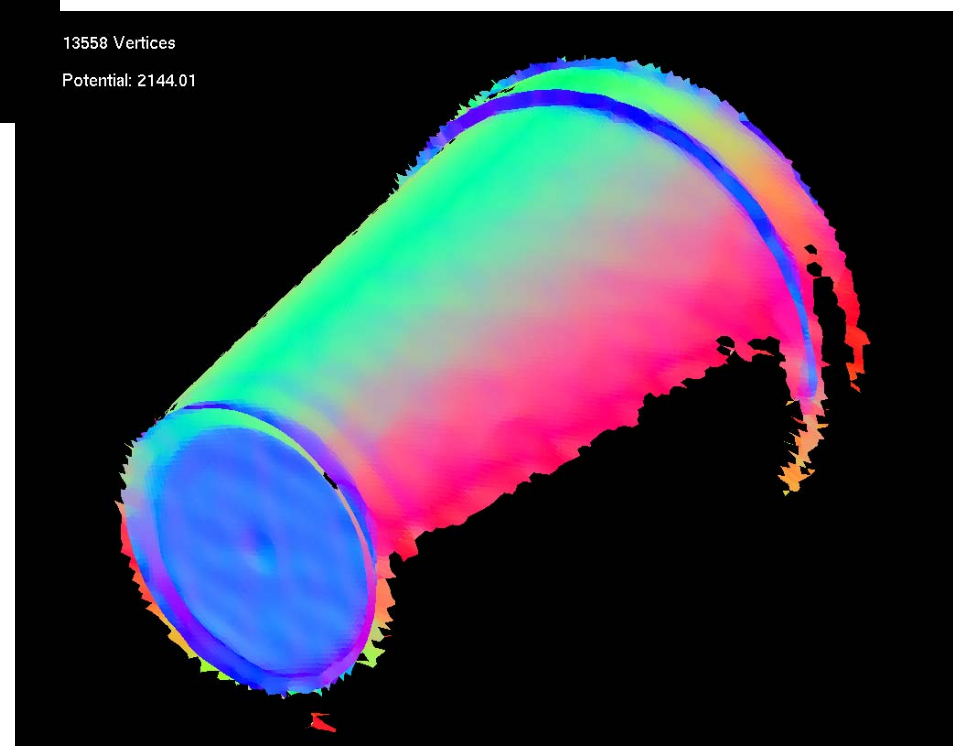
MRF Approach

MRF Approach

- Learning MRF potentials
- A learning algorithm to determine a suitable edge potential function from a training set of data
- The selection of this edge potential is the purpose of the learning algorithm


7351 Vertices
Potential: 9381.58
7351 Vertices
Potential: 870.31


13558 Vertices
Potential: 6765.28


13558 Vertices
Potential: 2144.01

Global Laplacian Smoothing (GLS)

Ji et al. CAD/Graphics'2005

Best Student Paper

Smoothing Problem Mathematically

- Find a smoothed surface with minimum fairing energy

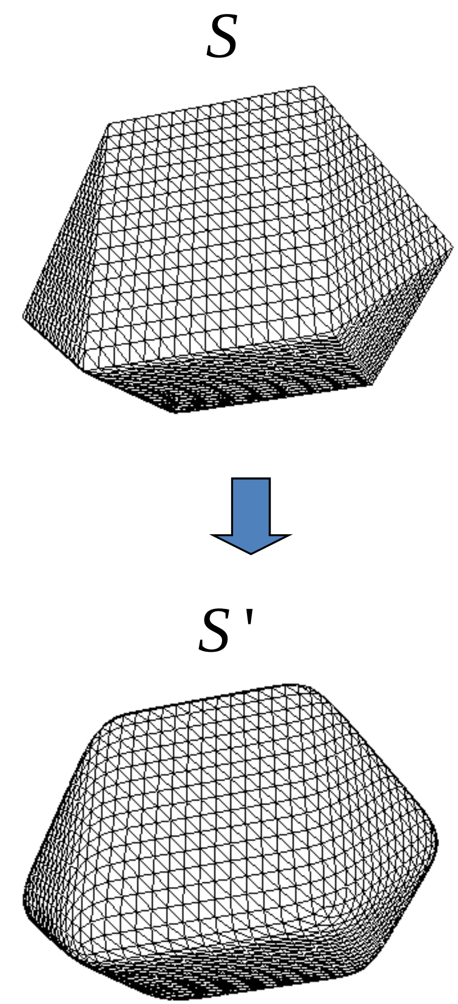
$$\min_{\mathcal{S}'} E(\mathcal{S}'),$$

- Fairing energy:

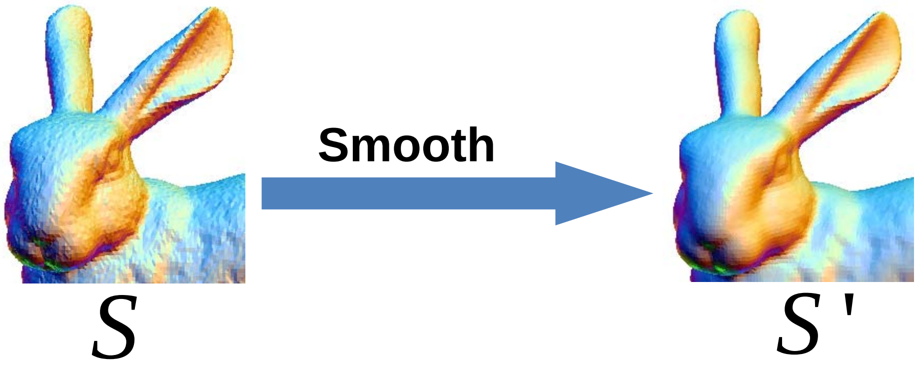
$$E(\mathcal{S}') = \underbrace{\alpha \int_\Omega \Psi(\mathcal{S}') du dv}_{\text{Smoothness constraint}} + \underbrace{\beta \int_\Omega (\mathcal{S}' - \mathcal{S})^2 du dv,}_{\text{Data fidelity}}$$

$$\int_\Omega \Psi(\mathcal{S}') du dv = \int_\Omega (\boldsymbol{F}_u^2 + \boldsymbol{F}_v^2) du dv,$$

$$\int_\Omega \Psi(\mathcal{S}') du dv = \int_\Omega (\boldsymbol{F}_{uu}^2 + 2\boldsymbol{F}_{uv}^2 + \boldsymbol{F}_{vv}^2) du dv.$$



Our Approach



- Global
  - Global Laplacian operator
  - Global shape preservation
- Non-iterative
- Feature preserving

Smoothing Problem

- A global optimization problem
  - Minimize smoothness energy within some tolerance

$$\min_{S'} \quad E(S')$$

  - A mathematical model

$$E(S') = \alpha \int_{\Omega} \Psi(S')dudv + \beta \int_{\Omega} (S' - S)dudv$$

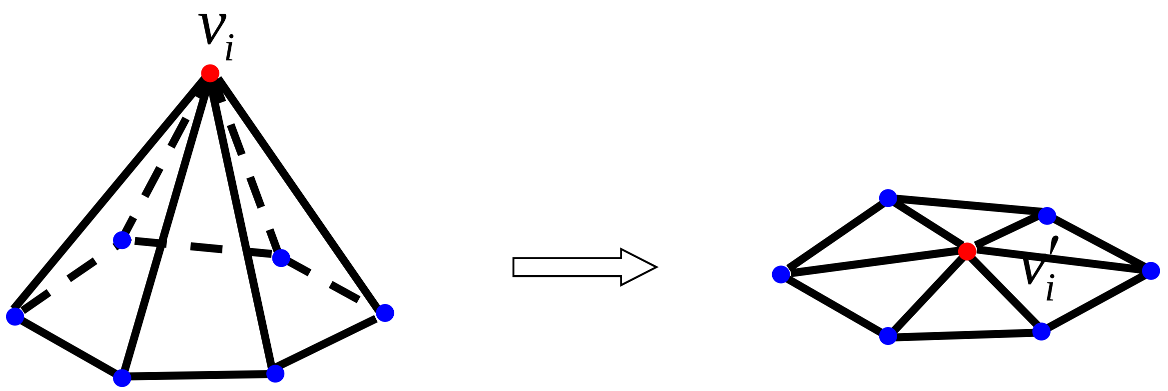
- **Smoothness term**

  membrane, thin-plate…

- **Fidelity term**

Local Lapacian Fairness

- Local discrete Laplacian smoothing operator

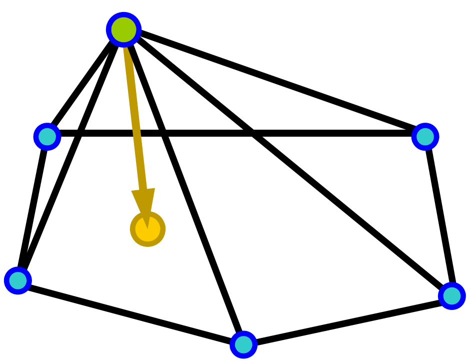


$$L(v_i) = v_i - \sum_{j \in N(i)} \omega_{ij} v_j = 0$$

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

Laplacian of Mesh

- Discrete Laplacians



$$L(v_i) = v_i - \sum_{j \in N(i)} \omega_{ij} v_j = 0$$

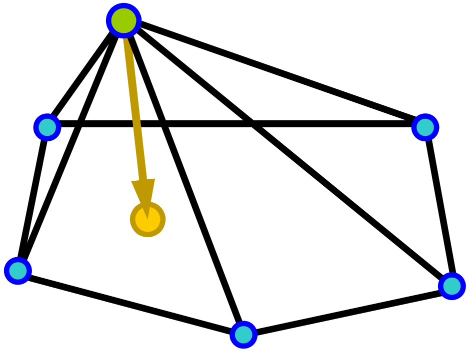
$$\mathbf{L}\,\mathbf{x} = 0$$

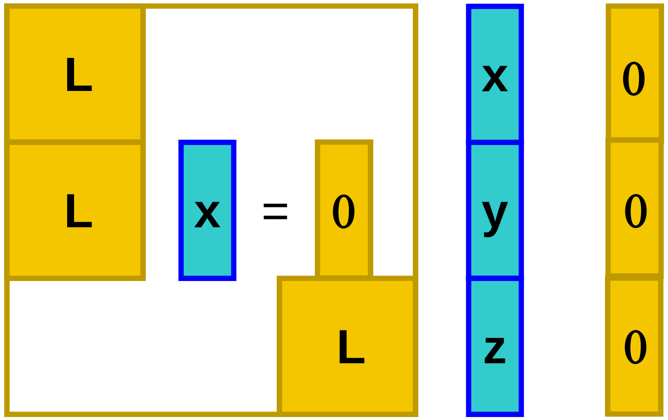
$$L_{ij} = \begin{cases} 1, & i = j, \\ -\omega_{ij}, & (i,j) \in E, \\ 0, & \text{other}. \end{cases}$$

- Laplacian of the mesh

Laplacian of Mesh

- Surface reconstruction





$$L(v_i) = v_i - \sum_{j \in N(i)} \omega_{ij} v_j = 0$$

Properties of Laplacian

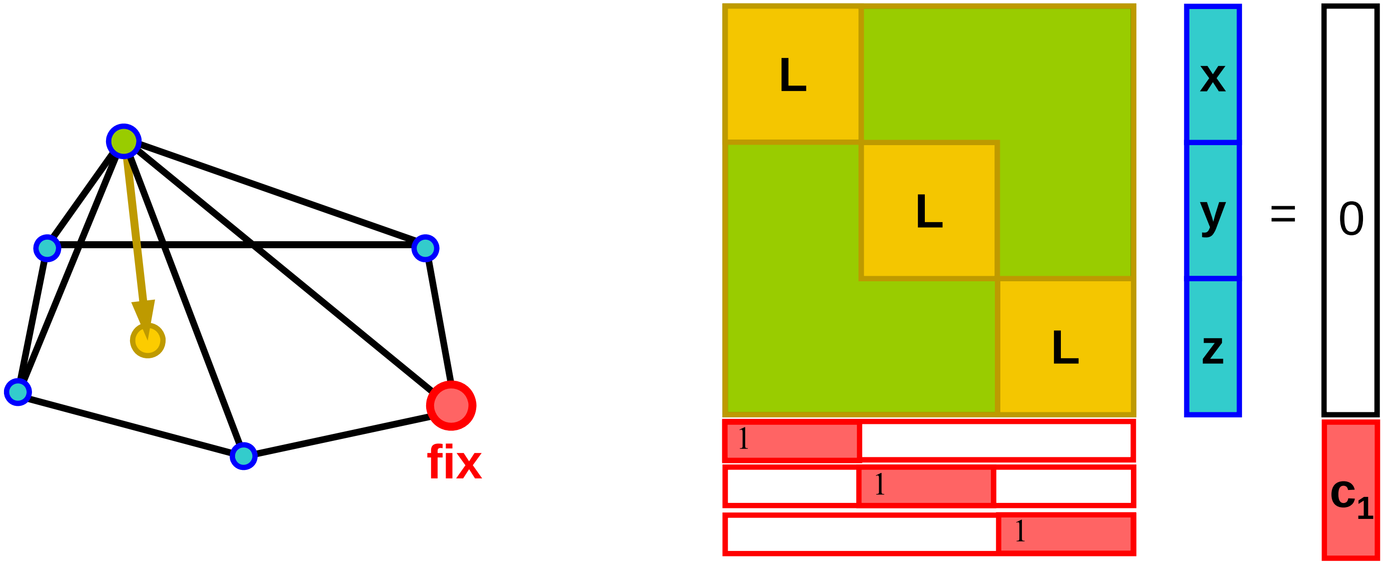
$$\begin{pmatrix} & & \\ & \boldsymbol{L} & \\ & & \end{pmatrix} \begin{pmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \\ \vdots \\ \boldsymbol{v}_3 \end{pmatrix} = \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \vdots \\ \boldsymbol{0} \end{pmatrix}$$

Rank(L) = n-k

- k is the number of connected components of the mesh
- Need to add some constraints

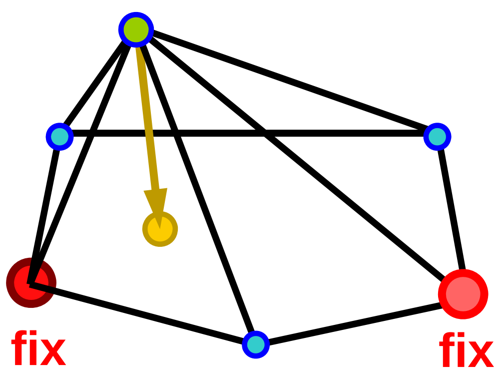
Vertex Constraints

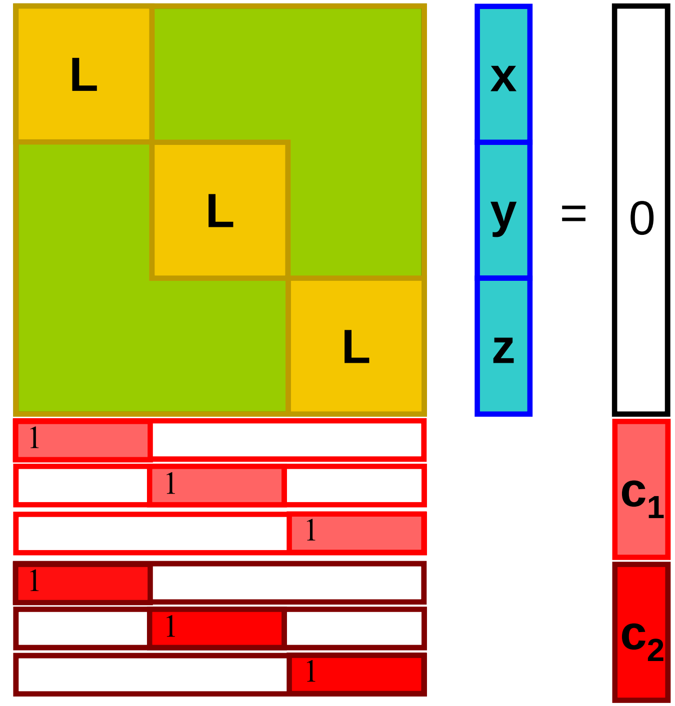
- Add position constraint for one vertex



Vertex Constraints

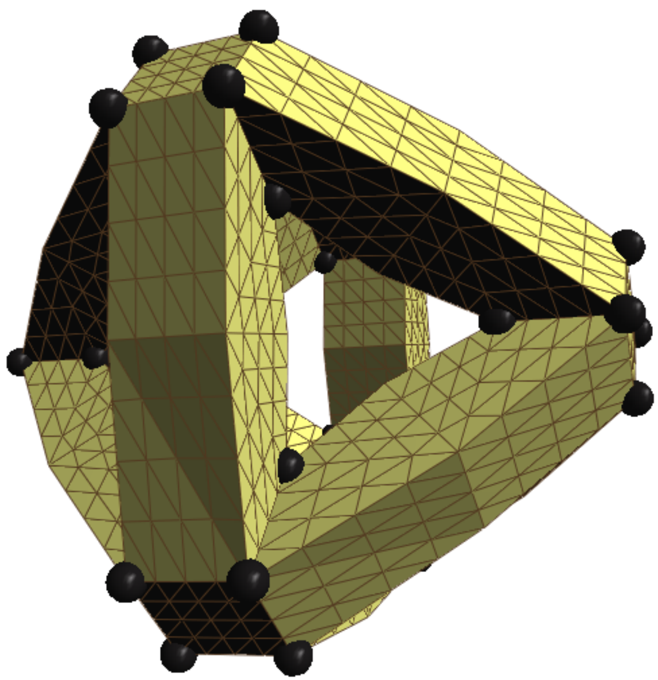
- Add position constraints for more vertices

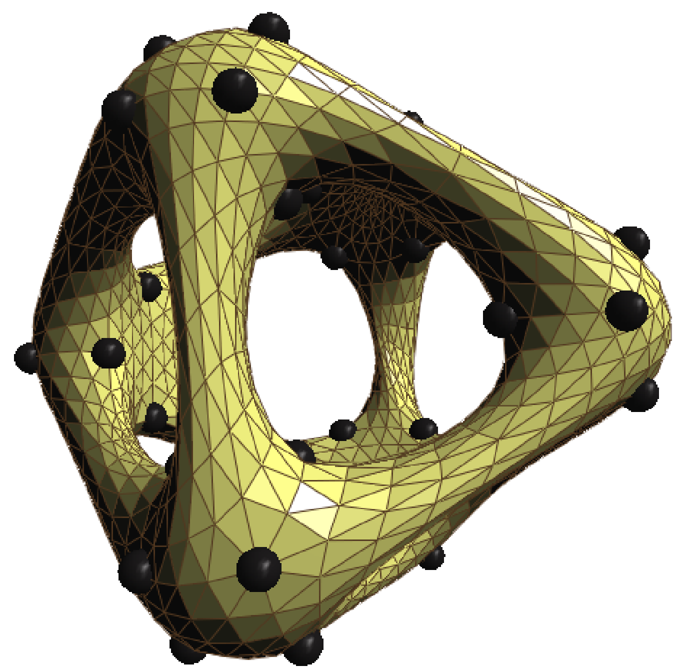




Adding Vertex Constraints

$$\min_{X'}\{\|LX'\|^2 + \mu^2 \sum_{i\in C} \left| v_i' - v_i \right|^2\}$$

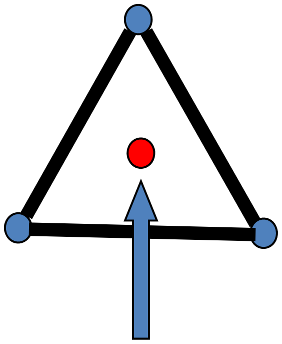




Face Constraints

$$
\begin{pmatrix} & & L & \\ 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix}
\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}
=
\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ c_1 \\ c_2 \\ t_1 \\ t_2 \end{pmatrix}
$$

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

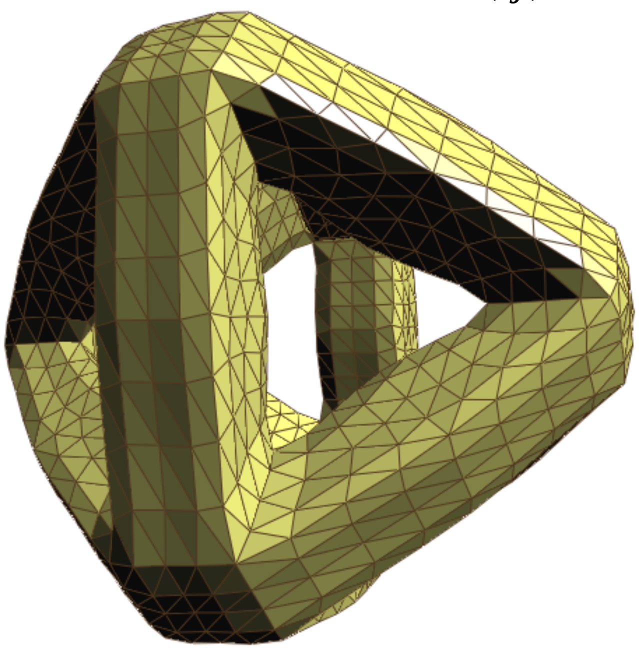


barycenter

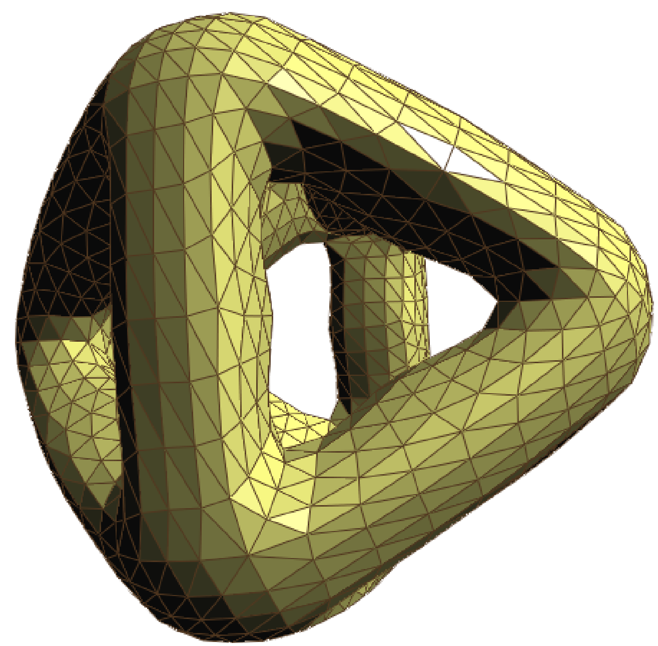
$$v_{center} = \frac{1}{3}\left(v_i + v_j + v_k\right)$$

Adding Face Constraints

$$\min_{X'}\{\|LX'\|^2 + \sum_{<i,j,k>\in T} \lambda^2 \left|(v_i' + v_j' + v_k') - (v_i + v_j + v_k)\right|^2\}$$



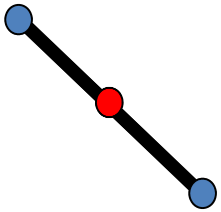
$\lambda = 0.5$



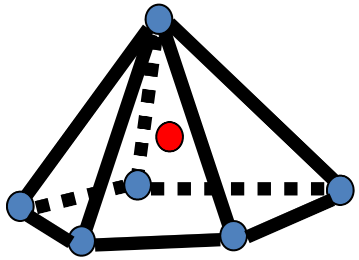
$\lambda = 0.3$

Other Constraints

- Edge constraints



- 1-ring barycenter constraints



- Other linear constraints

Minimizing Energy

$$\min_{X'}\left\{\left\|LX'\right\|^2 + \sum_{i\in C}\mu^2\left|v_i' - v_i\right|^2 + \sum_{<i,j,k>\in T}\lambda^2\left|(v_i' + v_j' + v_k') - (v_i + v_j + v_k)\right|^2\right\}$$

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

Least Square Solution

- An over-determined system:

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

- Normal equation:

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

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

One Channel Solution

- Very efficient solution by Cholesky factorization of $A^T A$:

$$\boldsymbol{A}^T \boldsymbol{A} = \boldsymbol{R}^T \boldsymbol{R}$$

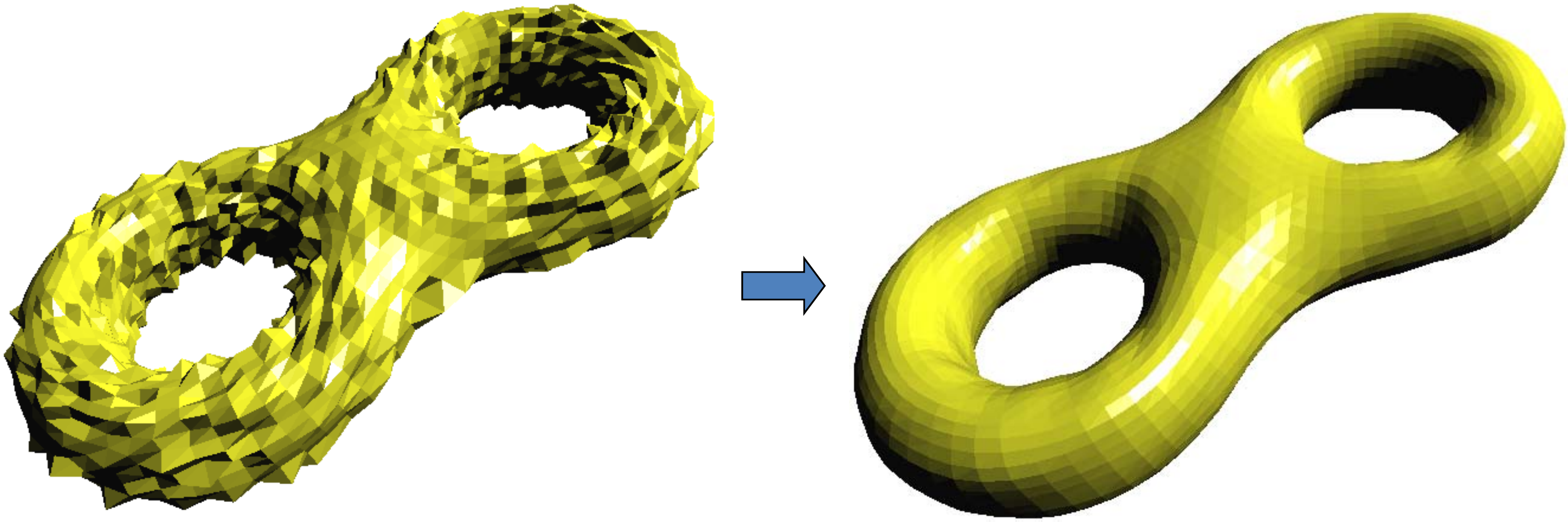
R is upper-triangular and sparse

Once R is computed, solving for **x**, **y**, **z** by back-substitution:

$$\boldsymbol{R}^T \xi = \boldsymbol{A}^T \mathbf{b}$$

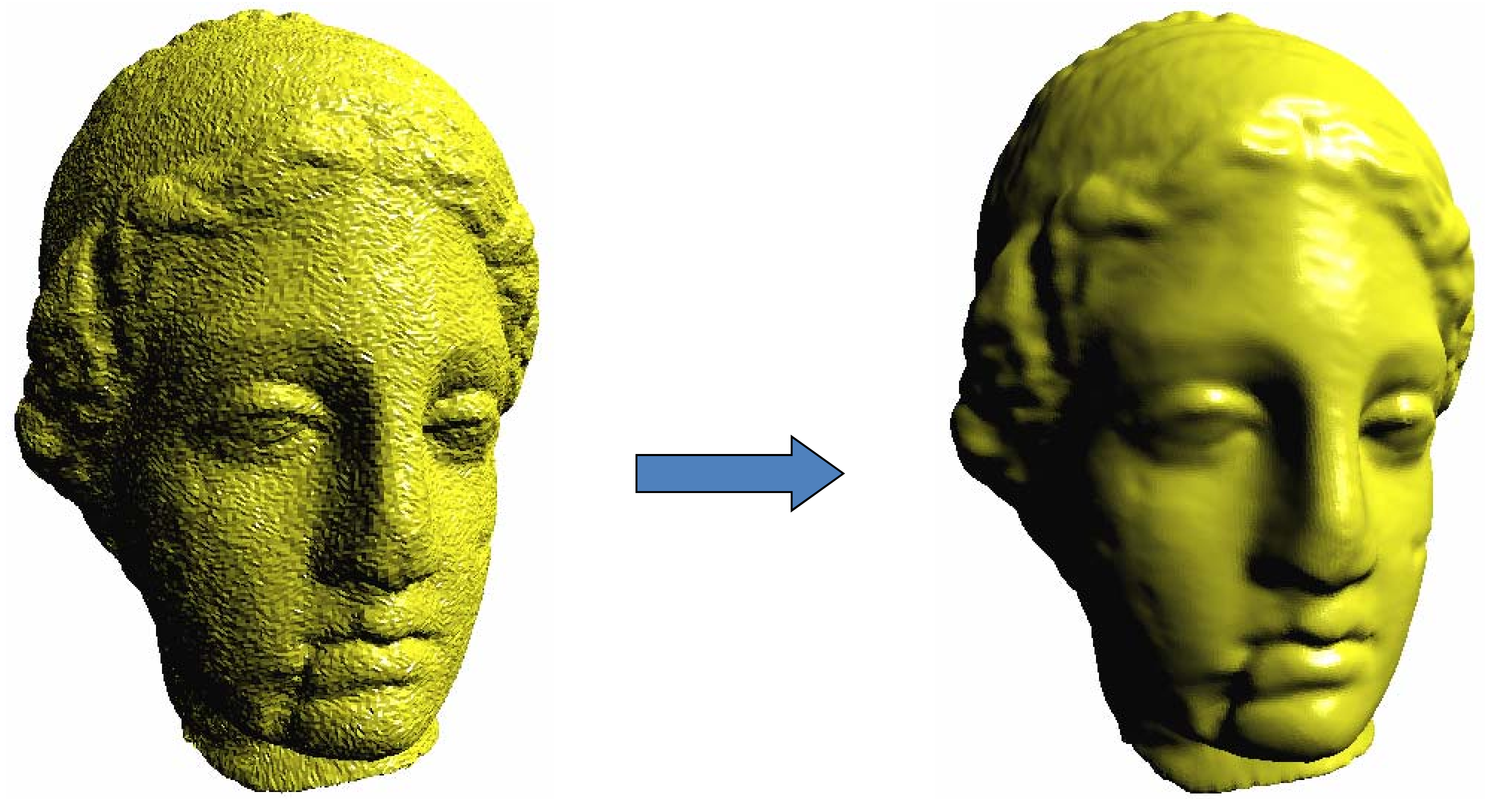
$$\boldsymbol{R}\mathbf{x} = \xi$$

Results



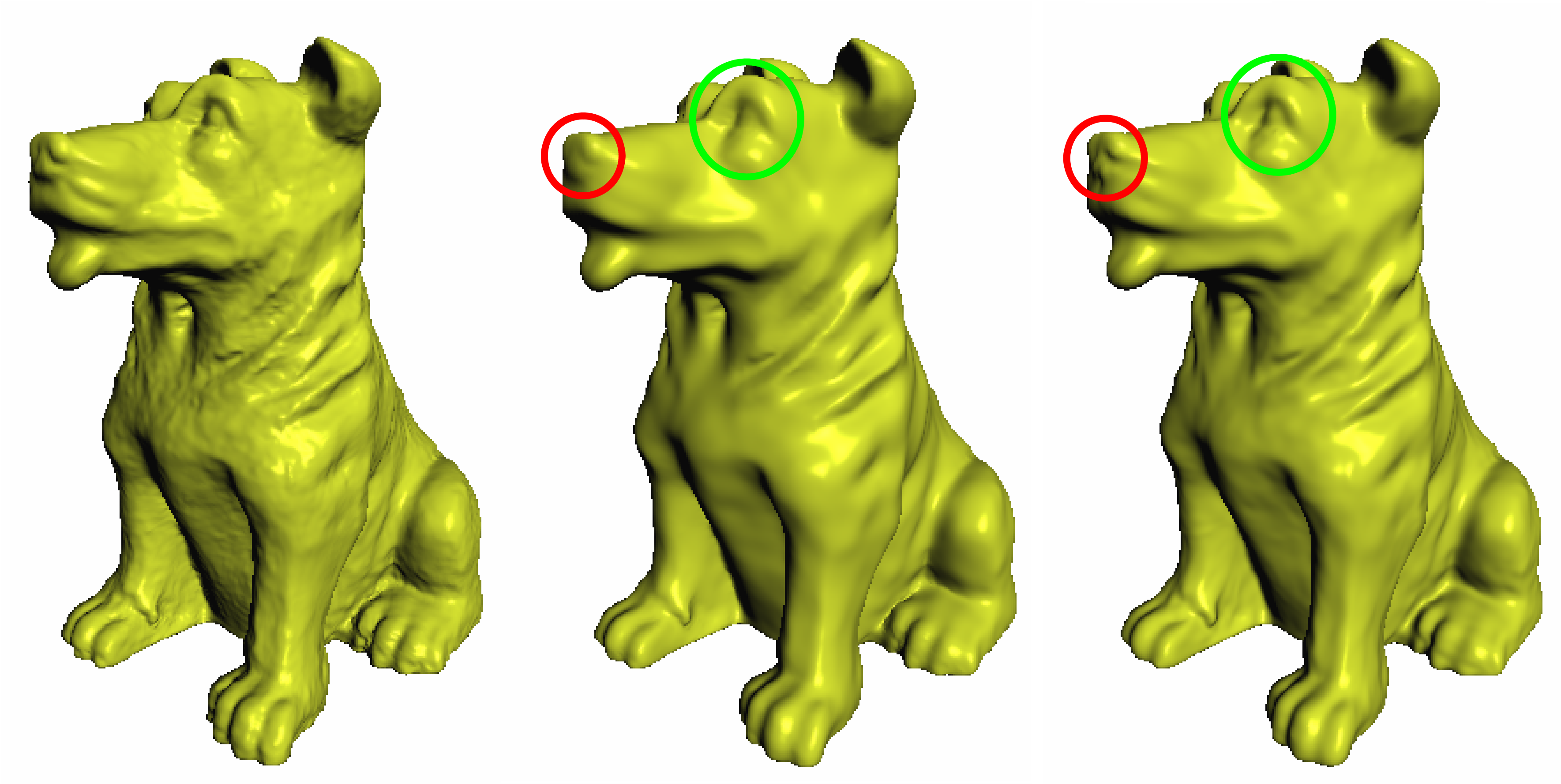
‘8’-like mesh model
3070 vertices, 6144 triangles

Results



Venus head model
134359 vertices, 268714 triangles

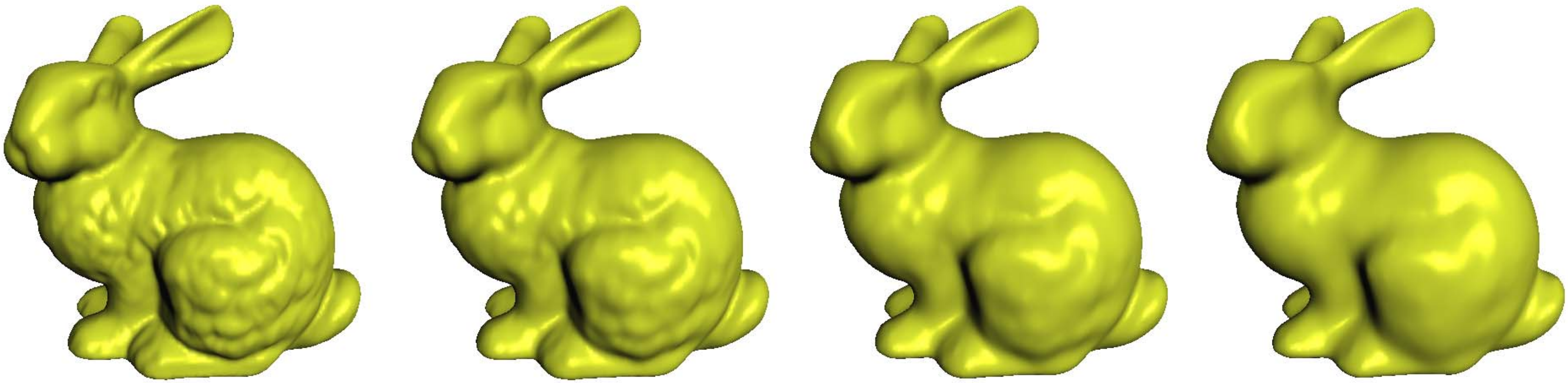
Comparisons



Original noisy mesh   Bilateral approach   Our Approach

Dog model, 195586 vertices, 391168 triangles

LoD Smoothing



Applying our algorithm to the bunny model with different parameters.

Summary

- A new smoothing framework
- Global Laplacian operator
- Non-iterative
- Feature preserving
  - Vertex constraints
  - Face constraints
  - Other linear constraints

Summary

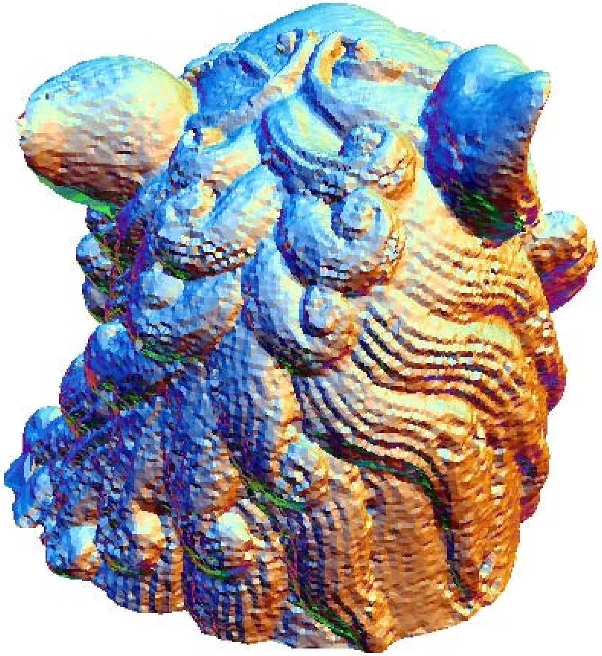
- Filtering
  - Laplacian filtering
  - Bilateral filtering
- Quantity
  - Vertex positions
  - Normal
  - Laplacian vectors

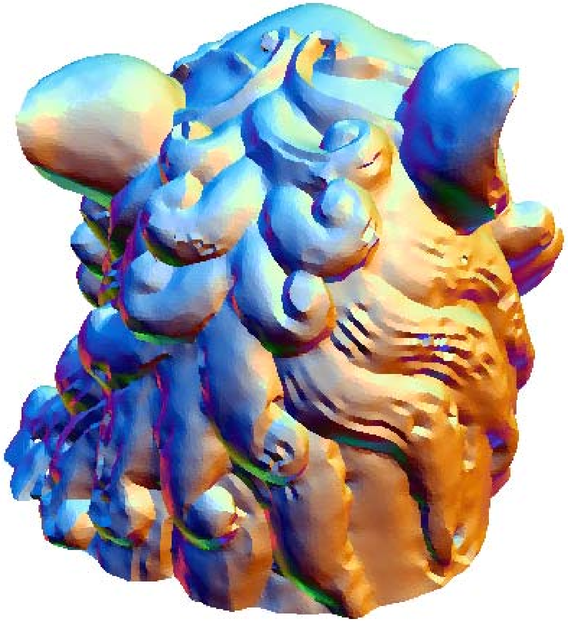
Many Problems Remain

Mesh smoothing remains to be an active research area

Photo

Scanned mesh





Smoothed mesh

Discussions