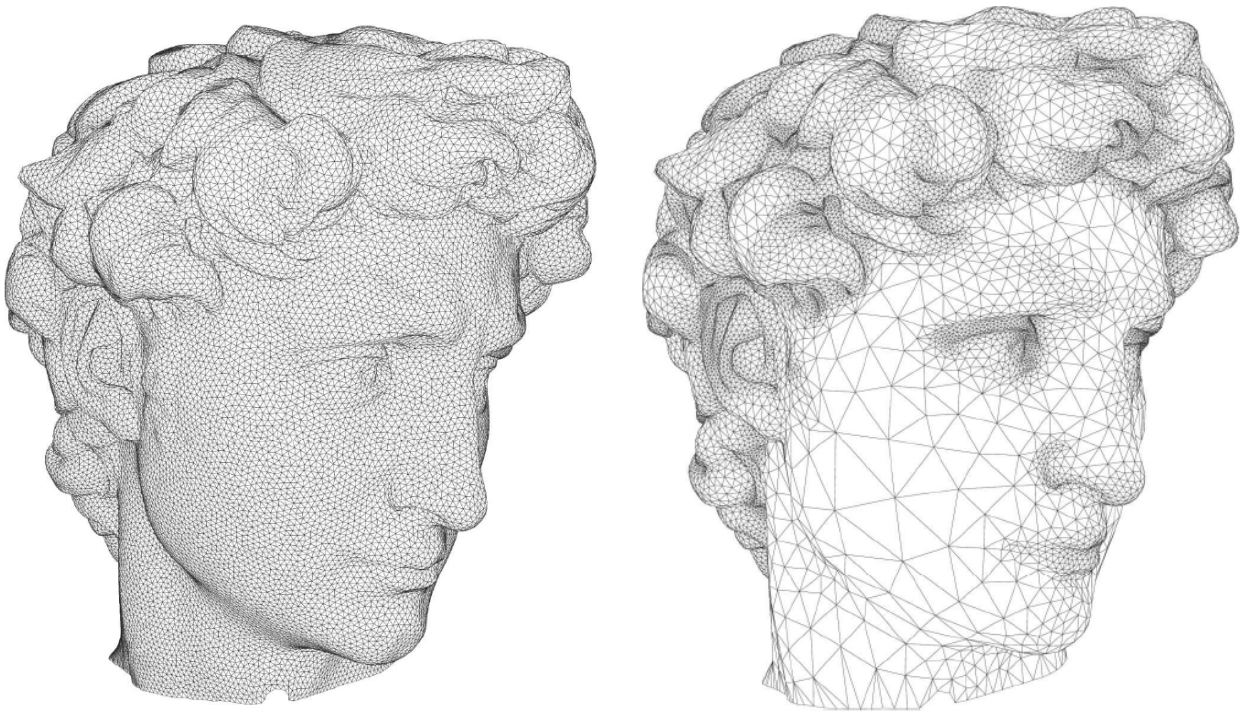
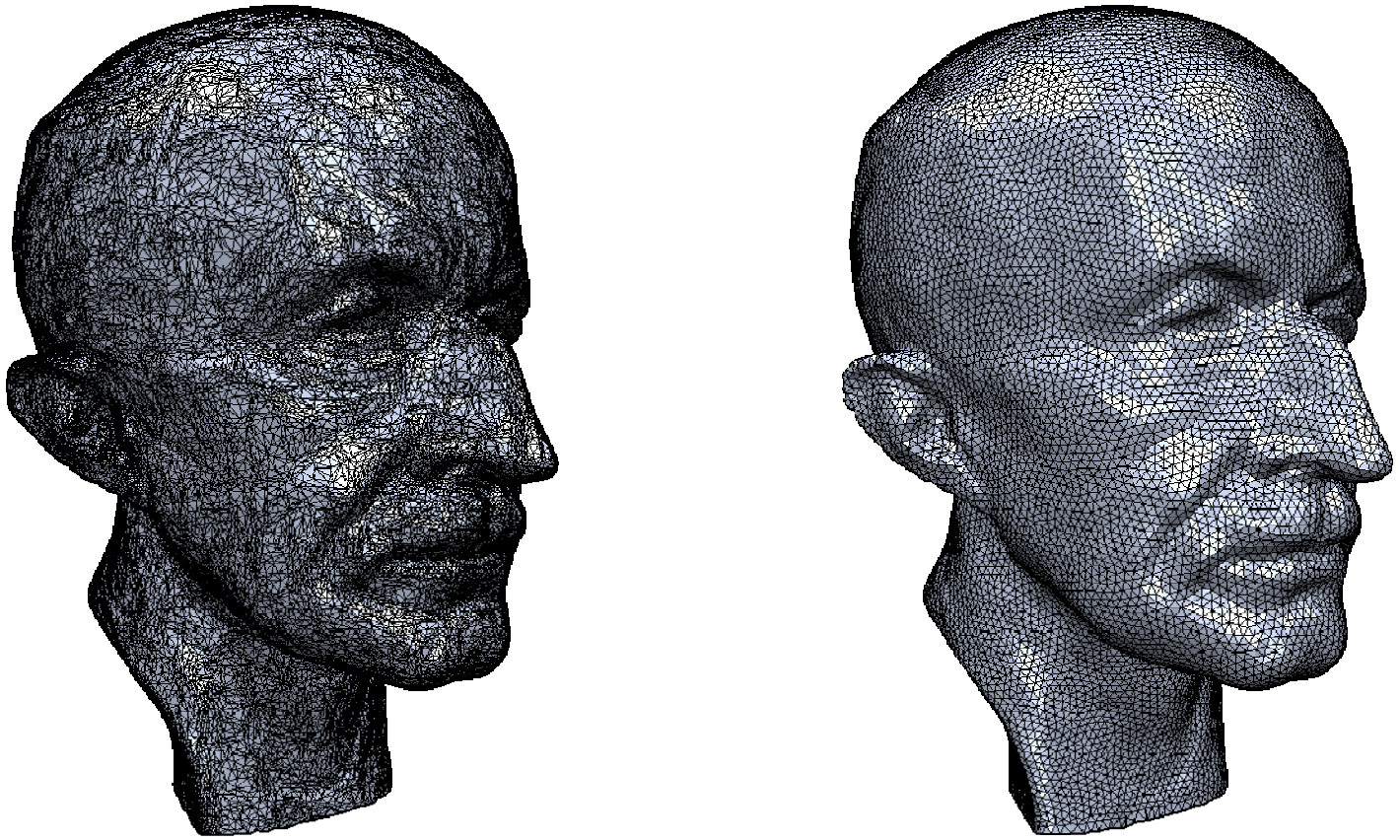
Remeshing I



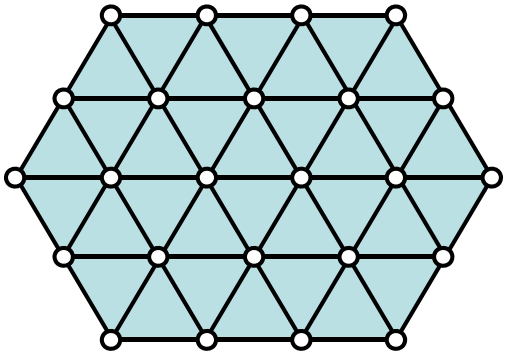
Remeshing

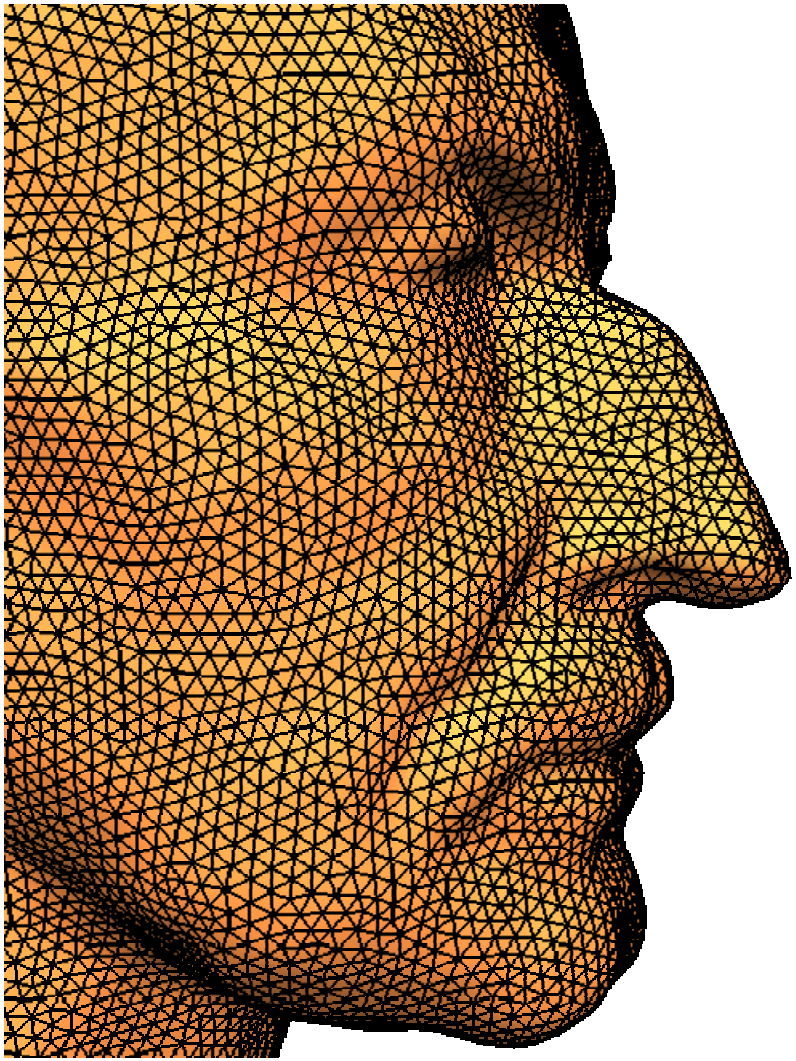
Given a 3D mesh, find a “better” discrete representation of the underlying surface



What is a good mesh?

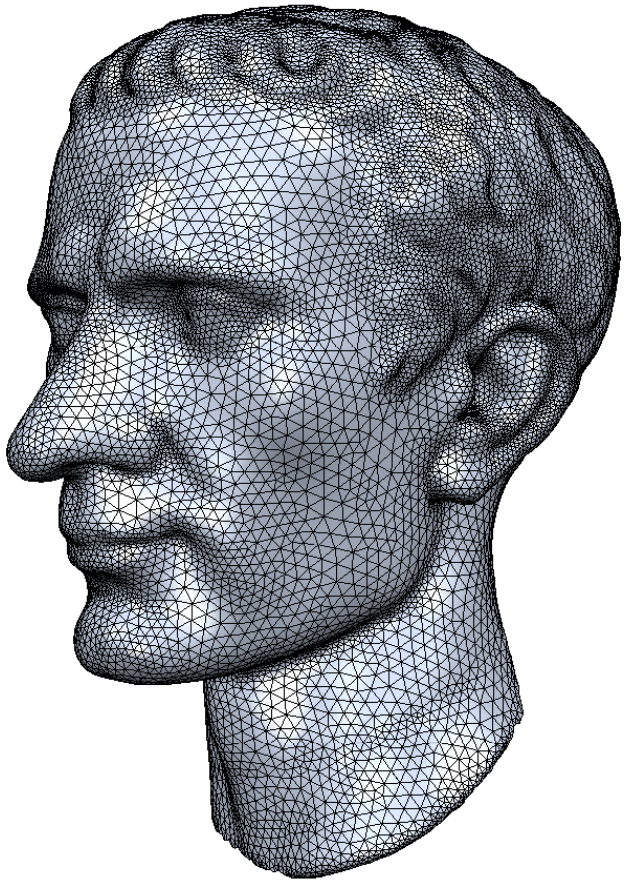
- Equal edge lengths
- Equilateral triangles
- Valence close to 6





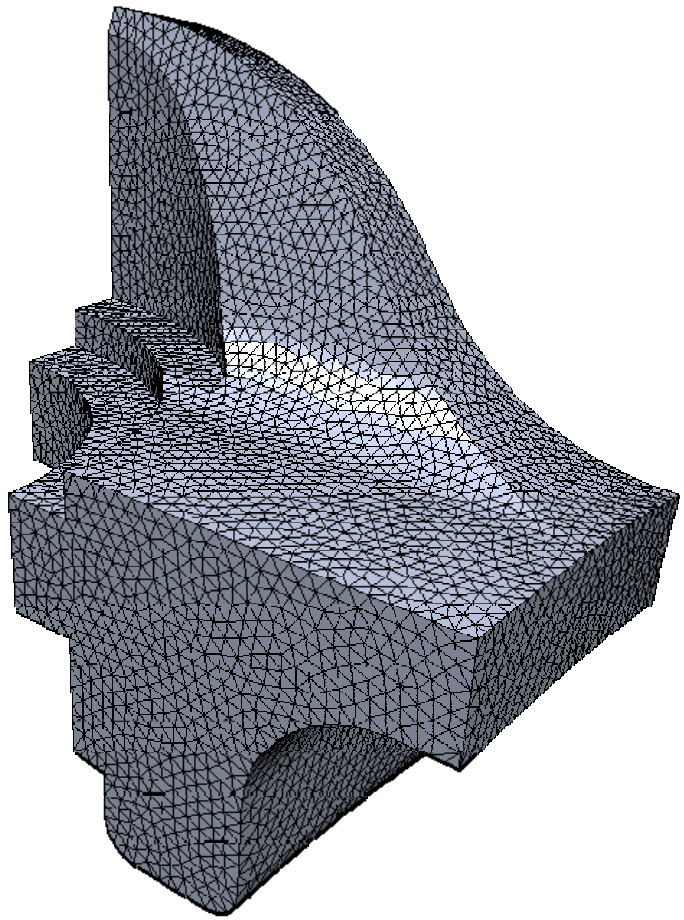
What is a good mesh?

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



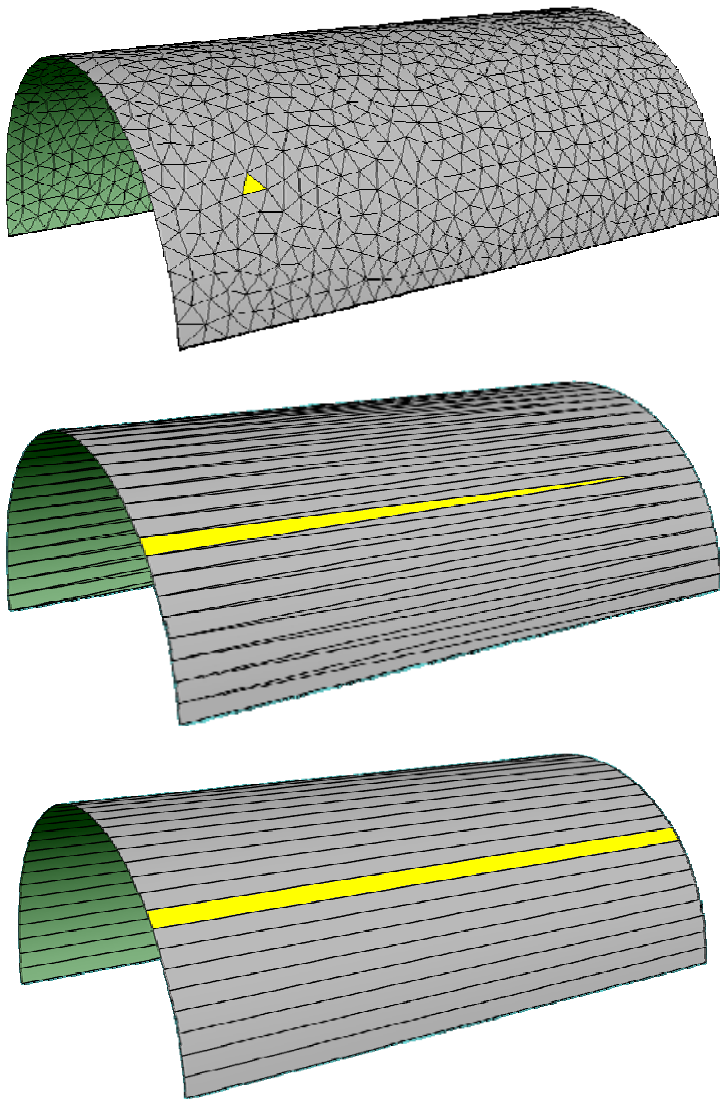
What is a good mesh?

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



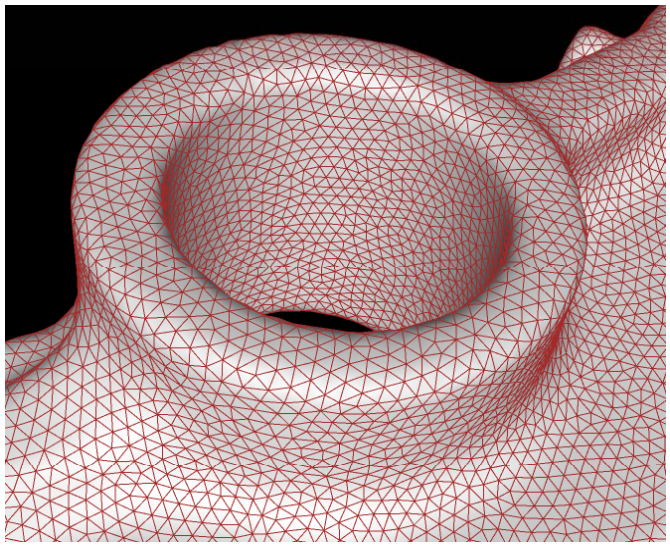
What is a good mesh?

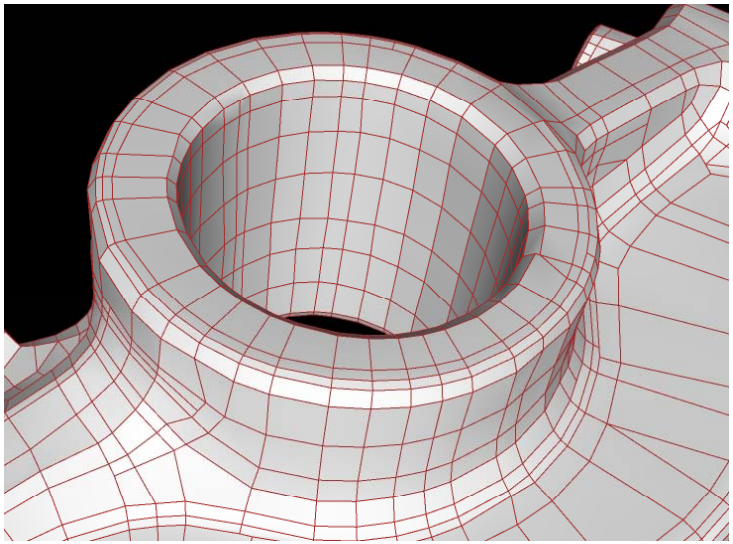
- Equal edge lengths
- Equilateral triangles
- Valence close to 6
- Uniform vs. adaptive sampling
- Feature preservation
- Alignment to curvature lines
- Isotropic vs. anisotropic



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



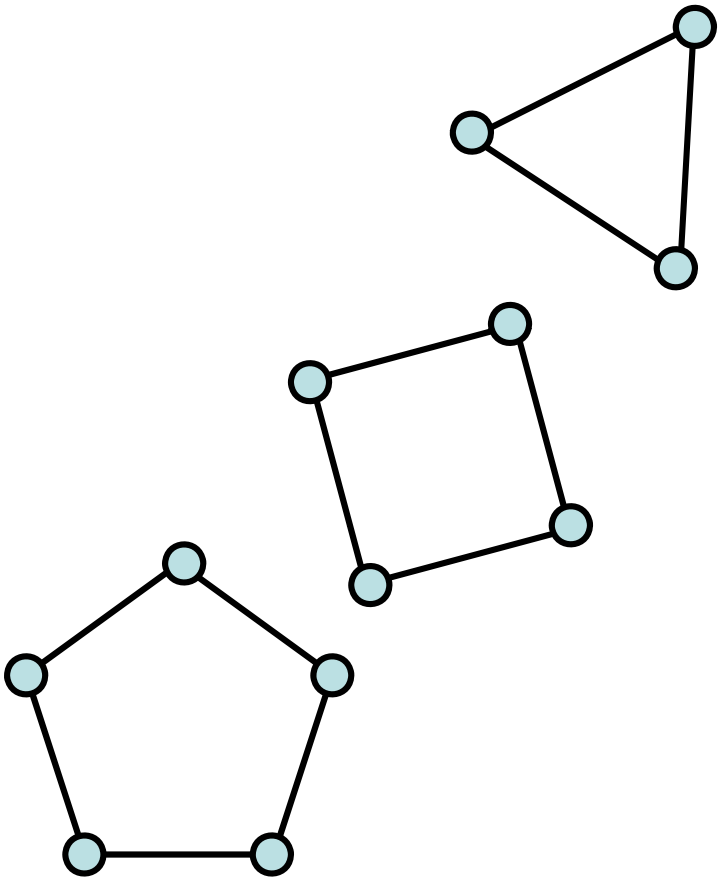


Applications

- Design
- Reverse engineering
- Simulation
- Visualization

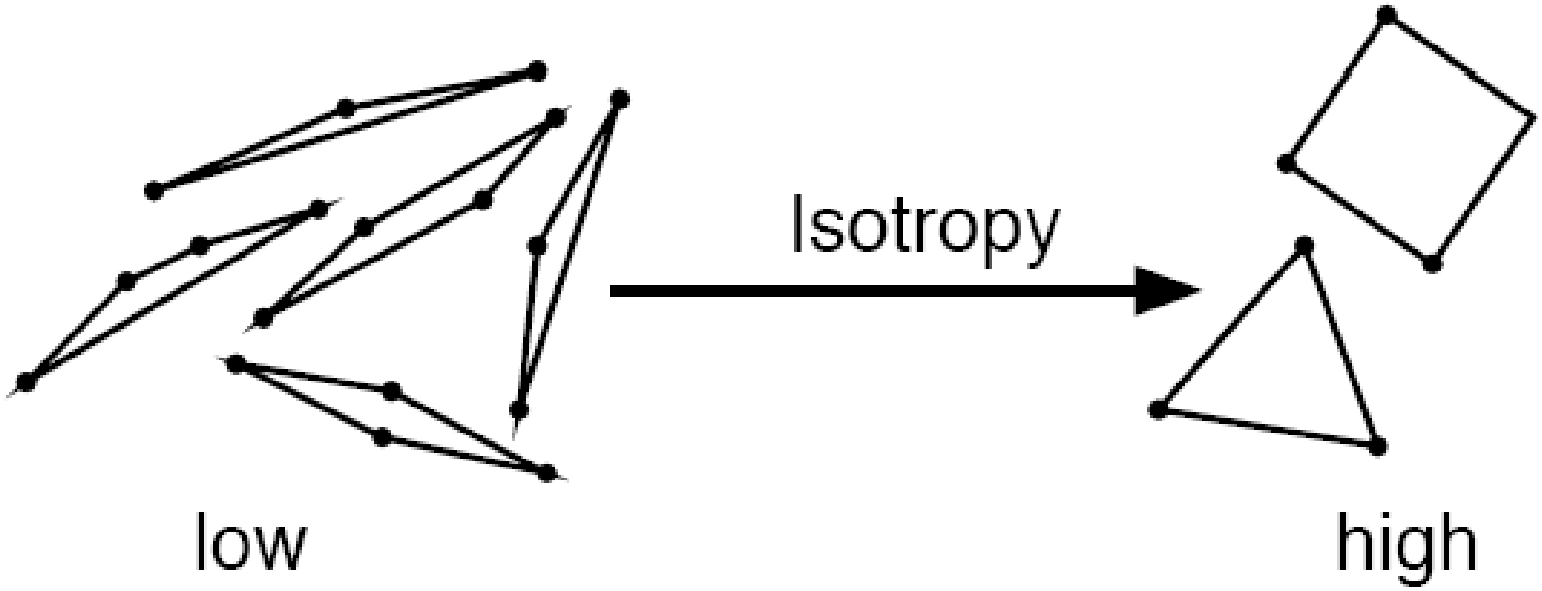
Local Structure

- Element **type**
  - Triangle
  - Quadrangle
  - Polygon



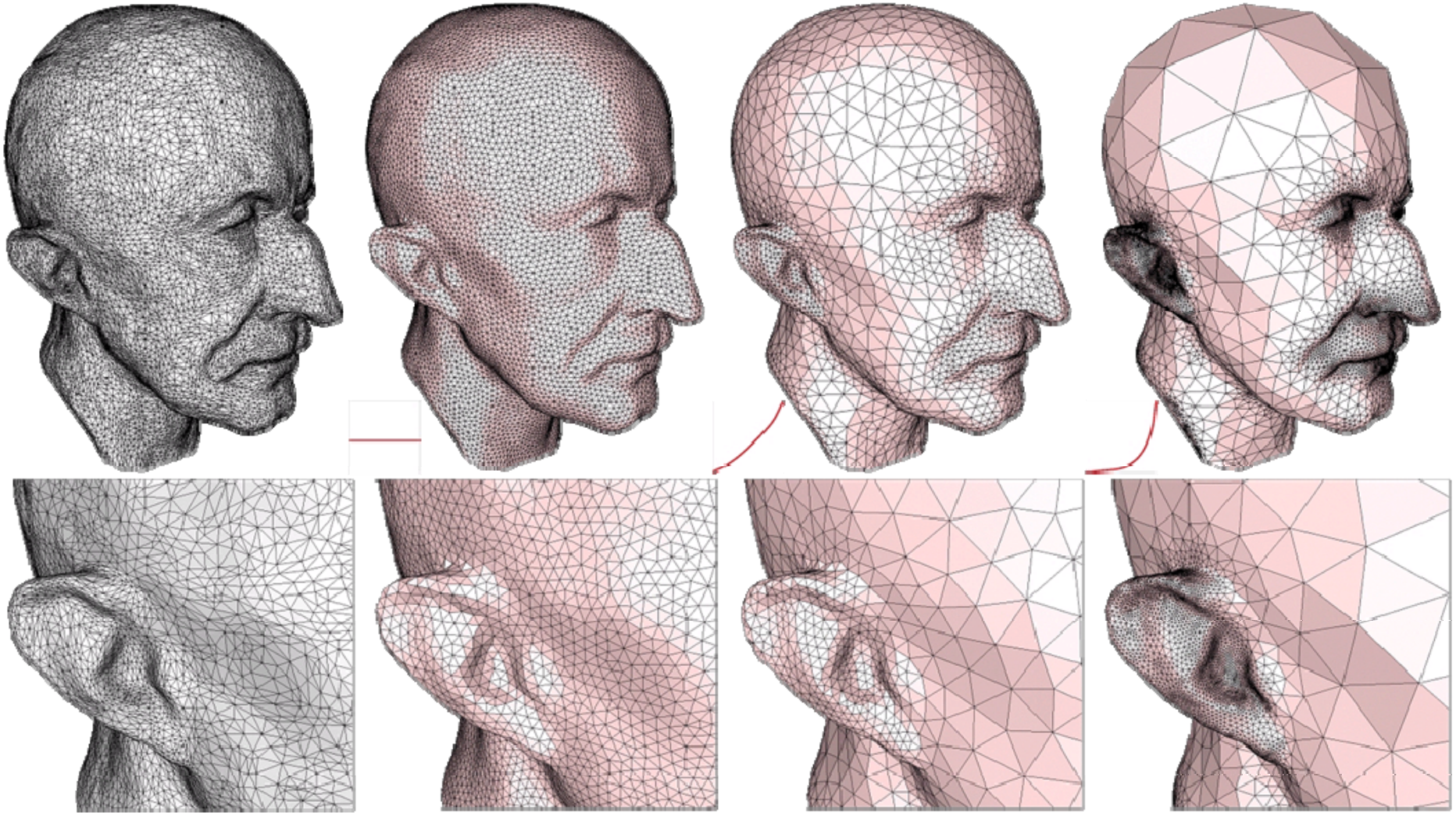
Local Structure

- Element **shape** (isotropy vs anisotropy)



Local Structure

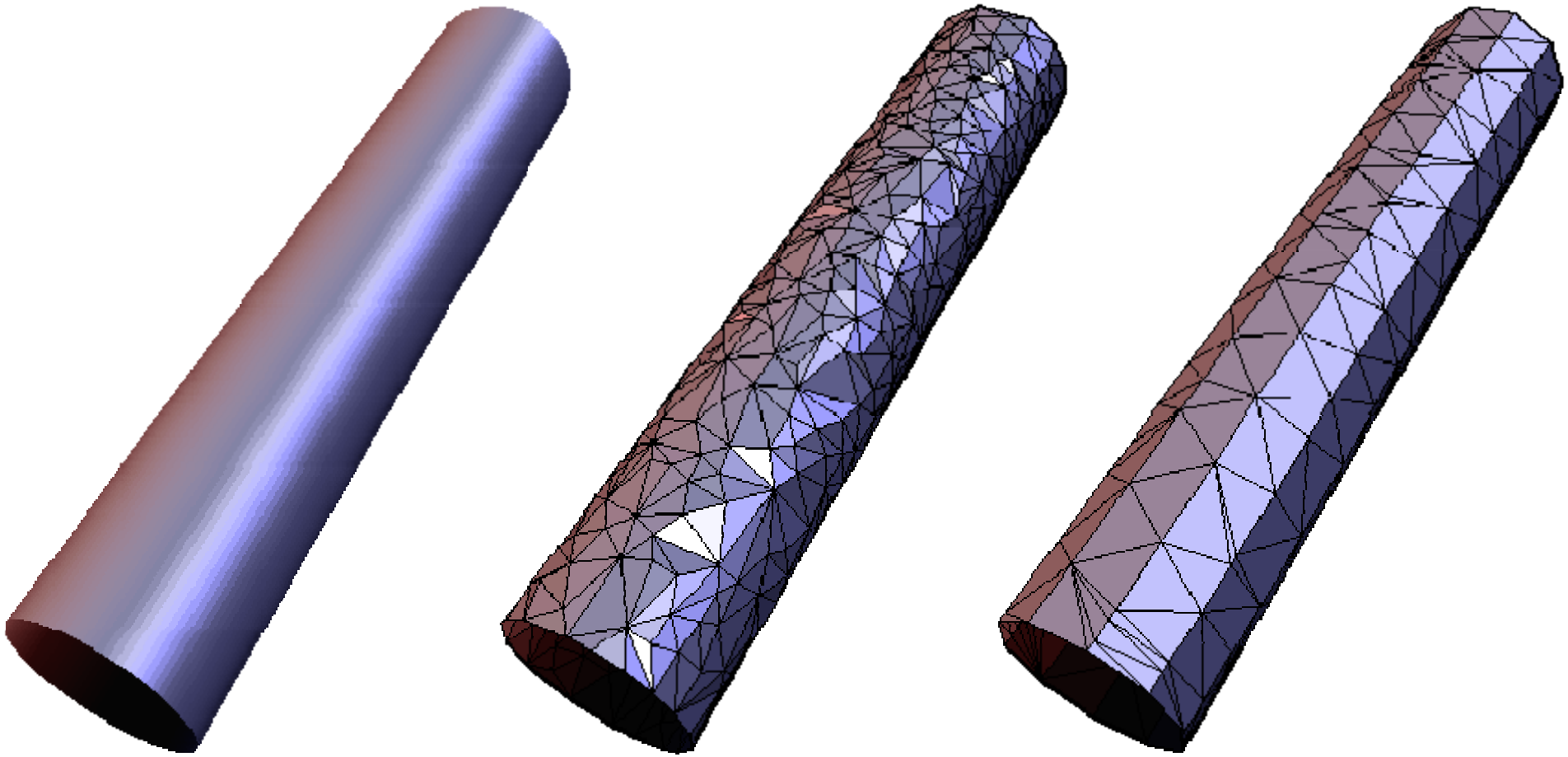
Element distribution (**sizing**, **grading**)



Input Uniform Adaptive

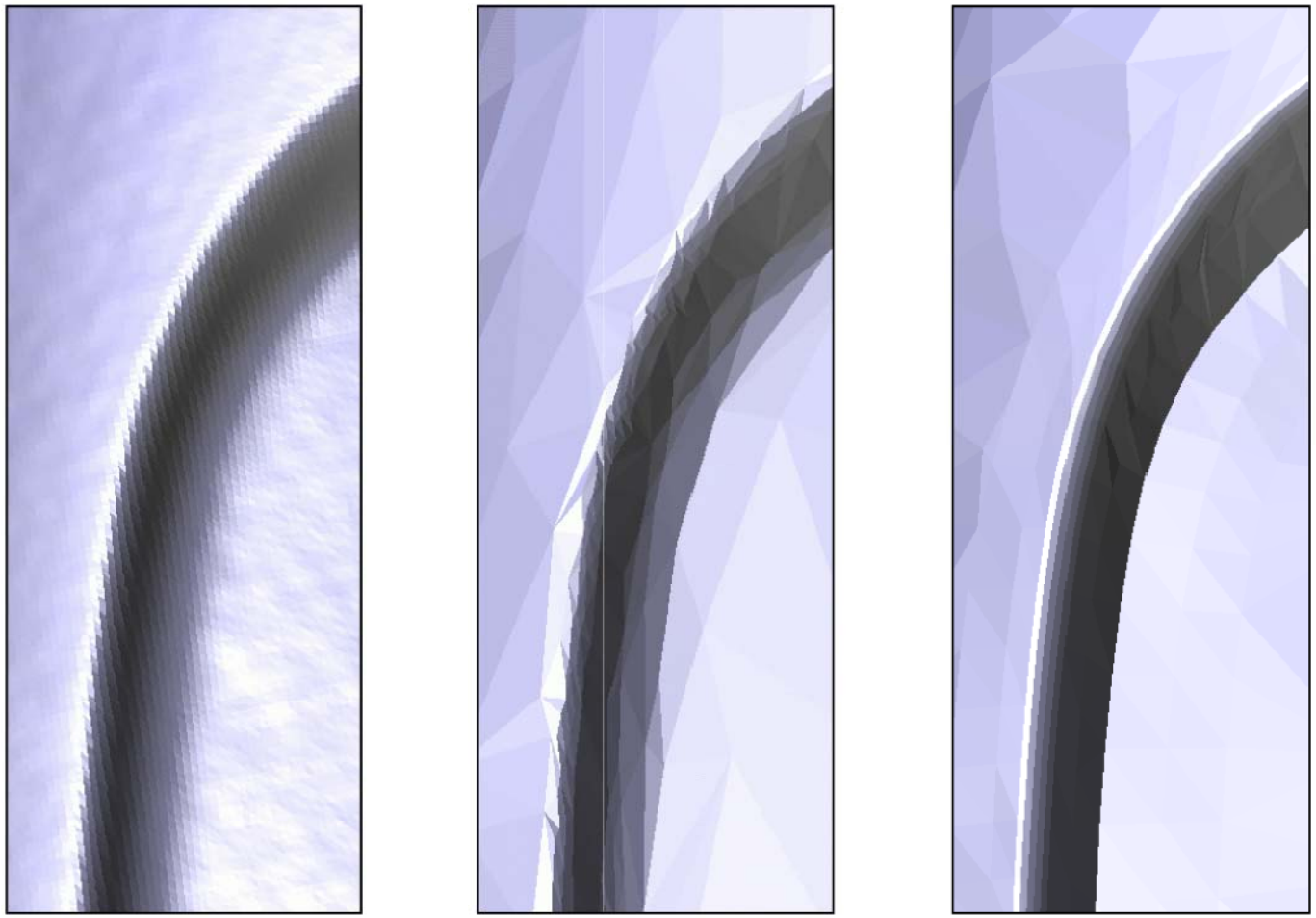
Local Structure

- Element **orientation**



Local Structure

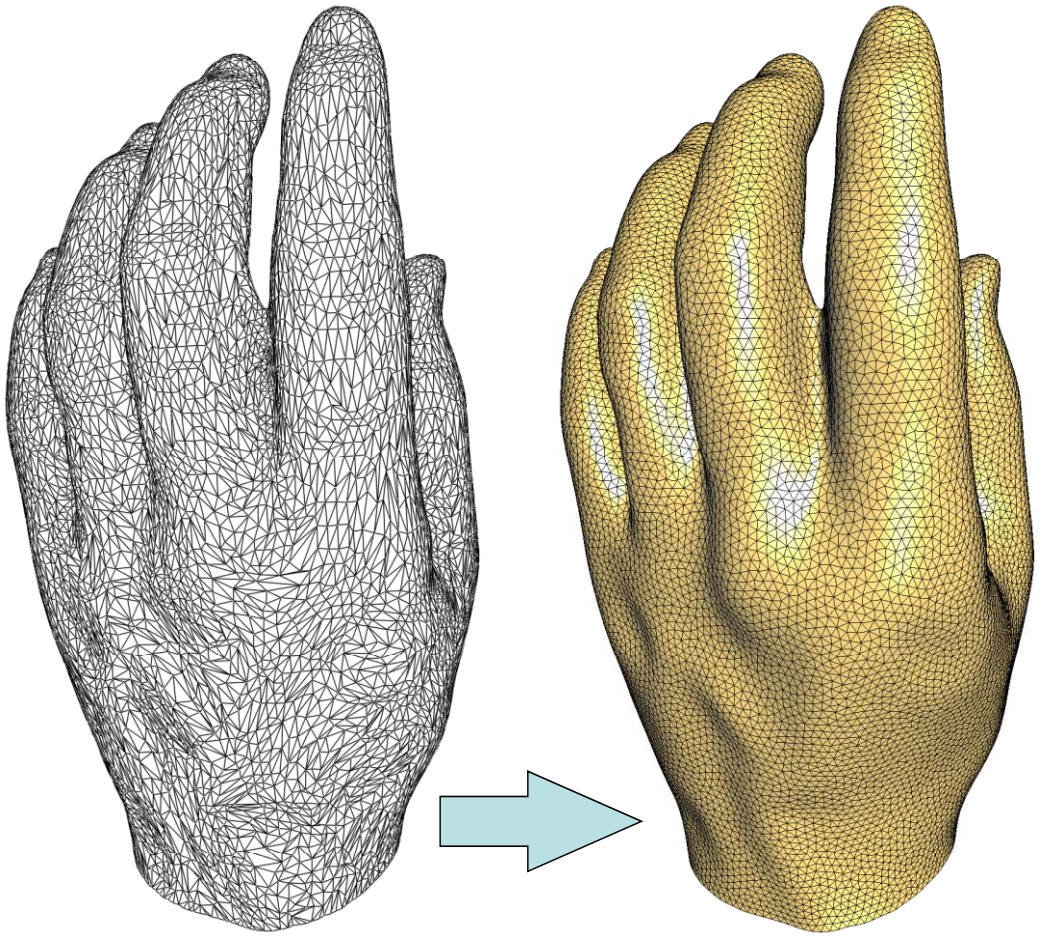
- Element orientation



Isotropic Remeshing

Well-shaped elements

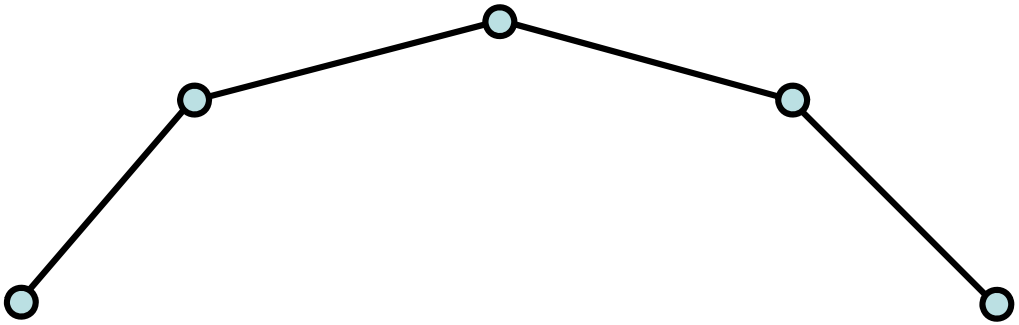
– for processing & simulation (numerical stability & efficiency)



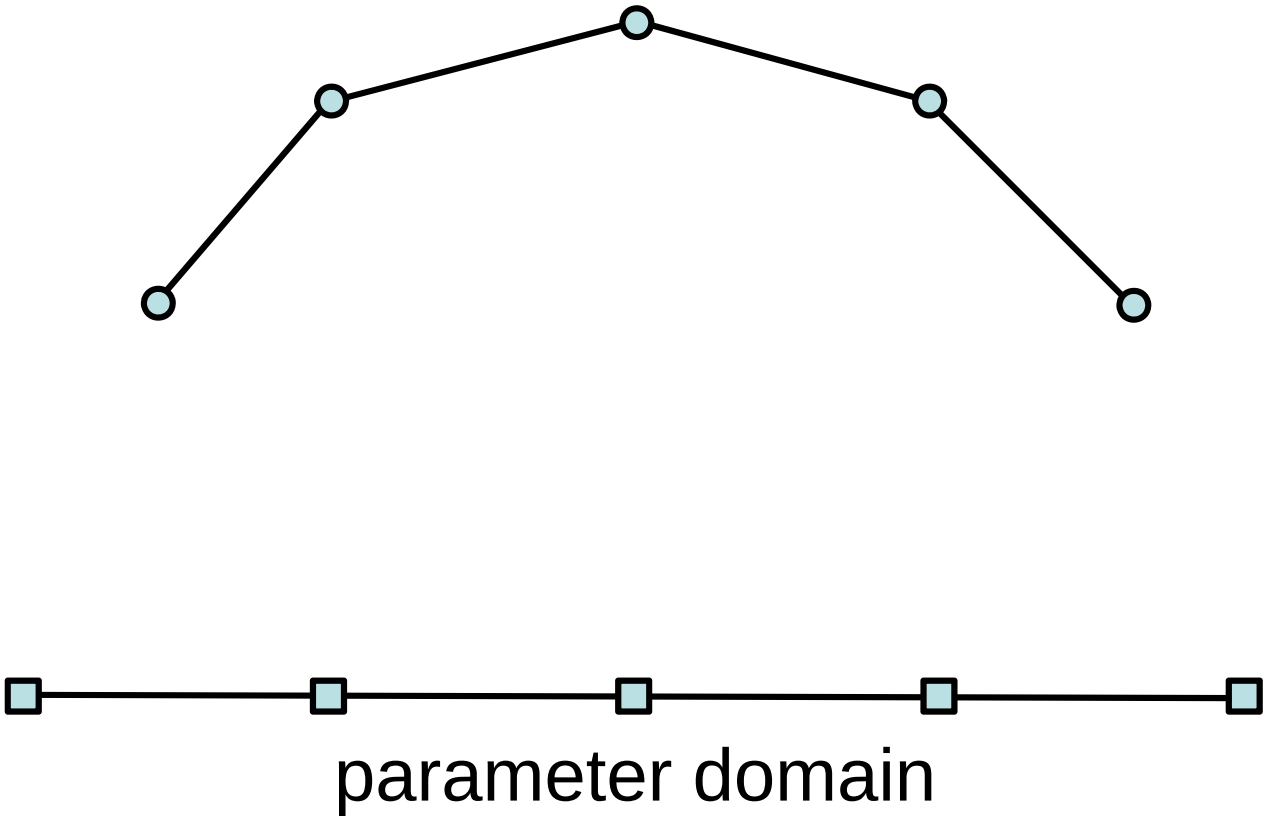
Two Fundamental Approaches

- **Parameterization-based**
  - map to 2D domain / 2D problem
  - computationally more expensive
  - works even for coarse resolution remeshing
- **Surface-oriented**
  - operate directly on the surface
  - treat surface as a set of points / polygons in space
  - efficient for high resolution remeshing

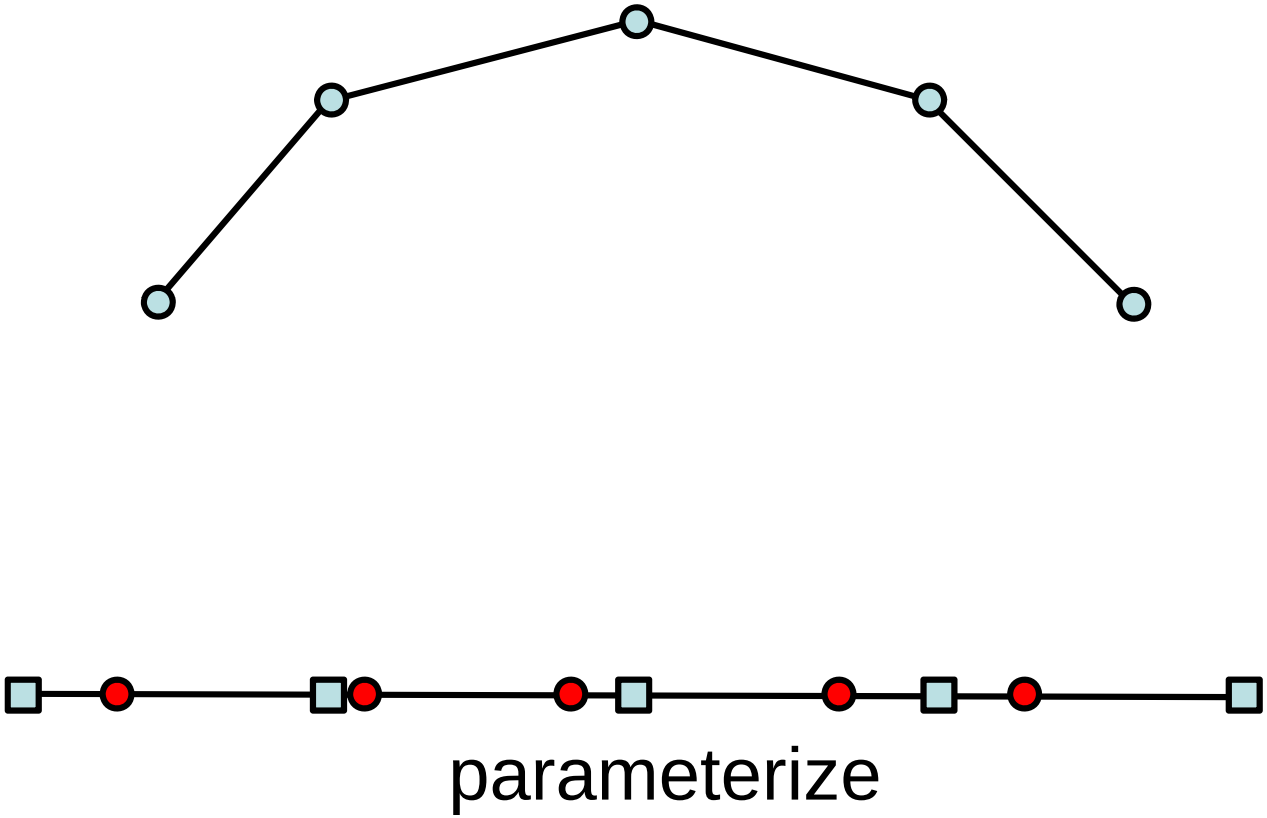
Parameterization Based



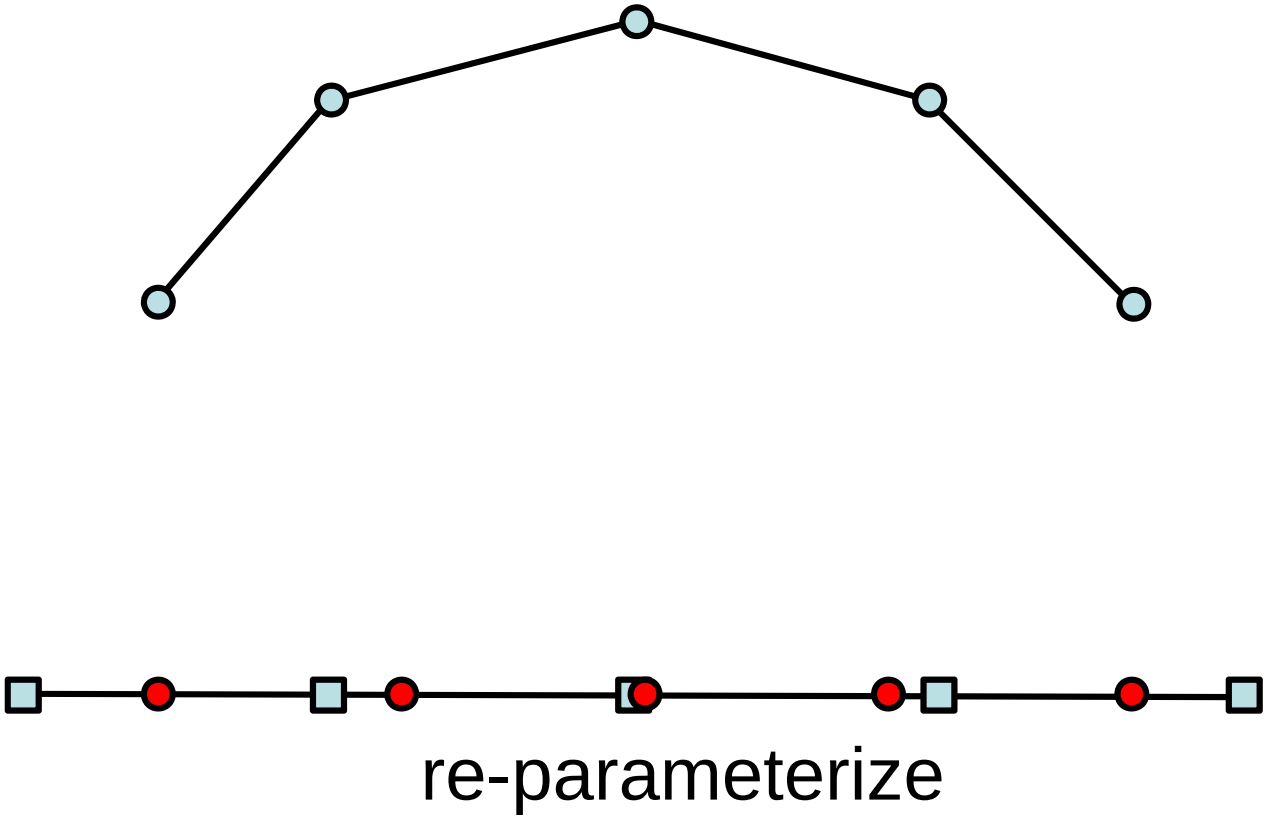
Parameterization Based



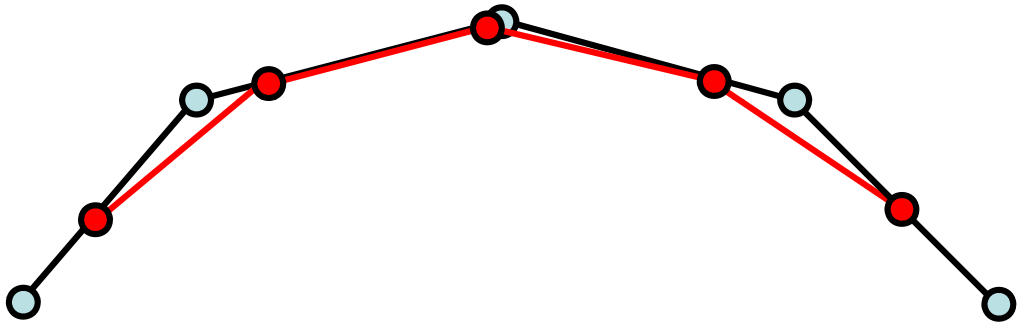
Parameterization Based



Parameterization Based

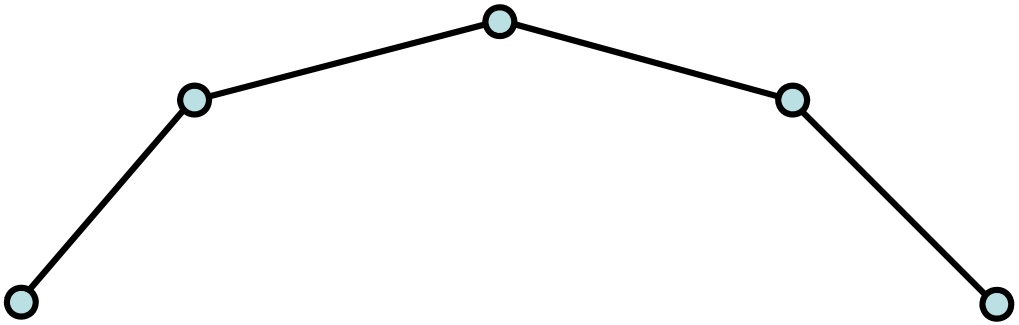


Parameterization Based

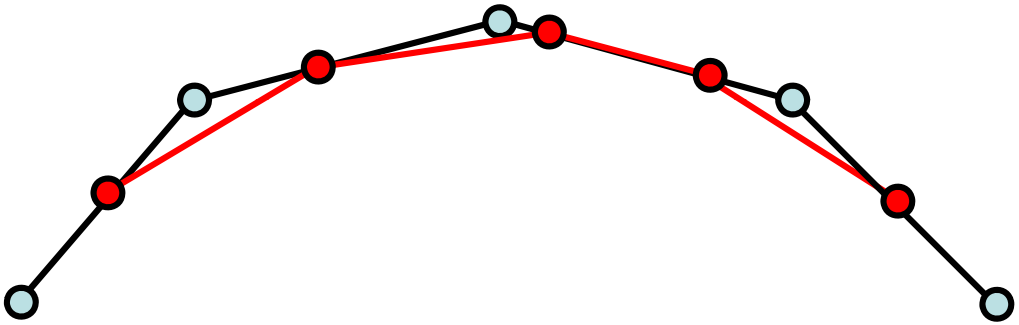


apply

Surface Oriented

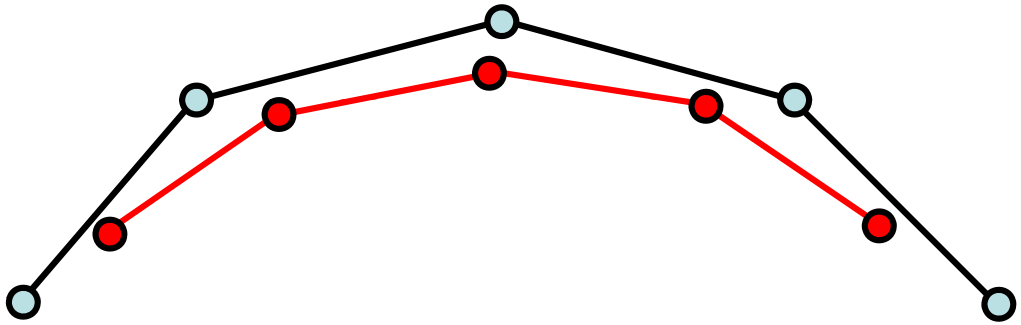


Surface Oriented



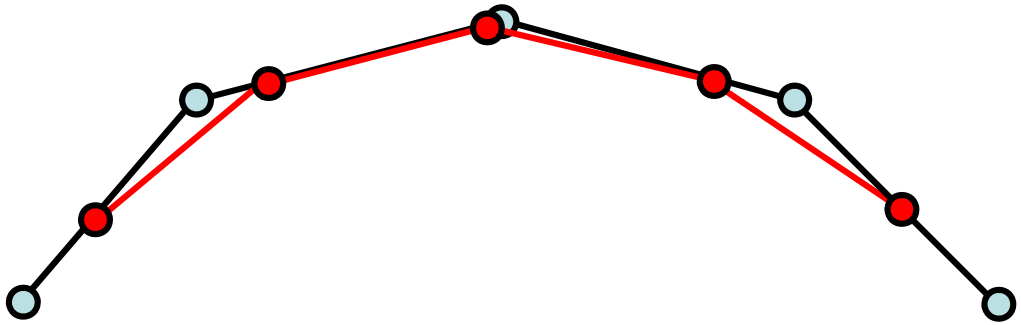
sample

Surface Oriented



filter

Surface Oriented



back-project

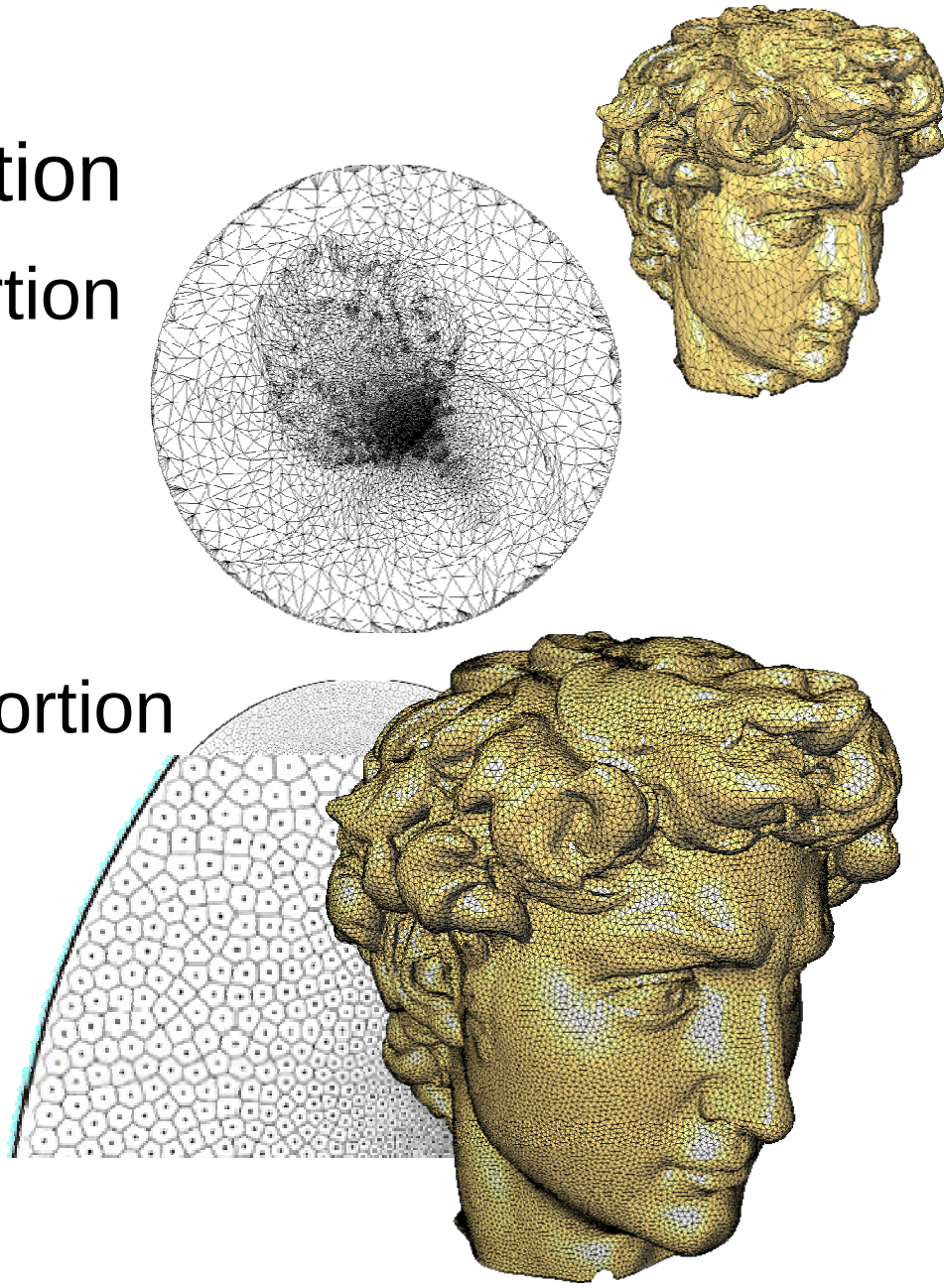
Variational Methods

- Minimize a global energy
  - Both discrete and continuous degrees of freedom
    - Connectivity, geometry
- What to optimize
  - Edge lengths, angles, triangle shapes,…
- Isotropic point sampling → well shaped triangles
  - Simple and effective

Parameterization-Based Remeshing

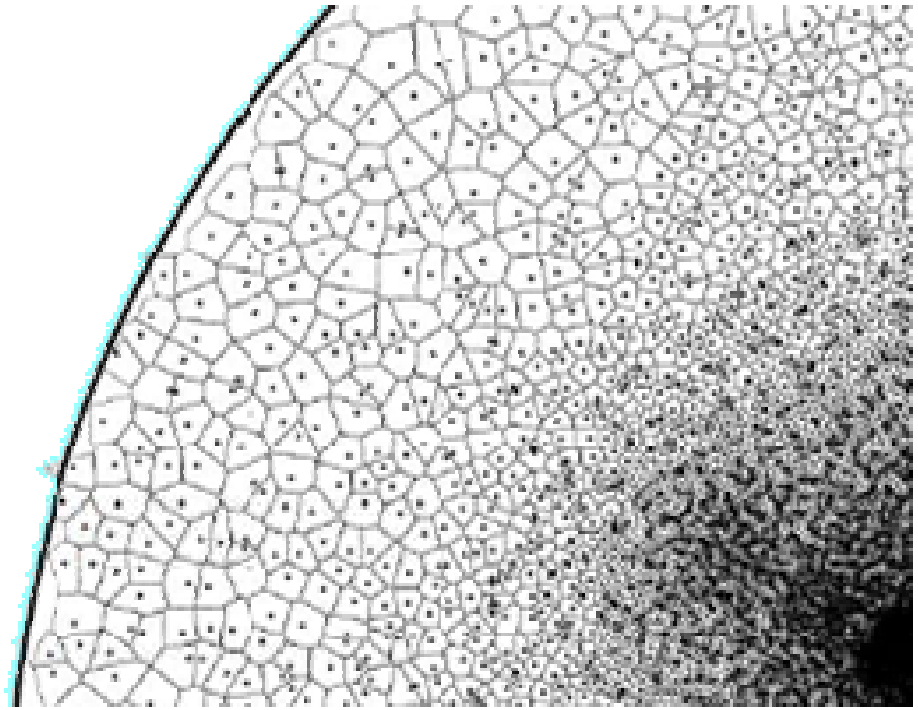
[Alliez et al. ‘03]

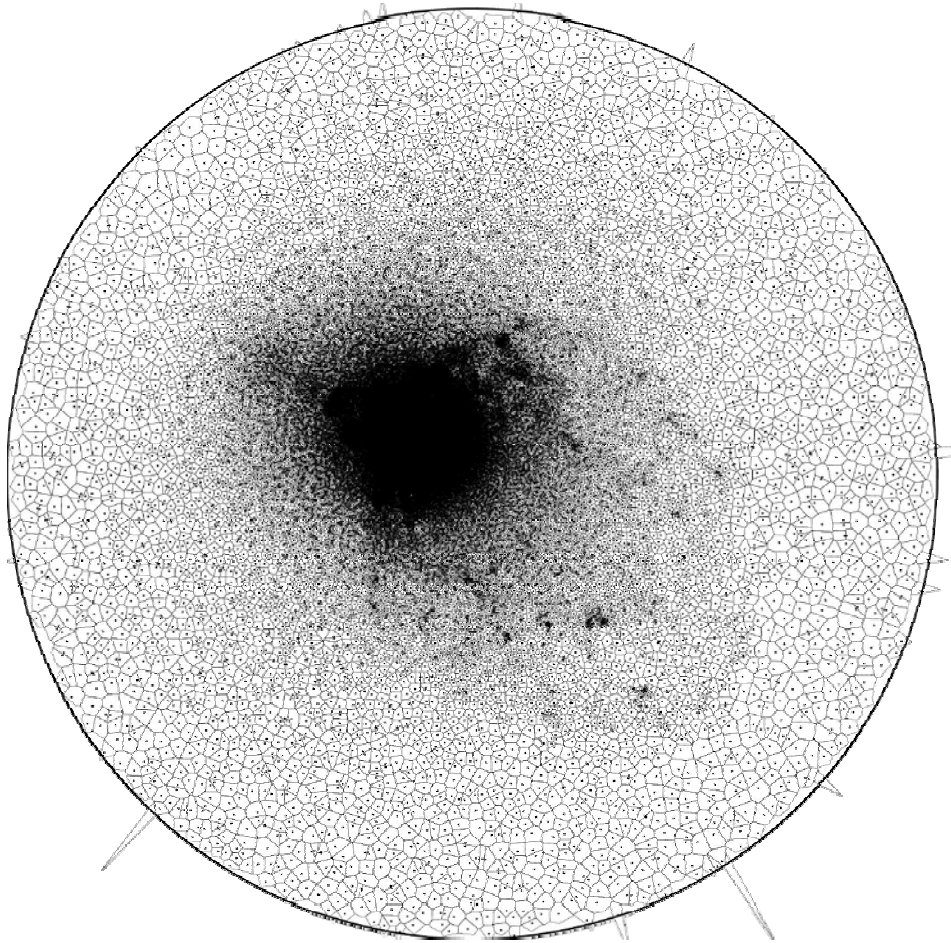
- Compute 2D parameterization
  - Conformal: only area distortion
- Sample 2D domain
  - Density based on area distortion
- Triangulate
- Project back to 3D



Isotropic 2D Sampling

- Density based random sampling
  - Does not guarantee uniform distance between samples



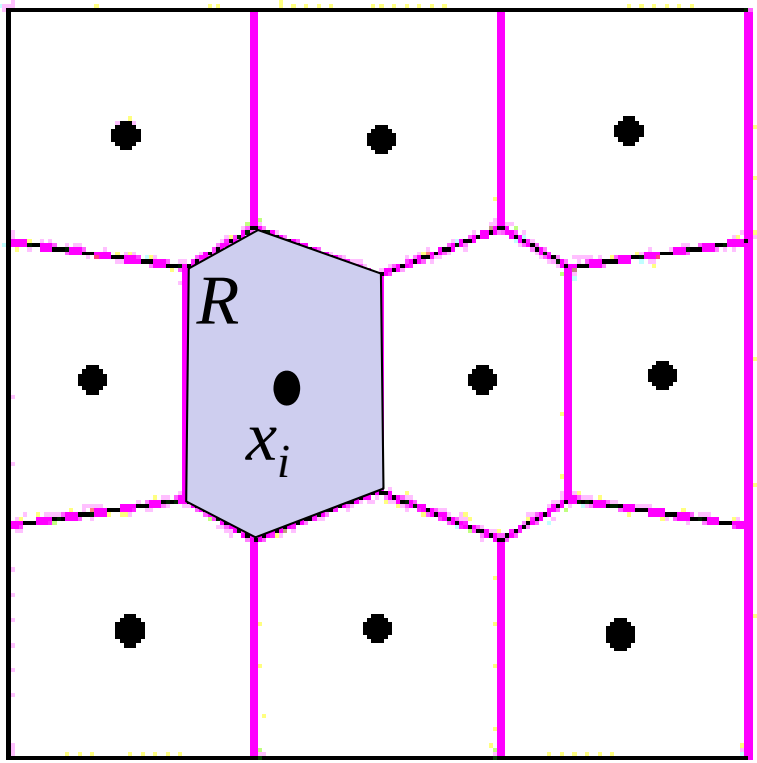


Sampling Energy

- Given sites x_i and regions R_i, minimize

$$E(x_1,..,x_k,R_1,...,R_k) = \sum_{i=1..k} \int_{x \in R_i} \| x - x_i \|^2 \, dx$$

- Spreads out points



Sampling Energy

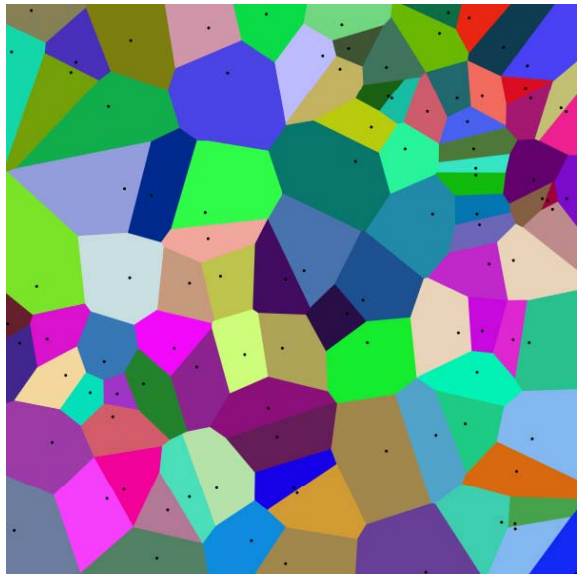
- Given sites x_i and regions R_i, minimize

$$E(x_1,..,x_k,R_1,...,R_k) = \sum_{i=1..k} \int_{x \in R_i} \| x - x_i \|^2 \, dx$$

- If sites are fixed, energy is minimized by the ***Voronoi Tesselation***

  Voronoi cell R_i =

  All points closer to $\mathbf{x}_i$ than to other $\mathbf{x}_j$



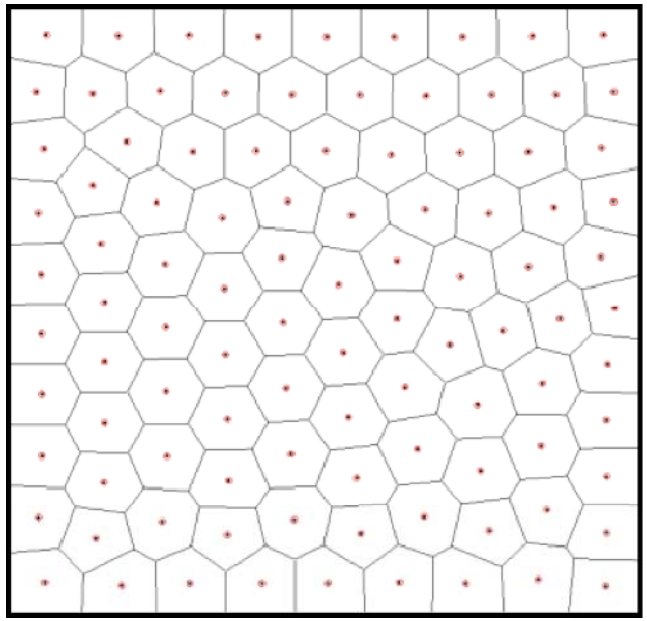
Sampling Energy

- Given sites x_i and regions R_i, minimize

$$E(x_1,..,x_k,R_1,...,R_k) = \sum_{i=1..k} \int_{x \in R_i} \| x - x_i \|^2 \, dx$$

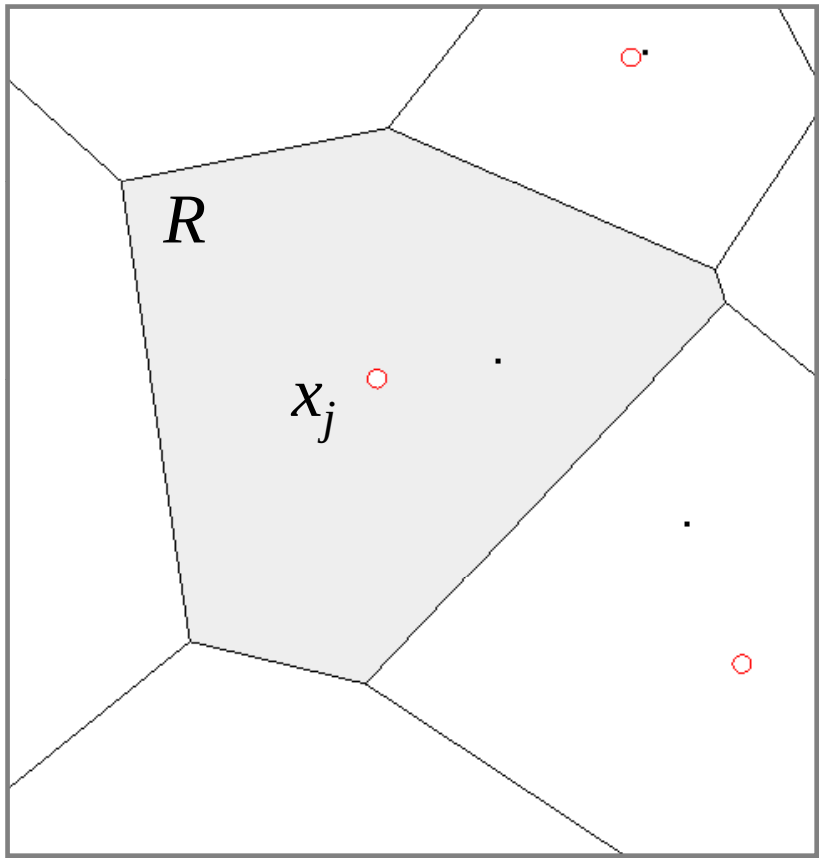
- If sites are fixed, energy is minimized by the Voronoi Tesselation

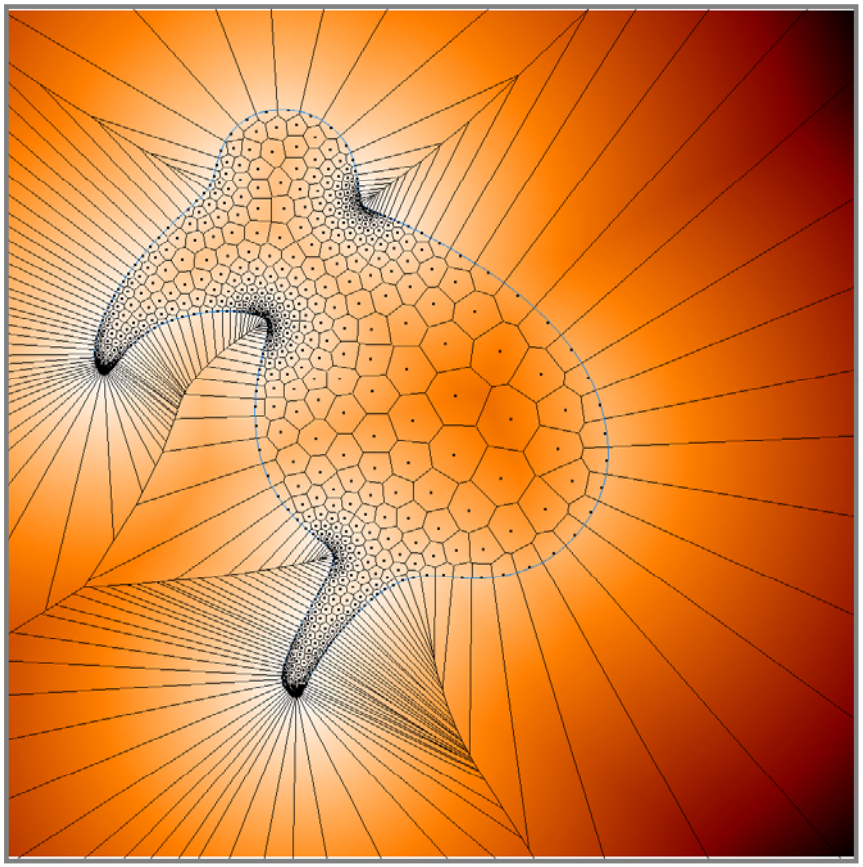
- Among all Voronoi tesselations, energy is minimized when sites are ***centroids of cells = Centroidal Voronoi Tesselation***



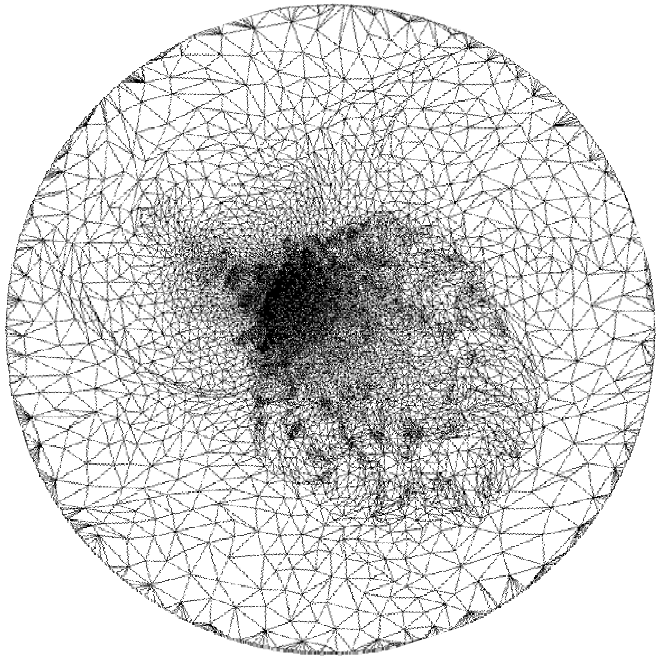
Varying Density

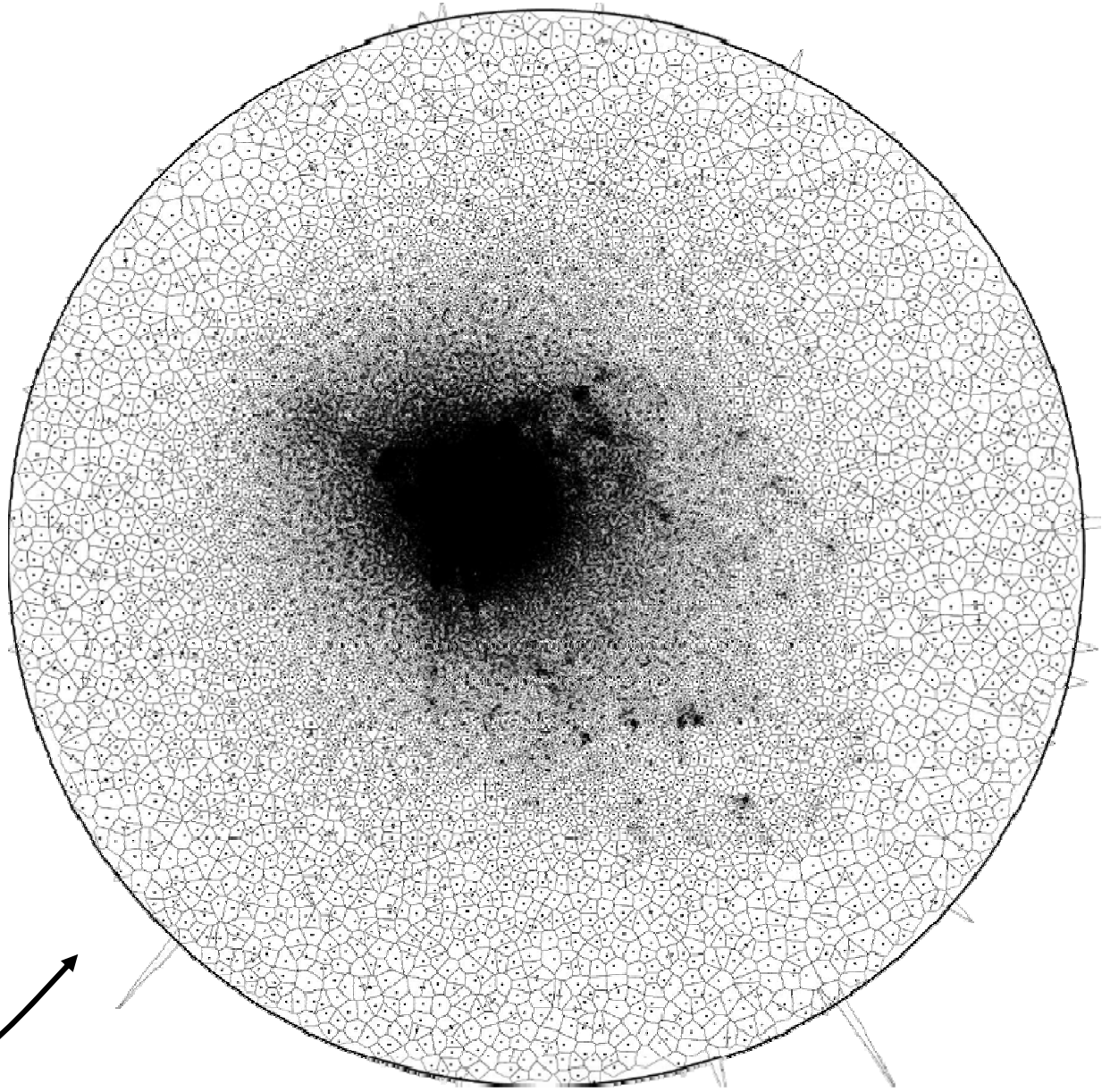
$$E(x_1,..,x_k,R_1,...,R_k) = \sum_{i=1..k} \int_{x \in R_i} \rho(x) \| x - x_i \|^2 \, dx$$





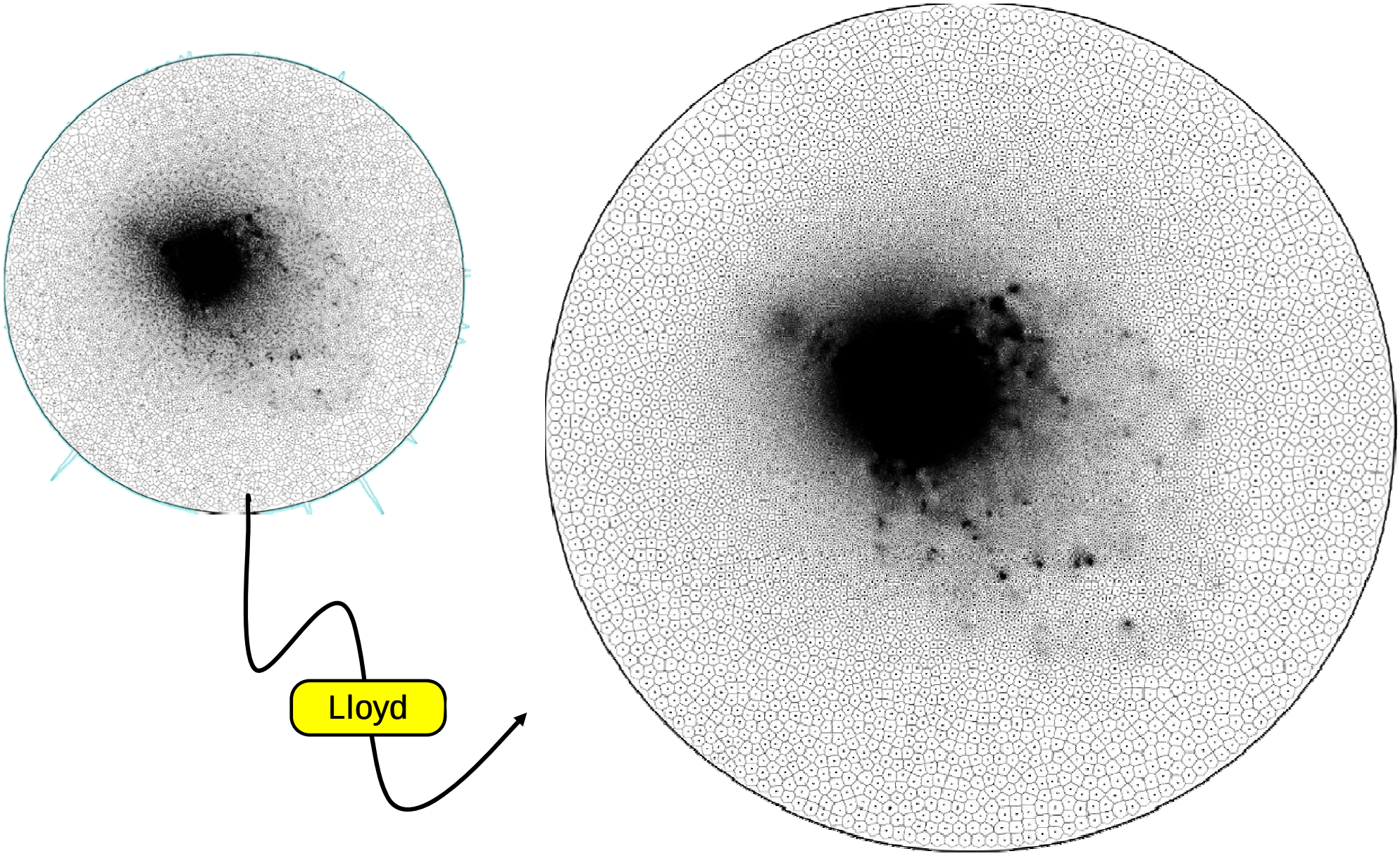
Initial Sample Scatter





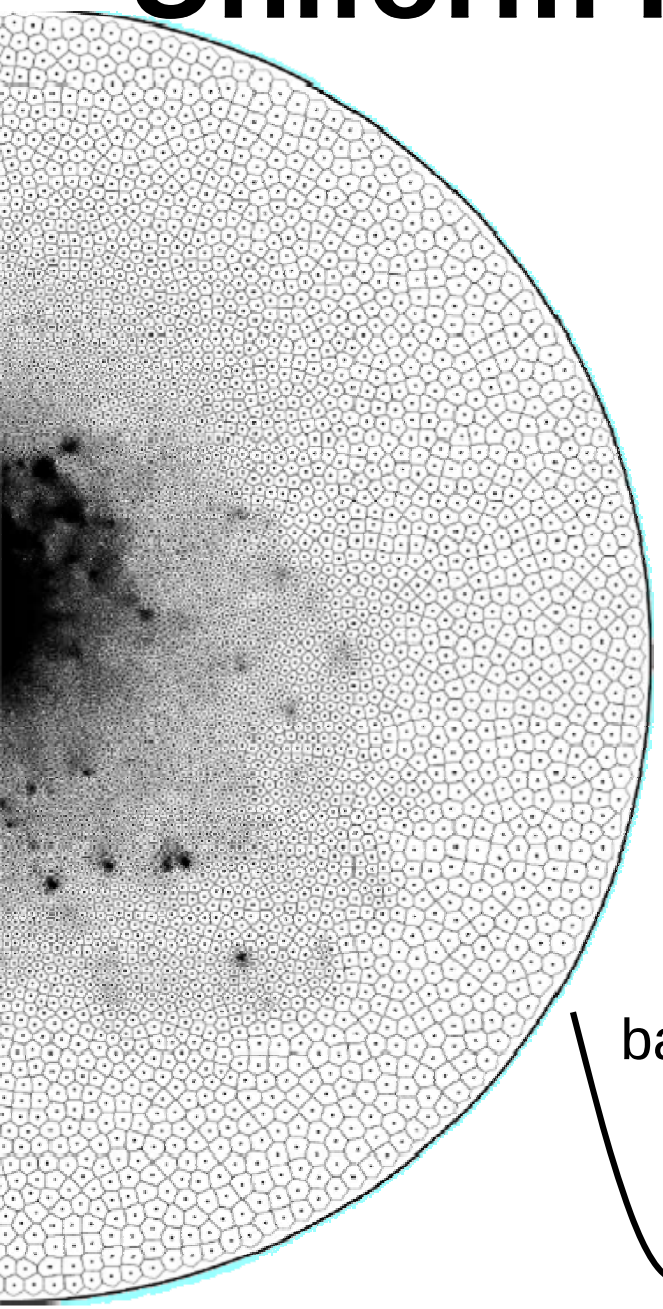
Voronoi tesselation

Optimized Sample Placement

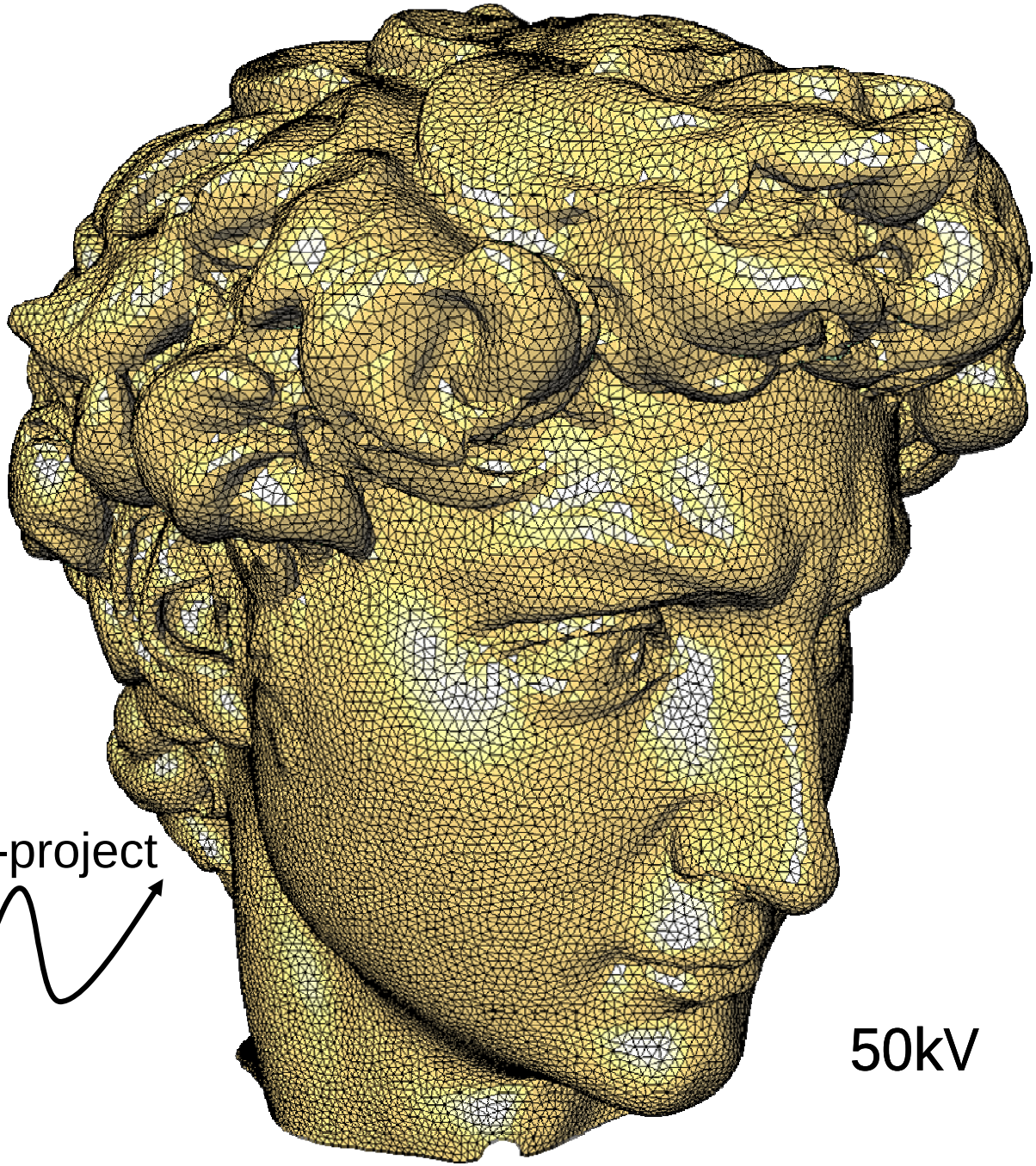


centroidal Voronoi tesselation

Uniform Remeshing



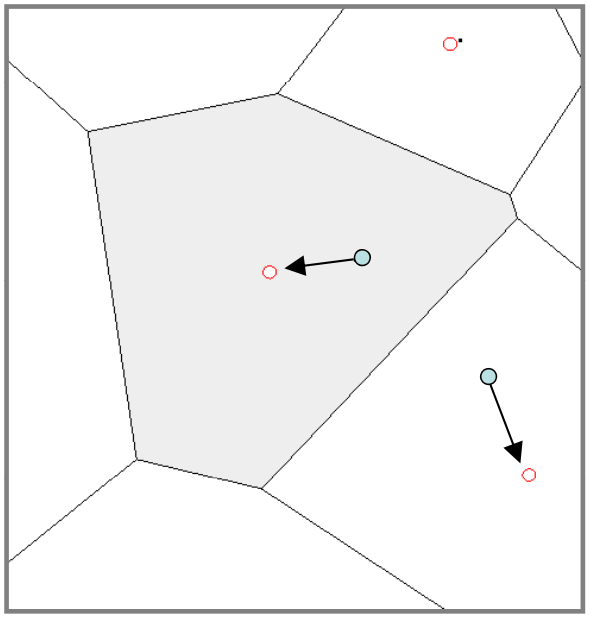
back-project



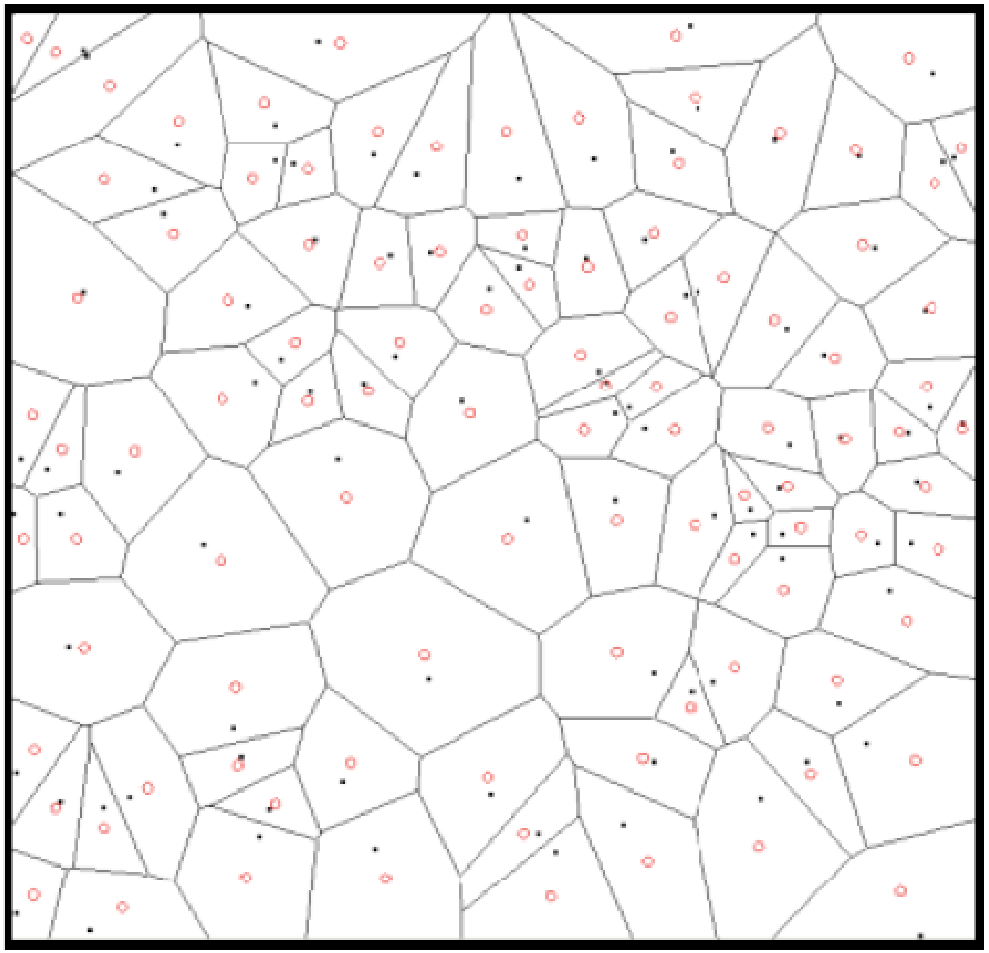
50kV

Lloyd Algorithm

- **Alternate**:
  - Voronoi partitioning
  - Move sites to respective centroids

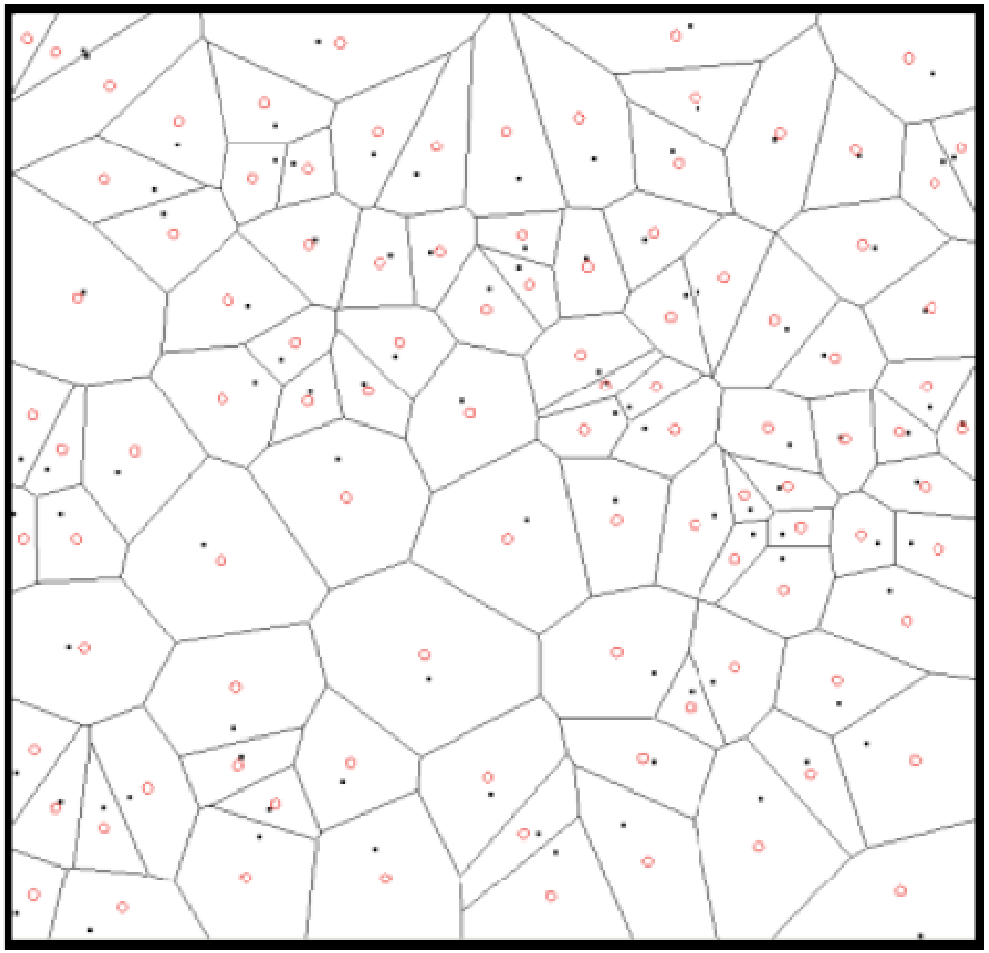


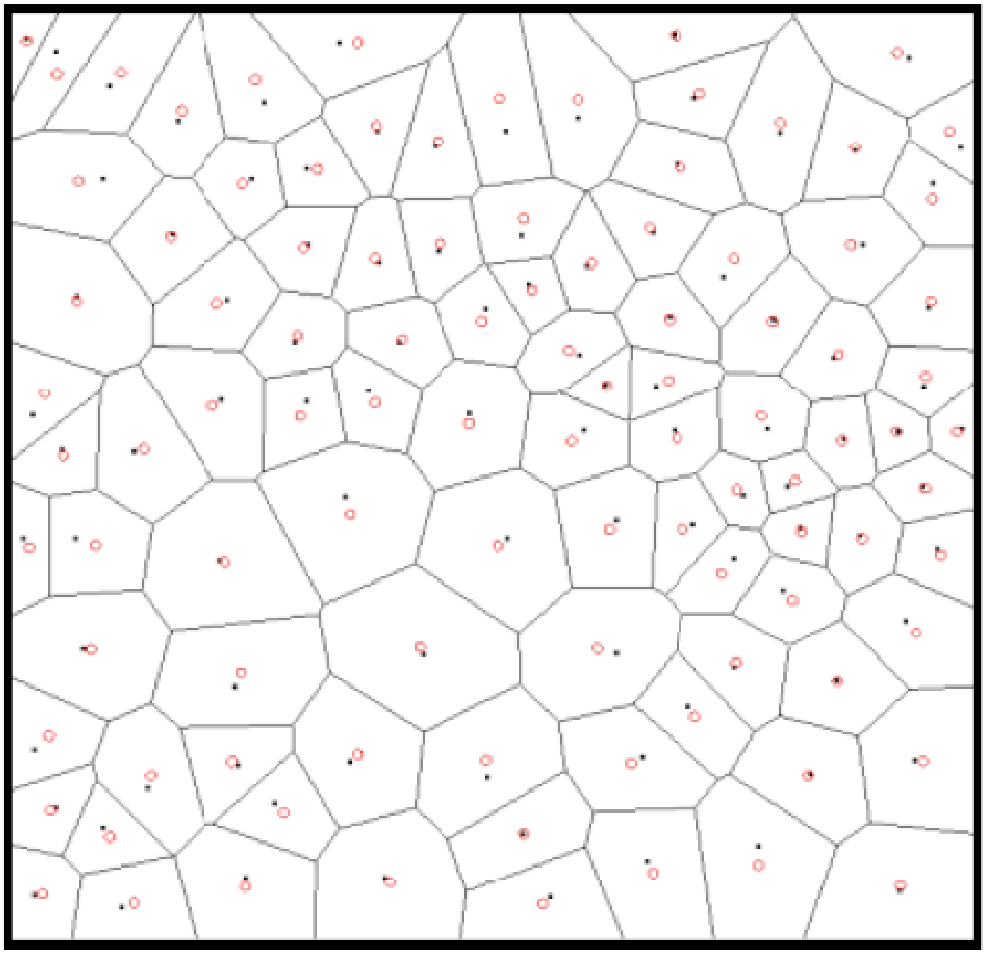
Centroidal Voronoi Diagrams



demo

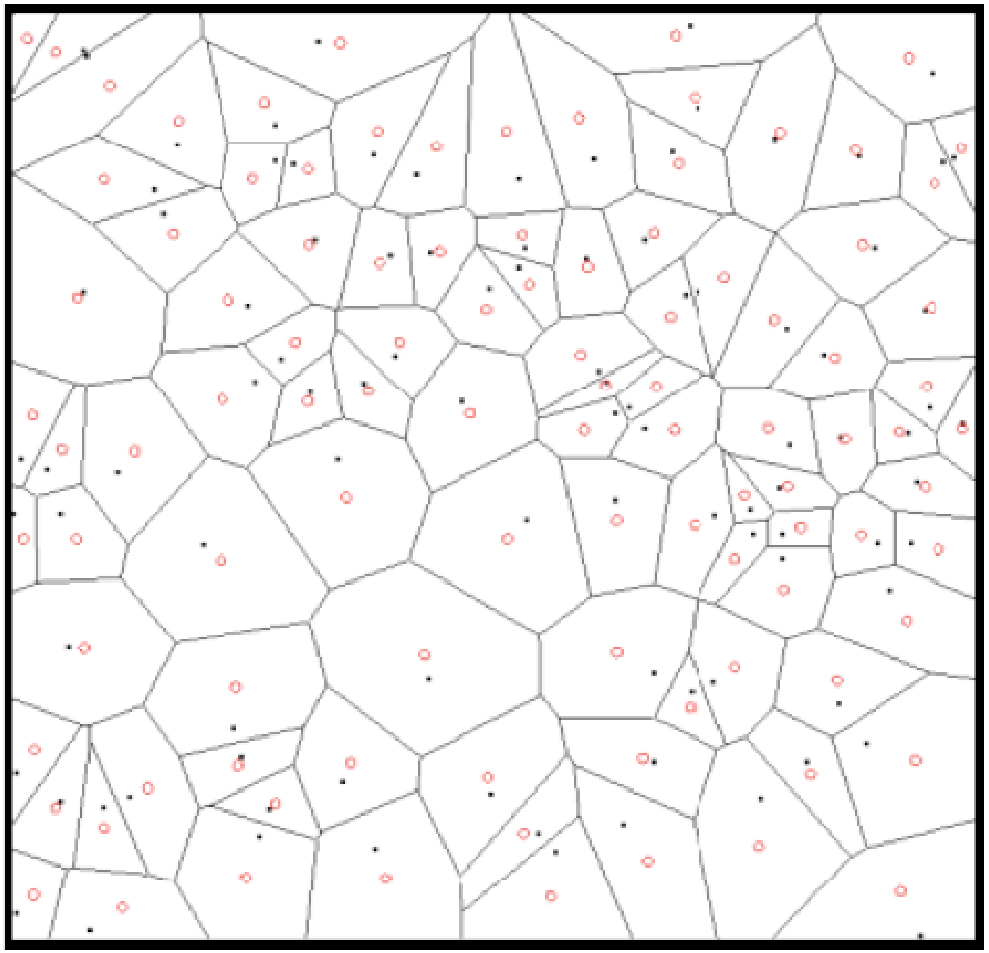
Centroidal Voronoi Diagrams

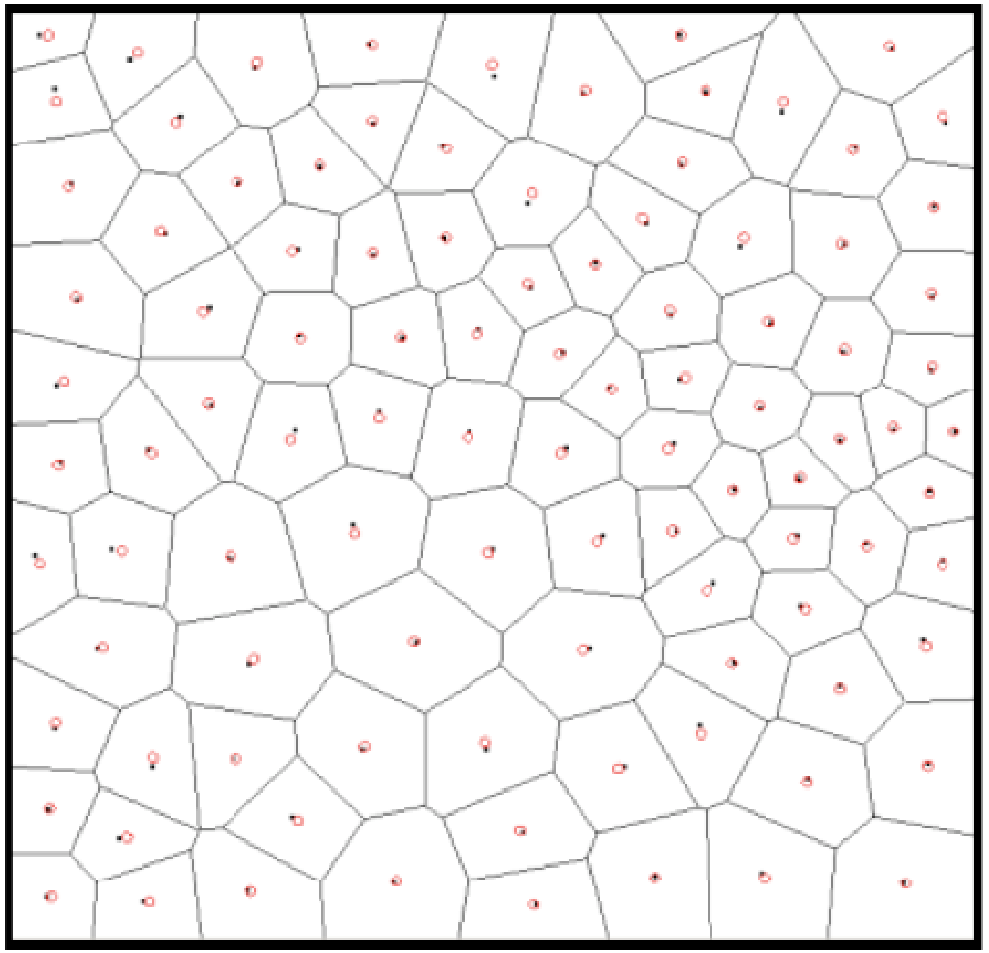




demo

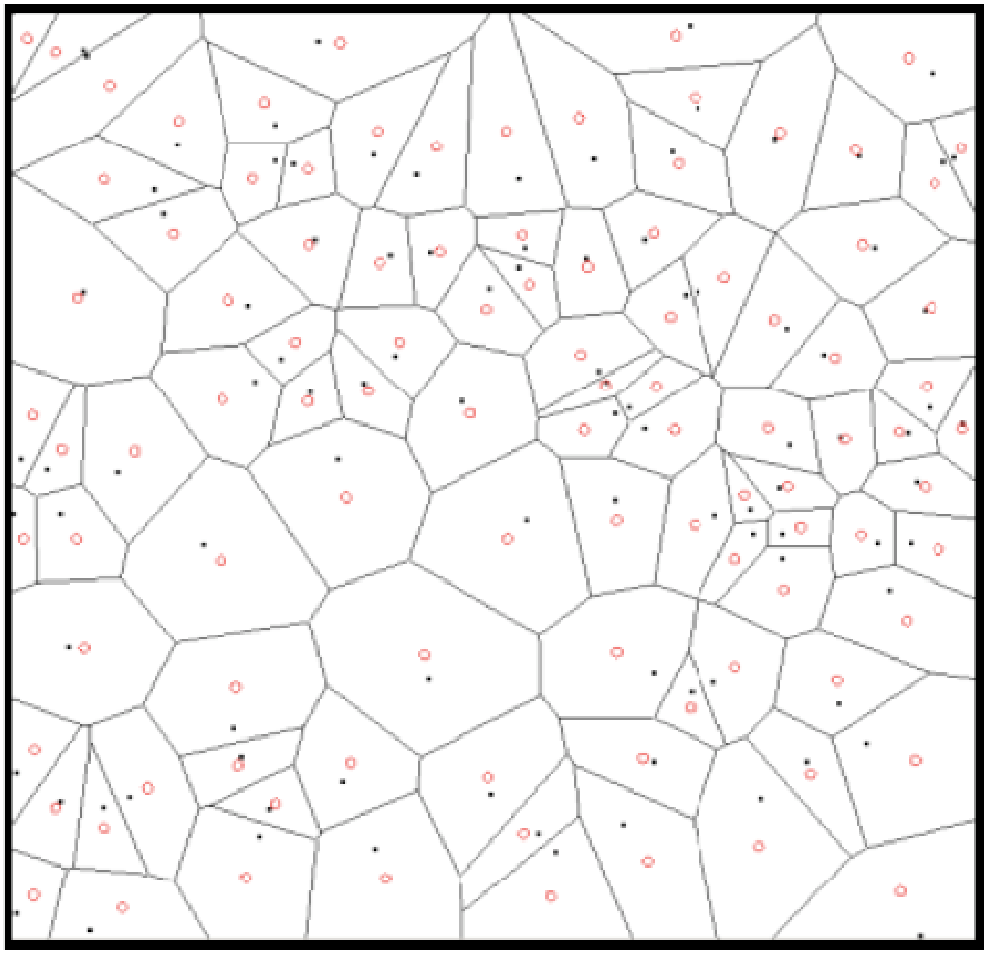
Centroidal Voronoi Diagrams

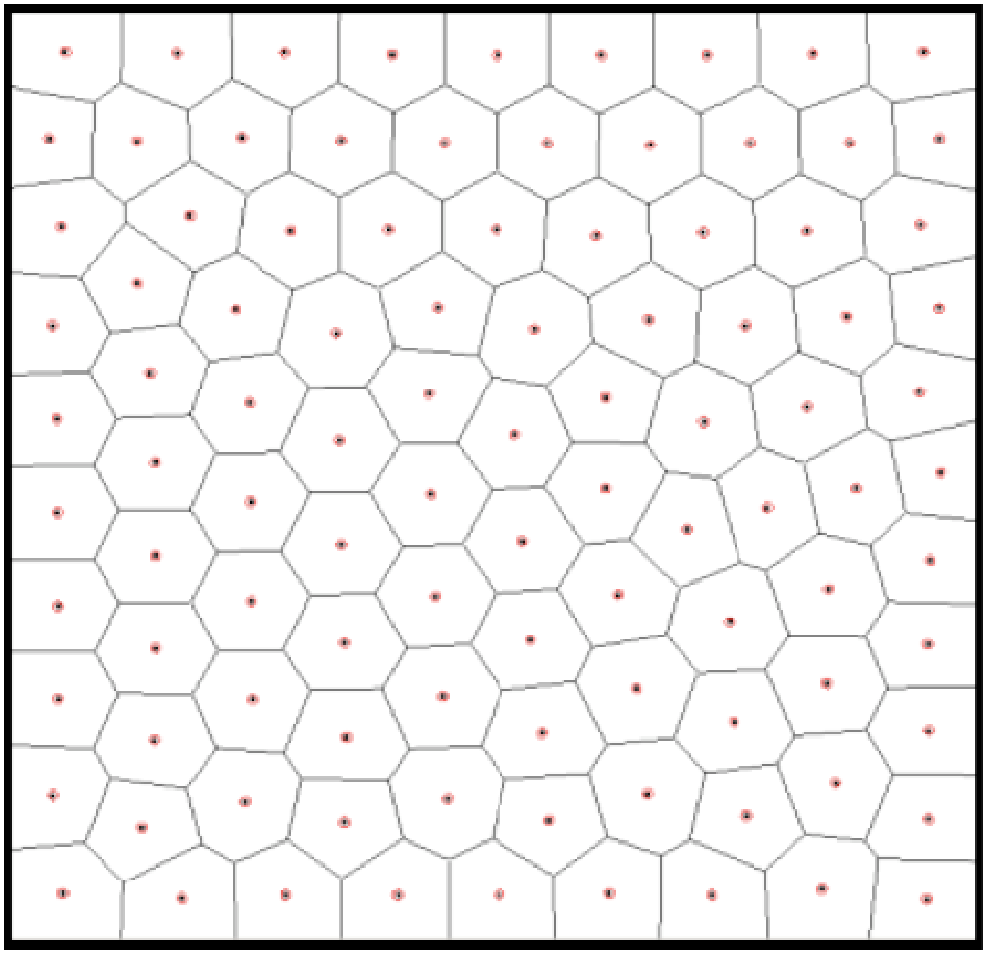




demo

Centroidal Voronoi Diagrams





demo

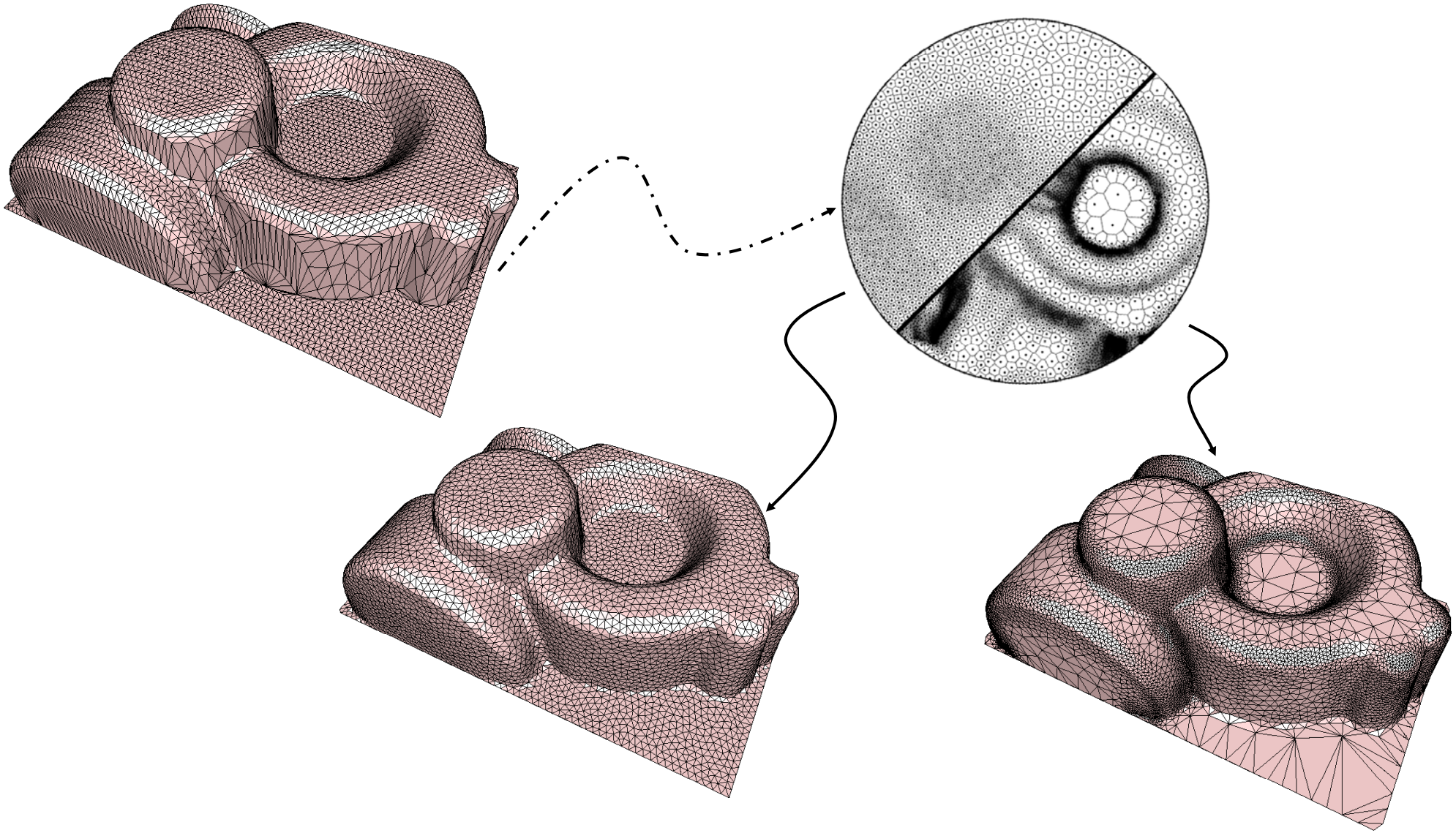
Centroidal Voronoi Tesselation

- Lloyd converges slowly
  - Stop when points “stop” moving

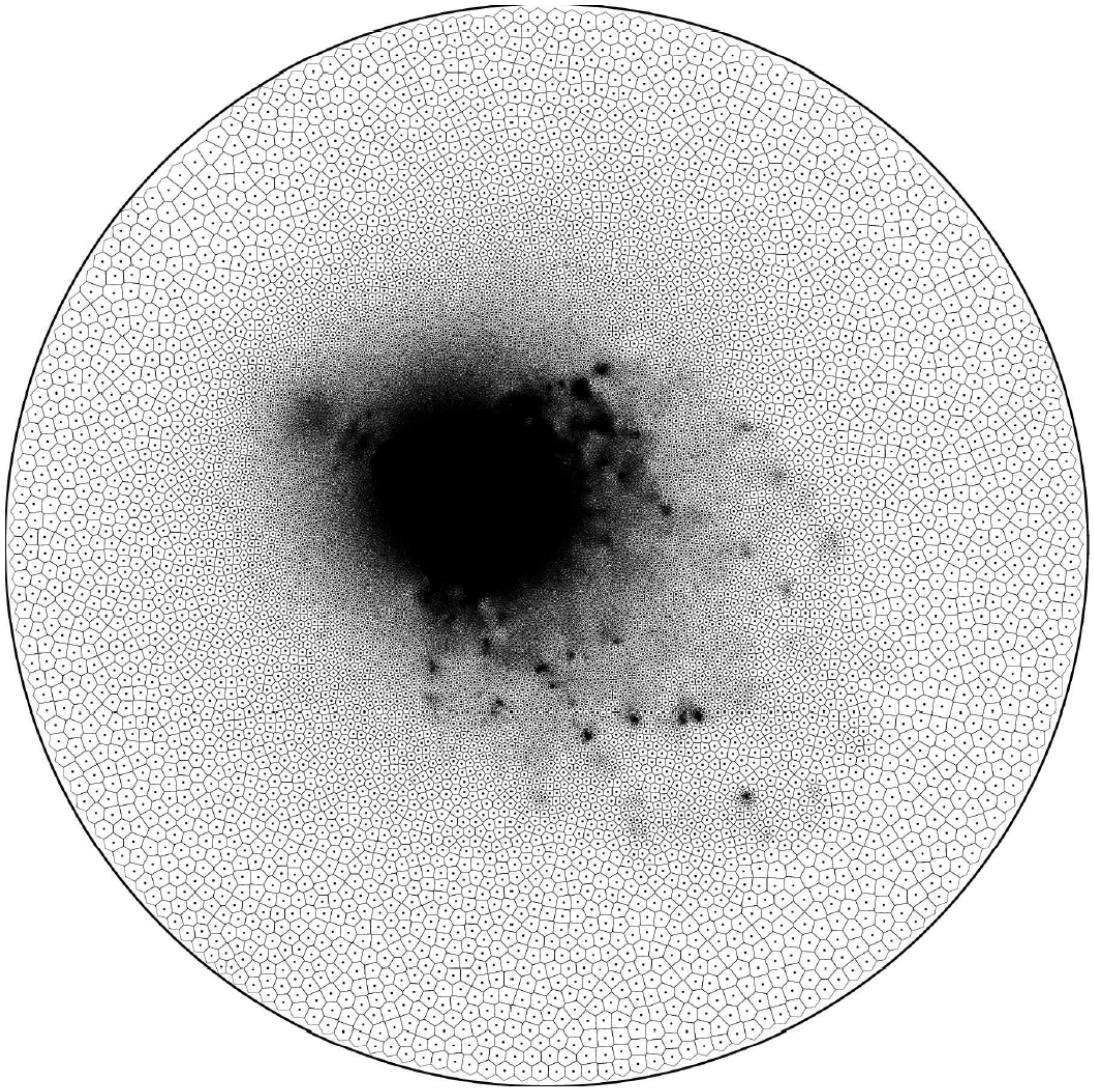
- Recent fast algorithm: direct optimization of the energy using quasi-Newton

  “On centroidal voronoi tessellation—energy smoothness and fast computation” [Liu et al., TOG ’09]

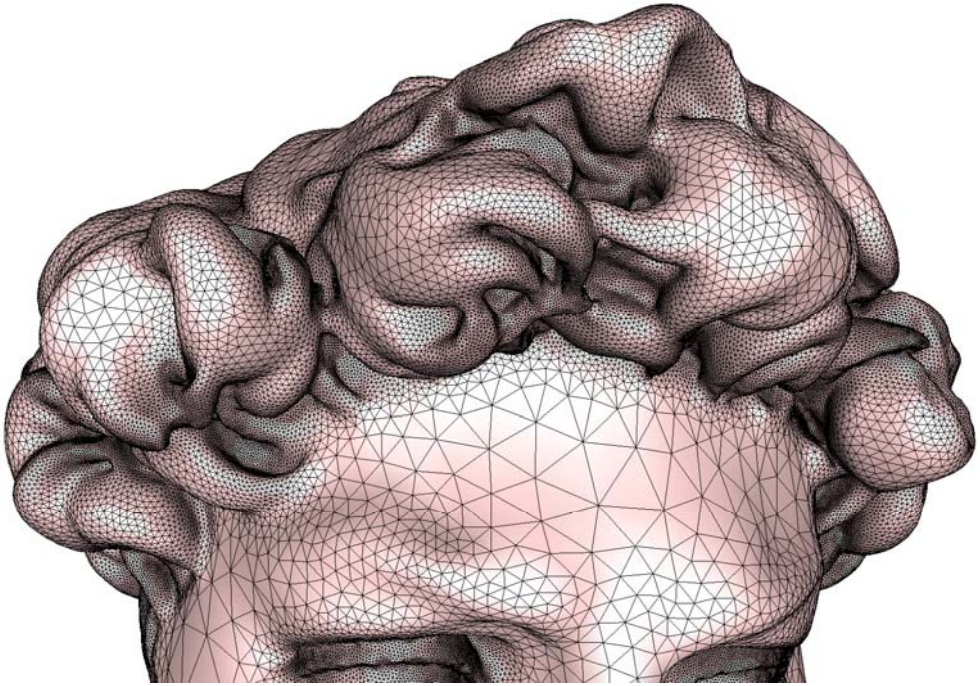
Uniform vs. Adaptive

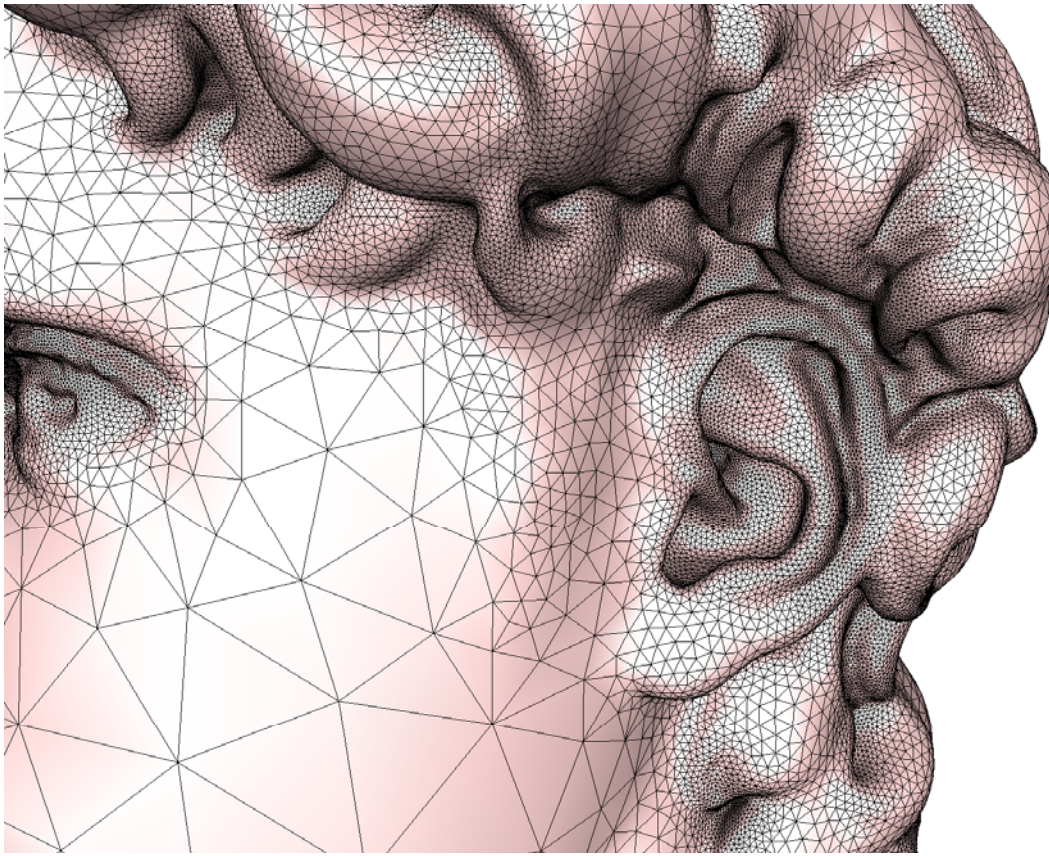


Uniform Sampling



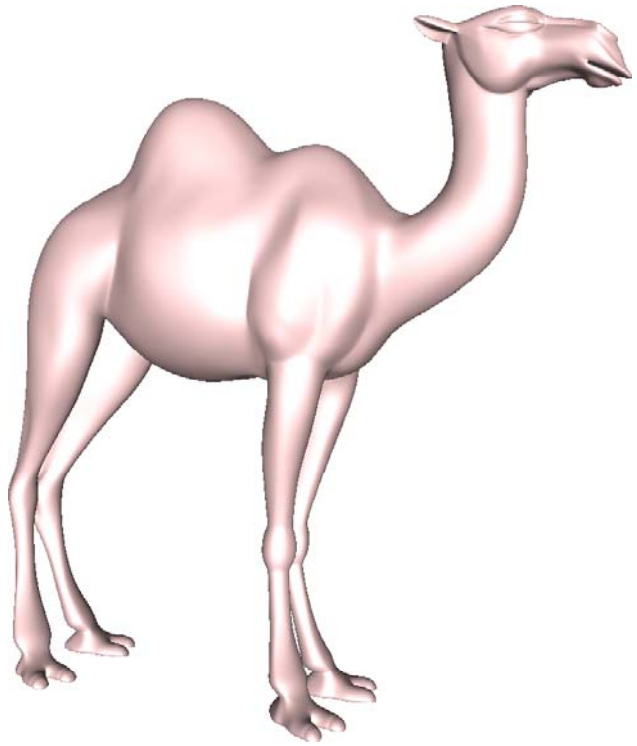
Adaptive Sampling



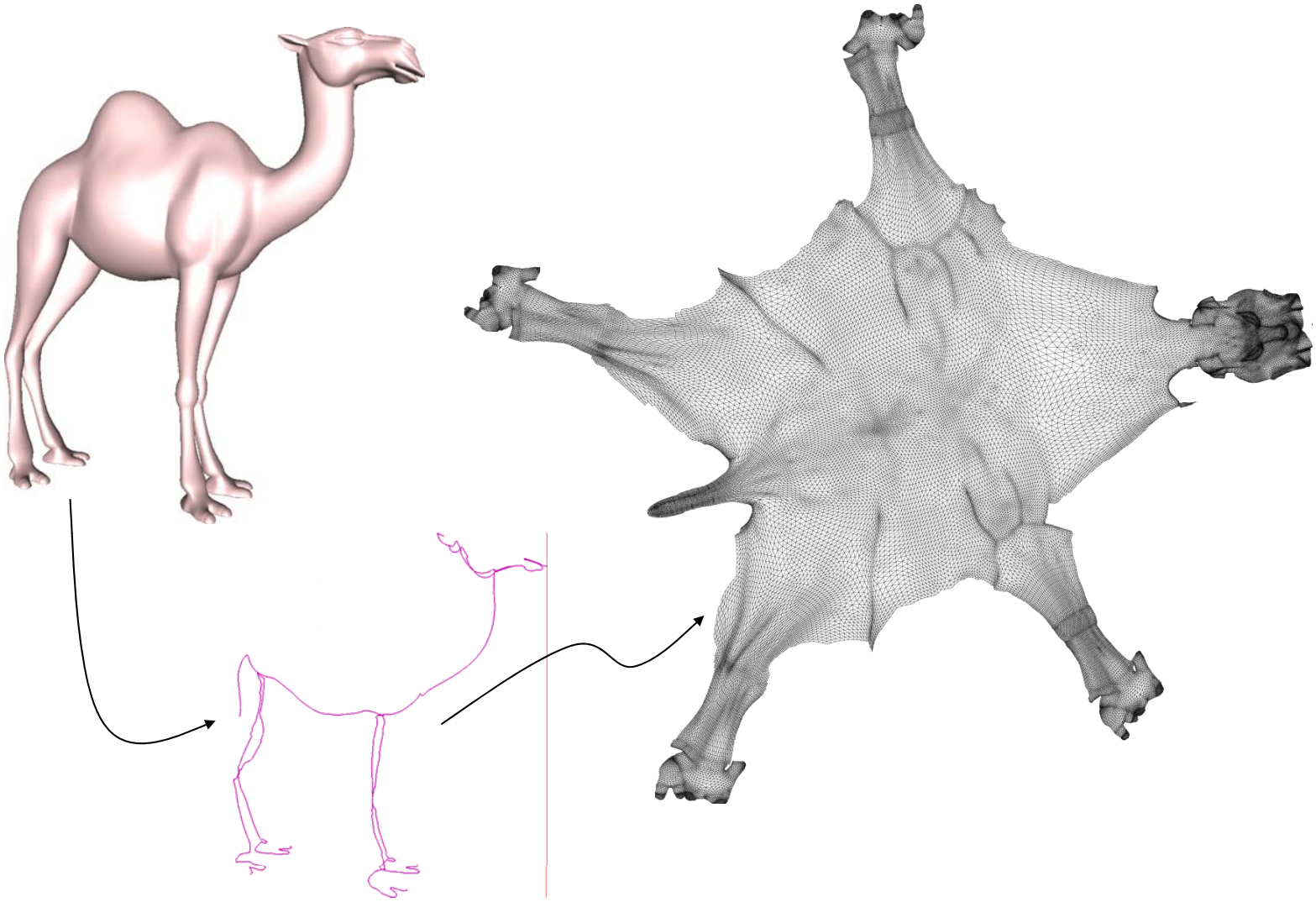


Limitations

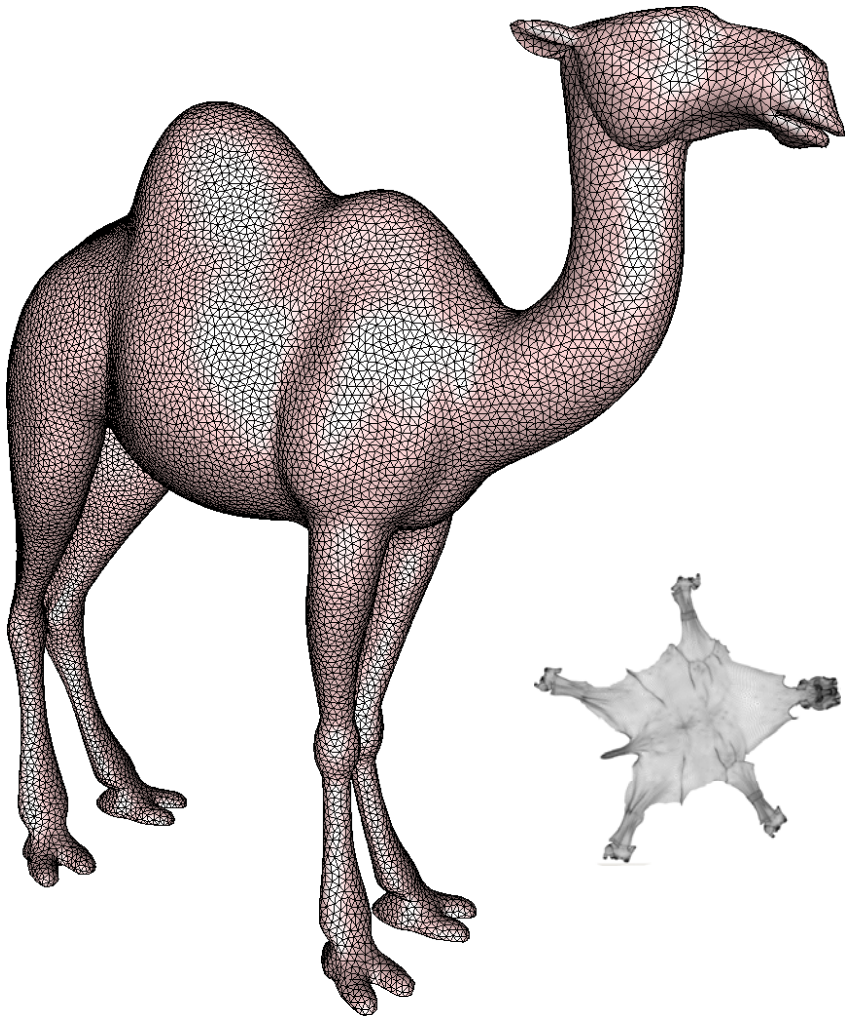
- Closed meshes
  - Need a good cut
  - Free boundary parameteriztion
  - Stitch seams afterwards
- Protruding legs
  - Sampling
  - Numerical problems

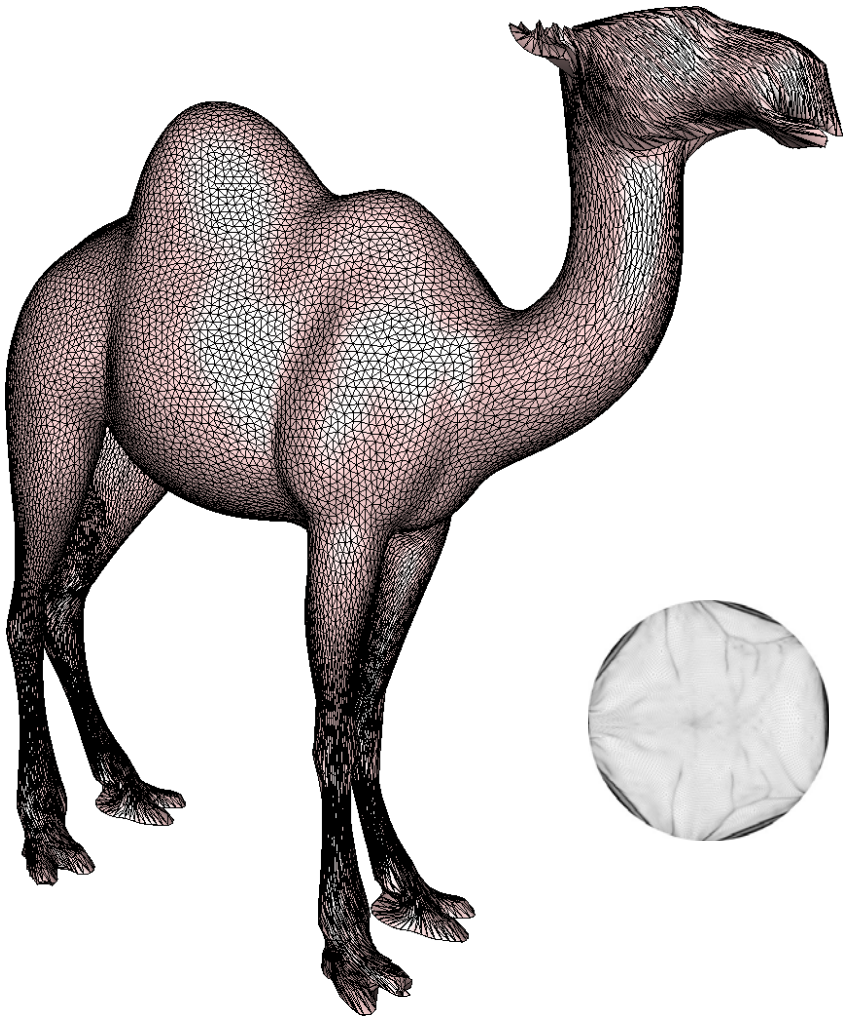


Smart Cut, Free Boundary

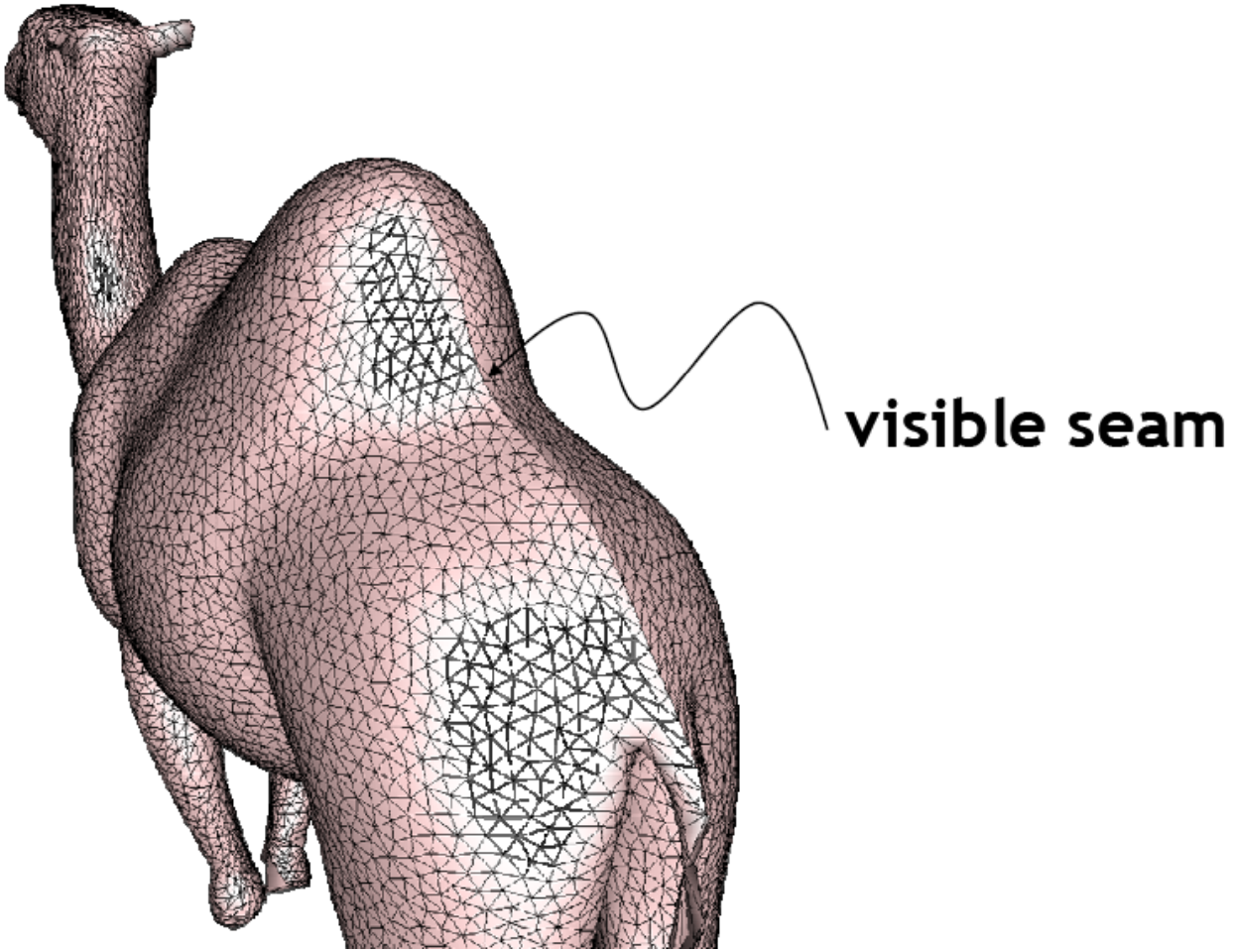


Free vs. Fixed Boundary

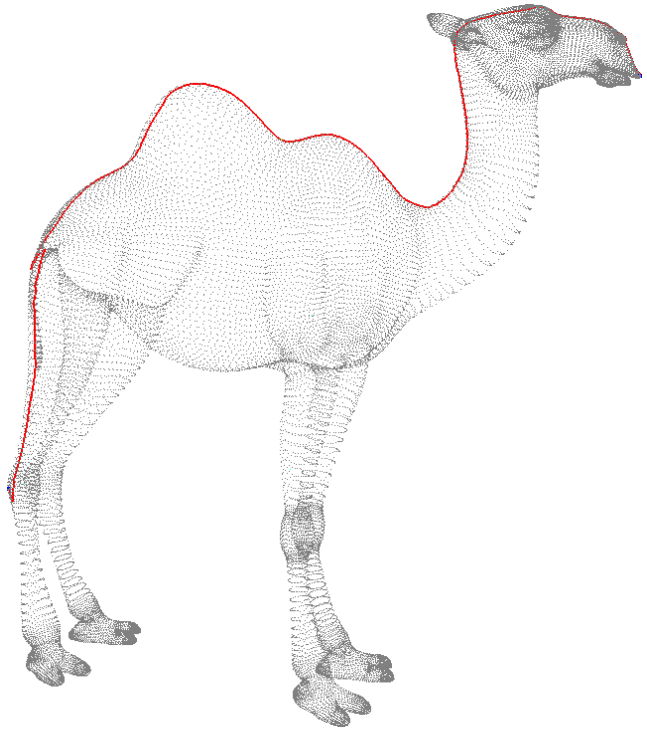


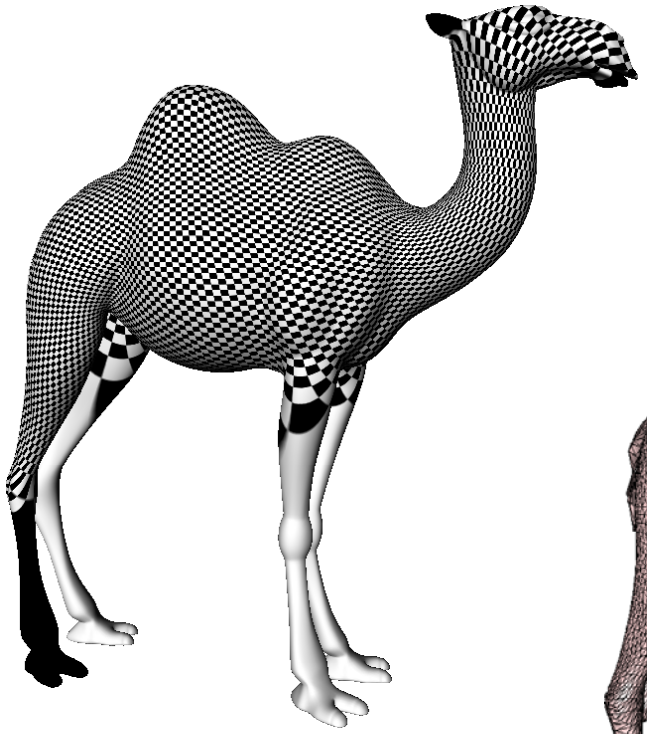


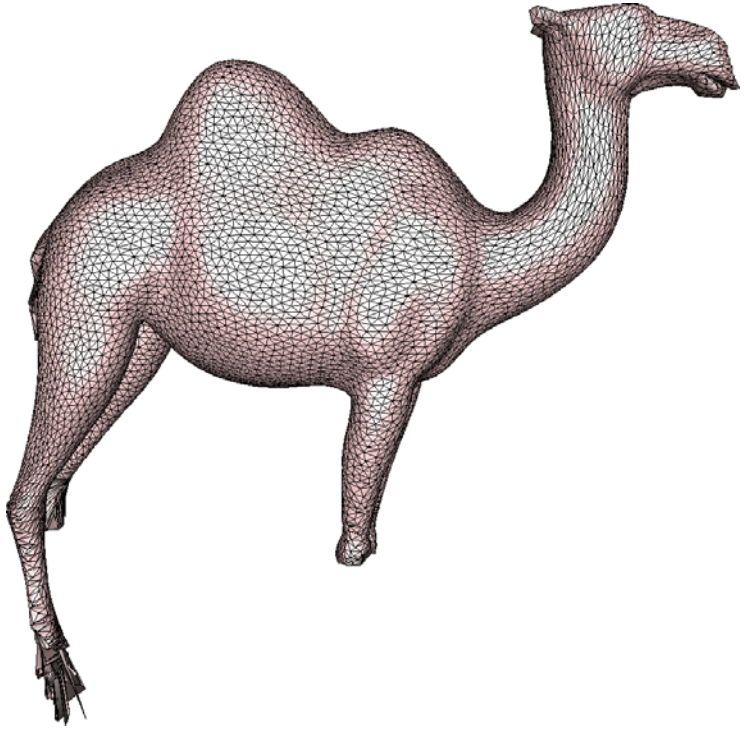
Visible Seams



Naive Cut, Numerical Problems



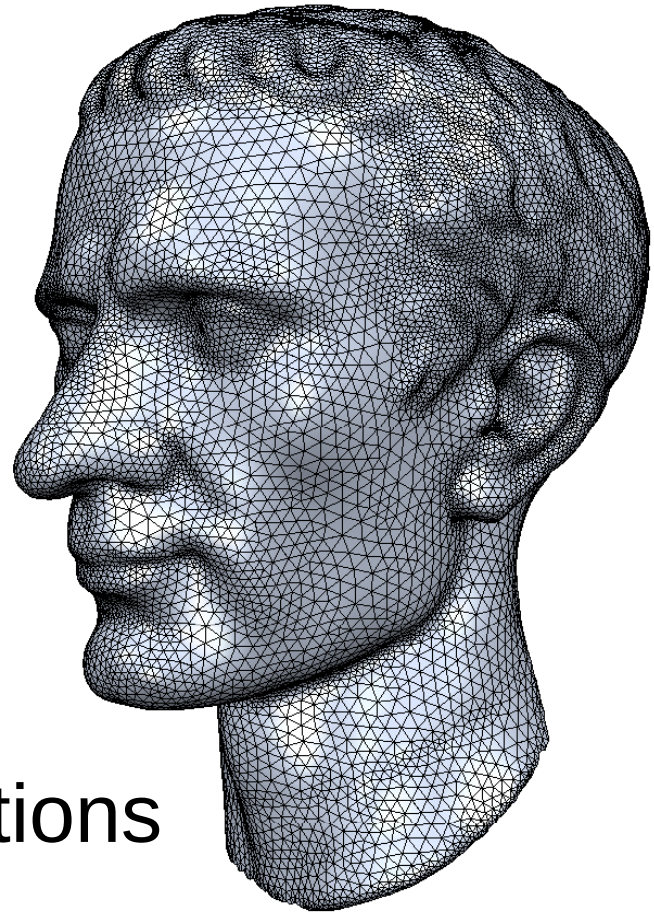




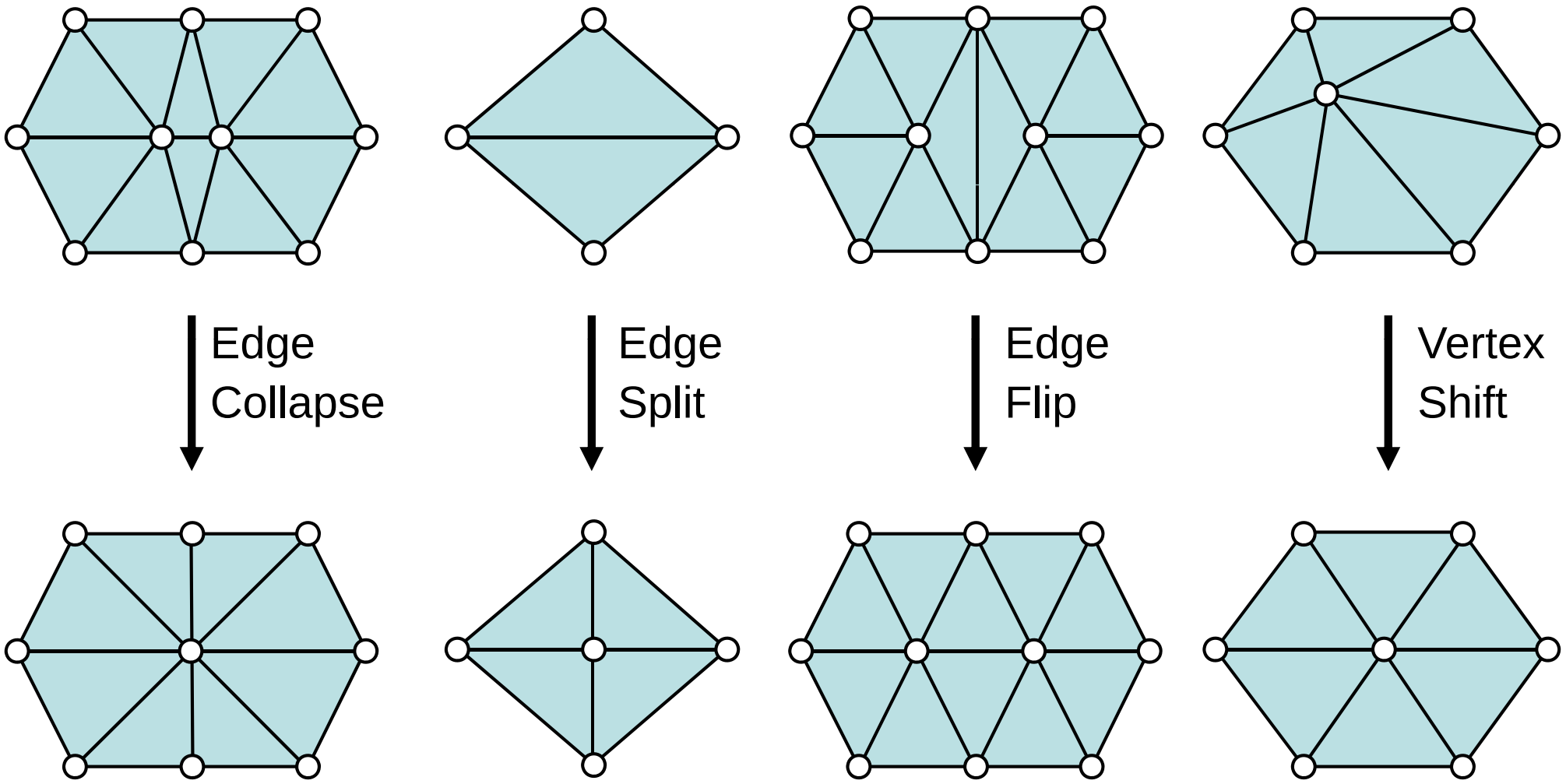
Direct Surface Remeshing

[Botsch et al. ’04]

- Avoid global parameterization
  - Numerically very sensitive
  - Topological restrictions
- Avoid local parameterizations
  - Expensive computations
- Use local operators & back-projections
  - Resampling of 100k triangles in < 5s



Local Remeshing Operators



Isotropic Remeshing

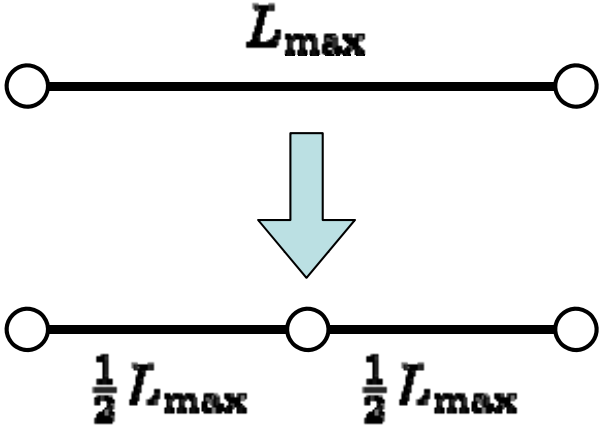
Specify target edge length L

Compute edge length range $[L_{min}, L_{max}]$

Iterate:

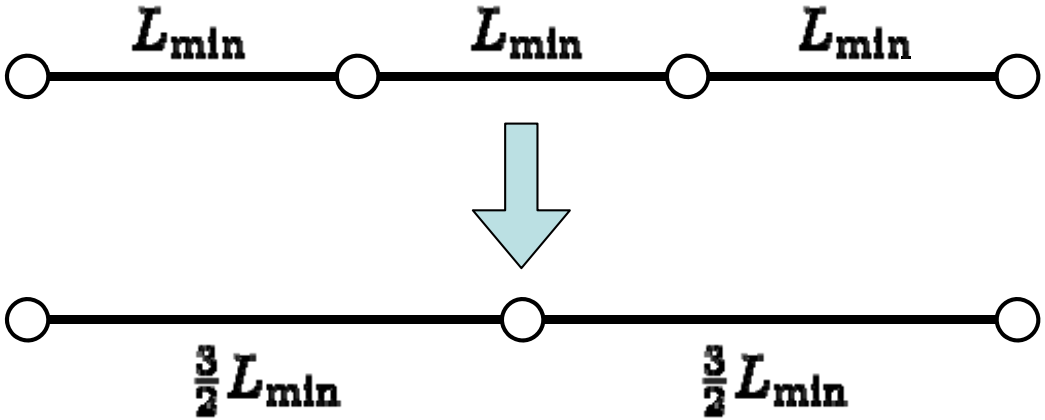
1. Split edges longer than L_{max}
2. Collapse edges shorter than L_{min}
3. Flip edges to get closer to valence 6
4. Vertex shift by tangential relaxation
5. Project vertices onto reference mesh

Edge Collapse / Split



$$|L_{max} - L| = \left|\frac{1}{2}L_{max} - L\right|$$

$$\Rightarrow L_{max} = \frac{4}{3}L$$

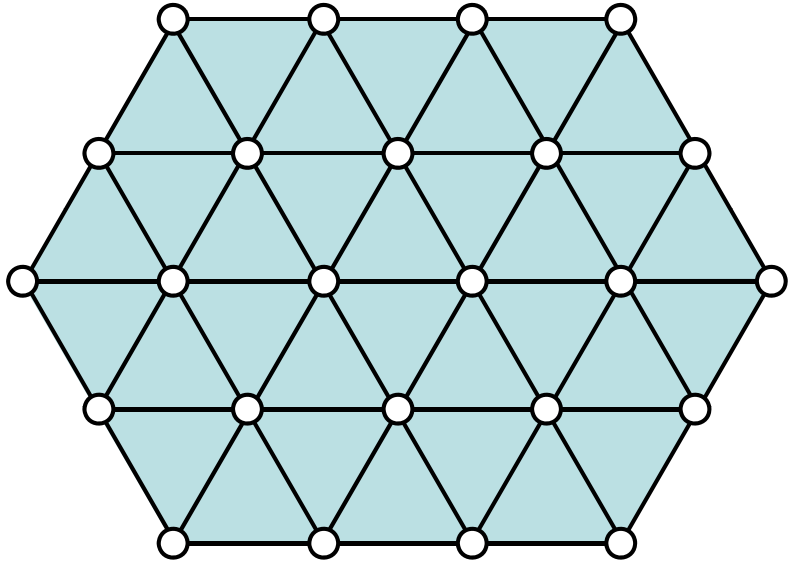


$$|L_{min} - L| = \left|\frac{3}{2}L_{min} - L\right|$$

$$\Rightarrow L_{min} = \frac{4}{5}L$$

Edge Flip

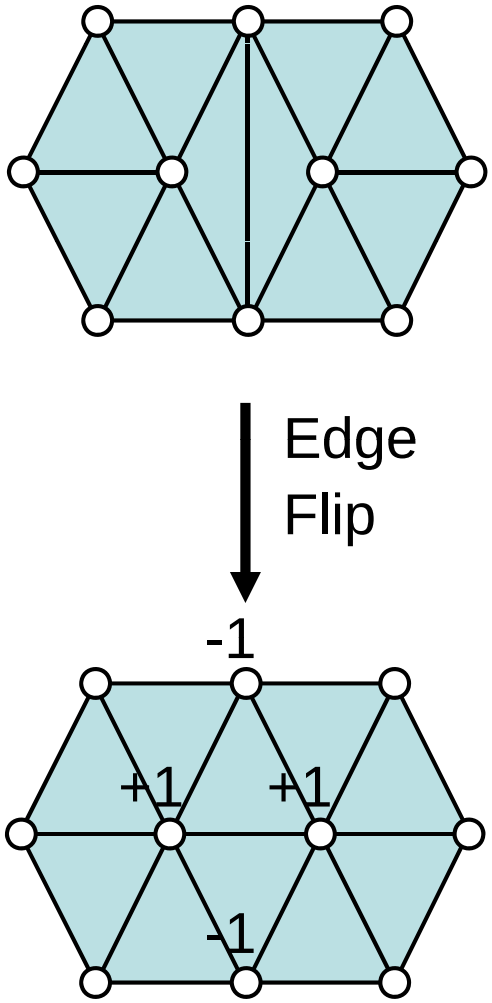
- Improve valences
  - Avg. valence is 6 (Euler)
  - Reduce variation
- Optimal valence is
  - 6 for interior vertices
  - 4 for boundary vertices



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

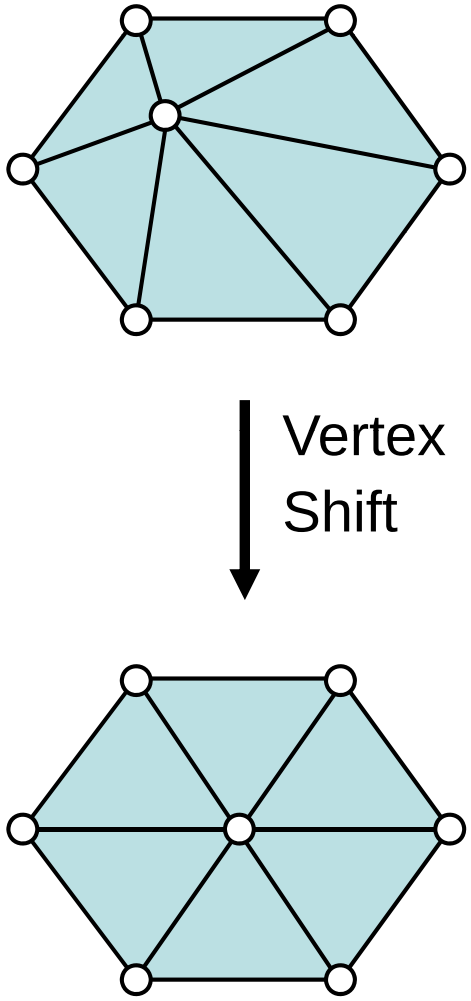
$$\sum_{i=1}^{4} (\text{valence}(v_i) - \text{opt\_valence}(v_i))^2$$



Vertex Shift

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

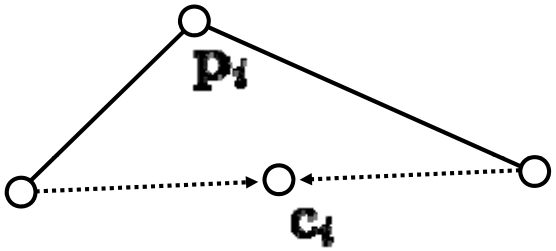
$$\mathbf{c}_i = \frac{1}{\text{valence}(v_i)} \sum_{j \in N(v_i)} \mathbf{p}_j$$



Vertex Shift

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

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



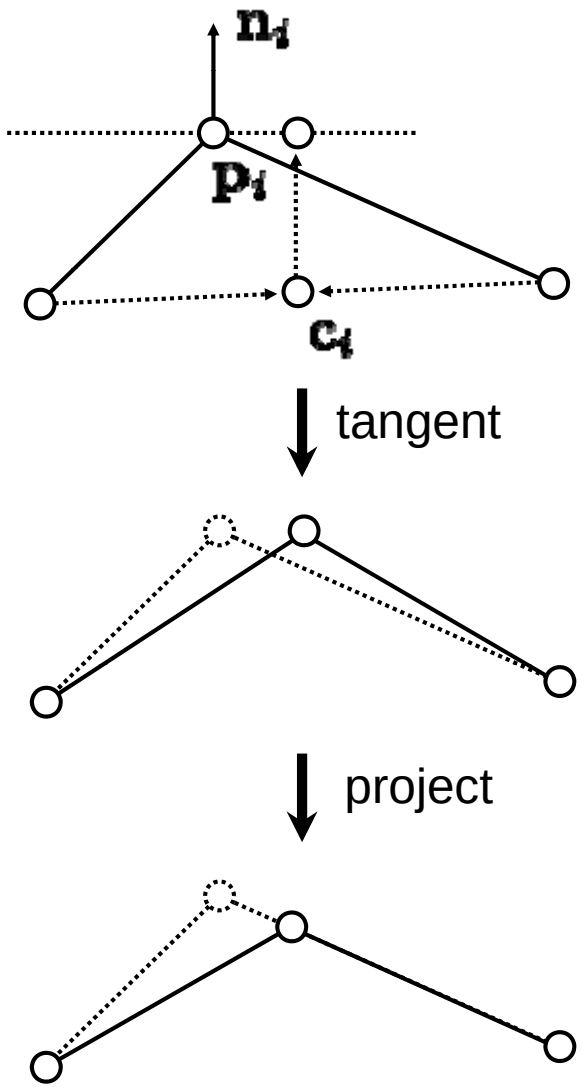
Vertex Shift

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

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

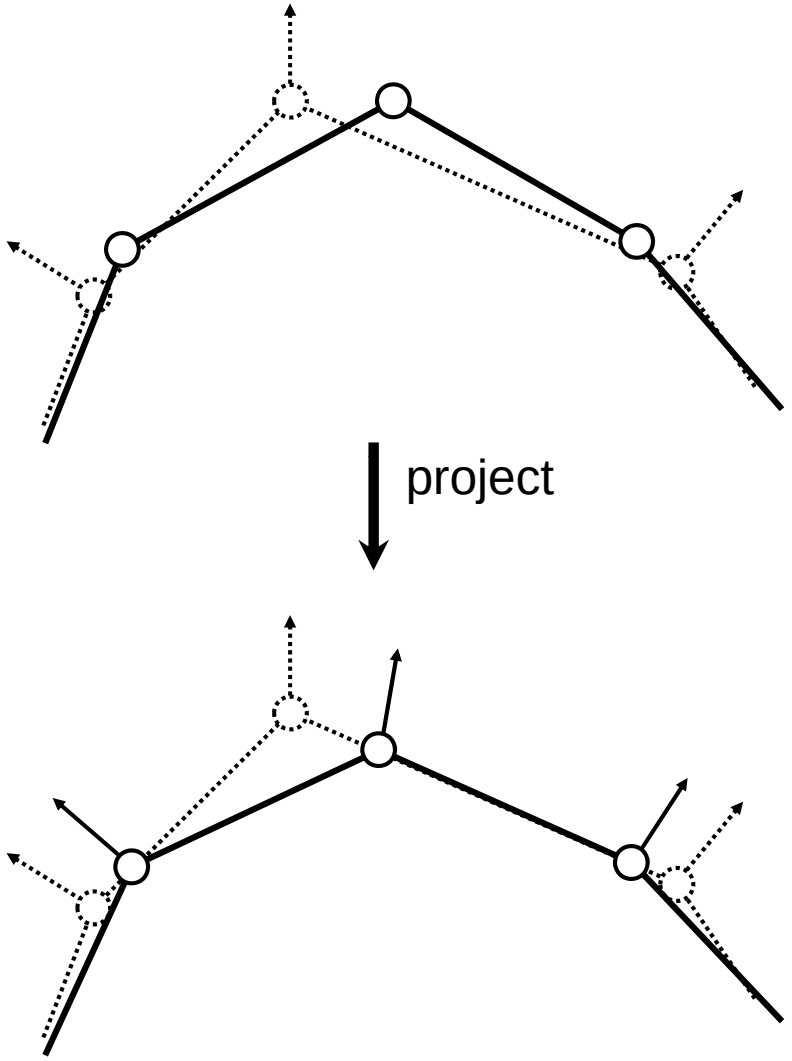
- Keep vertex (approx.) on surface
  - Restrict movement to tangent plane

$$\mathbf{p}_i \leftarrow \mathbf{p}_i - \mathbf{n}_i \mathbf{n}_i^T (\mathbf{c}_i - \mathbf{p}_i)$$

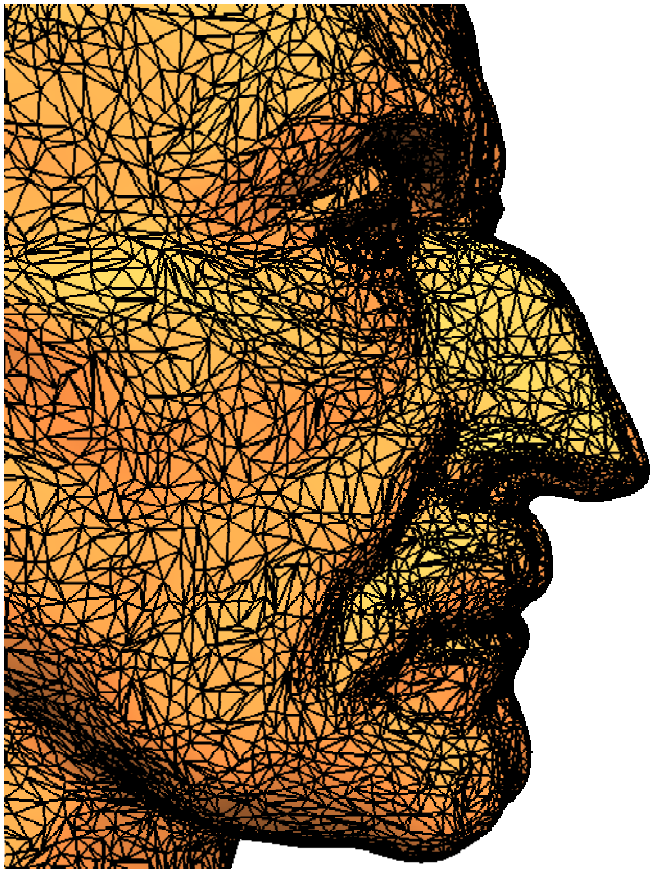


Vertex Projection

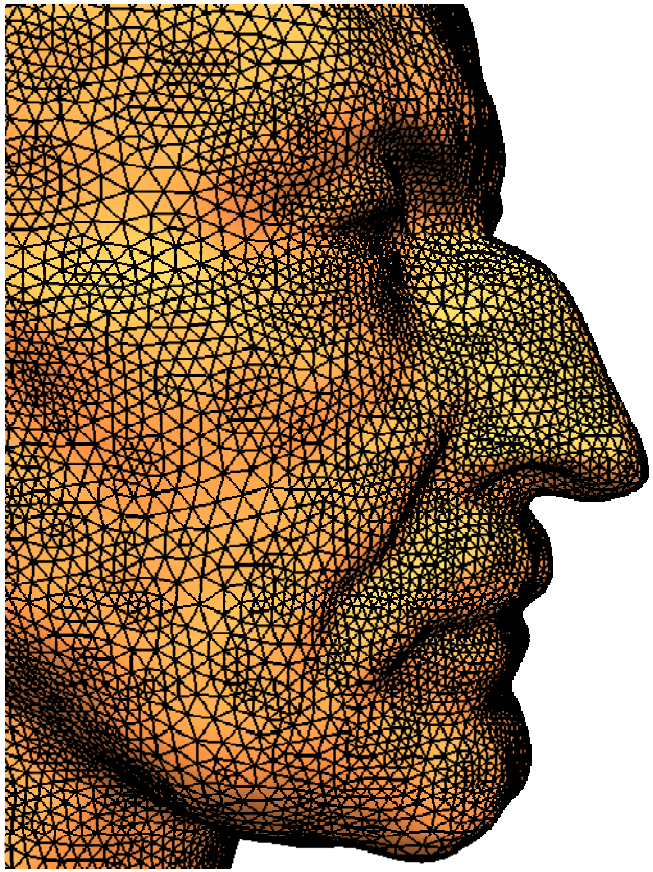
- Project vertices onto original reference mesh
- Assign position & interpolated normal



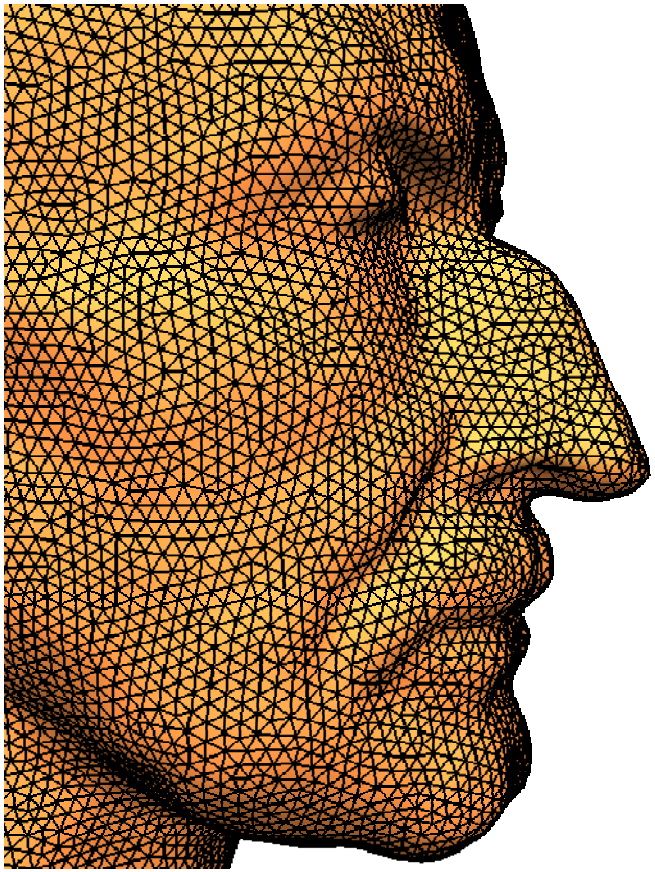
Remeshing Results



Original

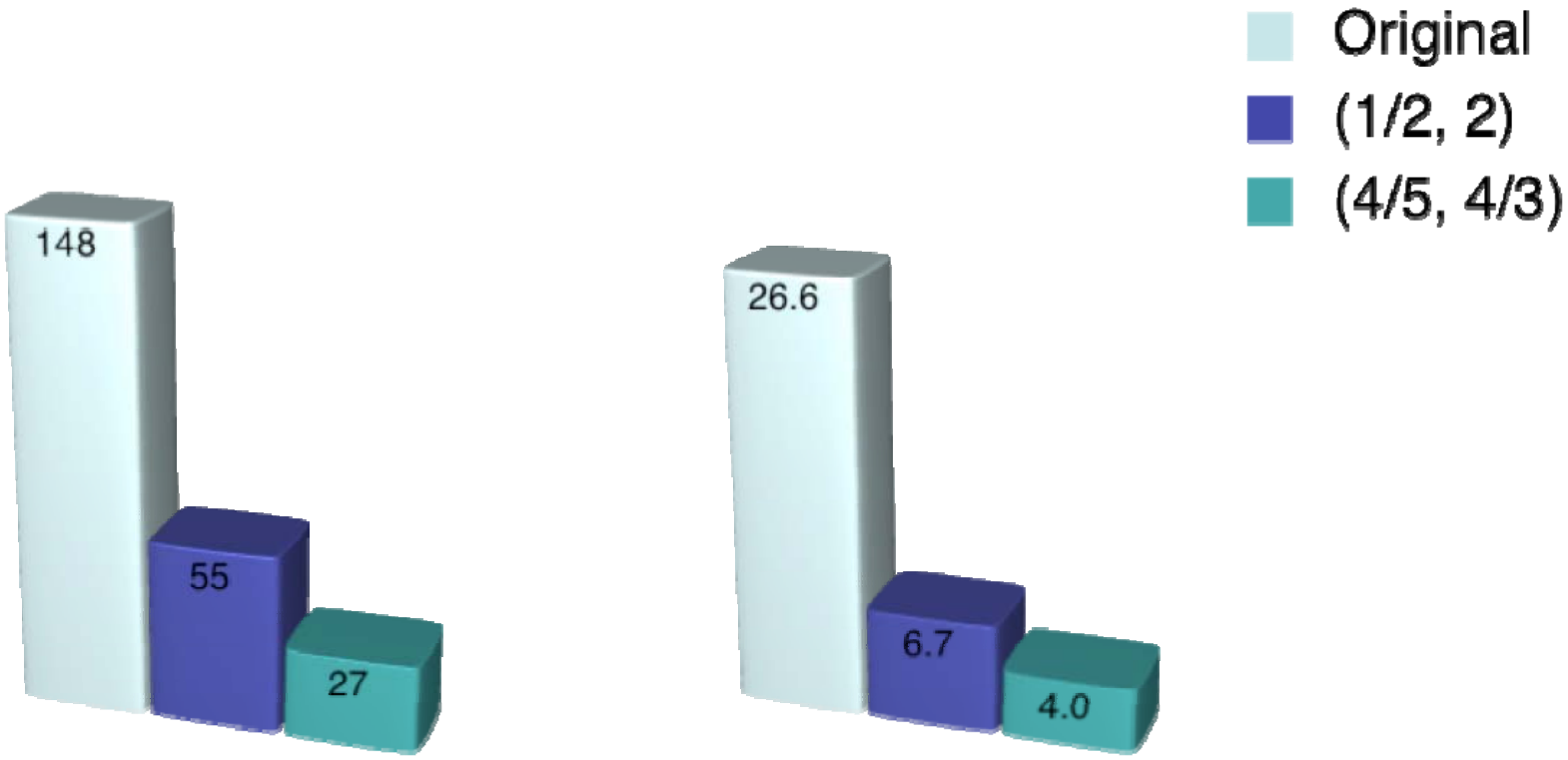


$\left(\frac{1}{2}, 2\right)$



$\left(\frac{4}{5}, \frac{4}{3}\right)$

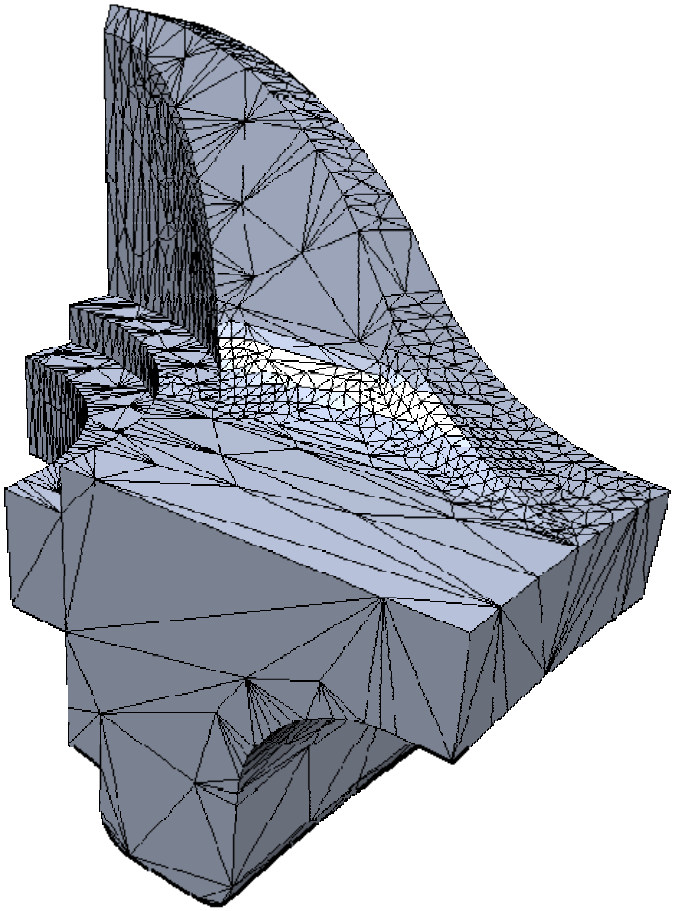
Remeshing Results

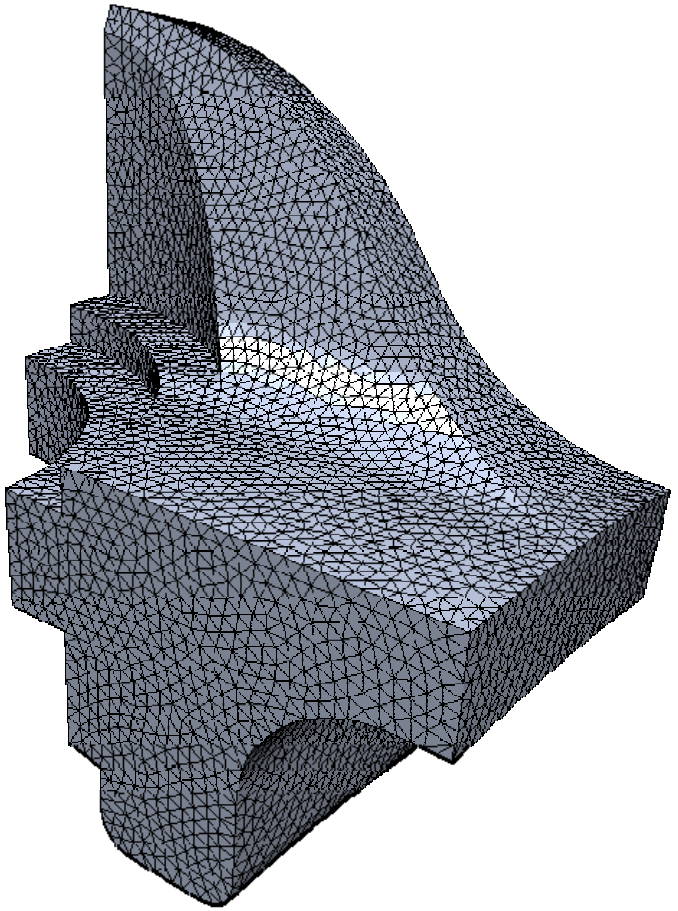


Edge length deviation (%)

Deviation from 60º

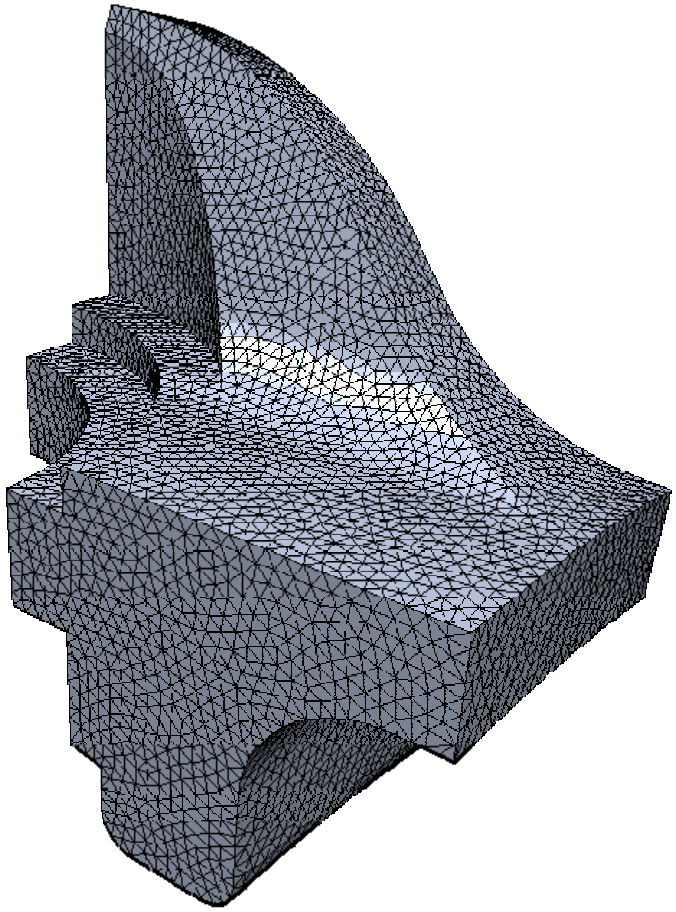
Feature Preservation?





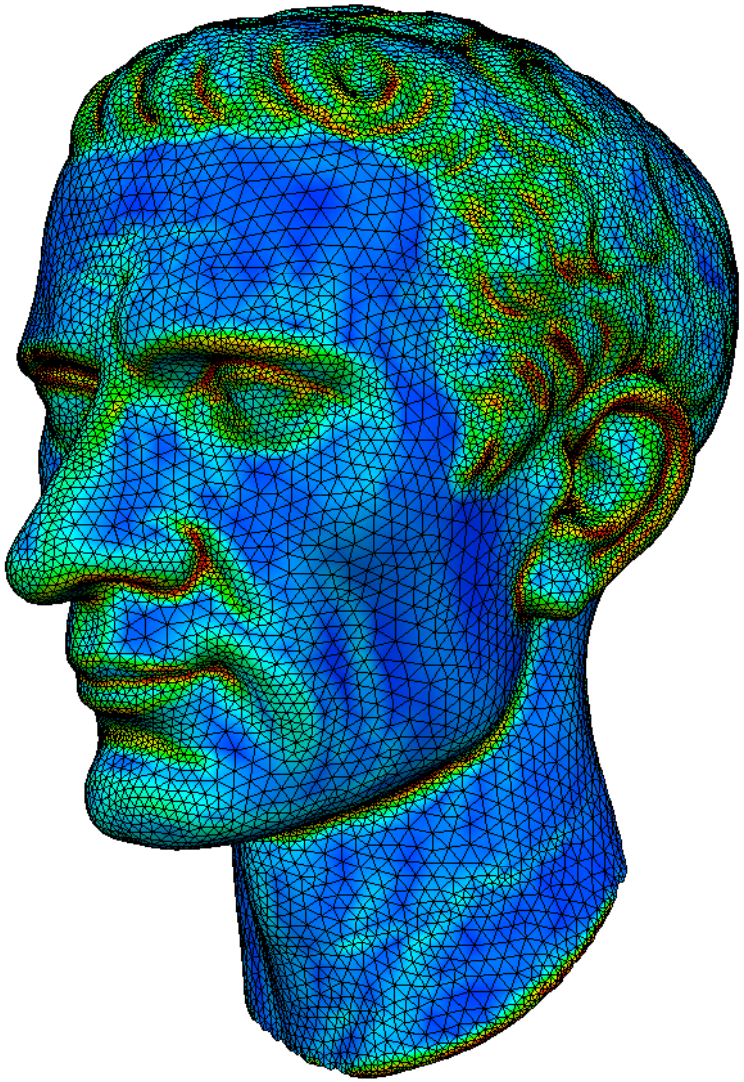
Feature Preservation

- Define features
  - Sharp edges
  - Material boundaries
- Adjust local operators
  - Don’t move corners
  - Collapse only along features
  - Don’t flip feature edges
  - Univariate smoothing
  - Project to feature curves



Adaptive Remeshing

- Precompute max. curvature on reference mesh
- Target edge length locally determined by curvature
- Adjust split / collapse criteria



Next Time:
Quad Dominant Meshing