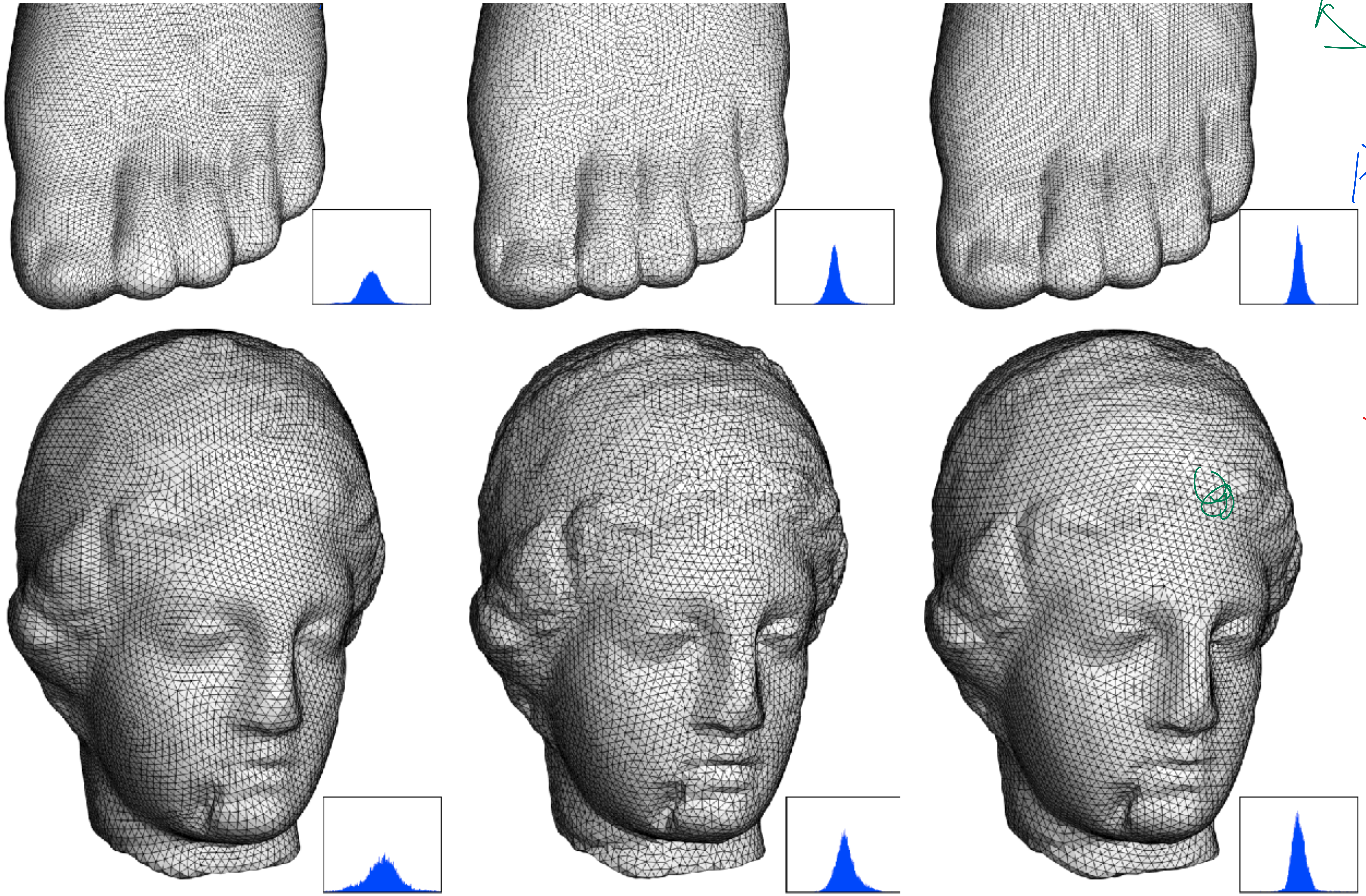
Remeshing

$[VR(P)]$

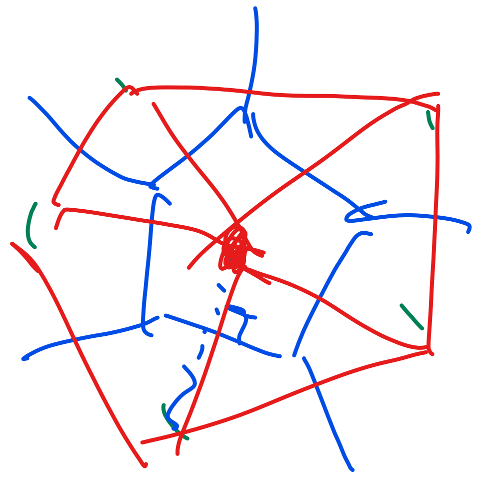
VORONOI REGIONS

DELAUNAY TRIANGULATION



$p_i \to \{(i,j,k)\}$

maximum min angle

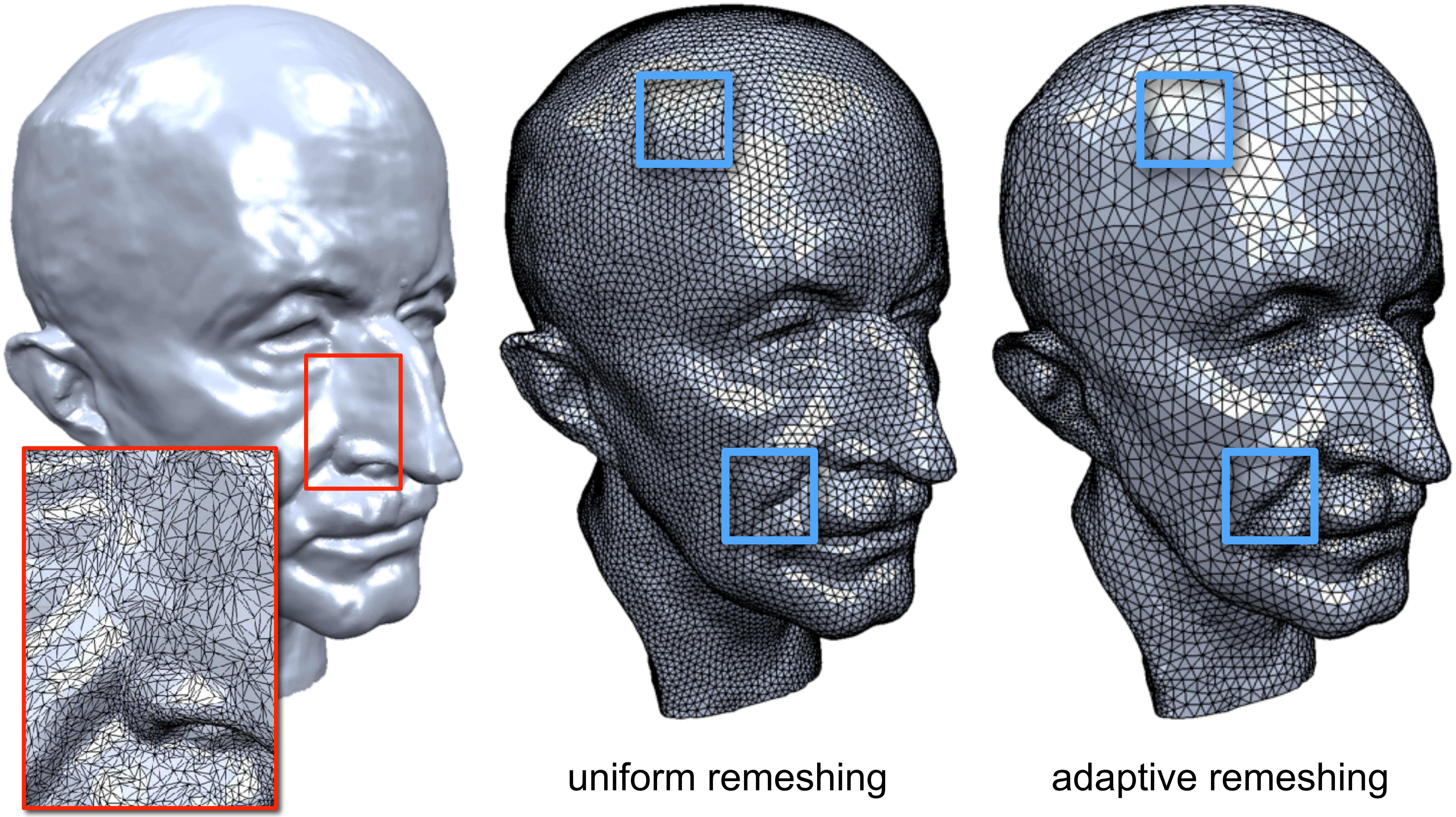


Remeshing

- Problem setting:
  – *Given a 3D mesh,* ***improve*** *its triangulation while preserving its geometry.*
- Why?
  – **Robustness** of numerical computations
- How?
  – Avoid very small / large triangle **angles**
  – Avoid very small / large triangle **areas**

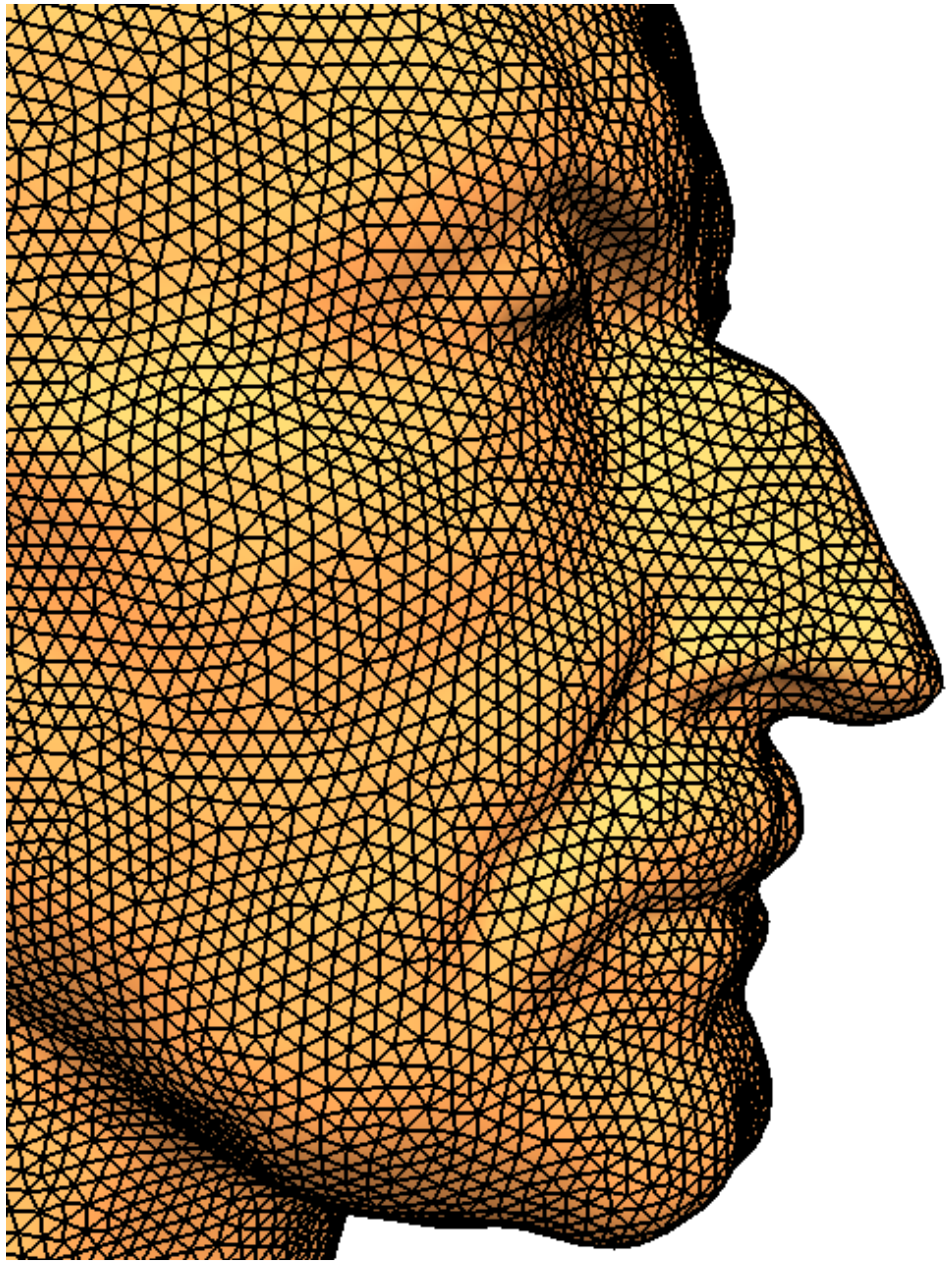
Isotropic Remeshing

length preserved



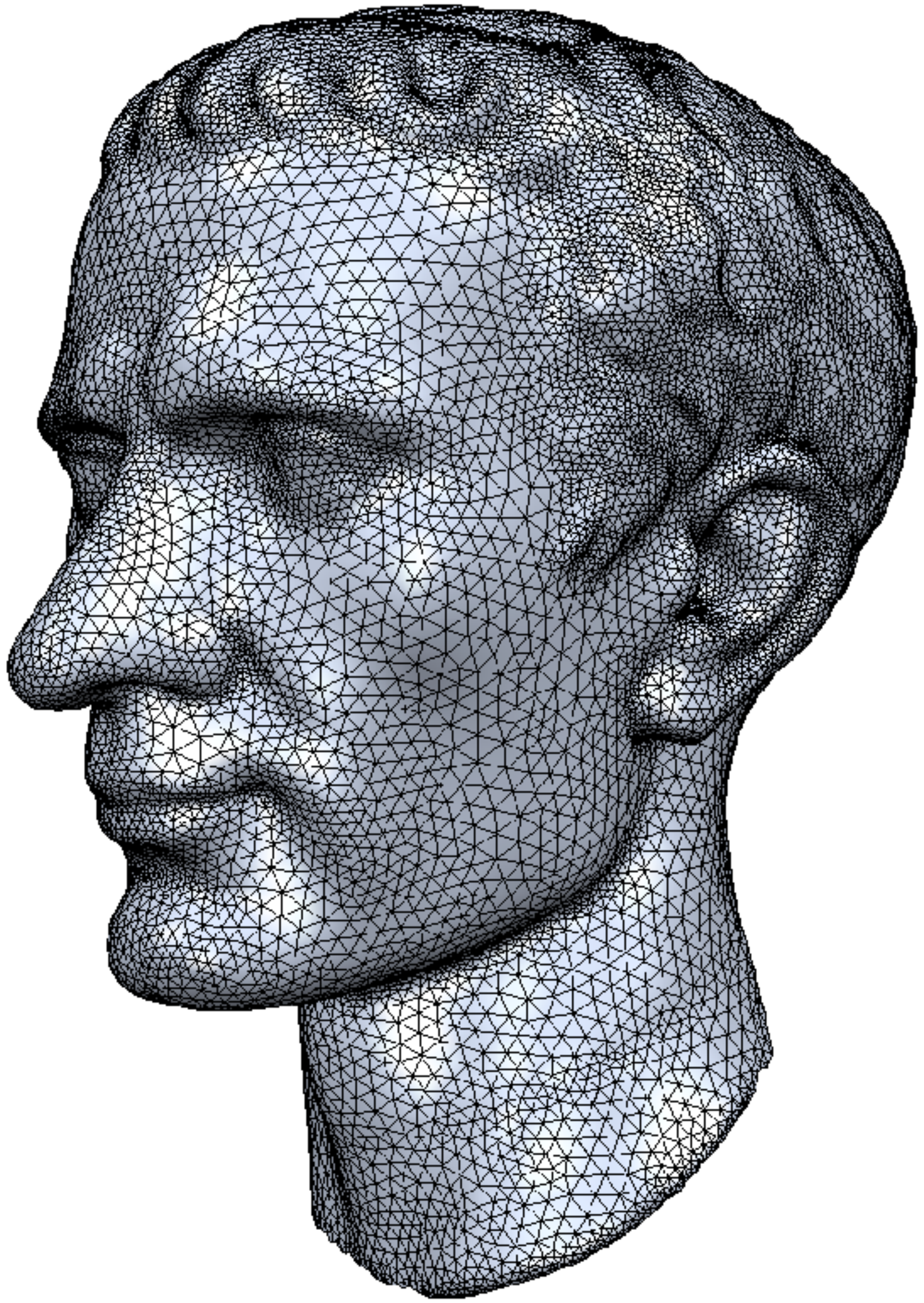
What is a good mesh?

- Equal edge lengths
- Equilateral triangles
- Valence close to 6



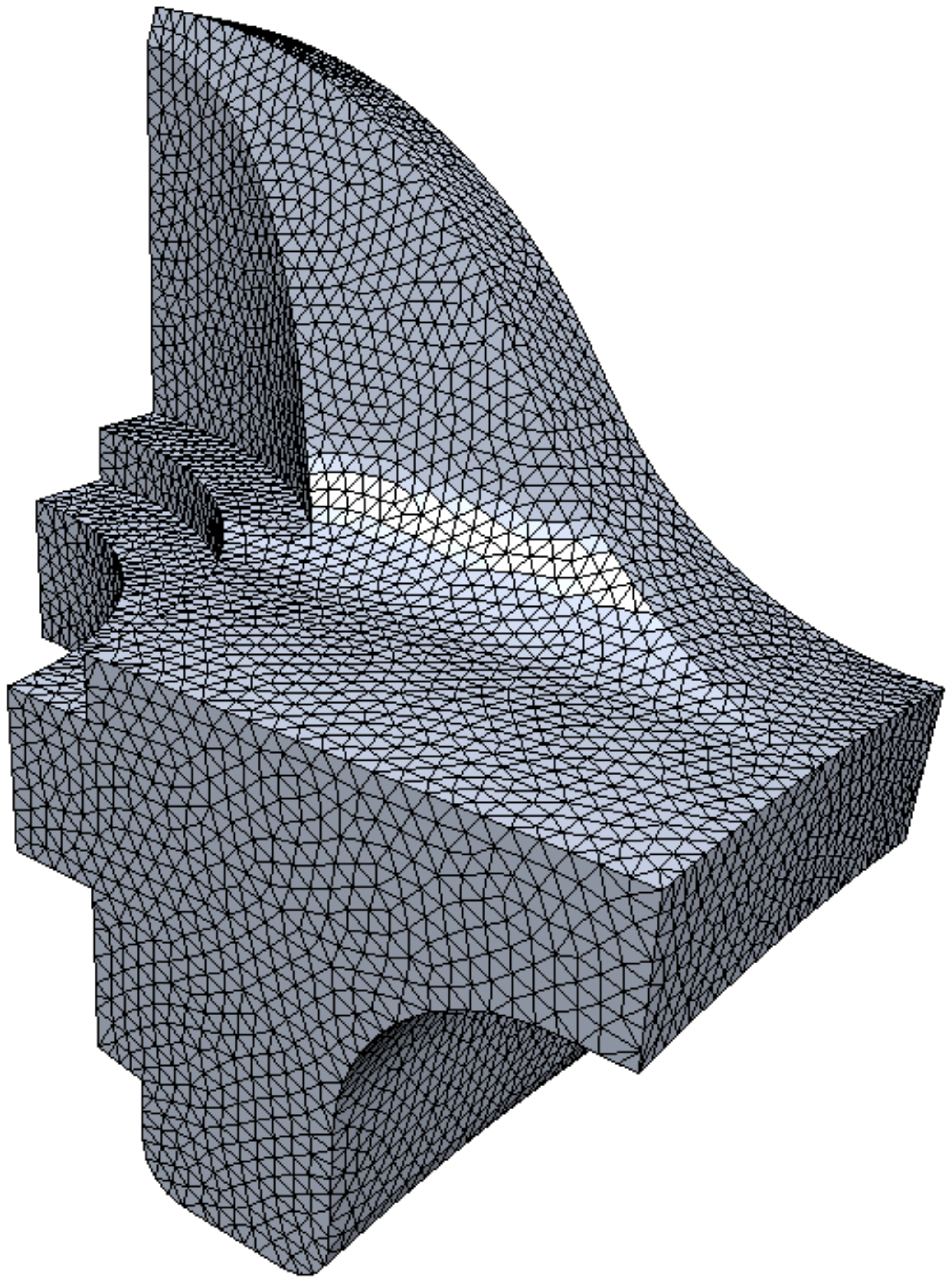
What is a good mesh?

- Equal edge lengths
- Equilateral triangles
- Valence close to 6
- Uniform vs. adaptive sampling



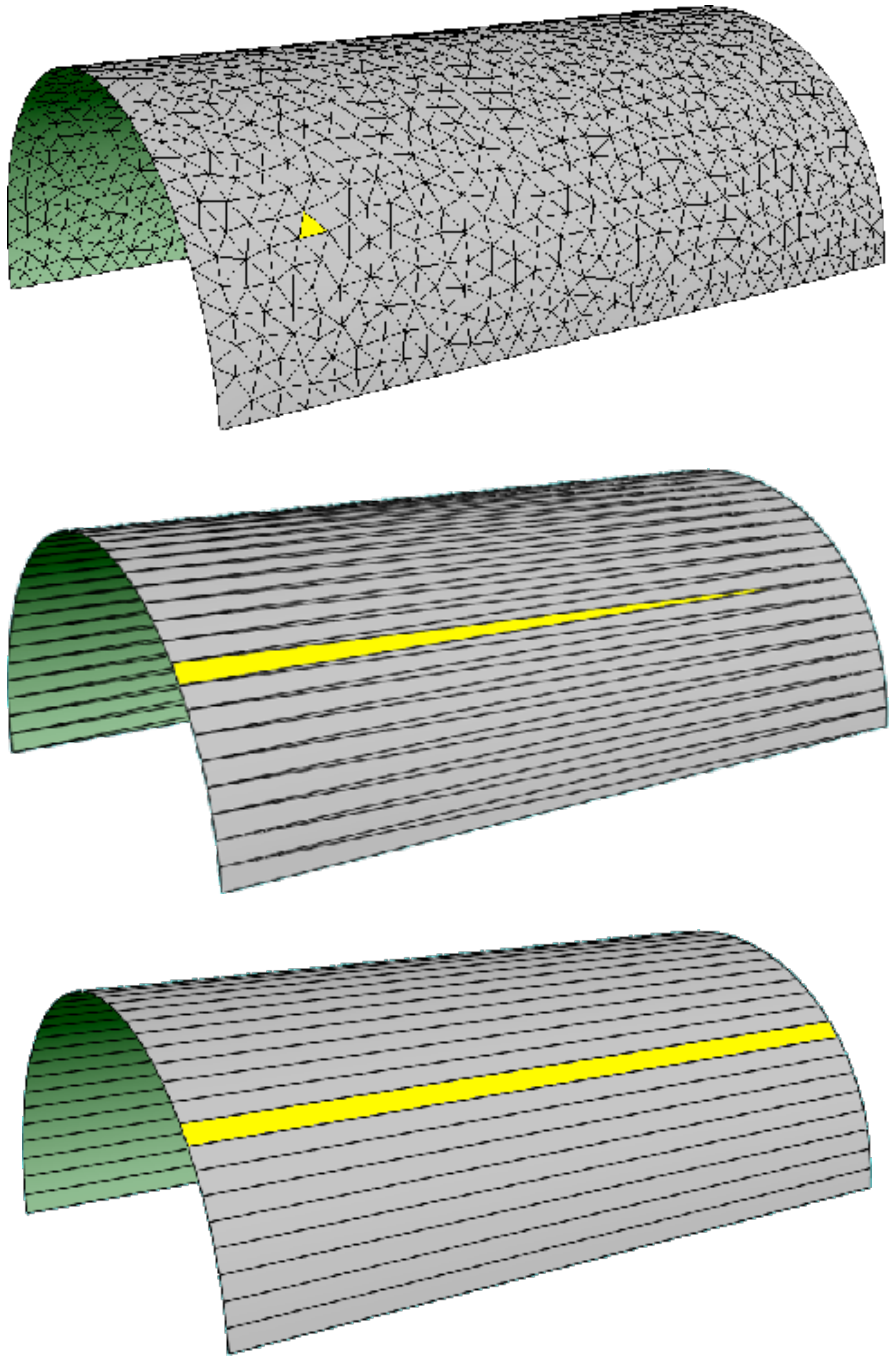
What is a good mesh?

- Equal edge lengths
- Equilateral triangles
- Valence close to 6
- Uniform vs. adaptive sampling
- Feature preservation



What is a good mesh?

- Equal edge lengths
- Equilateral triangles
- Valence close to 6
- Uniform vs. adaptive sampling
- Feature preservation
- Alignment to curvature lines
- Isotropic vs. anisotropic
- Triangles vs. quadrangles



Two Primary Approaches

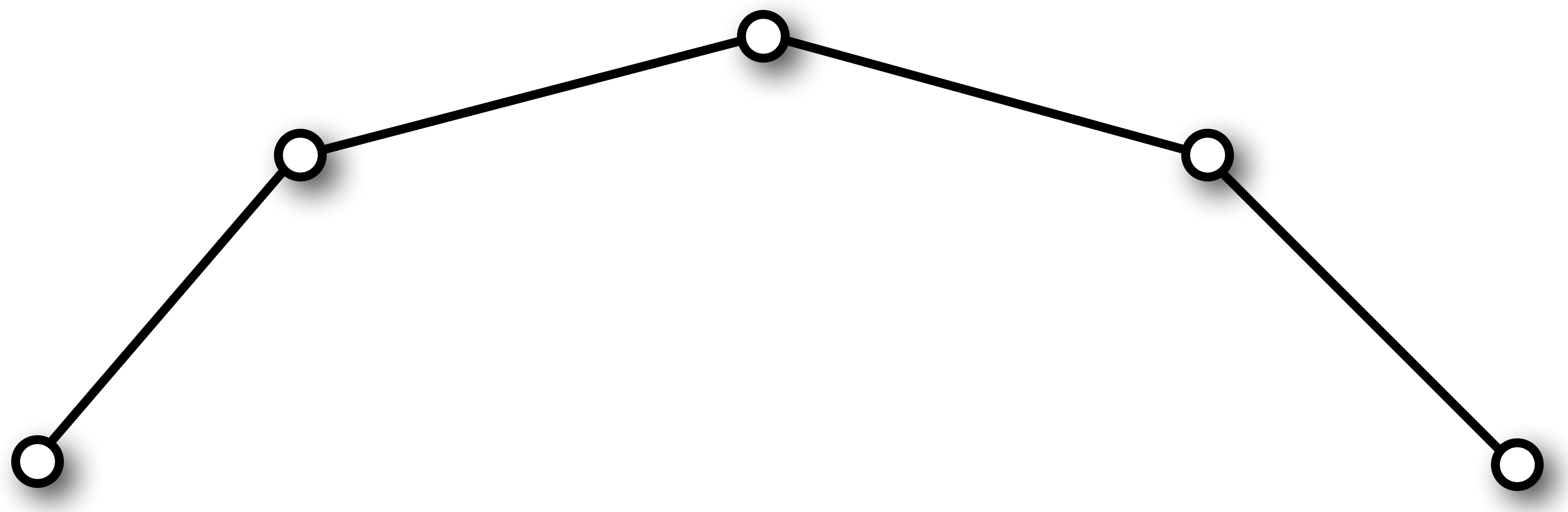
1. Parametrization based 3D → 2D → 1D

- map to 2D domain / 2D problem
- computationally more expensive
- works even for coarse resolution remeshing

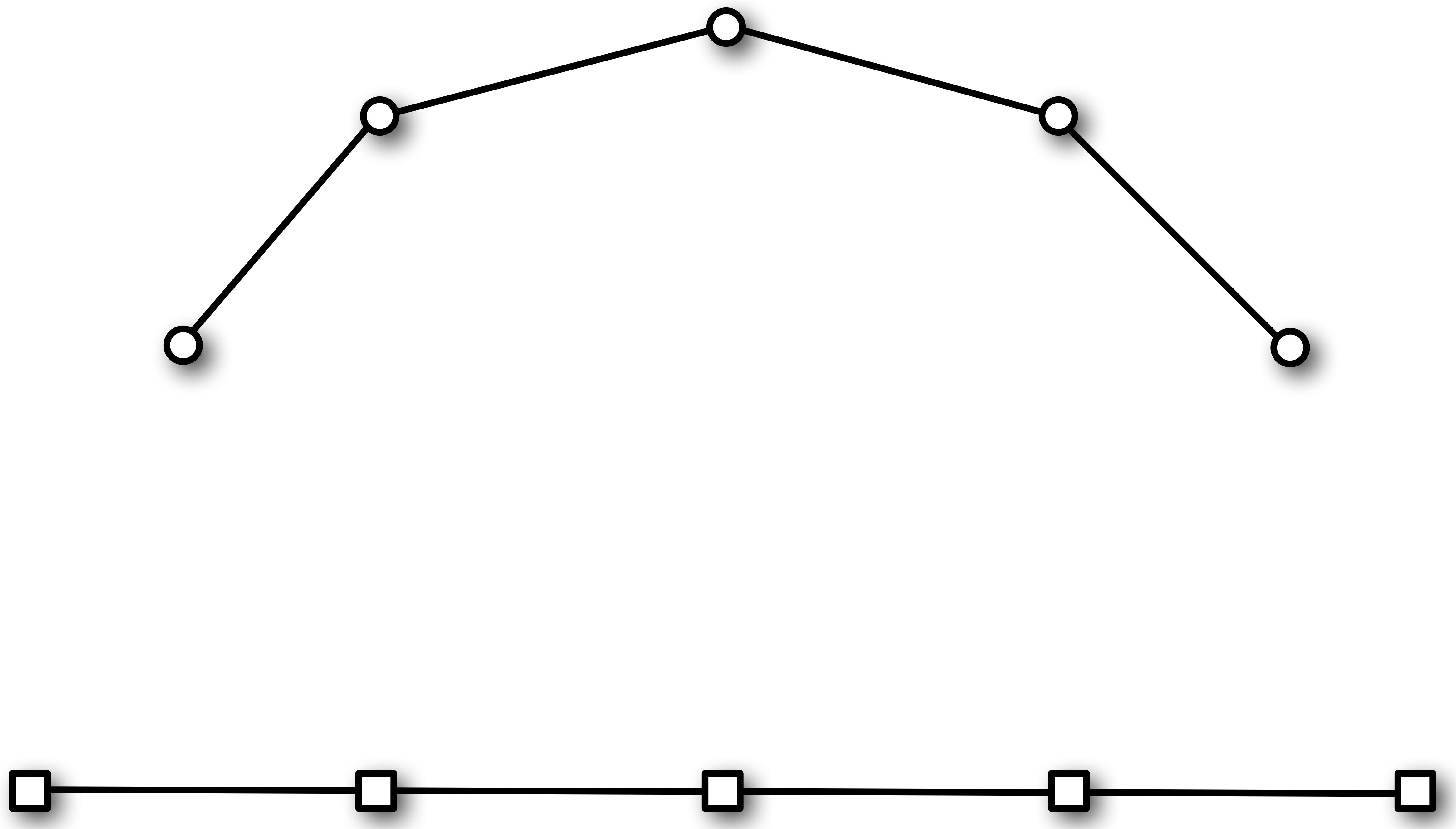
2. Surface based 3D

- operate *directly* on the surface
- treat surface as a set of points / polygons in space
- efficient for high resolution remeshing

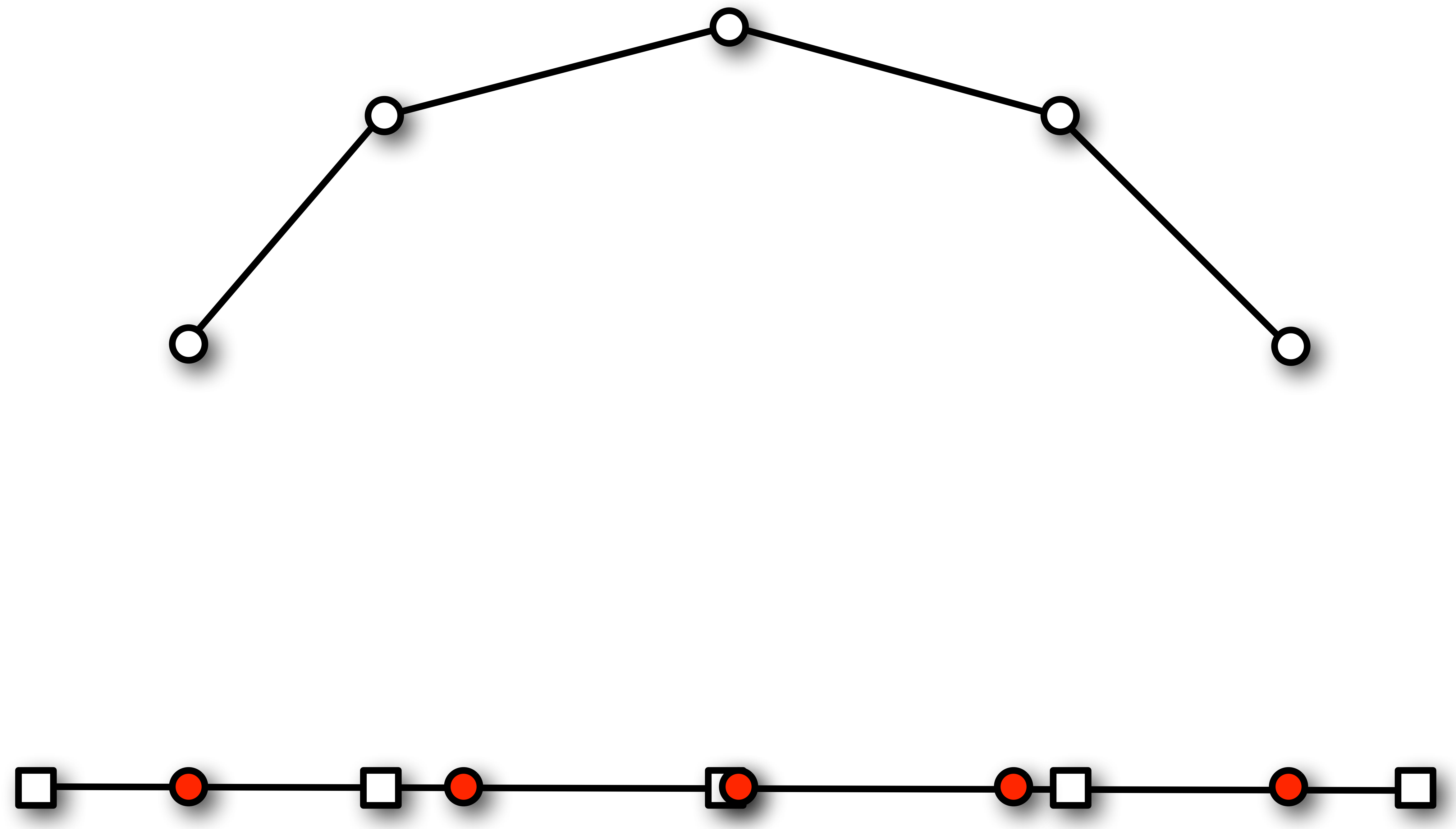
1. Parametrization Based



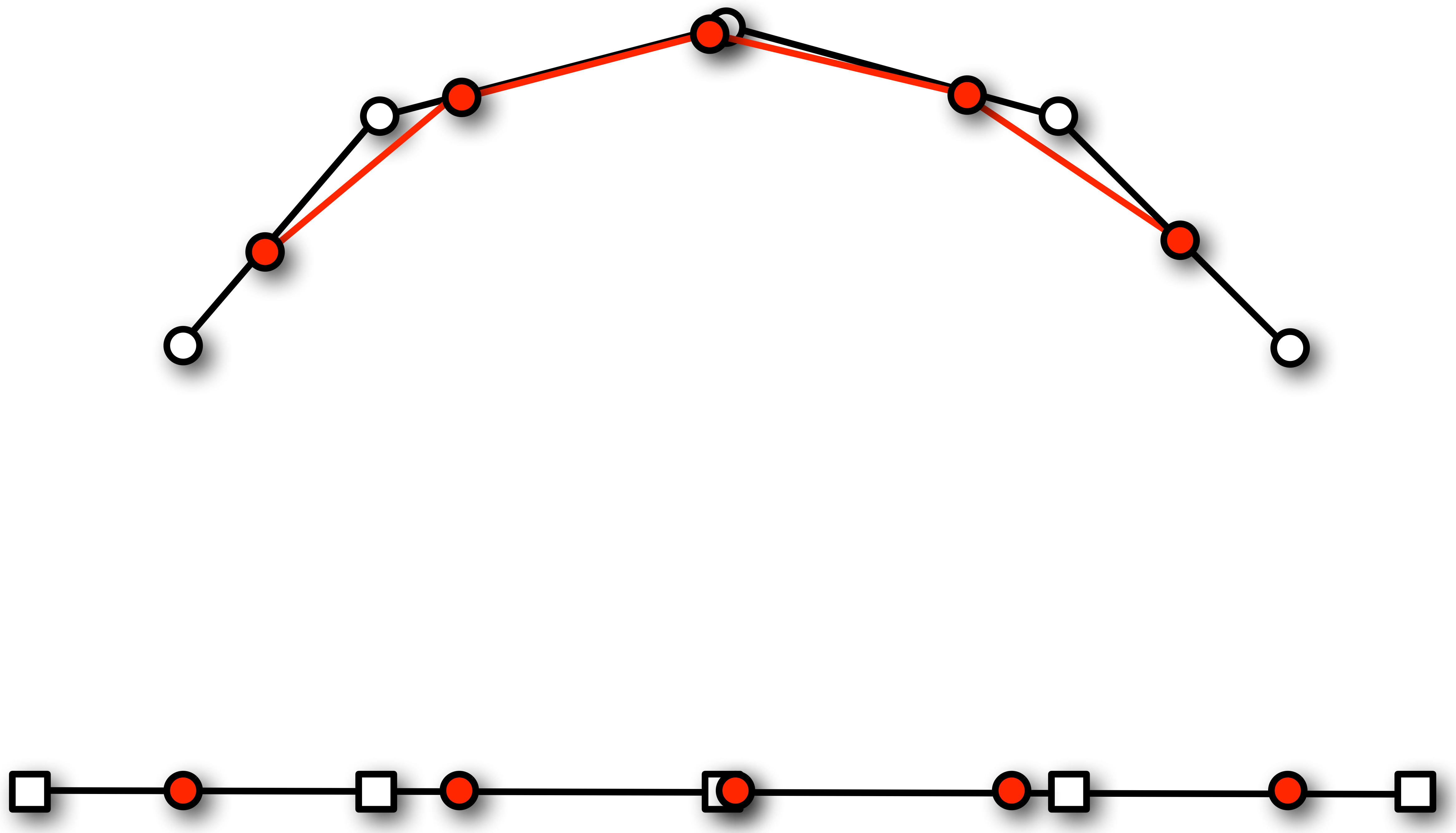
1. Parametrization Based



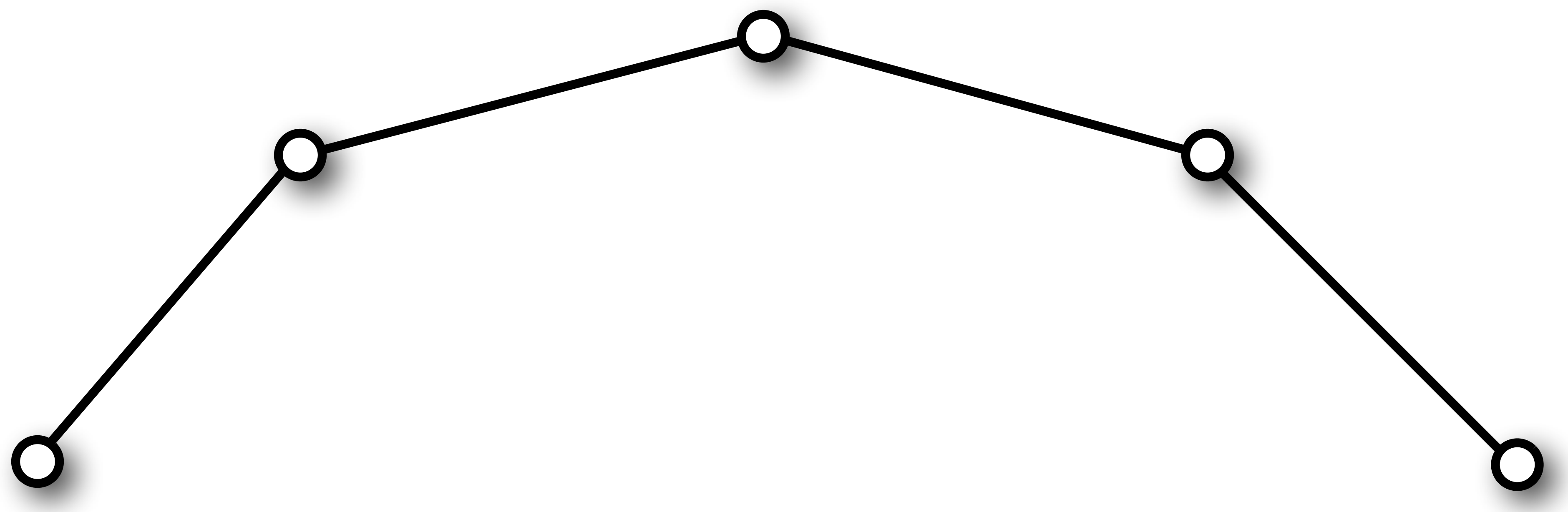
1. Parametrization Based



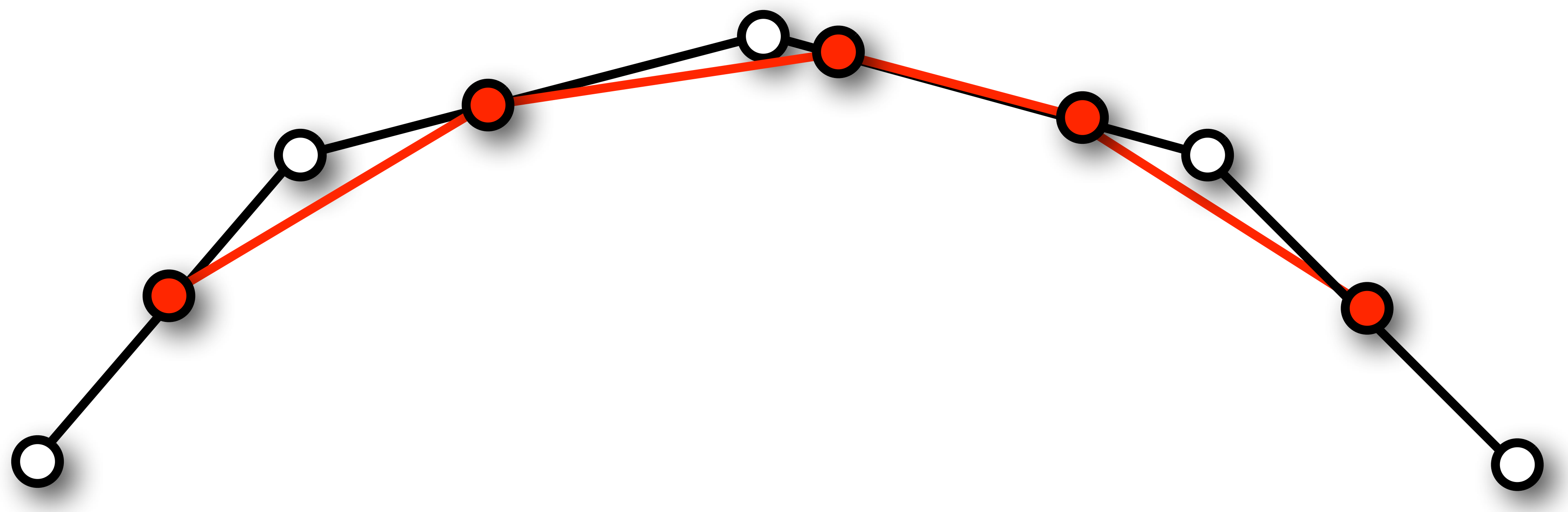
1. Parametrization Based



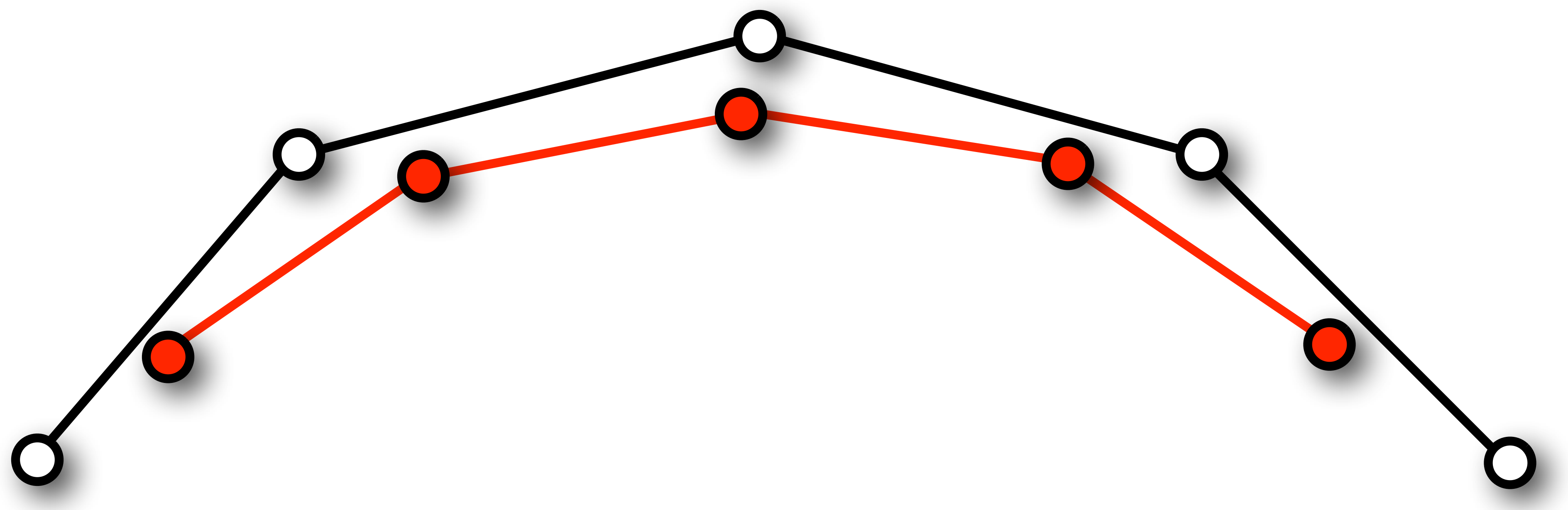
2. Surface Based



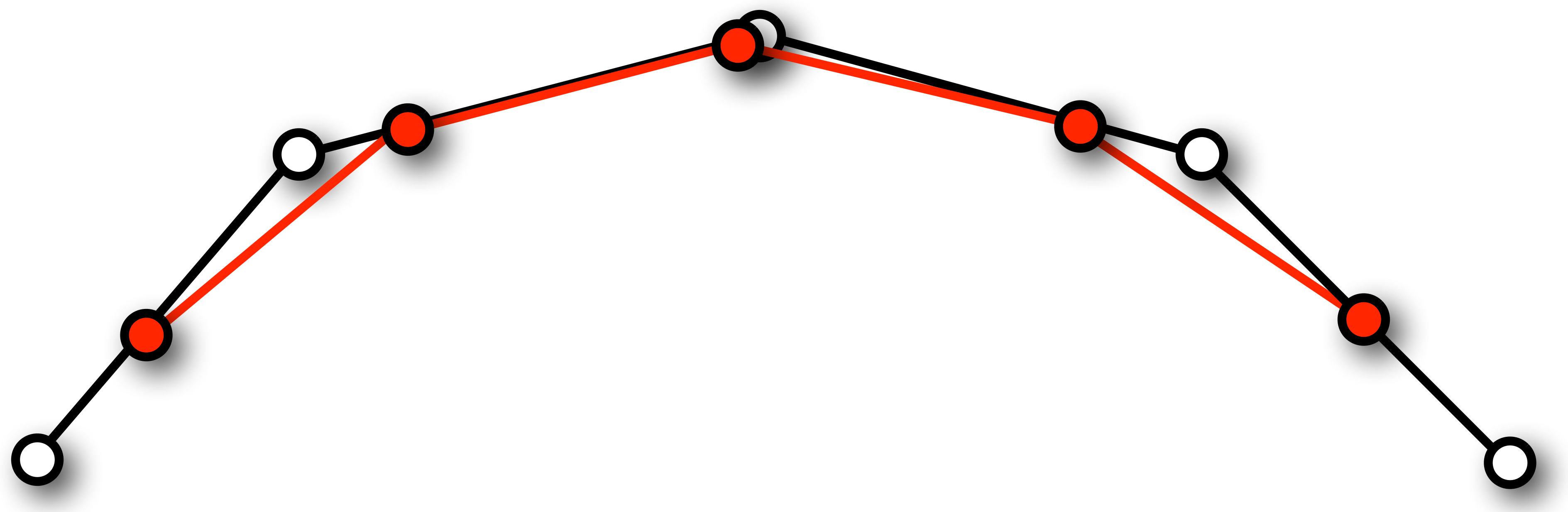
2. Surface Based



2. Surface Based



2. Surface Based

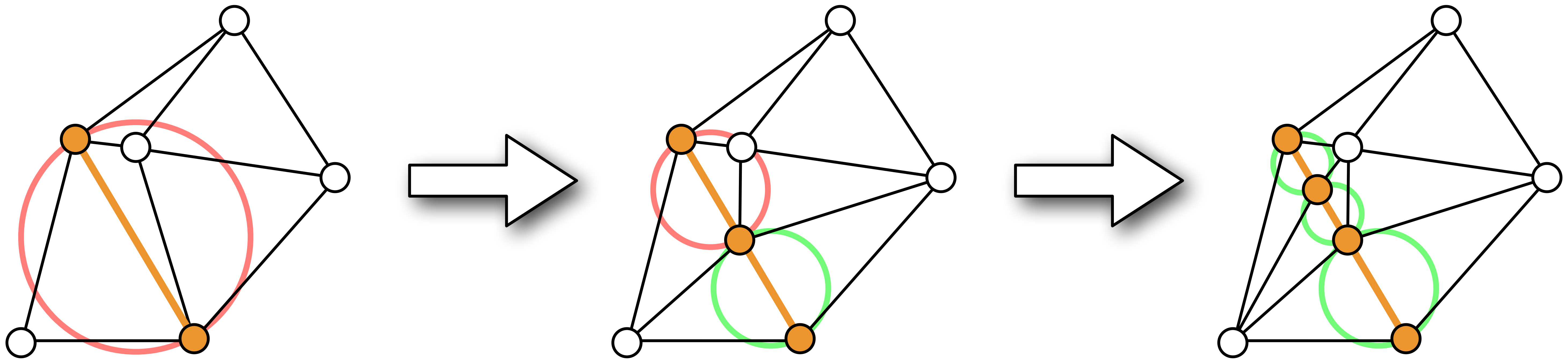


Overview

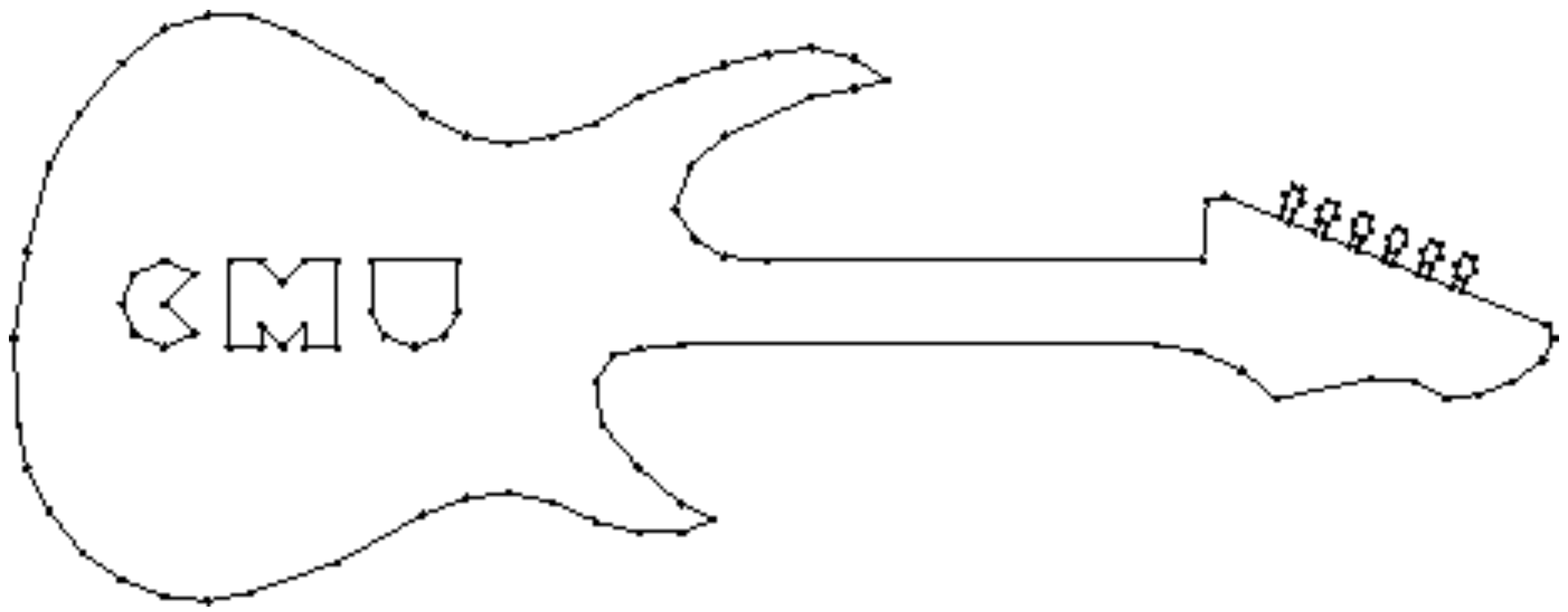
- **2D Meshing**
- Parameterization-Based Remeshing
- Surface-Based Remeshing

Constrained Delaunay Triangulation

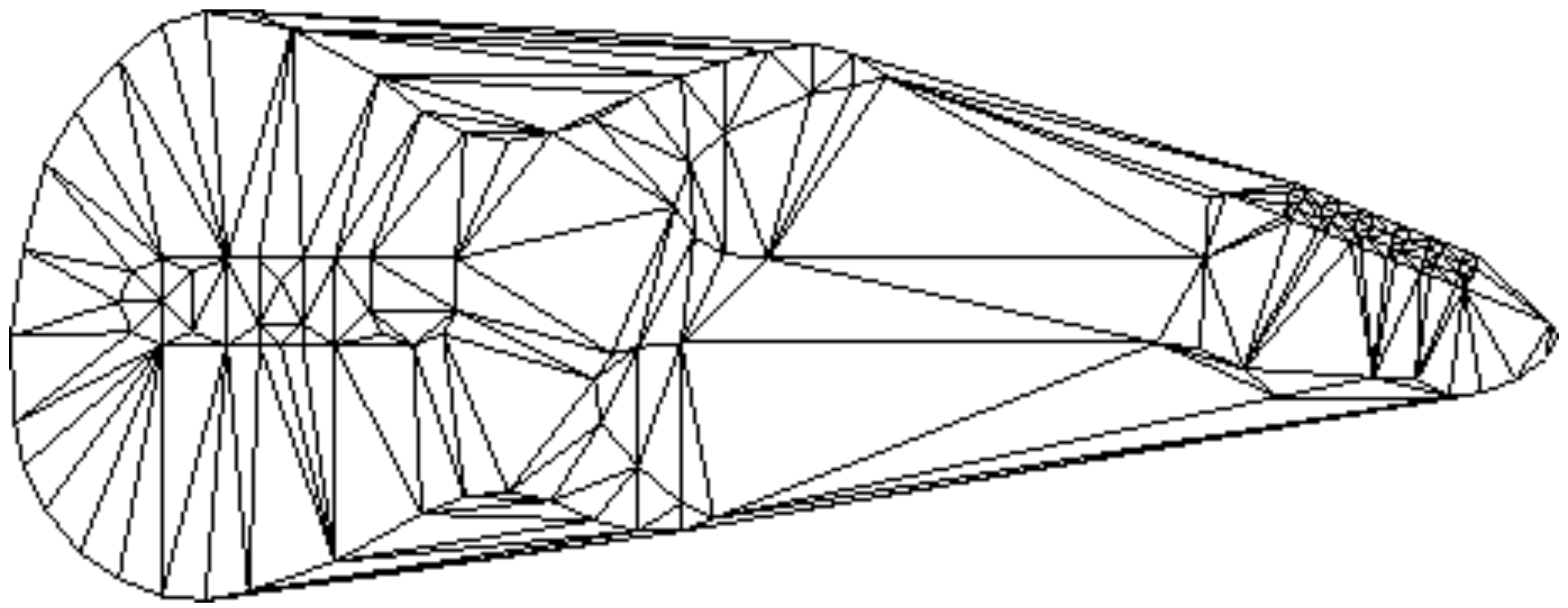
- Enforce certain edges in triangulation
  - Either prevent flipping (→ bad triangles)
  - Or subdivide edges sufficiently (→ many triangles)



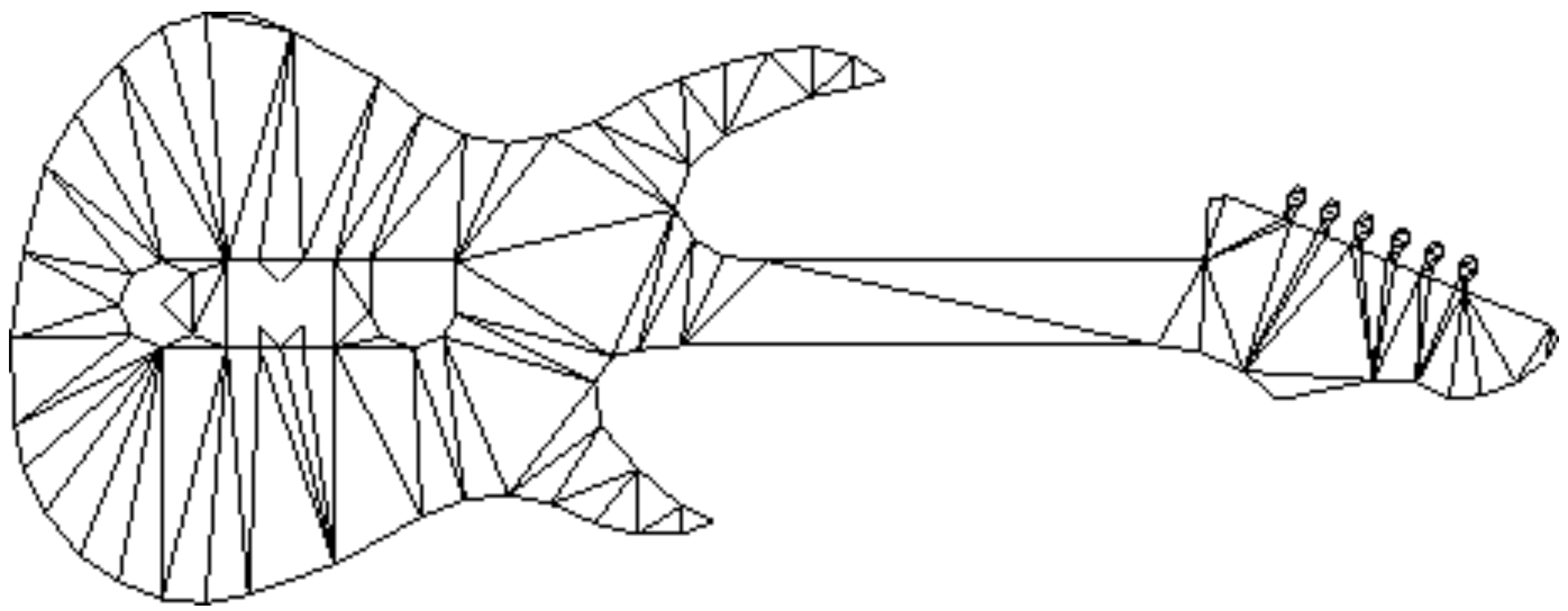
2D Meshing



Input points & edges



Constrained Delaunay Triangulation

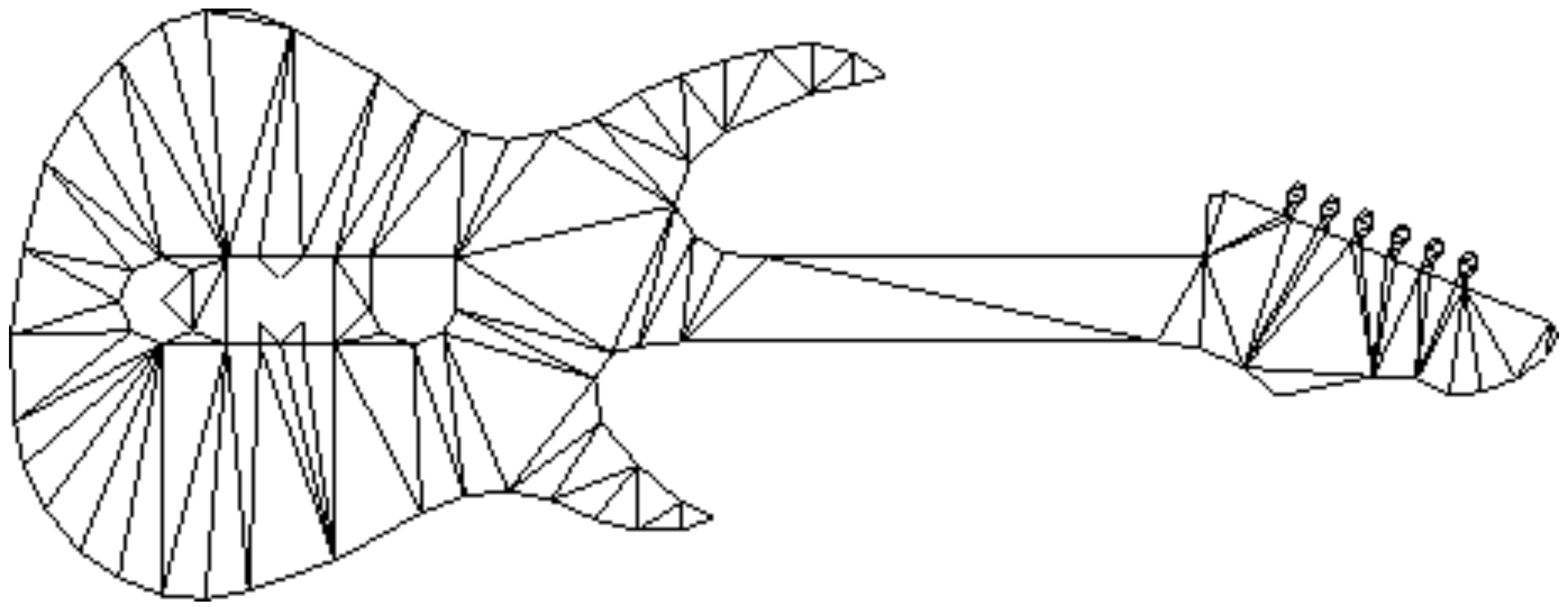


Remove exterior

[J. Shewchuck]

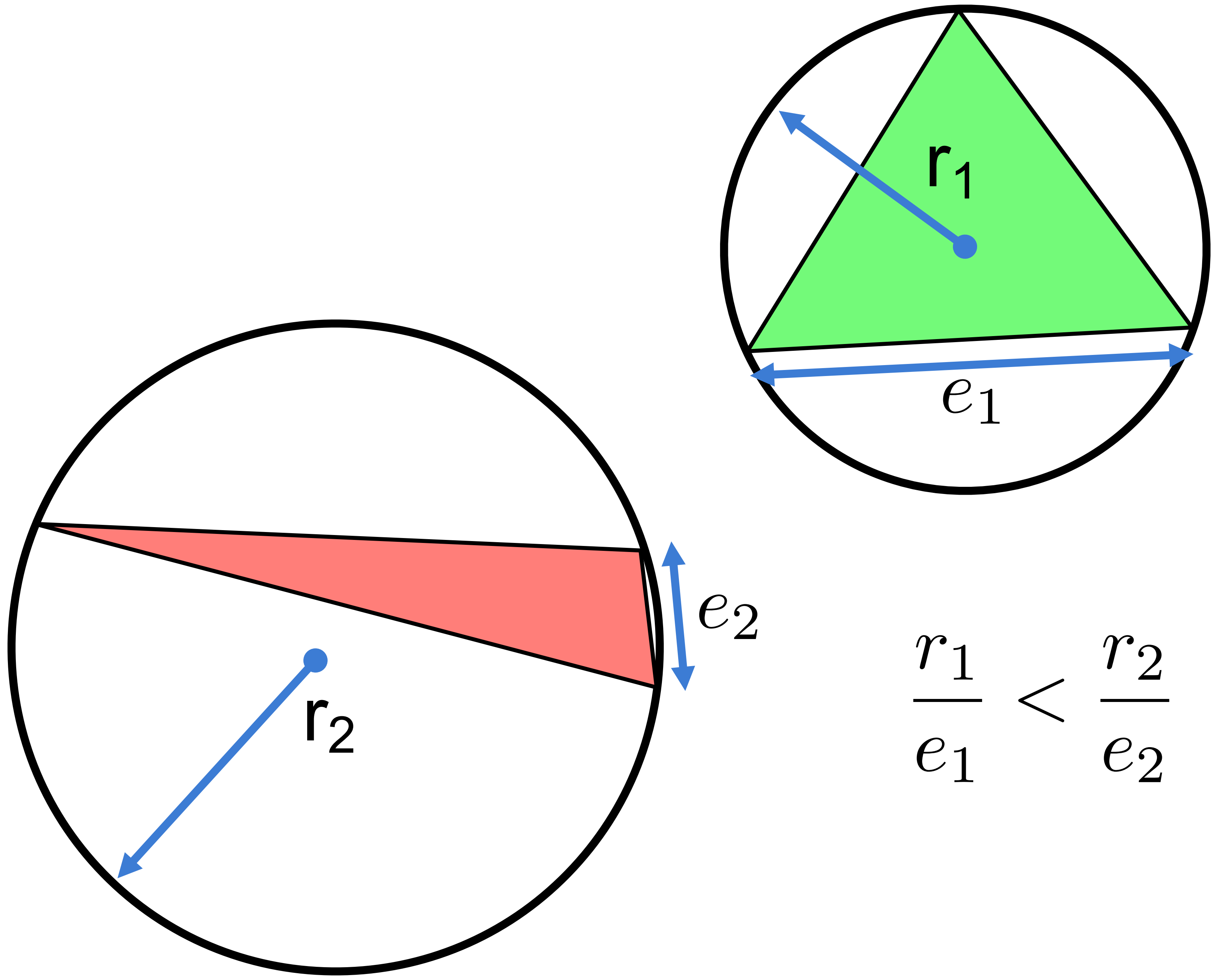
2D Meshing

- 2D Delaunay triangulation
  - *Maximizes minimum angle*
  - Optimal triangulation for given set of vertices
  - Why can there still be bad triangles?



Delaunay Refinement

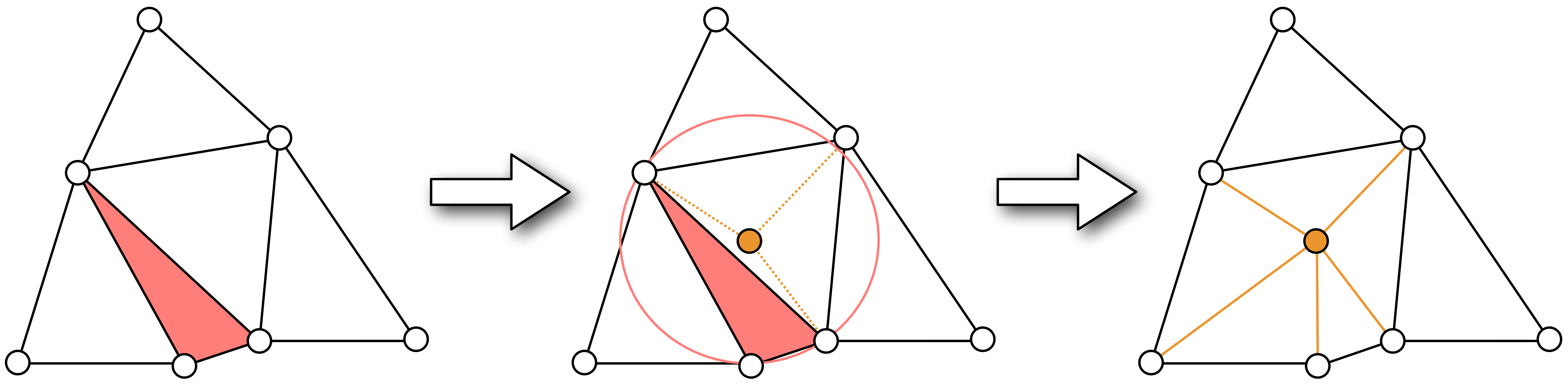
- Delaunay triangulation might contain bad triangles, depending on vertex distribution
- Refine triangulation
  - Insert new vertices, eliminate bad triangles
- Measure triangle quality by
  - circum-radius / shortest-edge
  - smallest inner angles



$$\frac{r_1}{e_1} < \frac{r_2}{e_2}$$

Delaunay Refinement

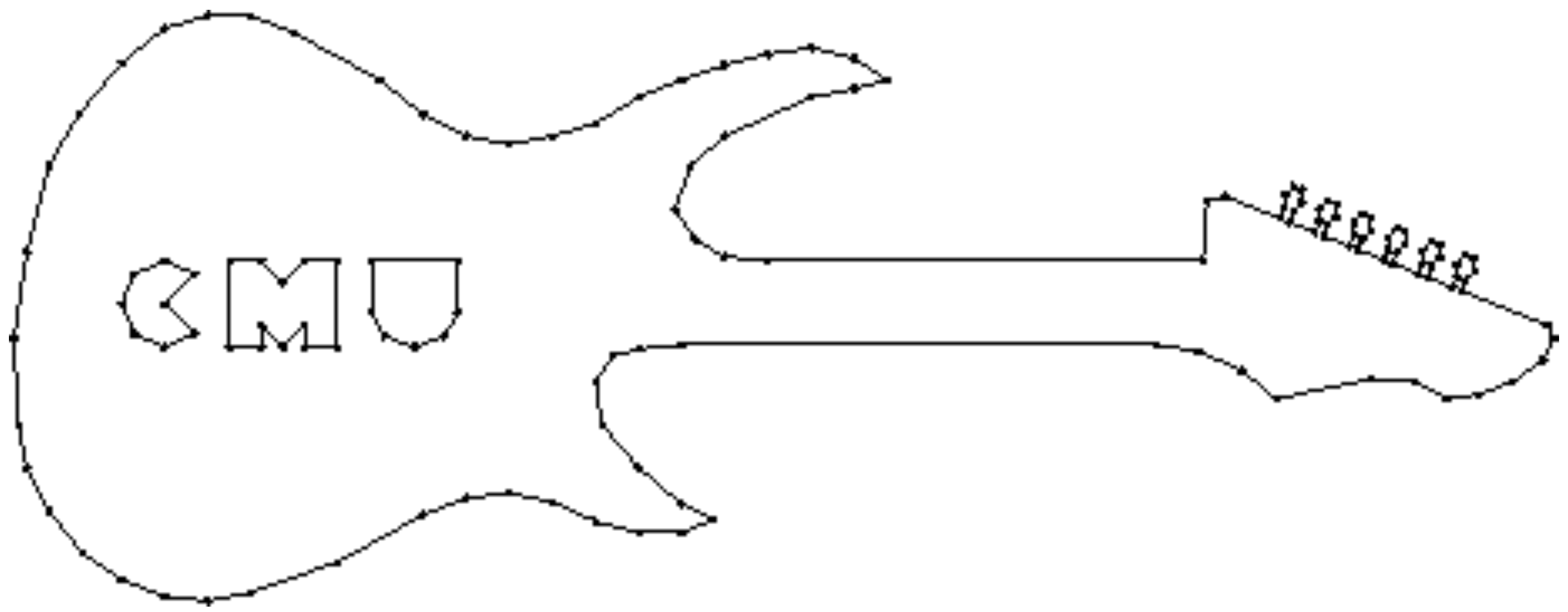
- Eliminate “bad” triangles by inserting their circum-center into the triangulation
- Bad triangle will fail the empty-circumcircle test and will therefore be removed



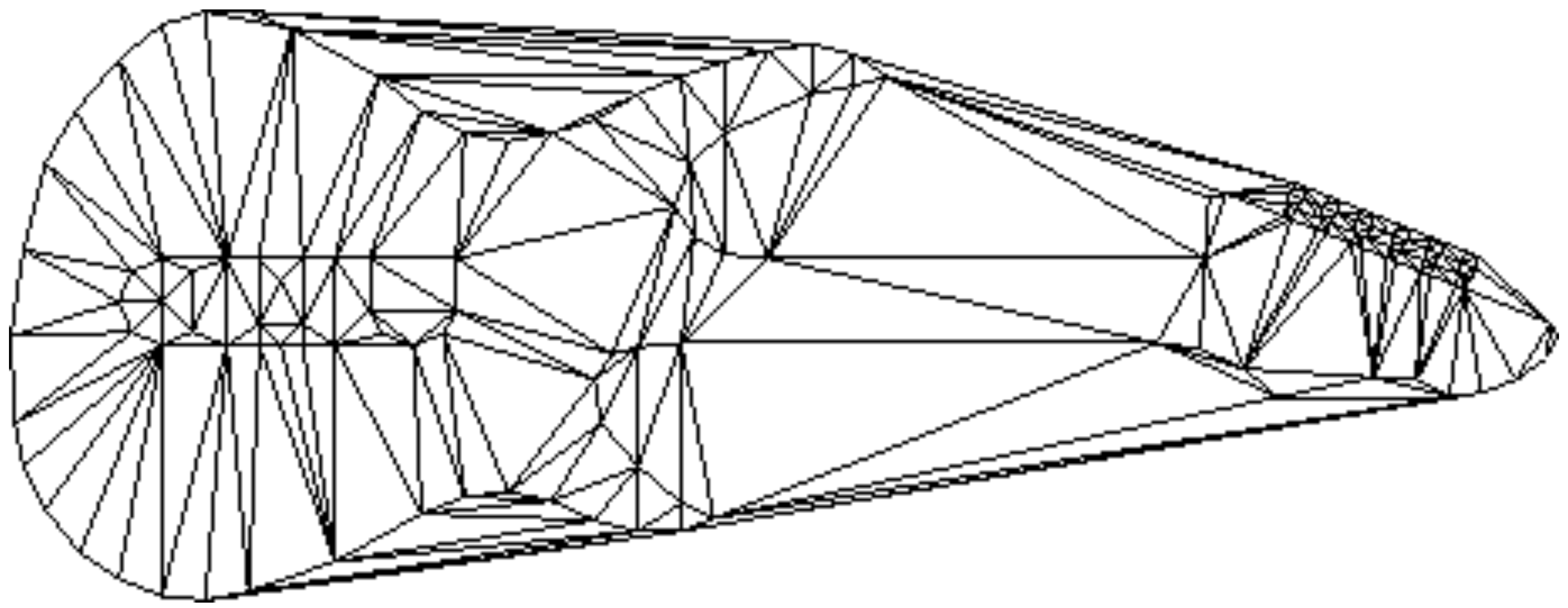
Delaunay Refinement

```
while (!finished)
{
  T = find worst triangle
  if T is too bad
    insert T's circum-center
  else
    finished = true
}
```

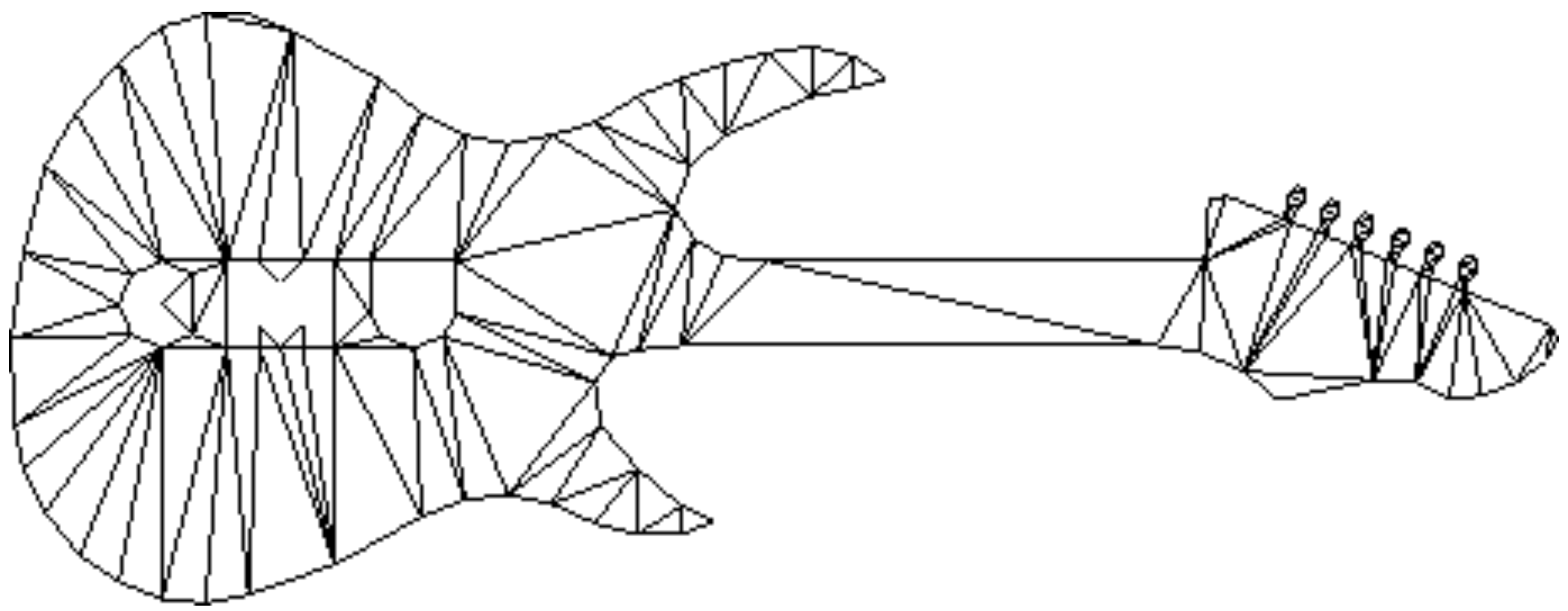
2D Meshing



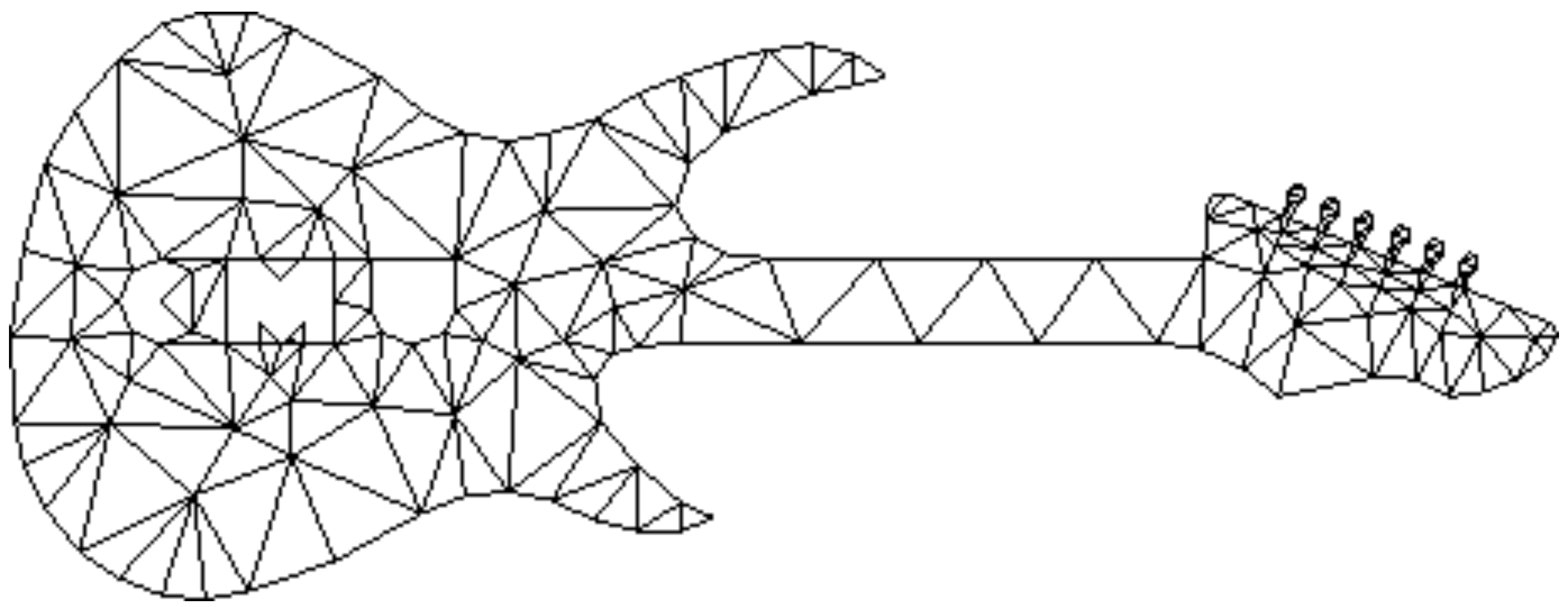
Input points & edges



Constrained Delaunay Triangulation



Remove exterior



Delaunay Refinement

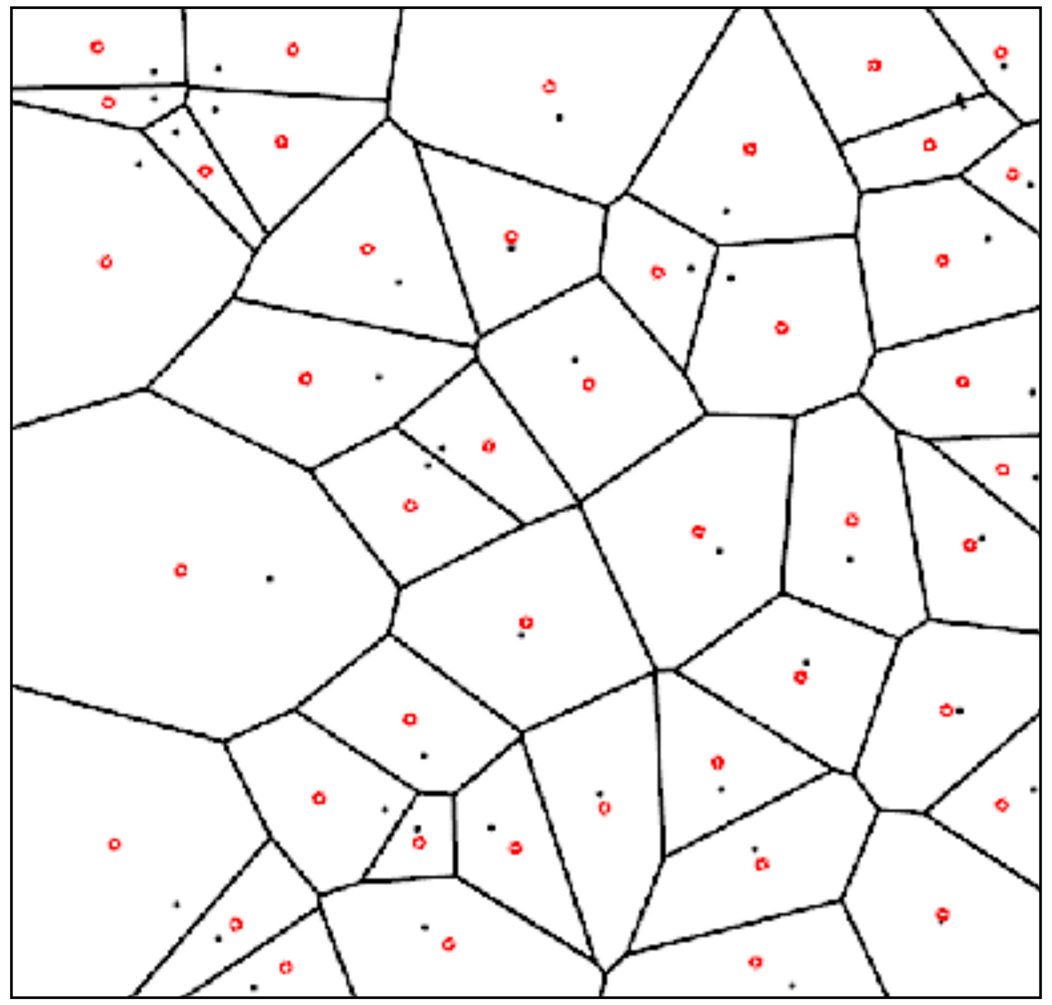
[J. Shewchuck]

DT/VD

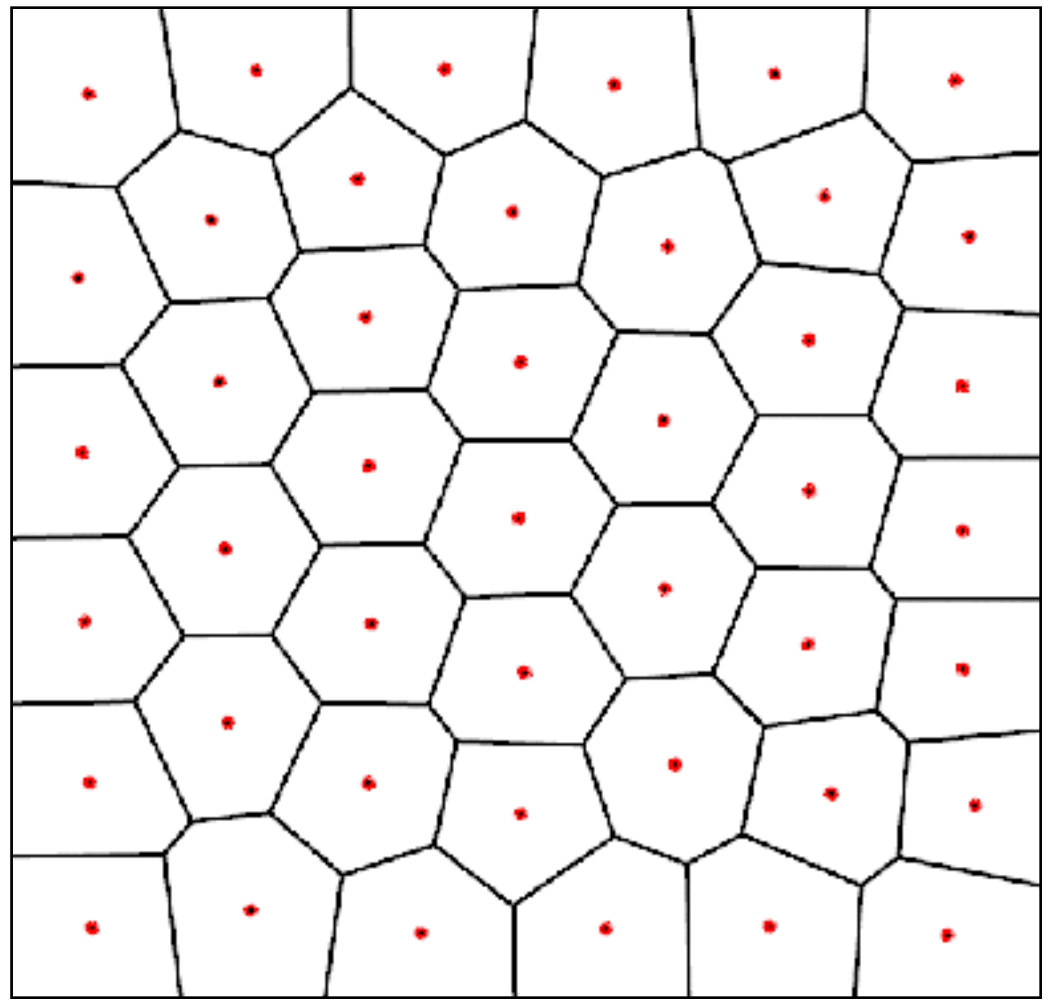
DT/VD

Centroidal Voronoi Diagrams

- How can we get a more *regular* triangulation?
  - *Delaunay Triangulation* (DT) of *Centroidal Voronoi Diagram* (CVD)
- Definition:
  - All points are centroids of their Voronoi cells



VD



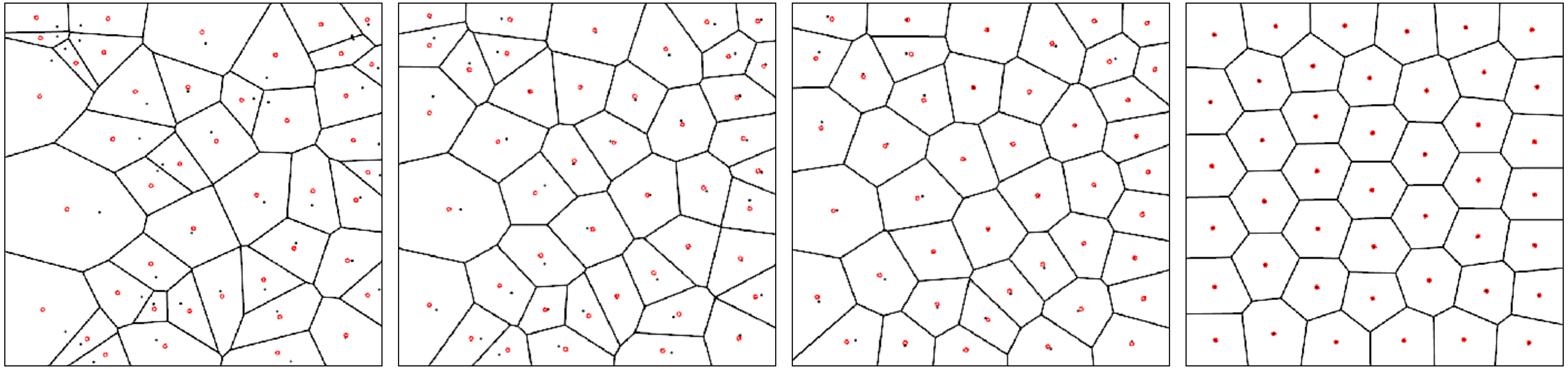
CVD

Centroidal Voronoi Diagrams

Compute CVD by iterative ***Lloyd relaxation***

1. Compute Voronoi diagram of given point set $\{\mathbf{p}_i\}$
2. Move $\mathbf{p}_i$ to centroids $\mathbf{c}_i$ of their Voronoi cells V_i

$$\mathbf{p}_i \leftarrow \mathbf{c}_i = \frac{\int_{V_i} \mathbf{x} \cdot \rho(\mathbf{x}) \, \mathrm{d}\mathbf{x}}{\int_{V_i} \rho(\mathbf{x}) \, \mathrm{d}\mathbf{x}}$$



Centroidal Voronoi Diagrams

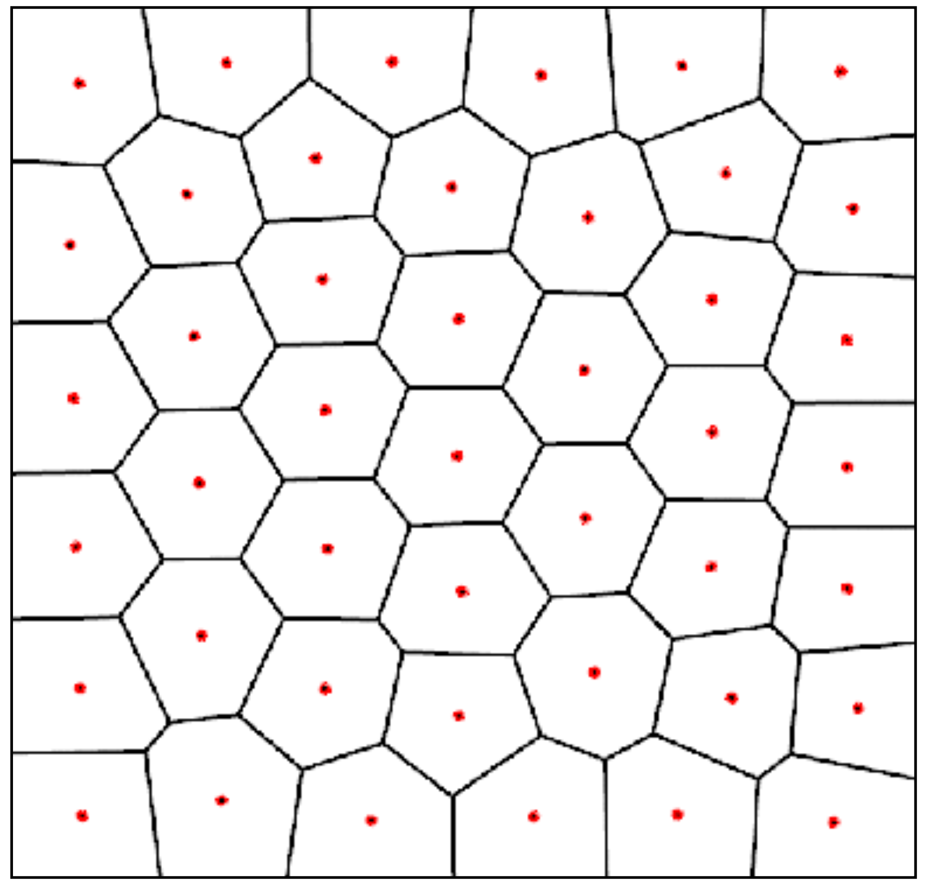
Compute CVD by iterative ***Lloyd relaxation***

1. Compute Voronoi diagram of given points $\mathbf{p}_i$
2. Move $\mathbf{p}_i$ to centroids $\mathbf{c}_i$ of their Voronoi cells V_i

$$\mathbf{p}_i \leftarrow \mathbf{c}_i = \frac{\int_{V_i} \mathbf{x} \cdot \rho(\mathbf{x})\ \mathrm{d}\mathbf{x}}{\int_{V_i} \rho(\mathbf{x})\ \mathrm{d}\mathbf{x}}$$

- CVD maximizes compactness
  - Minimizes this energy

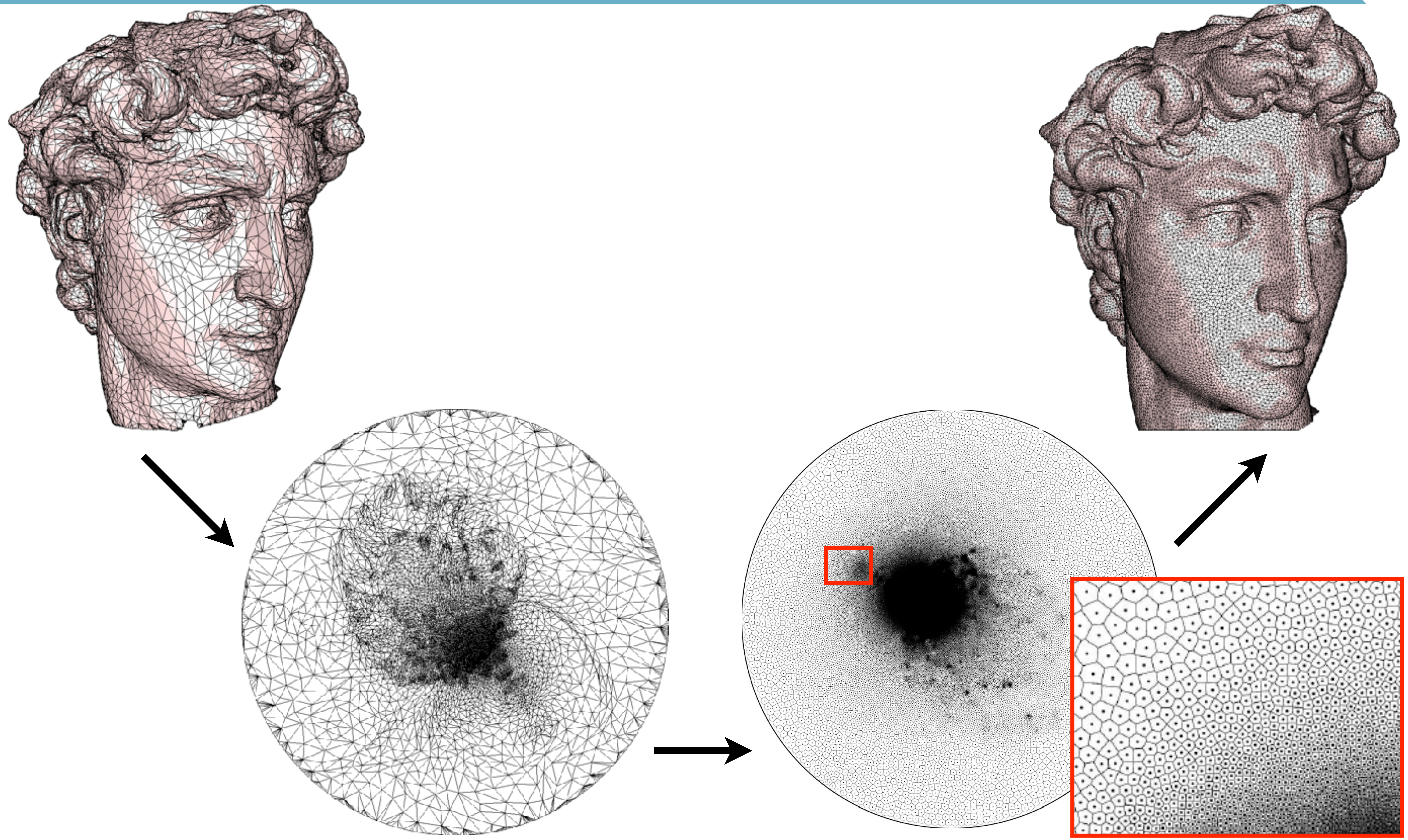
$$\sum_i \int_{V_i} \rho(\mathbf{x}) \left\|\mathbf{x} - \mathbf{p}_i\right\|^2 \mathrm{d}\mathbf{x} \rightarrow \min$$



Overview

- 2D Meshing
- **Parameterization-Based Remeshing**
- Surface-Based Remeshing

Parameterization-Based Remeshing



[Alliez et al., *"Isotropic surface remeshing"*, SMI 2003]

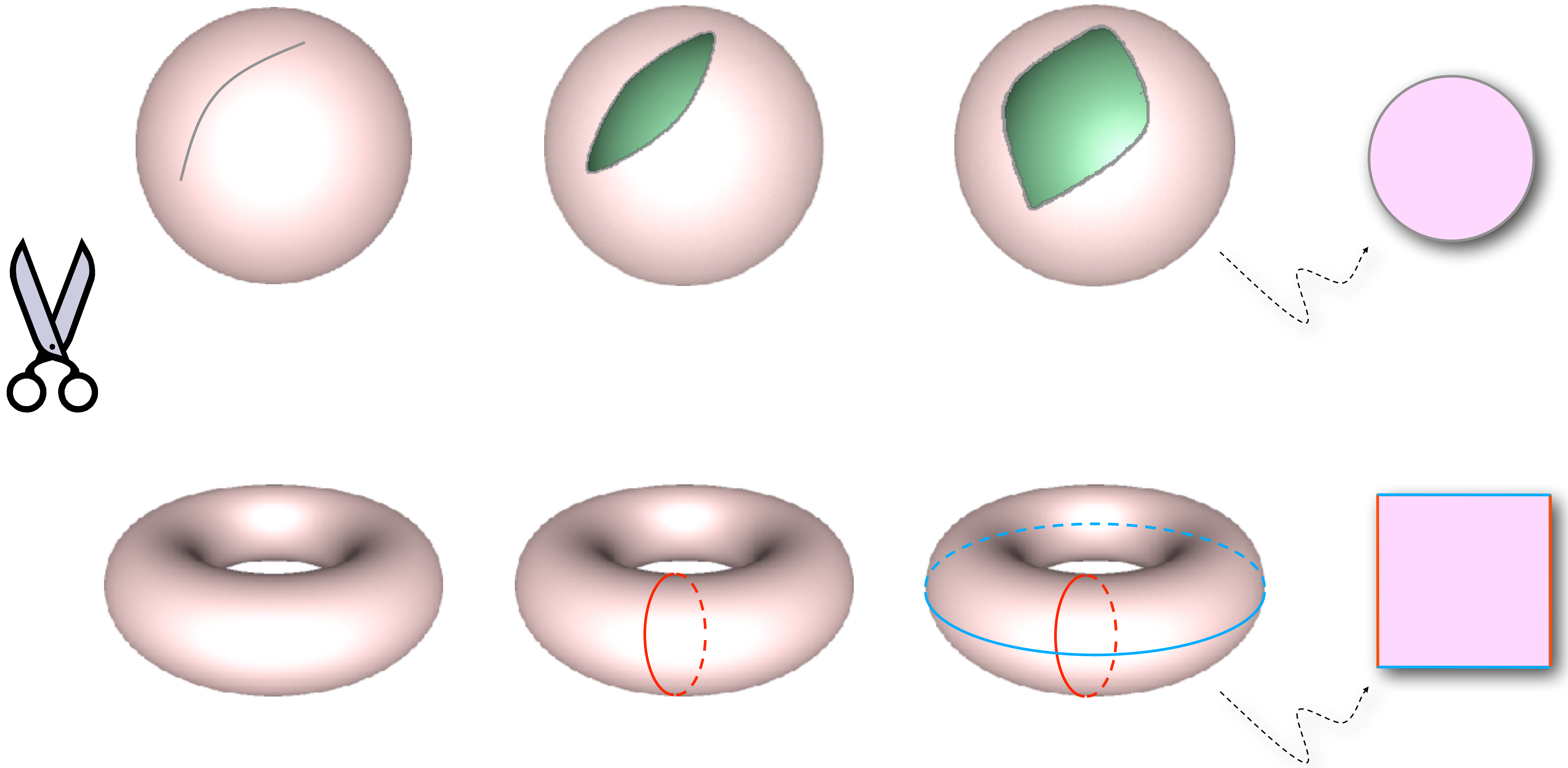
Use a Global Parameterization

- **Motivation:** 2D remeshing is much easier
  - Distribution of sample points
  - Centroidal Voronoi diagram
  - Delaunay triangulation

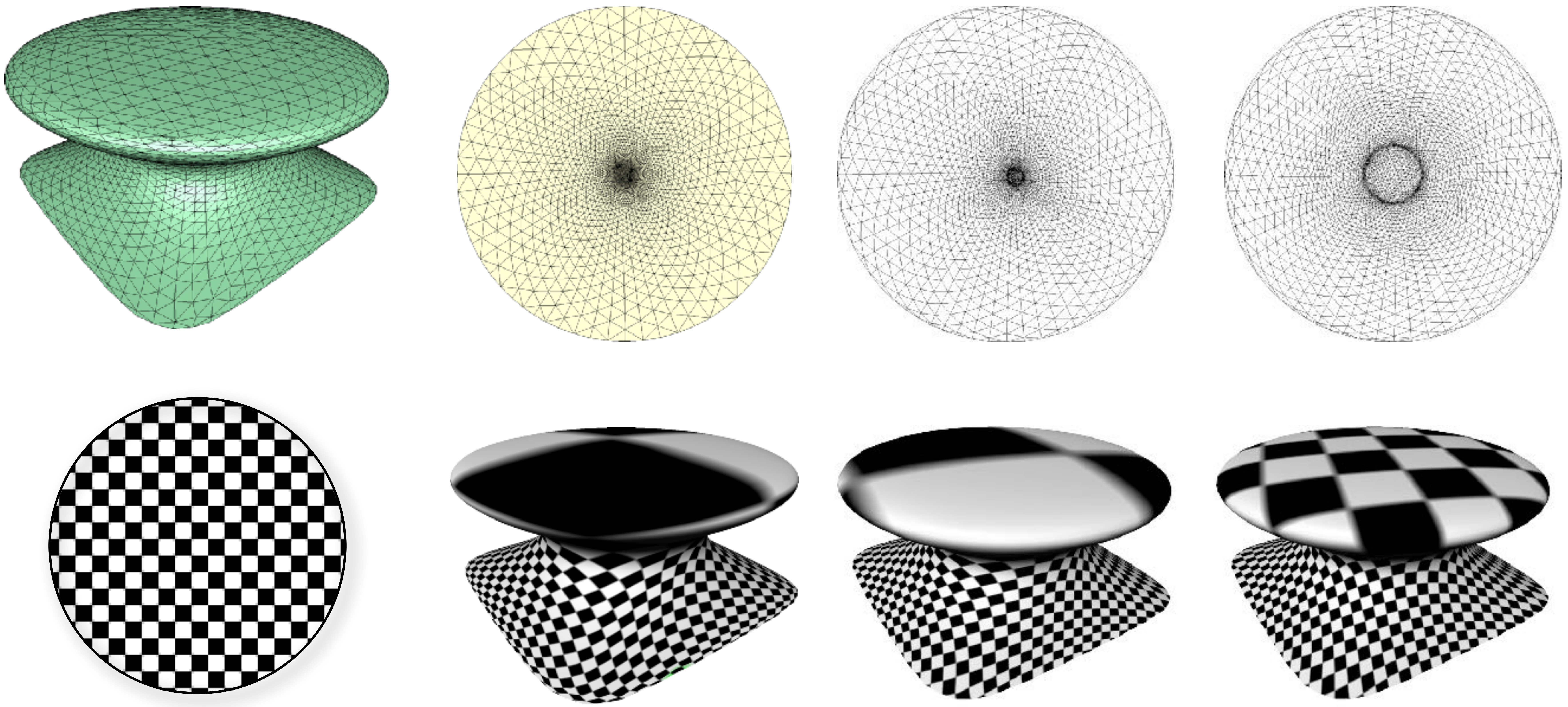
- Which parameterization method to use?
  - Minimize distortion (conformal, isometric, …)
  - Cut path for higher genus models
  - Free versus fixed boundary

Need disk-like topology

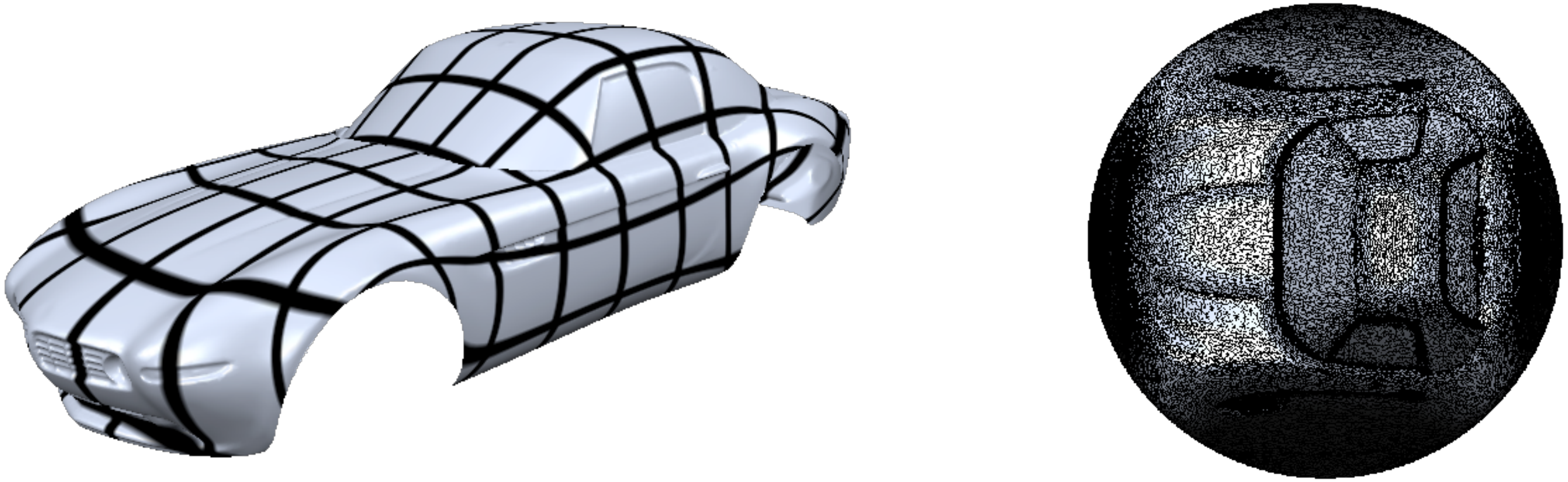
- Introduce cuts on the mesh

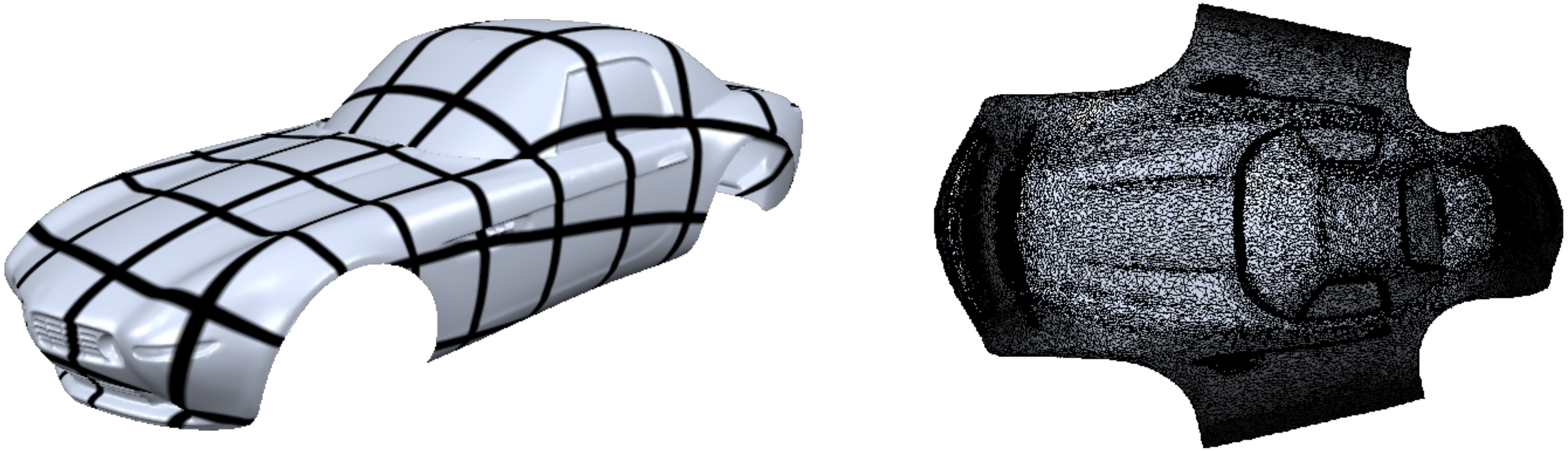


Parameterization Methods



Fixed vs. Free Boundary



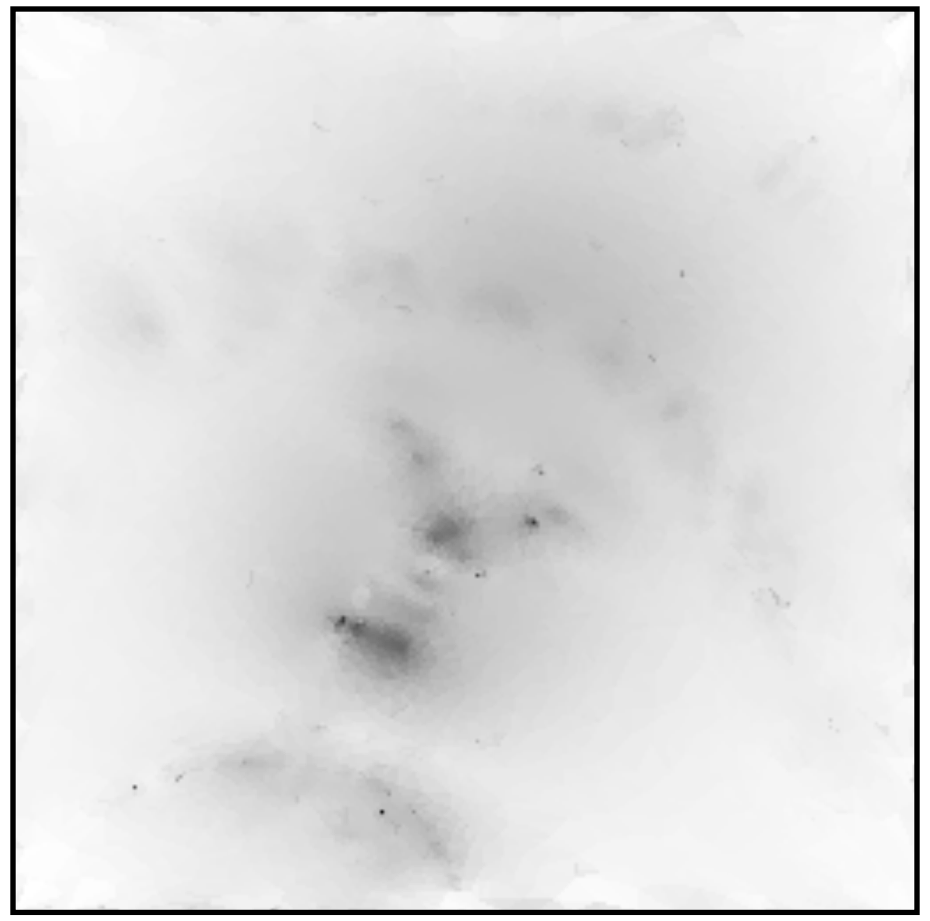


Initial Sampling

- Randomly sample 2D triangles
  - Weighted by triangle area and density ρ
  - **Density**: curvature or user-defined sizing field
- Compensate area distortion when sampling in the parameter domain
  - Distortion = triangle_area3D / triangle_area2D

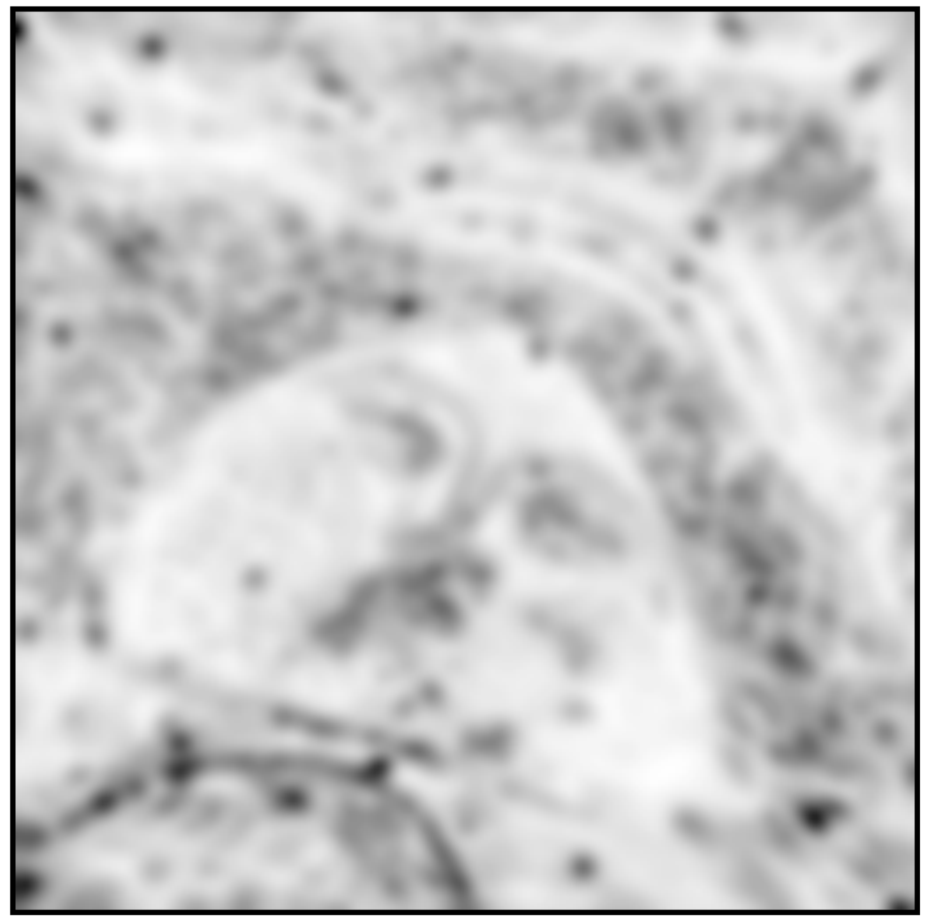
Initial Sampling

- Compose importance map



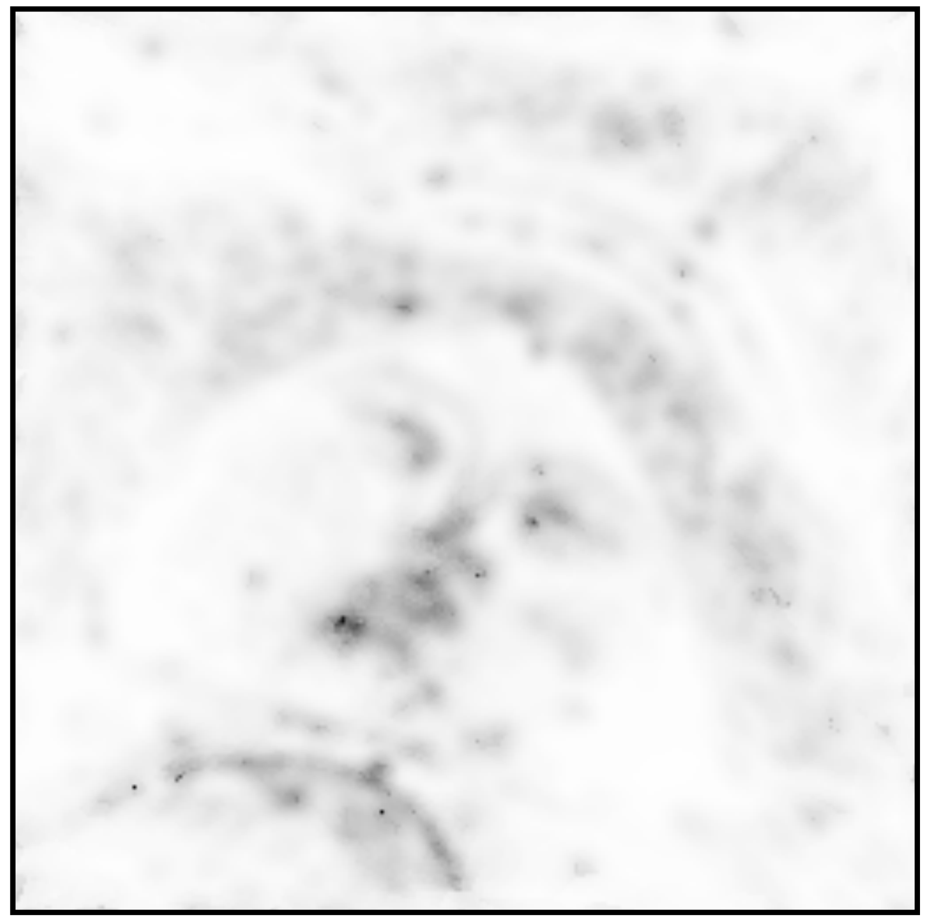
Area stretch

×



Mean curvature

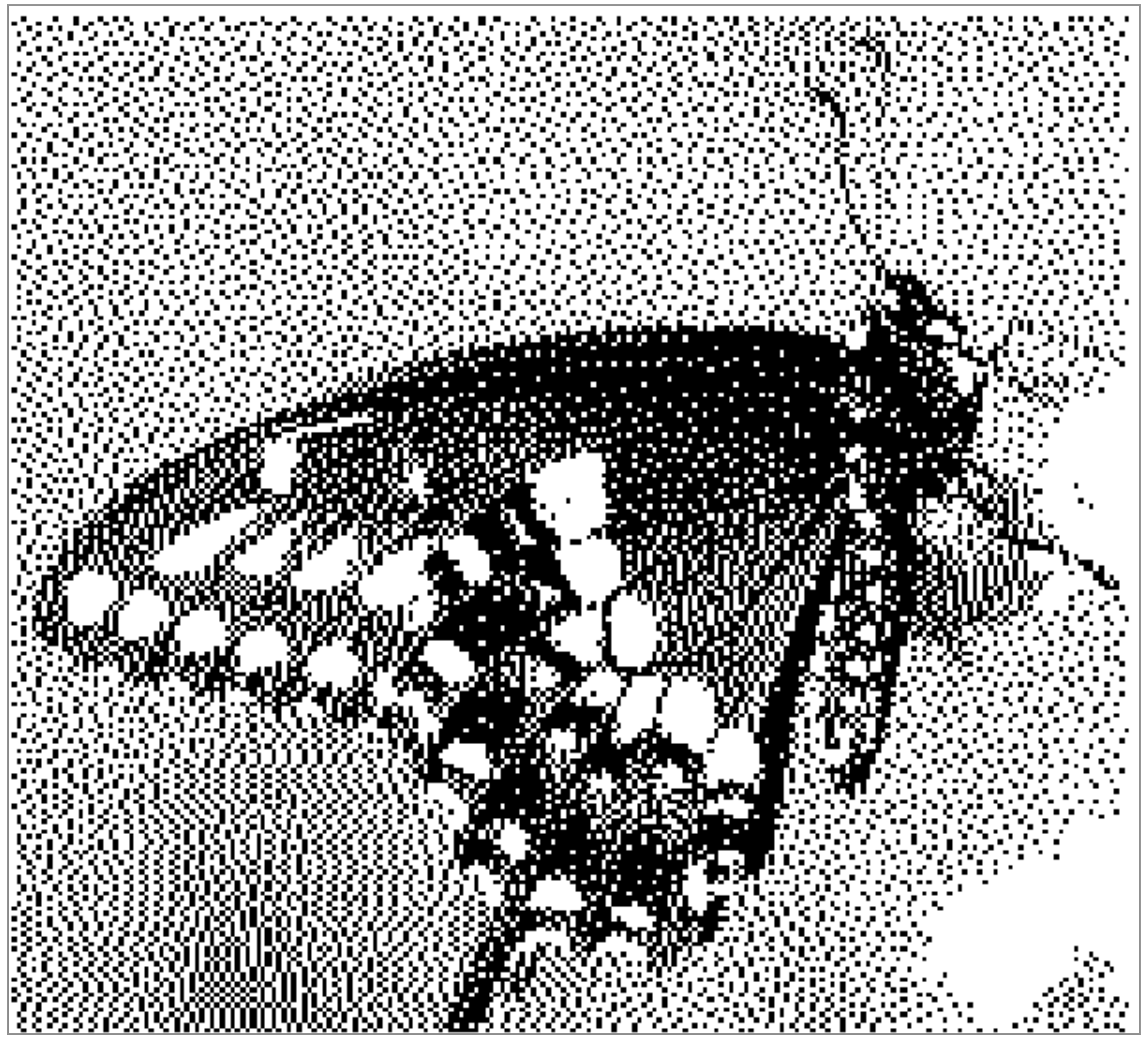
=



Importance map

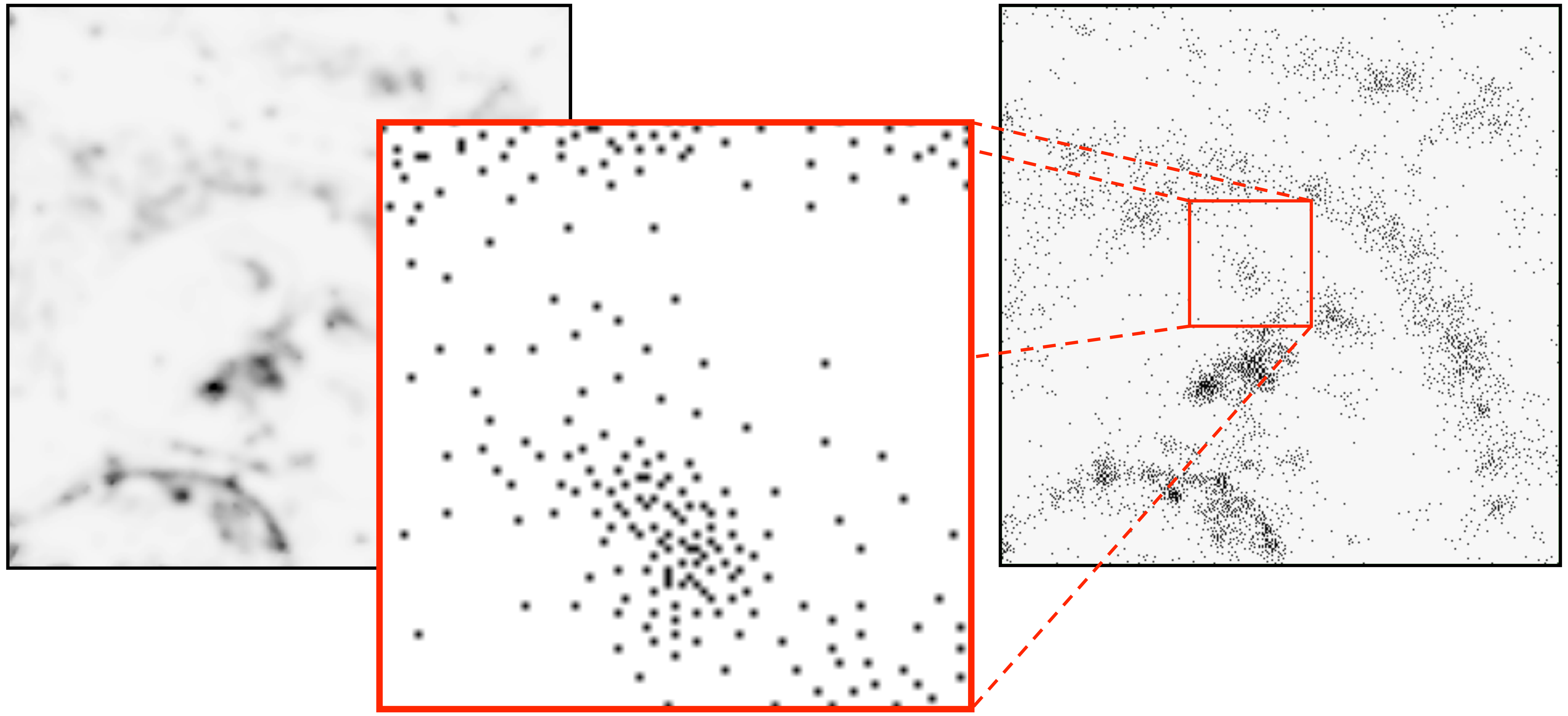
Initial Sampling

- 2D error diffusion on importance map
  - Use half-toning or dithering techniques
  - “Print grey images with B/W printer”



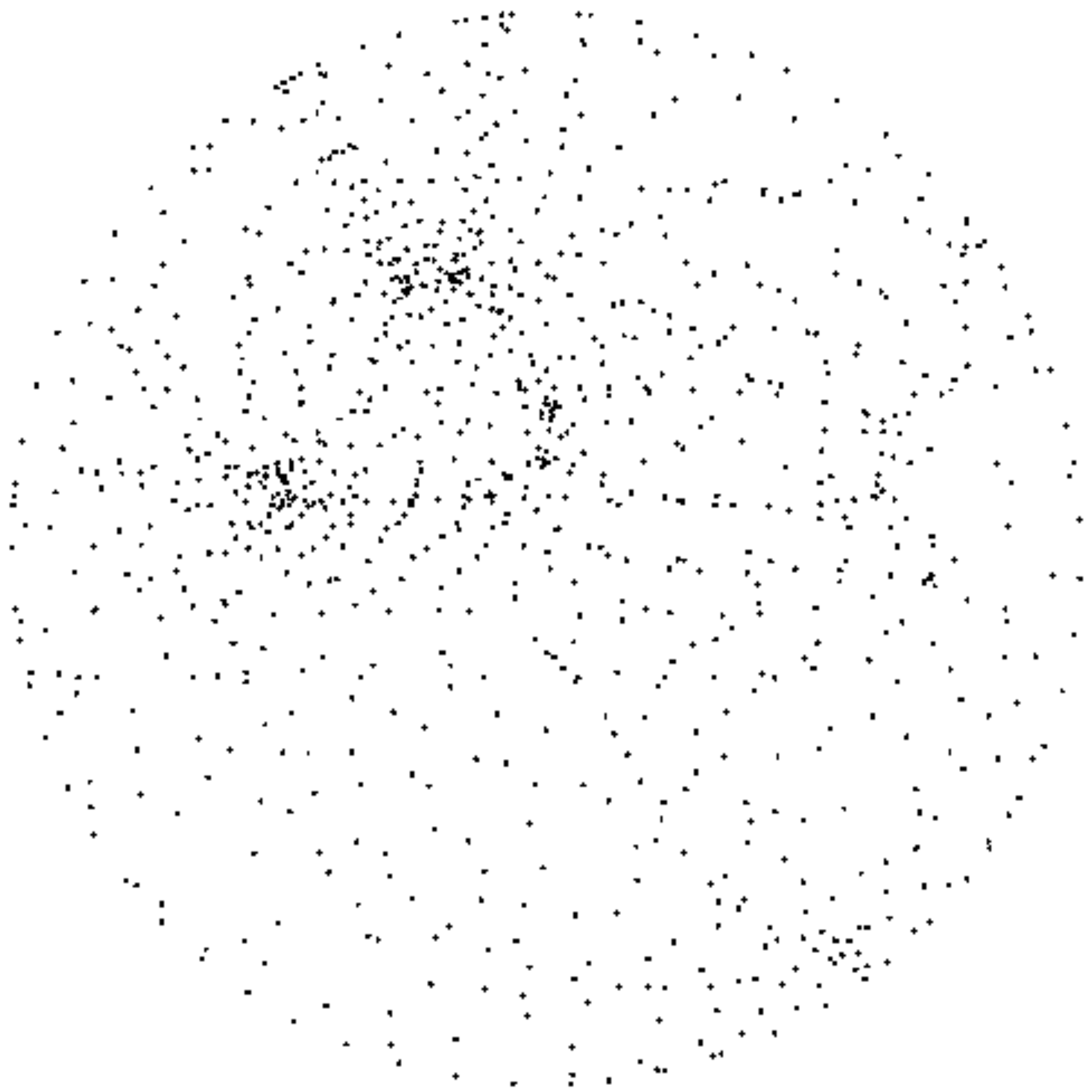
Initial Sampling

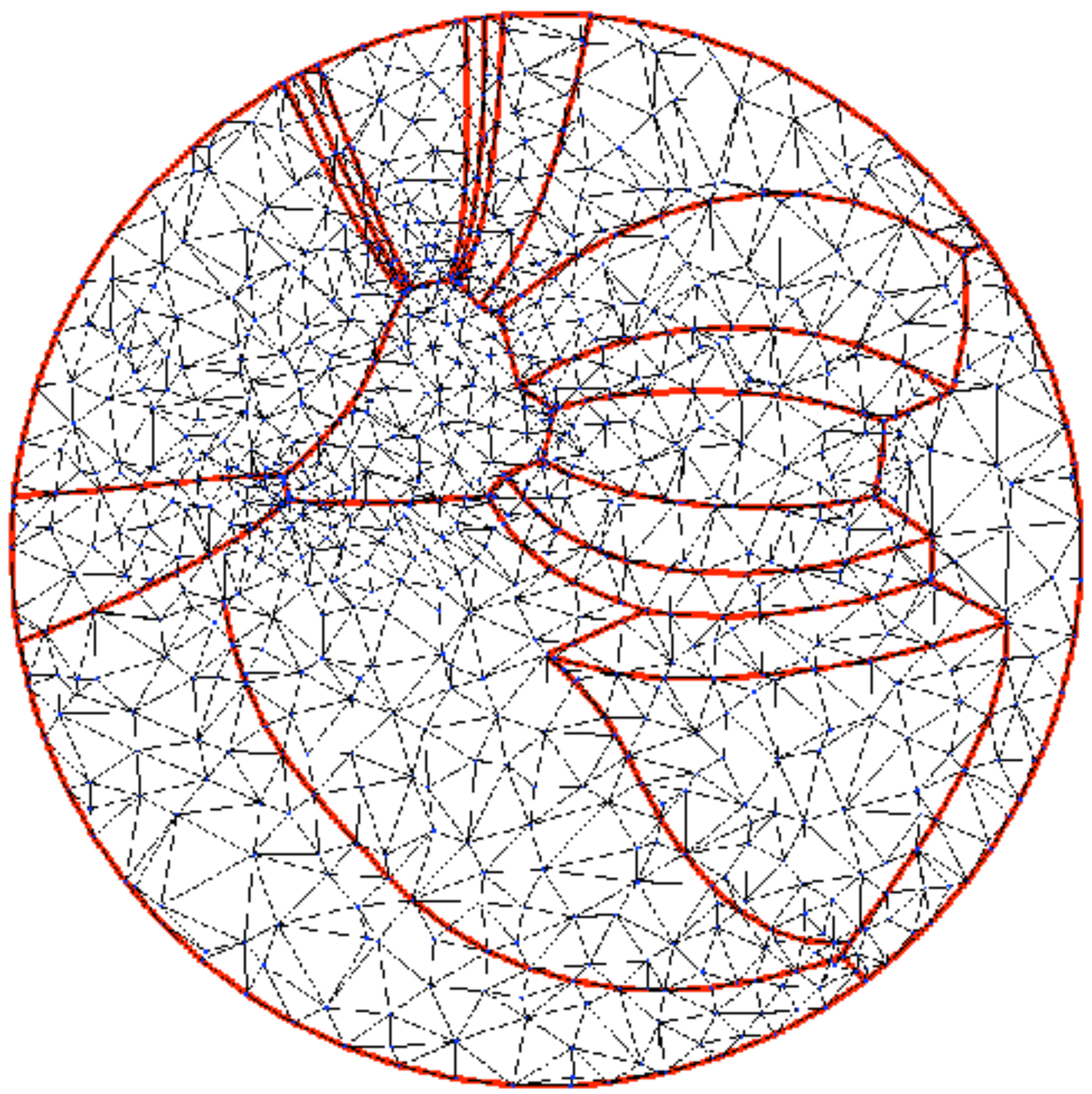
- 2D error diffusion on importance map
  - Use half-toning or dithering techniques
  - “Print grey images with B/W printer”



Initial Meshing

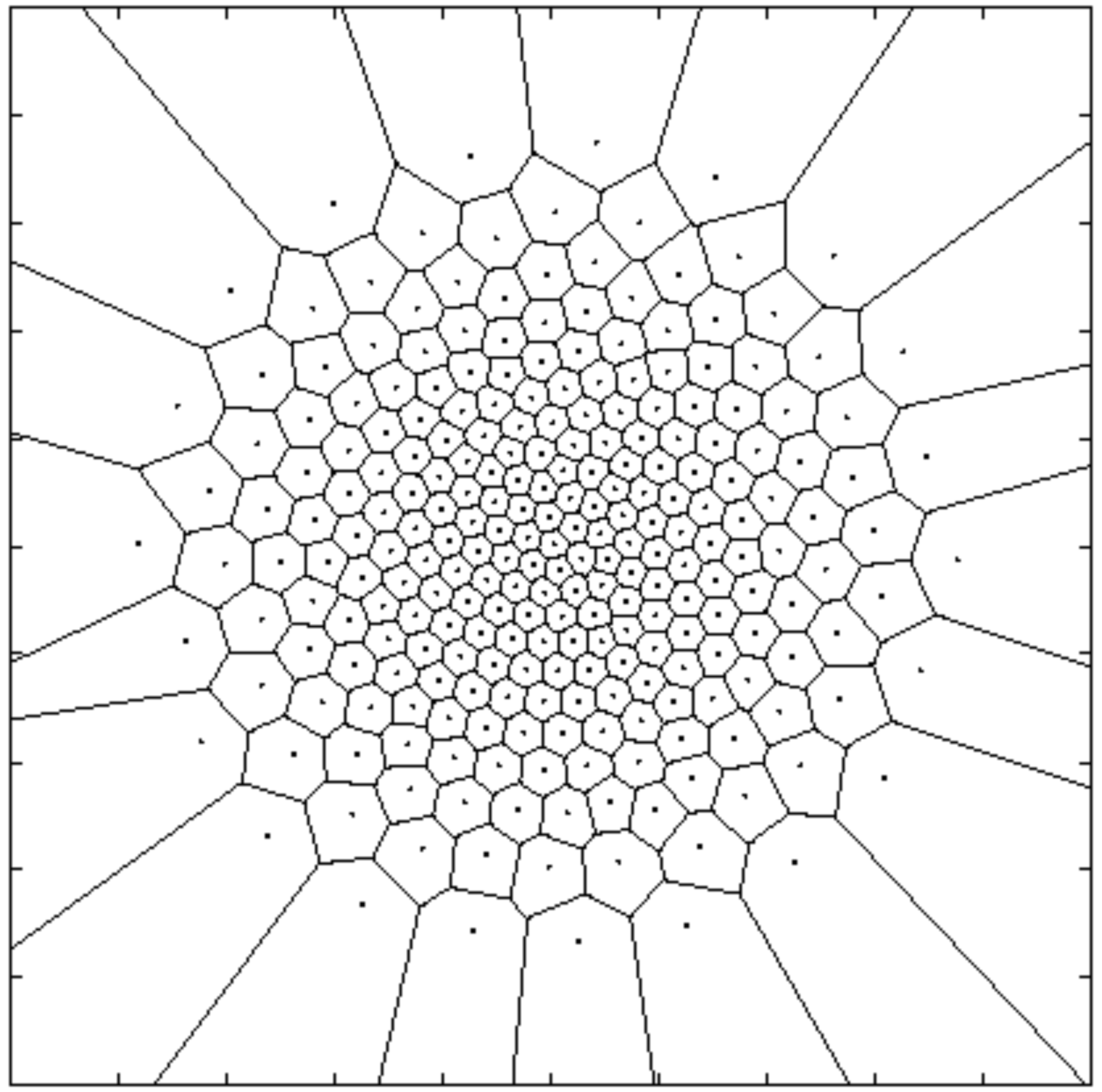
- 2D ***constrained*** Delaunay triangulation
  - Preserve feature edges / vertices
  - CGAL library provides robust implementation



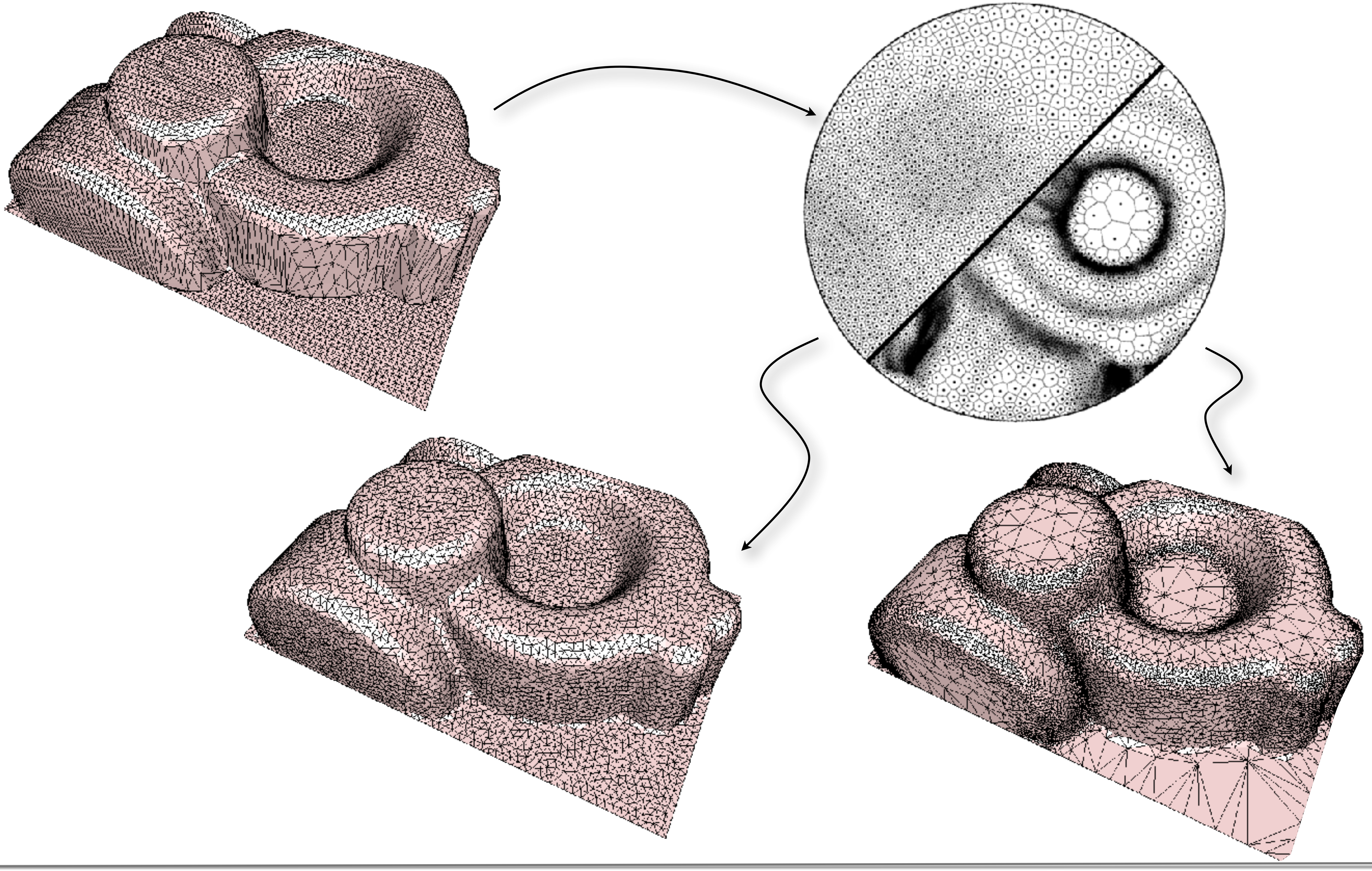


Optimize Sample Distribution

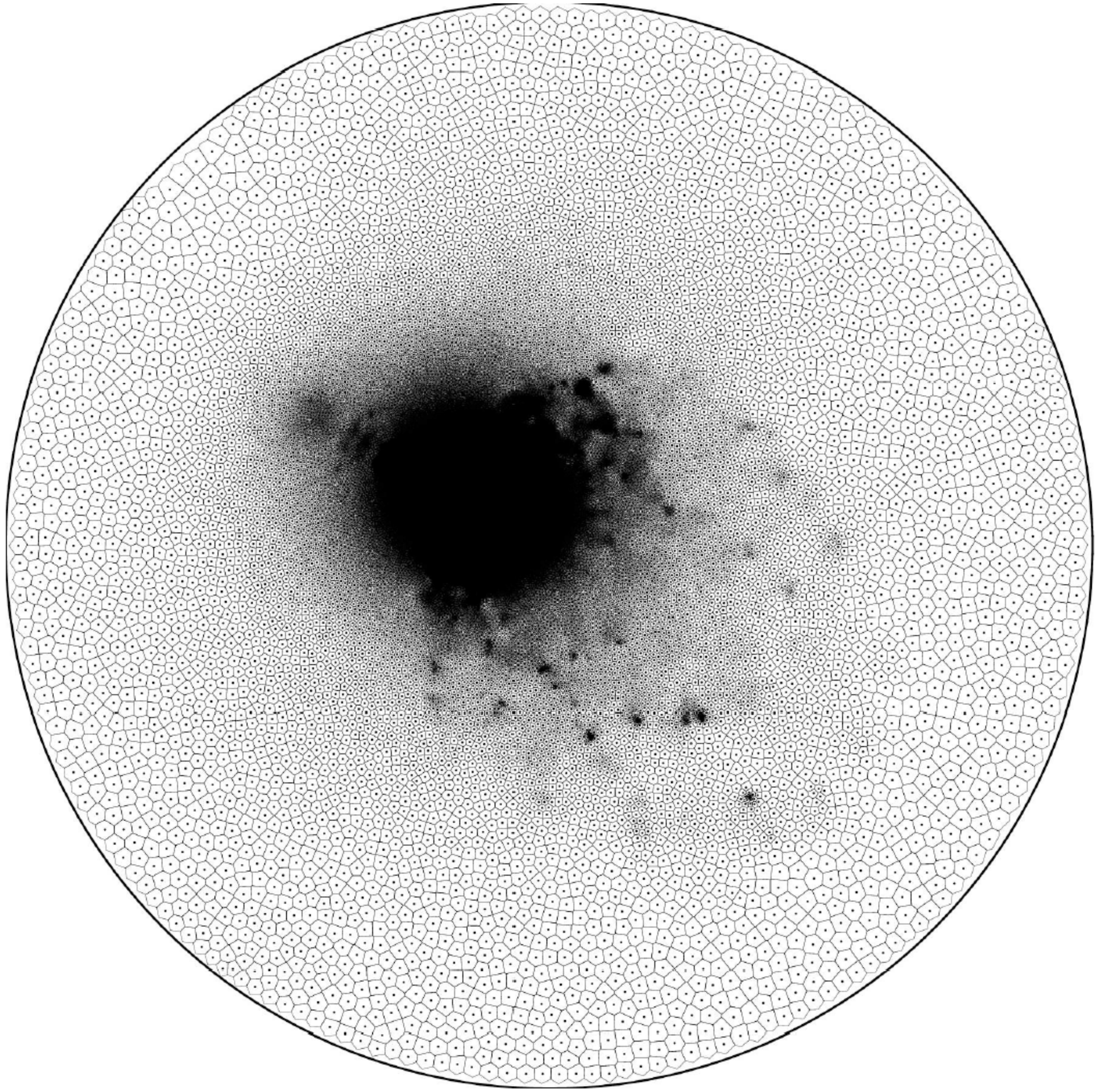
- Density-weighted centroidal Voronoi diagram
  - Equal mass enclosing
  - Tiles as compact as possible
  - Highly isotropic sampling
  - Lloyd clustering



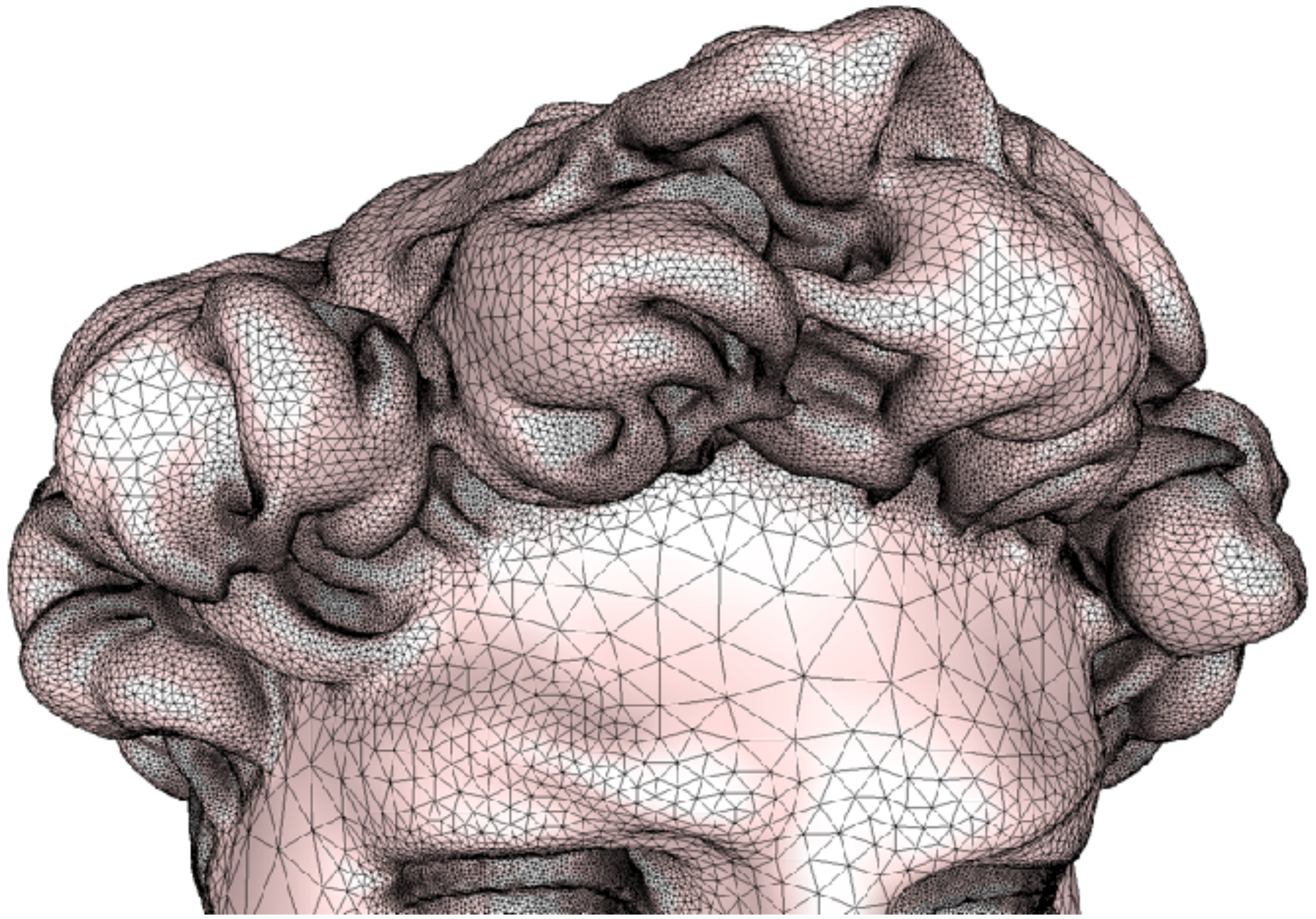
Uniform vs. Adaptive

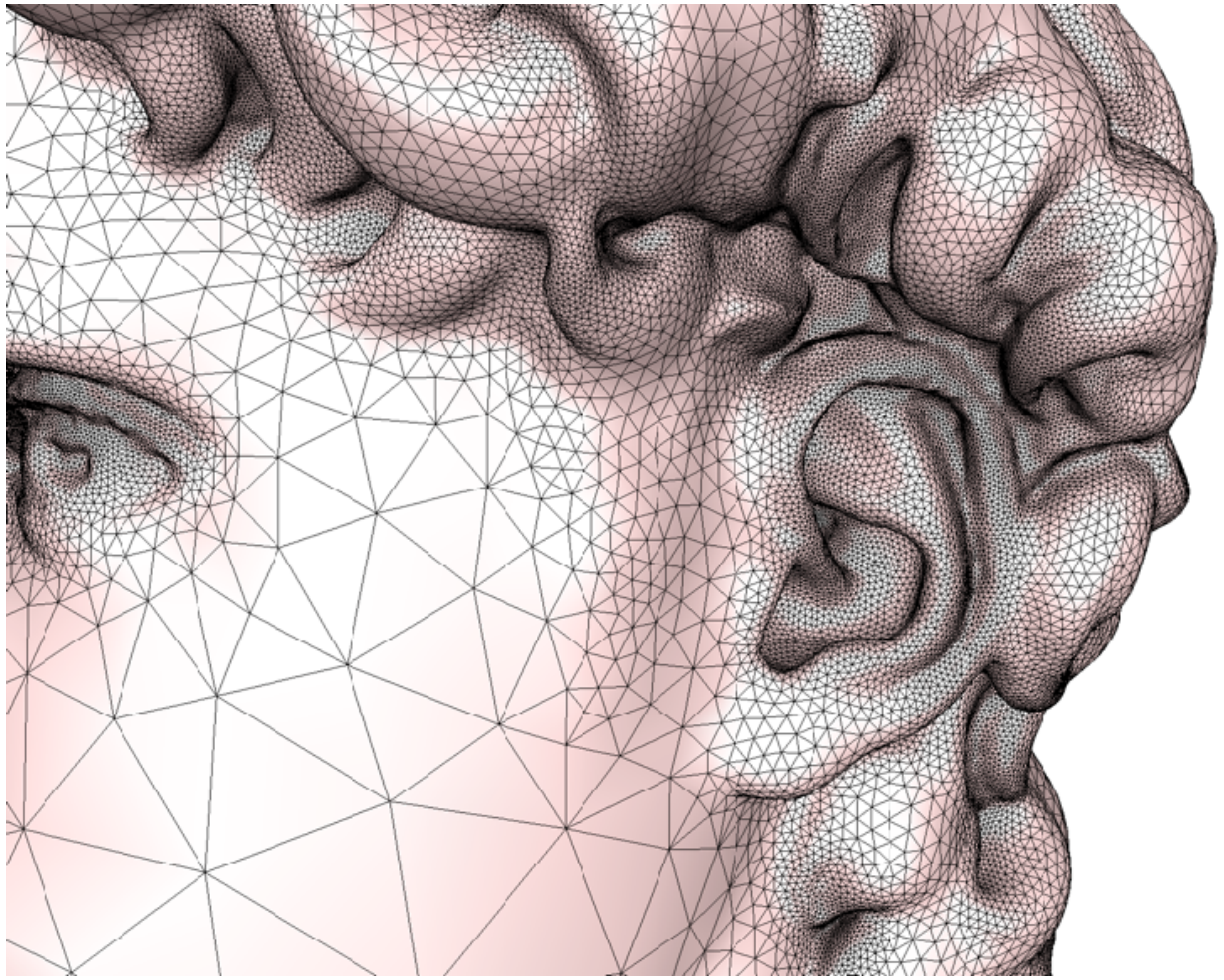


Uniform Sampling

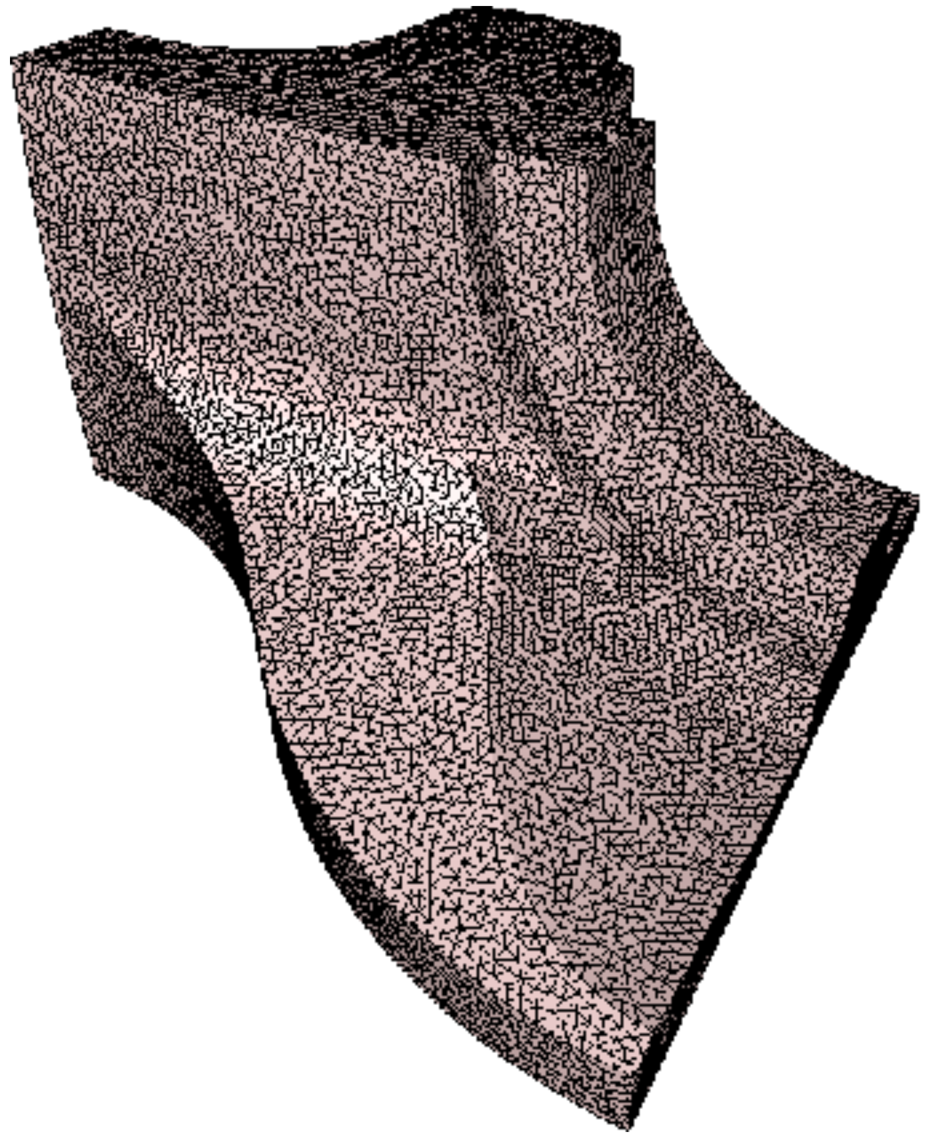


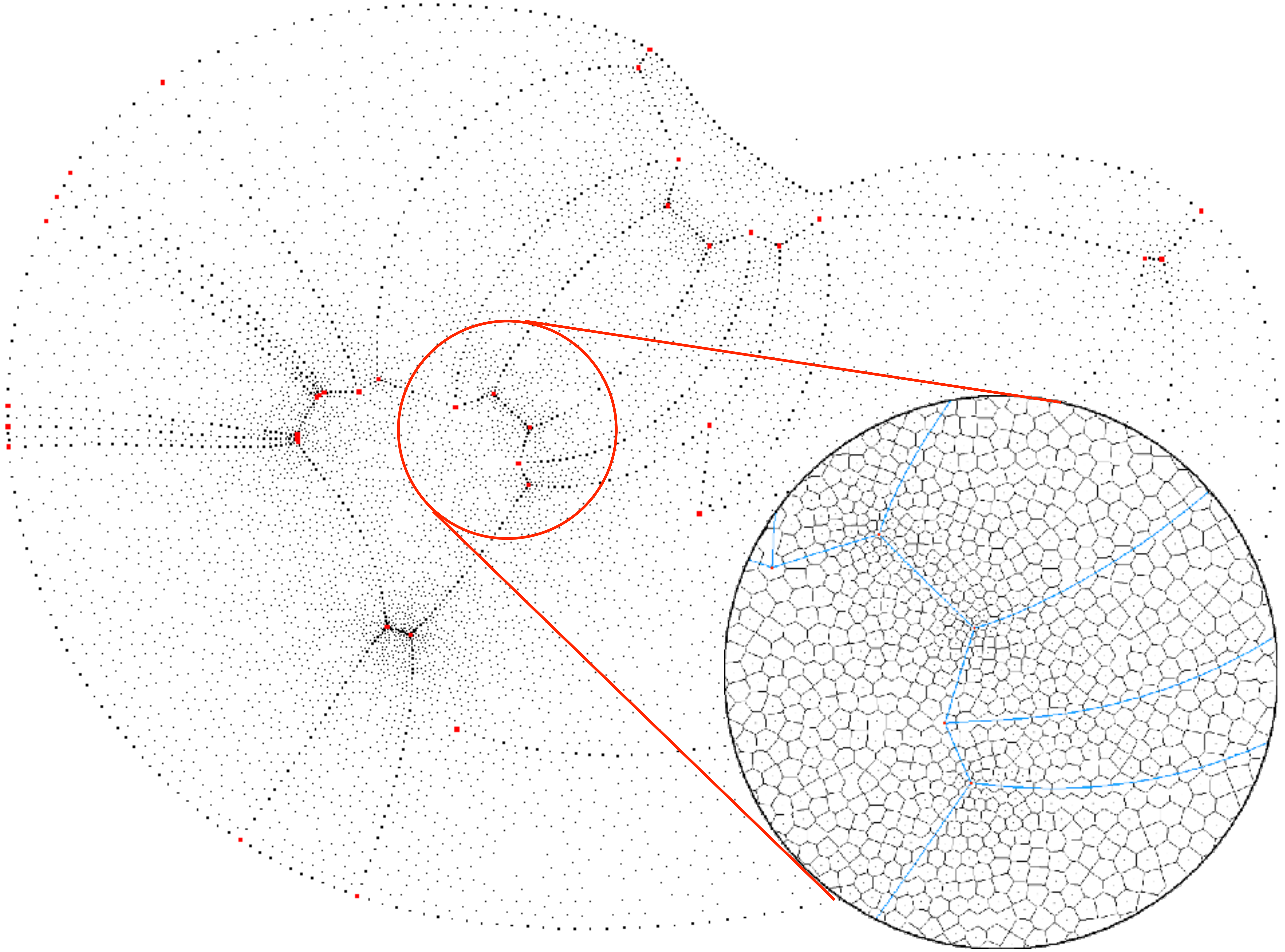
Adaptive Sampling





Sharp Features

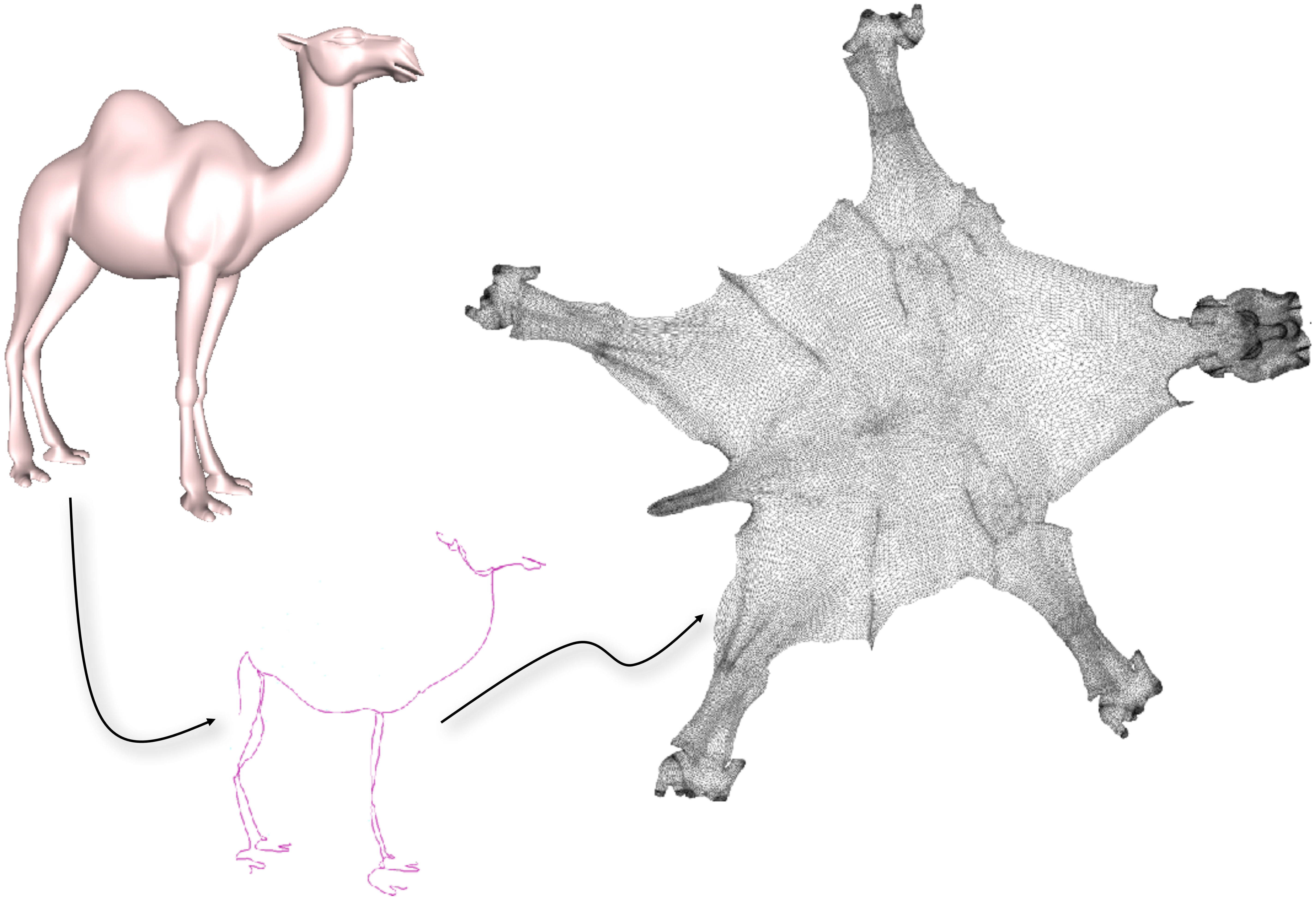




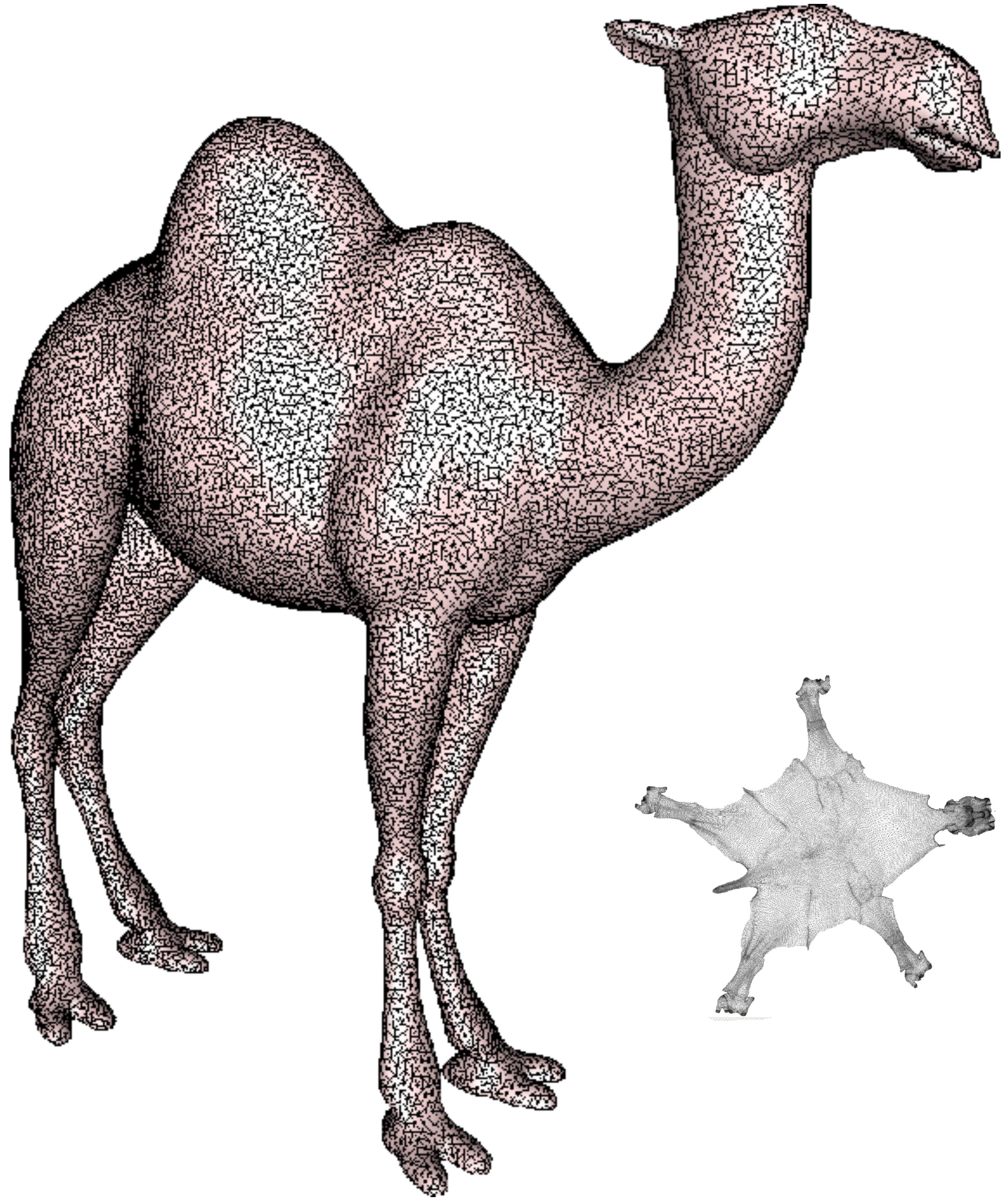
Limitations

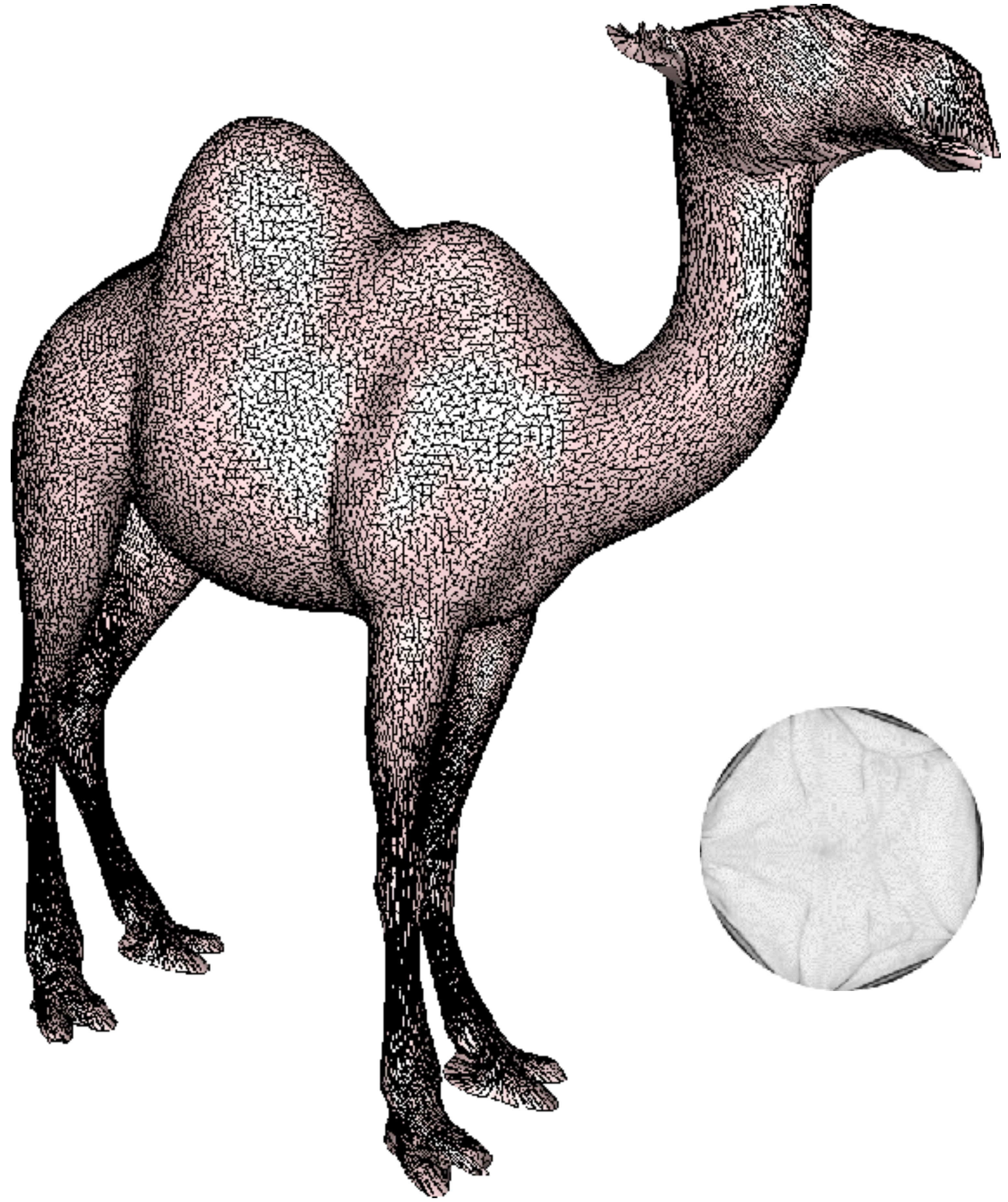
- Closed meshes
  - Need a good cut path
  - Free boundary parameterization
  - Stitch seams afterwards
- Protruding features
  - Legs, arms, ...
  - Numerical problems
  - Sampling fails

Smart Cut, Free Boundary

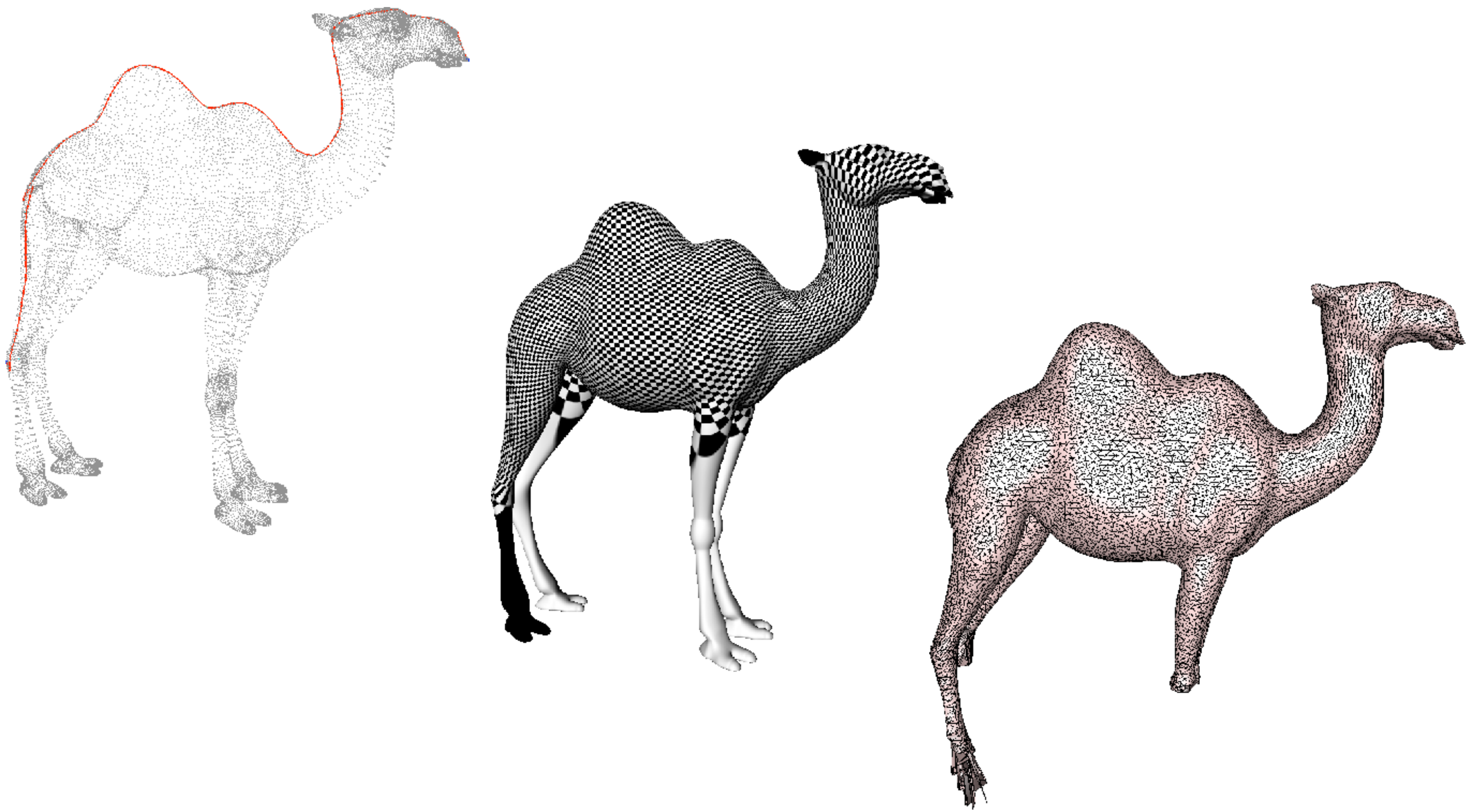


Free vs. Fixed Boundary





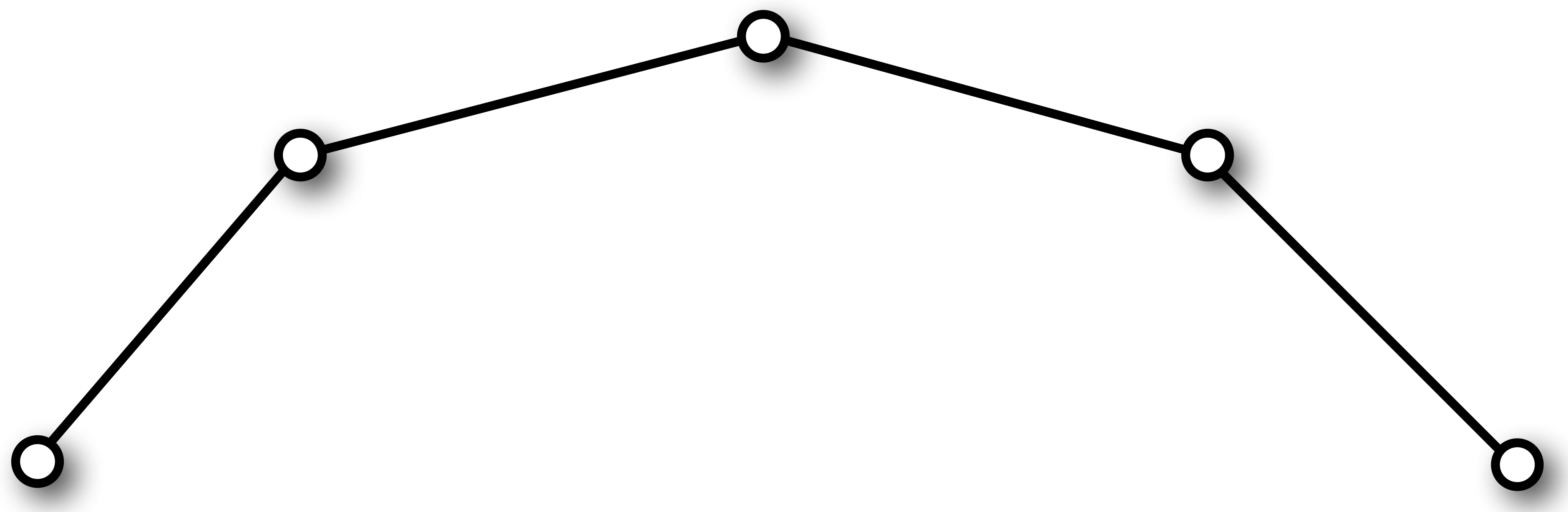
Naive Cut, Numerical Problems



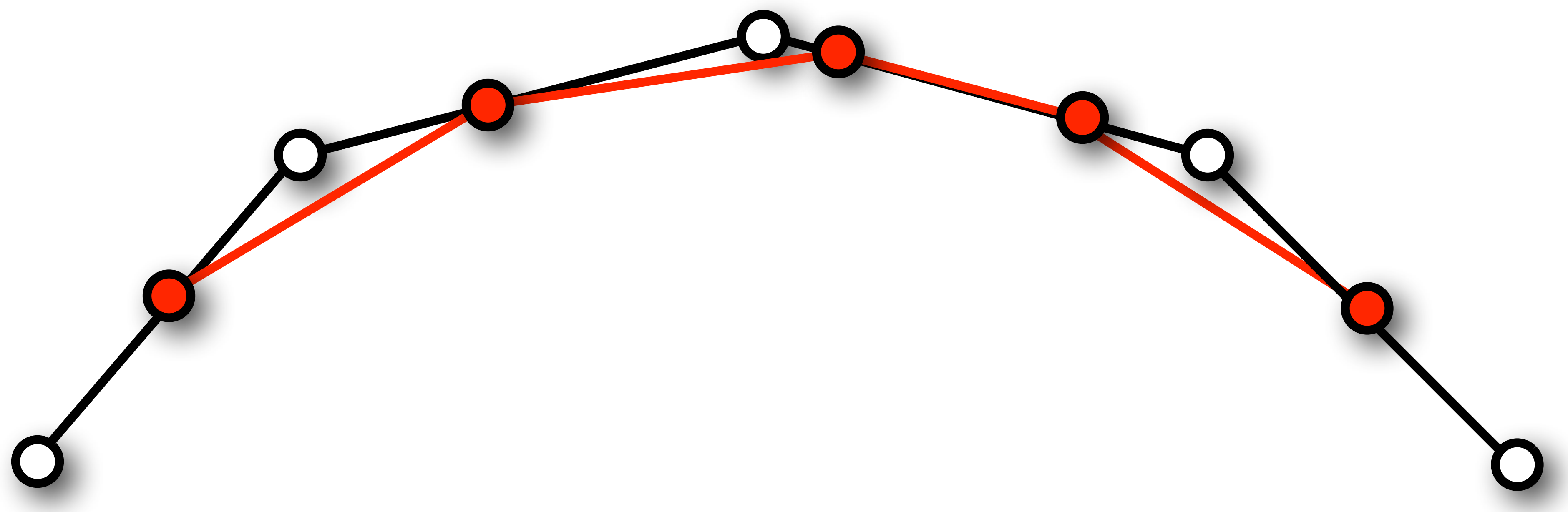
Overview

- 2D Meshing
- Parameterization-Based Remeshing
- **Surface-Based Remeshing**

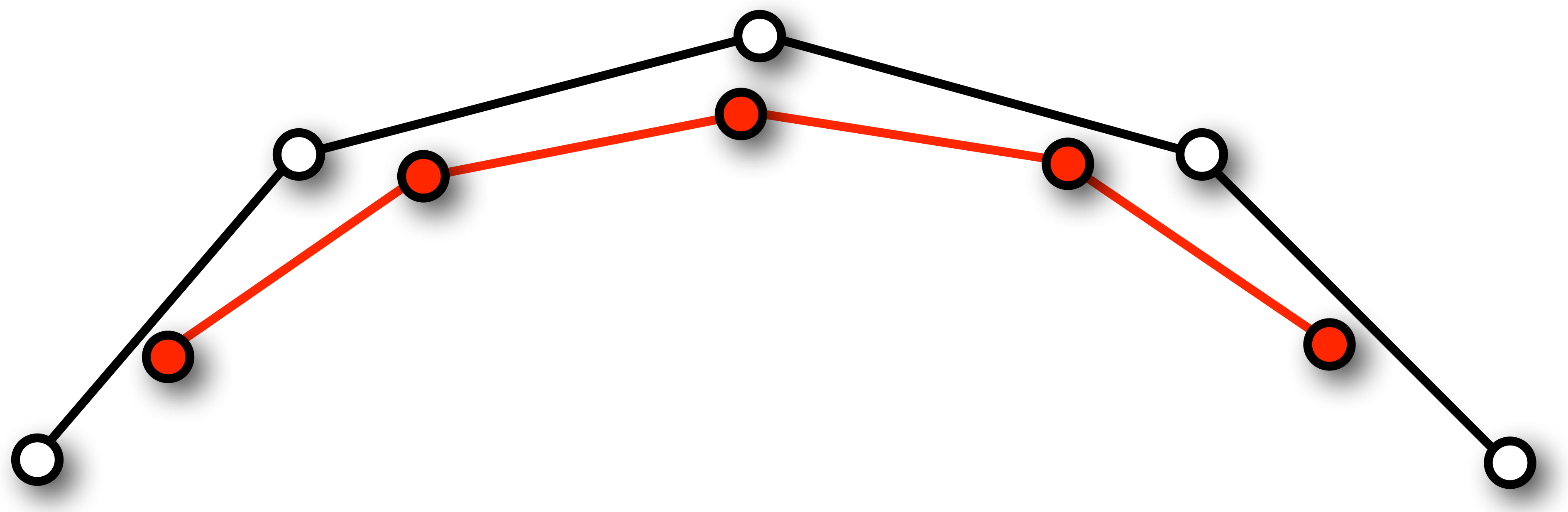
2. Surface Based



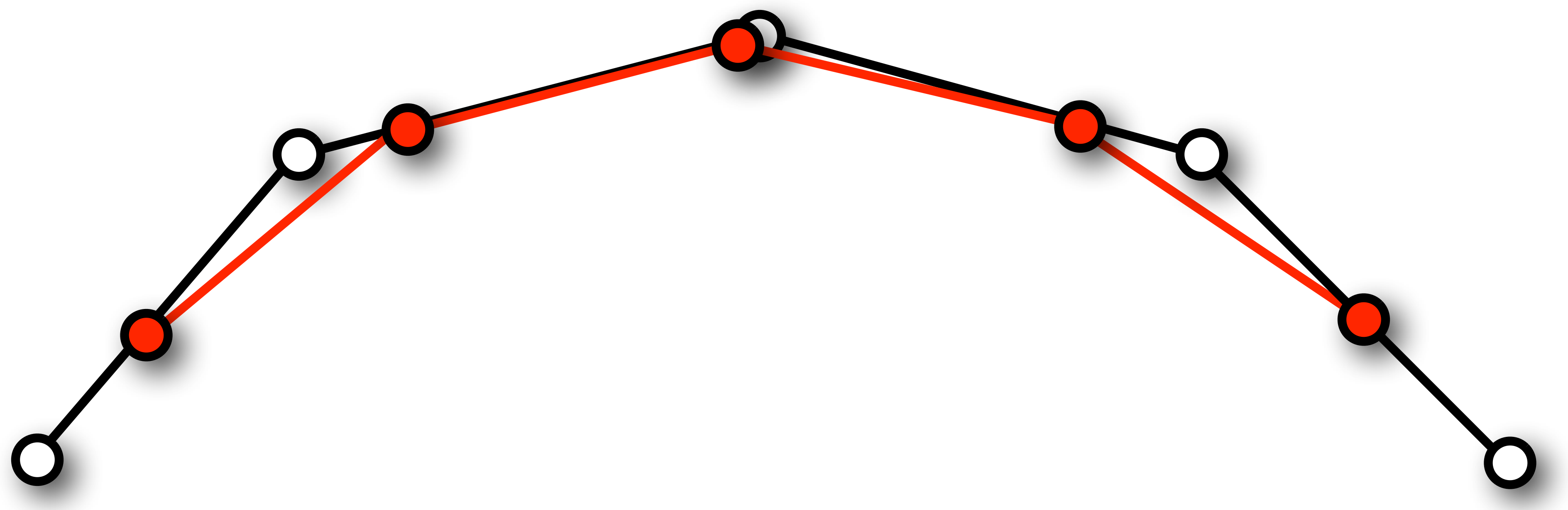
2. Surface Based



2. Surface Based



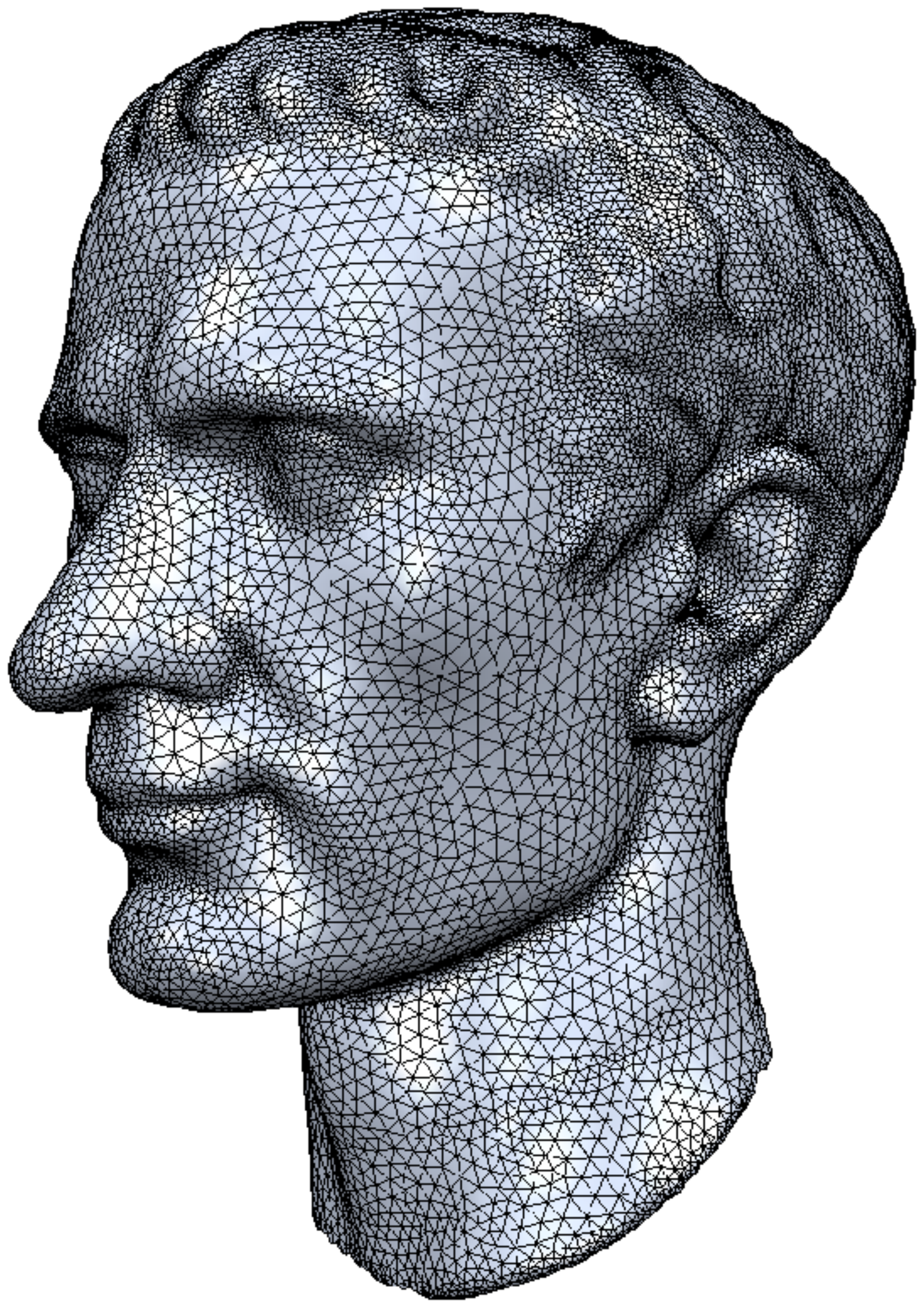
2. Surface Based



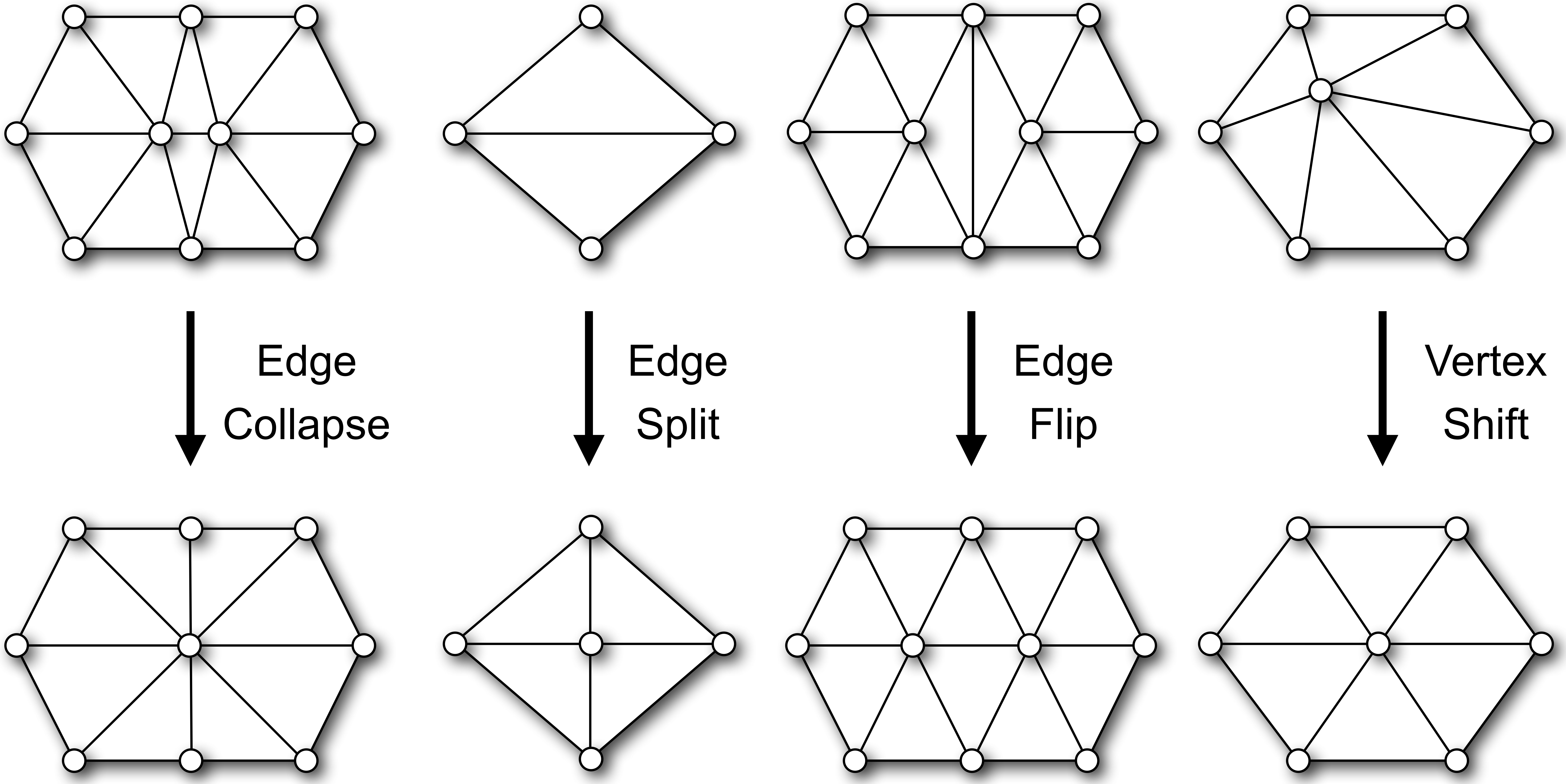
Direct Surface Remeshing

- ***Avoid*** global parametrization
  - Numerically very sensitive
  - Topological restrictions

- Surface-oriented remeshing
  - Perform ***local*** mesh updates
  - Use ***projections*** to keep vertices on the surface
  - Fast: e.g., Resampling of 100k triangles in less than 5s



Local Remeshing Operators



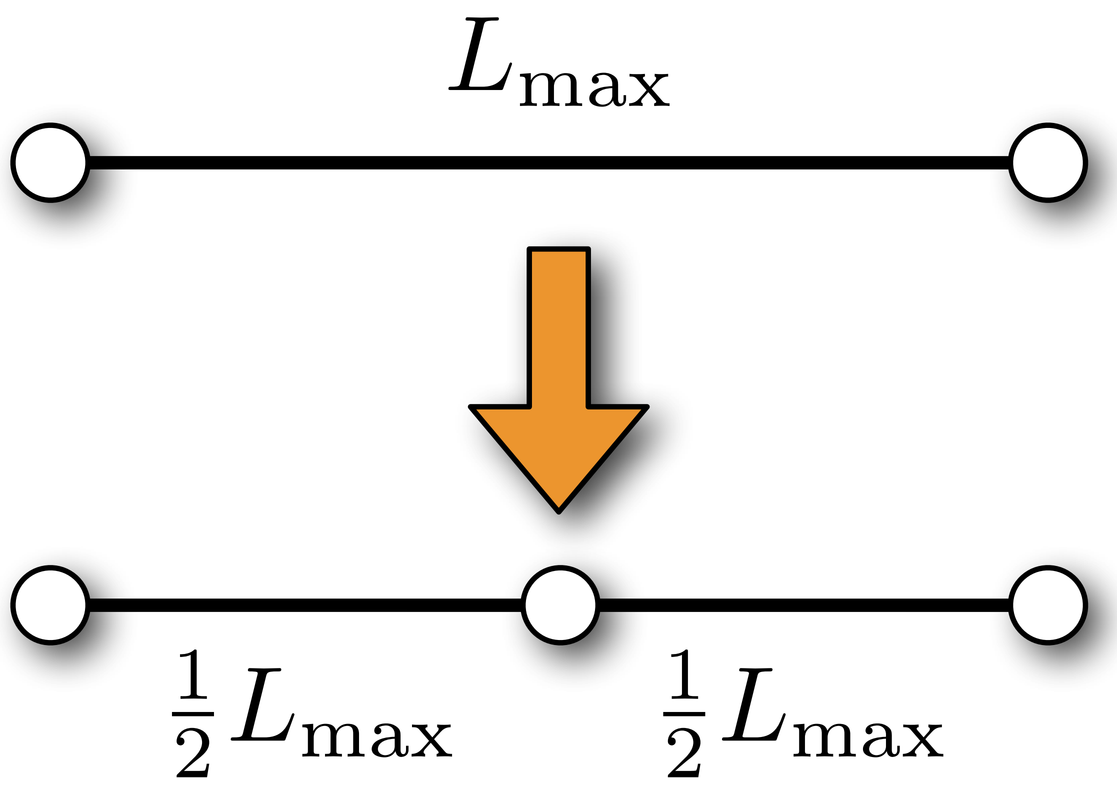
Isotropic Remeshing

Specify target edge length L

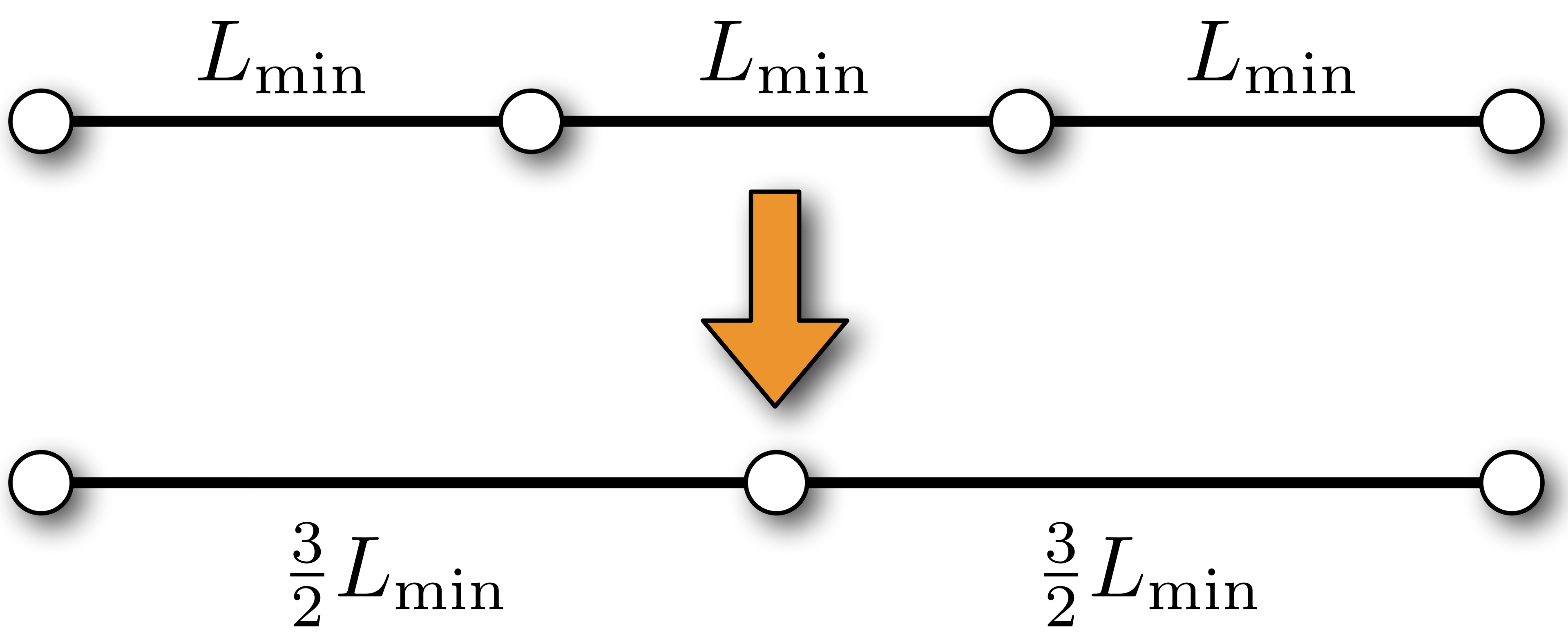
Iterate:

1. **Split** edges longer than L_{max}
2. **Collapse** edges shorter than L_{min}
3. **Flip** edges to get closer to valence 6
4. Vertex **shift** by tangential relaxation
5. **Project** vertices onto reference mesh

Edge Split/Collapse



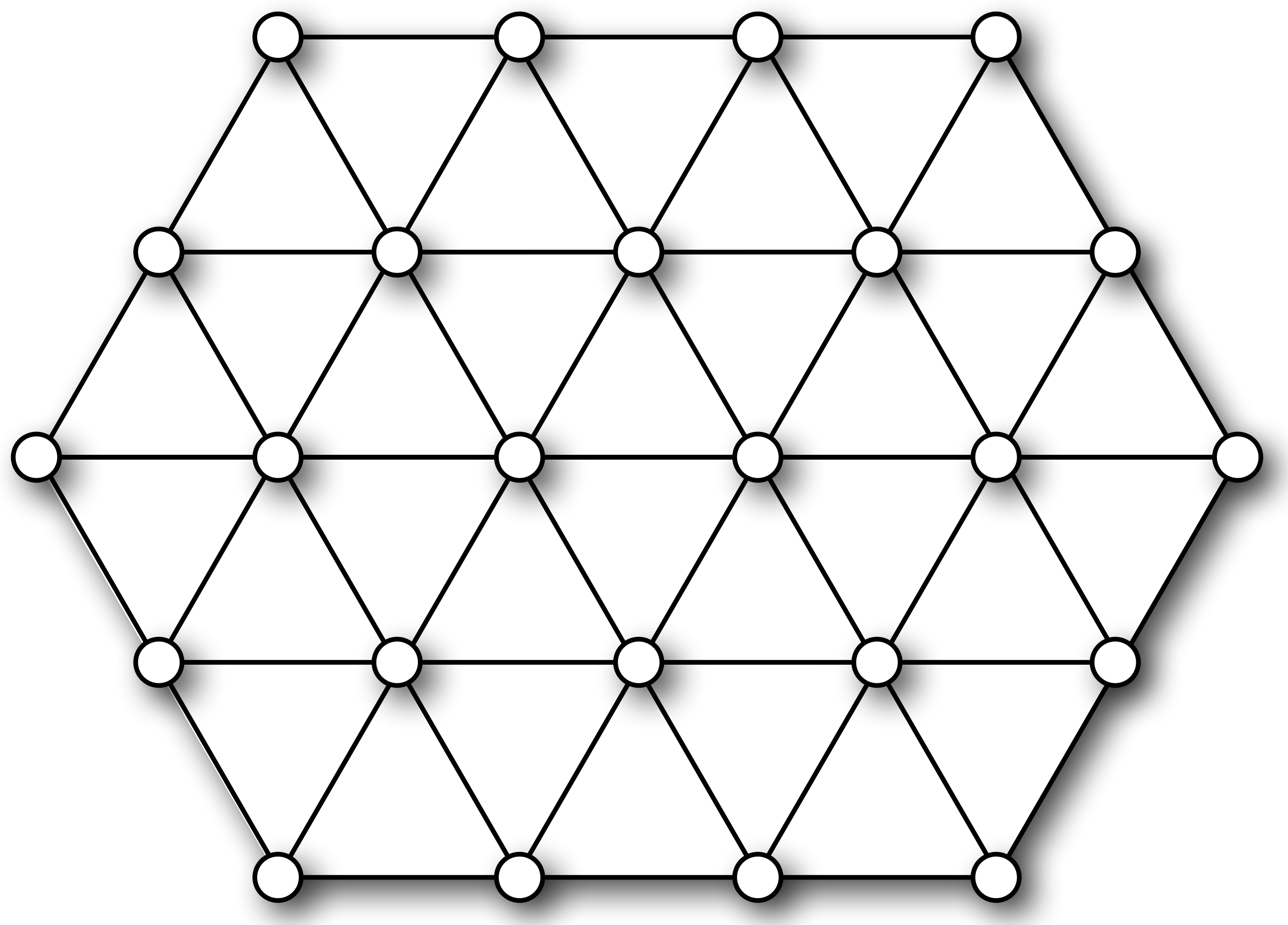
$$\begin{aligned} |L_{\max} - L| &= \left|\frac{1}{2}L_{\max} - L\right| \\ \Rightarrow L_{\max} &= \frac{4}{3}L \end{aligned}$$



$$\begin{aligned} |L_{\min} - L| &= \left|\frac{3}{2}L_{\min} - L\right| \\ \Rightarrow L_{\min} &= \frac{4}{5}L \end{aligned}$$

Edge Flip

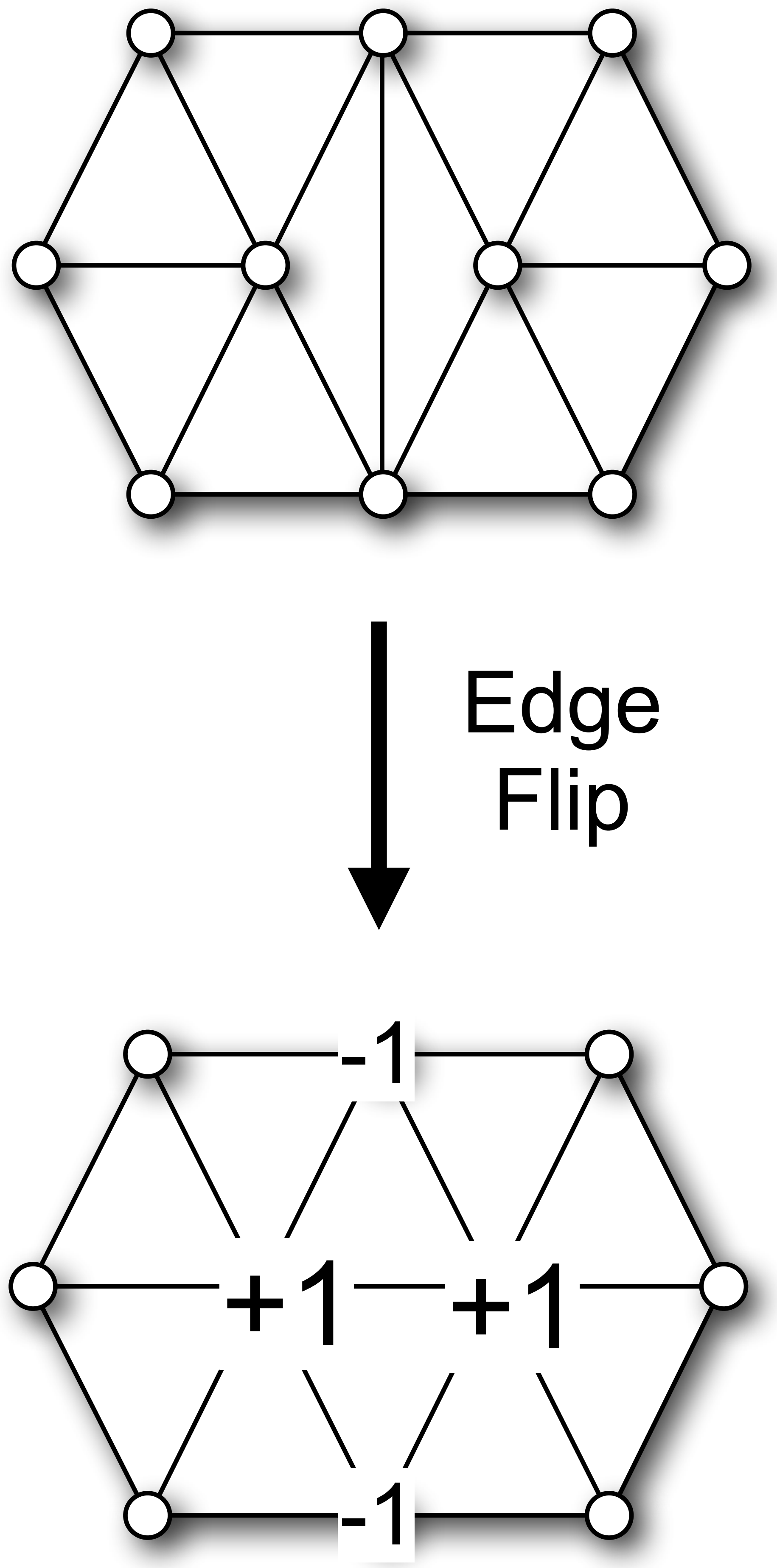
- Improve valences
  - Avg. valence is 6 (Euler)
  - Reduce variation
- Optimal valence is
  - 6 for interior vertices
  - 4 for boundary vertices



Edge Flip

- Improve valences
  - Avg. valence is 6 (Euler)
  - Reduce variation
- Optimal valence is
  - 6 for interior vertices
  - 4 for boundary vertices
- Minimize valence excess

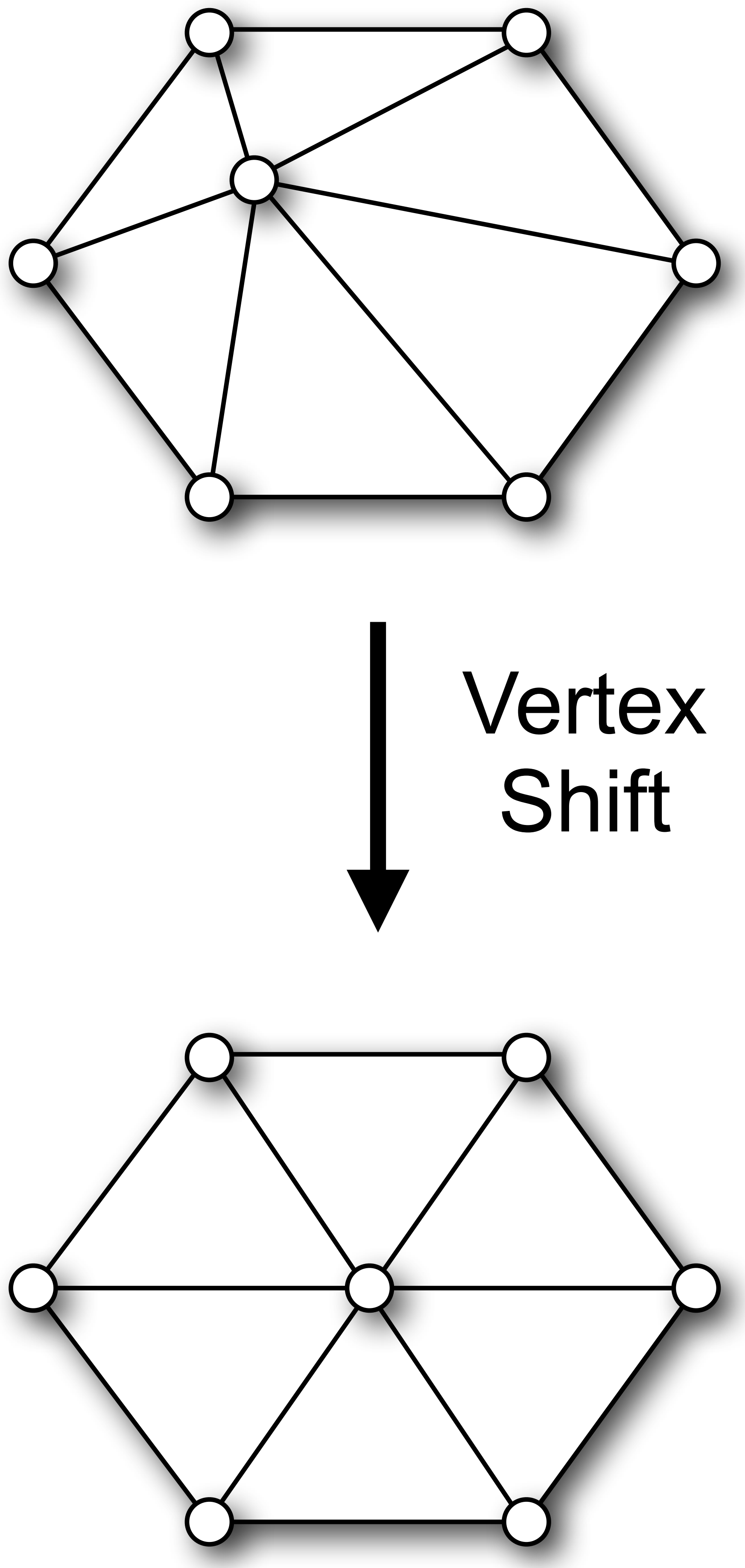
$$\sum_{i=1}^{4} (\text{valence}(v_i) - \text{opt\_valence}(v_i))^2$$



Vertex Shift

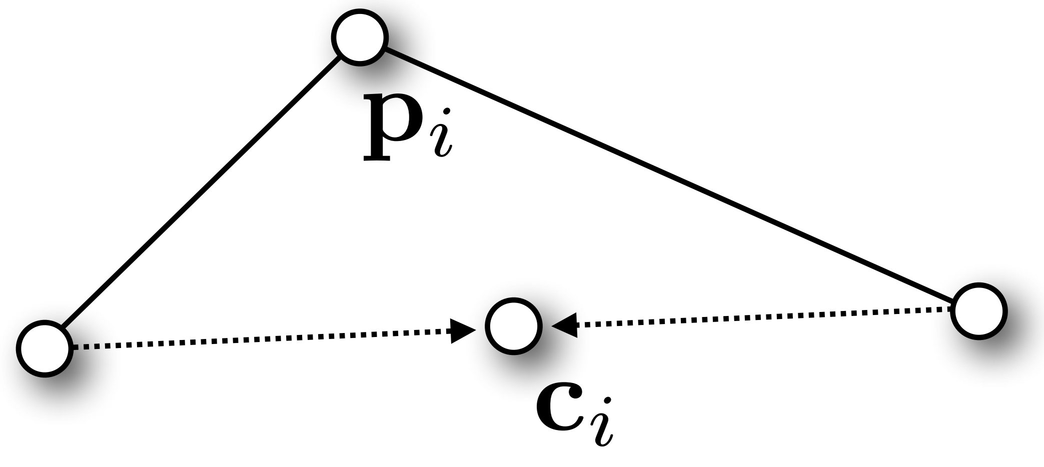
- Local “spring” relaxation
  - Uniform Laplacian smoothing
  - Barycenter of one-ring neighbors

$$\mathbf{c}_i = \frac{1}{\mathrm{valence}(v_i)} \sum_{j \in N(v_i)} \mathbf{p}_j$$



Vertex Shift

- Local “spring” relaxation
  - Uniform Laplacian smoothing
  - Barycenter of one-ring neighbors



$$\mathbf{c}_i = \frac{1}{\text{valence}(v_i)} \sum_{j \in N(v_i)} \mathbf{p}_j$$

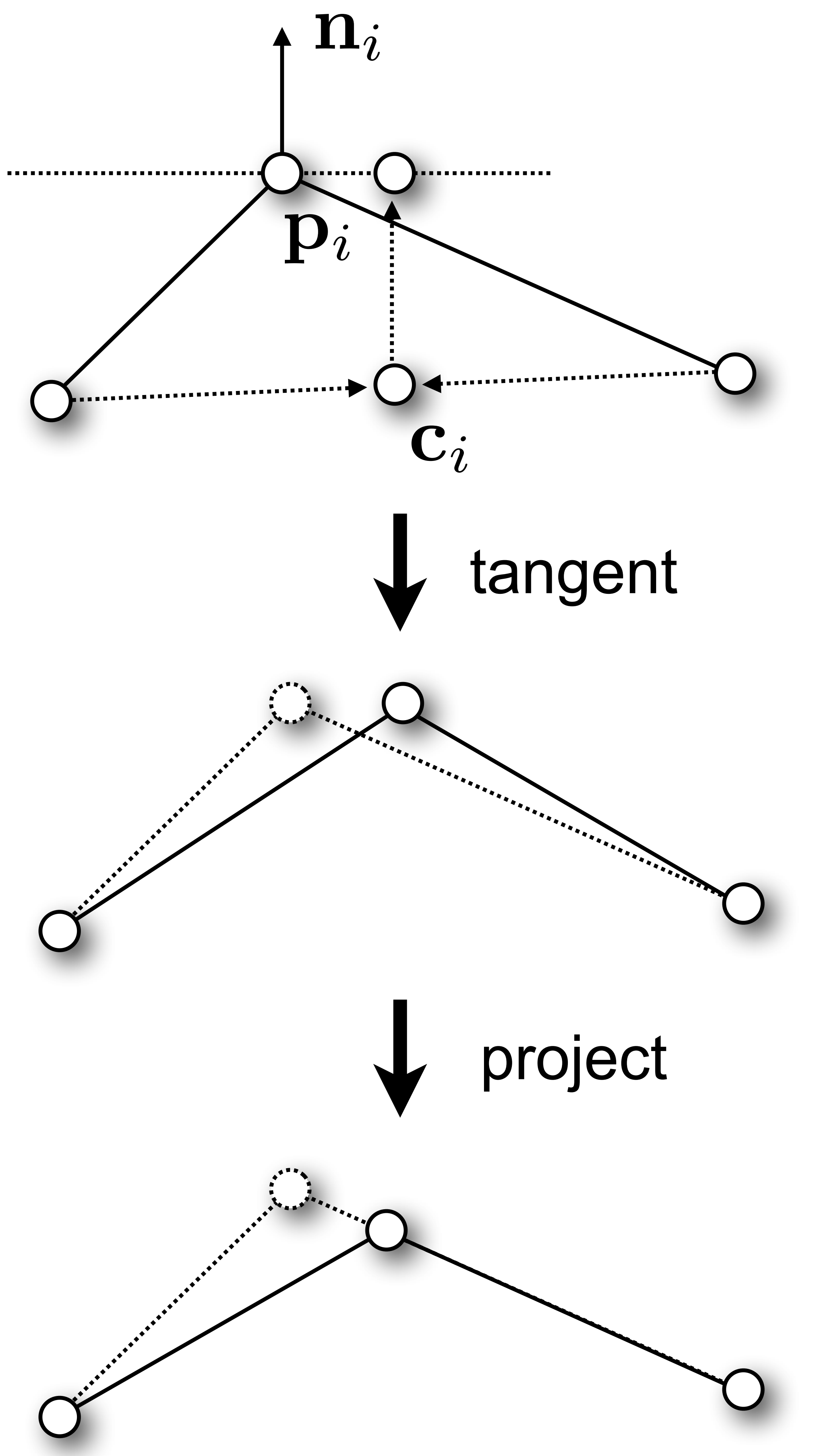
Vertex Shift

- Local "spring" relaxation
  - Uniform Laplacian smoothing
  - Barycenter of one-ring neighbors

$$\mathbf{c}_i = \frac{1}{\text{valence}(v_i)} \sum_{j \in N(v_i)} \mathbf{p}_j$$

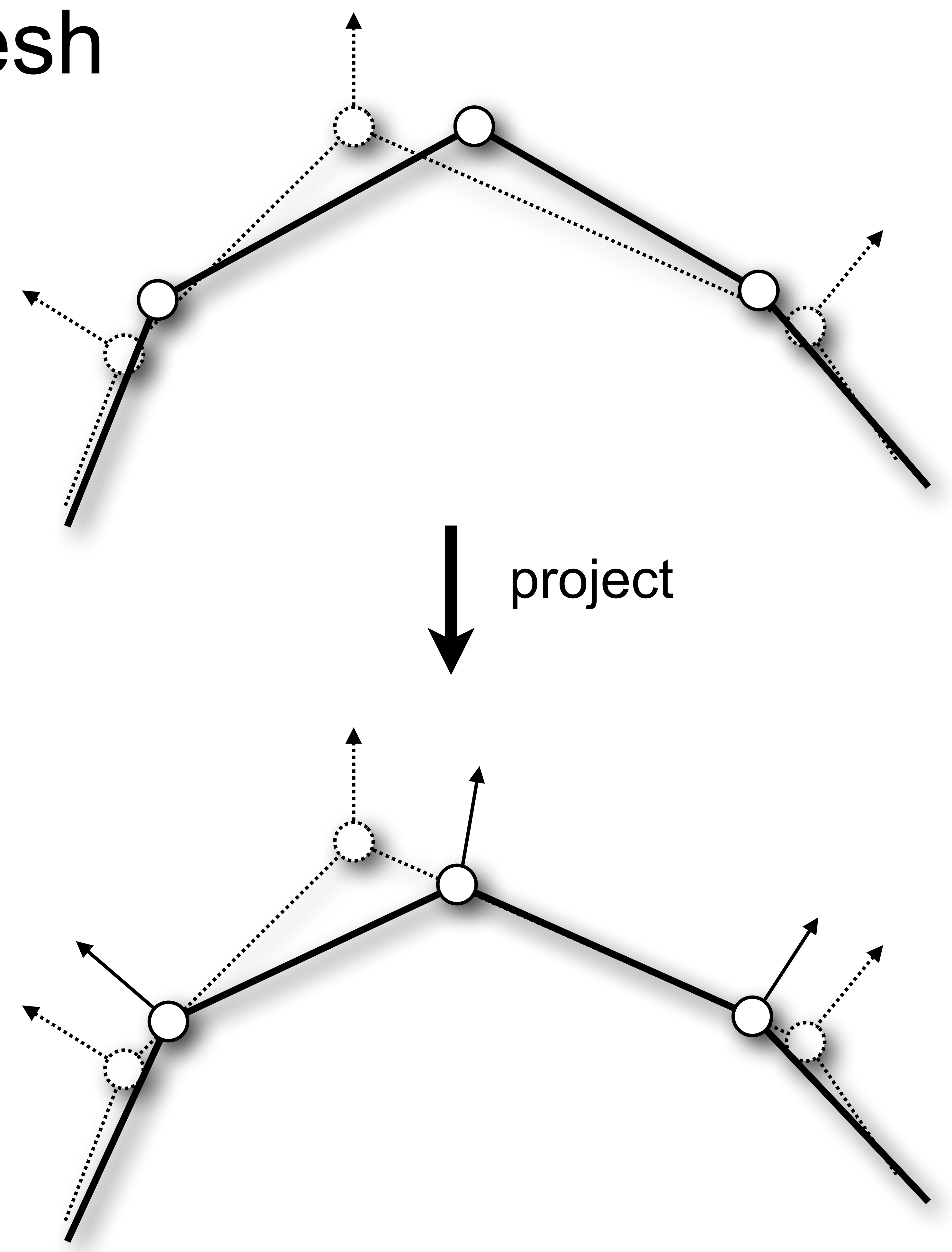
- Keep vertex (approx.) on surface
  - Restrict movement to tangent plane

$$\mathbf{p}_i \leftarrow \mathbf{p}_i + \lambda \left(\mathbf{I} - \mathbf{n}_i \mathbf{n}_i^T \right) (\mathbf{c}_i - \mathbf{p}_i)$$



Vertex Projection

- Project vertices onto original reference mesh
  - Find closest triangle
  - Use BSP → $O(\log n)$
- Compute position & normal by barycentric interpolation



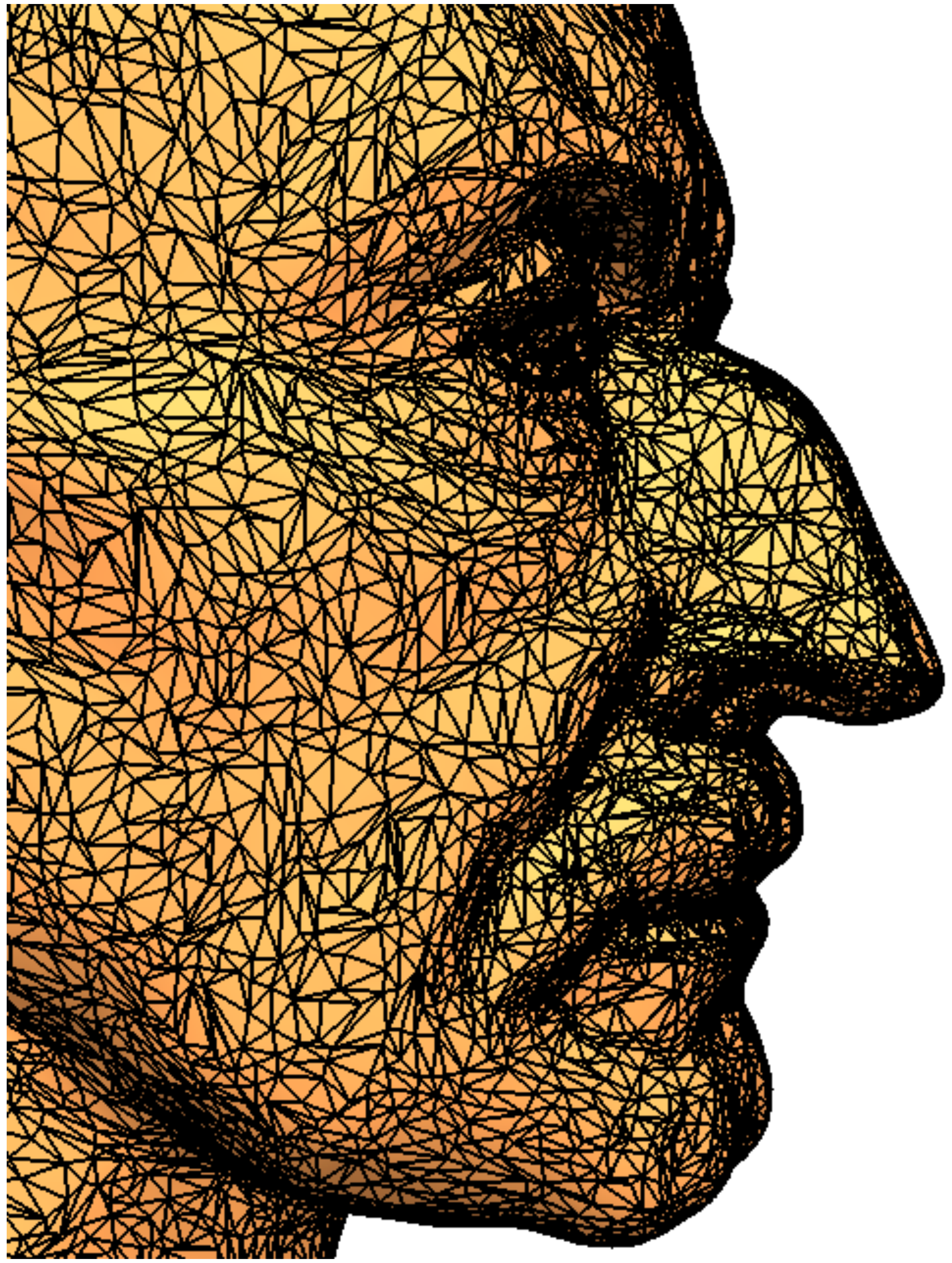
Isotropic Remeshing

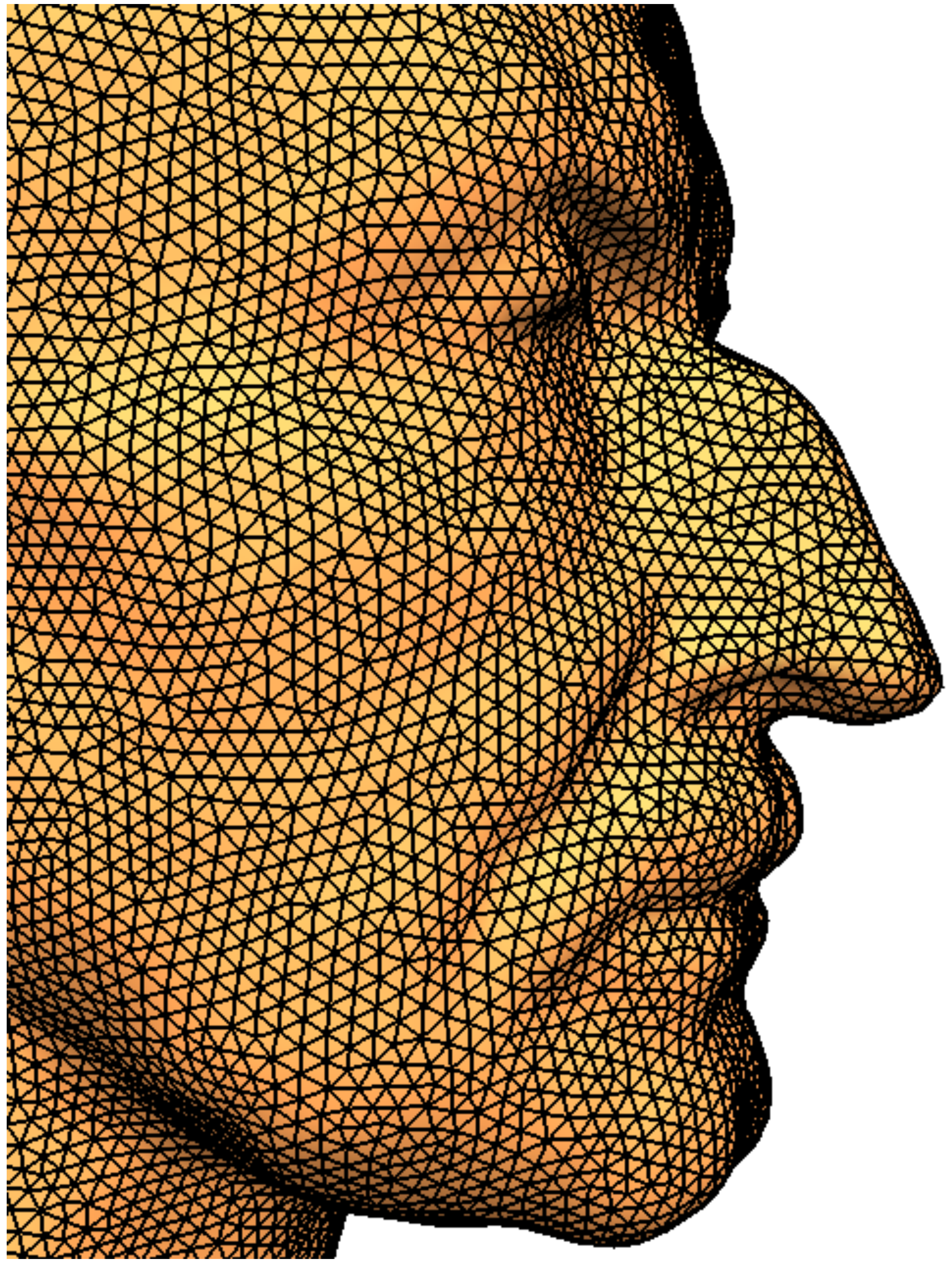
Specify target edge length L

Iterate:

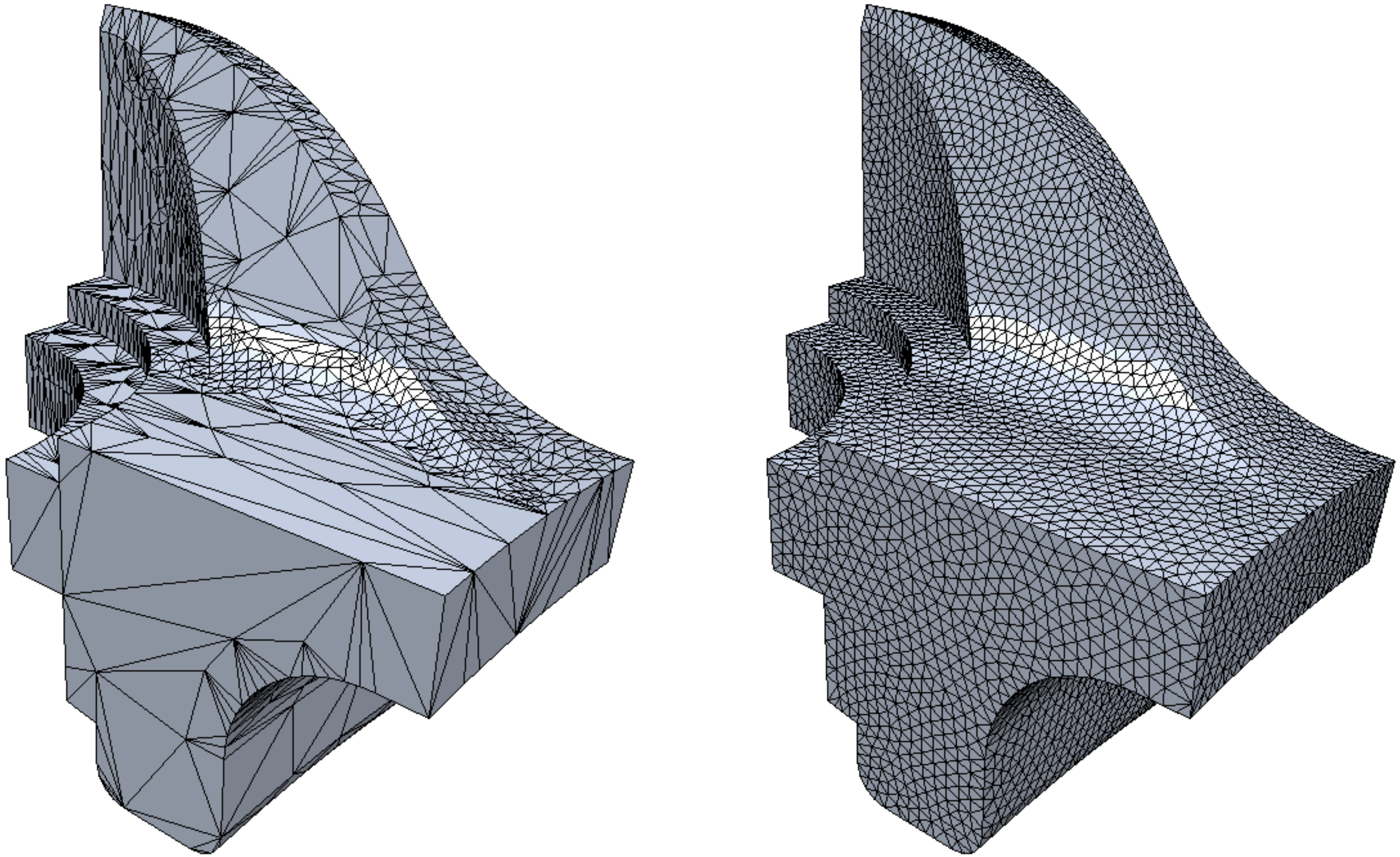
1. **Split** edges longer than L_{max}
2. **Collapse** edges shorter than L_{min}
3. **Flip** edges to get closer to valence 6
4. Vertex **shift** by tangential relaxation
5. **Project** vertices onto reference mesh

Remeshing Results





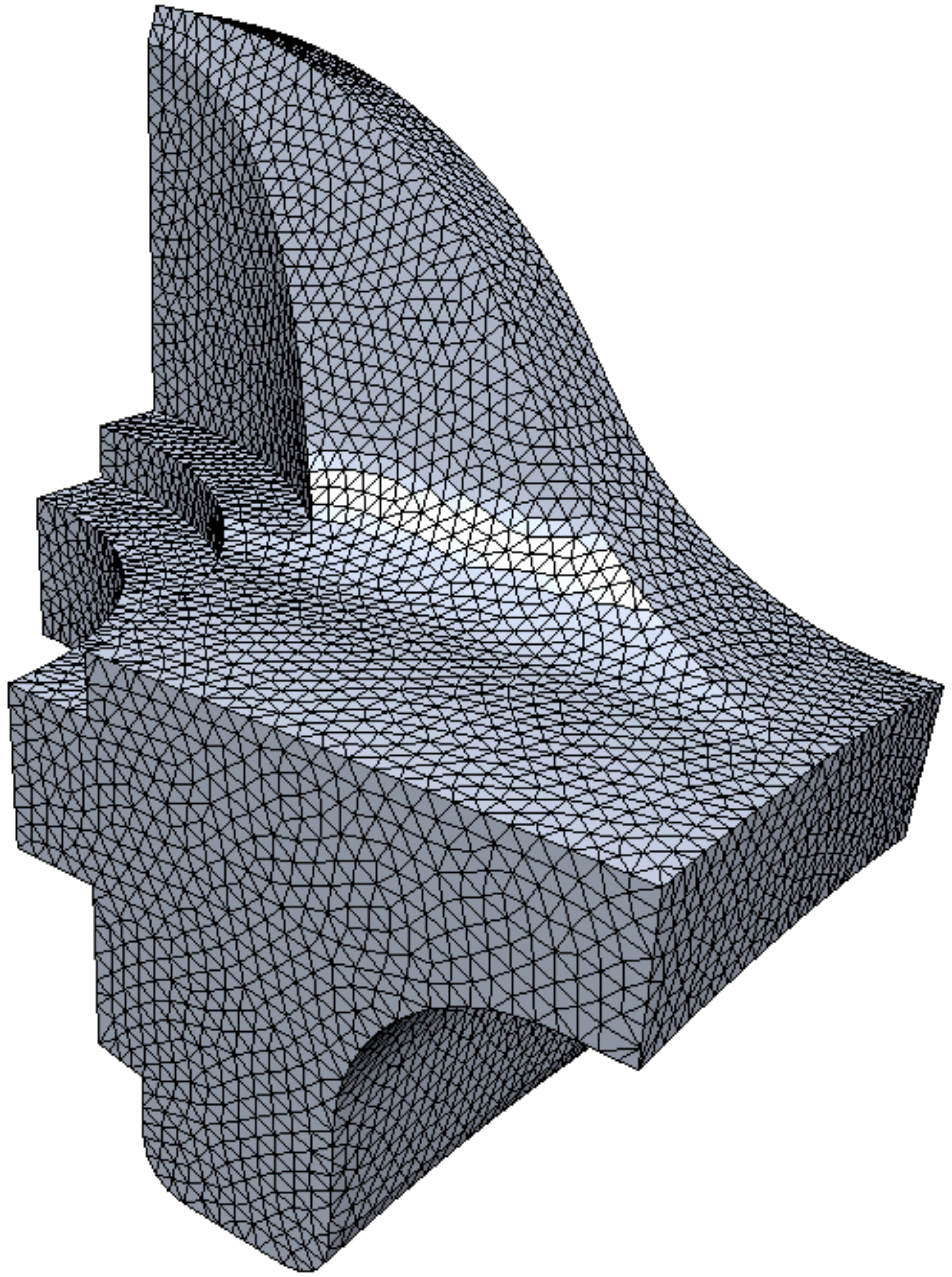
Feature Preservation?



Feature Preservation

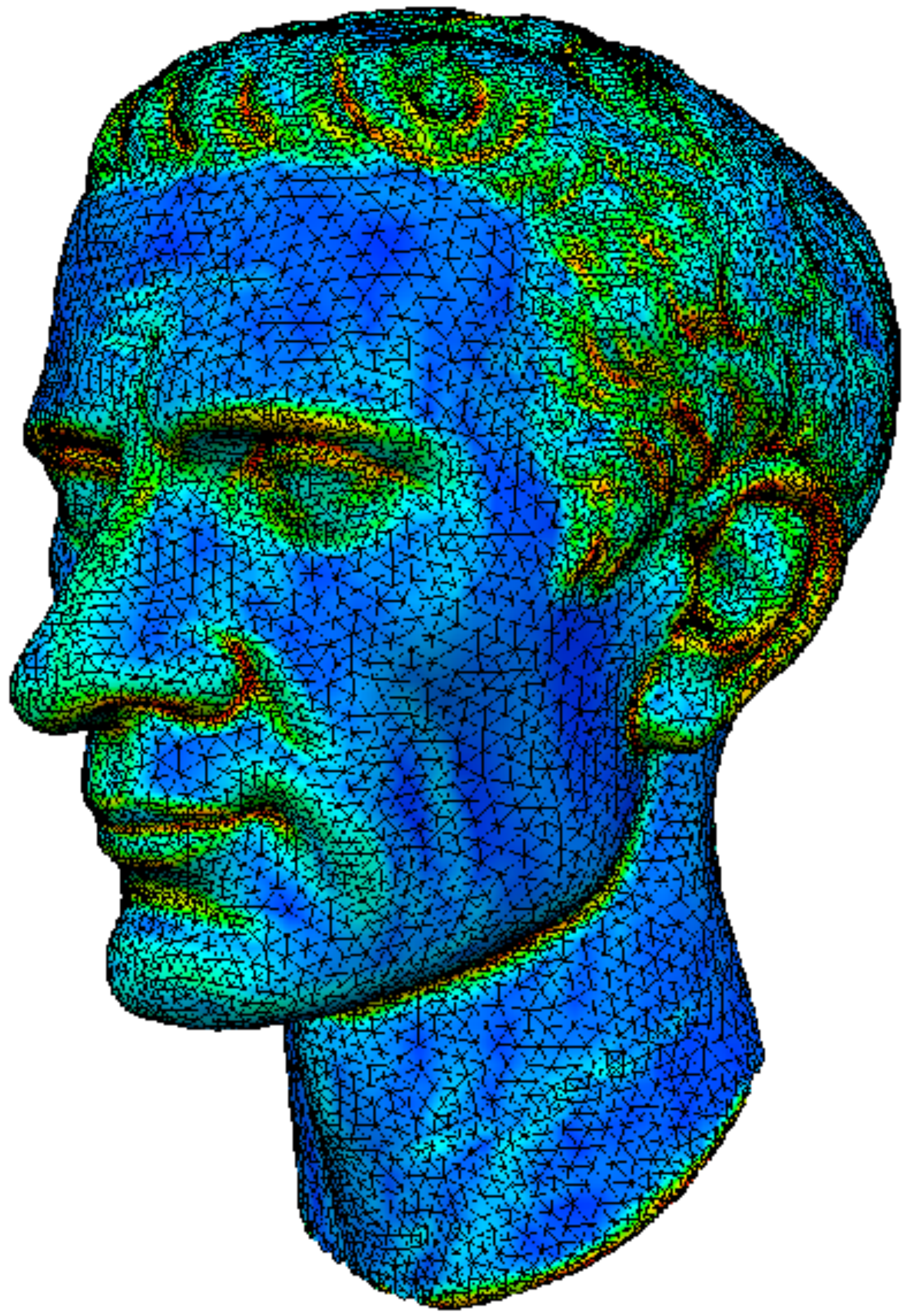
- Define feature edges / vertices
  - Large dihedral angles
  - Material boundaries

- Adjust local operators
  - Don’t flip feature edges
  - Collapse only along features
  - Univariate smoothing
  - Project to feature curves
  - Don’t touch feature vertices



Adaptive Remeshing

- Compute maximum principle curvature on reference mesh
- Determine **local** target edge length from max-curvature
- Adjust split & collapse criteria accordingly



Remeshing Approaches

- Parametrization based
  - Map complex 3D problem to simpler 2D problem
  - Parameterization is the difficult part
  - Computationally more involved

- Surface oriented
  - Local mesh operators on 3D surface
  - Project vertices onto reference surface
  - Projection works best on dense meshes

Remeshing Components

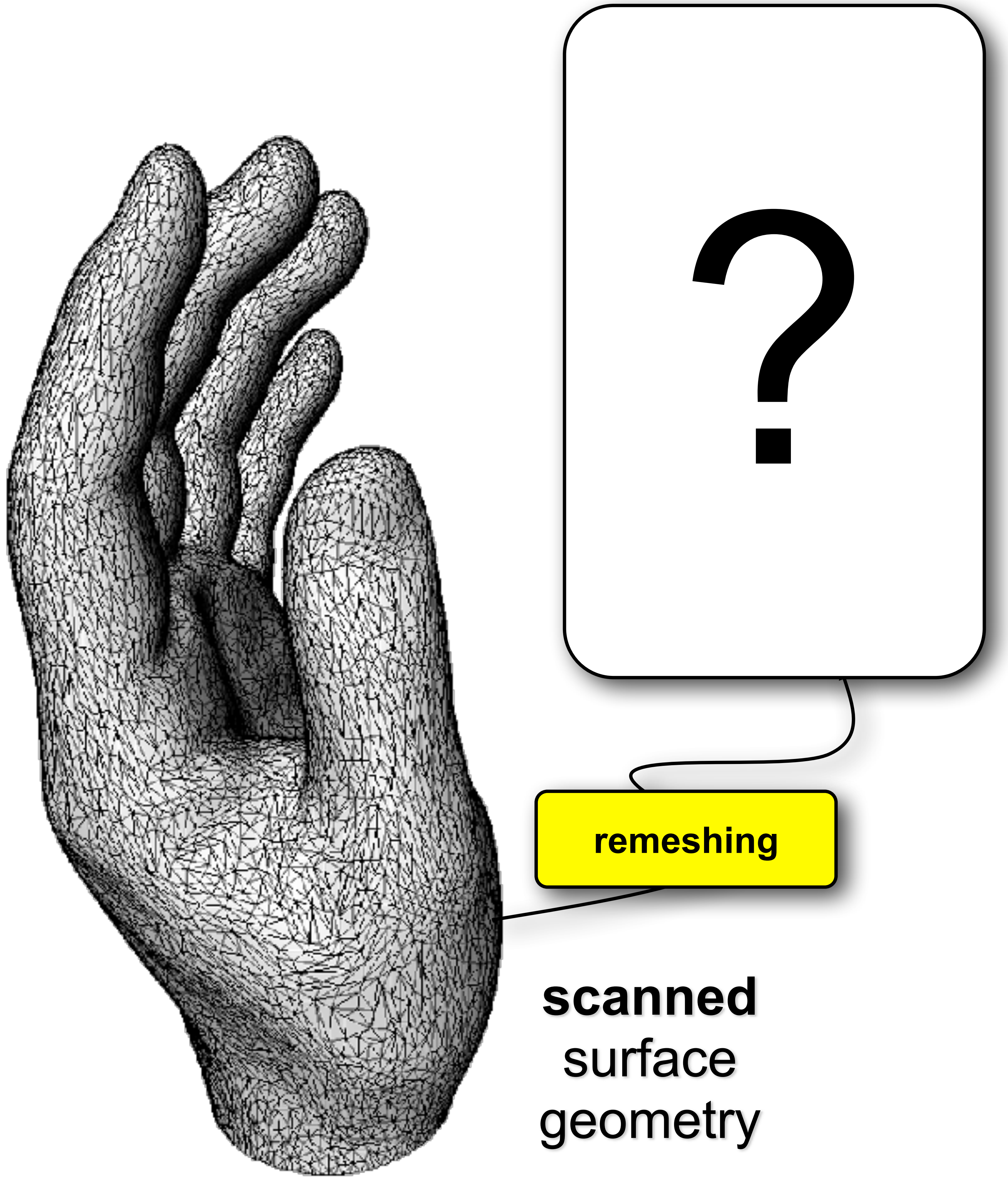
- **Correspondence**
  - Global / local parameterization vs. projections
- **Vertex density** control
  - Uniform vs. adaptive
- **Local alignment**
  - Optimal shape approximation (→ quad meshing)
- **Global alignment**
  - Preservation of sharp features

Overview

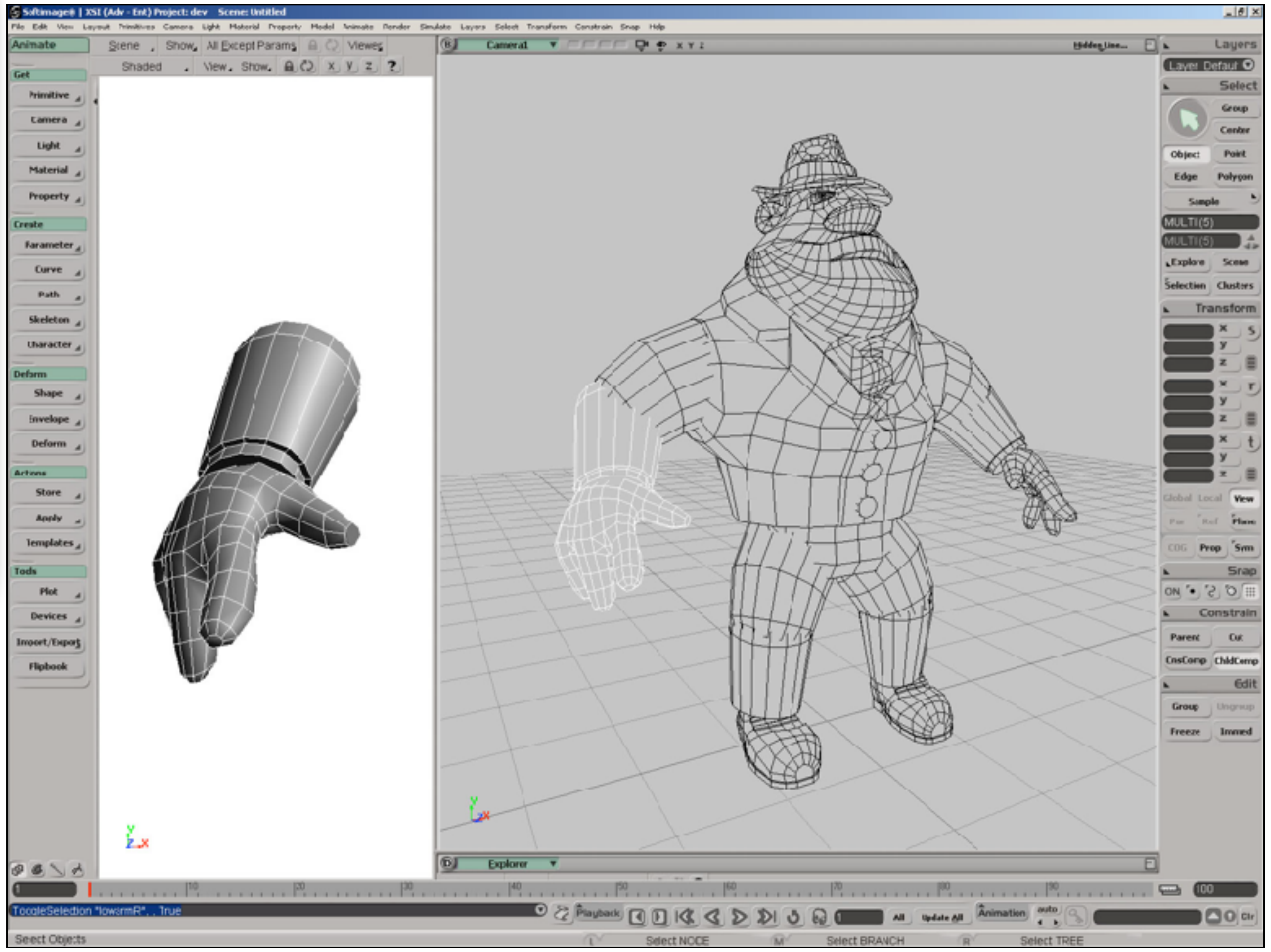
- Isotropic Remeshing
  - Parameterization-based [Alliez 2002]
  - Surface-based [Botsch 2004]

- **Anisotropic Remeshing**
  - Parameterization-based [Alliez 2003]
  - Surface-based [Marinov 2004]

Anisotropic Remeshing



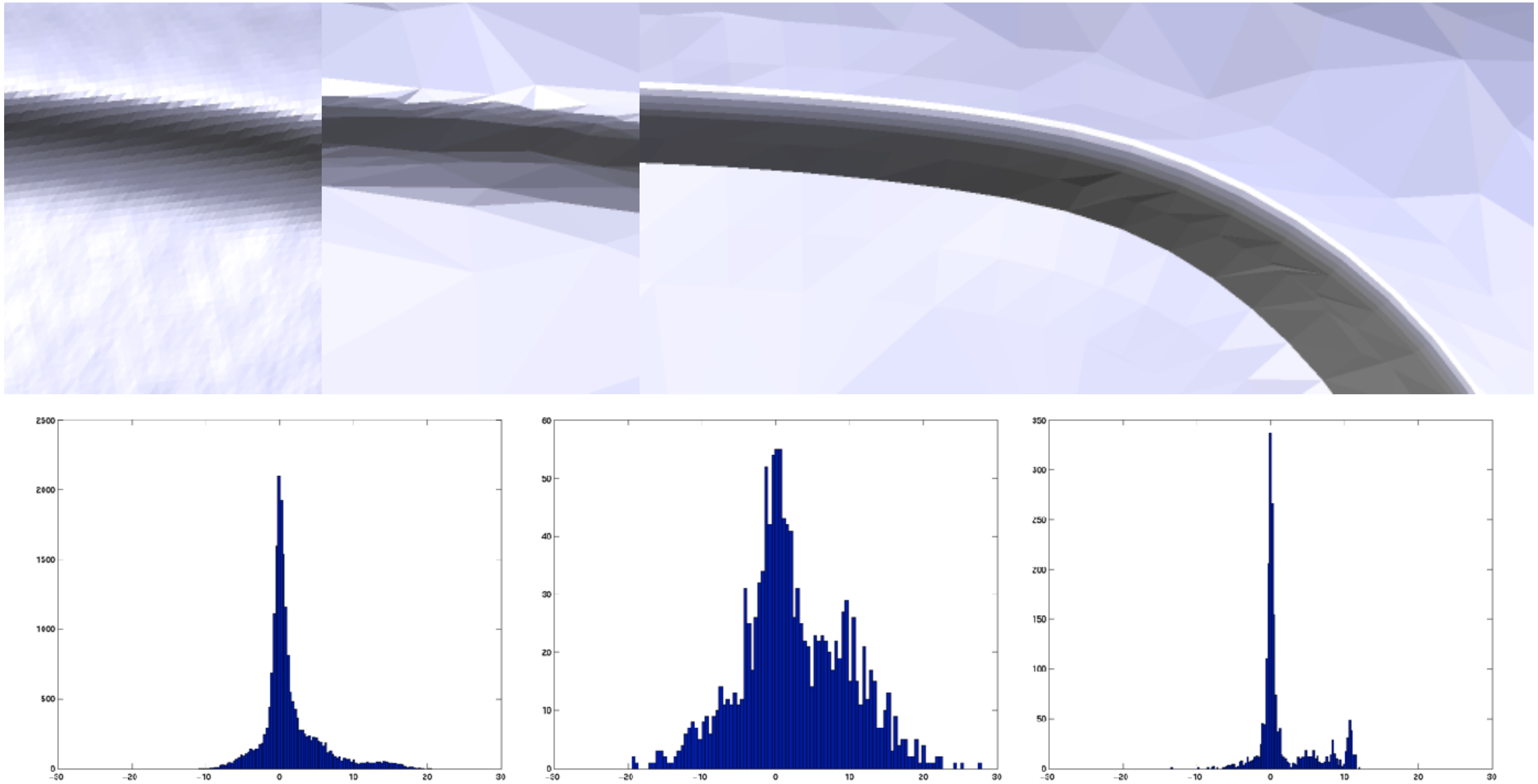
scanned surface geometry



carefully designed model

www.viewpoint.com

Normal Noise

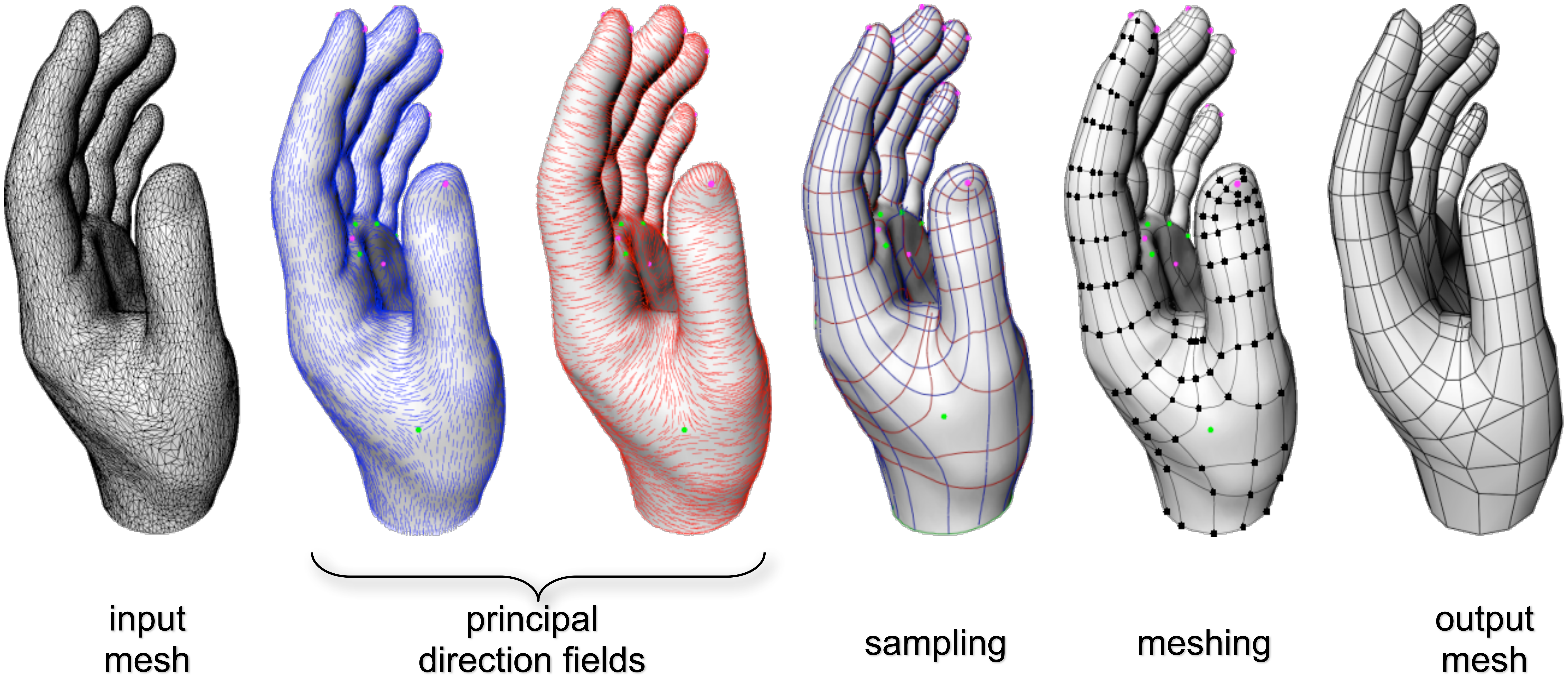


Anisotropic Remeshing

Overview

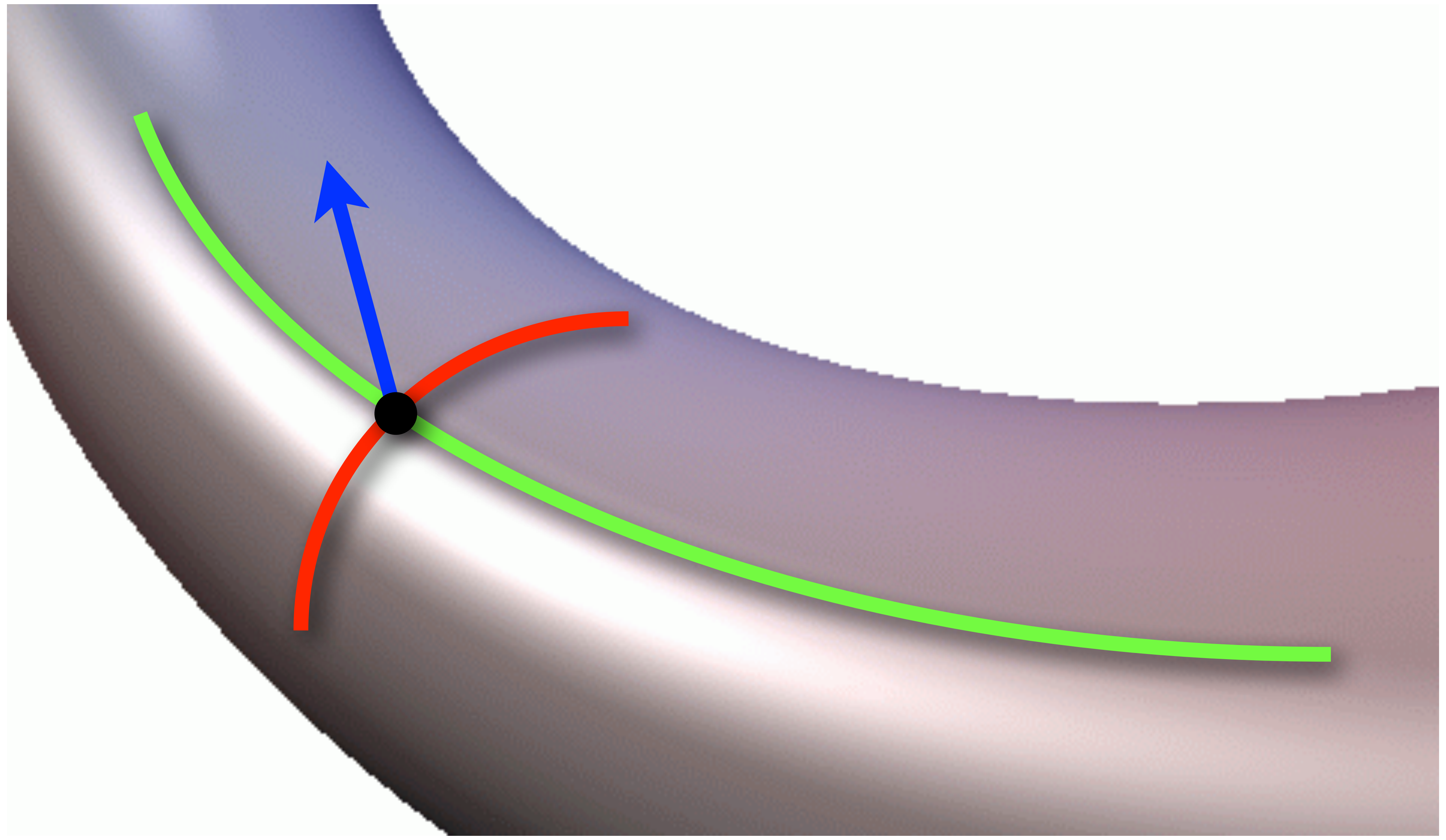
- Isotropic Remeshing
  - Parameterization-based [Alliez 2002]
  - Surface-based [Botsch 2004]
- Anisotropic Remeshing
  - **Parameterization-based** **[Alliez 2003]**
  - Surface-based [Marinov 2004]

Anistropic Remeshing



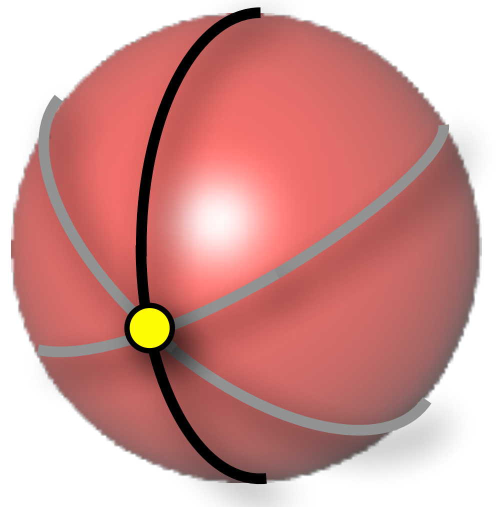
Anisotropy

- Differential geometry
  - 2nd fundamental form defines a local **orthogonal** frame (min- / max-curvature directions and the normal)

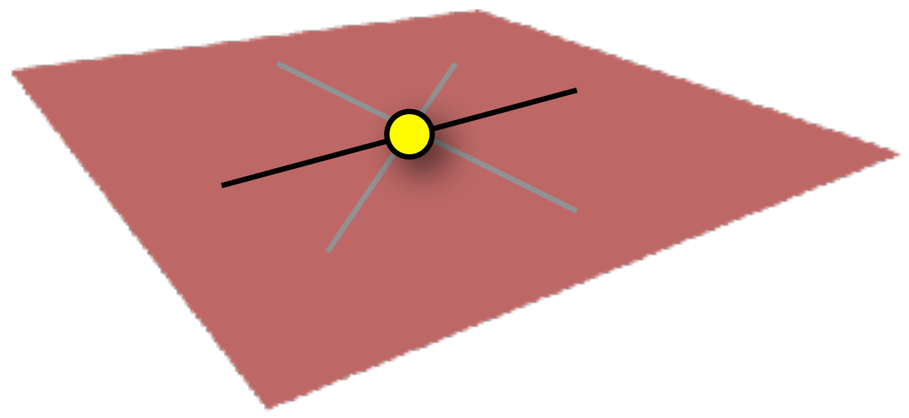


3D Curvature Tensor

Isotropic



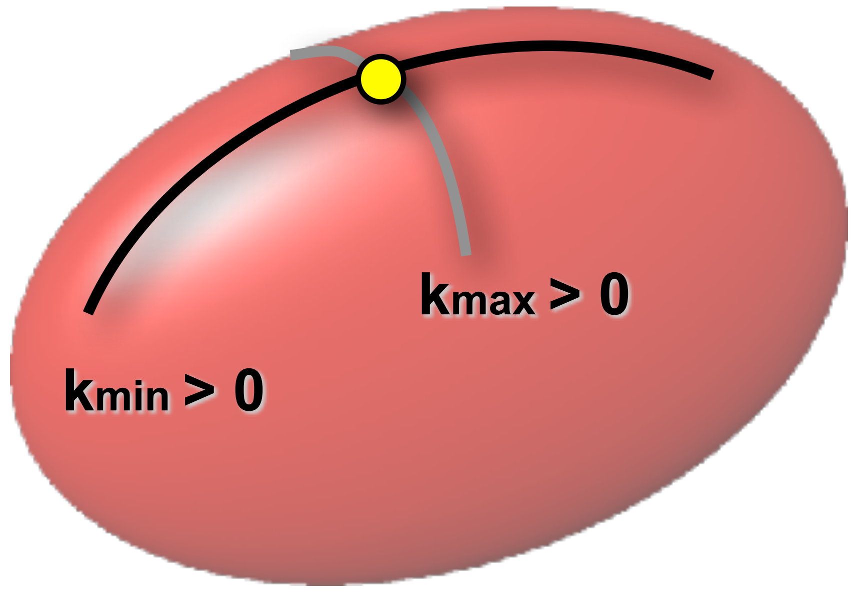
spherical



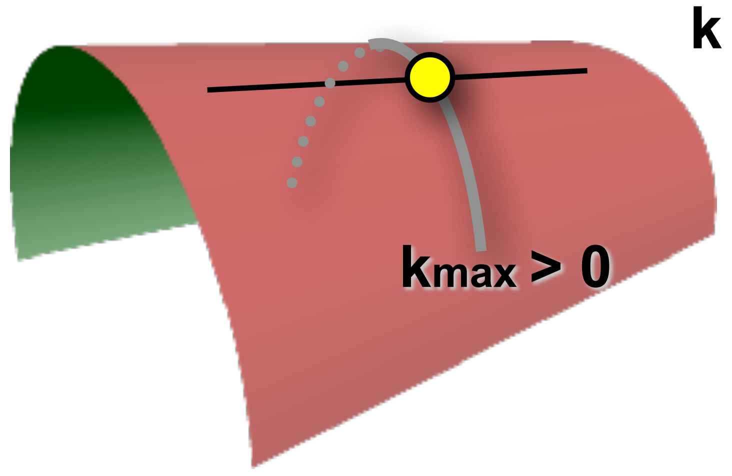
planar

Anisotropic

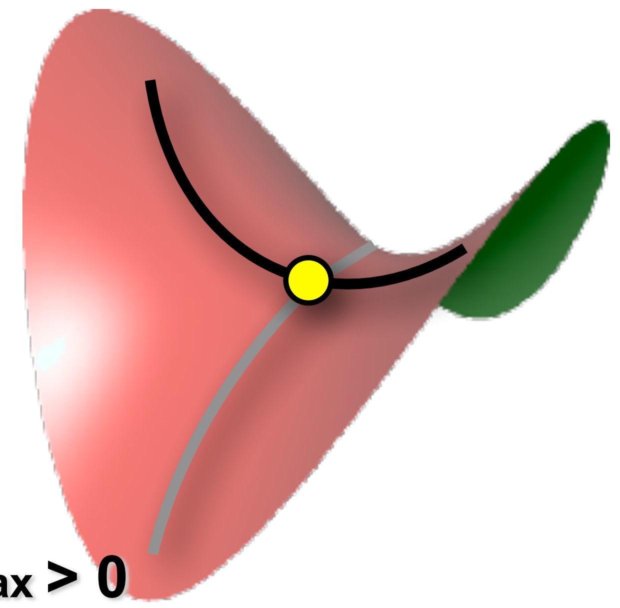
2 principal directions



elliptic

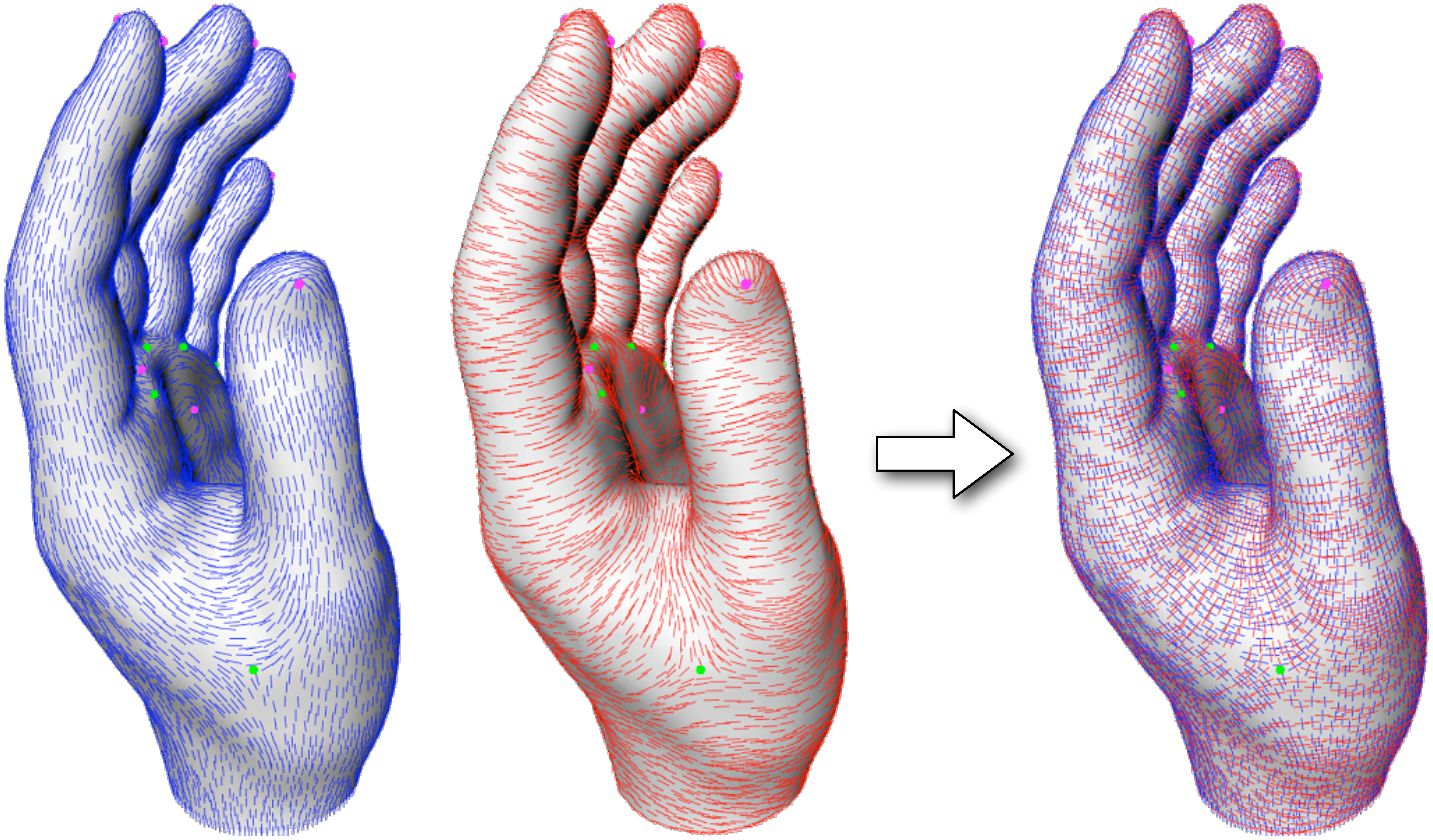


parabolic



hyperbolic

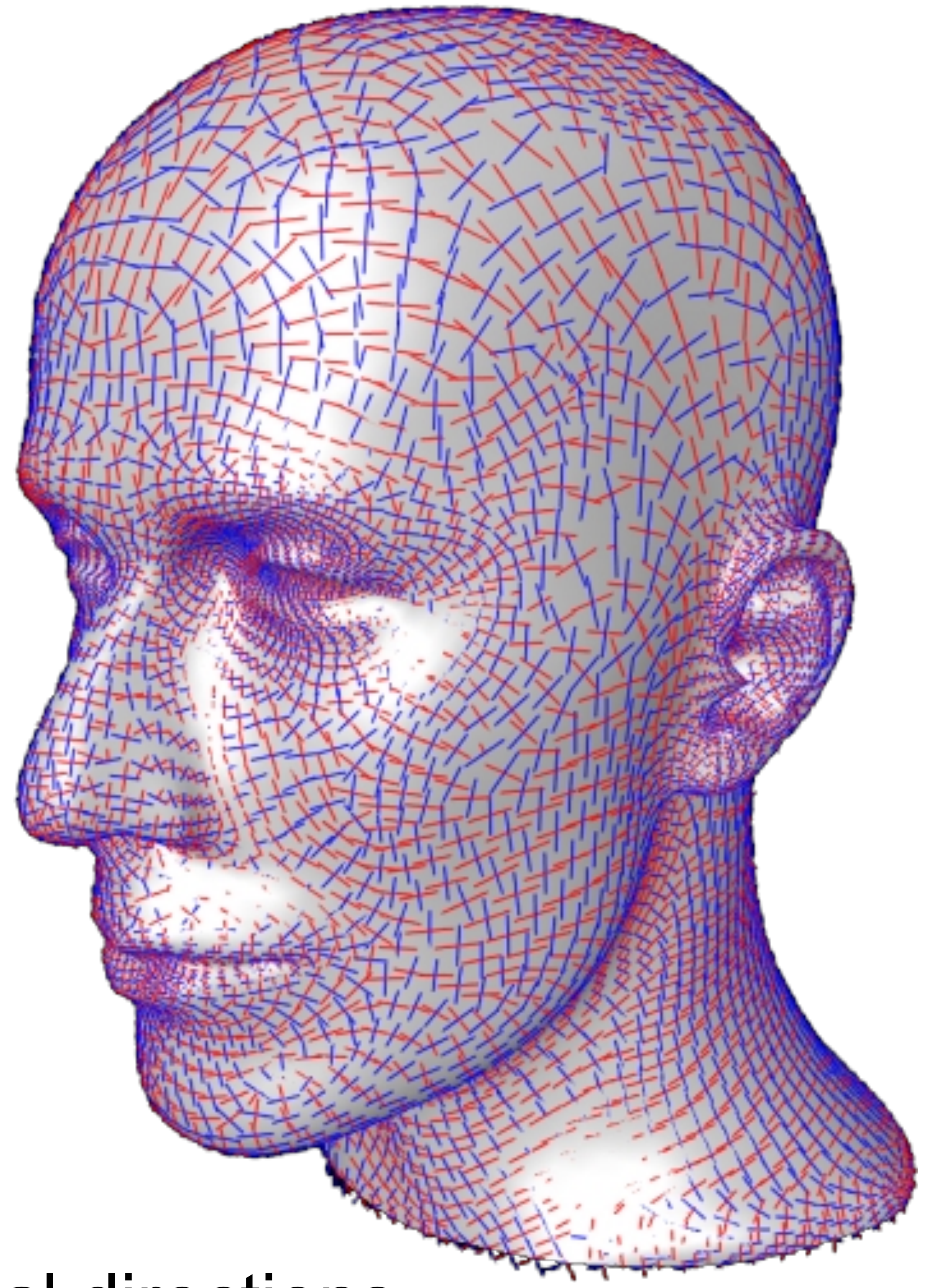
Principal Directions Fields



min curvature max curvature

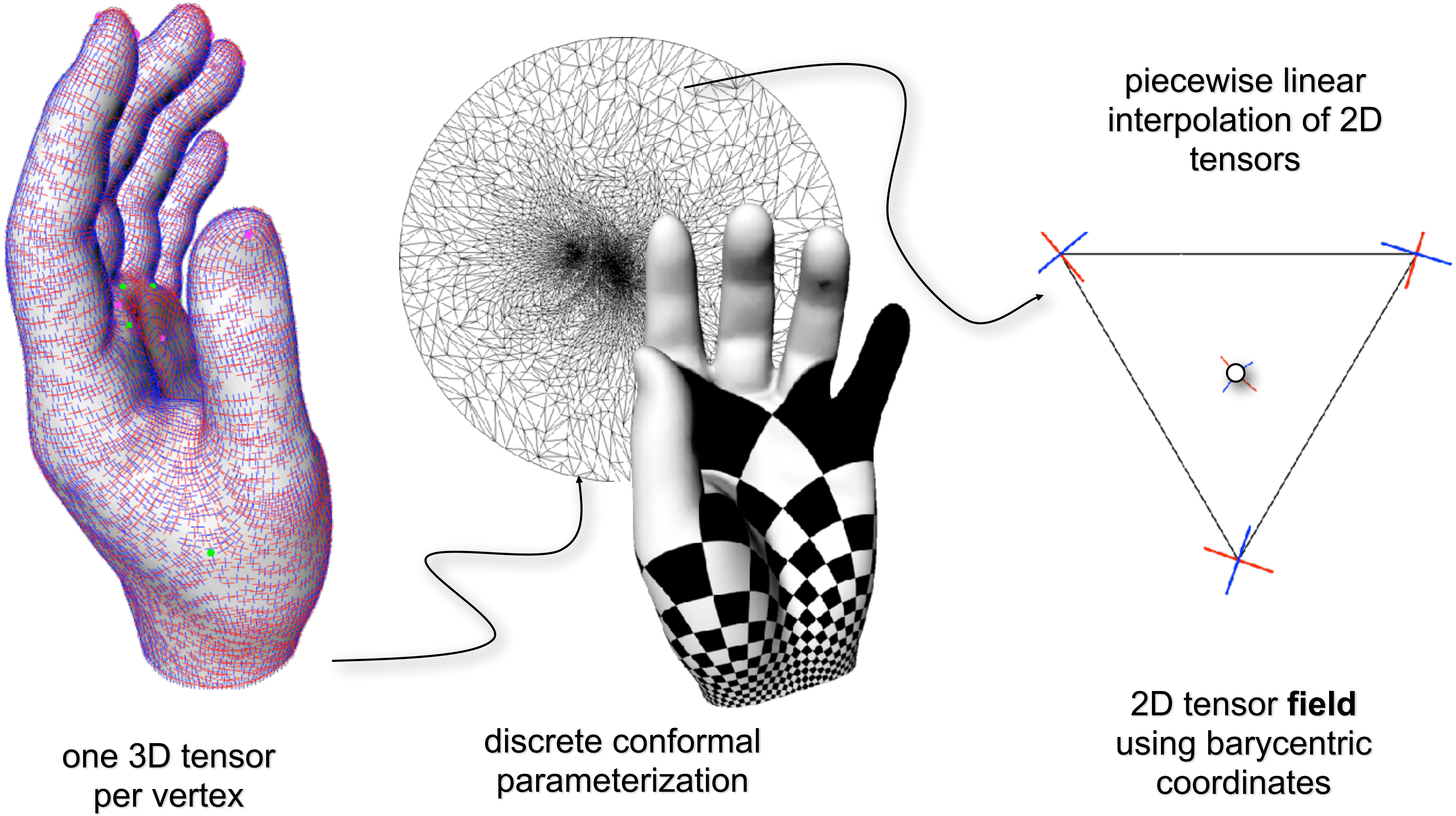
Principal Curvature Directions

- P. Alliez: *Estimating Curvature Tensors on Triangle Meshes* (source code)
  - http://www-sop.inria.fr/geometrica/team/Pierre.Alliez/demos/curvature/



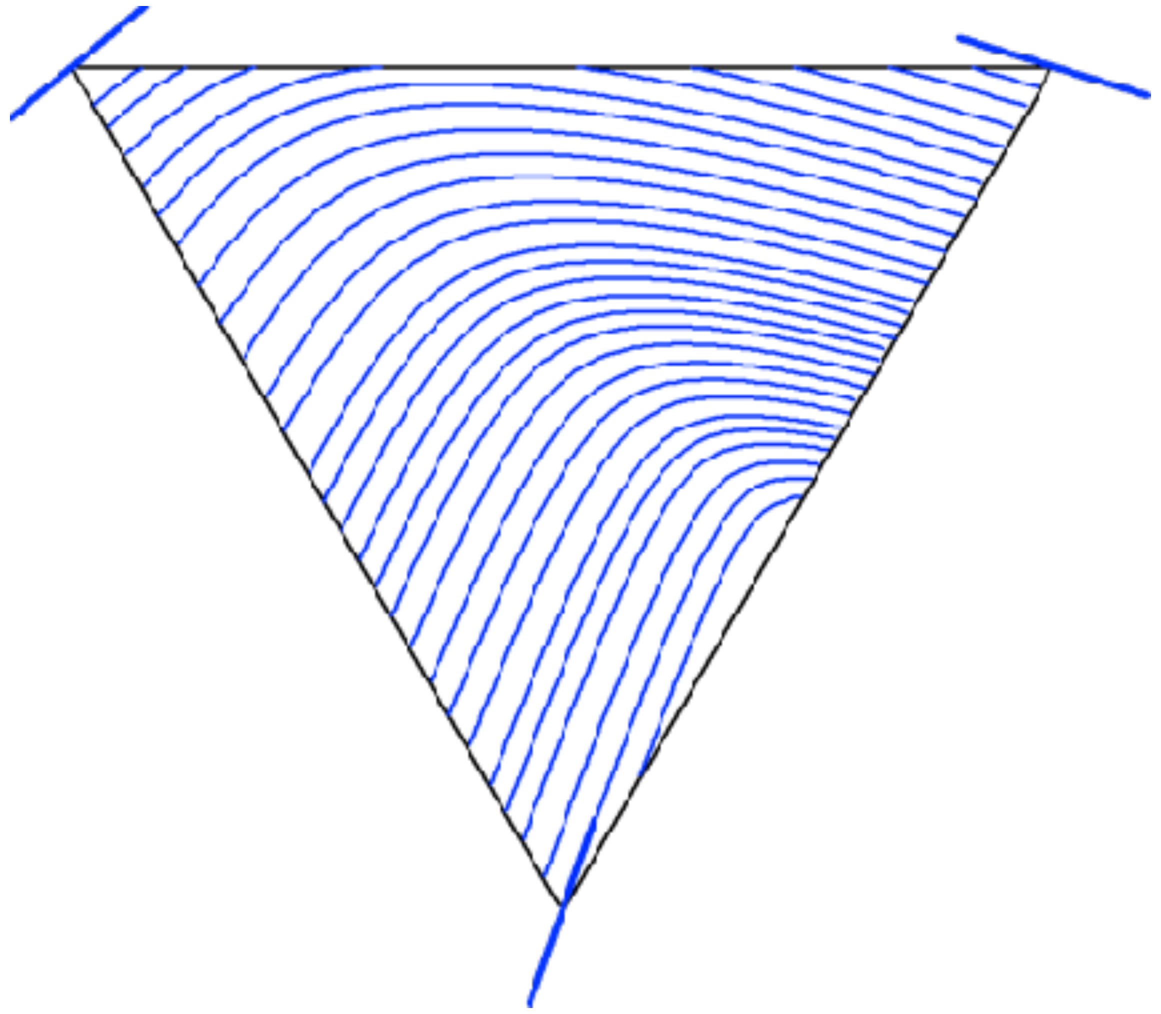
principal directions

Flattening to 2D

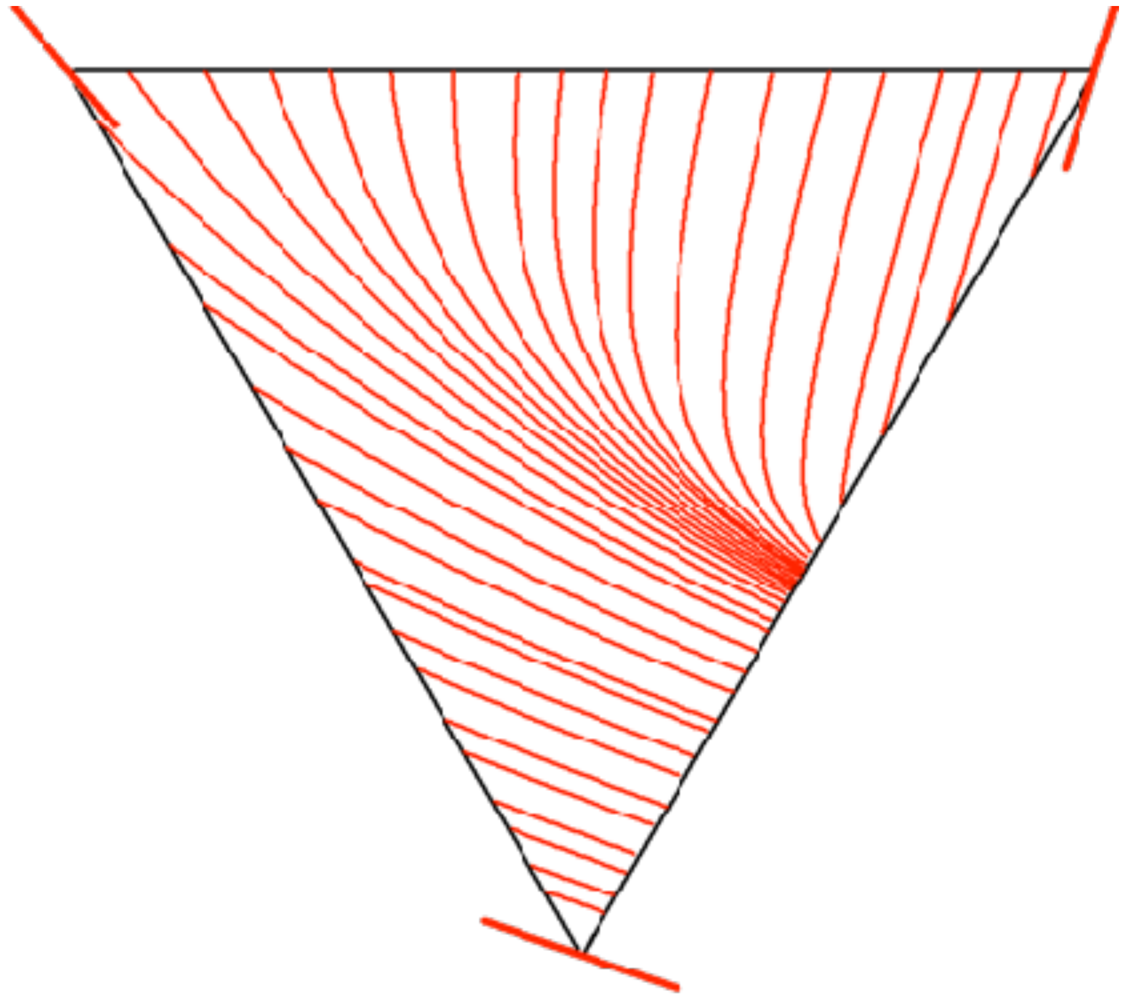


2D Direction Fields

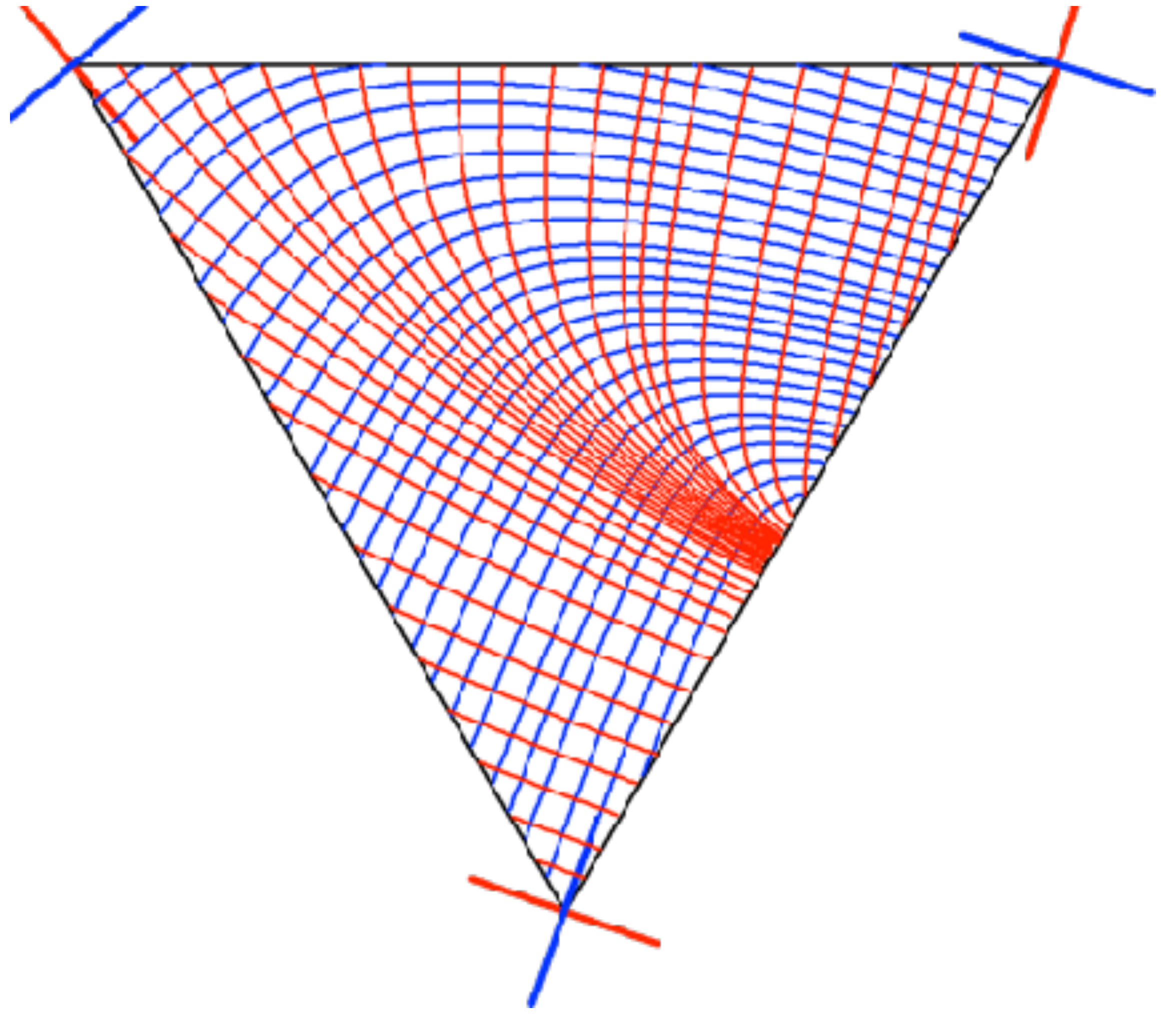
- Regular case



minor foliation



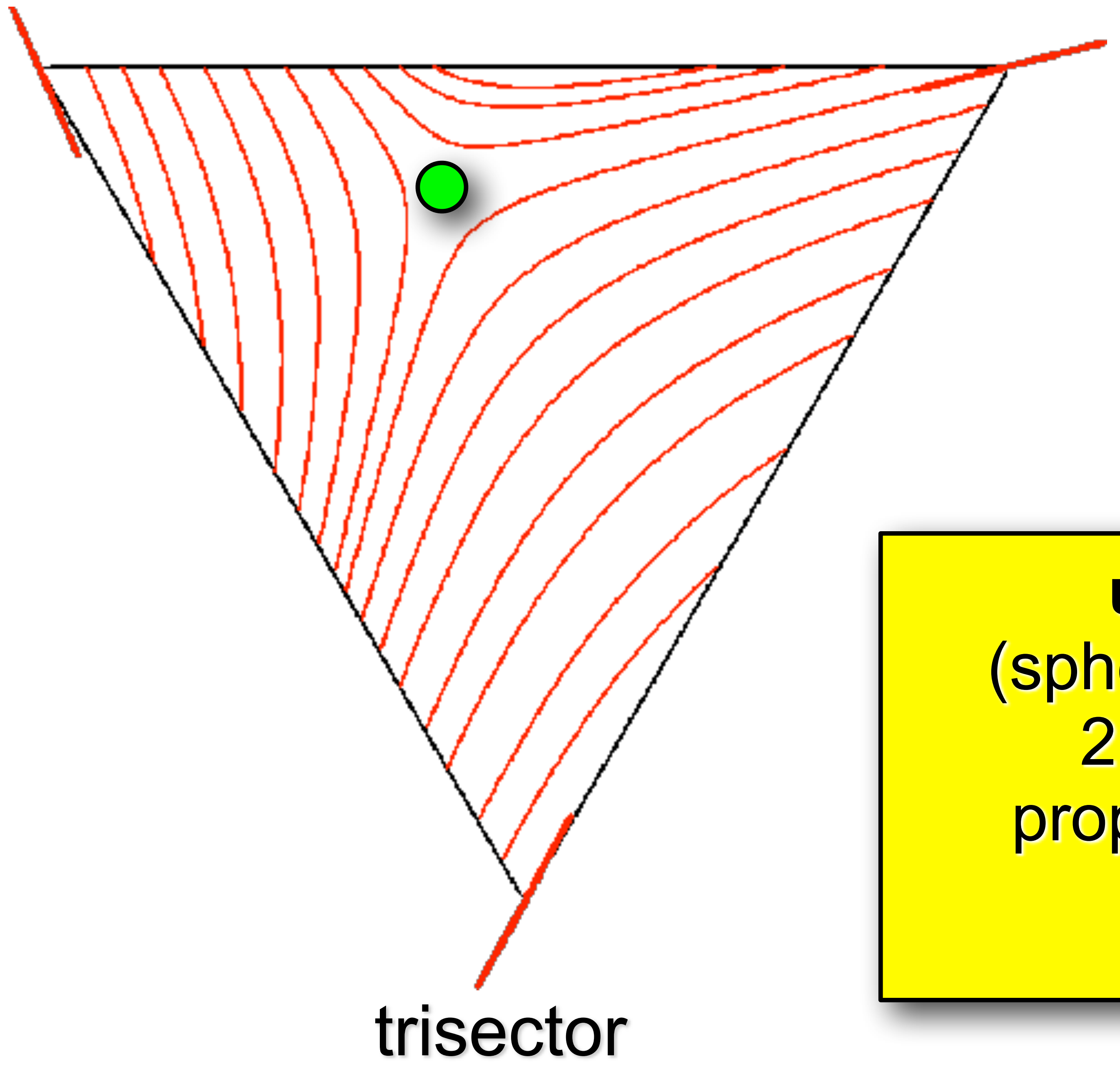
major foliation

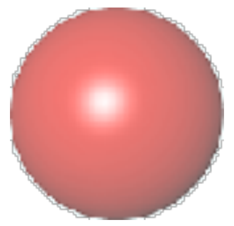


principal foliations

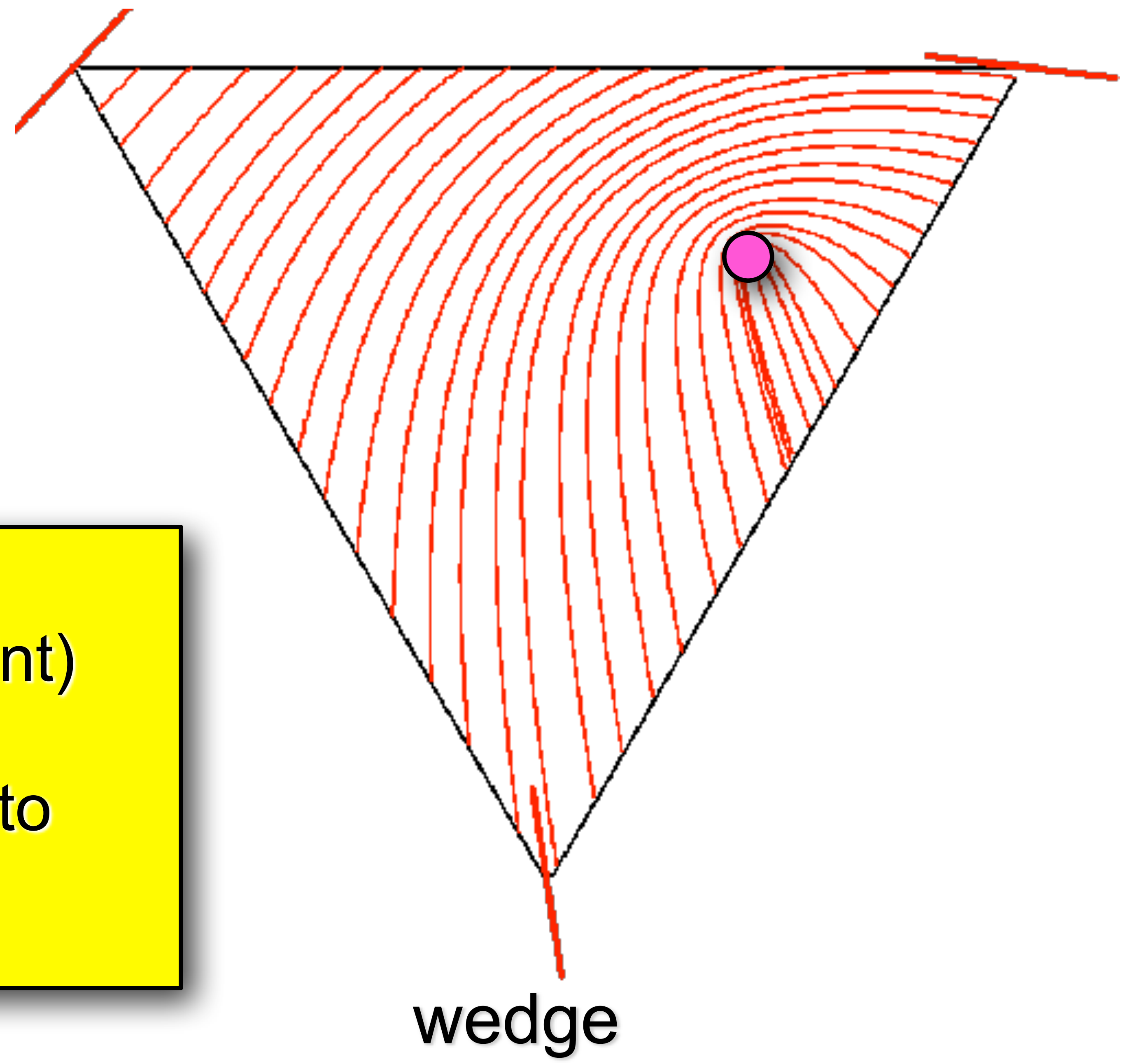
2D Direction Fields

- Singularities

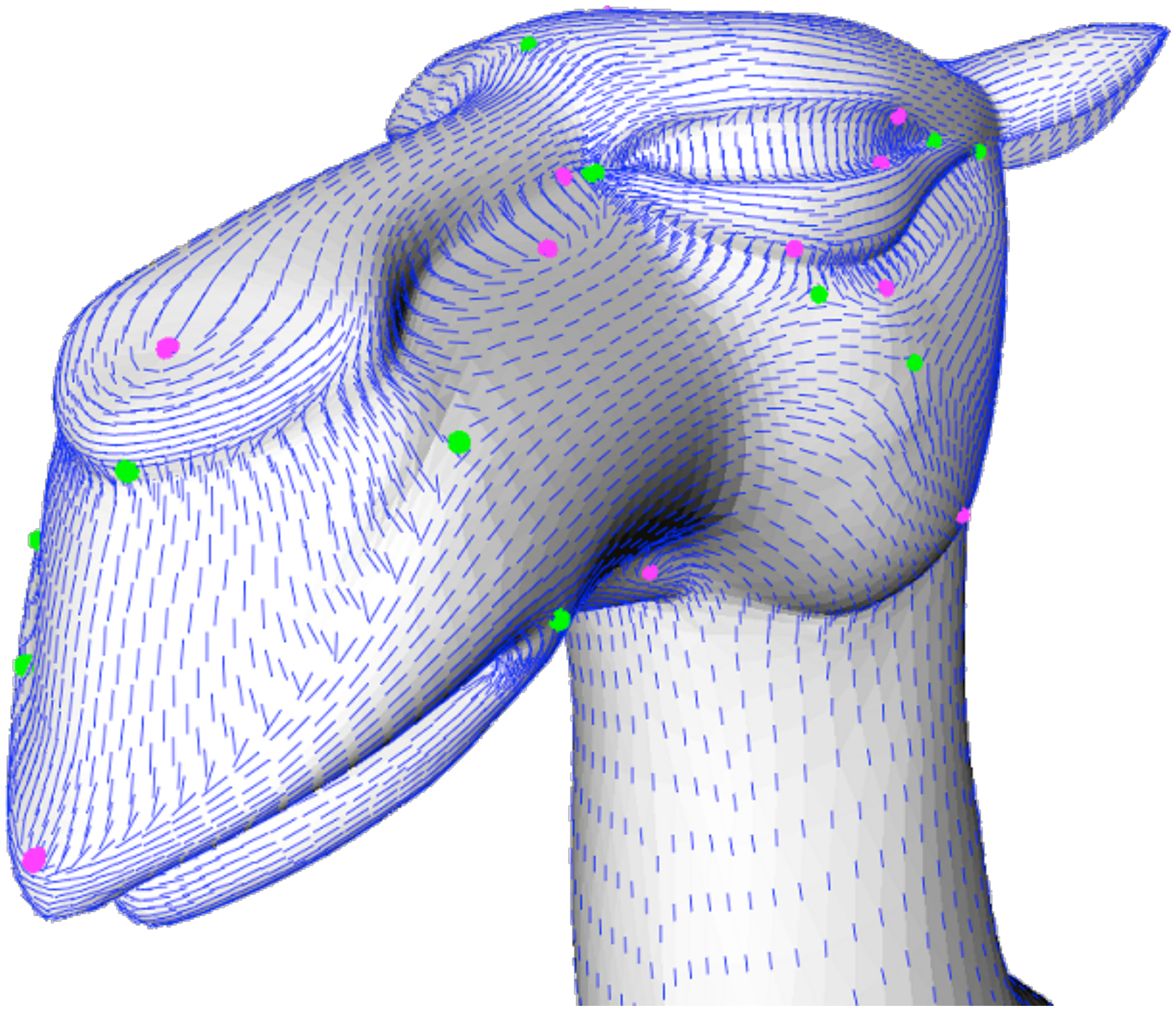




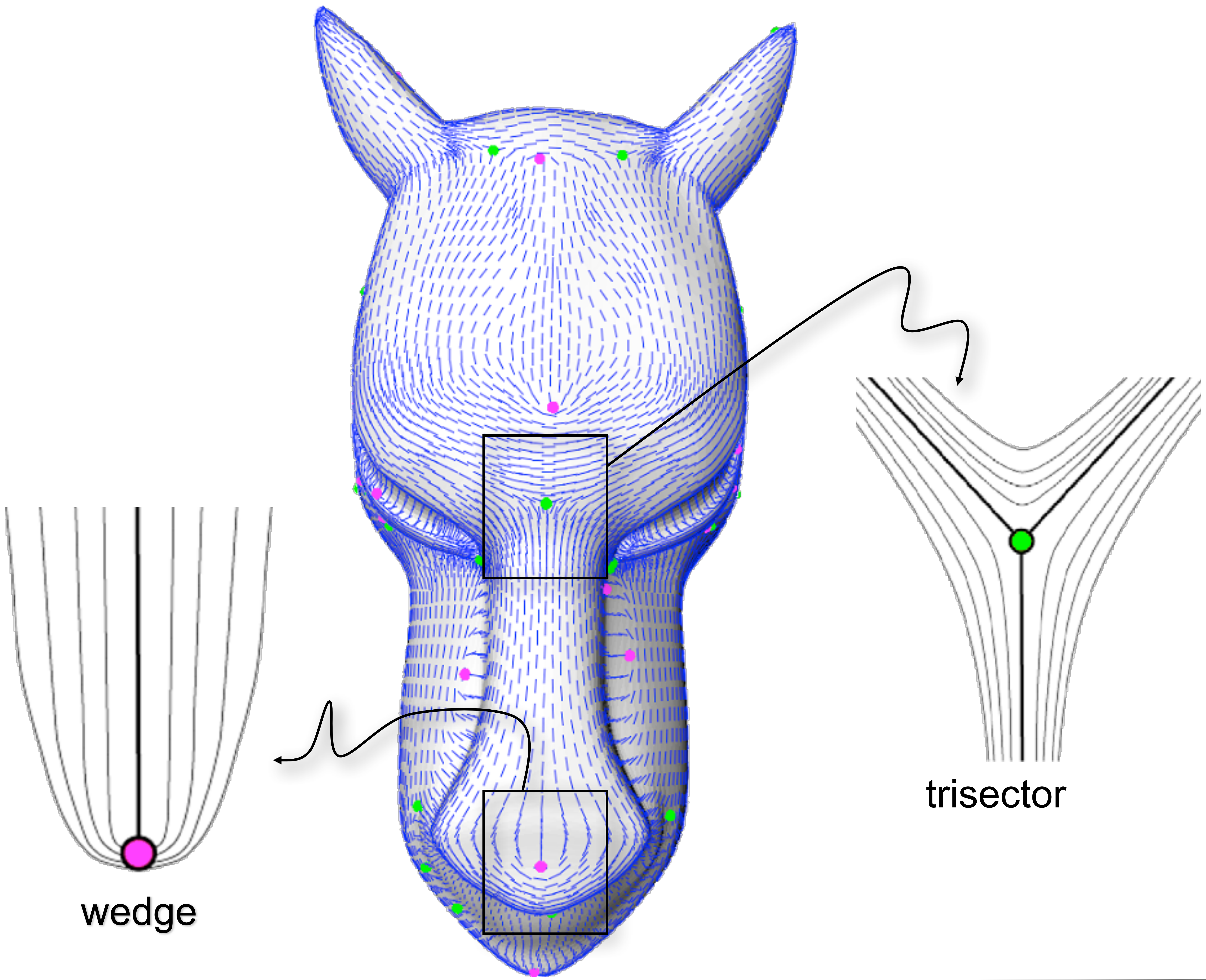
umbilic
(spherical point)
2D tensor
proportional to
identity



Umbilics

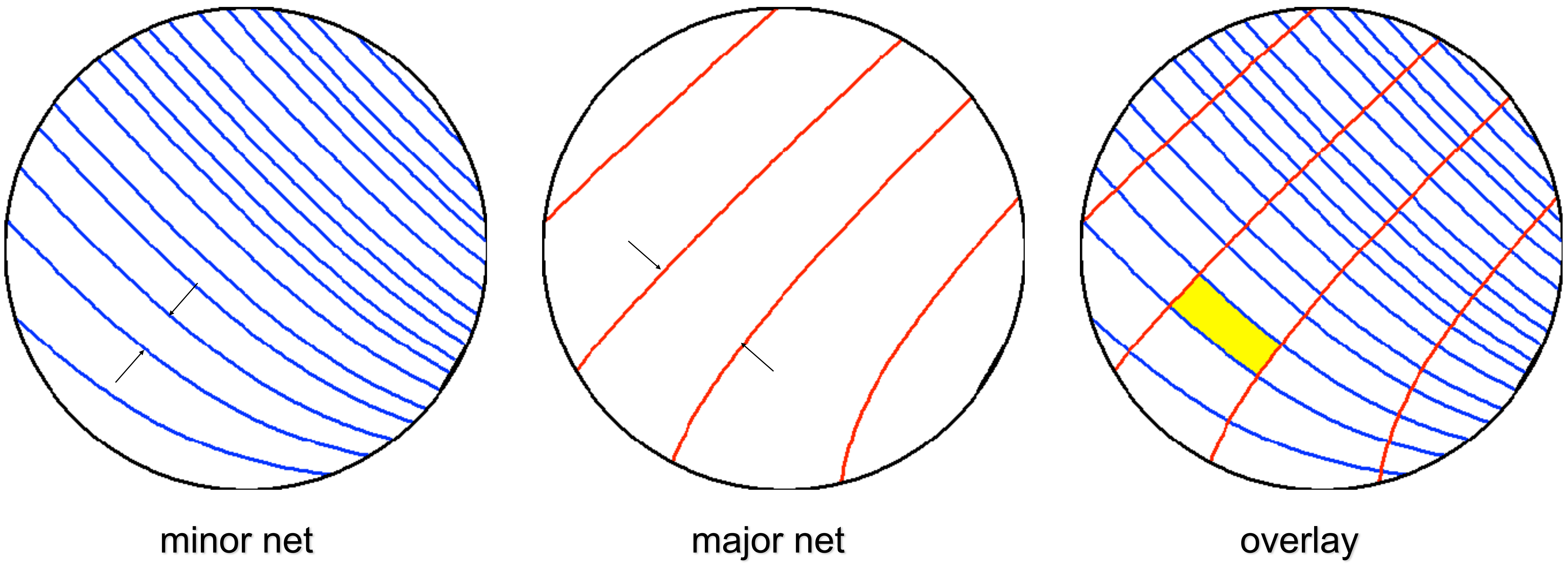


Umbilics



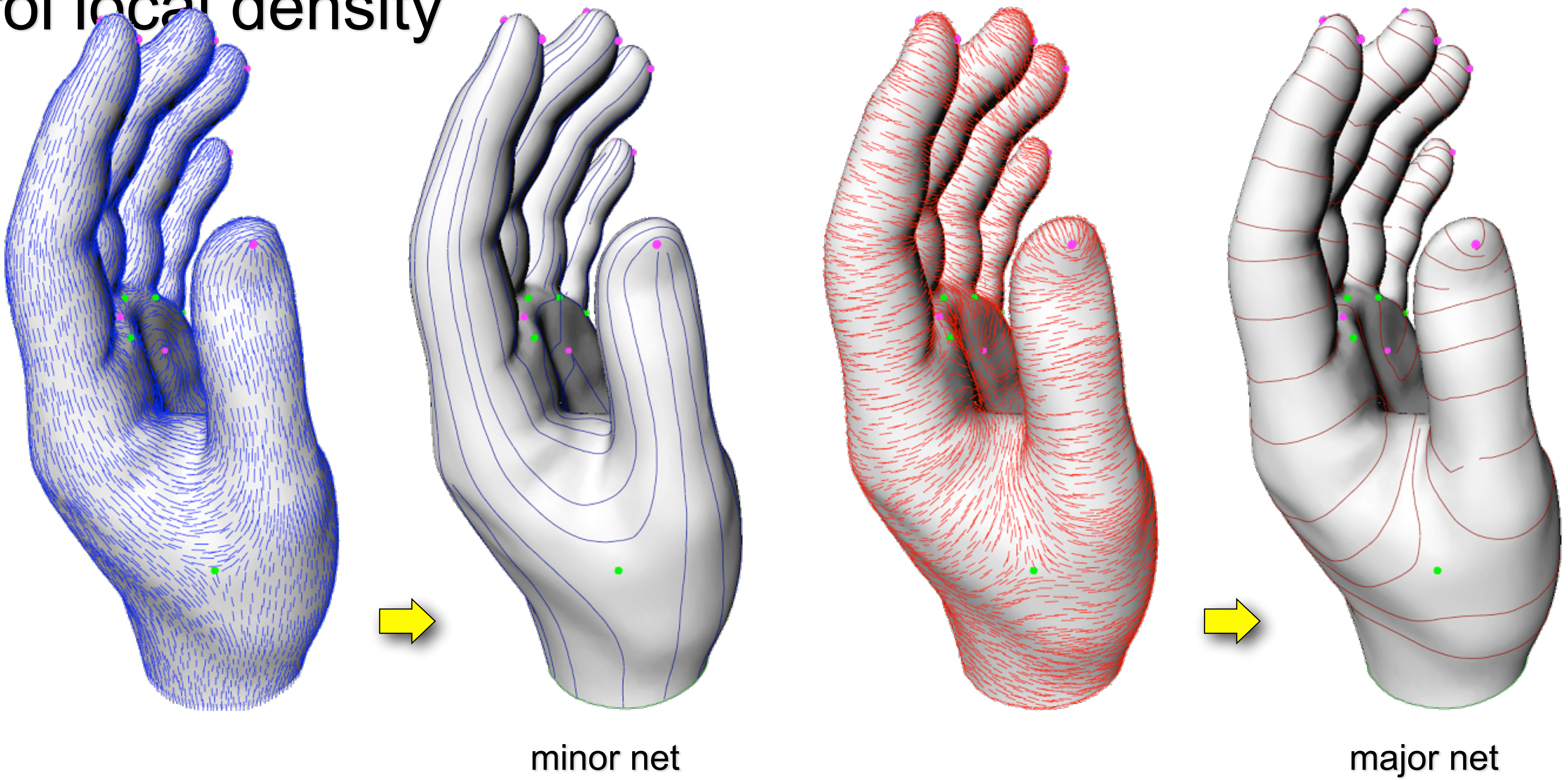
Lines of Curvature

- Control local density



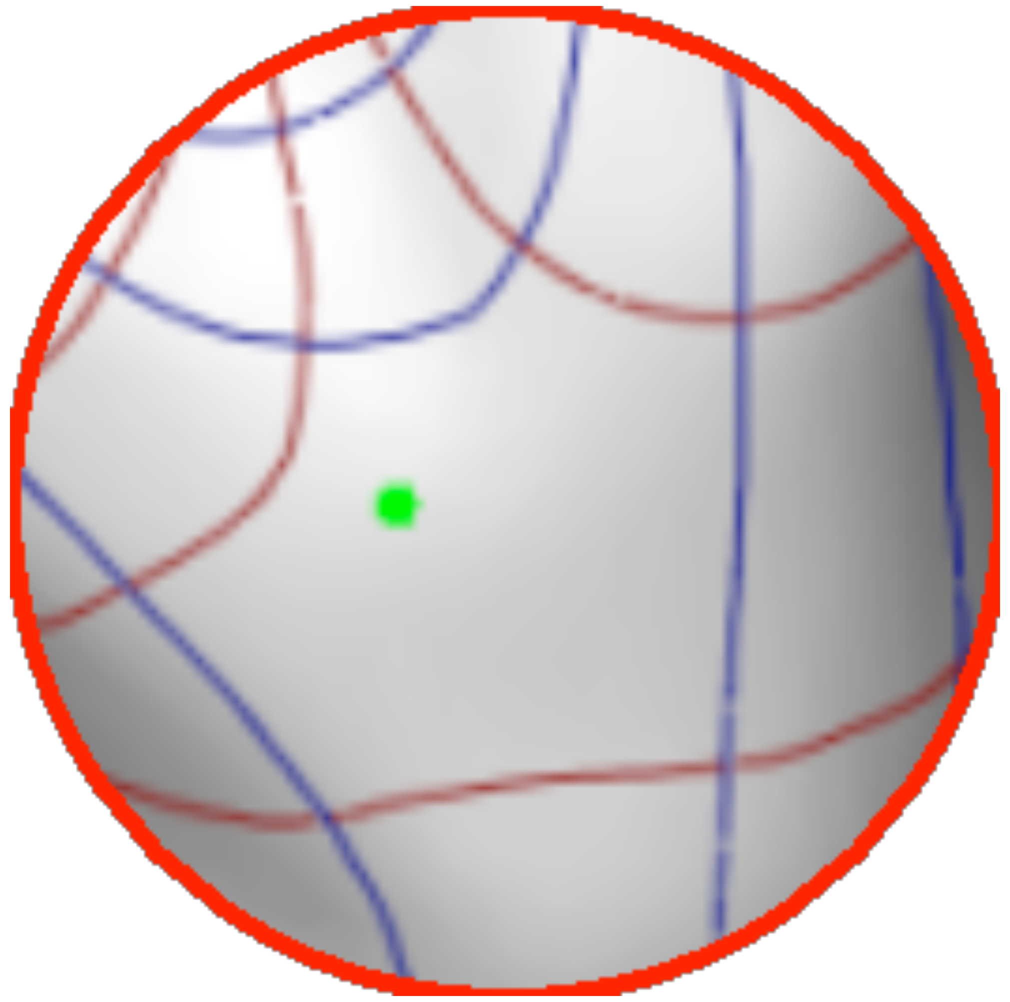
Lines of Curvature

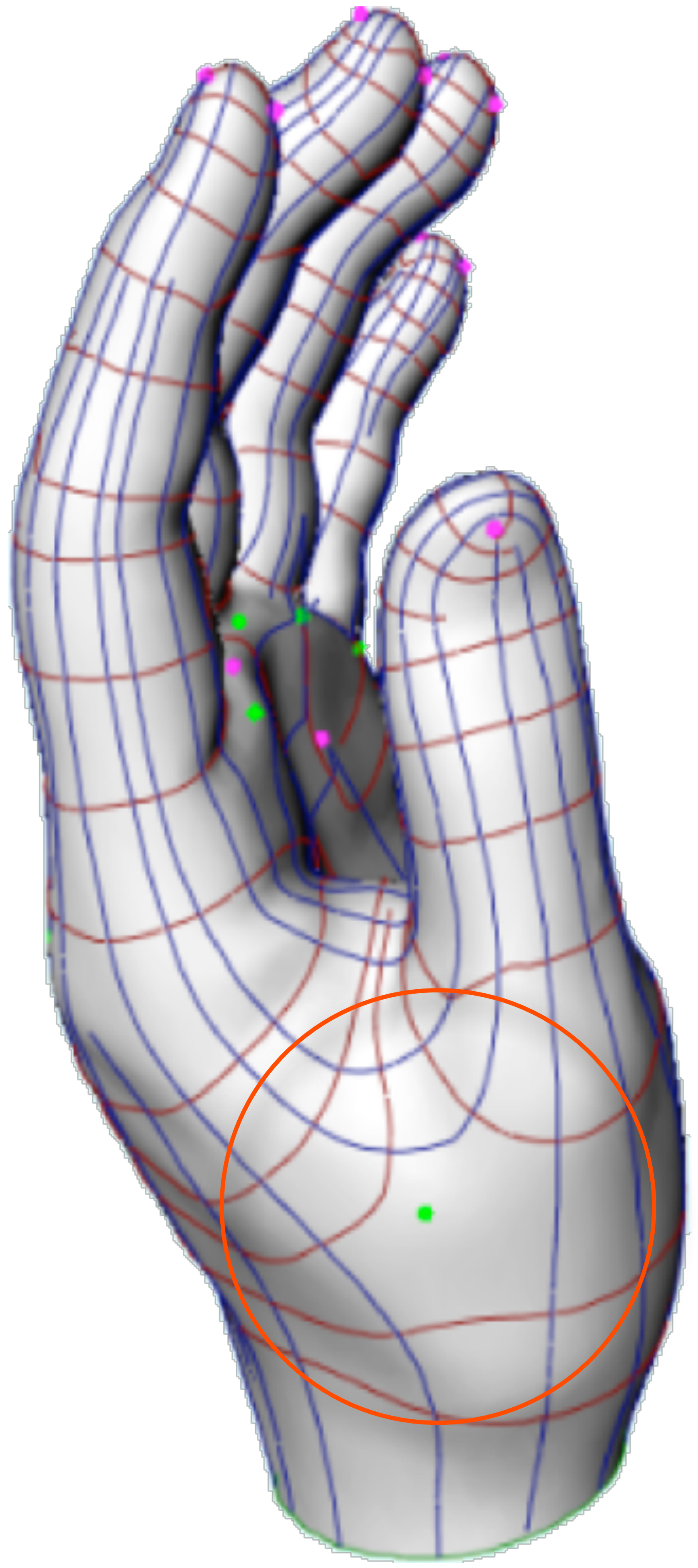
- Control local density



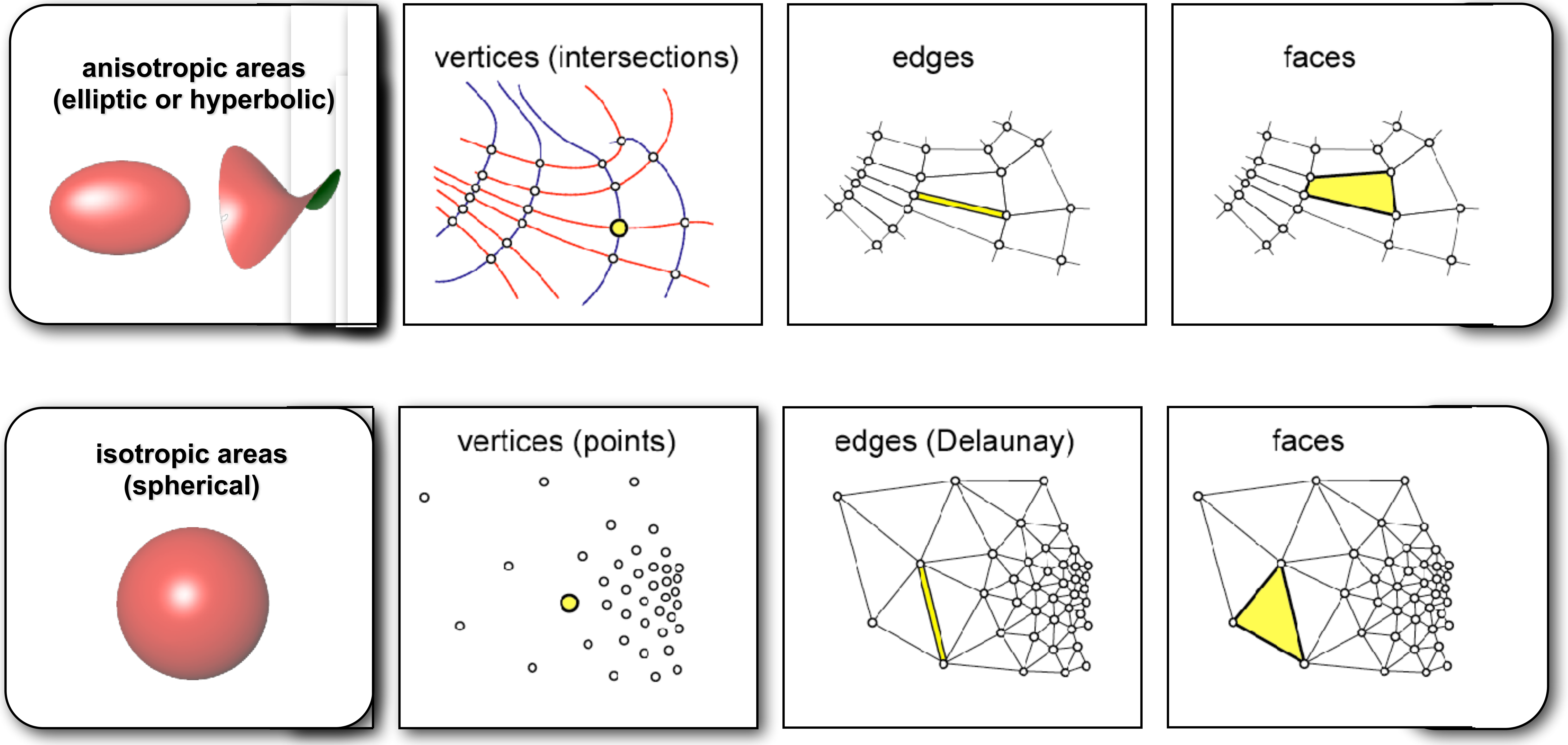
Overlay

- Overlay curvature lines in anisotropic regions
- Add umbilic points in isotropic regions



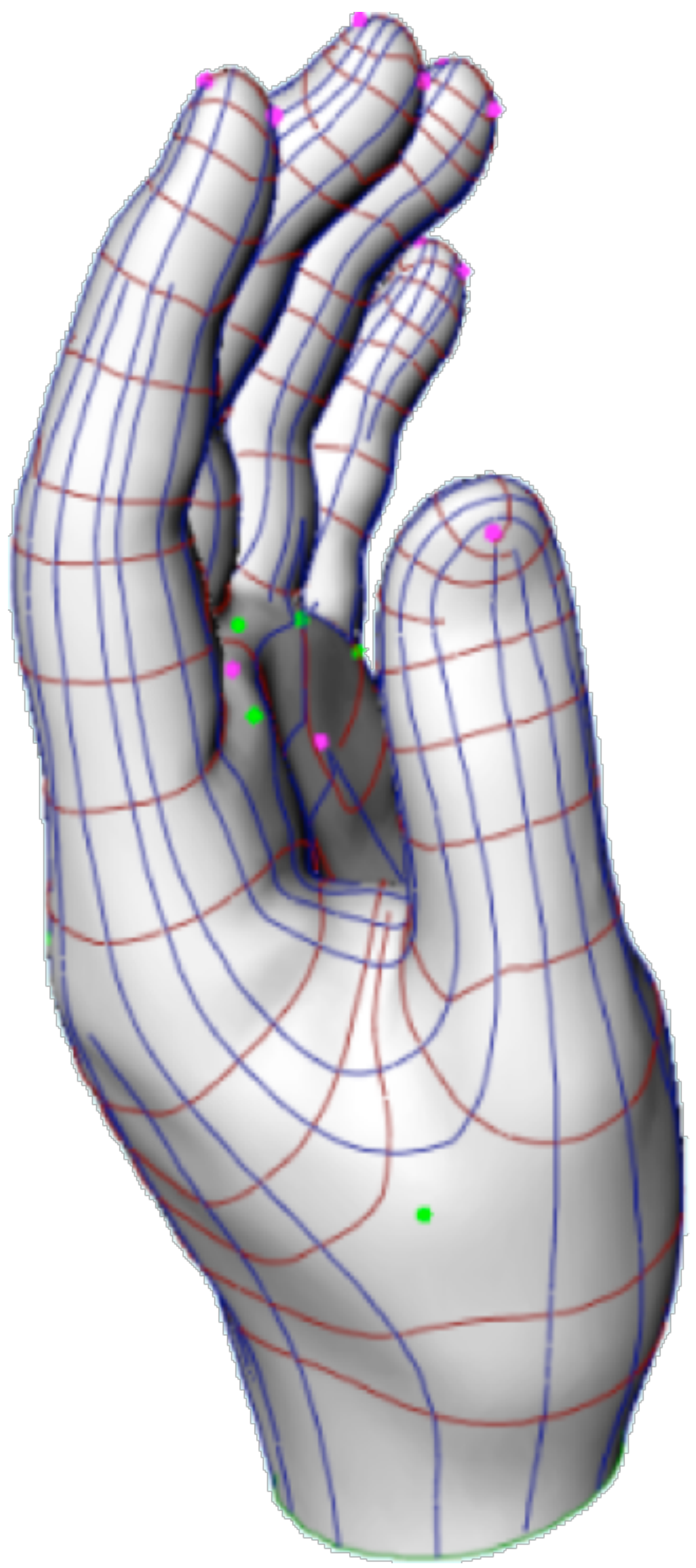


Meshing

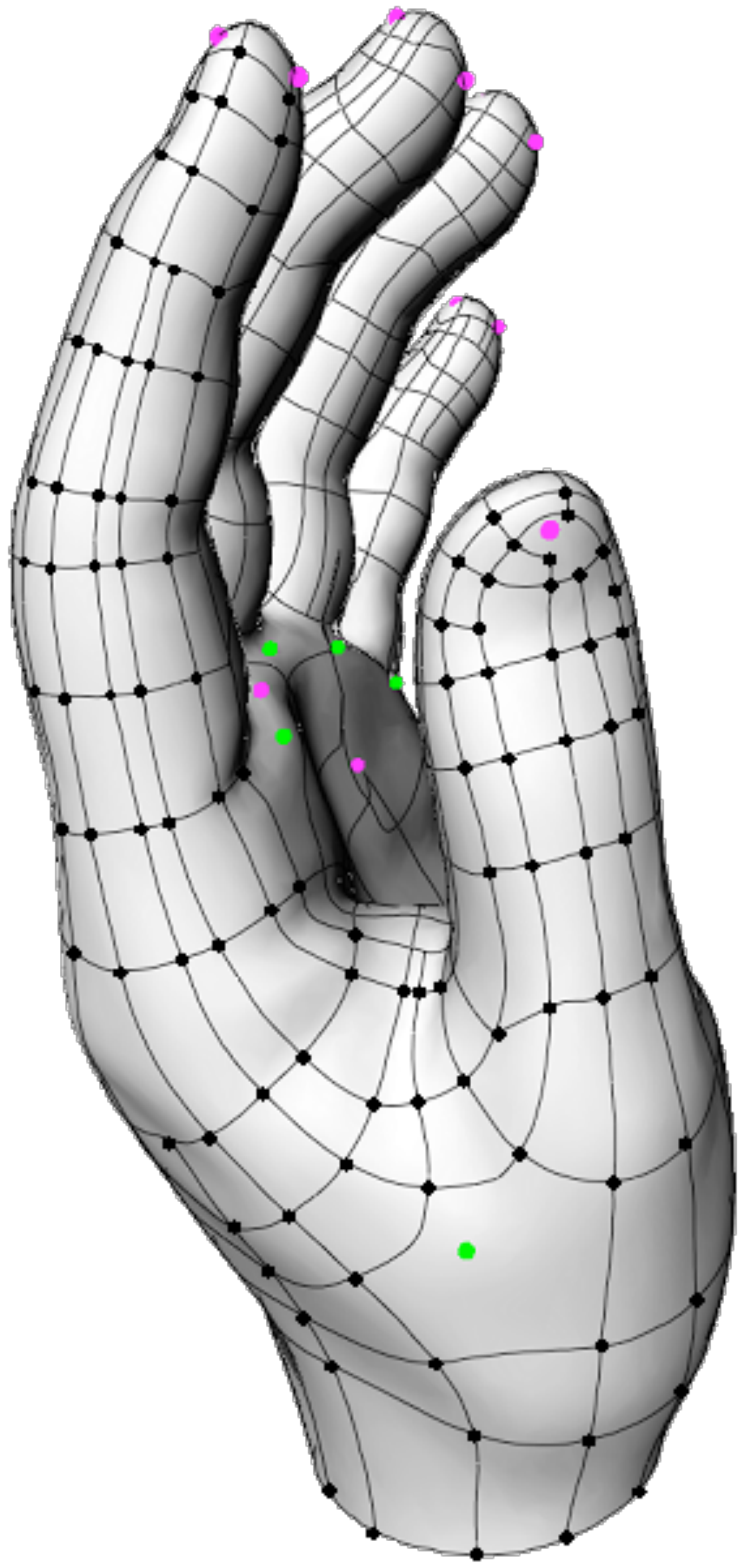


[CGAL] www.cgal.org

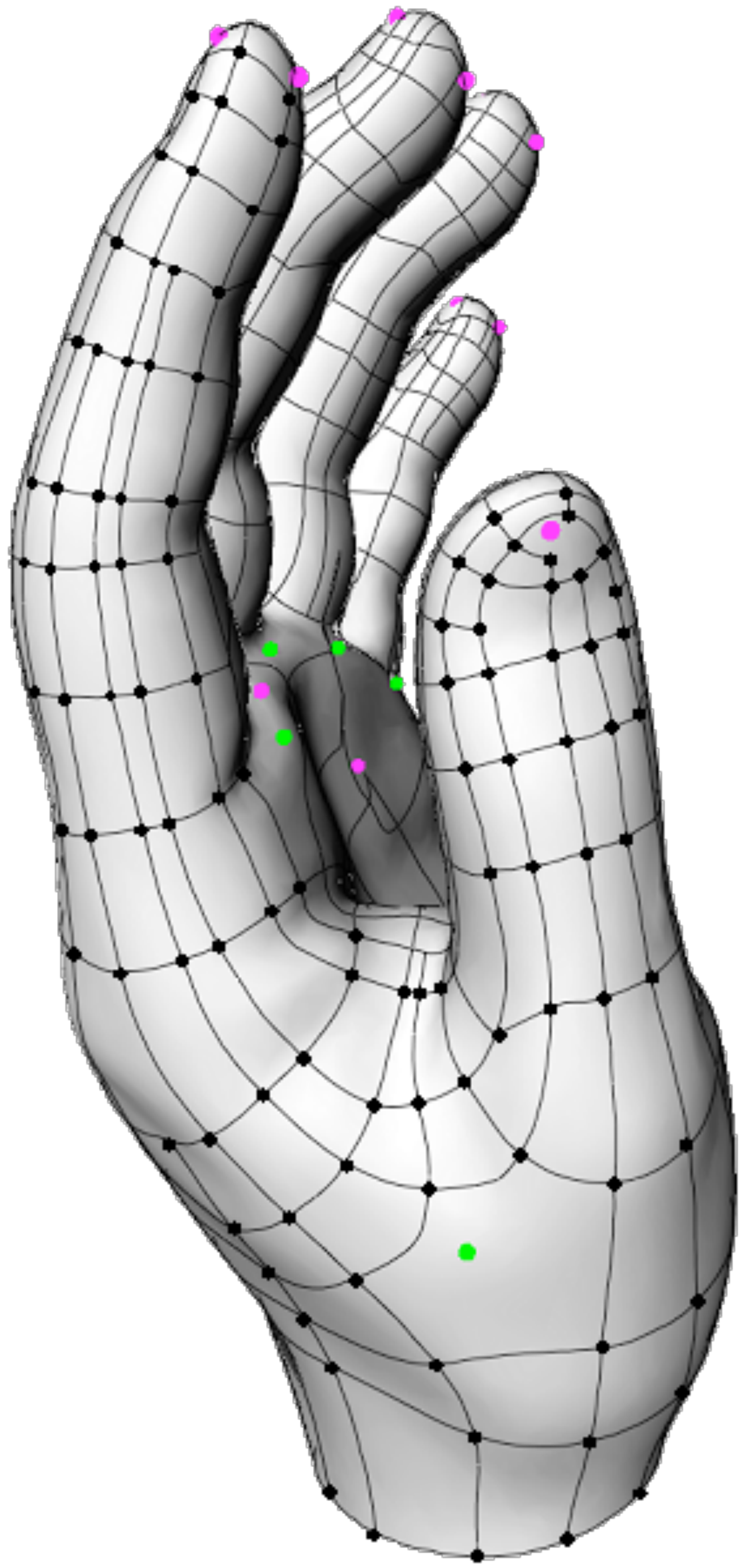
Vertices



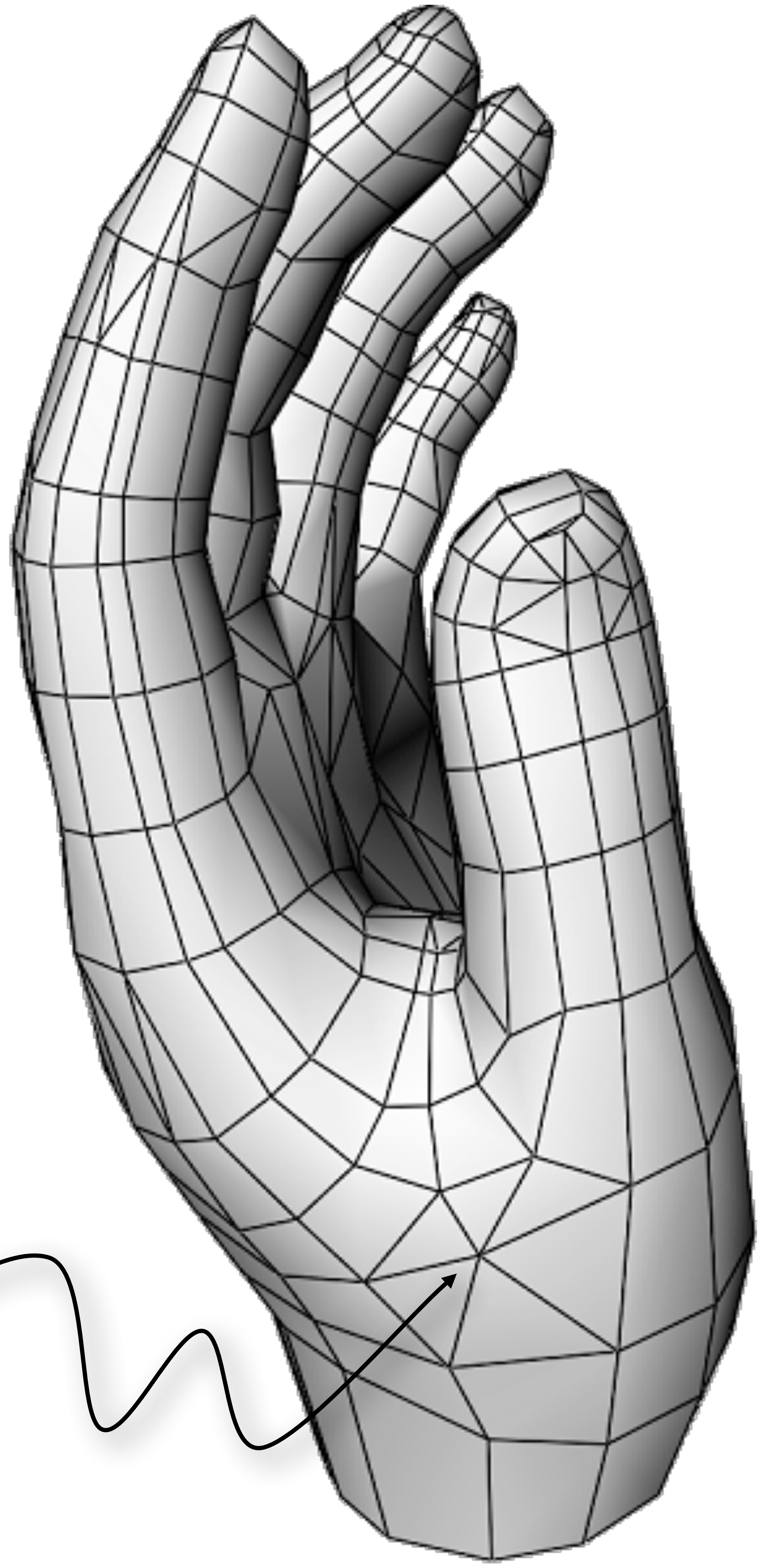
intersect lines of curvatures



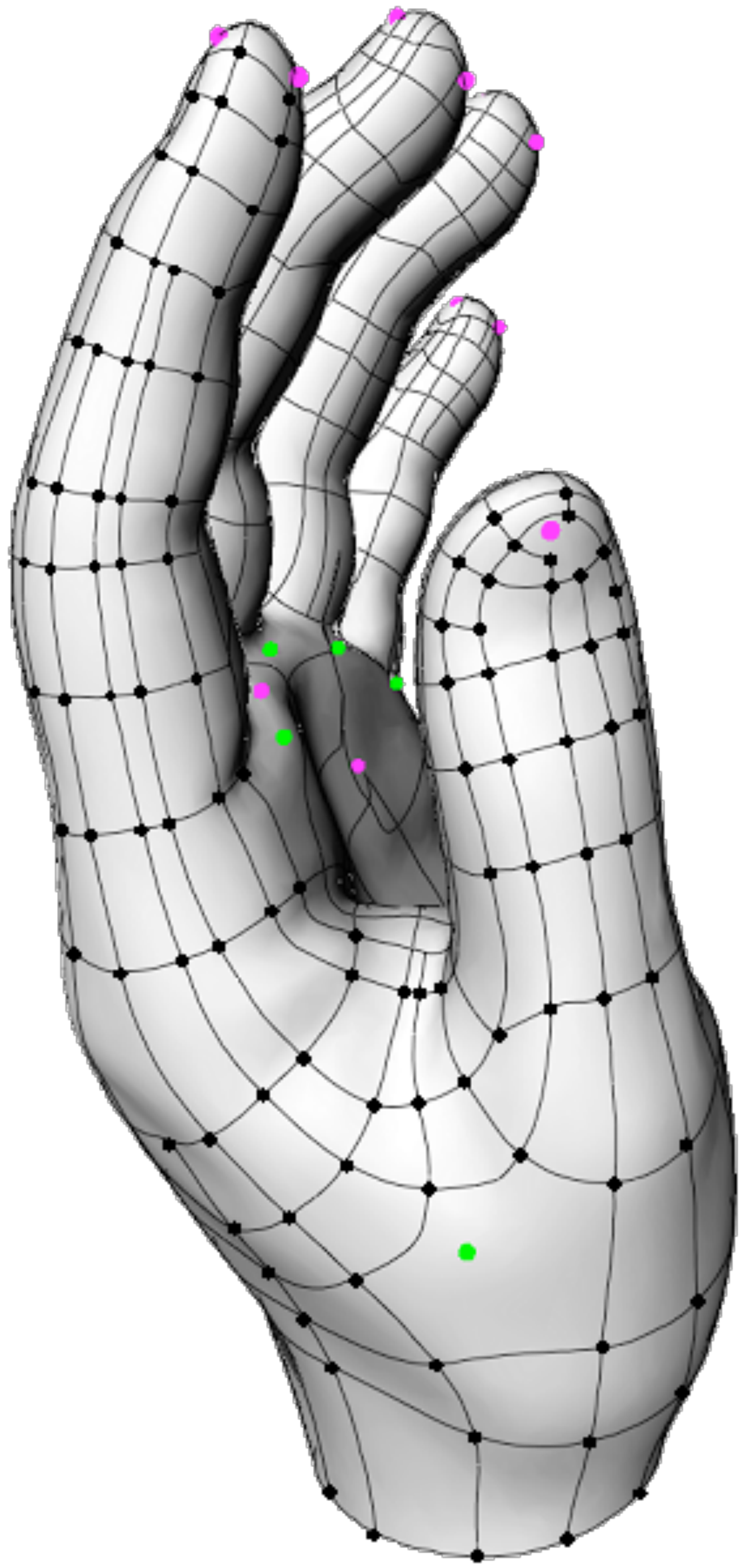
Edges

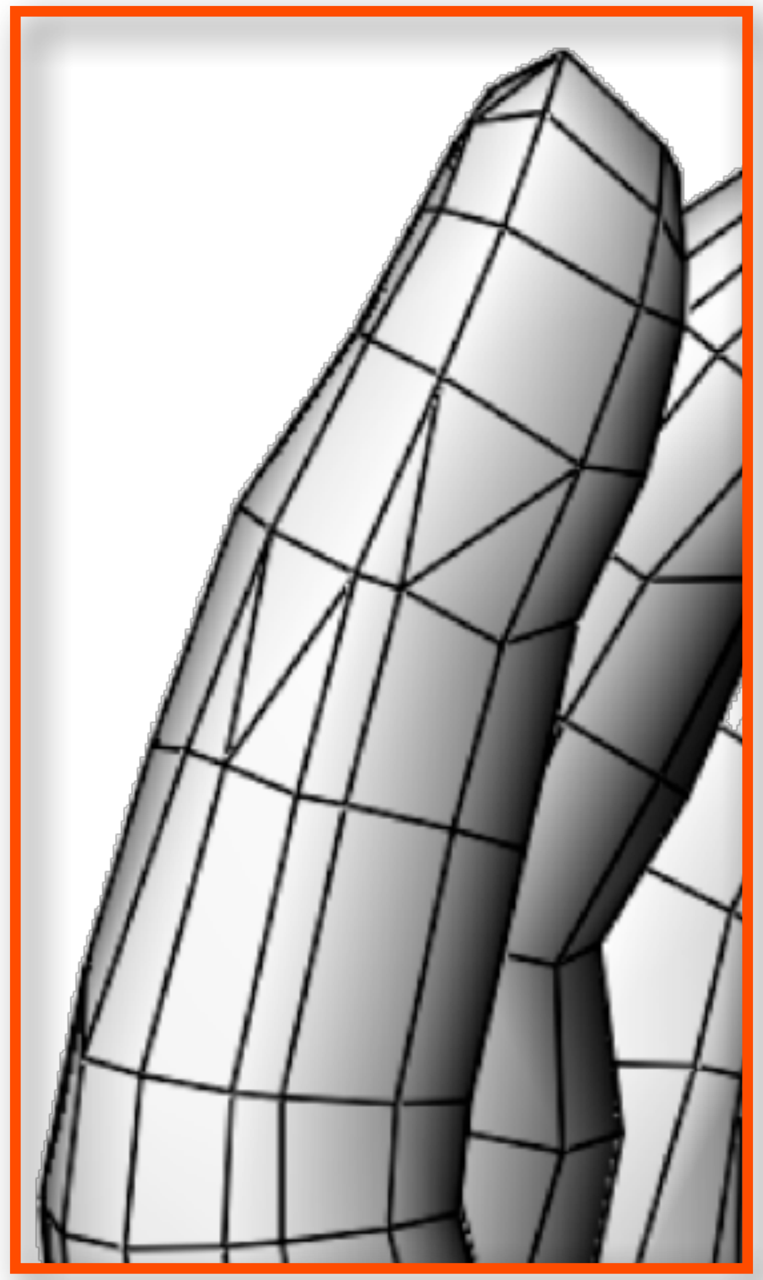


straighten lines of
curvatures
+
Delaunay triangulation
near umbilics

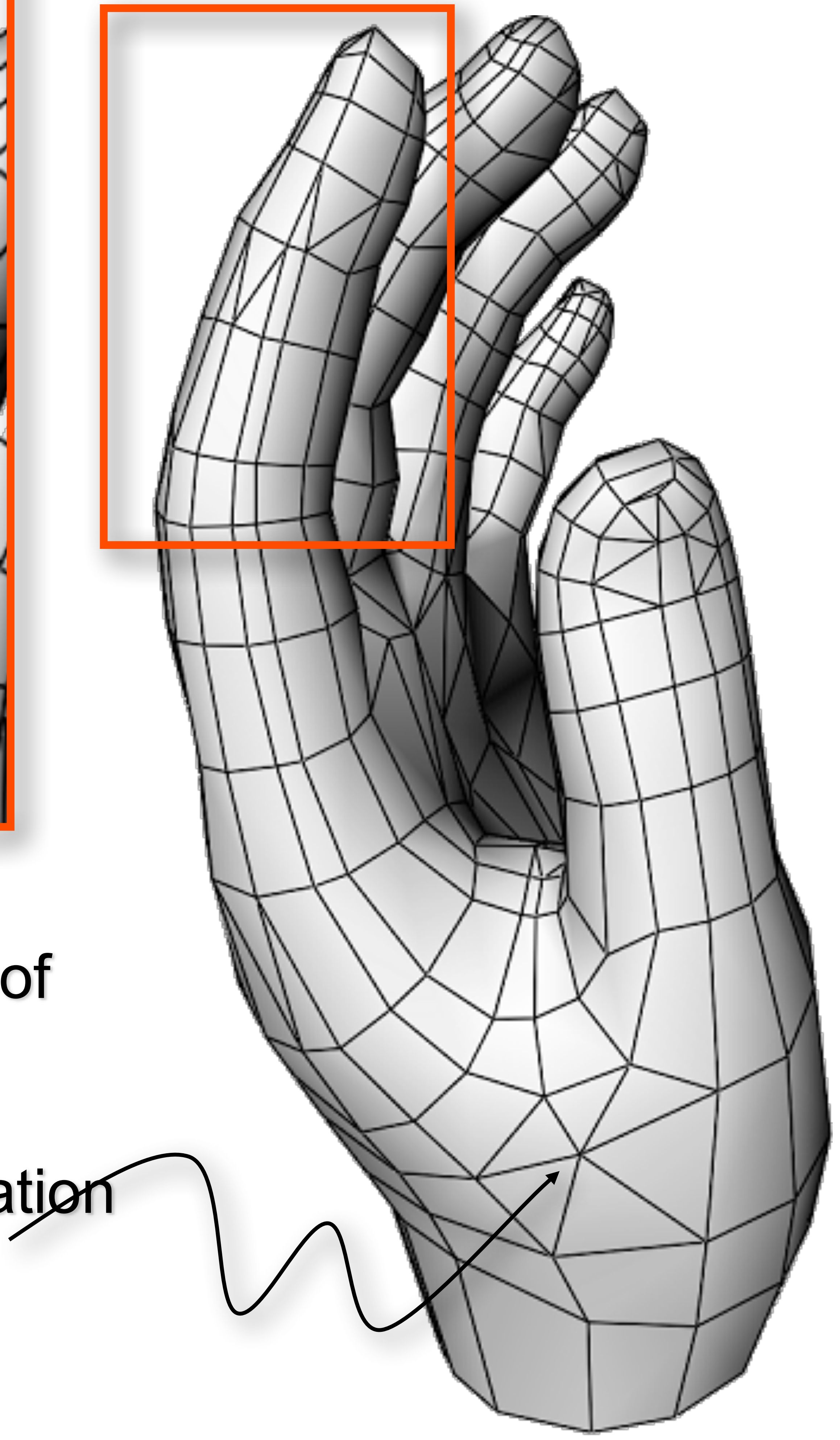


Edges

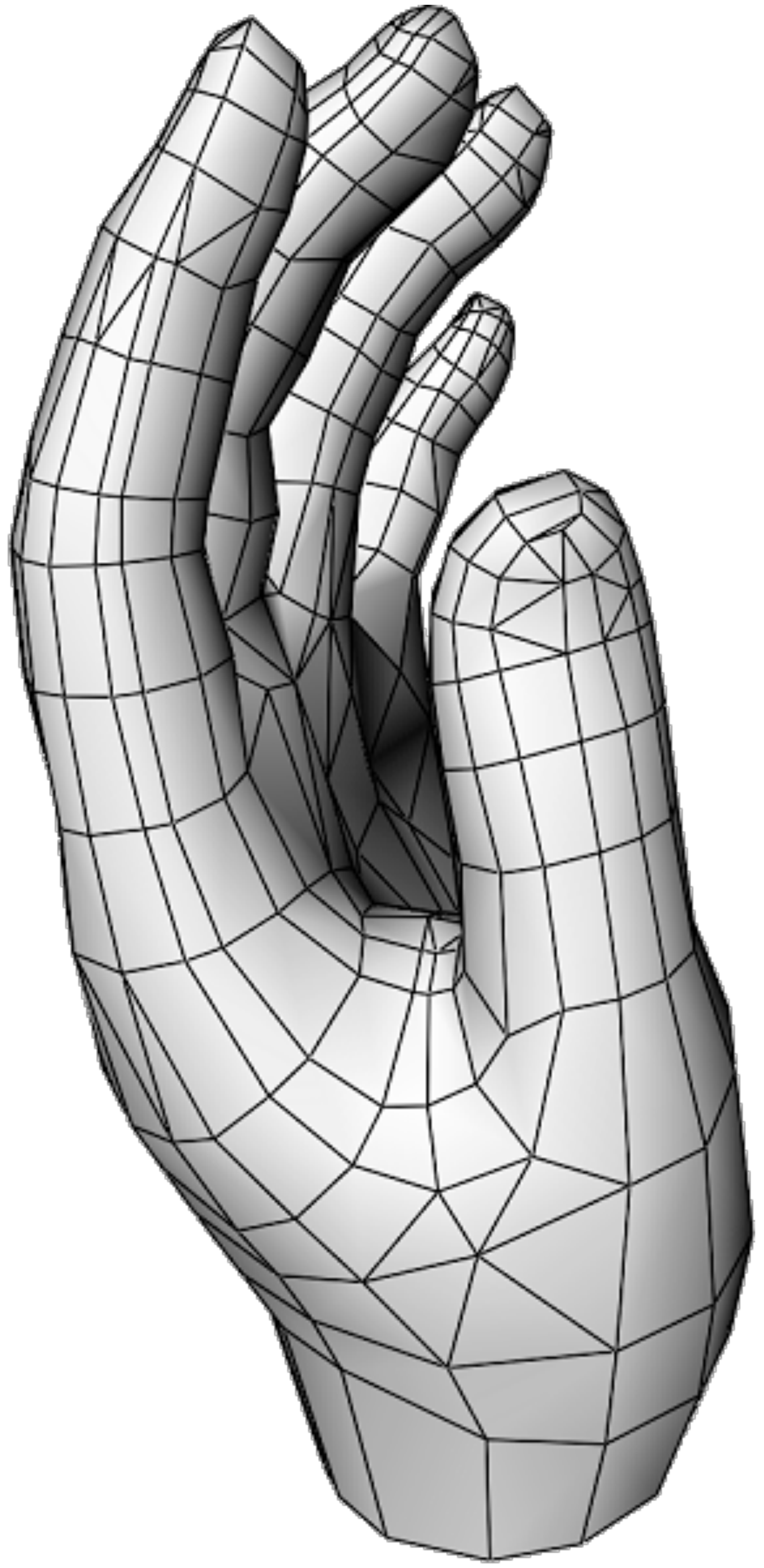


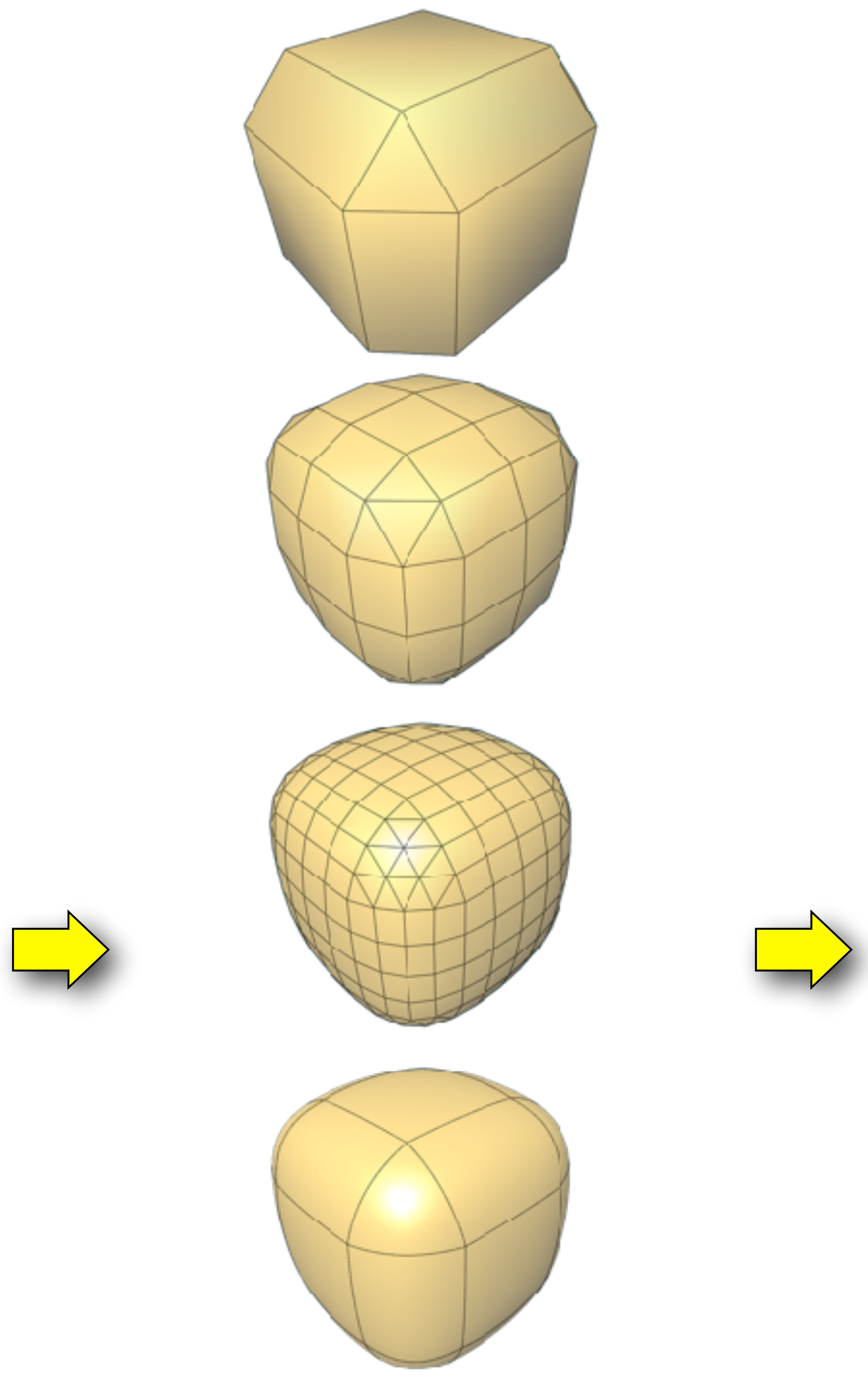


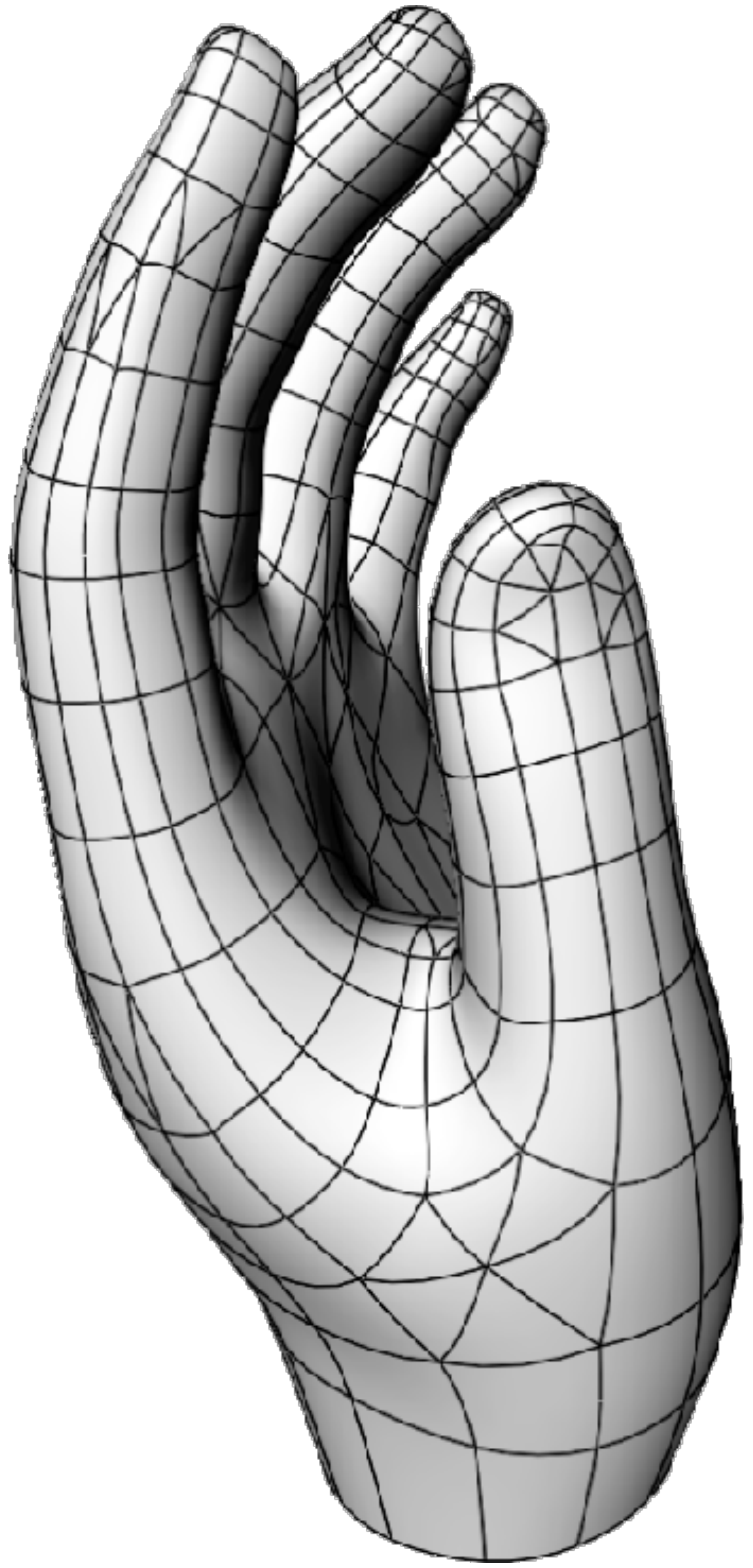
straighten lines of curvatures
+
Delaunay triangulation near umbilics



Smoothing by Subdivision

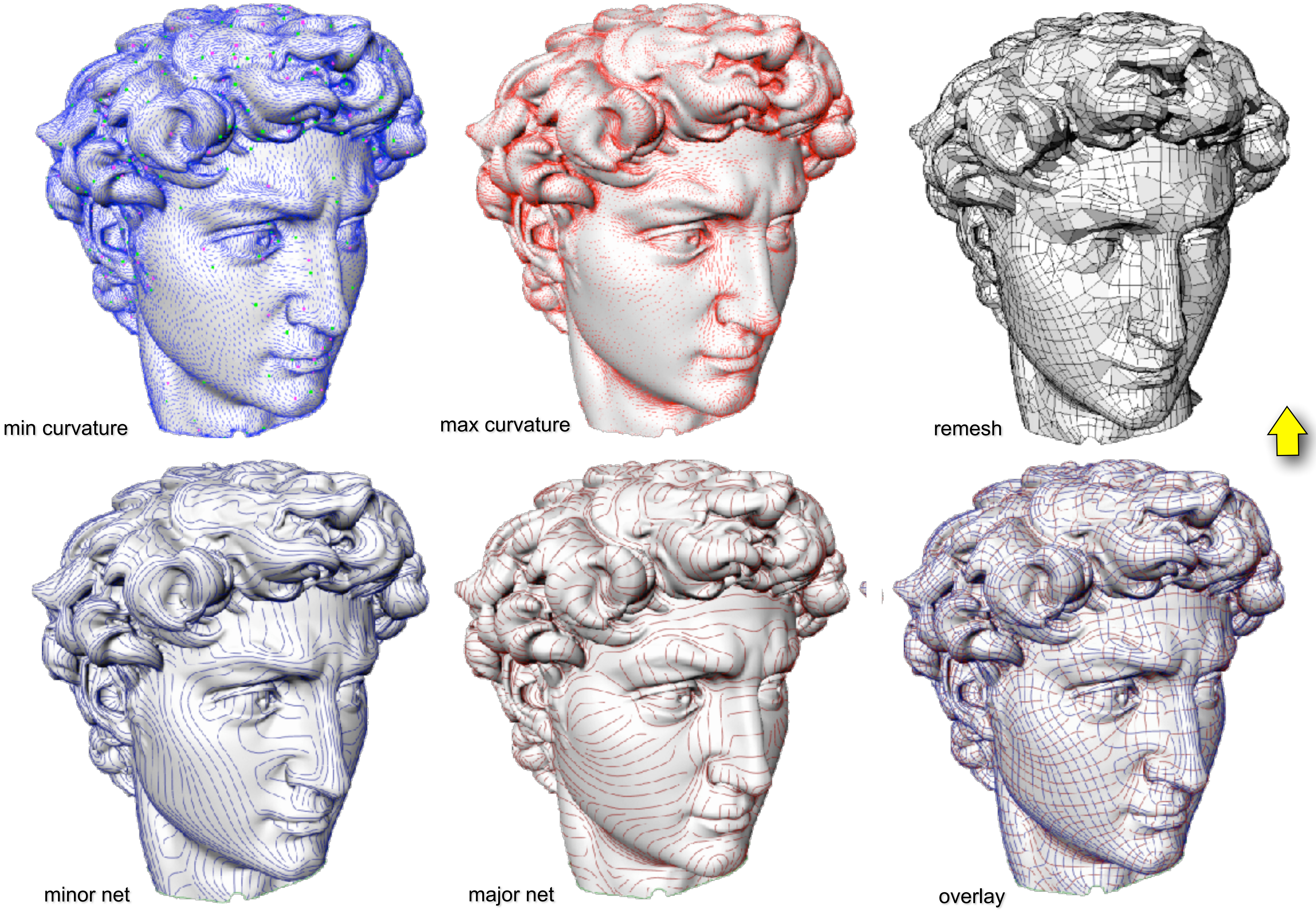






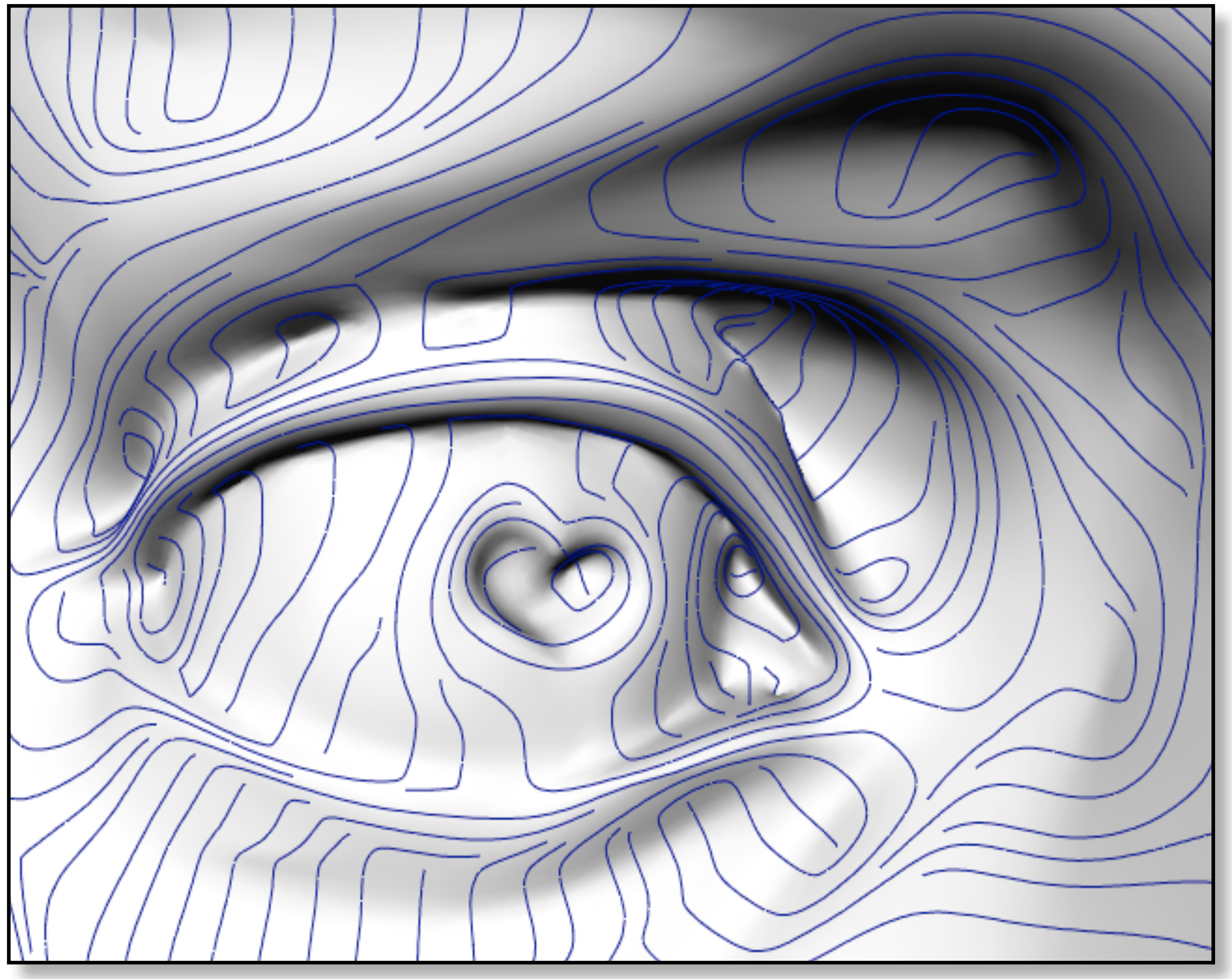
quad-triangle subdivision

Results

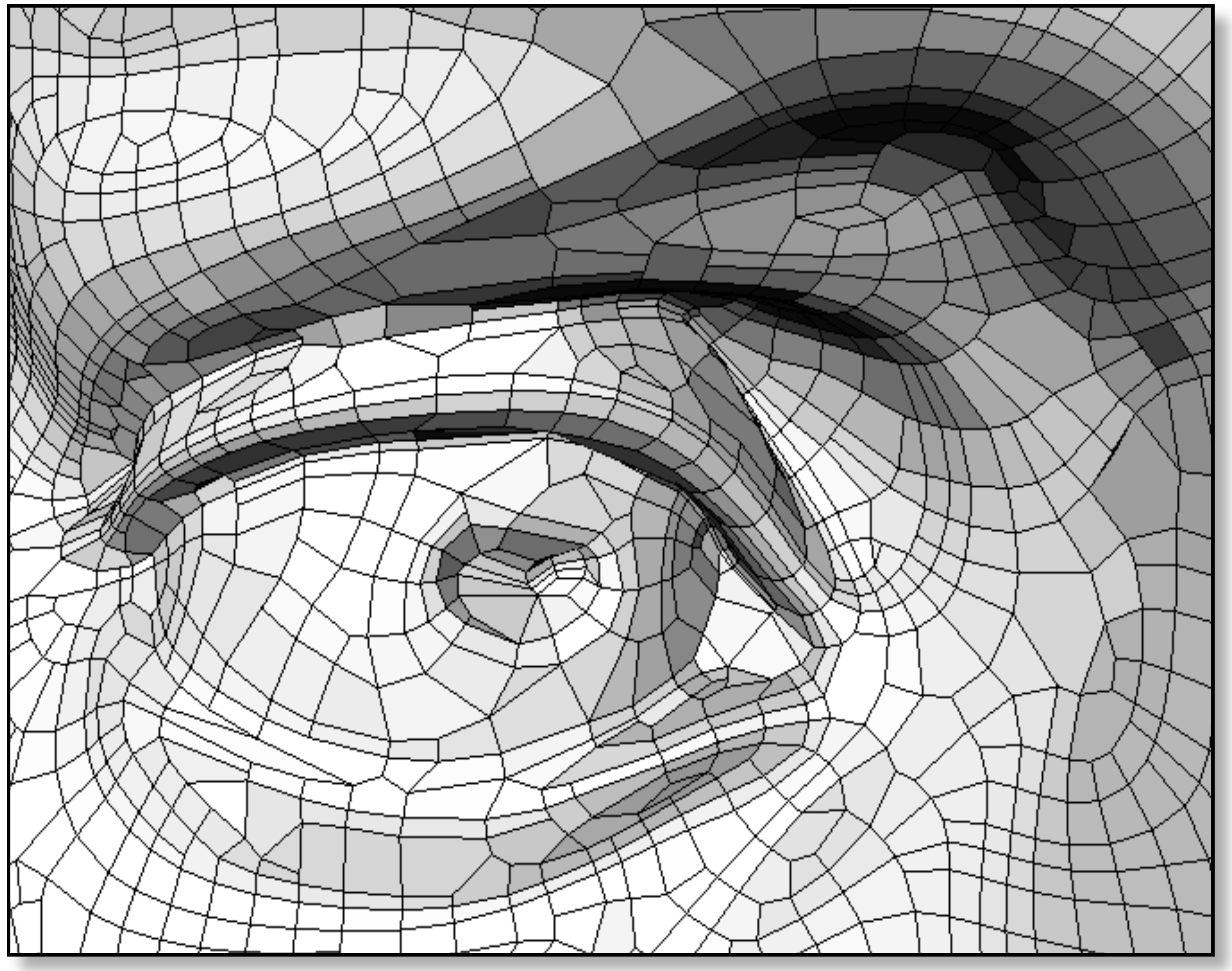


min curvature
max curvature
remesh
minor net
major net
overlay

Close-Up



minor net



remesh

Limitations

- **Global parameterization**
  - Smart cutting required
  - Low distortion parameterization
  - Numerical issues
- **Lines of curvature**
  - Robust curvature tensor computation
  - Tensor smoothing required
  - Optimal placement of streamlines difficult

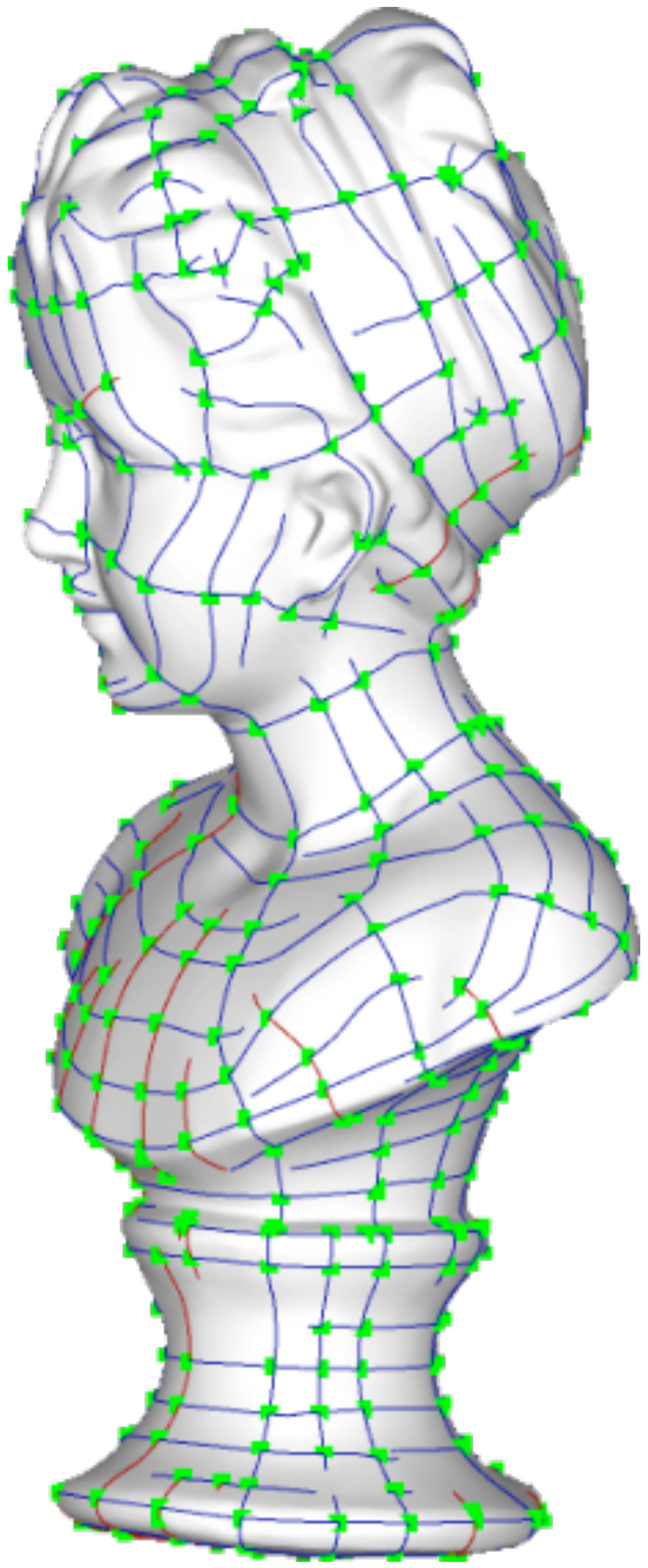
Overview

- Isotropic Remeshing
  - Parameterization-based [Alliez 2002]
  - Surface-based [Botsch 2004]

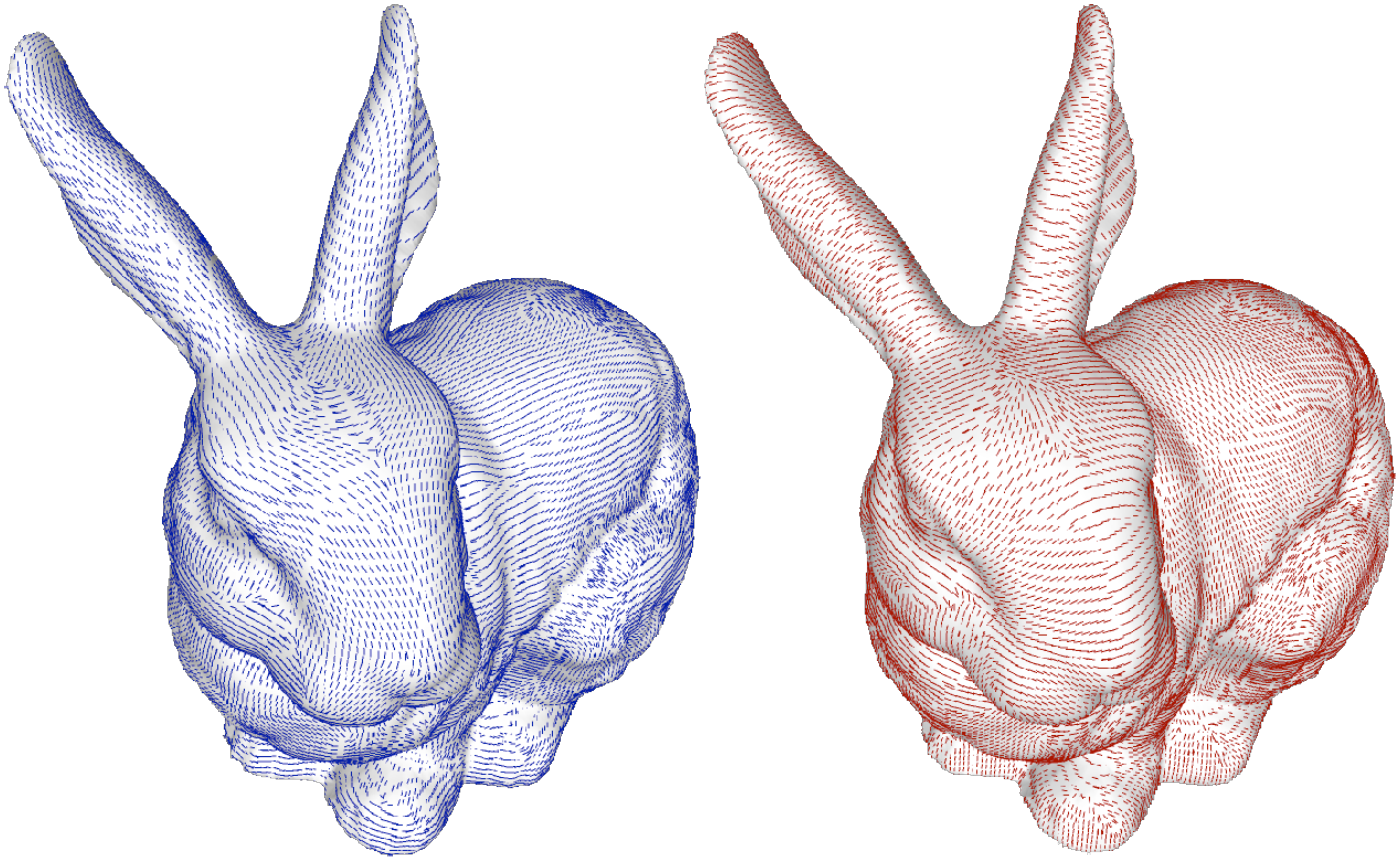
- Anisotropic Remeshing
  - Parameterization-based [Alliez 2003]
  - **Surface-based** **[Marinov 2004]**

Marinov: Direct Aniso. Remeshing

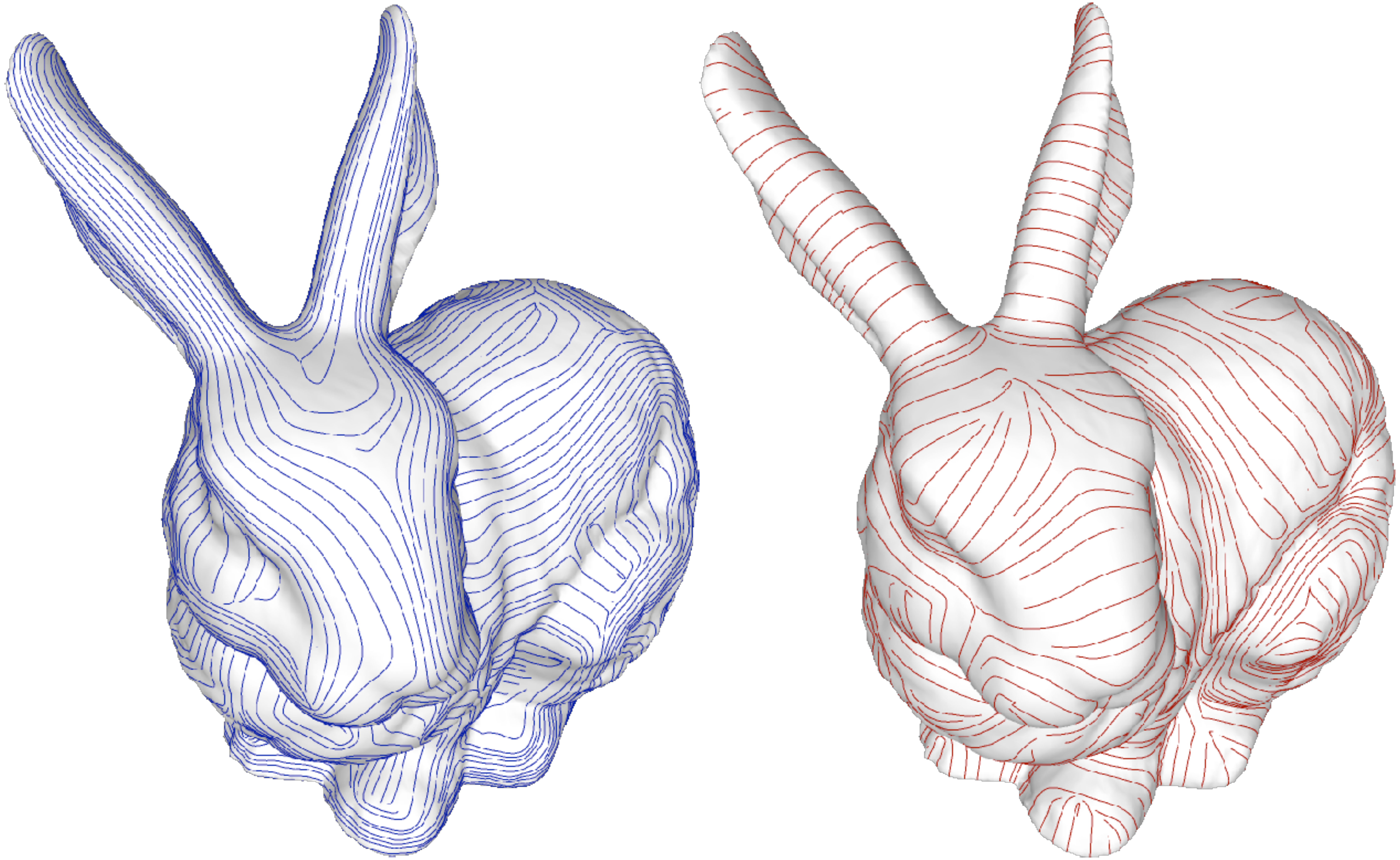
- Compute **curvature tensor** on the mesh
  - Smoothing in the tangent plane
- Integrate **curvature lines** on the mesh
  - Follow min / max curvature in aniso. regions
  - Straight geodesics in isotropic regions
  - Line snapping
- **Mesh directly** on the mesh
  - Lines intersections give vertices
  - Convex partitioning



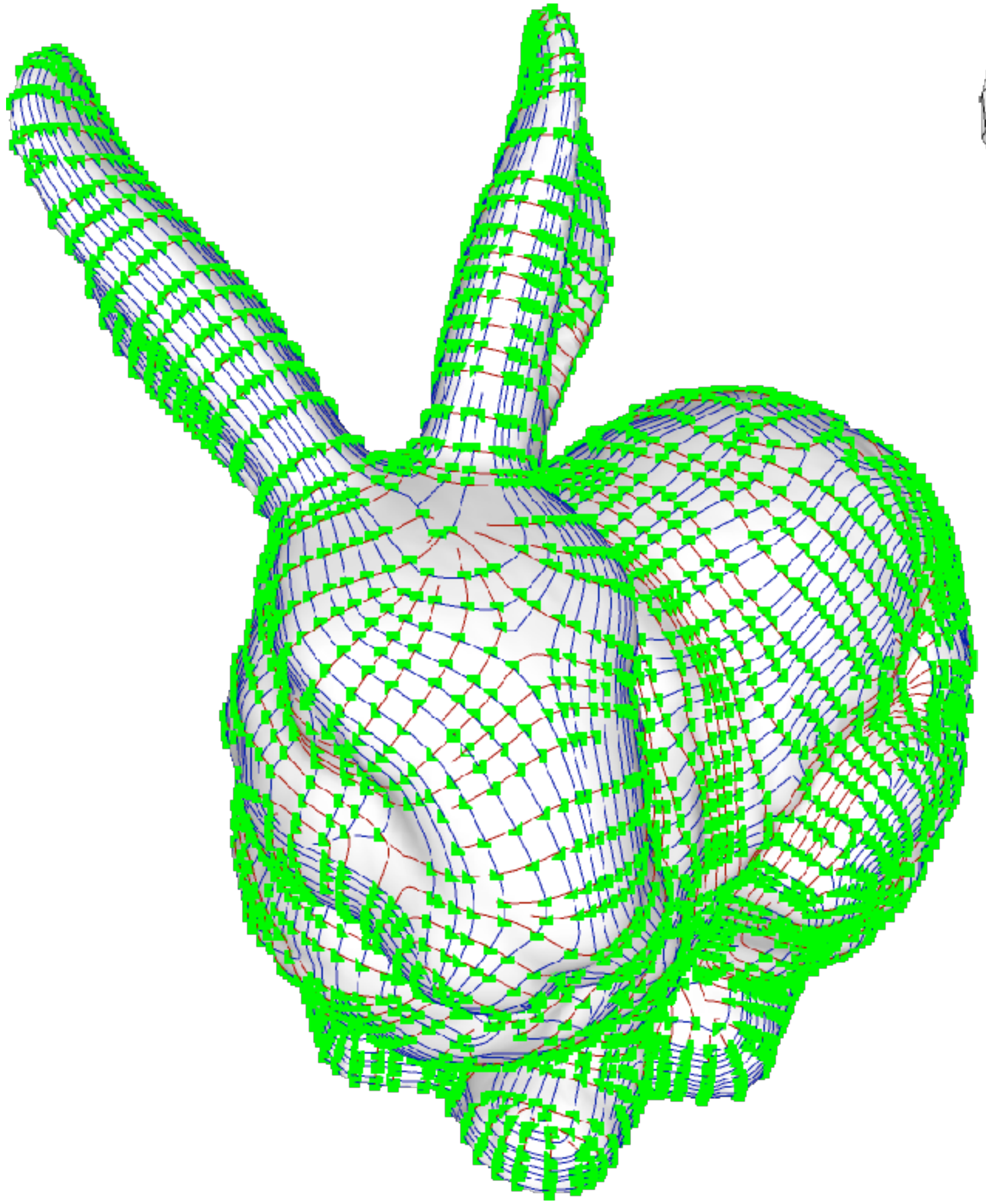
Curvature Tensors

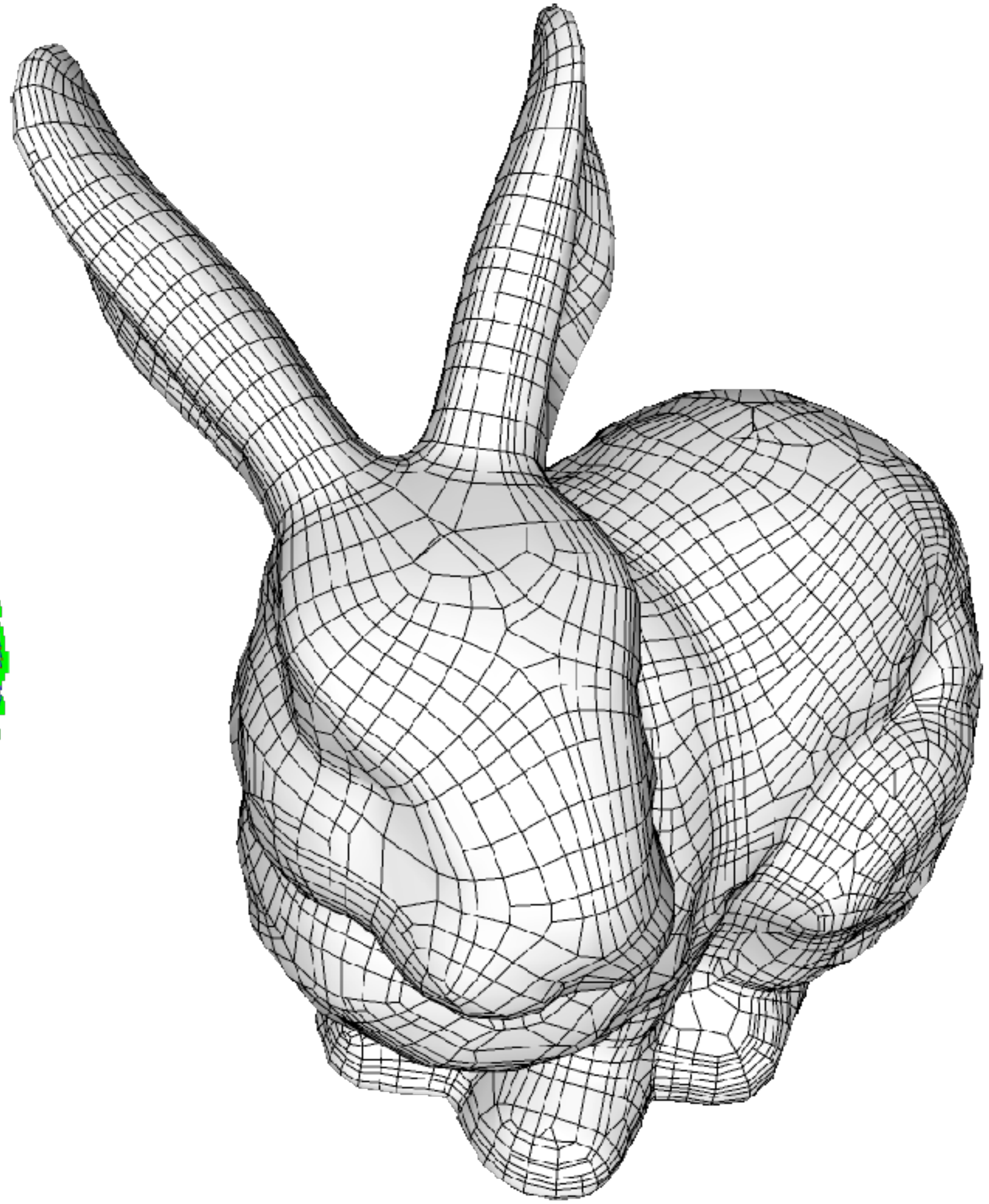


Curvature Line Integration

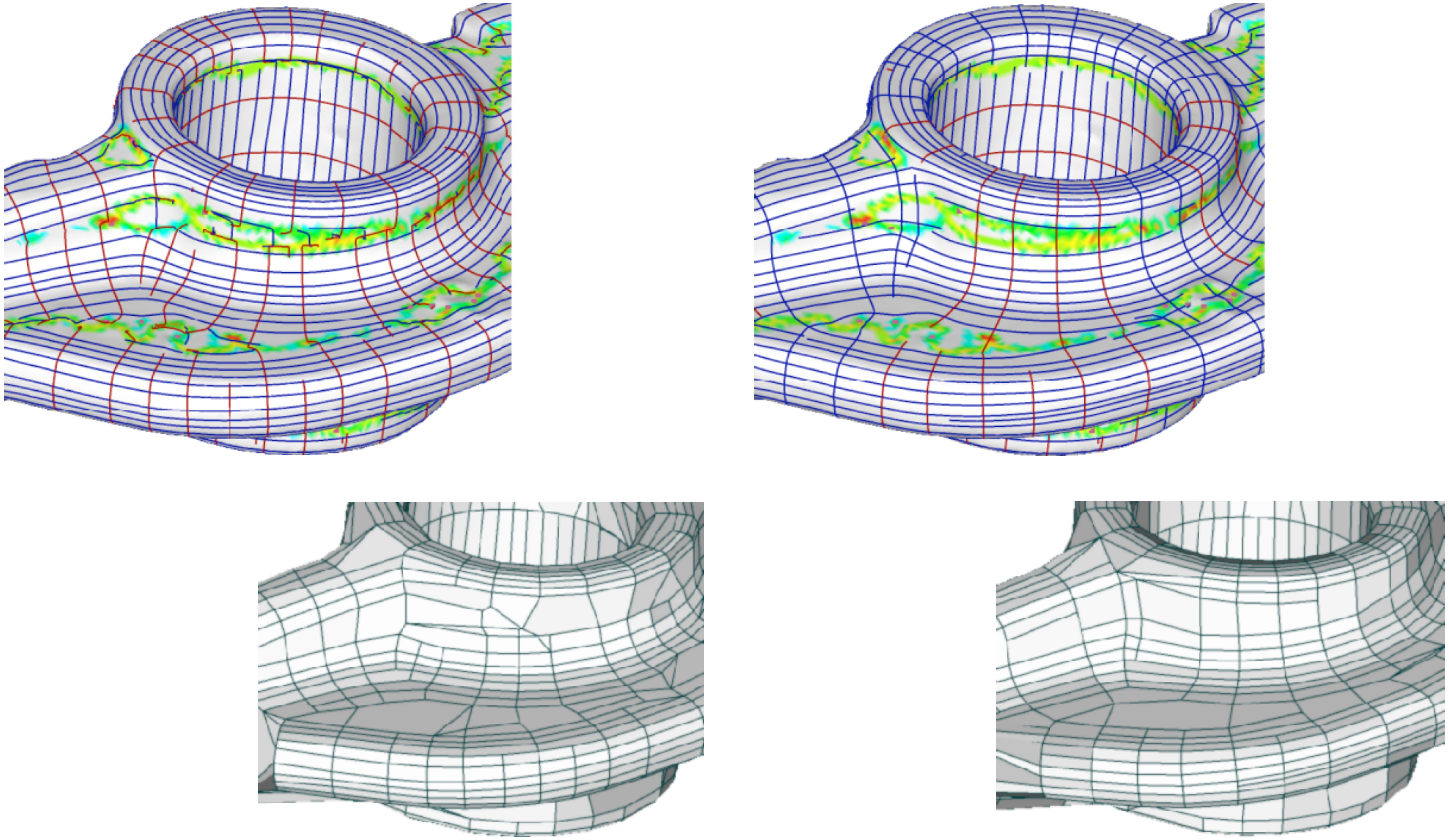


Meshing

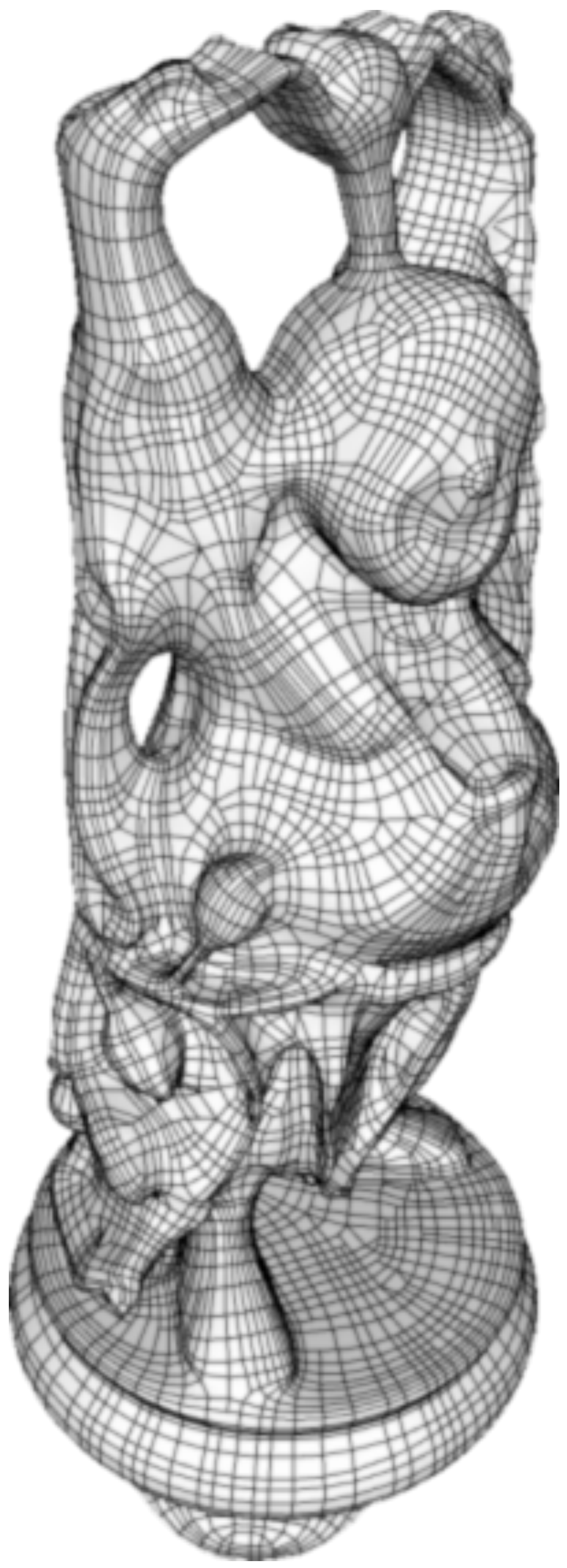


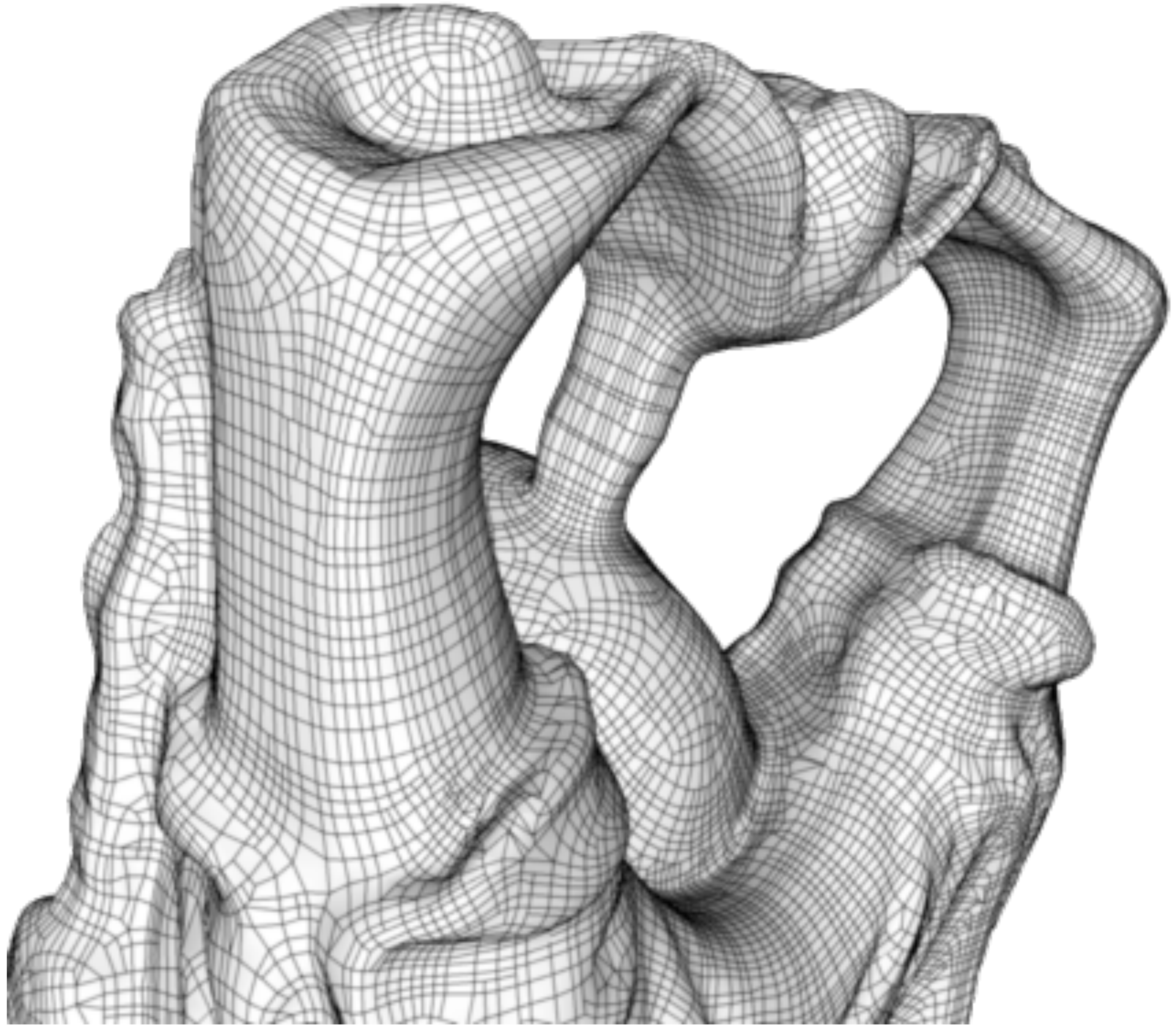


Confidence Estimation



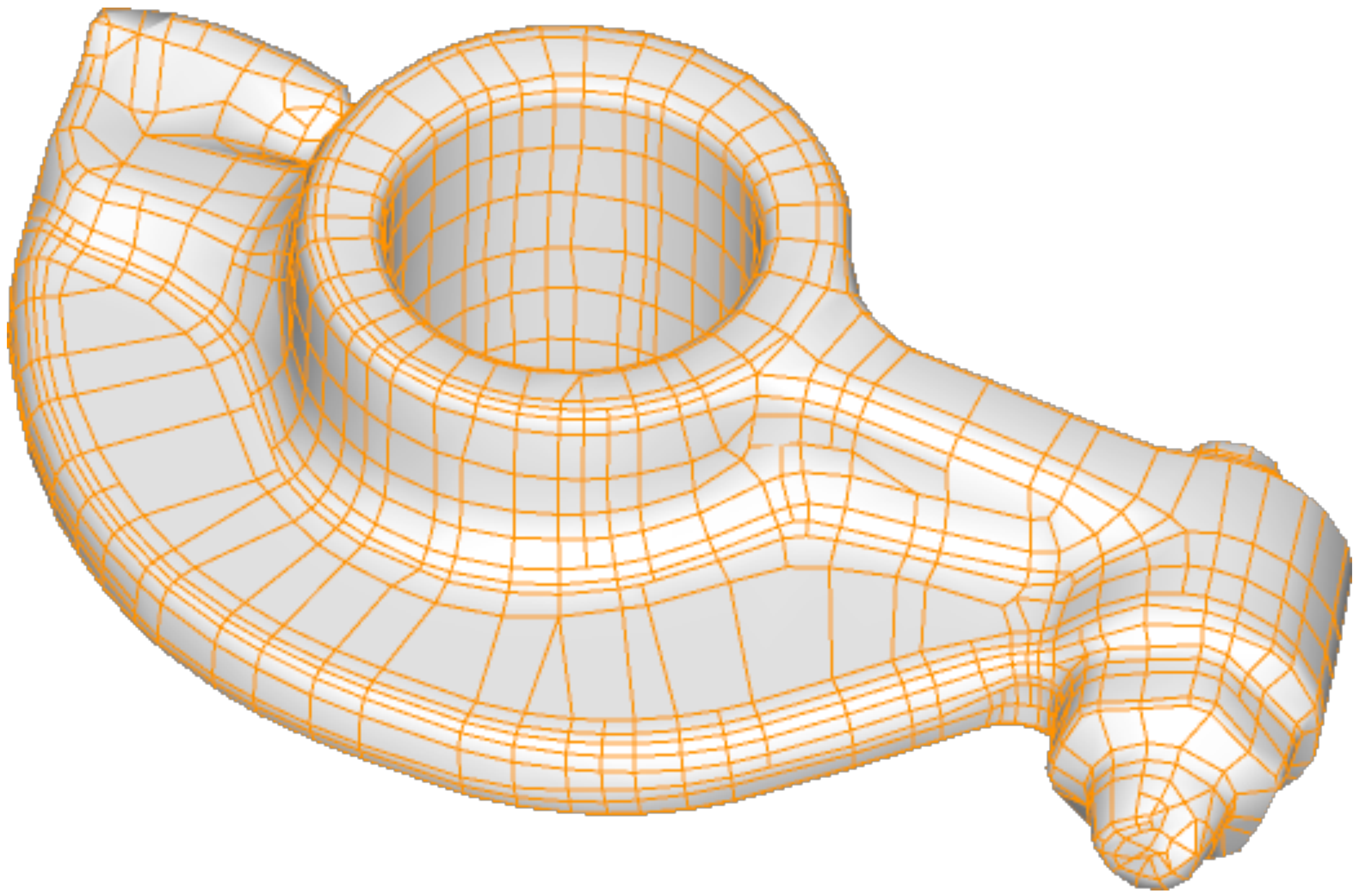
Quad Dominant Meshing





CAD Model Remeshing

- Direction propagation into isotropic regions
  - Confidence estimation
  - Line snapping



Two Fundamental Approaches

- Surface oriented
  - operate directly on the surface
  - treat surface as a set of points / polygons in space
  - efficient for high resolution remeshing

- Parametrization based
  - map to 2D domain / 2D problem
  - computationally more expensive (?)
  - works even for coarse resolution remeshing

Basic Operators

- Correspondence
  - Global vs. local parameterization
- Vertex density control
  - Uniform vs. adaptive
  - Isotropic vs. anisotropic
- Local alignment
  - Optimal shape approximation
- Global alignment
  - Feature sensitivity

References

- Alliez et al., *"Interactive geometry remeshing,"* SIGGRAPH 2002.
- Alliez et al., *"Isotropic surface remeshing,"* SMI 2003.
- Alliez et al., *"Anisotropic polygonal remeshing,"* SIGGRAPH 2003.
- Vorsatz et al, *"Dynamic remeshing and applications,"* Solid Modeling 2003.
- Botsch & Kobbelt, *"A remeshing approach to multiresolution modeling,"* Symp. on Geometry Processing 2004.
- Marinov et al., *"Direct anisotropic quad-dominant remeshing,"* Pacific Graphics 2004.
- **Alliez et al., *"Recent advances in remeshing of surfaces,"* AIM@Shape state of the art report, 2006.**