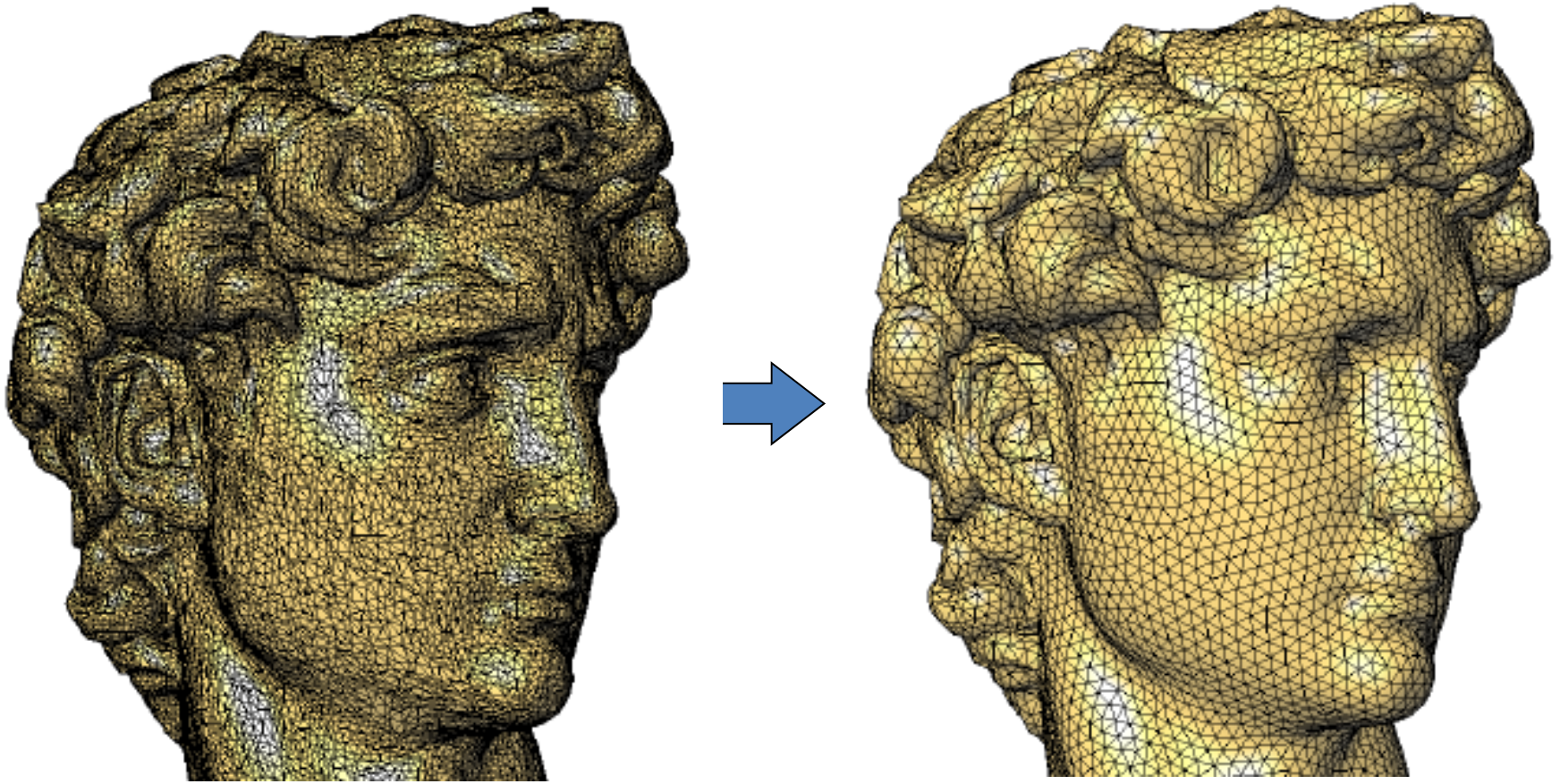
Graphics&Geometric Computing Lab
@USTC

Re-Meshing Surfaces

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

Remeshing

- Generate another mesh for the given mesh



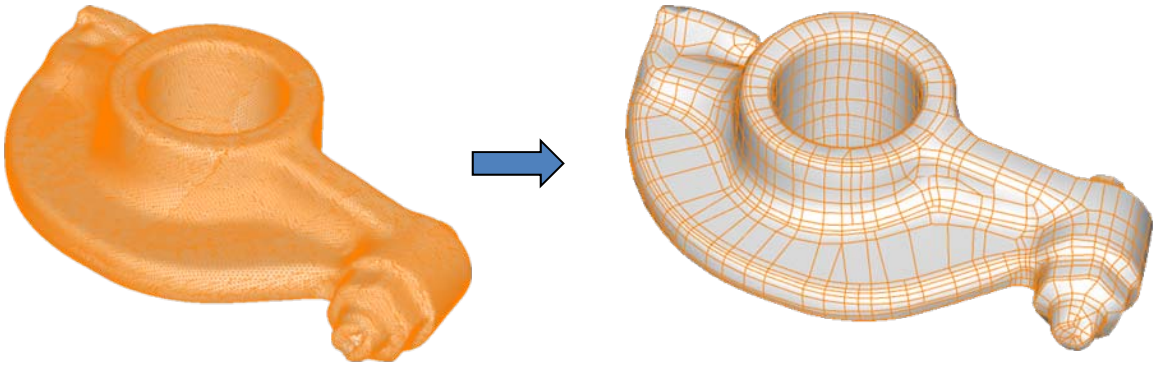
Re-Meshing

- No precise definition
  - Varies according to the targeted goal or application
  - Mesh generation
- Possible definition
  - Given an input mesh, generate another mesh
    - Good element quality
    - Approximating well the input

Motivation

- Computer graphics
  - NURBS patches in CAD/CAM
  - PDEs for fluid, cloth, …
- Finite elements
  - High-quality meshes for simulation

DAZ Productions

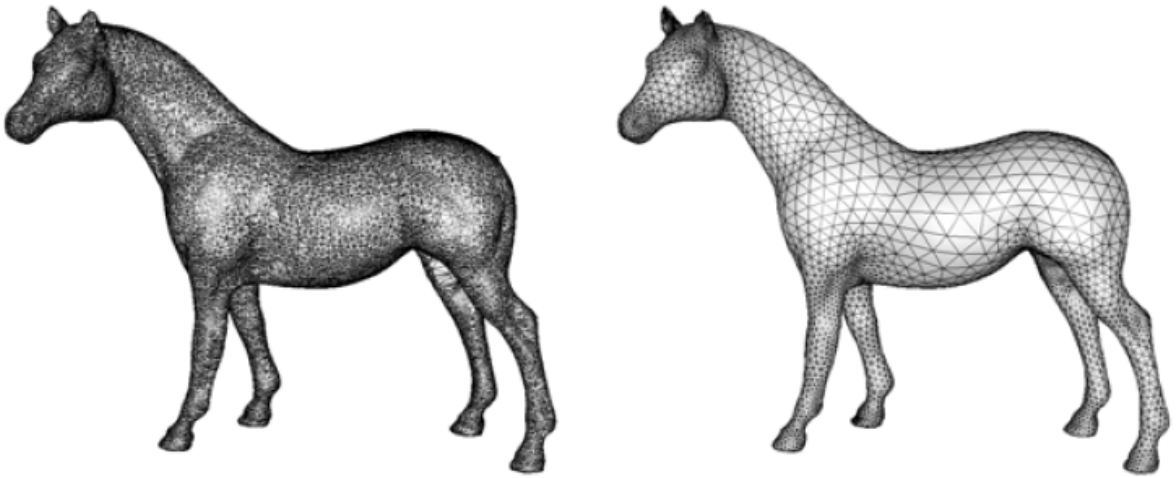


Applications

- Creation and editing
- Animation
- Metamorphosis
- Approximation
- Simulation
- Denoising
- Smoothing and fairing
- Efficient rendering
- Compression
- Feature recovery
- Levels of detail

Quality

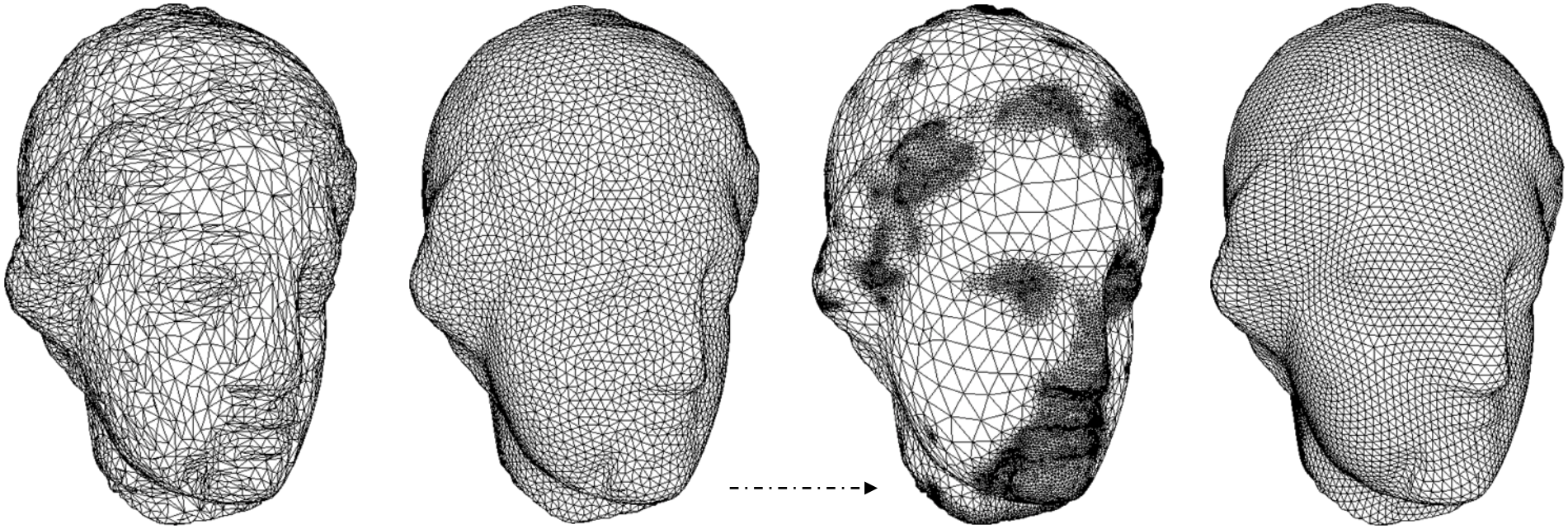
- Some criteria
  - Vertex sampling, grading, regularity, size and shape of elements
- A combination of these criteria



Quality

- Measure “closeness” to equilateral triangle
- Triangle quality measures
  - Ratio of in-radius to circum-radius
  - Smallest angle
  - Ratio of shortest edge to circum-radius

Generality of Remeshing



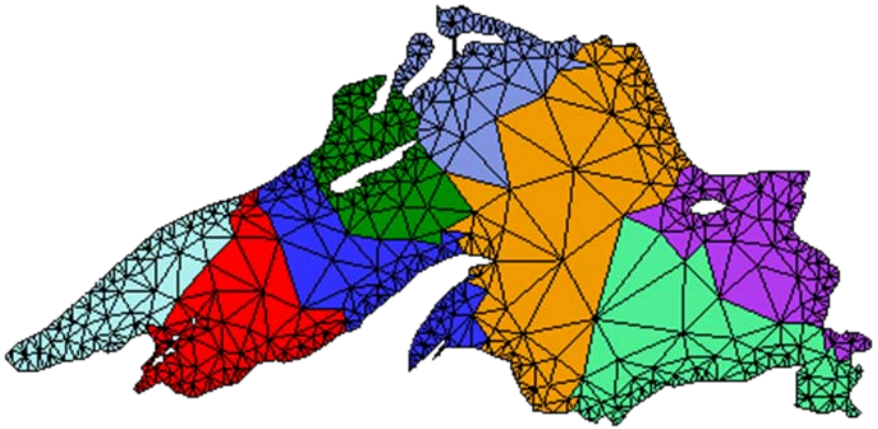
Outline

- Mesh generation in 2D
- Remeshing surfaces
  - Local
  - Global
- Remeshing classifications
  - Structured remeshing
  - Compatible remeshing
  - High quality remeshing
  - Feature remeshing
  - Error-driven remeshing
- Case studies

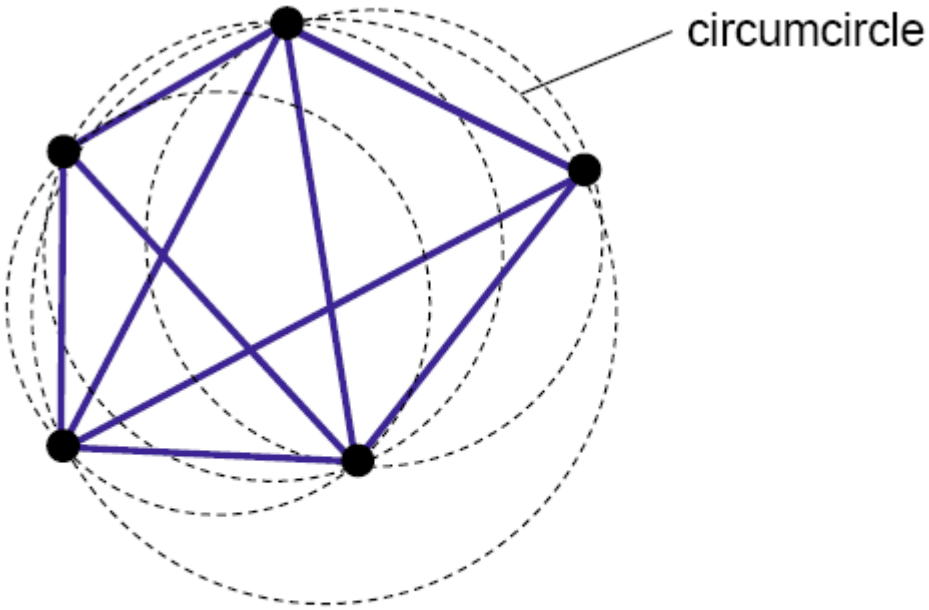
1. Mesh Generation in 2D

Meshing in 2D

- Input
  - Planar polygon
  - Optimal sizing
- Output
  - Triangular mesh
- Motivation
  - 2D problems
  - 3D problems reduced to 2D (parameterization)

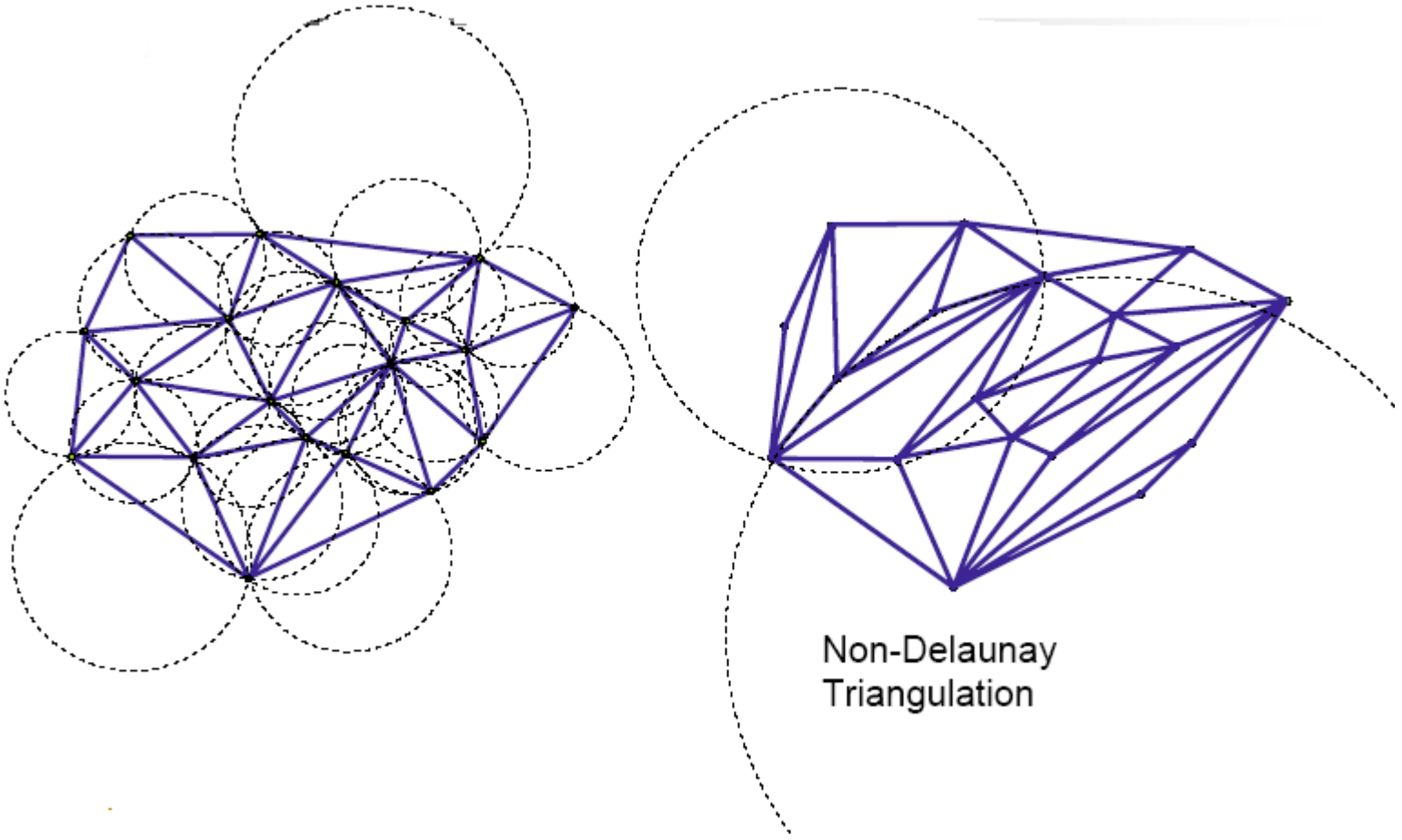


Delaunay Criterion



Empty Circle Property:
No other vertex is contained within the circumcircle of any triangle

Delaunay Triangulation



Delaunay Triangulation

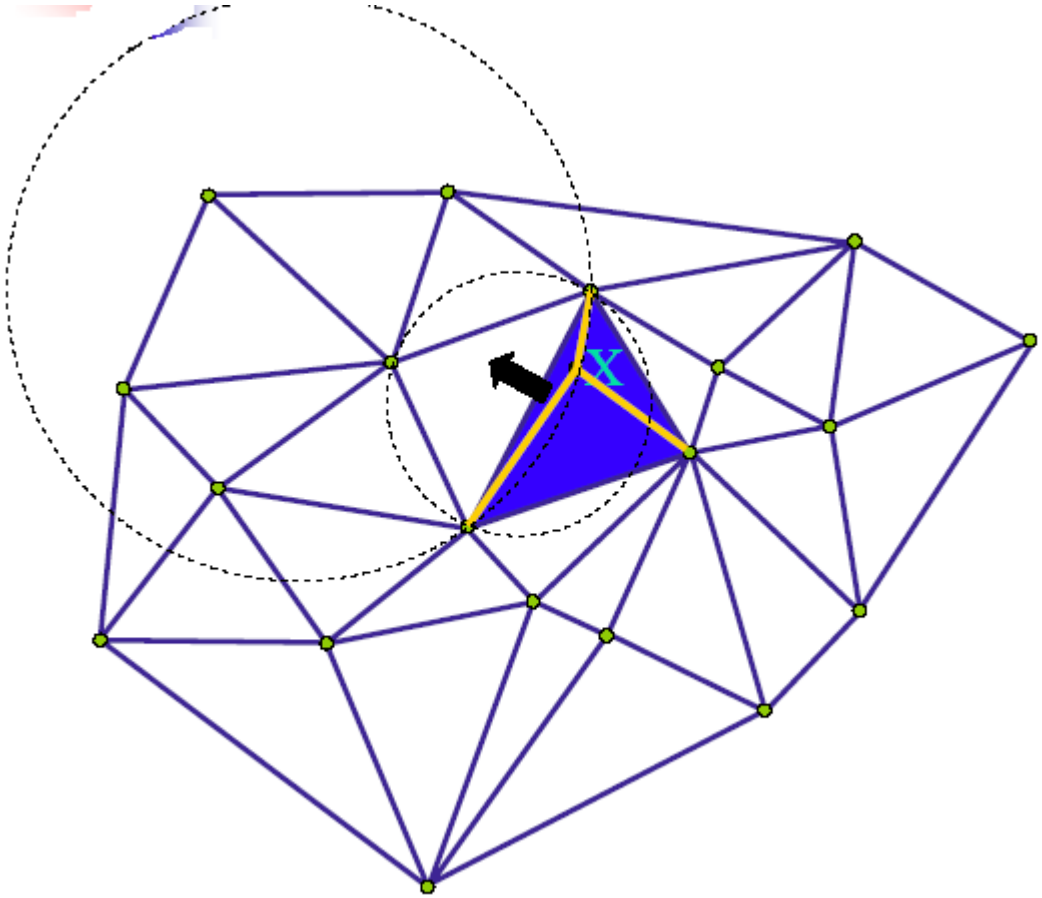
- Obeys empty-circle property
- Exists for any set of vertices
- Is unique (up to degenerate cases)
- Proven to provide best triangles in terms of quality for given vertex positions
- To test – enough to check pairs of triangles sharing common edge

Triangulation Methods

- Edge flip algorithm
  - Start with any triangulation of the vertices
  - Test all edges if satisfy Delaunay criterion
    - test triangles on both sides of edge
  - If edge does not satisfy it, flip edge
  - Repeat until all edges satisfy criterion
- Proven to terminate & give Delaunay mesh
- Slow $O(n^2)$
- Alternative – additive construction
  - Keep Delaunay mesh
  - Add one vertex at a time

Vertex Insertion

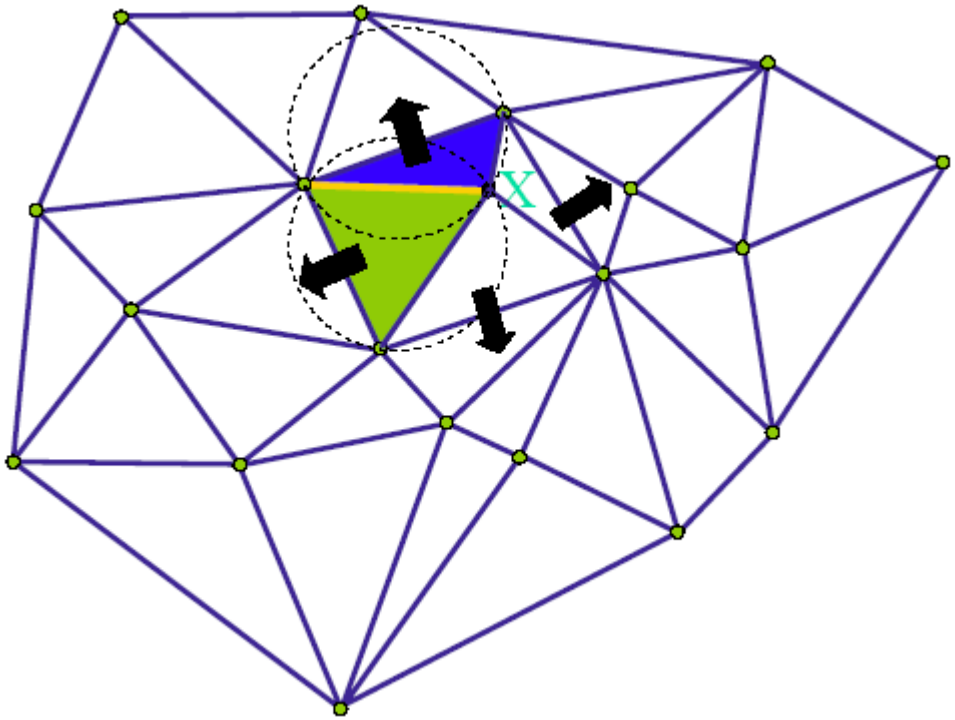
- Locate triangle containing X
- Subdivide triangle



Vertex Insertion

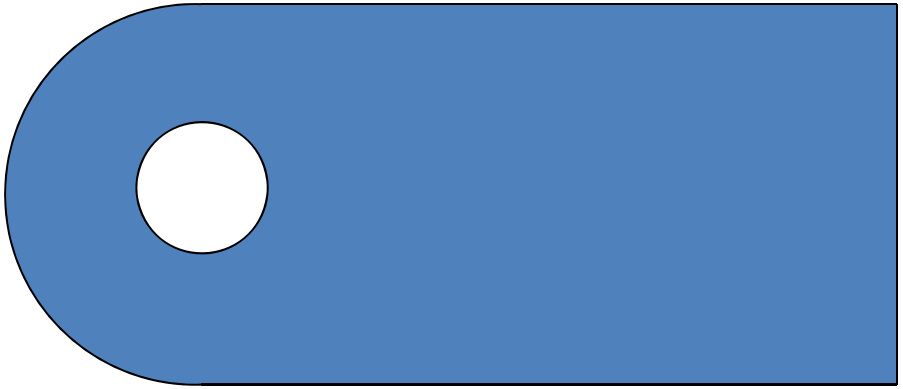
- Locate triangle containing X
- Subdivide triangle

- Recursively check empty-circle property
- Swap diagonal

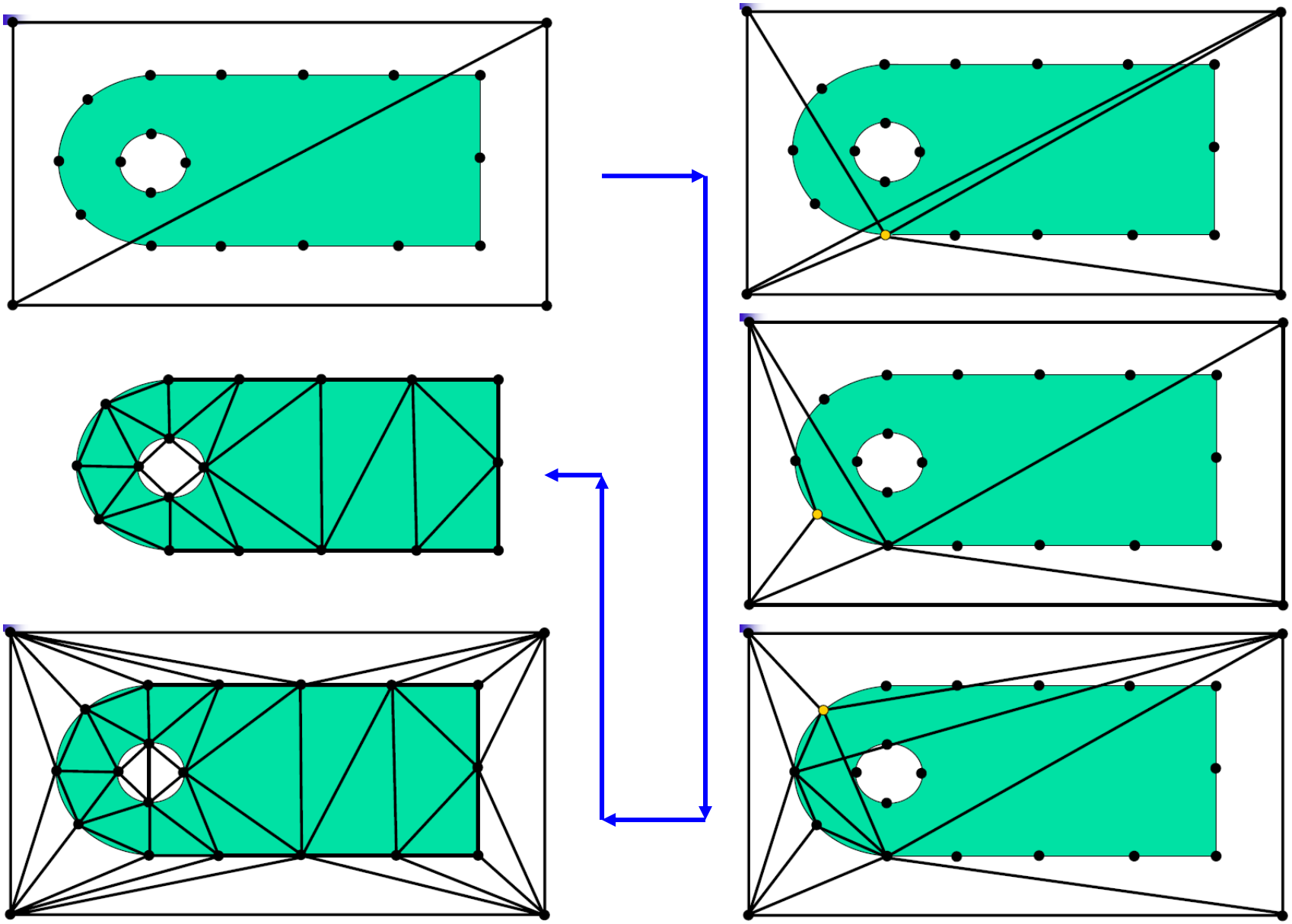


Boundary Insertion

- Place vertices on boundary at cord-length intervals based on sizing
- Create bounding triangles
- Insert vertices using Delaunay method
- Delete outside triangles



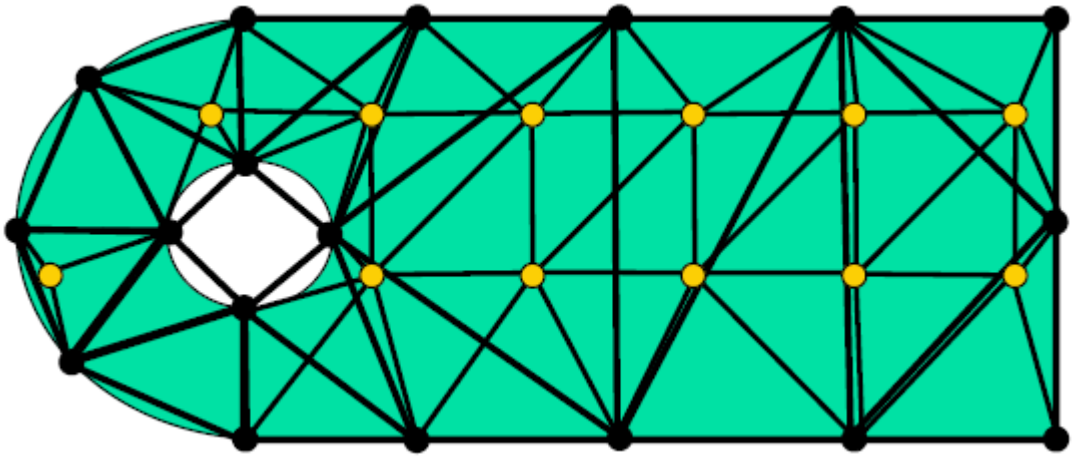
Boundary Insertion



Refinement

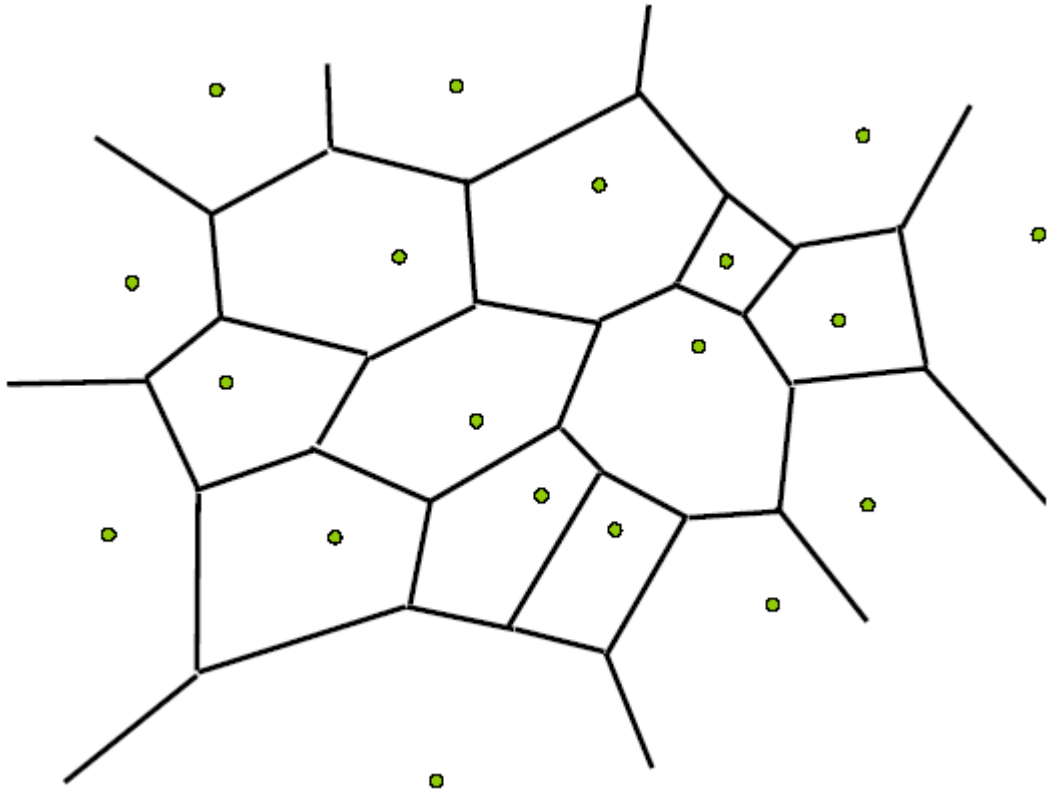
- Edge split
- Edge collapse

- Uniform sampling



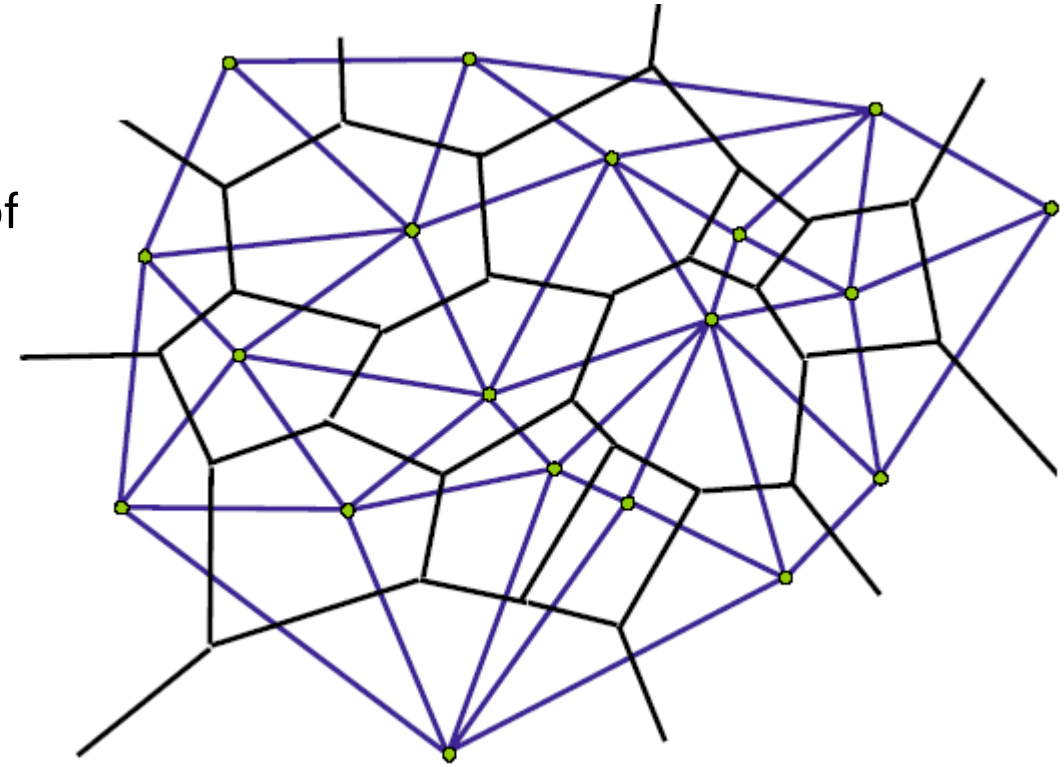
Voronoi Diagram

- Given set of vertices
  - union of all locations at equal distance from two or more vertices



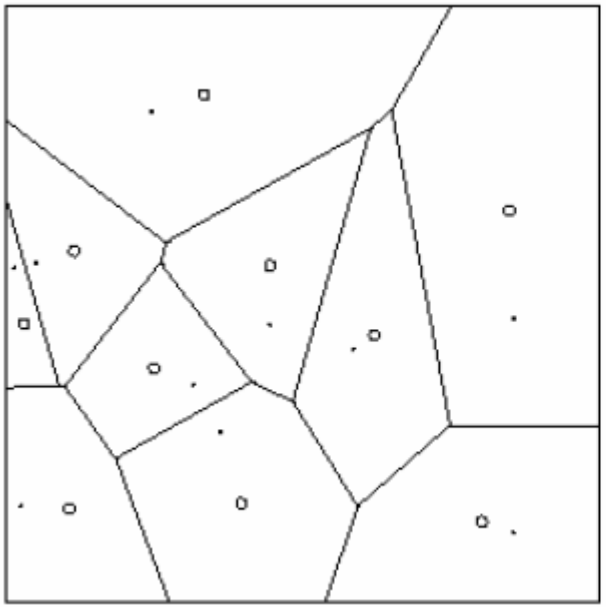
Voronoi Diagram

- Dual to Delaunay Triangulation
  - Vertices correspond to faces
  - Voronoi edges = perpendicular bisectors of Delaunay edges
- Can be constructed directly
- Easier – compute Dealunay & compute dual

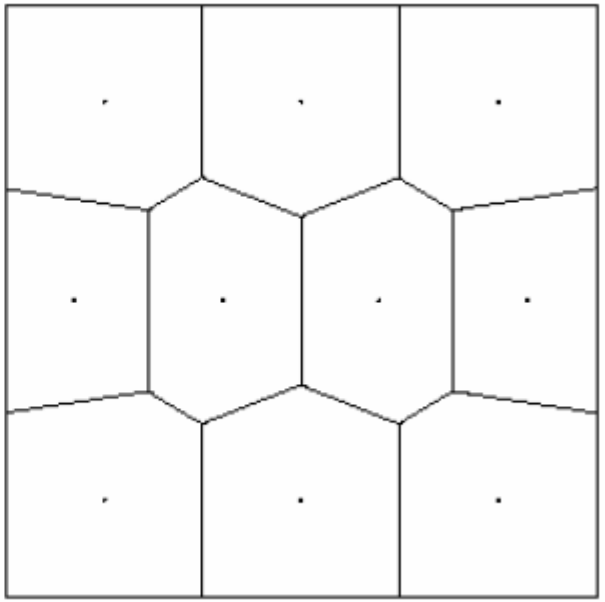


Centroidal Voronoi Diagram

- Vertices coincide with centroids



Ordinary Voronoi diagram



Centroidal Voronoi diagram

Centroidal Voronoi Diagram

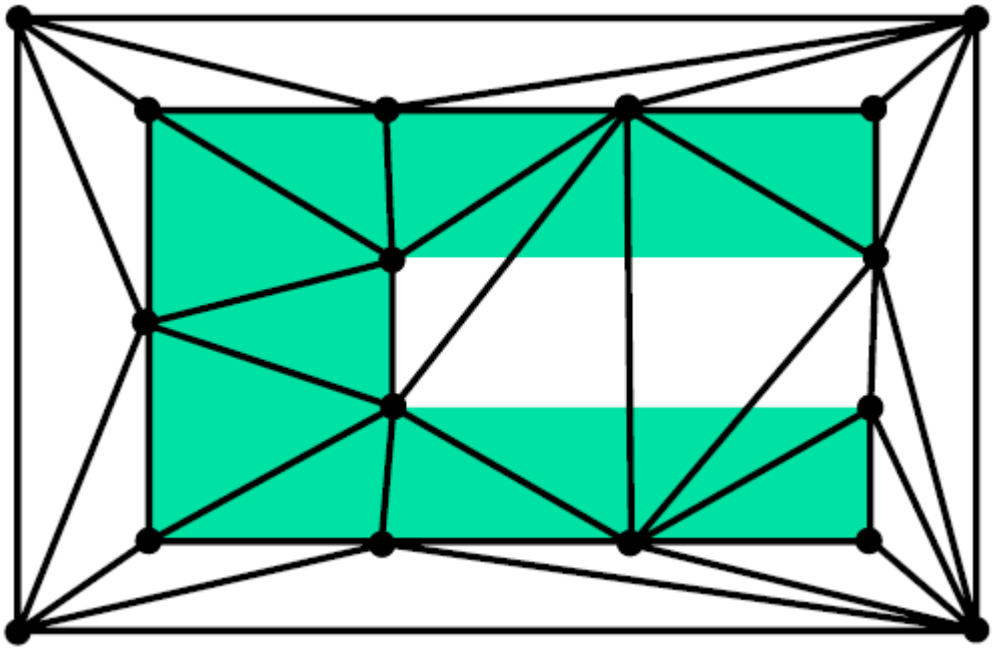
- To compute use Lloyd iterations:
  - Start with set of sites
  - Do
    - Compute VD
    - Compute centers of mass for each Voronoi cell
    - If sites = centers of mass
      - Stop
    - Set sites to centers of mass
    - Repeat
- Guaranteed to converge
- Provides optimal repartitioning of density among vertices

2D Meshing

- Place vertices on boundary
- Use sampling for initial placement inside
- Construct Delaunay triangulation
- Iterate
  - Refinement
  - Coarsening
  - Smoothing
  - Each time perform necessary edge flips

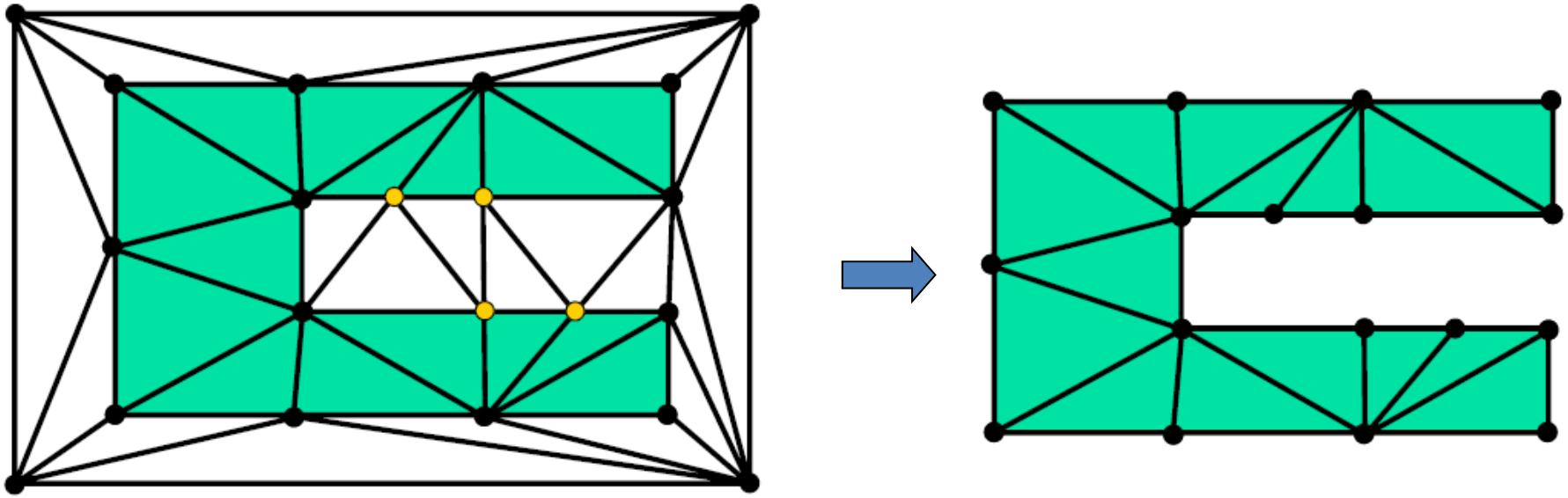
Boundary Recovery

- Delaunay triangulation does not have to obey polygon boundary



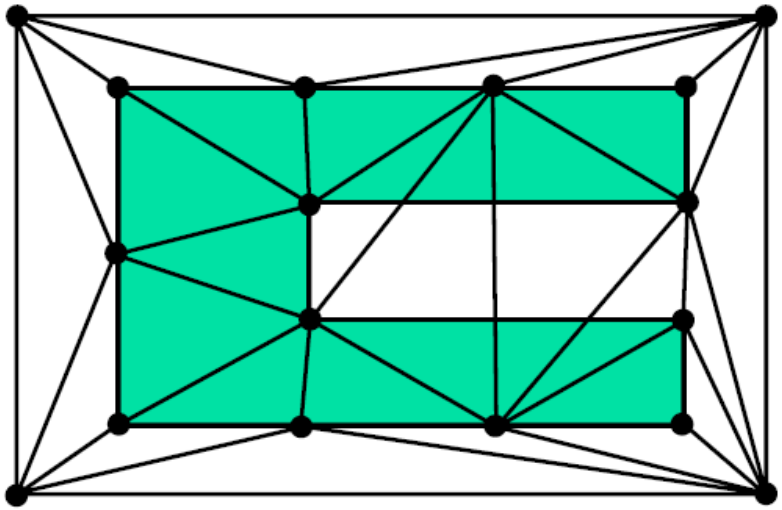
Boundary Conforming Delauanay

- Add vertices at intersections
- Repeat if necessary

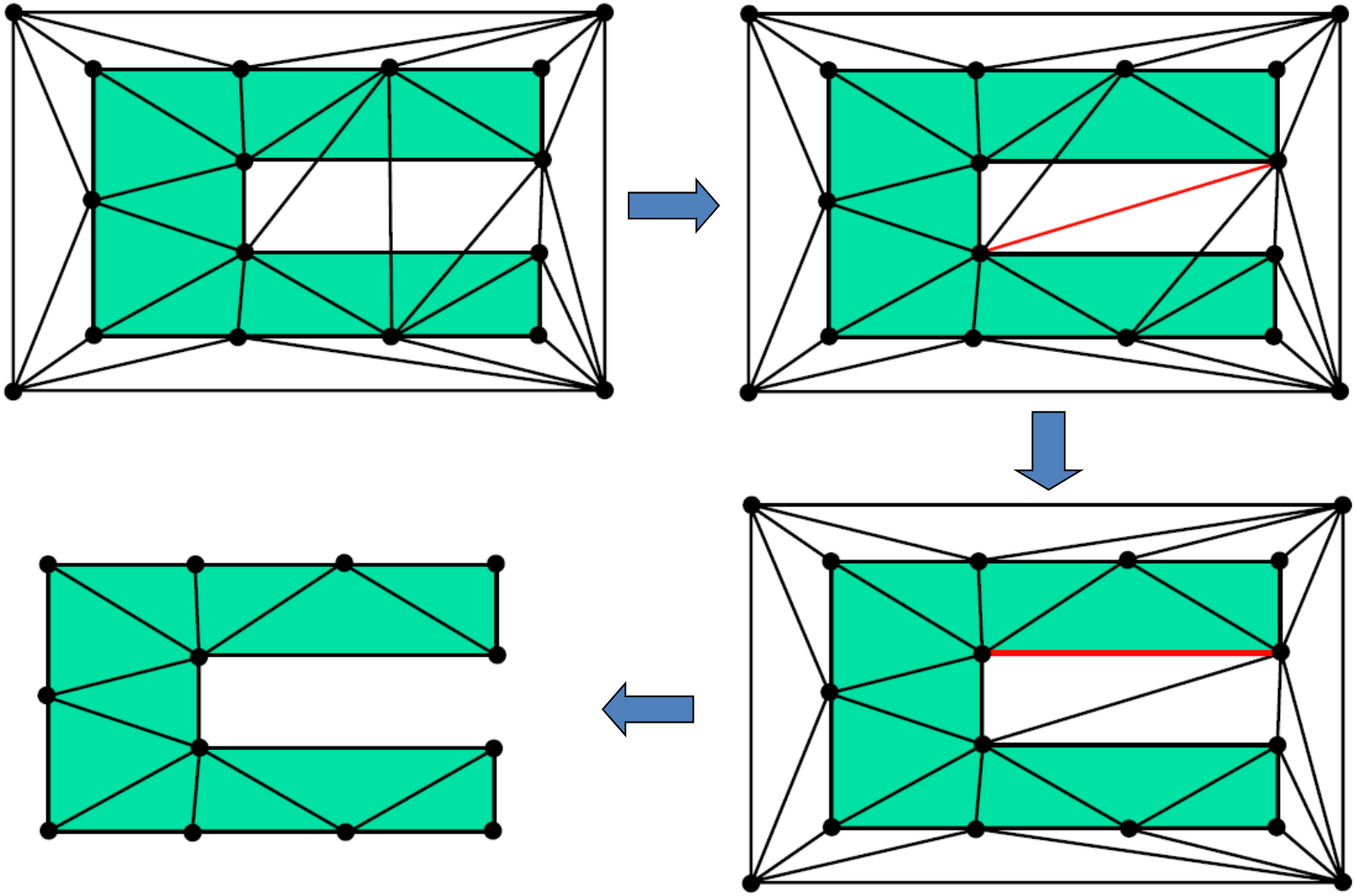


Boundary Recovery - Constrained

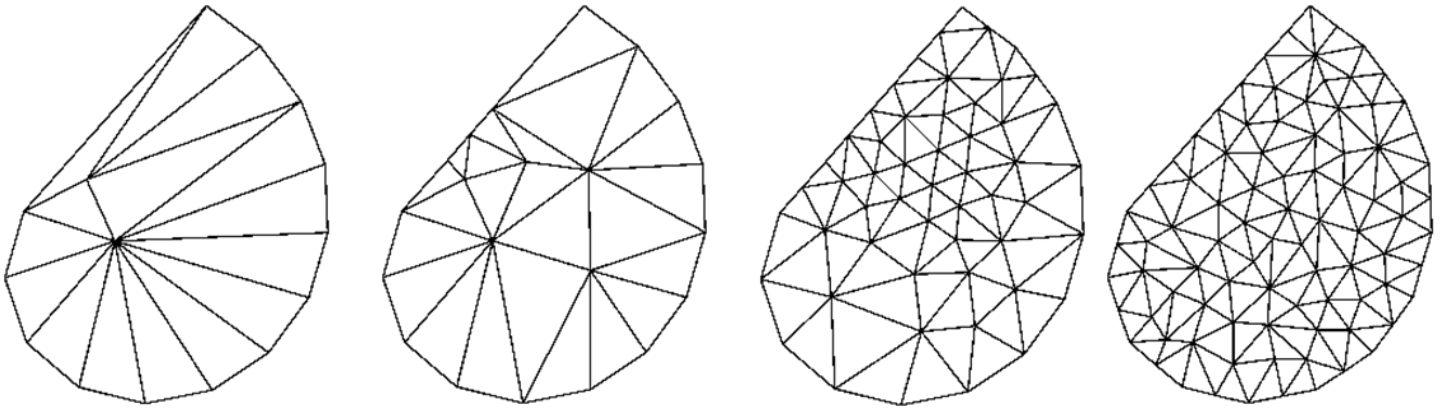
- Not always can add boundary vertices (shared edges)
  - Swap edges between adjacent pairs of triangles
  - Repeat till recover the boundary
- Does not maintain Delaunay criterion !!!

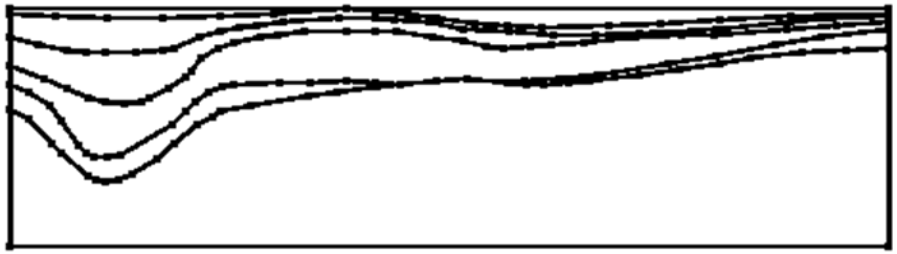


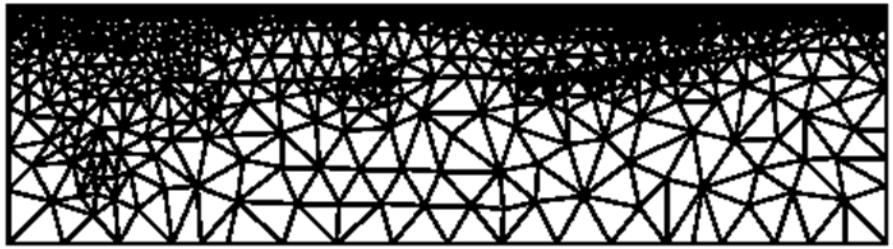
Boundary Recovery - Constrained



Examples

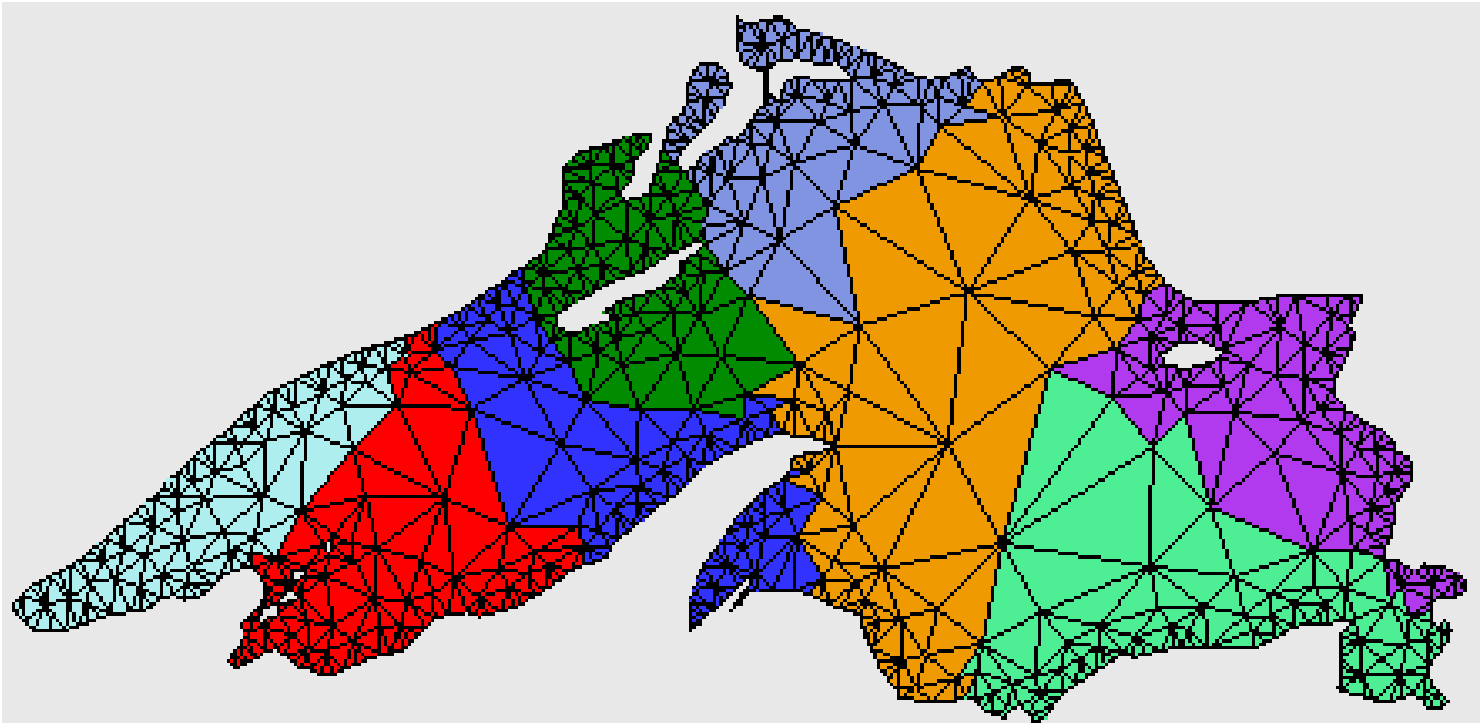






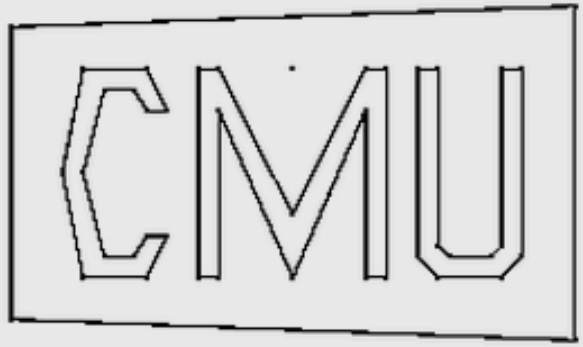
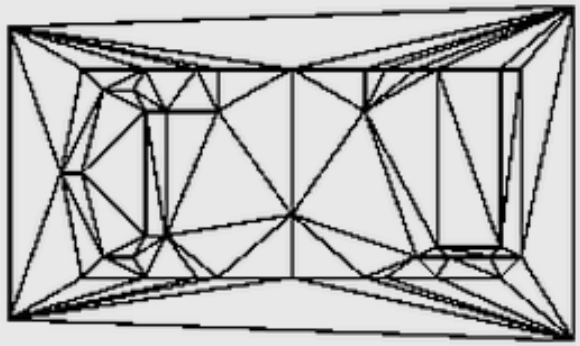
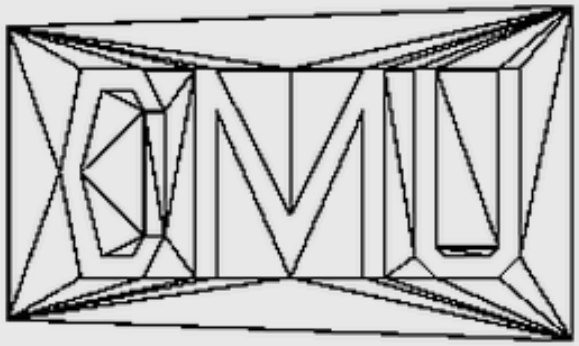
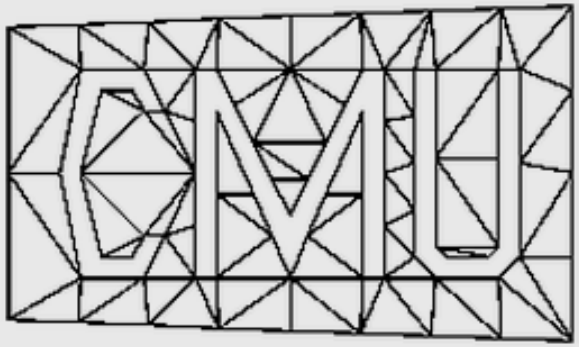
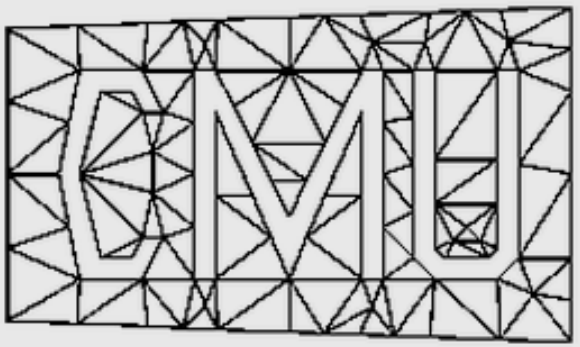
Use the library “Triangle”!

- http://www.cs.cmu.edu/~quake/triangle.html



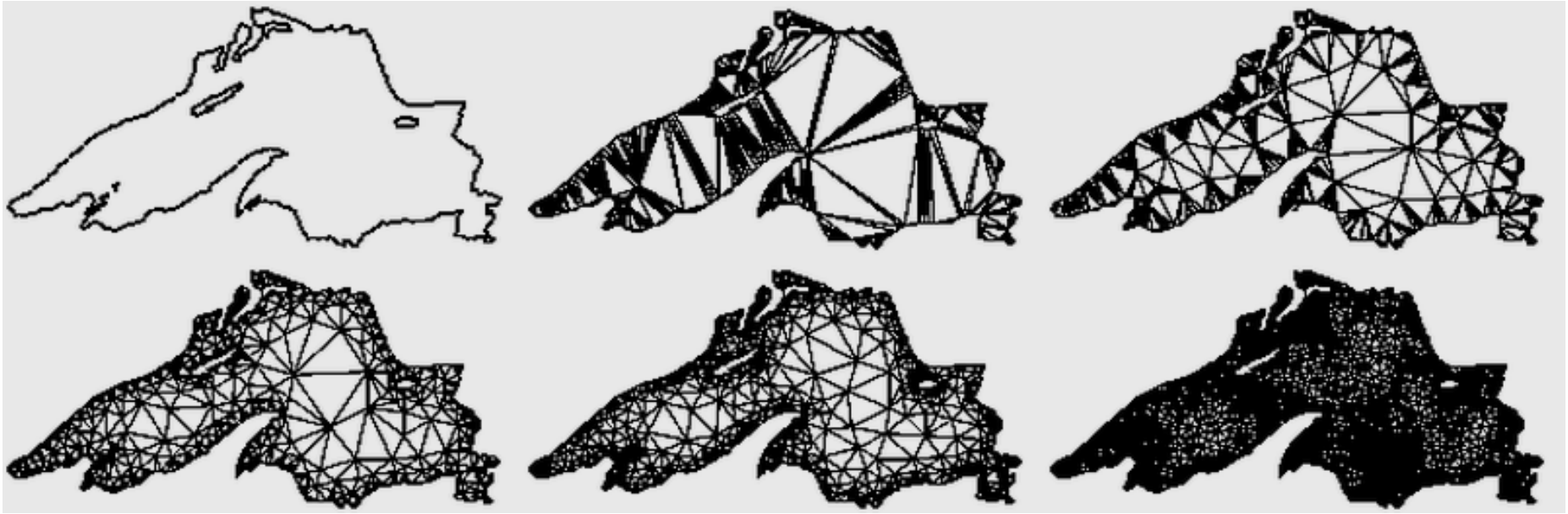
Examples of “Triangle”

- Planar Straight Line Graph (PSLG)
- a Delaunay triangulation of its vertices
- a constrained Delaunay triangulation of the PSLG
- a conforming Delaunay triangulation of the PSLG
- a quality conforming DT of the PSLG with no angle smaller than 25 degree

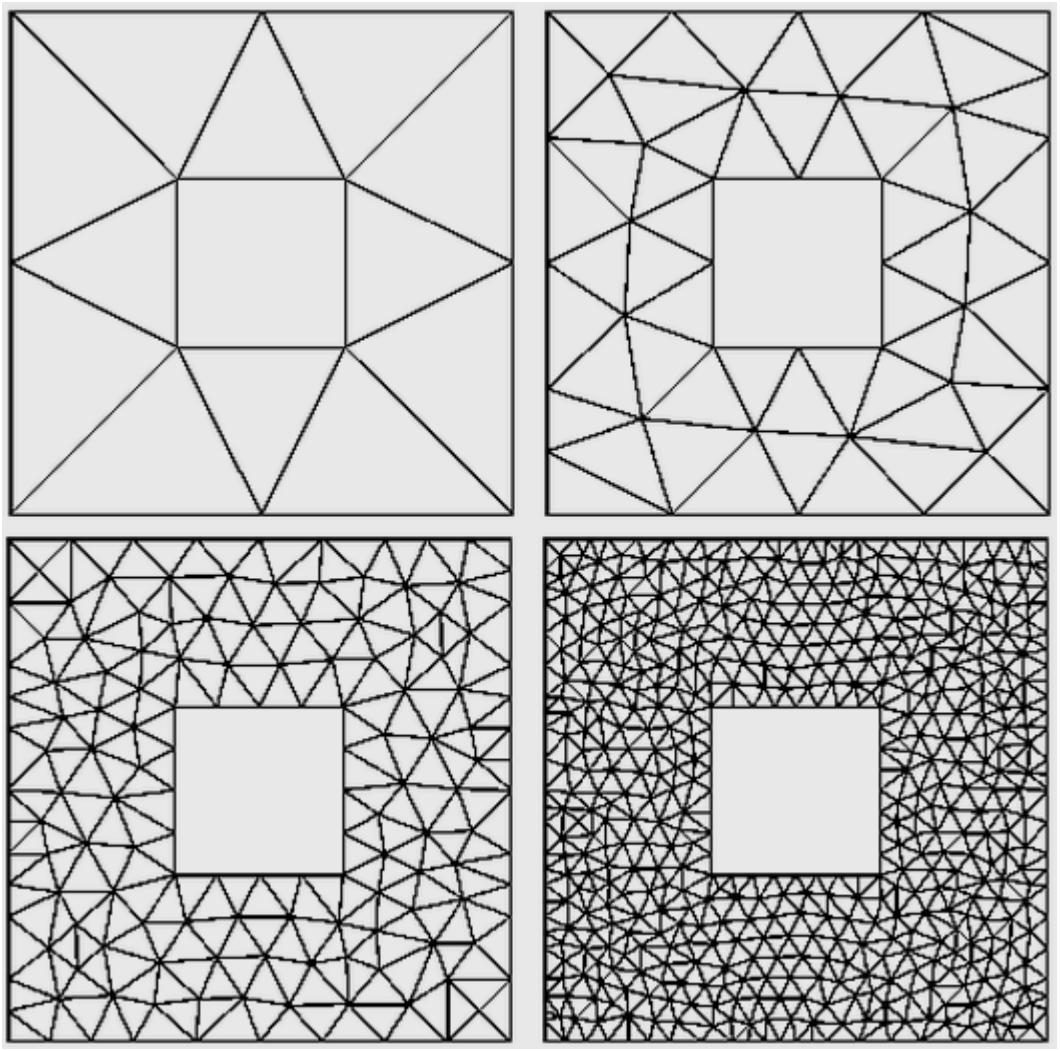
Examples of “Triangle”

- a PSLG of Lake Superior
- triangulations having minimum angles of 0, 5, 15, 25, and 33.8 degrees.



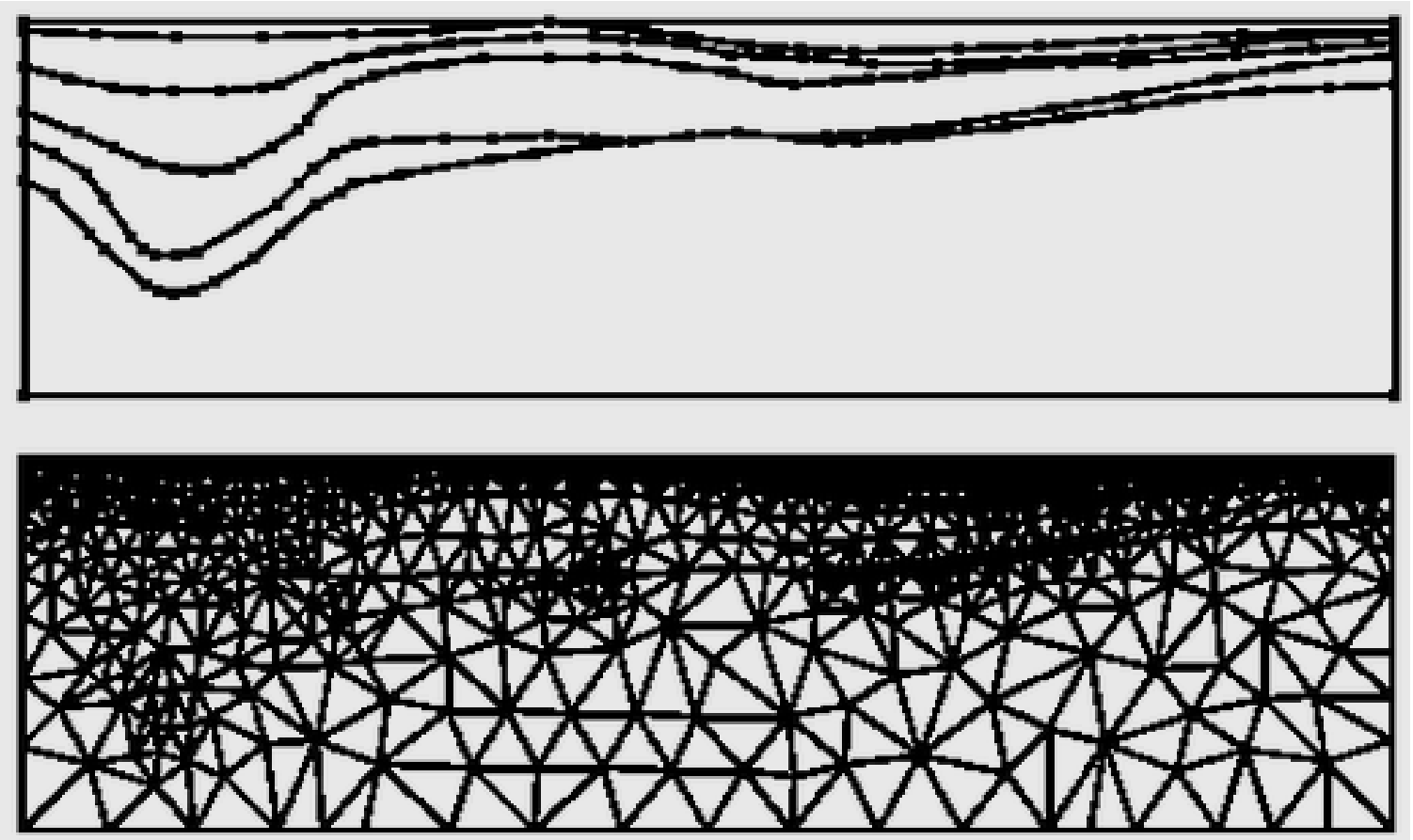
Examples of “Triangle”

- Maximum area constraints



Examples of “Triangle”

- Different maximum triangle area constraints



2. Re-Meshing Surface (Triangular meshes)

Hausdorff Metric

- Given two sets P and Q

$$H_P(Q) = \max_{p \in P} \min_{q \in Q} \|p - q\|$$

$$H(P, Q) = \max(H_P(Q), H_Q(P))$$

- Approximate by measuring distance from vertices
  - For each vertex on new mesh measure distance to old surface
  - For each vertex on old mesh measure distance to new surface
- Another approximation (upper bound)
  - Measure distance from new/old vertices to nearest old/new vertex

Approaches

- Local – perform sequence of operations to improve existing mesh
  - Use local parameterization
  - Mesh optimization
- Global – 2D reduction
  - Segment surface into parameterizable pieces
  - Parameterize in 2D
  - Mesh in 2D (Delaunay)
  - Project back

2.1 Local Remeshing

Mesh optimization

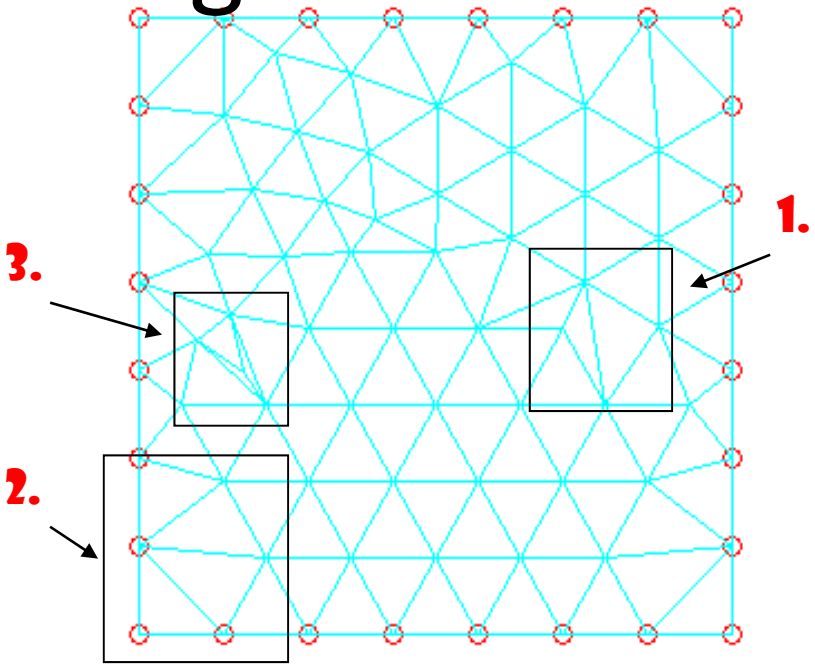
Local Remeshing

- Perform sequence of operations to improve existing mesh
- Connectivity modification
  - Edge swap/split
  - Vertex insertion/removal
- Geometry modification
  - Mesh smoothing
- Repeat operations

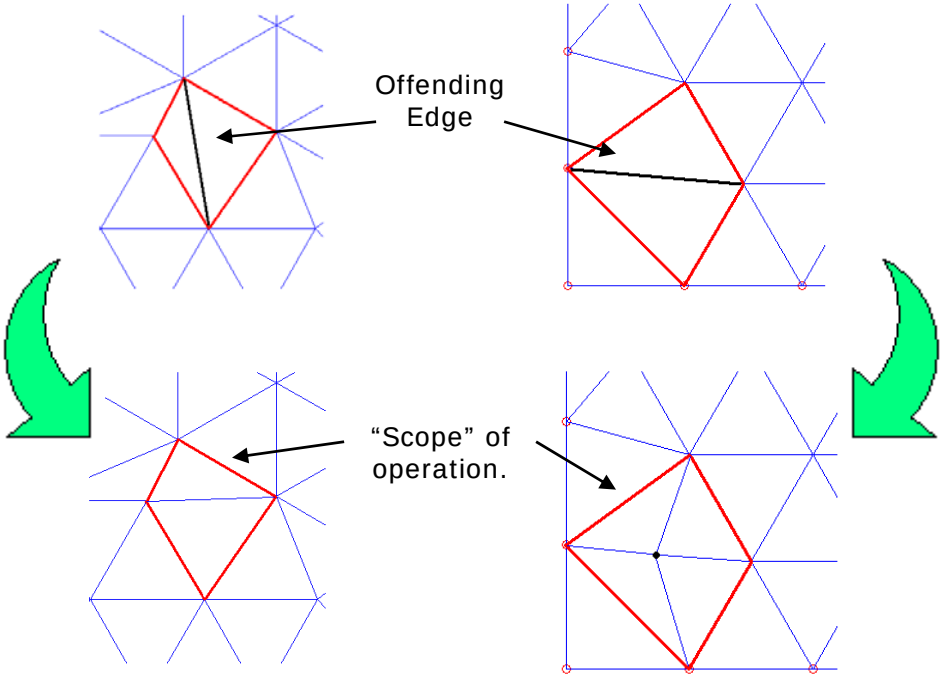
Local Remeshing

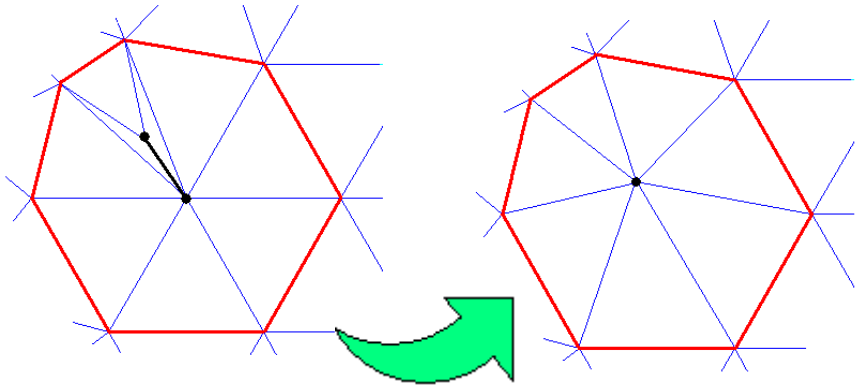
- Topology changes
- Local correction strategies

Algorithms



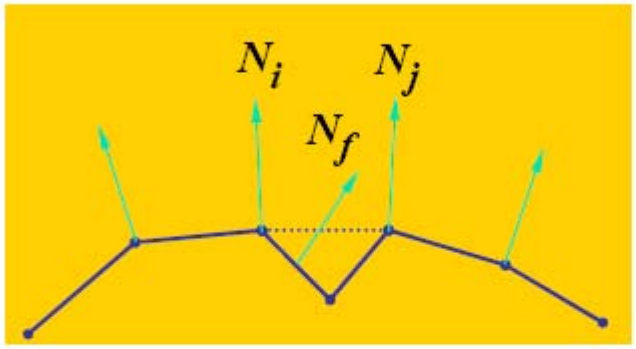
1. **Flip** an edge.
2. **Split** an edge.
3. **Collapse** an edge.

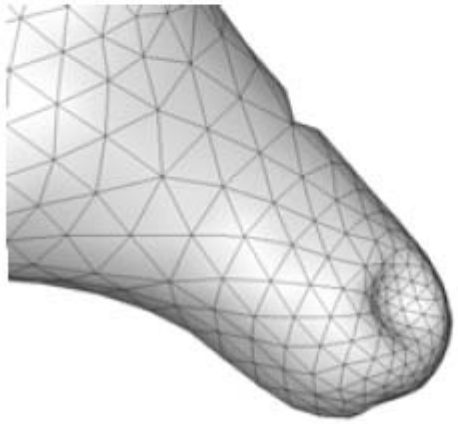


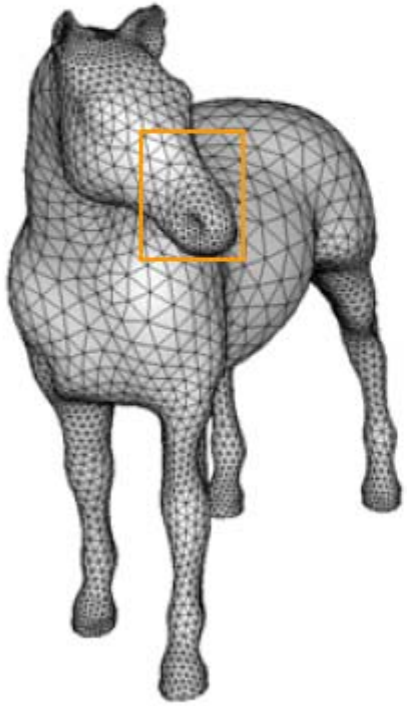


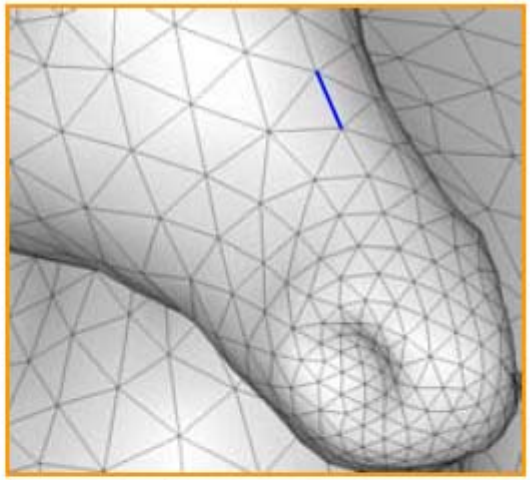
Normal Error

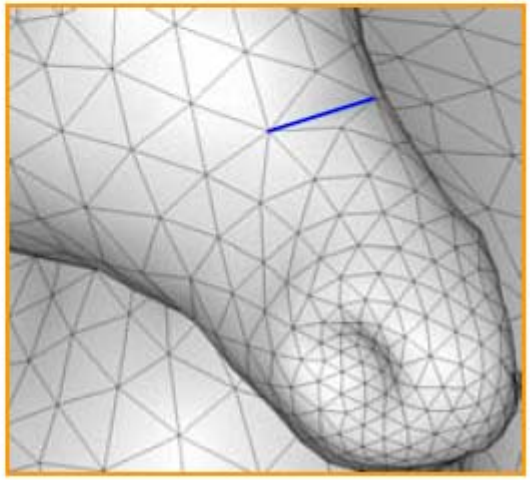
$$E_{smth}(f) = \max_{i \in \{1,2,3\}} \langle N_f, N_{v_i} \rangle < \cos \theta_{smth}$$





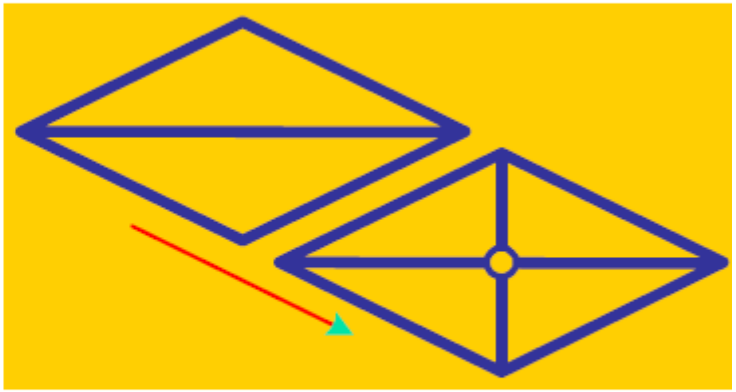






Local Remeshing: Enrichment

- Add vertices – reach desired sizing or element count
- Strategies
  - Long edge split – insert mid-points
  - Other: random, face split (circumcenter)
- Find vertex locations
  - Location on current mesh
  - Project to **original** mesh
  - For better accuracy/smoothness use approximate surface

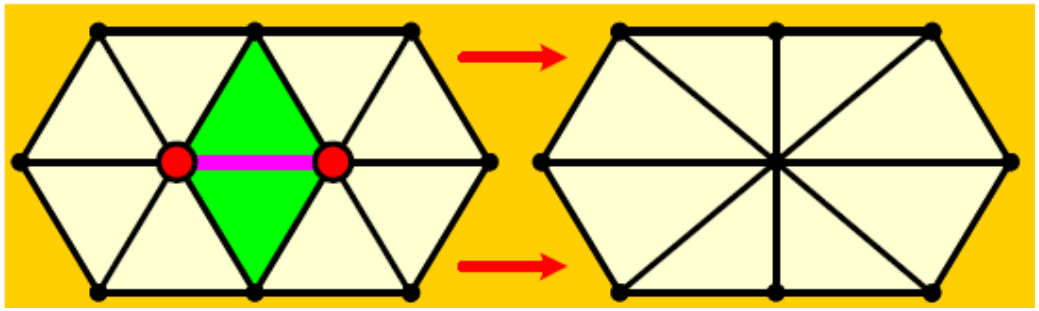


Hard to achieve good spacing

- Improve by smoothing

Local Remeshing: Simplification

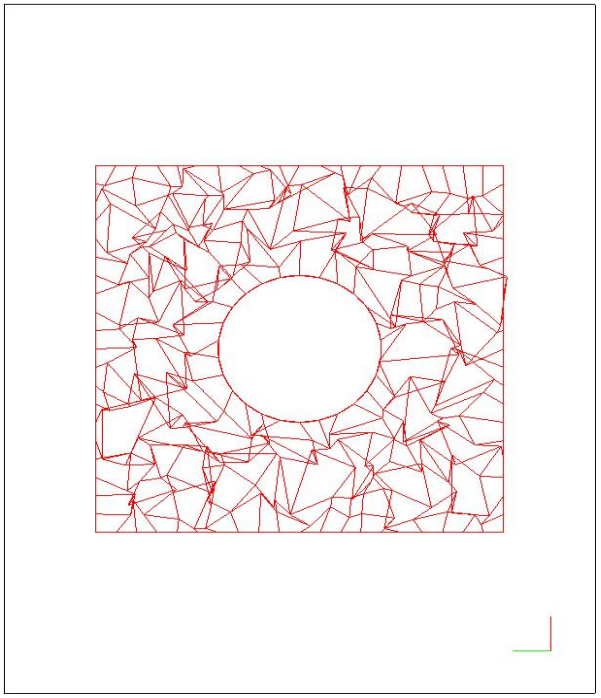
- Remove – similar to simplification
  - Vertex removal – remove vertex & triangulate generated cavity
  - Edge collapse
    - Project new vertex to original surface as in enrichment
- Error metrics to preserve original surface shape + sizing specs
  - Same as in simplification - quadrics
  - Or normal based

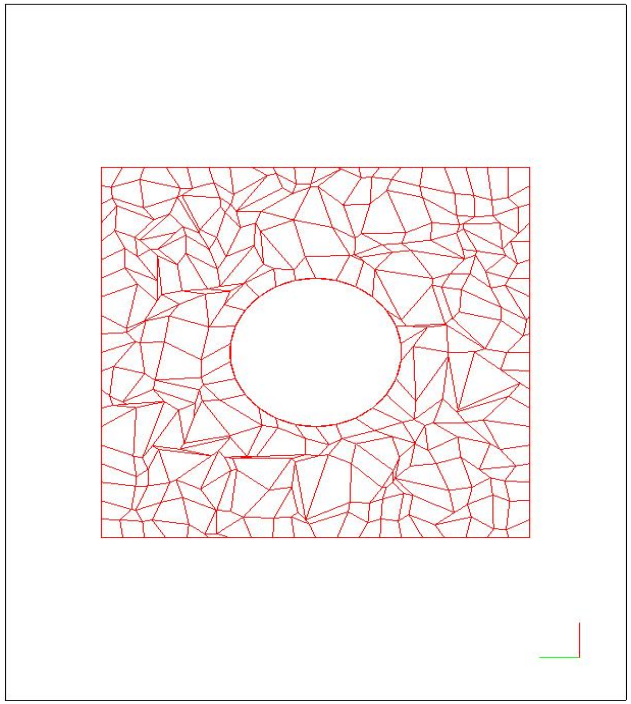


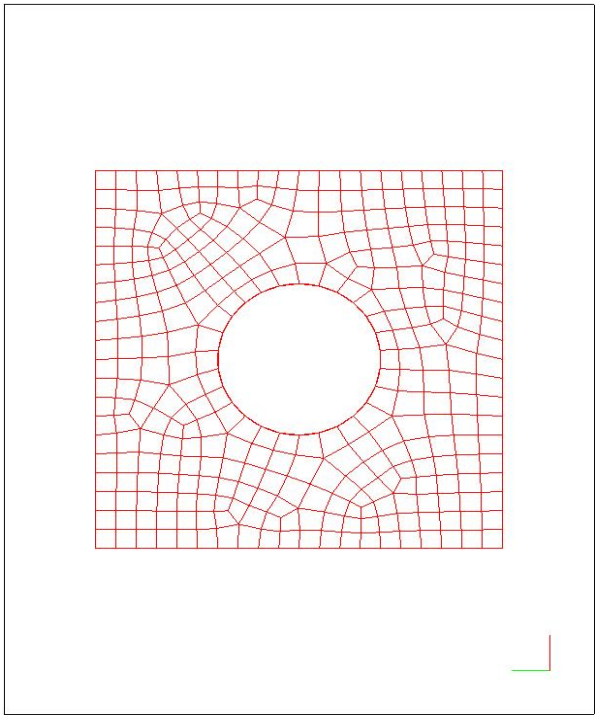
Local Remeshing: Smoothing

- Move vertices ON surface to improve sizing/quality
- Relocation:
  - Compute vertex location as function of neighbors in new mesh
    - E.g. weighted average (similar weights as in parameterization)
    - Compute in
      - Directly in 3D
      - Or in 2D using local parameterization of new mesh
  - Project to original mesh (approximate surface)
    - same as for enrichment
  - **Check error**
    - If too large, keep previous location

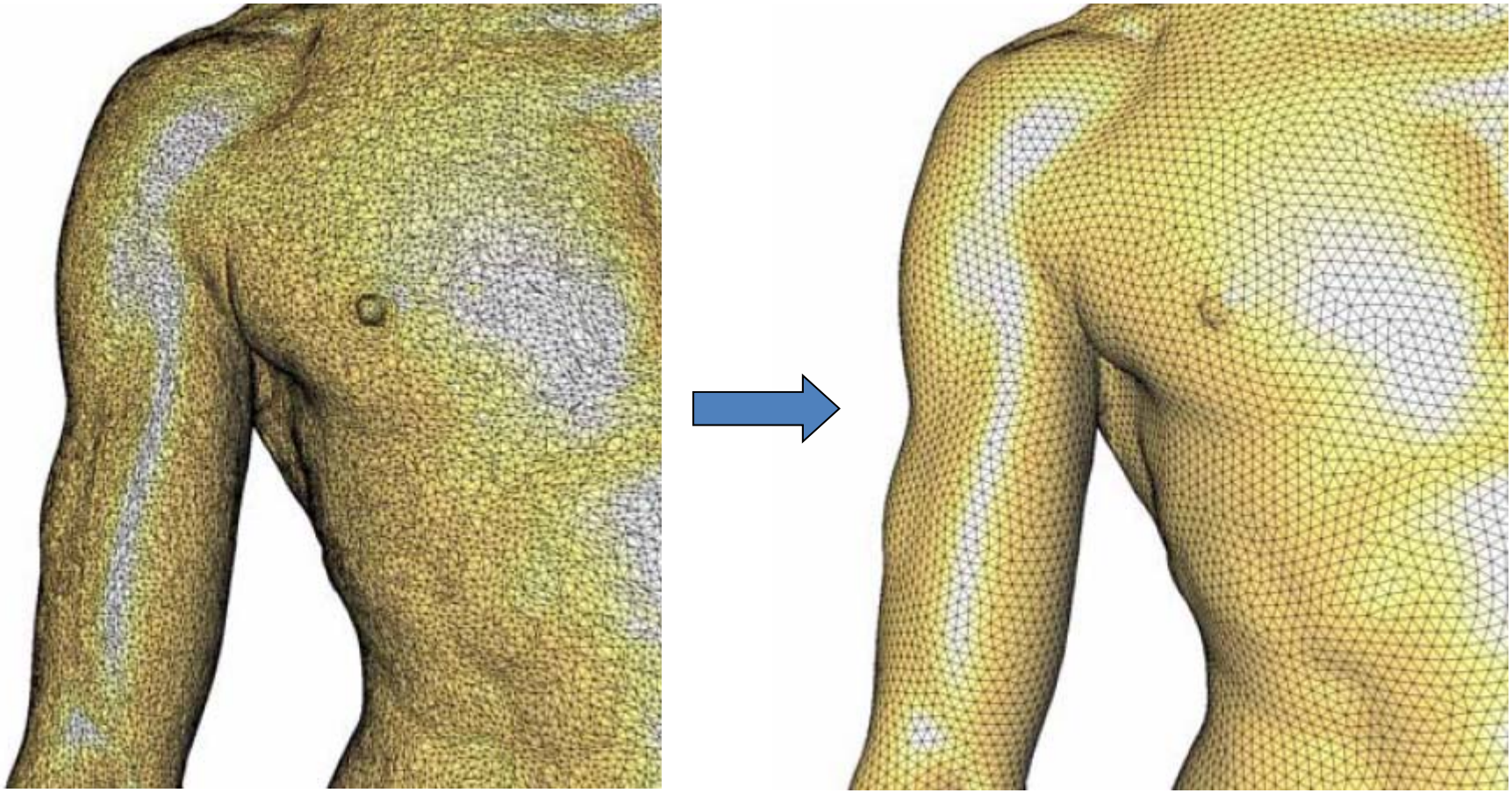
Example







Example



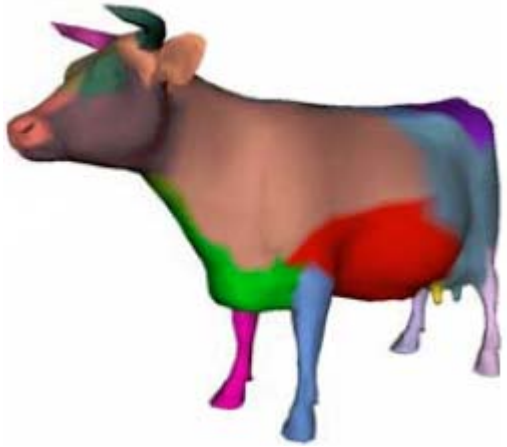
Local Methods Properties

- Preserve topology
- Fast
- Simple to implement
  - Depending on choice of local operations
- Hard to place distance/quality bounds
- Hard to find GOOD spacing of vertices
  - WCVD does the trick but at a cost…
- Heuristic
  - How many iterations of each operation to do?
- **Bag of tricks….**

2.2 Global Remeshing

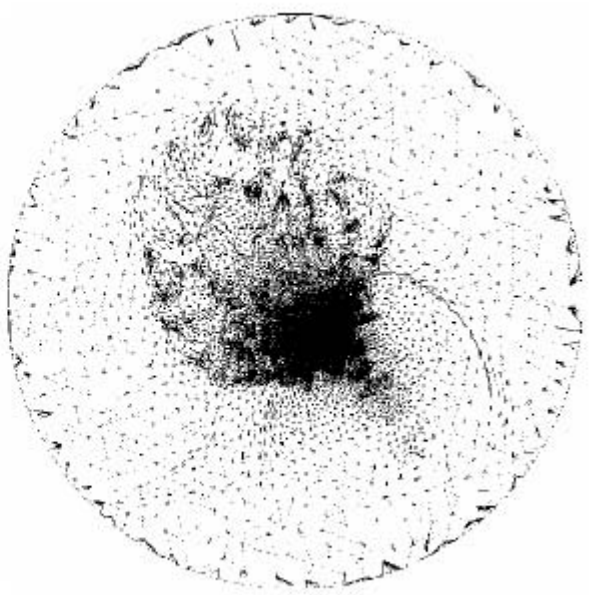
Global: Reduction to 2D

- Segment surface into parameterizable patches
- Parameterize each patch in 2D
  - Parametric distortion determines 3D mesh quality
- Mesh patches in 2D (Delaunay)
  - Take parametric distortion into account (sizing)
  - Take care of shared boundaries
- Project back



Parameterization

- Fixed boundary approach
- Boundary free approach



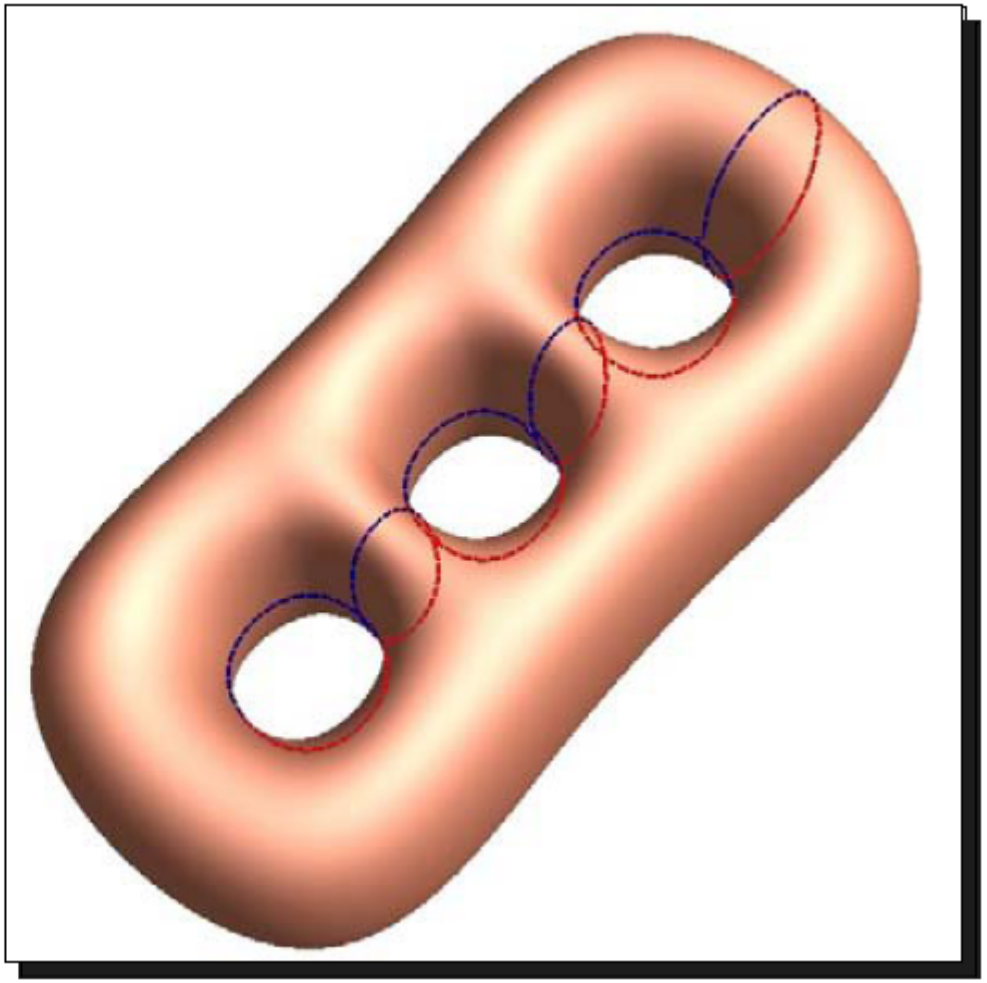
Segmentation

- Make parameterizable patches
  - Cut closed & high genus surfaces
- Less distortion (curvature) simplifies meshing
- Fixed boundary parameterization – should segment into convex patches

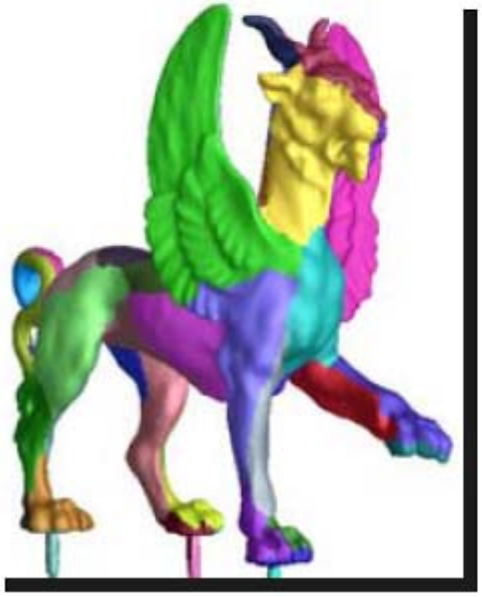
Will be discussed in detail in class “Mesh segmentation”

Genus>0

- Finding shortest cut NP-hard
- Use heuristics
- Algorithm
  - Start from seed face
  - Propagate face front (BFS)
    - Each time front meets itself (at edge e)
      - Add e to Cut Path
  - Prune Cut Path

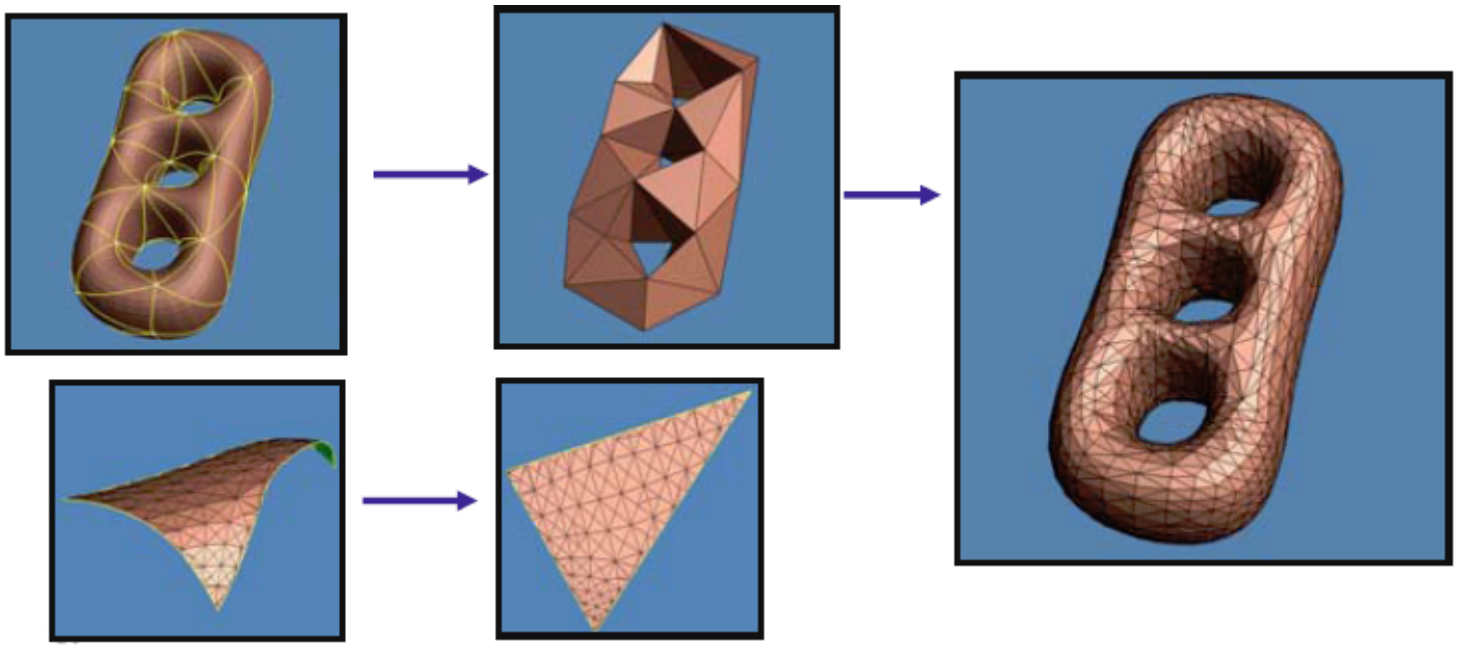


Examples



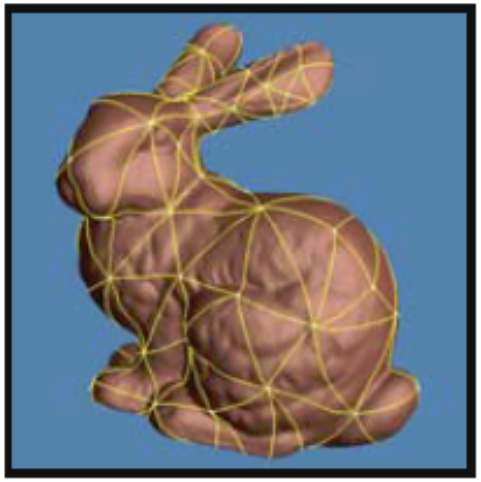
Segment Based Remeshing

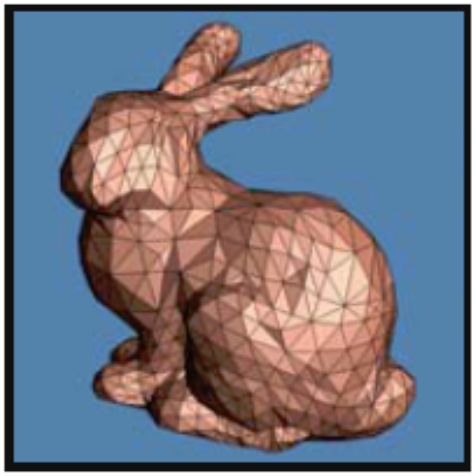
- Map each triangular patch to corresponding triangle
  - Use harmonic/mean-value weights
- Mesh each triangle using subdivision connectivity
  - Guarantees conformity



Segment Based Remeshing

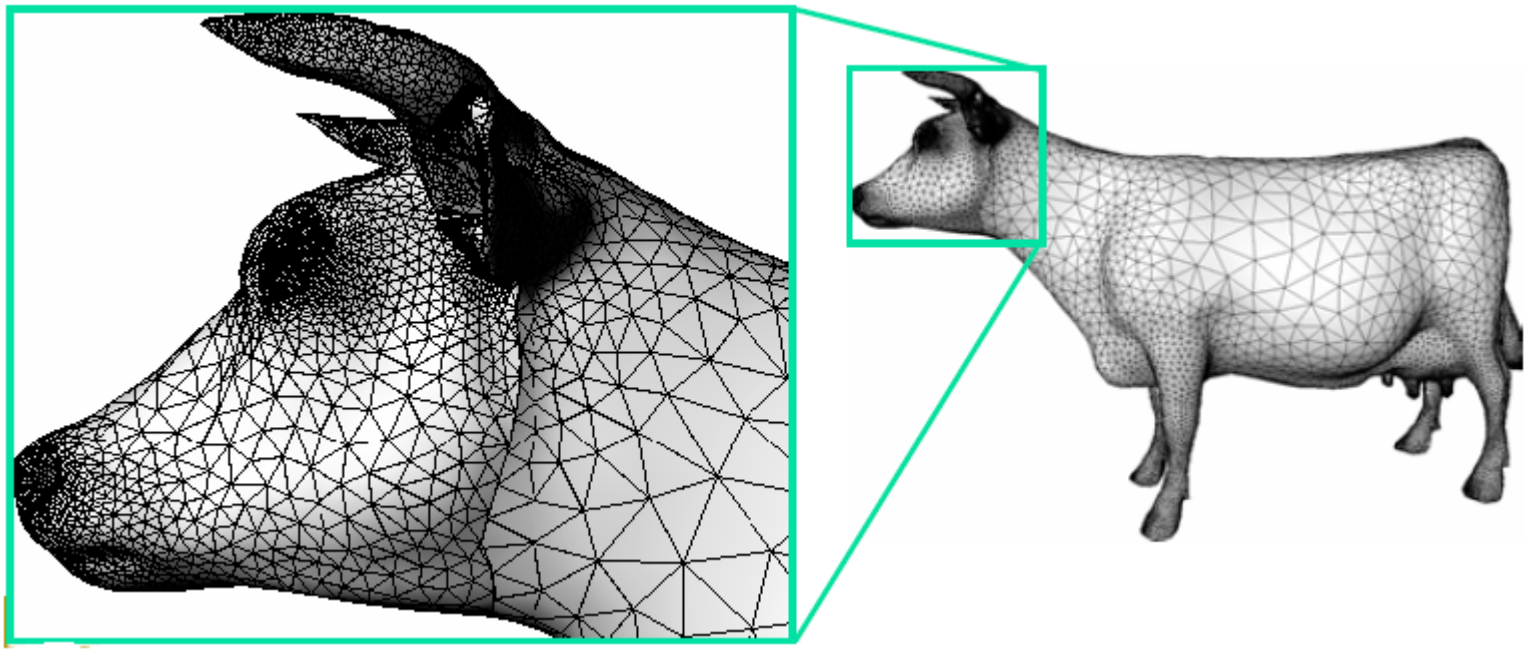
- Drawbacks
  - No sizing control
  - Approximation & quality depend on patch planarity & shape





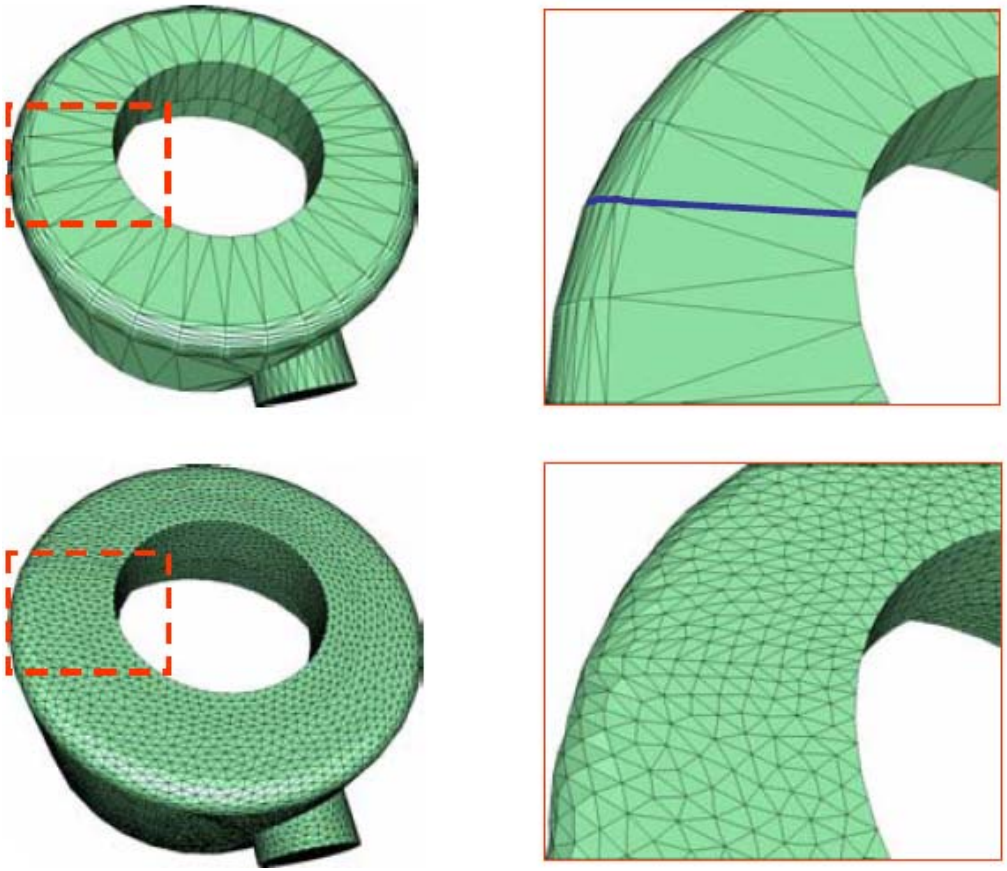
Boundary Consistency

- Need conformal mesh on boundary
- Place vertices conformally on shared boundaries



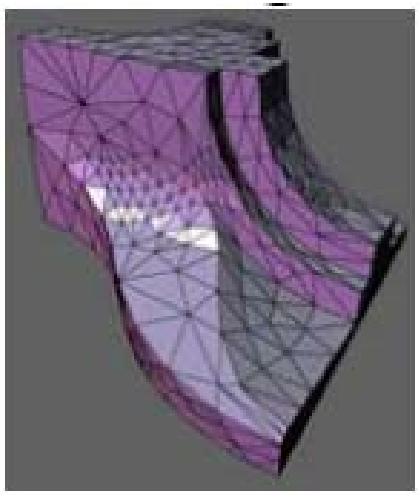
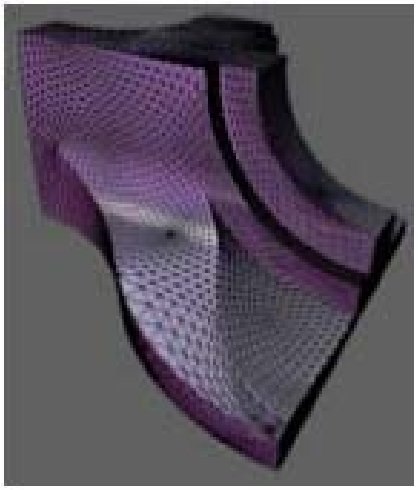
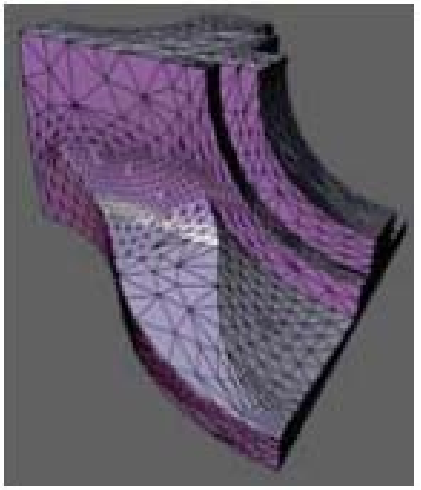
Conformal Boundaries

- Conformal but visible…



Features

- Preserving features – locate surface creases and prevent removing them
- Special handling by segmentation and/or 2D meshing

Global Methods - Properties

- Strongly depends on parameterization quality
  - In turn depends on segmentation
- Bottleneck
  - Parameterization
- Better shape control/spacing
- Better theoretical basis
  - Quality
  - Approximation
- Typically more complex to implement

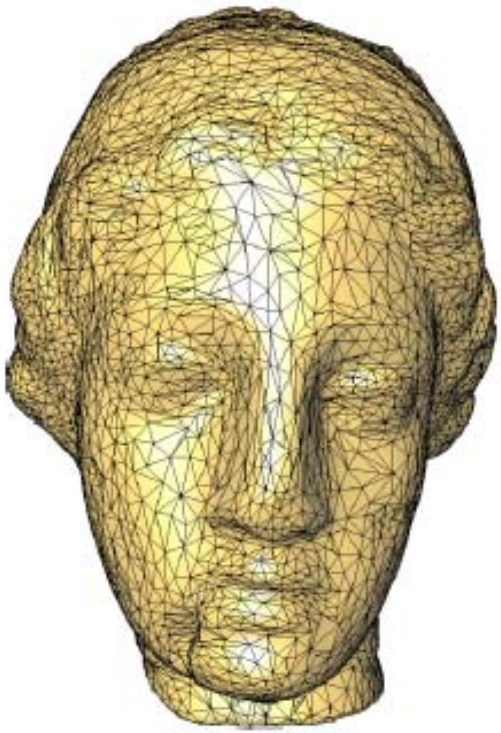
3. Remeshing Classifications

Remeshing Techinques

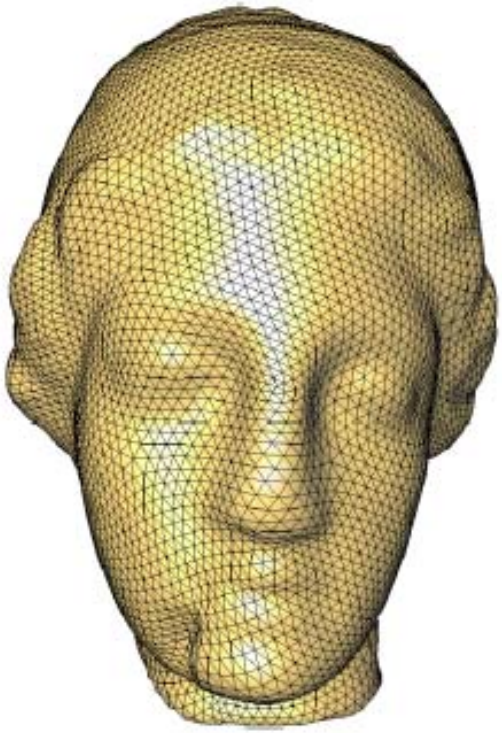
- Structured remeshing
- Compatible remeshing
- High quality remeshing
- Feature remeshing
- Error-driven remeshing

3.1 Structured Remeshing

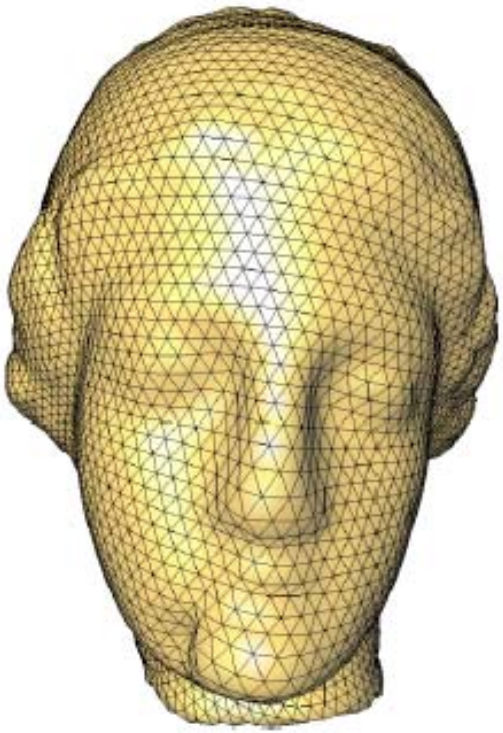
Structured Remeshing



Irregular



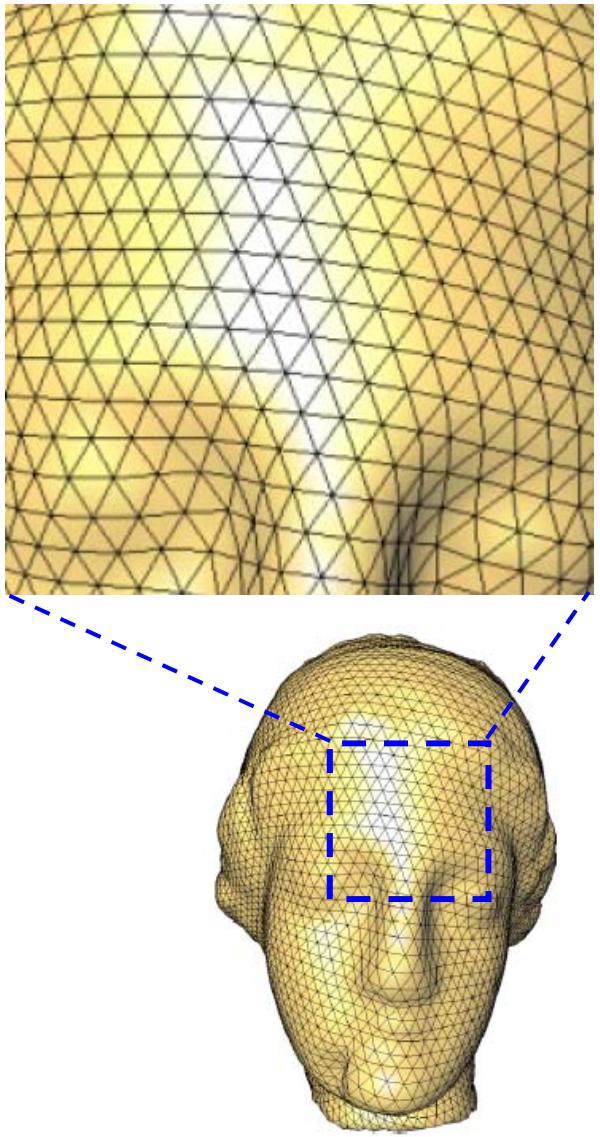
Semi-irregular



Regular

Regular Meshes

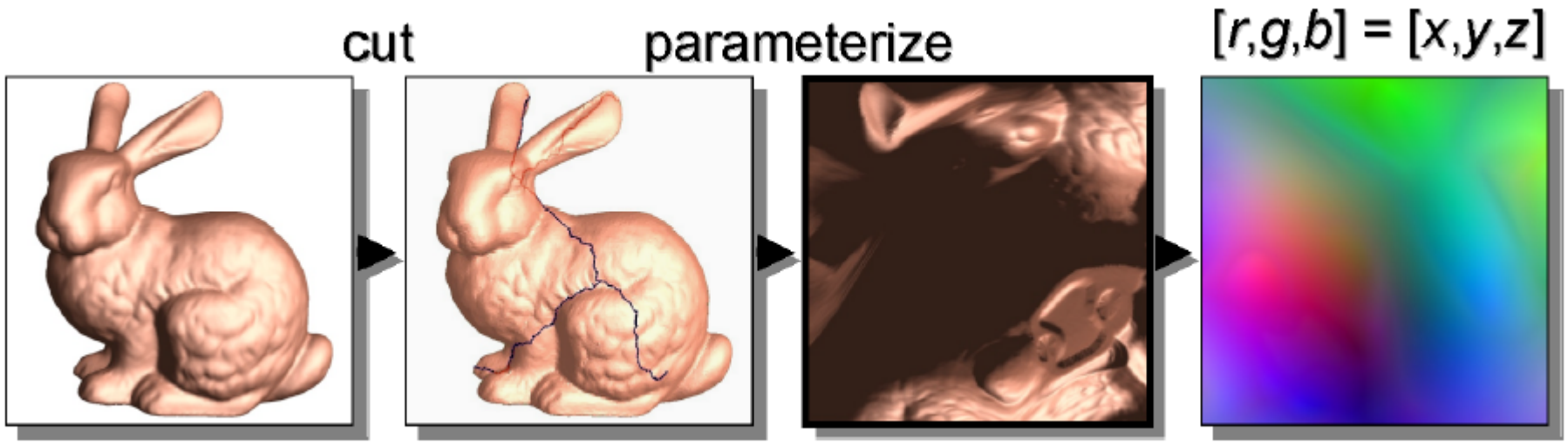
- Regular Meshes
  - Regular vertices
  - Extraordinary vertices
- Advantages
  - Simple connectivity graph
  - Efficient traversal and storage



Completely Regular

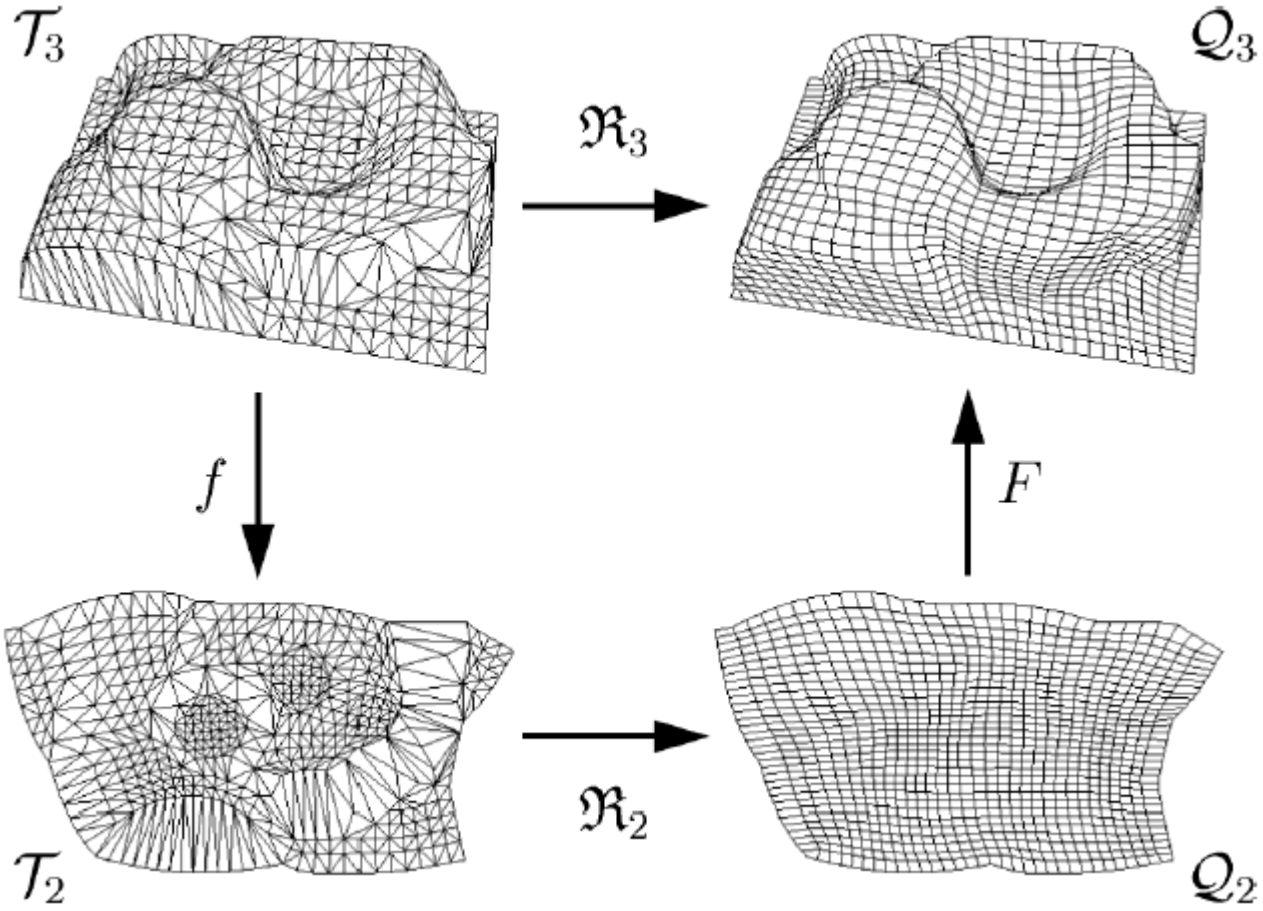
- Geometry Image

[Gu et al., Siggraph 2002]



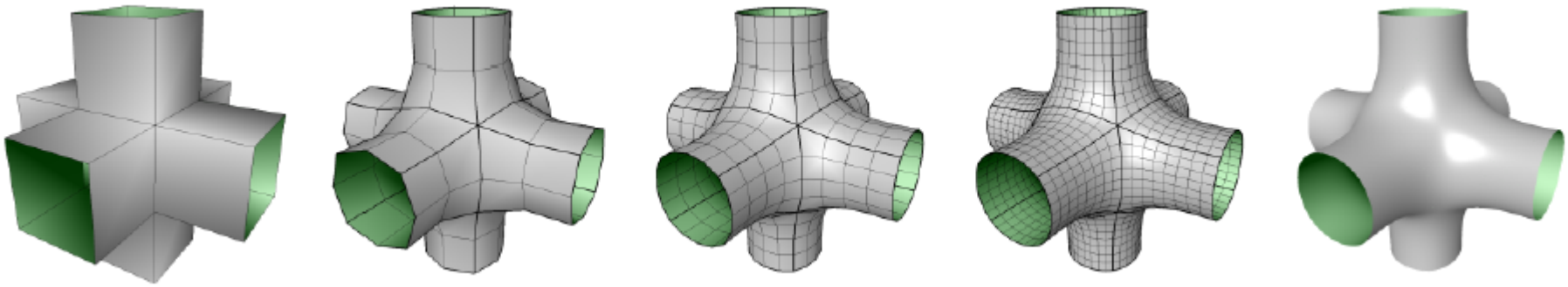
Quadrilateral Remeshing

[Hormann and Greiner, VMV 2000]



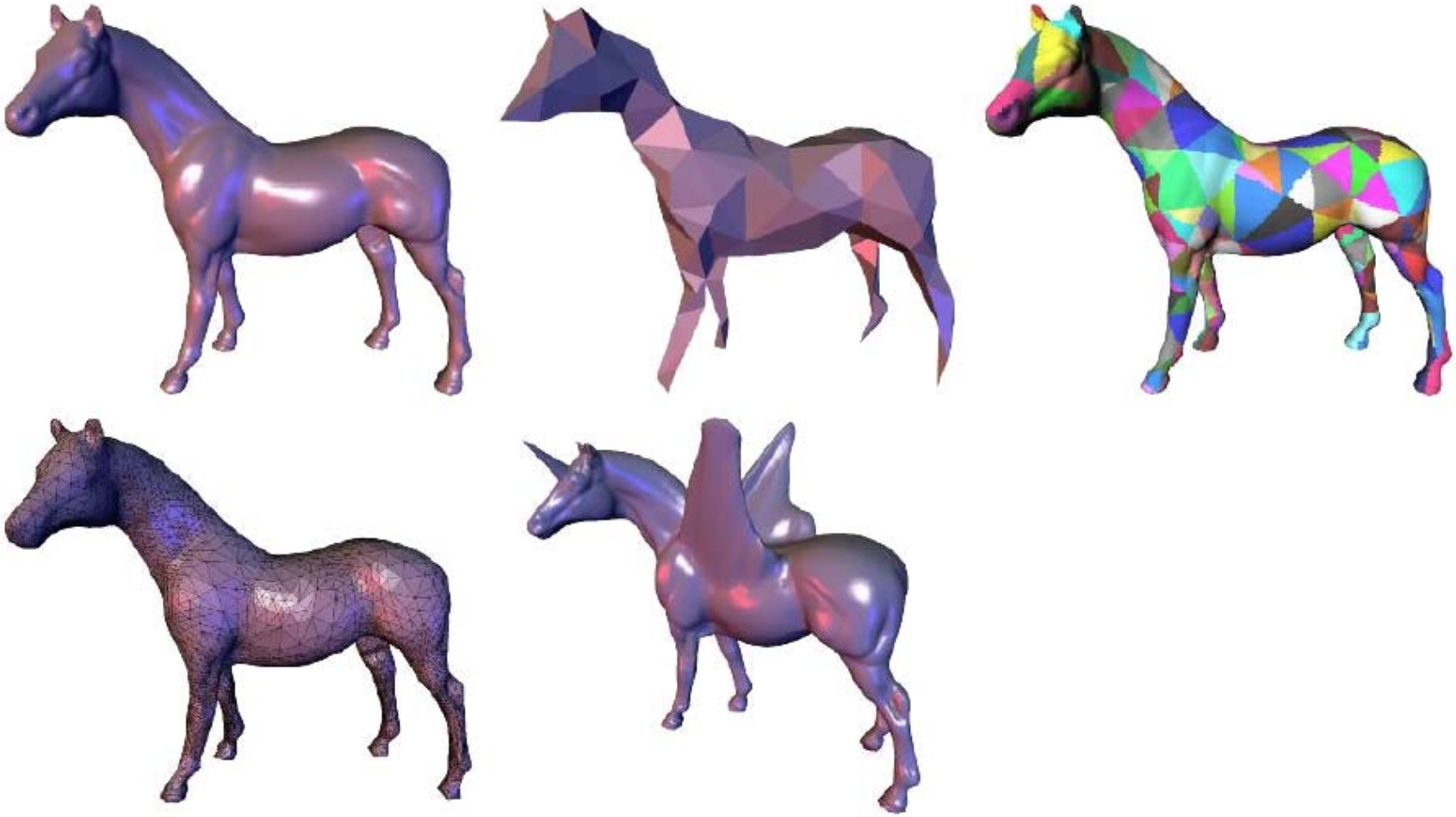
Semi-regular Remeshing

- Recursive subdivision



MAPS

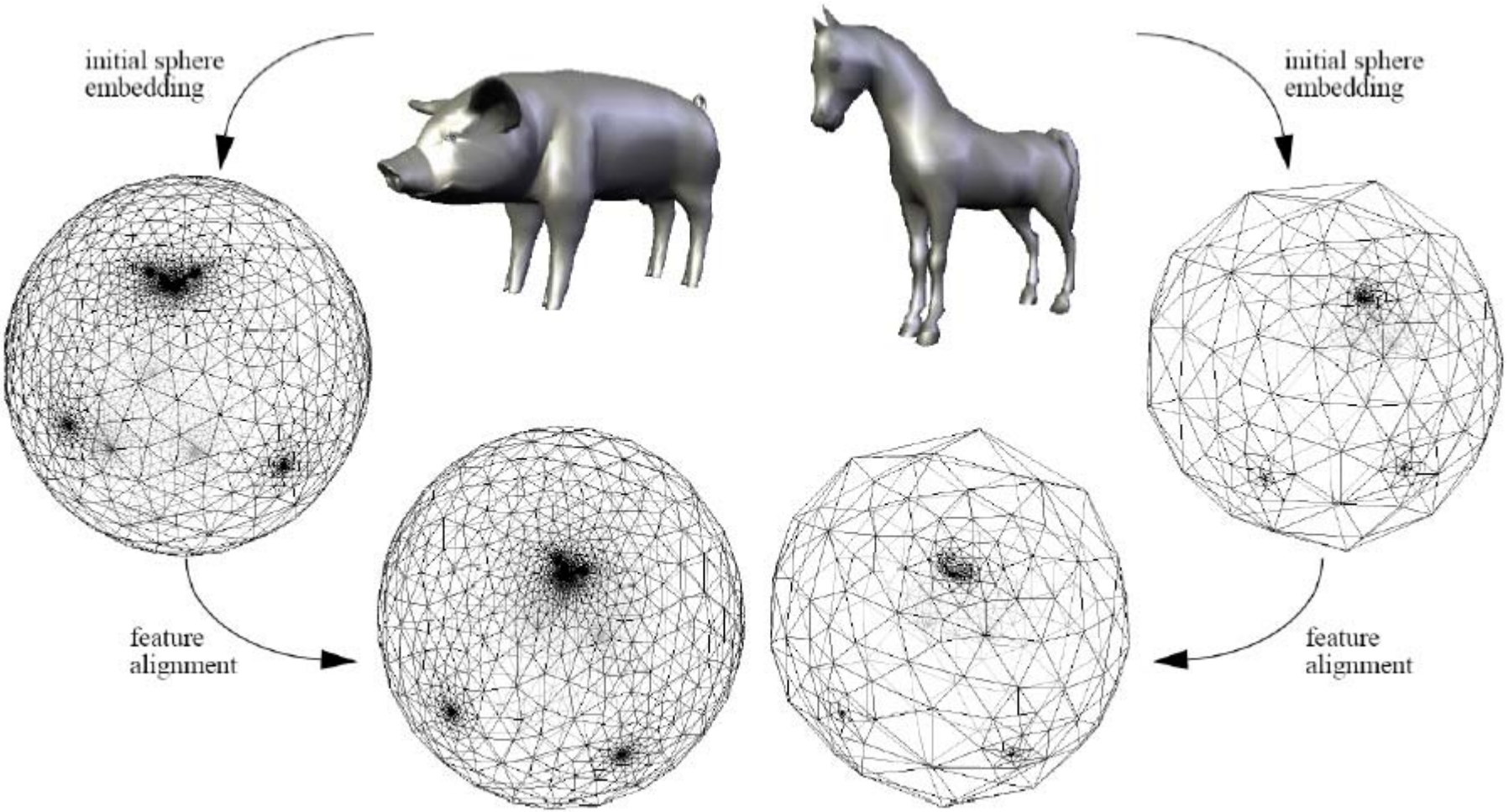
[Lee et al., Siggraph 1998]



3.2 Compatible remeshing

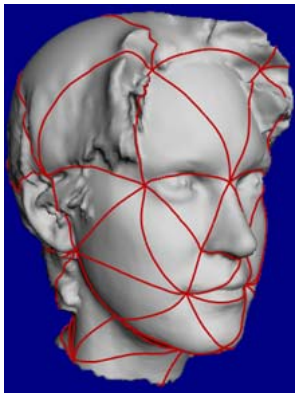
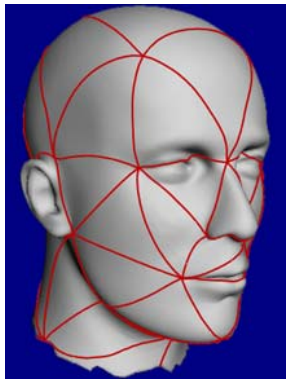
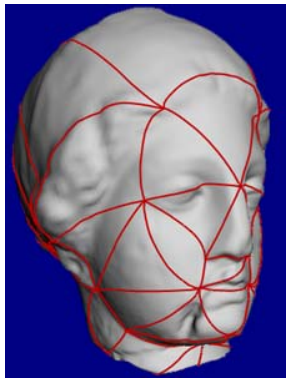
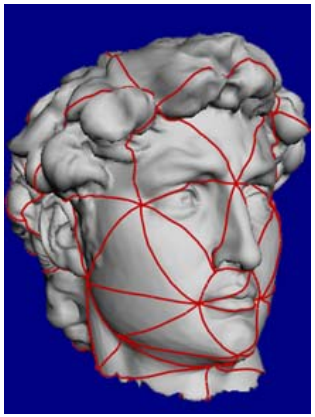
See details in classes of “Surface Parameterization” and “Mesh Morphing”

Joint Parameterization



Compatible remeshing

- Consistent mesh parameterizaiton
- Cross parameterizatin
- Inter-surface mapping
- Polycube maps
- Manifold parameterization

See details in classes of “Surface Parameterization” and “Mesh Morphing”

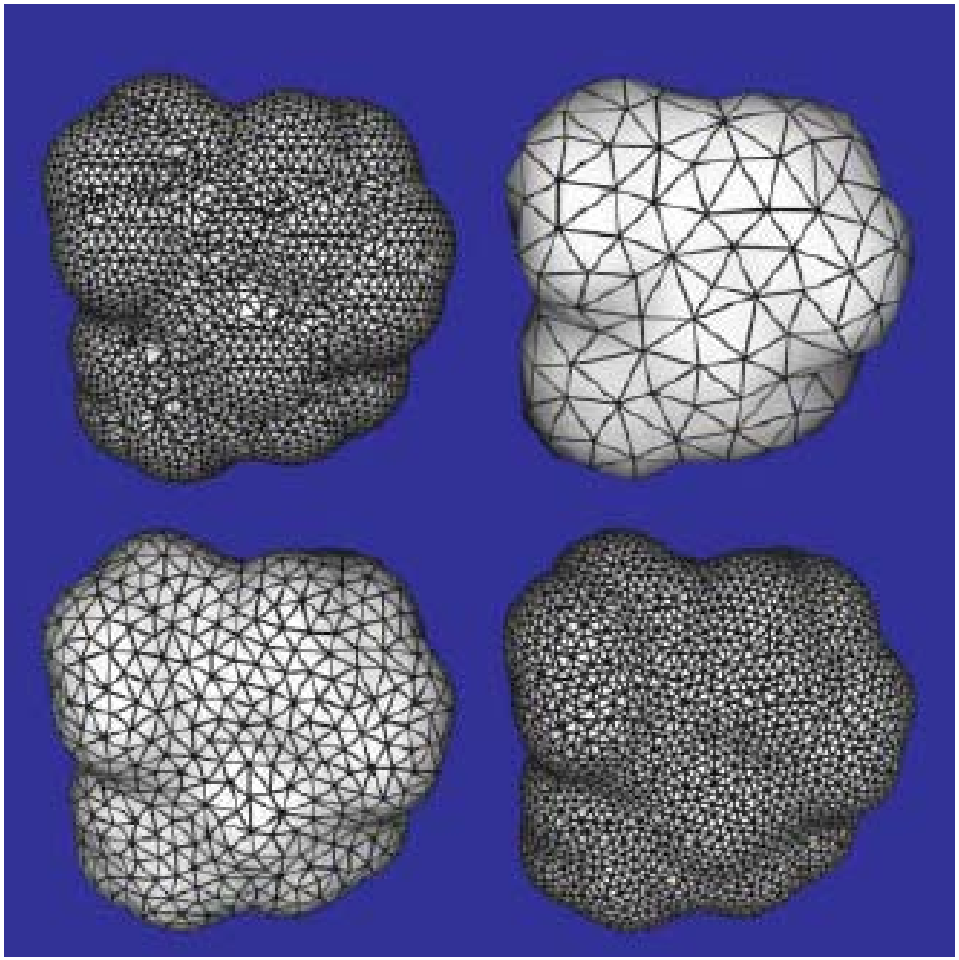
3.3 High Quality Remeshing

High Quality

- Well shaped elements
  - well-shaped triangle has aspect ratio as close to 1 as possible
- Uniform or isotropic sampling
  - sampling is locally uniform in all directions
- Smooth gradation sampling
  - if the sampling density is not uniform -- it should vary in a smooth manner

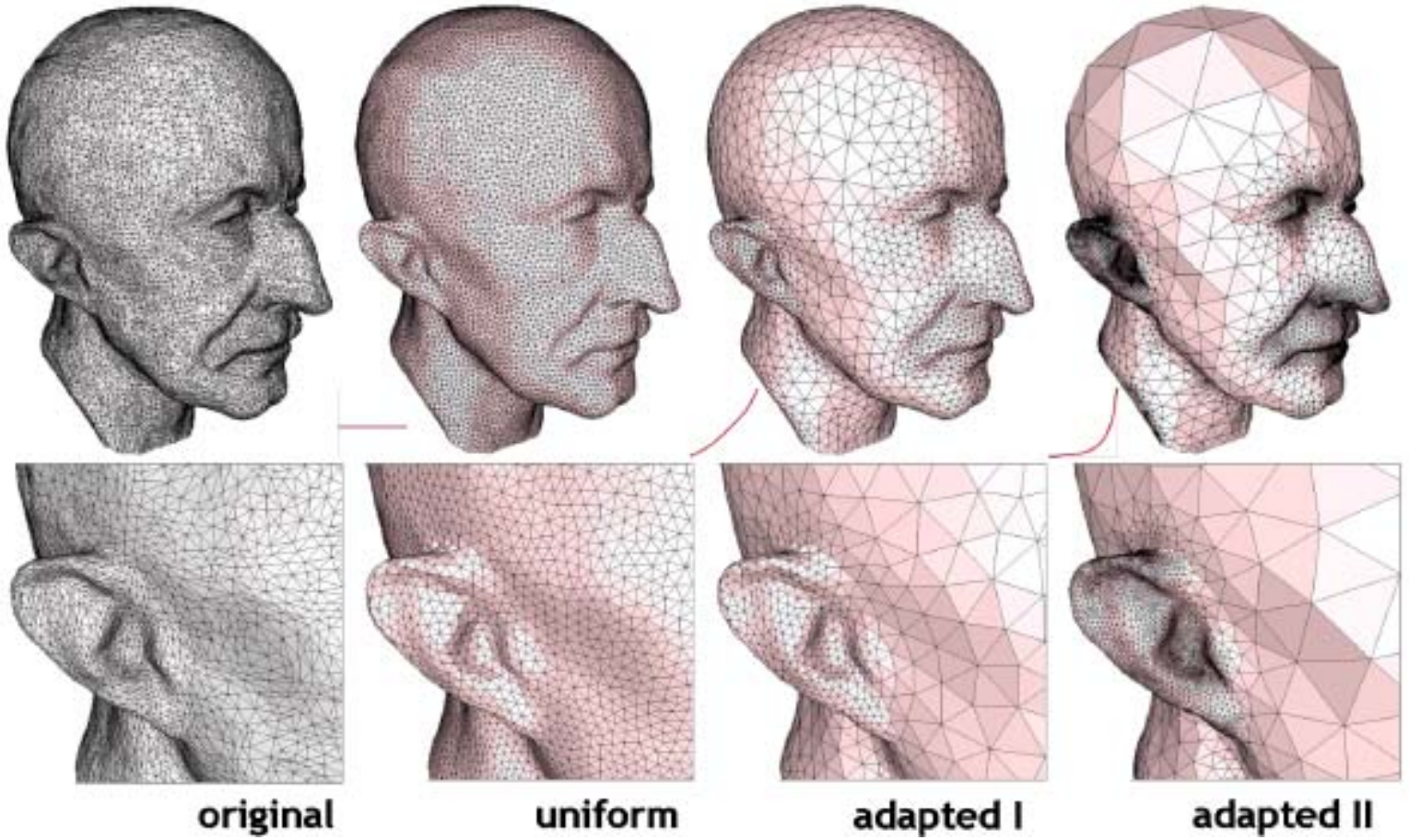
Re-Tiling

[Turk, Siggraph 1992]

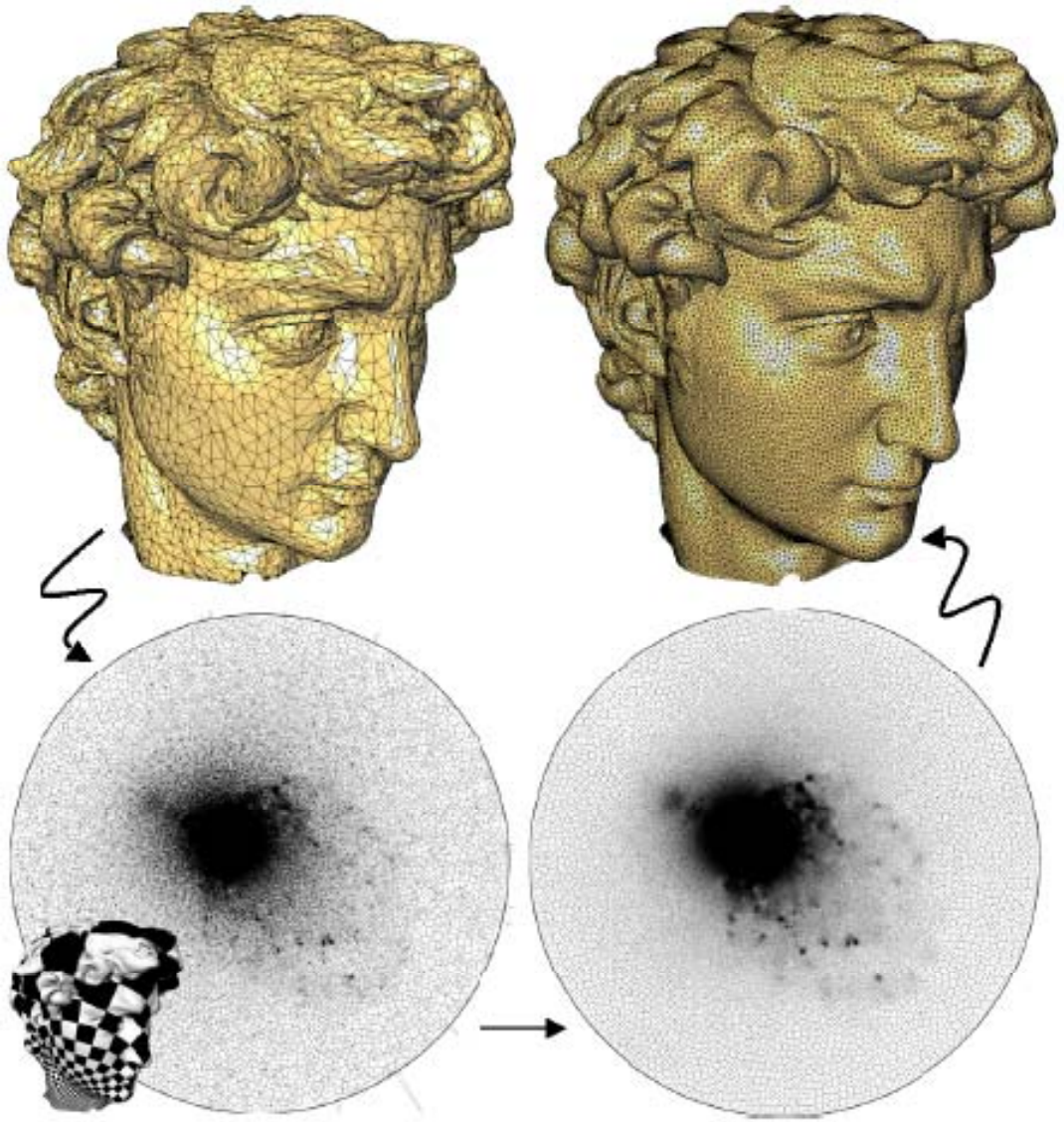


Interactive Geometry Remeshing

[Alliez et al., Siggraph 2002]



Lloyd Relaxation

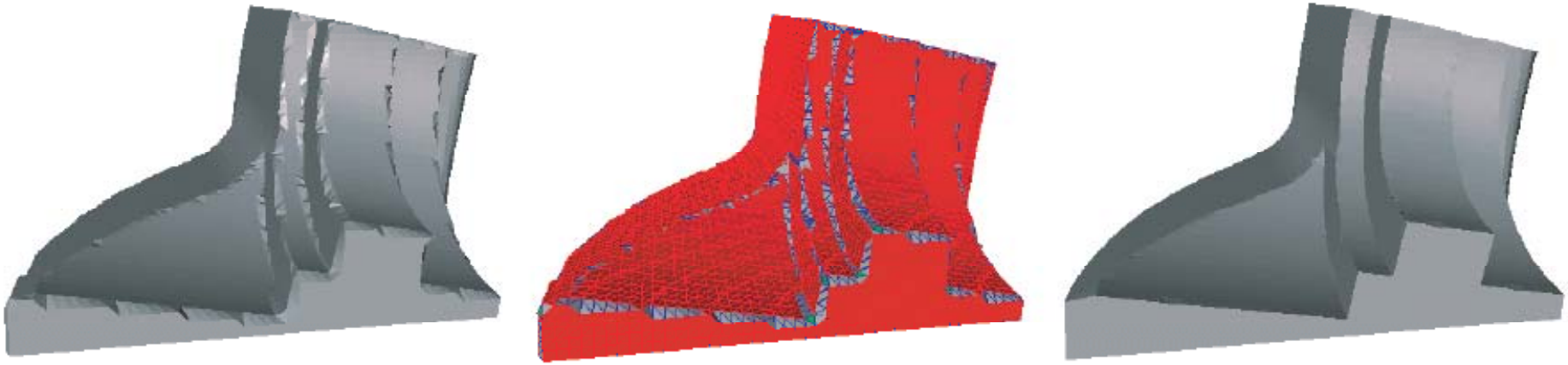


[Alliez et al., SMI 2003]

3.4 Feature Remeshing

Feature Preserving/Enhancement

[Attene et al., Siggraph 2003]

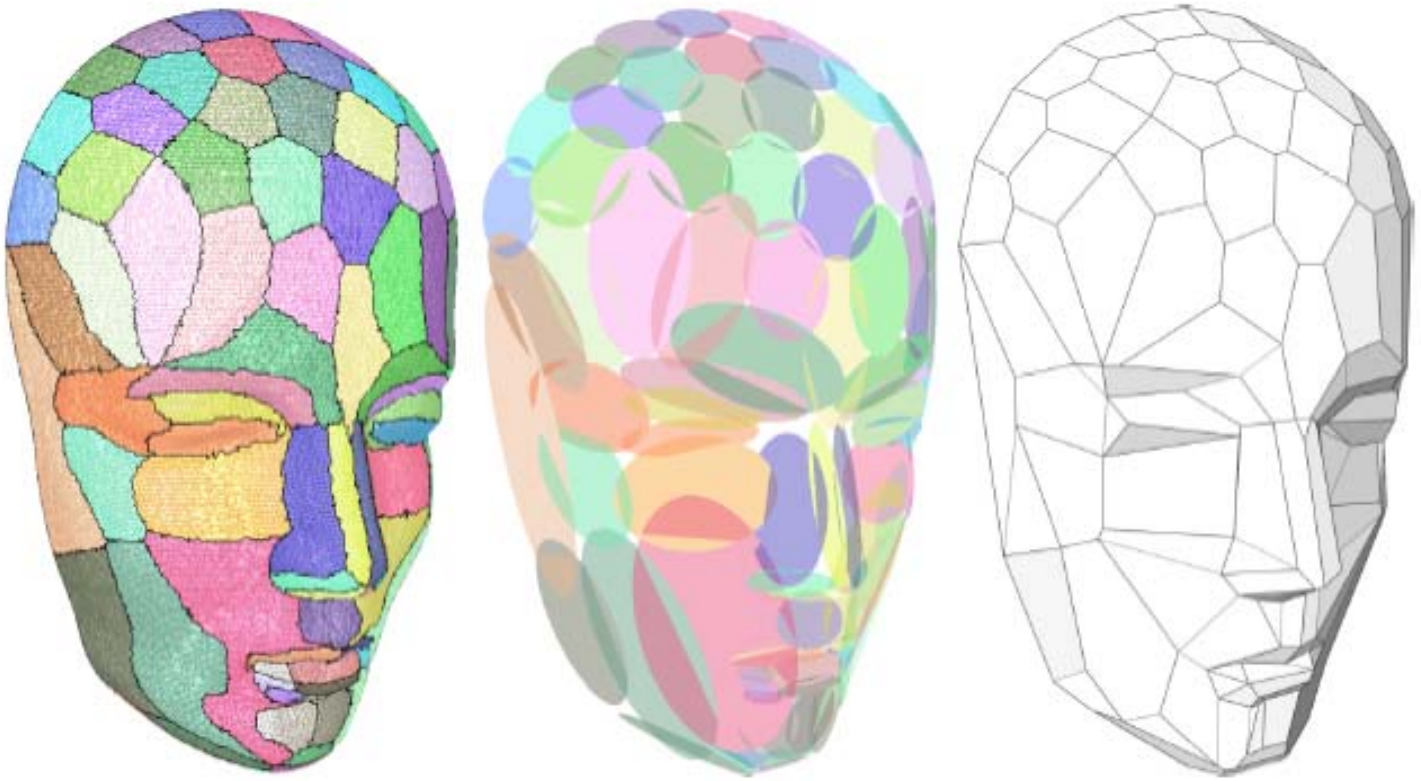


3.5 Error-Driven Remeshing

Error-driven Remeshing

- Variational surface approximation (VSA)
  - Remeshing into planar patches

[Cohen-Steiner et al, Siggraph 2004]



4. Case Studies

Case Study

4.1 Interactive Geometry Remeshing

[Alliez et al., Siggraph 2002]

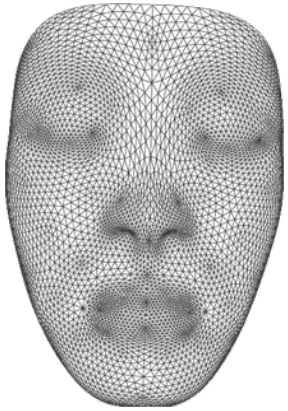
Main Ideas

- Work in parameter space:
  - 2D space, much easier/faster!
- Use a density map to design the sampling:
  - Density map can be computed and/or painted
- Avoid long optimizations as much as we can:
  - Error diffusion for near-optimal vertex placement

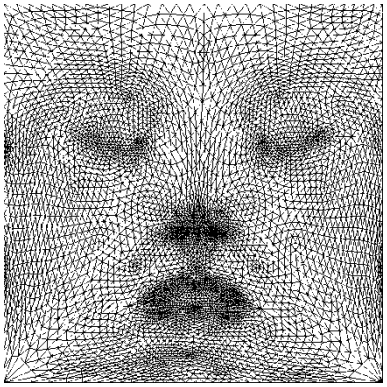
Pipeline

- Geometry Analysis - input-dependent
  - **Parameterization (remove embedding)**
  - **Geometry Maps (2D images to substitute for 3D)**
- Remeshing Design – real-time
  - **Flexible Design (use conventional DSP tools)**
  - **Realtime Resampling (use error diffusion)**
- Mesh Generation - output-dependent
  - **Triangulation and Reprojection (2D back to 3D)**
  - **Final Optimization (only if needed!)**

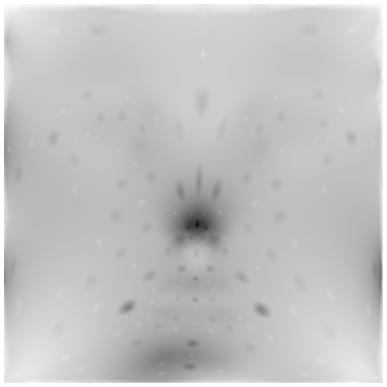
1. Geometry Analysis



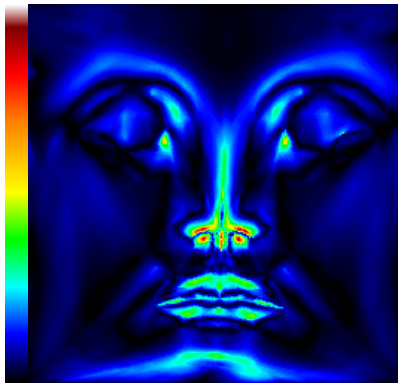
- Creating the parametrization charts
- Computing 2D geometry maps



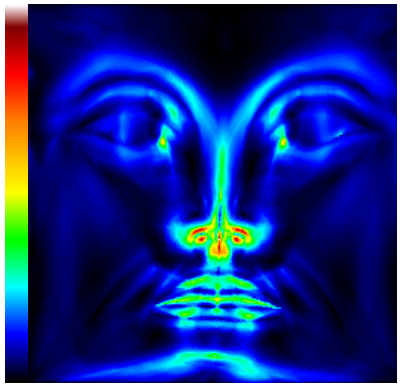
Parametric domain



Area stretching



Mean curvature



Gaussian curvature

2. Remeshing Design

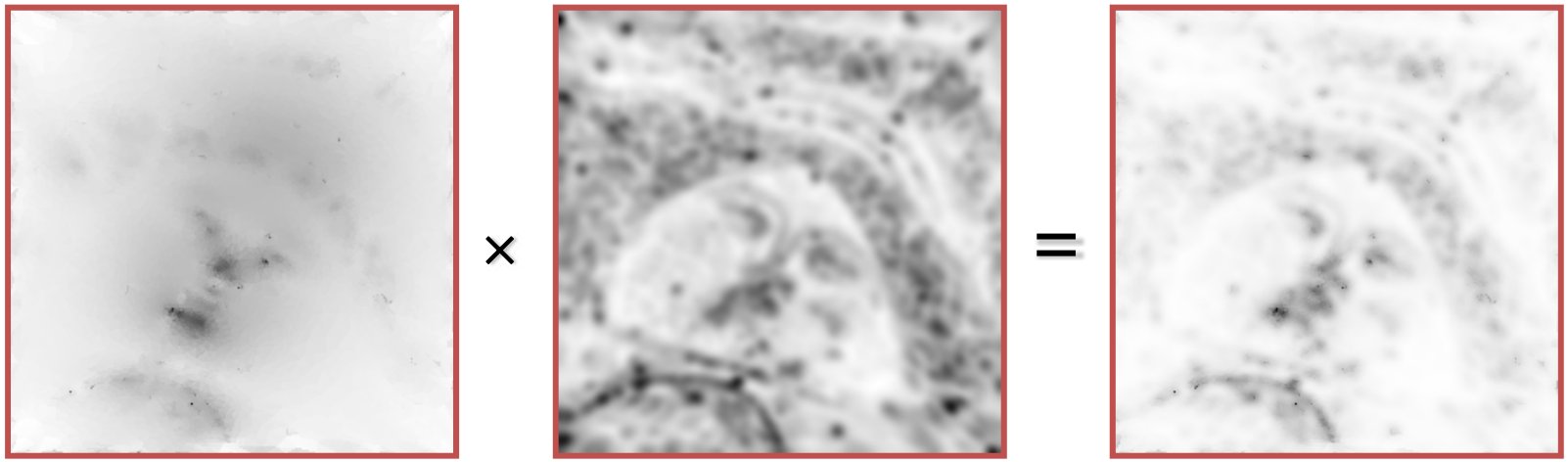
Design of the desired vertex density

– Select a sampling criteria

- **Can use any combination of precomputed maps**
- **Or any user-defined, spray-painted map**

– Multiply (pixel by pixel) by the area map

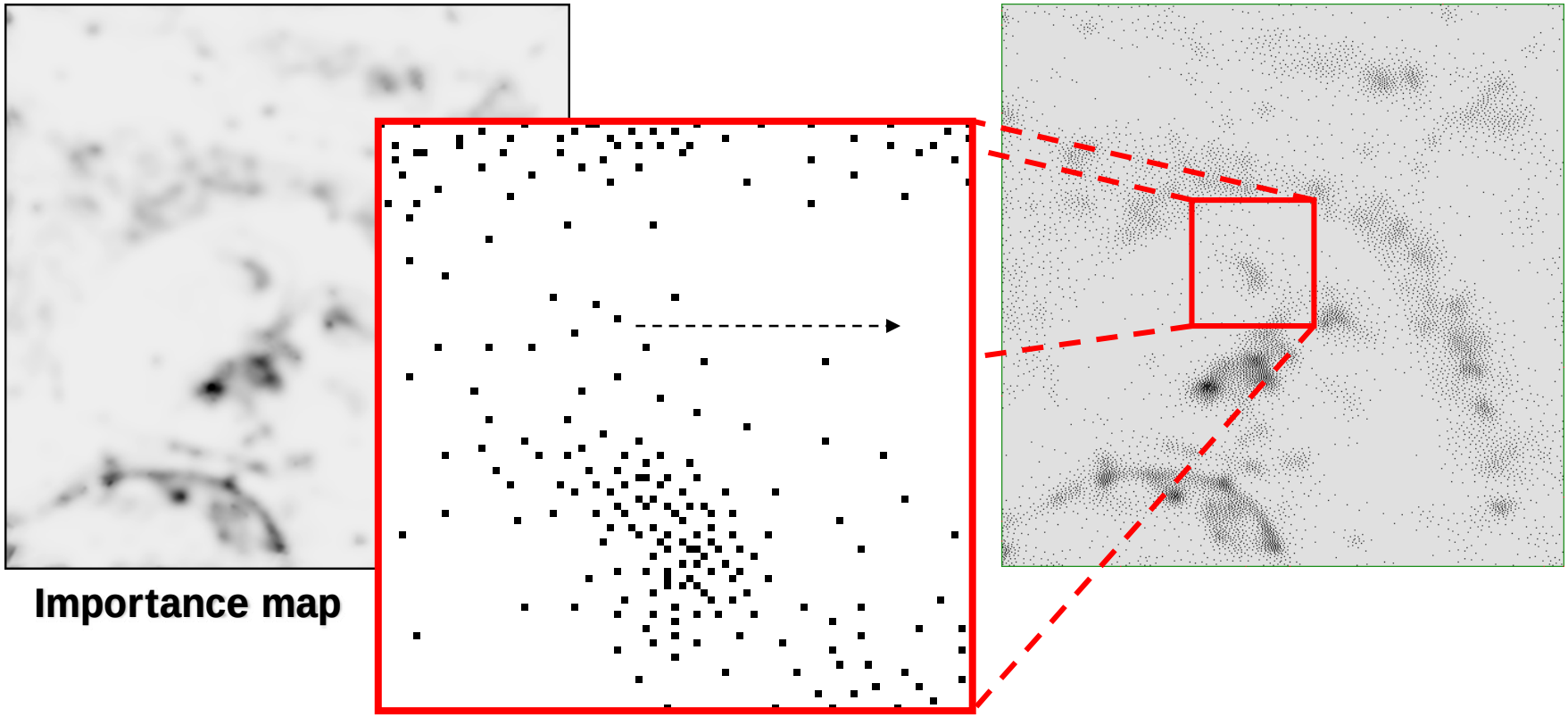
⇨ Importance map (sampling space)



Real-time Resampling

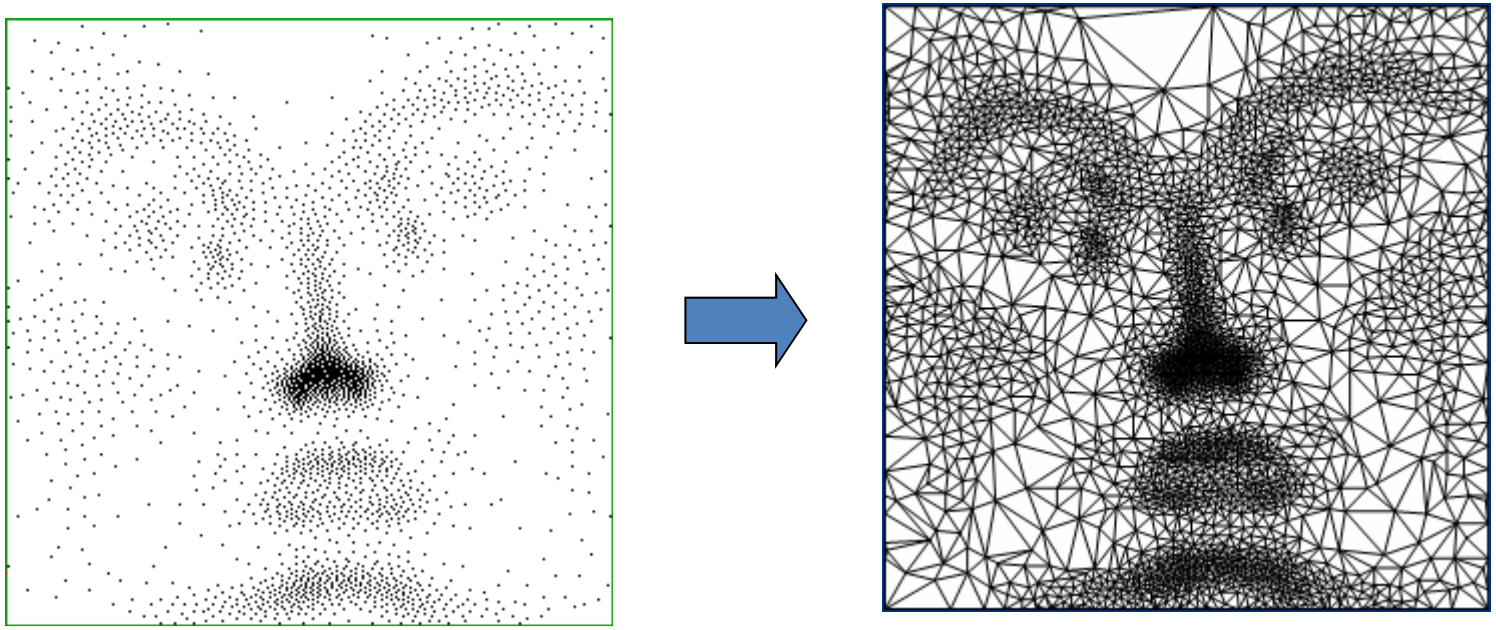
– 512×512 picture in 15ms

Independent of vertex budget!



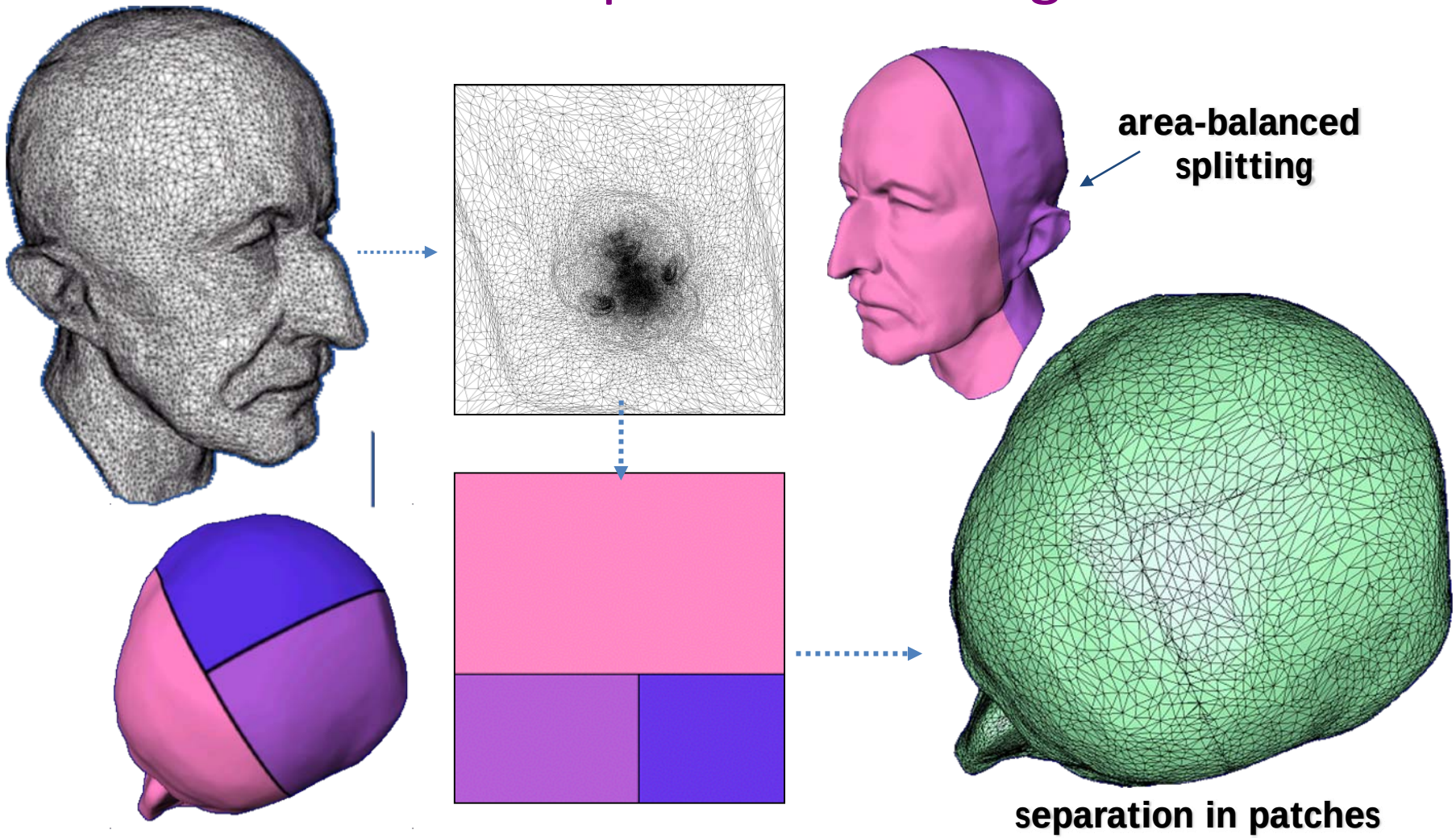
3. Mesh Generation

- Triangulate in parameter space
- Mesh optimization

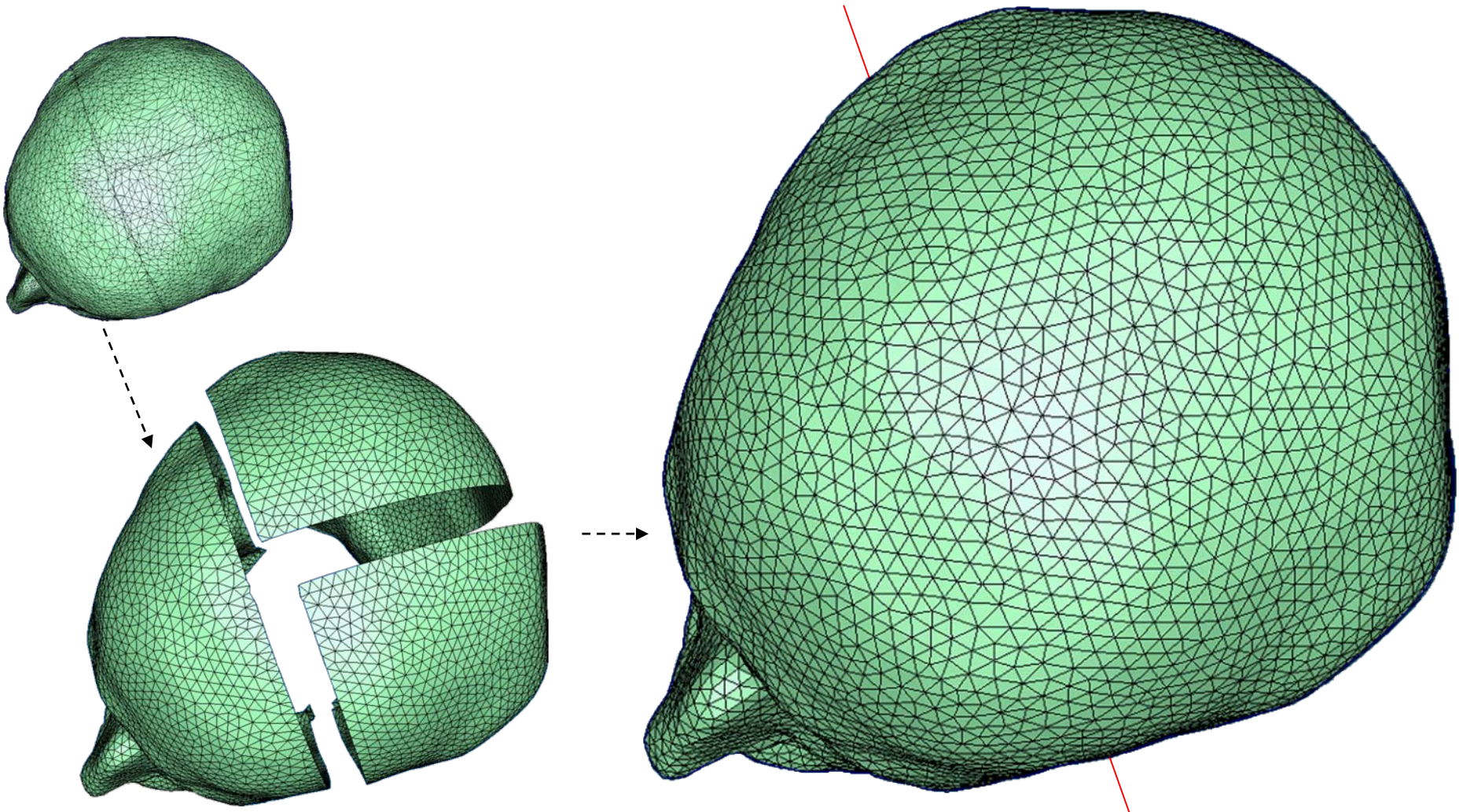


Remeshing Example

The area stretch map can drive tiling



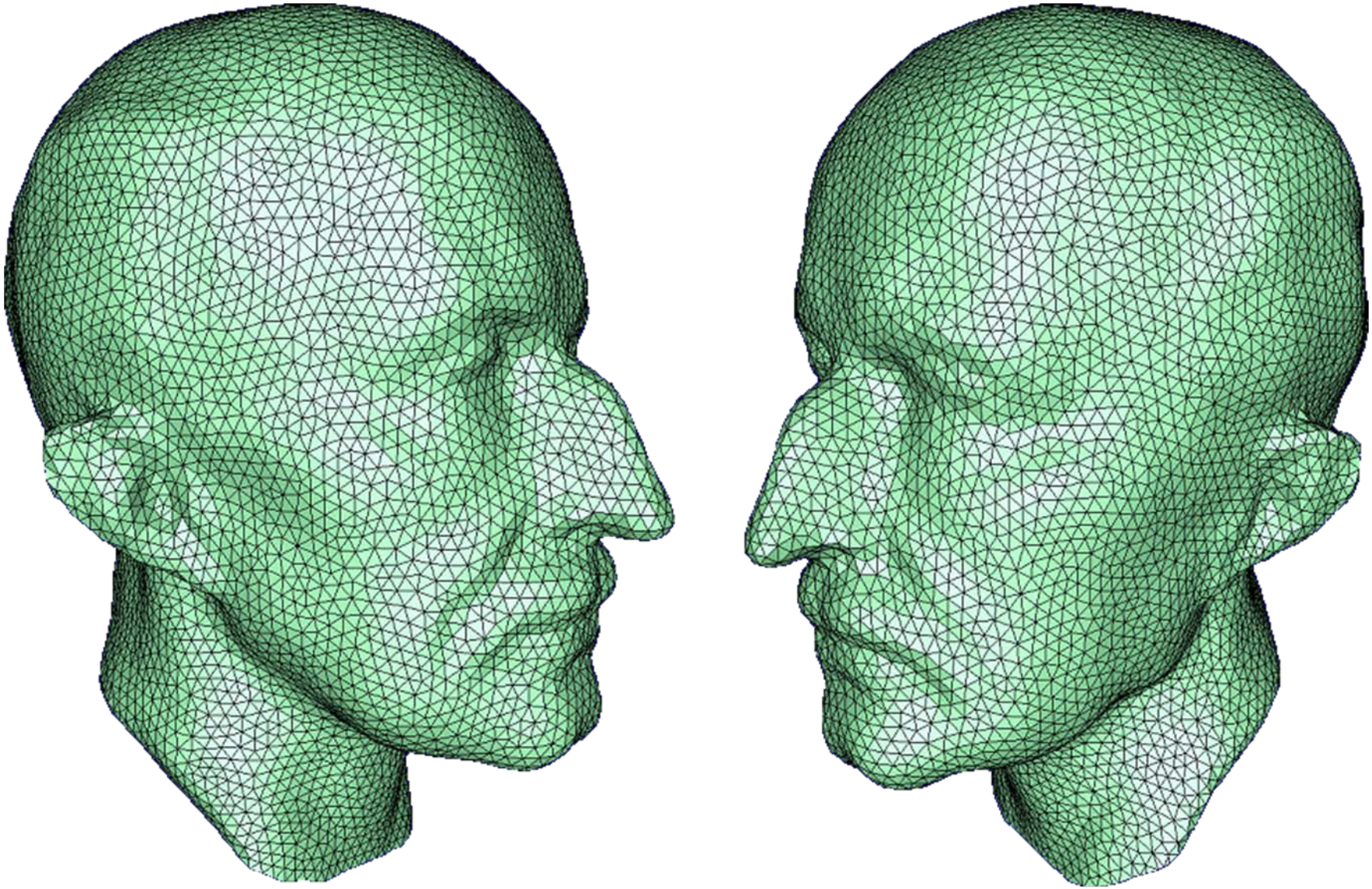
Remeshing Example



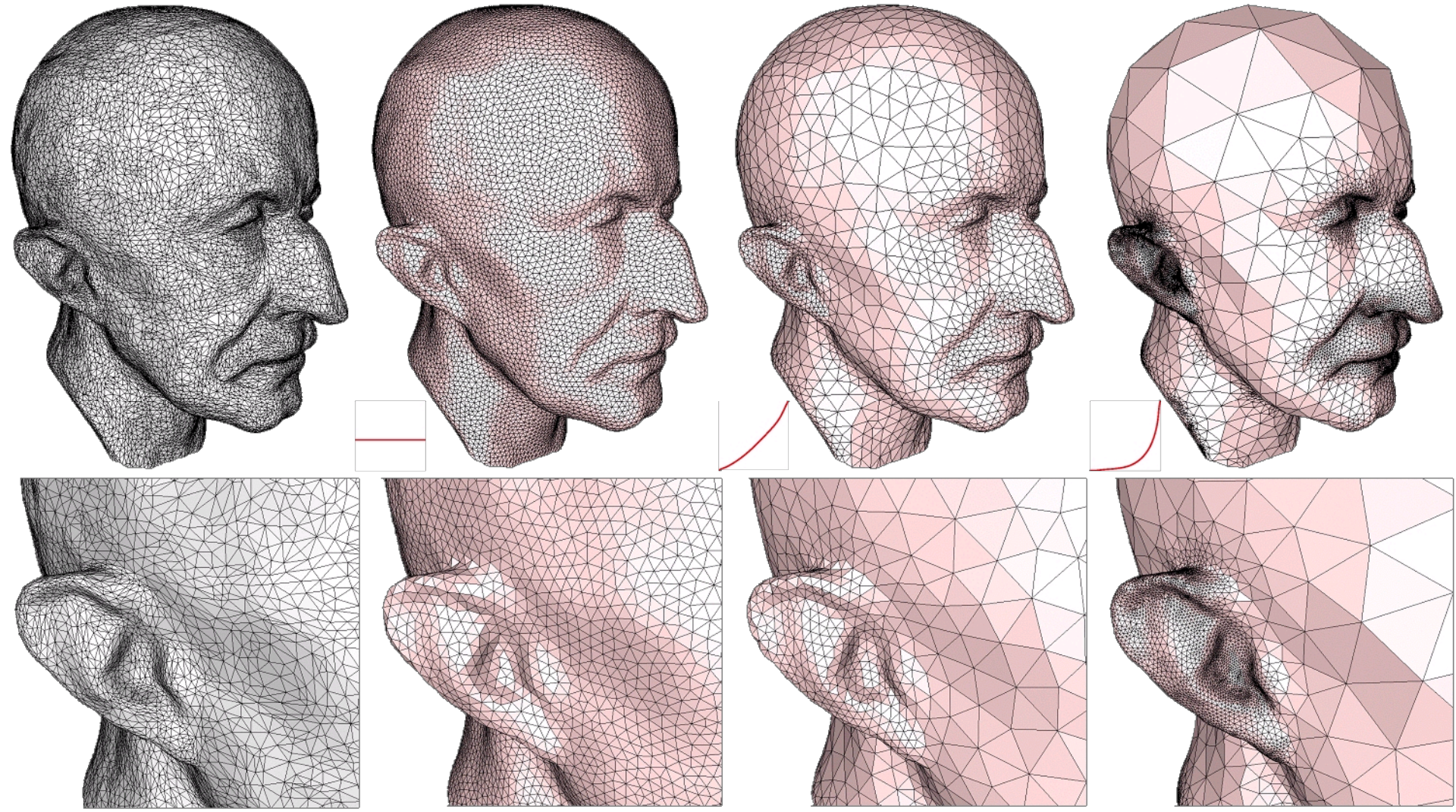
Remeshing patch by patch

And after tangential smoothing...

Remeshing Example



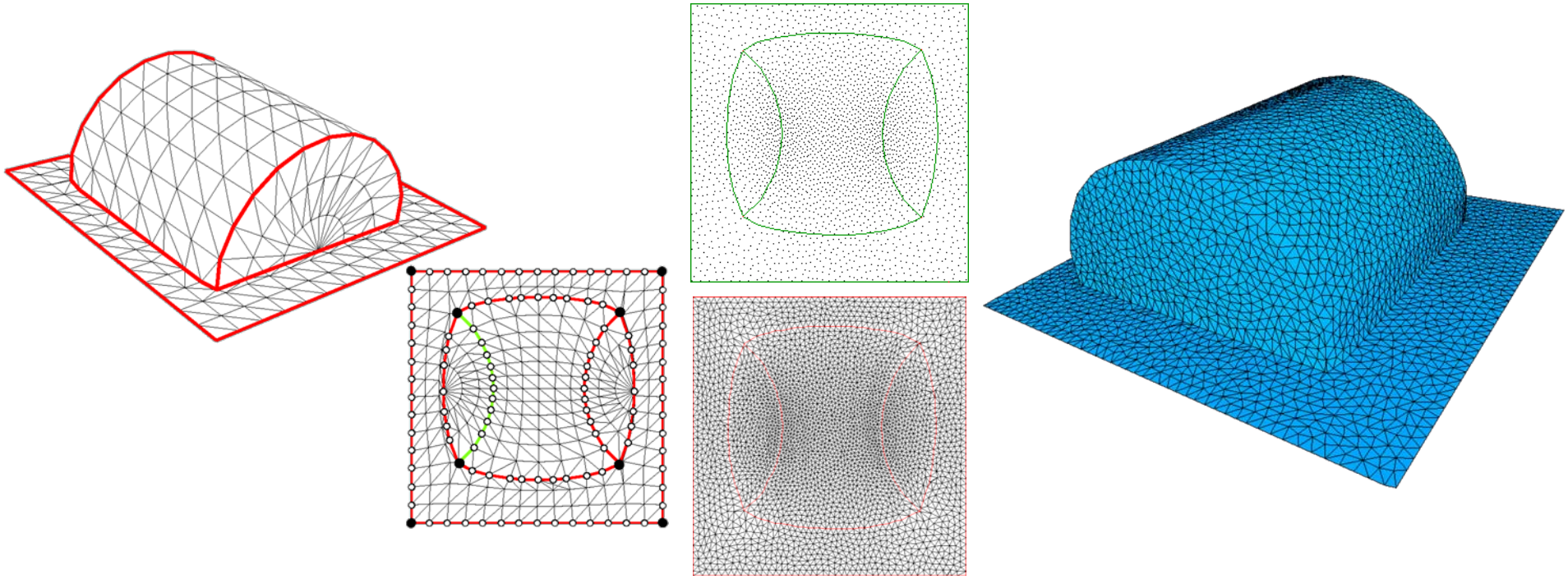
More Examples



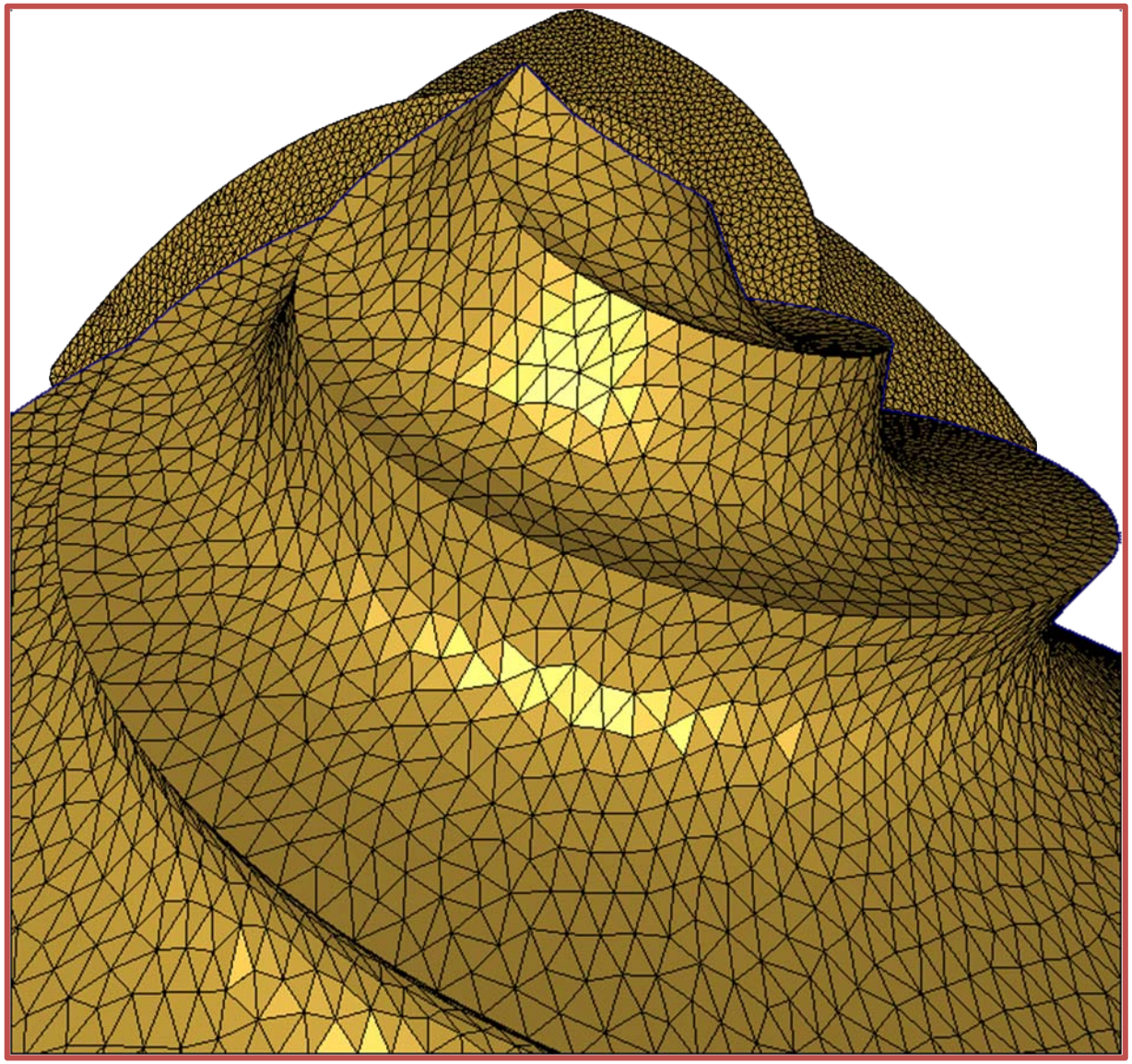
Preserving Features

Using a *Feature Skeleton*

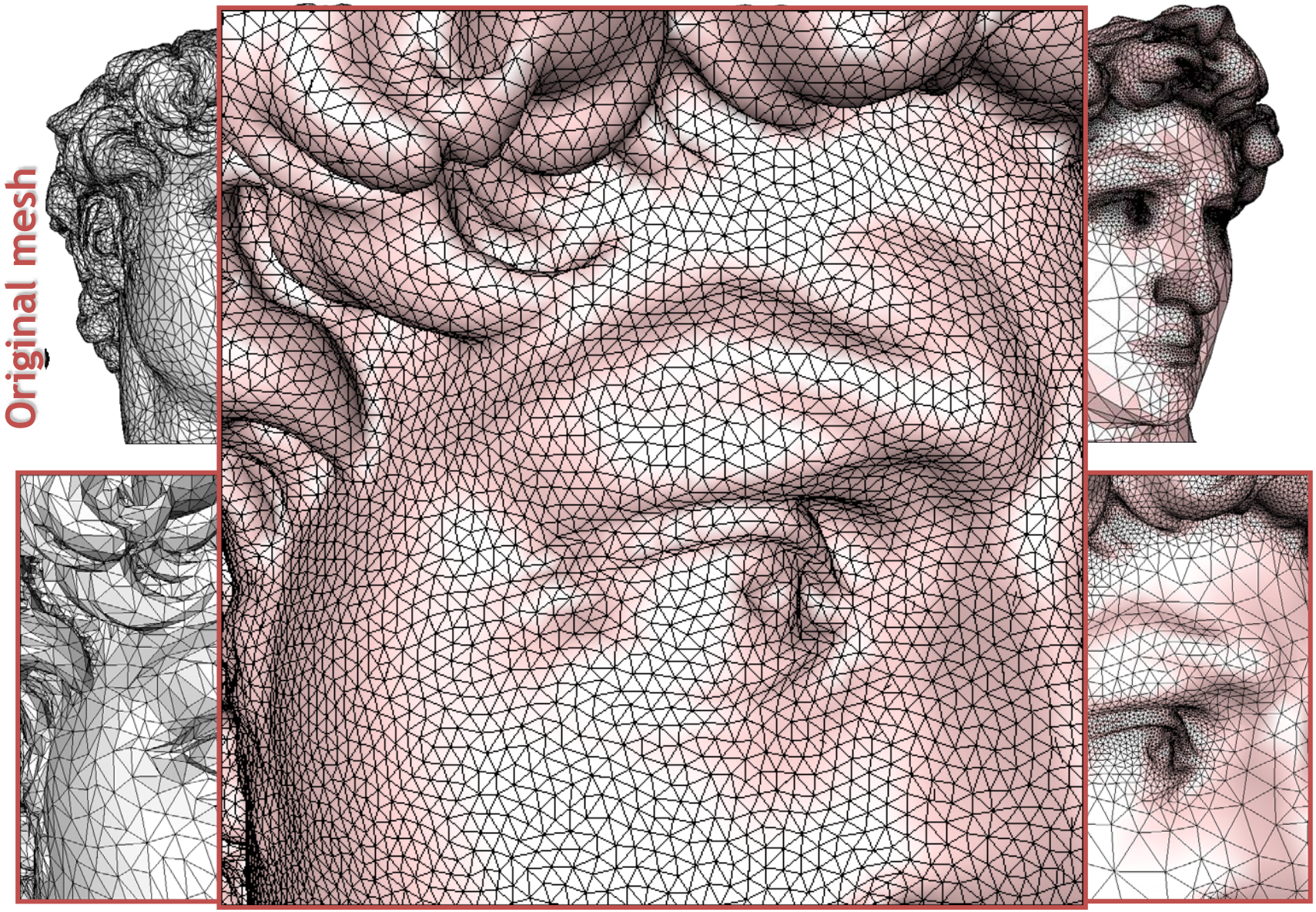
- Extract feature graph
- 1D error diffusion along features
- Constrained Delaunay triangulation



Example With Sharp Edges

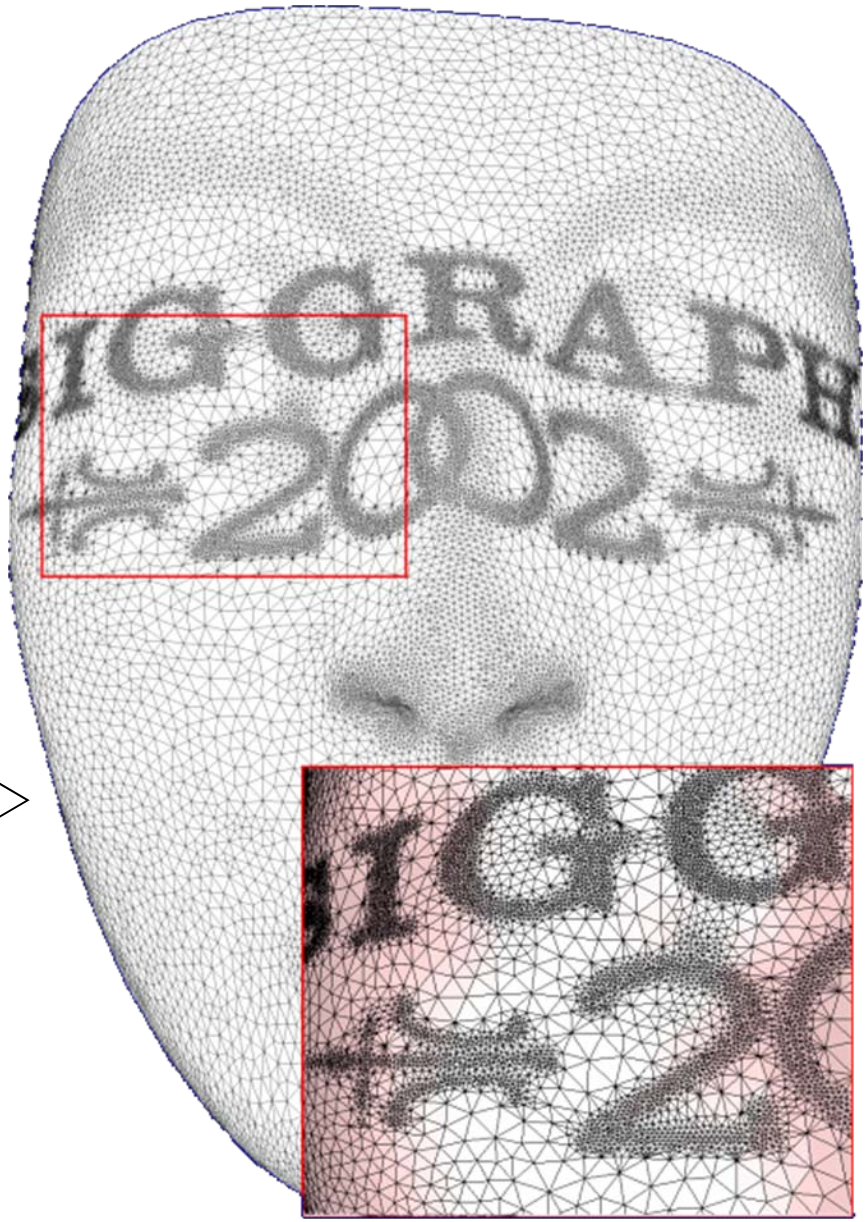


Remeshing Example



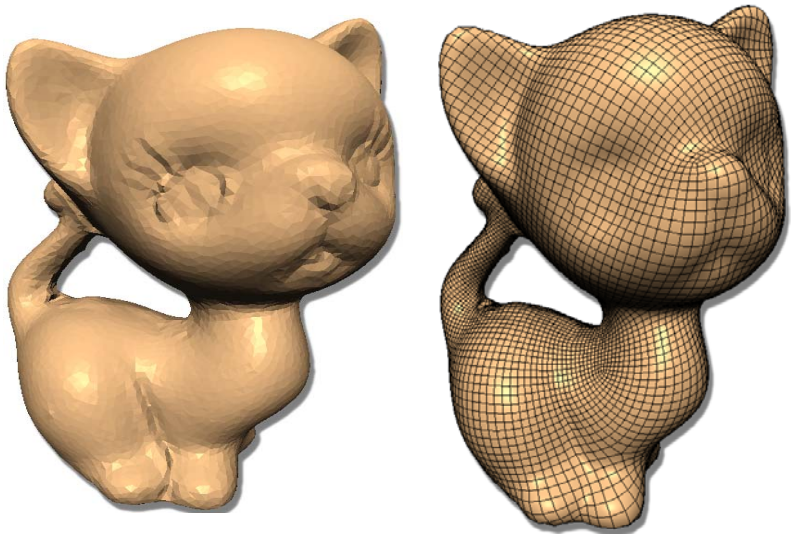
User-Defined Maps

paint either in the importance map or directly on the mesh



More…

- Quad Meshing
  - Quadrangulation



- Tetrahedral Meshing
  - Volume meshing

Summary

- Remeshing is to generate high quality meshes
- Many applications
- Still a hot topic
- Trends
  - Local → Global (shape analysis)
  - Greedy → Optimization
  - Properties (feature) preserving
  - Application dependent

Discussions