(Re)Meshing

Motivations

A key technique in many industrial applications involving numerical simulation and geometric modeling

- Think about the discrete harmonic mapping and mean value coordinates
- Recent efforts to handle arbitrary irregular / non uniform meshes.
- Most scanned surfaces need undergo complete remeshing before further processing.

Related Work and Observation

Most remeshing techniques proceed by:

- simplification / refinement (adaptation)
- optimization
- resampling (point sampling)

Control over:

- vertex density
- shape of elements
- etc.

Local Structure of Meshes

Local structure: described by the type, shape, orientation, and element distribution:

- Element type: triangles, quadrangles…
- Element shape: isotropic (locally uniform in all directions), anisotropic
- Element density: uniform distribution, nonuniform/adaptive distribution
- Element alignment and orientation: sampling sharp features, orientation of anisotropic elements

Global Structure of Meshes

Regularity of meshes

- Irregular meshes:
- Semiregular meshes: usually produced by regular subdivision of a coarse initial mesh
- Highly regular meshes: most vertices are regular
- Regular mesh: can compactly be replaced as a 2D array
  - Can be generated only for a very limited number of input models (topological torus)

Meshing Algorithms

Classified as: greedy, variational, and incremental methods

- Greedy (decimation): one local change (e.g. vertex insertion) at a time until the goal is satisfied
- Variational (relaxation): minimizing an energy functional such that low levels of this energy correspond to good positions of vertices
- Incremental: combines both above methods

Paper:

"Isotropic Surface Remeshing"

by Pierre Alliez, Eric Colin de Verdiere, Olivier Devillers, and Martin Isenburg

IEEE International Conference on Shape Modeling and Applications, 2003

- Explore the problem of *isotropic* surface (re)sampling.
- Provide a new remeshing tool for geometric modeling and simulation.

Previous work

Two different fields:

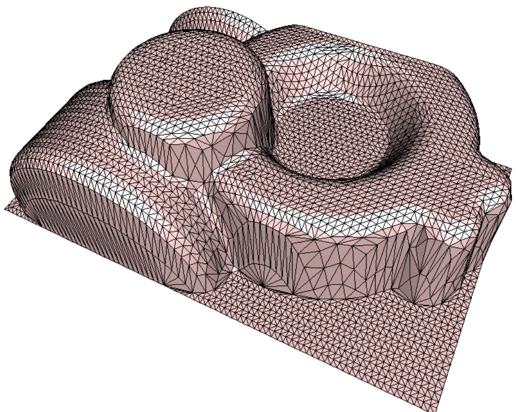
Finite Element community:

High-quality meshes
for simulation

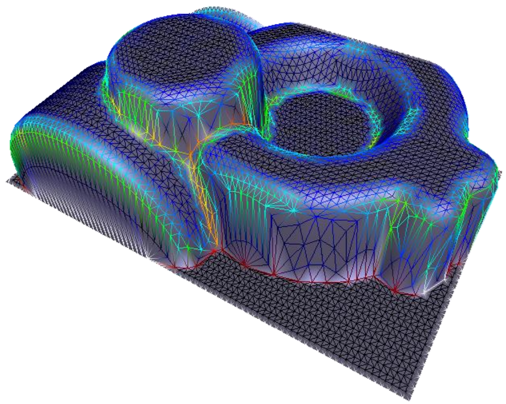
Computer Graphics community:

Geometric modeling
for effective processing and fast display

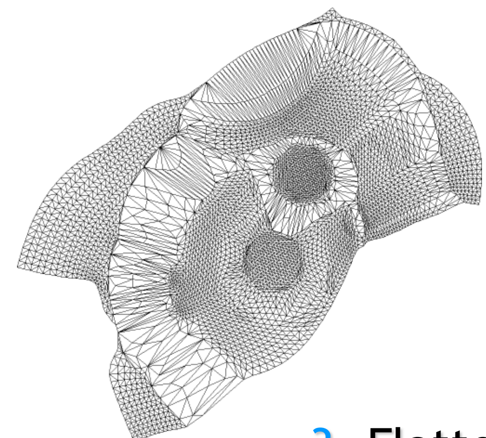
Proposed technique



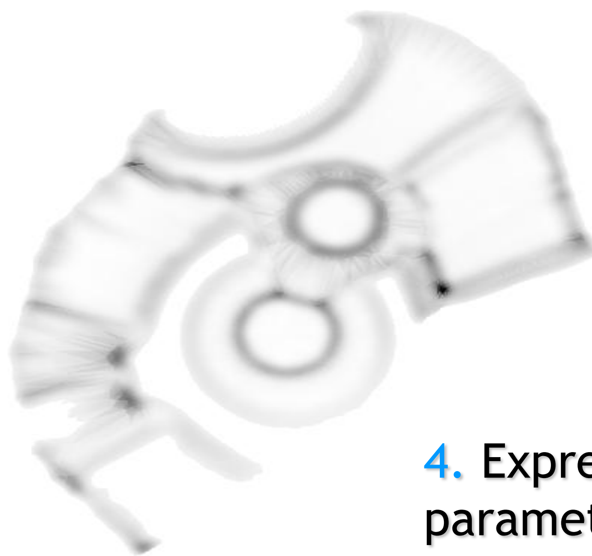
1. Original model



2. Measure density



3. Flatten it



4. Express density function in parameter space
5. Resample this function

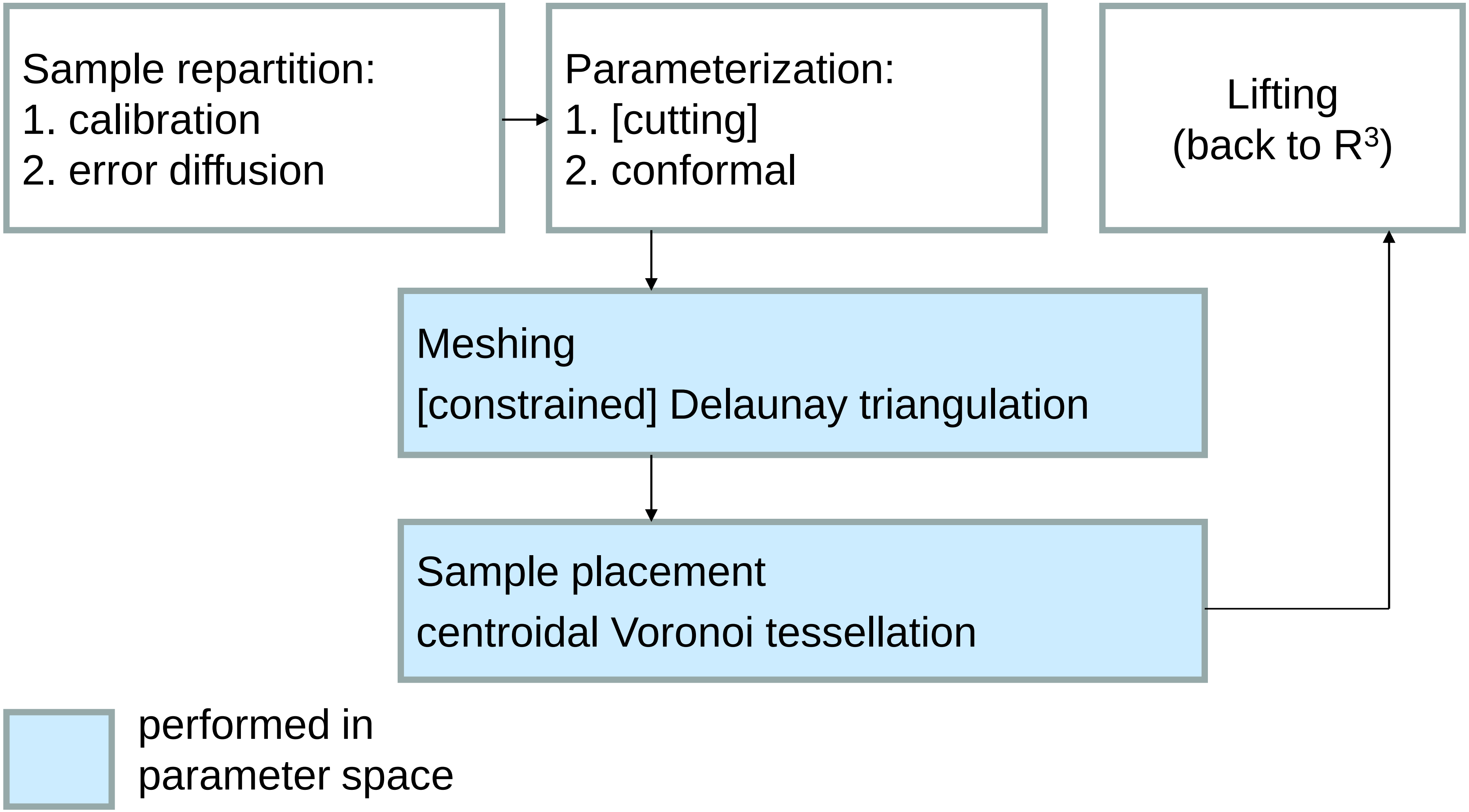
More precisely...

- Resample
  - in accordance with a density function
  - isotropic
- Match sample budget

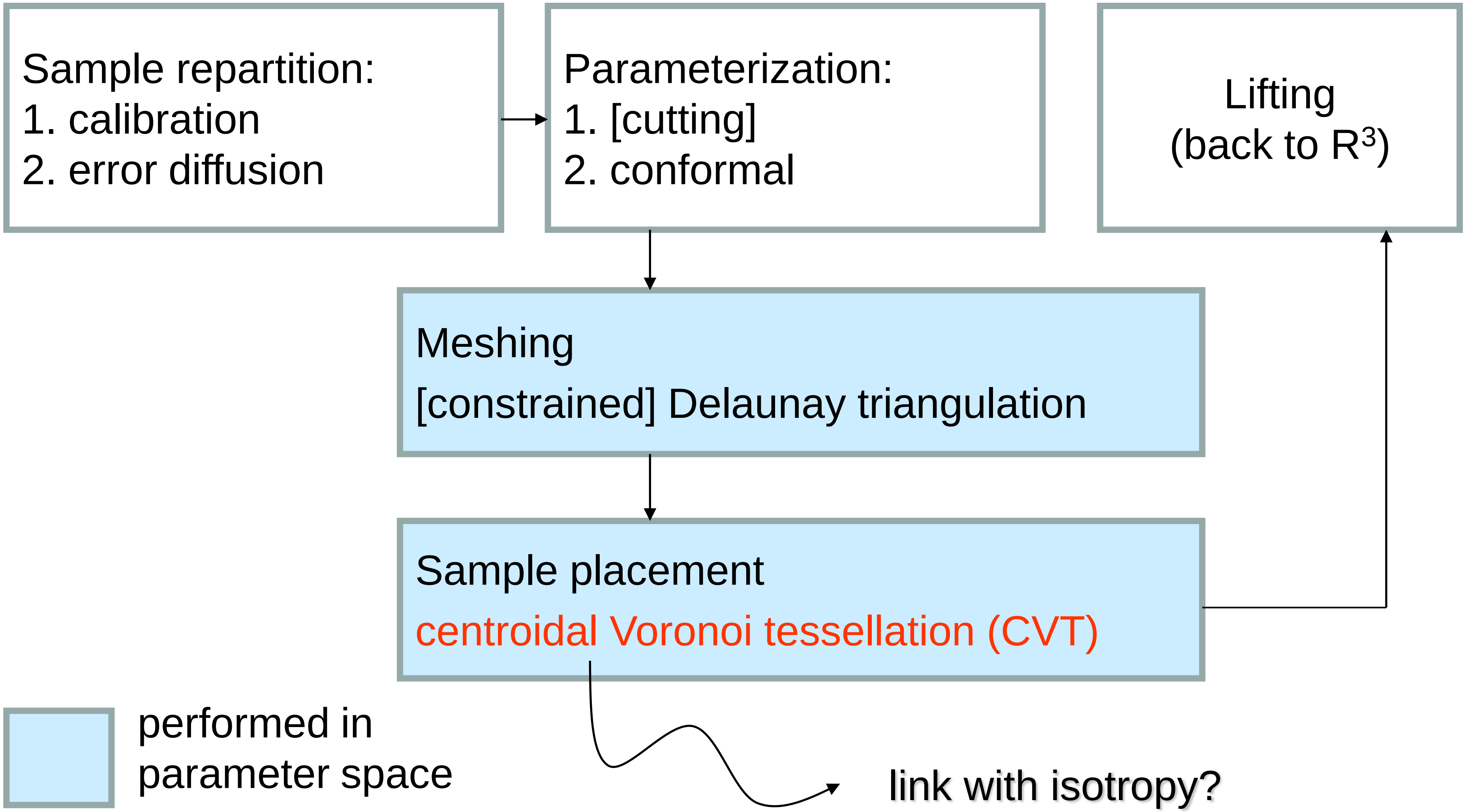
Solution

  - resample in parameter space
  - use good parameterization
  - compensate for distortion

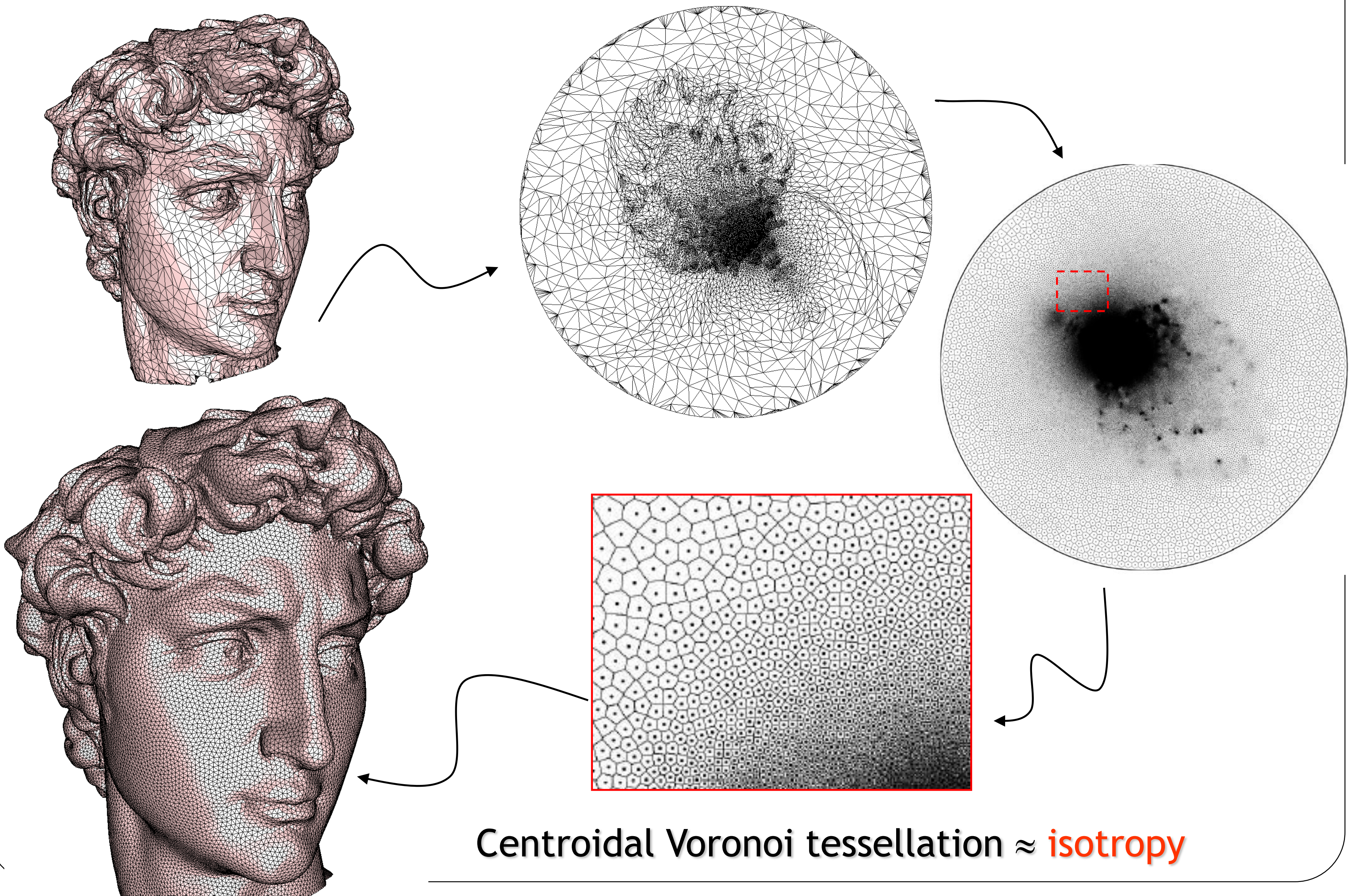
Remeshing Pipeline



Remeshing Pipeline



Key idea



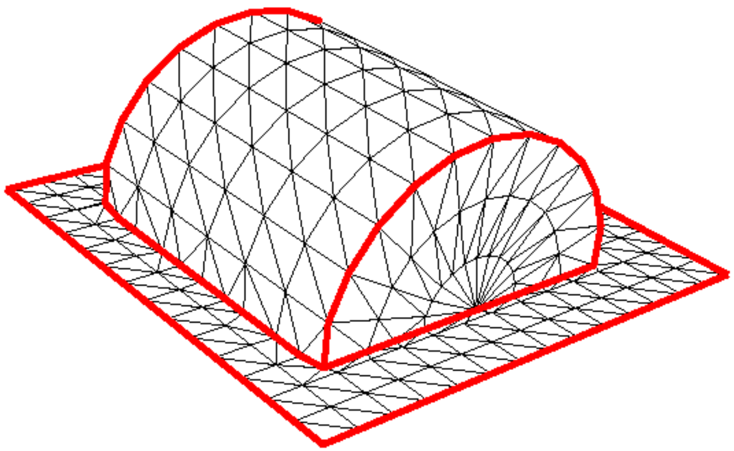
Centroidal Voronoi tessellation ≈ isotropy

Preliminaries

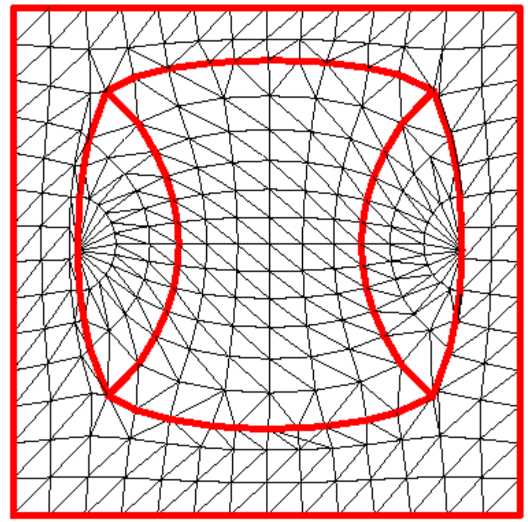
Input

- Triangle surface mesh with:
  - tagged feature edges
  - tagged corners
  - density function on:
    - feature edges (sharp,boundary,cut)
    - facets
- Vertex budget (#samples)
- Note:
  - the user *specifies* a density function
  - we focus on resampling & remeshing

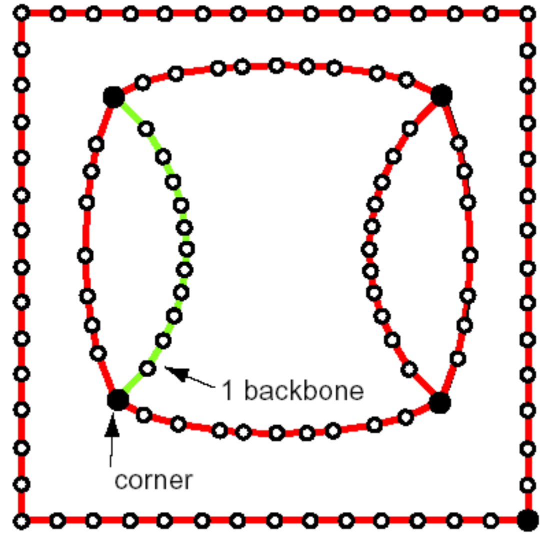
Feature (edges) skeleton



Original model



Parameterization and tagged edges



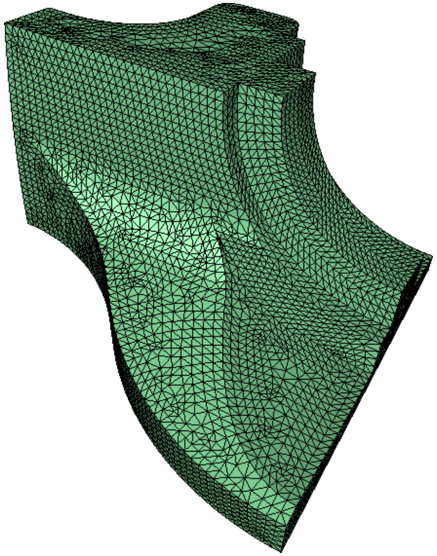
Feature skeleton

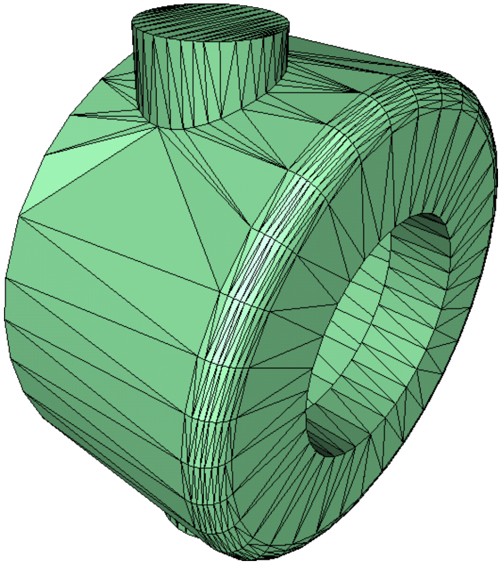
0-manifolds: corners

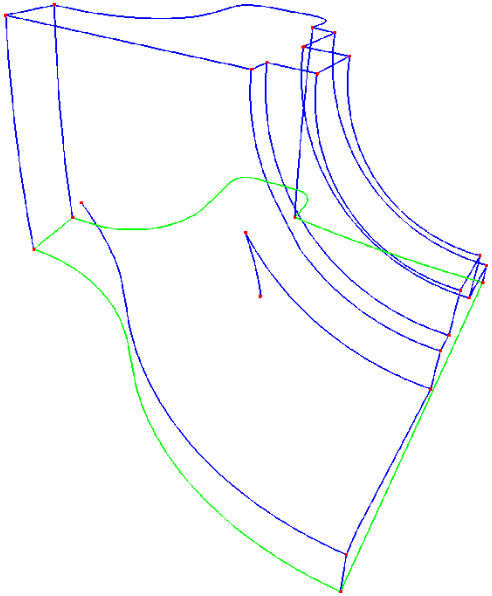
1-manifolds: backbone: feature edges chained together

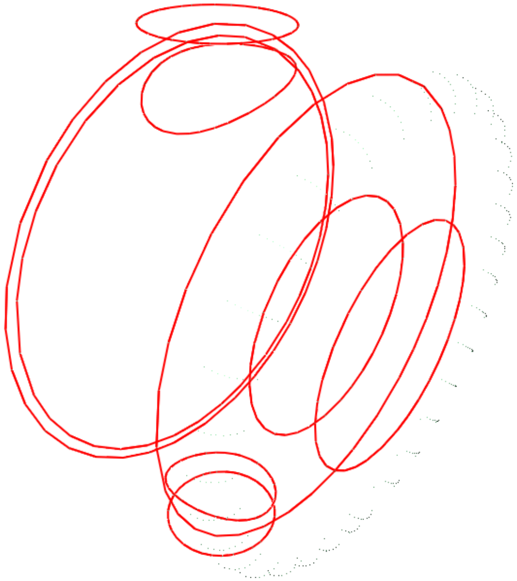
feature skeleton: corners + backbones

More feature skeletons

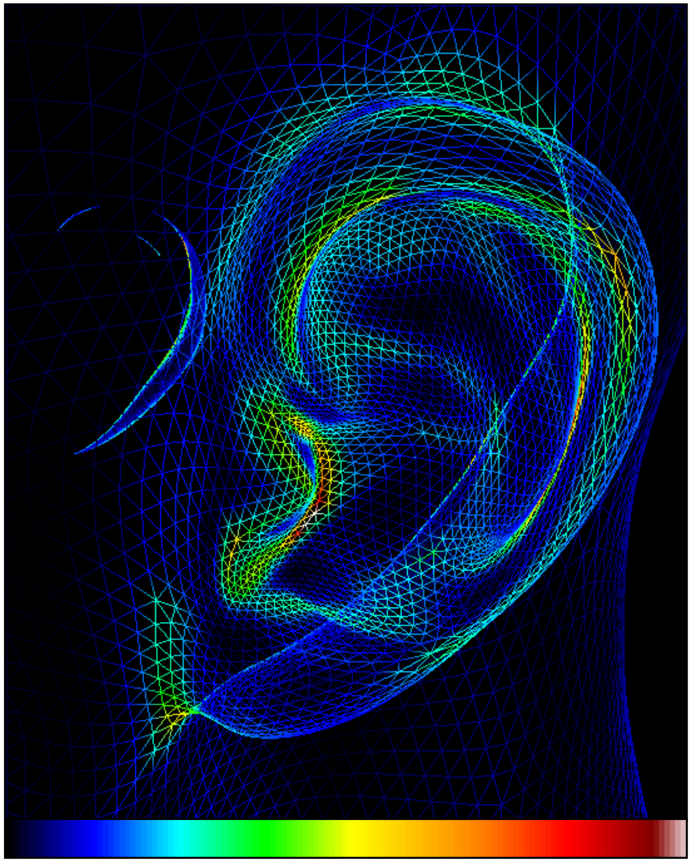








Example of density function



1. Curvature related density function

Discrete Differential-Geometry Operators

for Triangulated 2-Manifolds. [Meyer, Desbrun, Schröder, Barr]

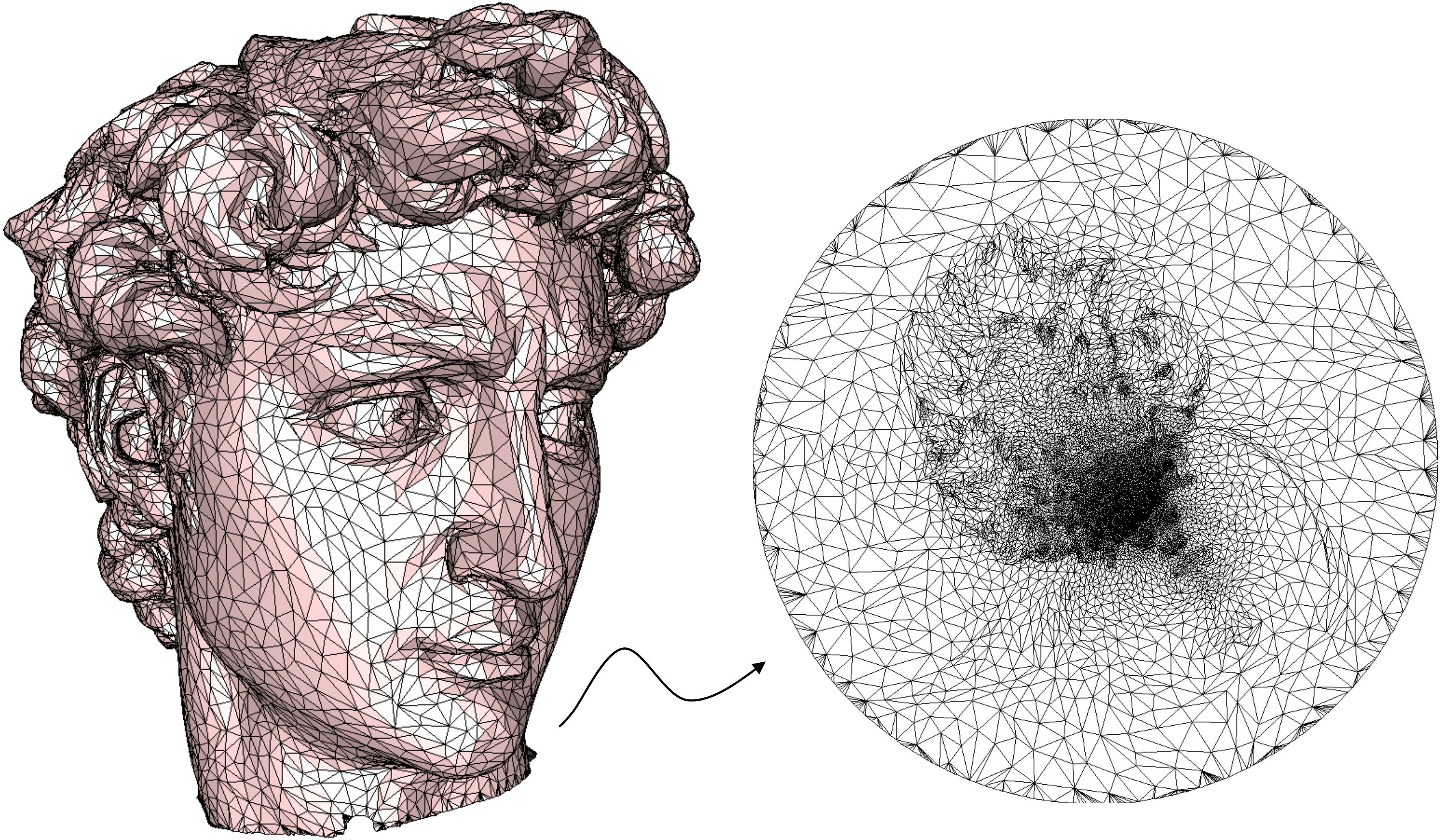
Algorithm

1. parameterization
2. sampling
3. meshing
4. sample placement

-> slow, precise

Motivation
Previous work
Contributions
Algorithm
- parameterization
- sample repartition
- meshing
- sample placement

Parameterization

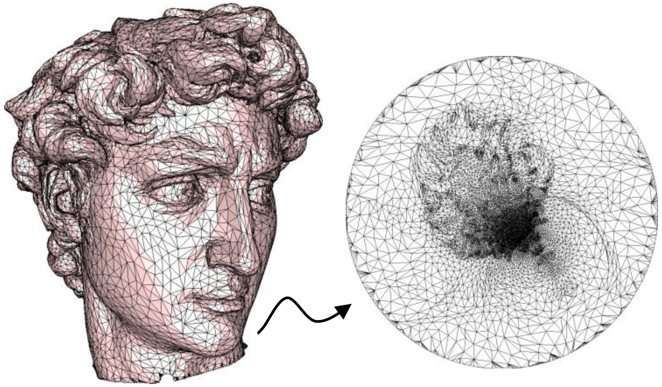


~remove embedding (back to manifold)

Parameterization

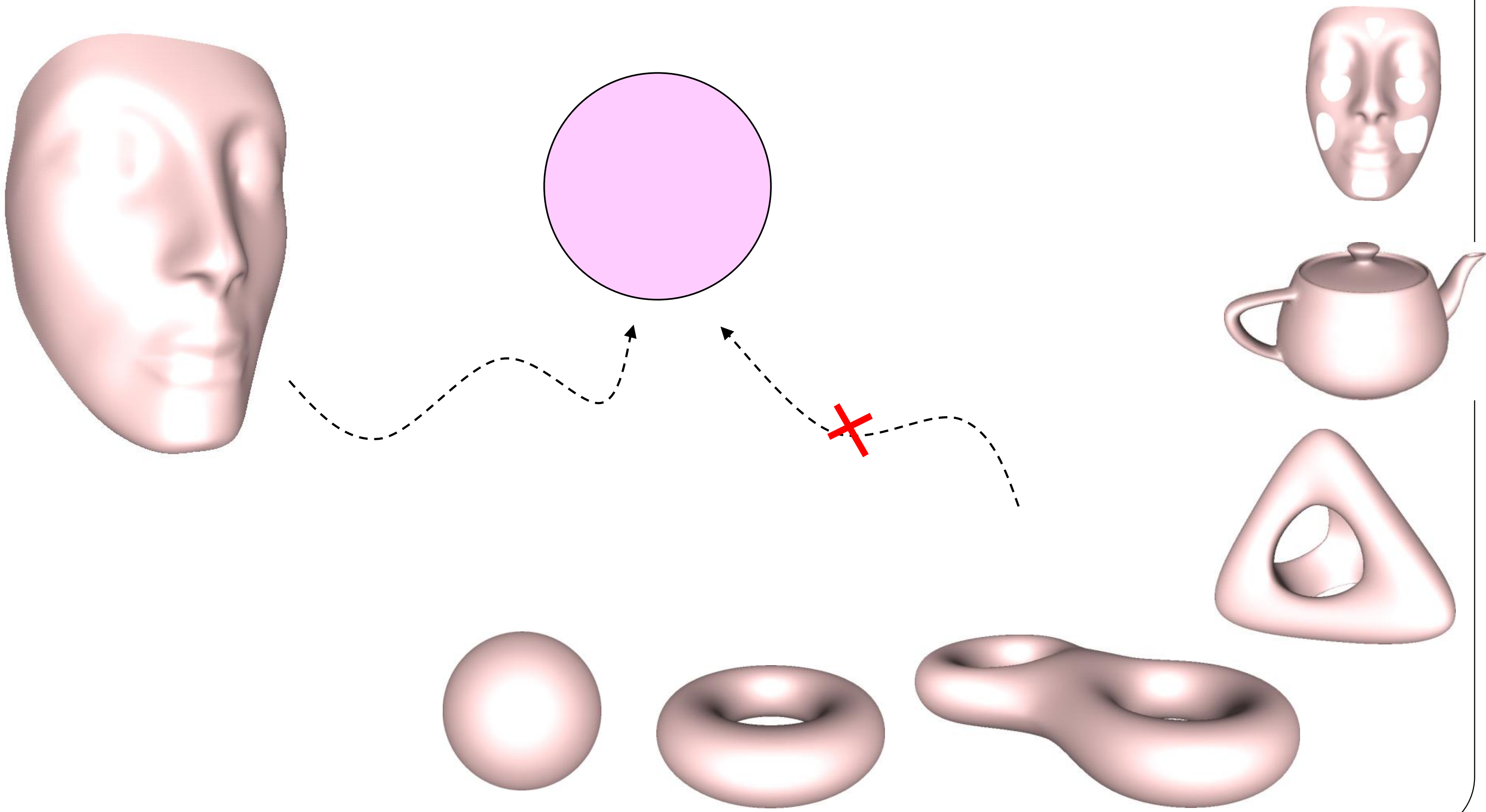
Motivations:

- Map signals
  - modulation
  - enrich geometry (GPU)
- Algorithms in flatland
  - analysis
  - sampling
  - meshing
  - etc.

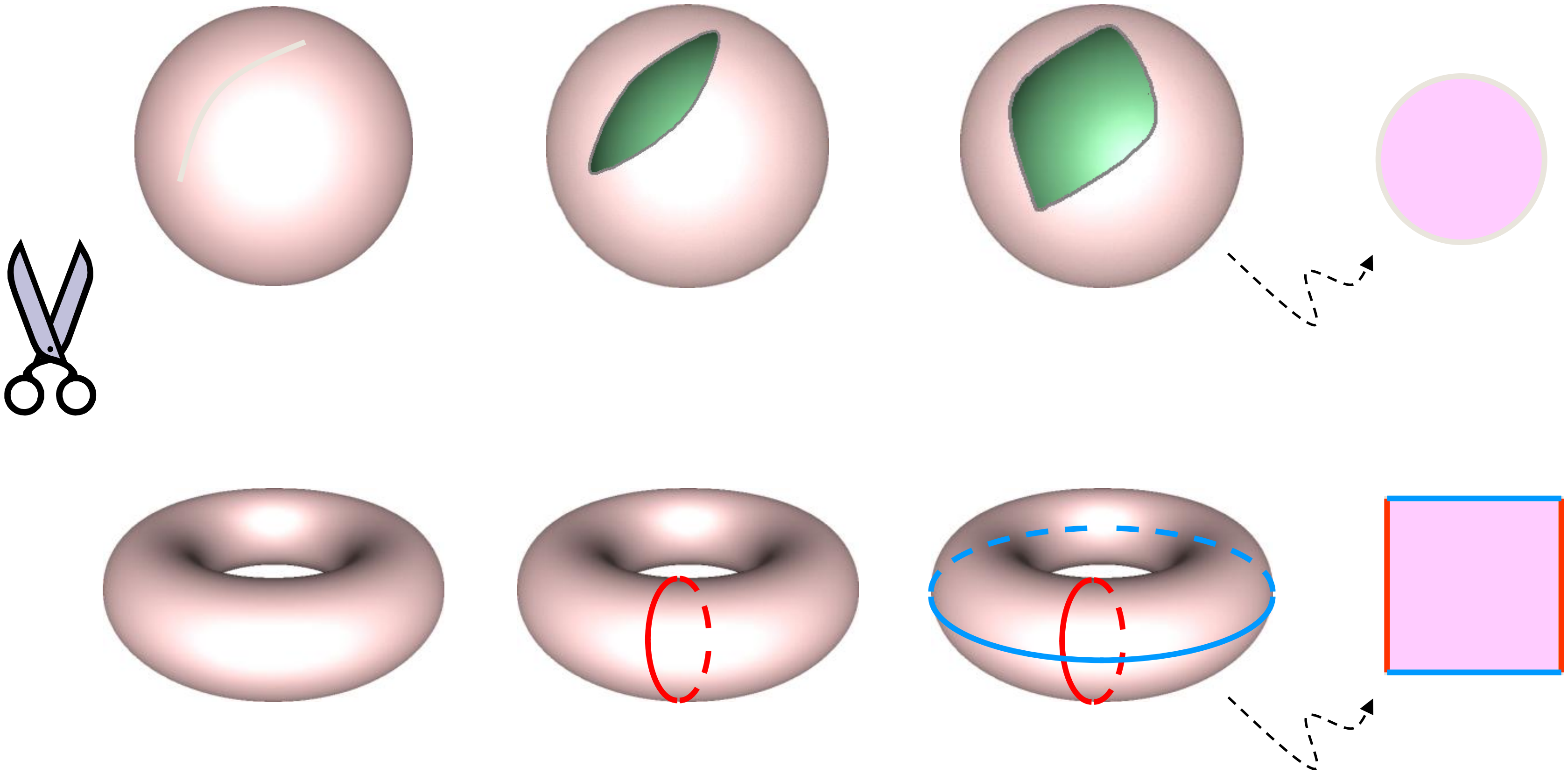


Parameterization

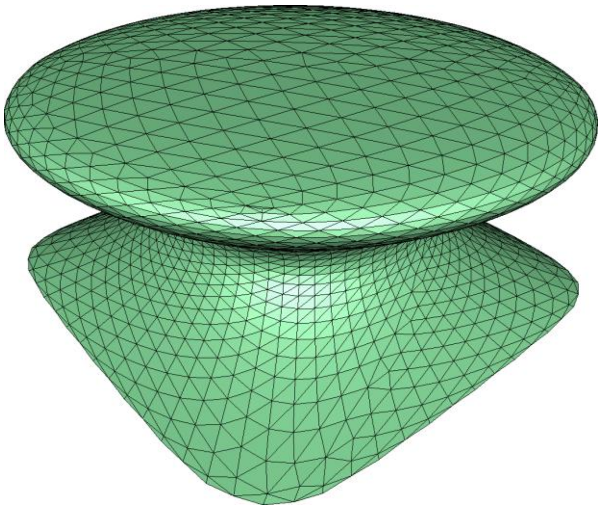
-> simple domain

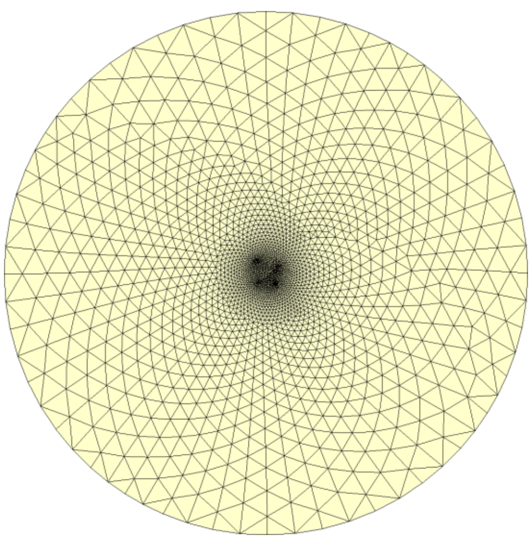


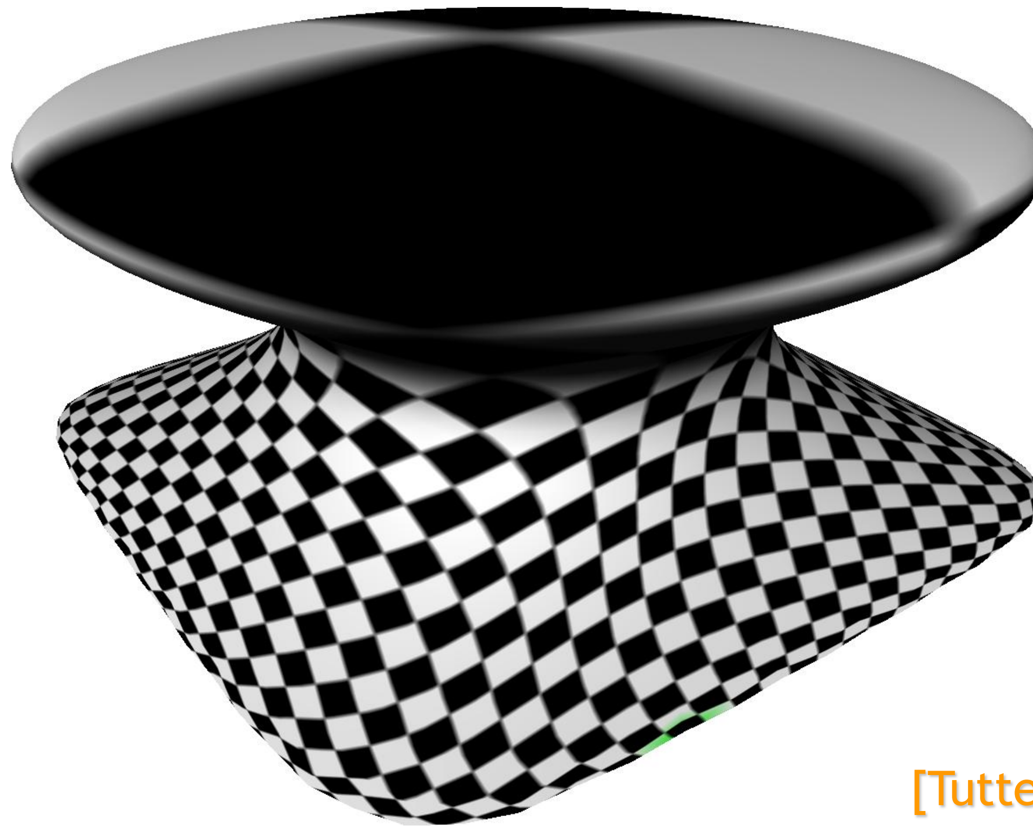
Surface cutting



Convex embedding

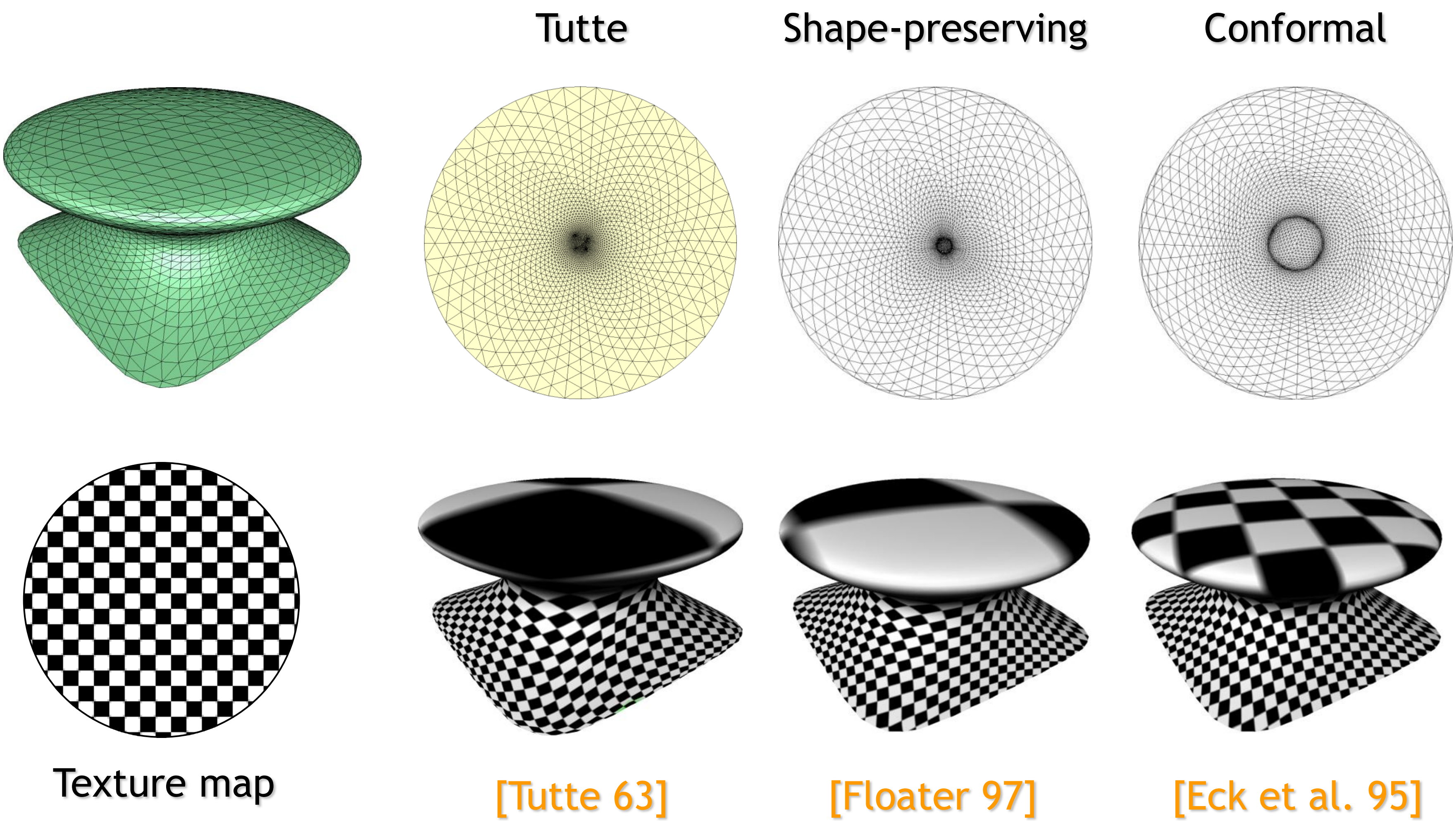




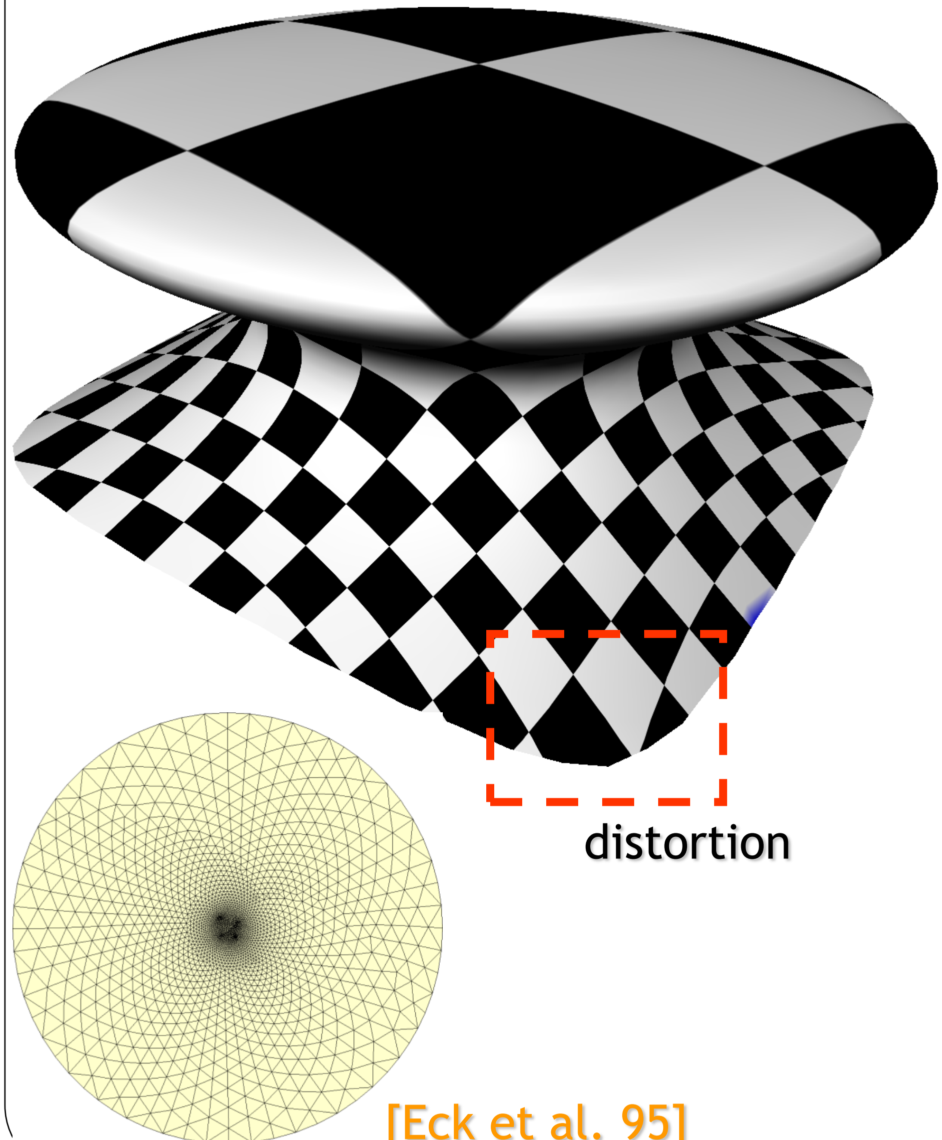


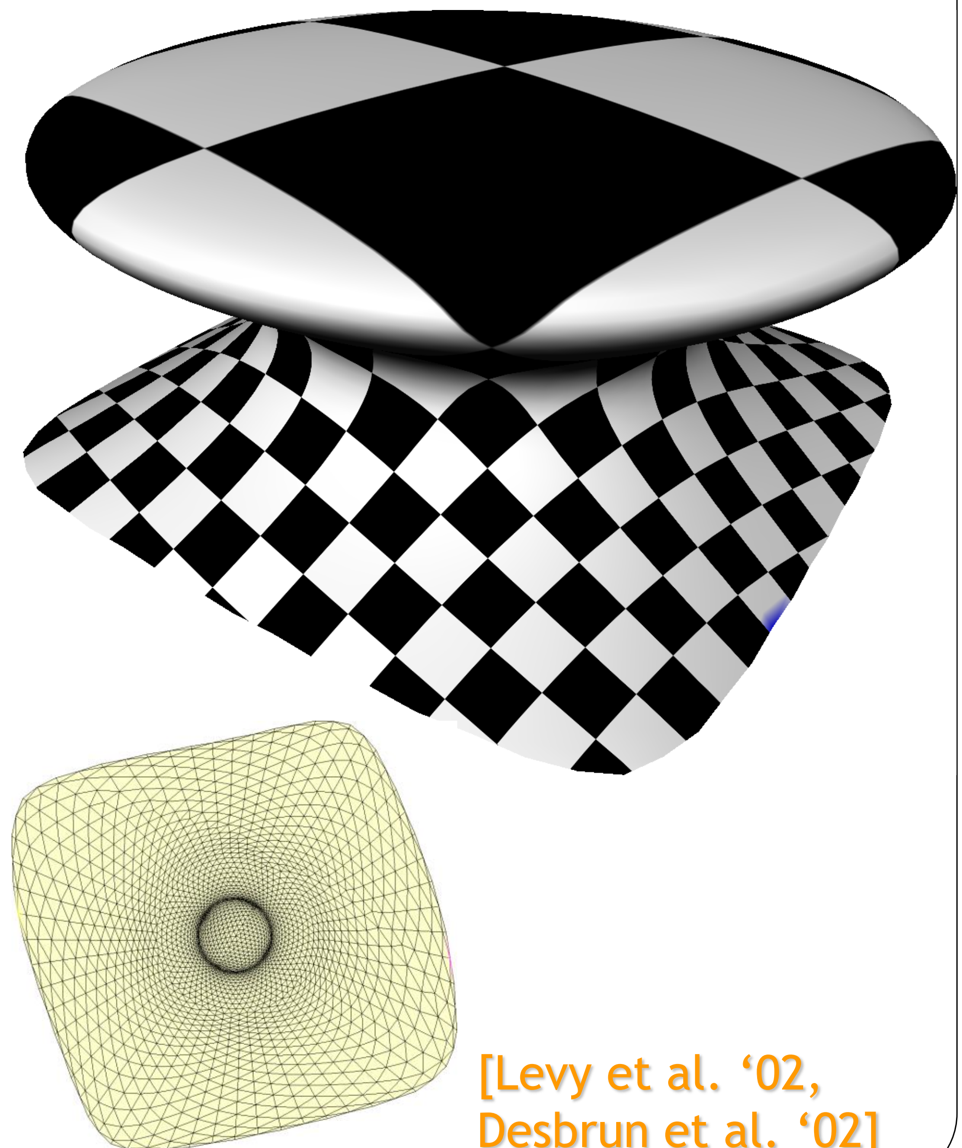
[Tutte]

More parameterizations

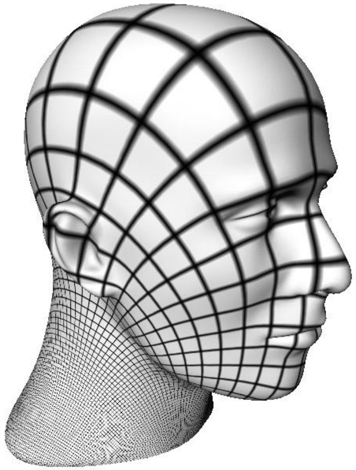


Conformal: fixed vs free boundary

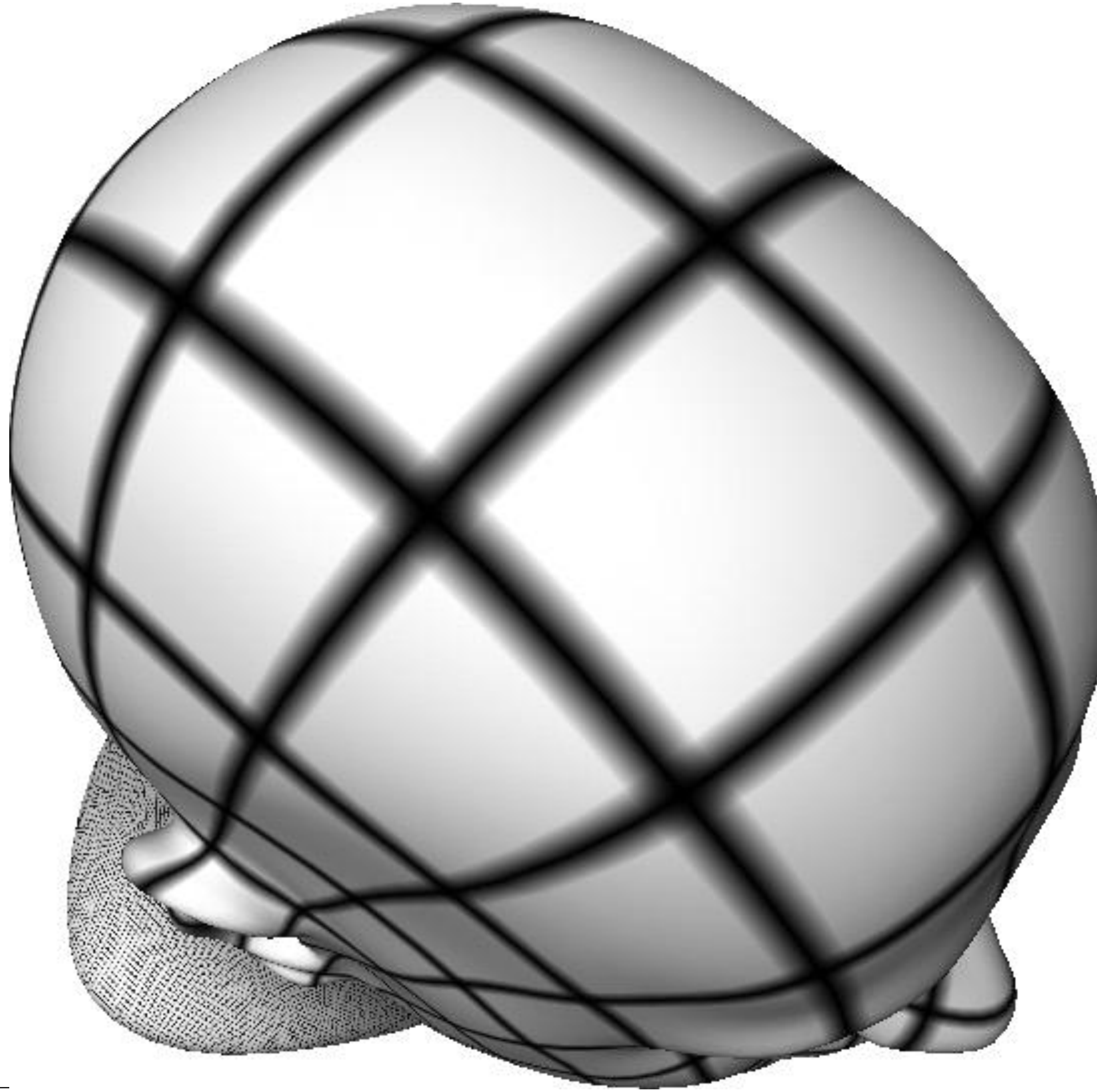




Preservation of angles

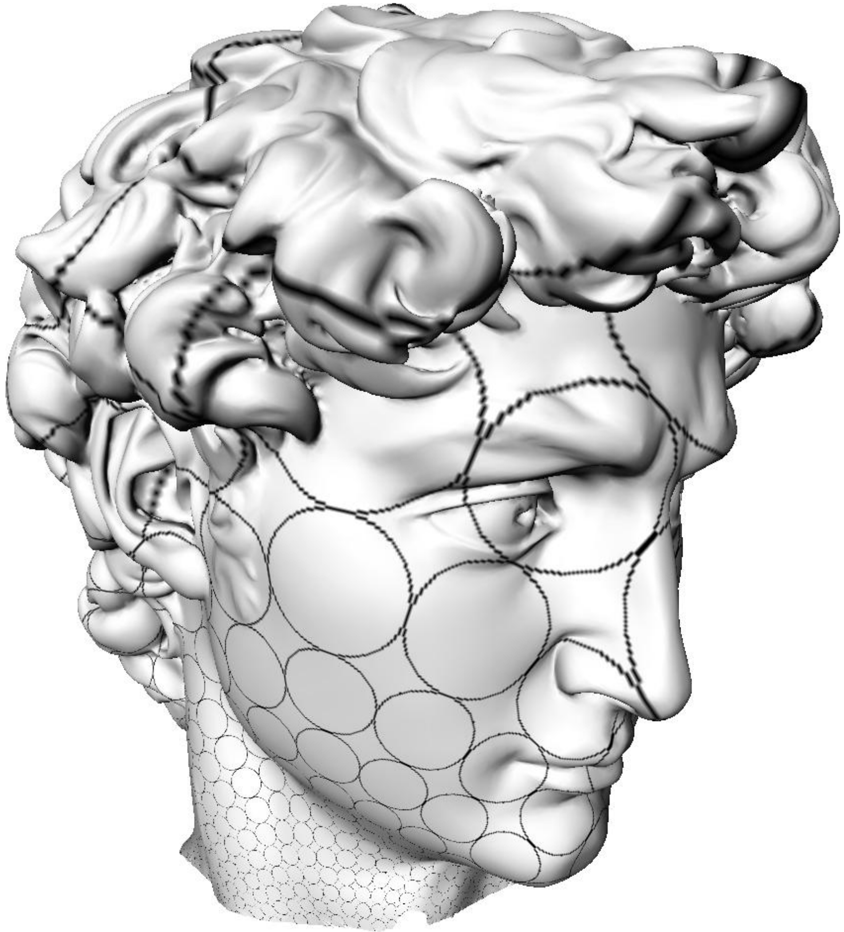


Isoparametric lines

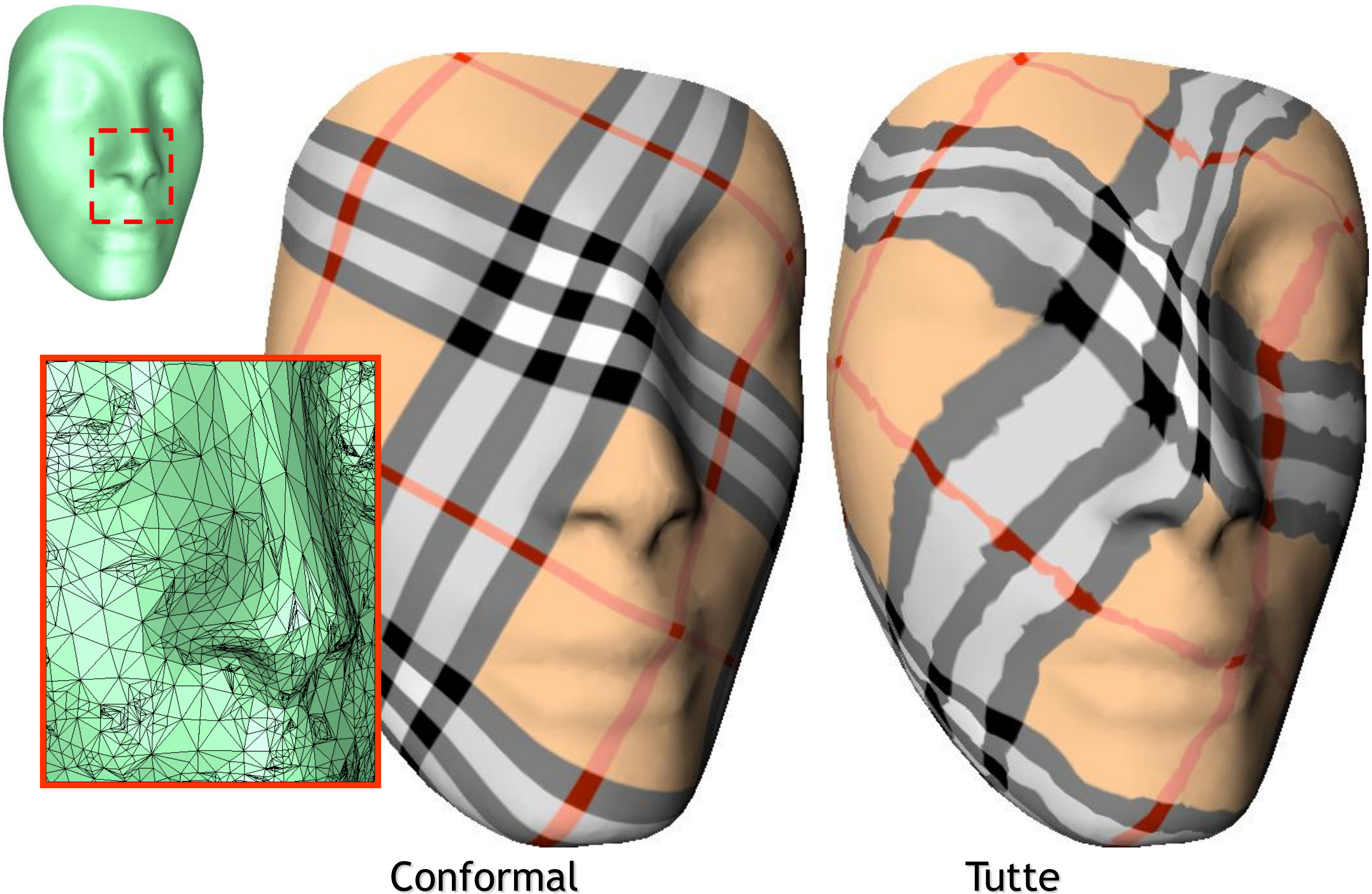


Conformality

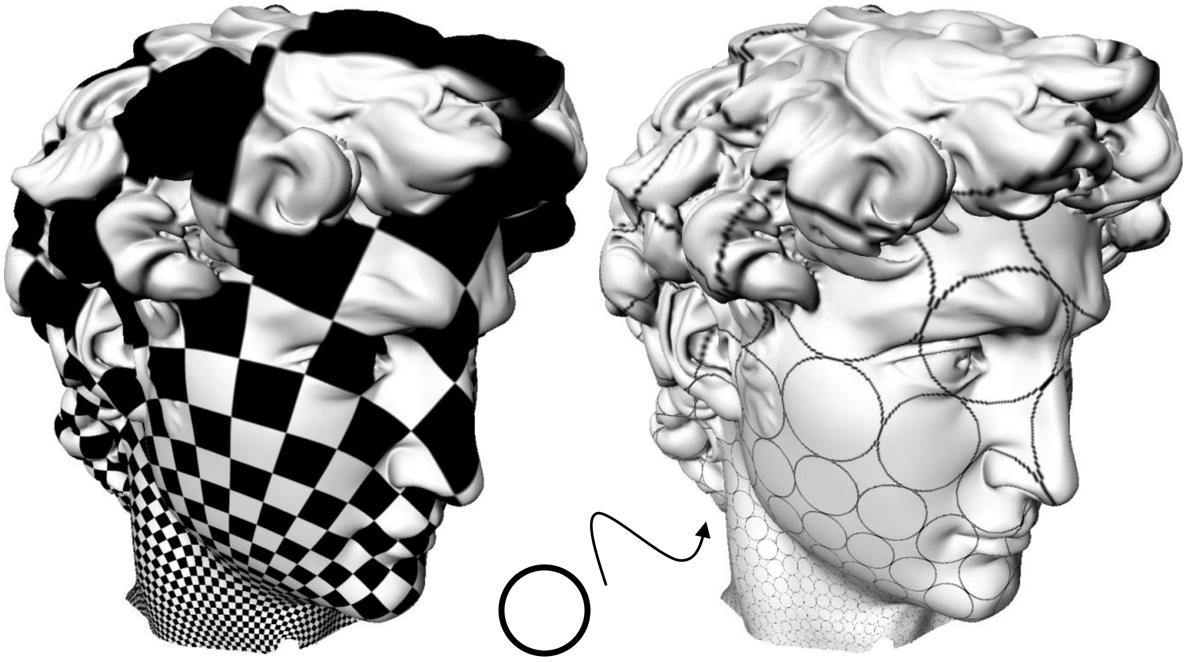
Angle-preserving + locally isotropic



Behavior w.r.t. sampling



"A well-shaped element in parameter space will not be deformed too much once lifted in embedding space"



Motivation

Previous work

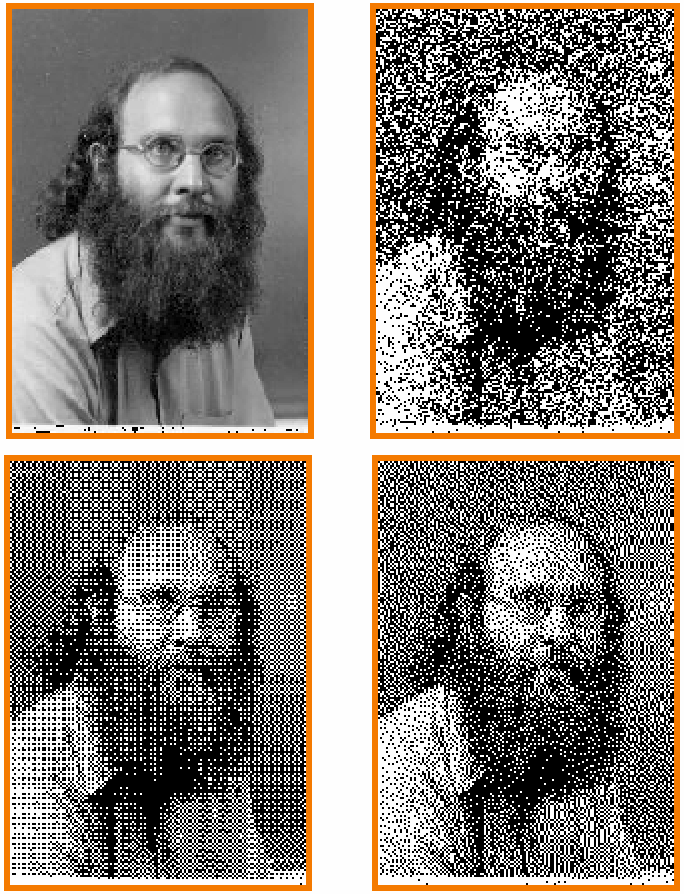
Contributions

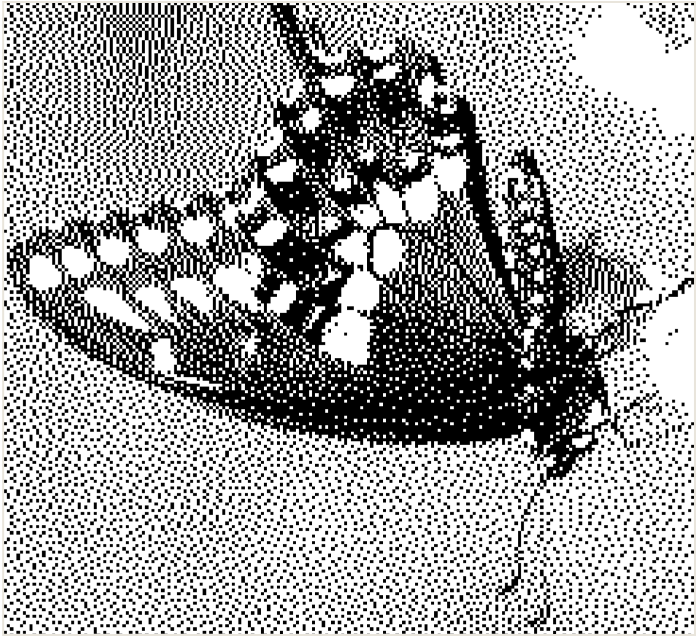
Algorithm

- parameterization
- sample repartition
- meshing
- sample placement

Sample repartition

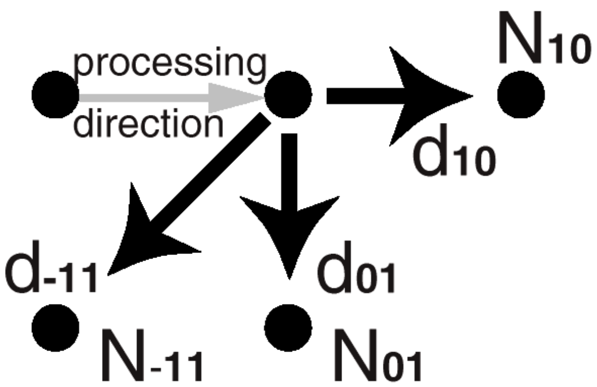
Use core principle of error diffusion

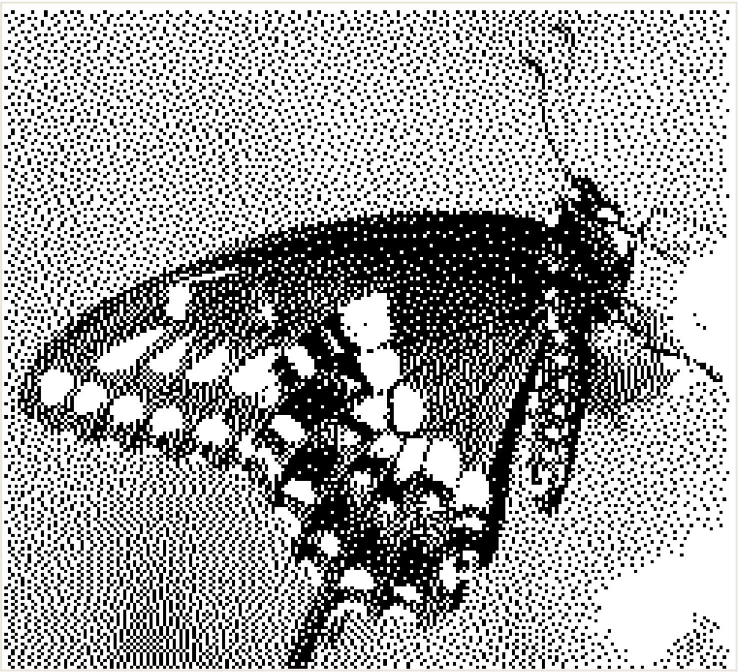




[Ostromoukhov '01]

Error diffusion



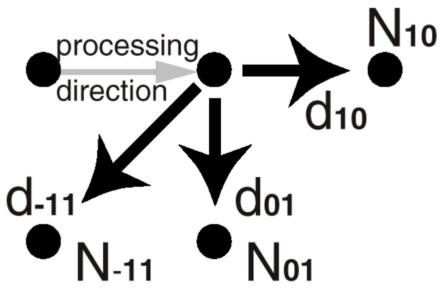


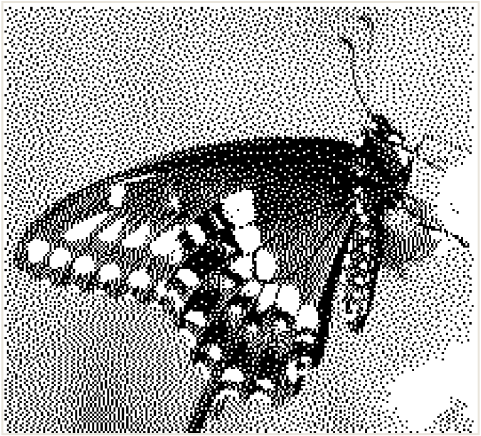
Two main components:

1. Processing path (proximity+absence of teleport)
2. Coefficients of diffusion

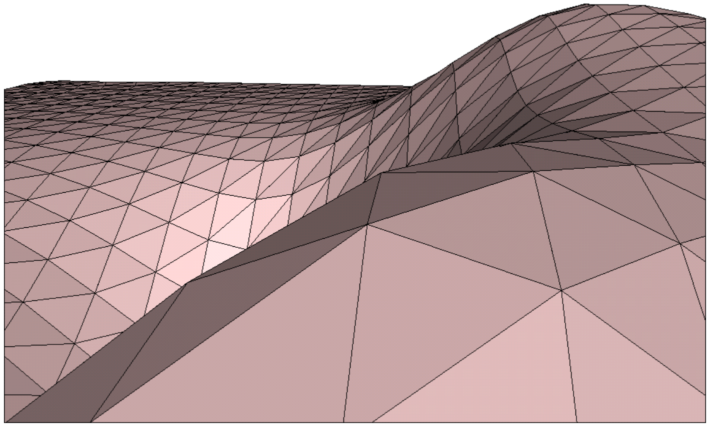
Sample repartition

Idea: generalize error diffusion over discrete surfaces





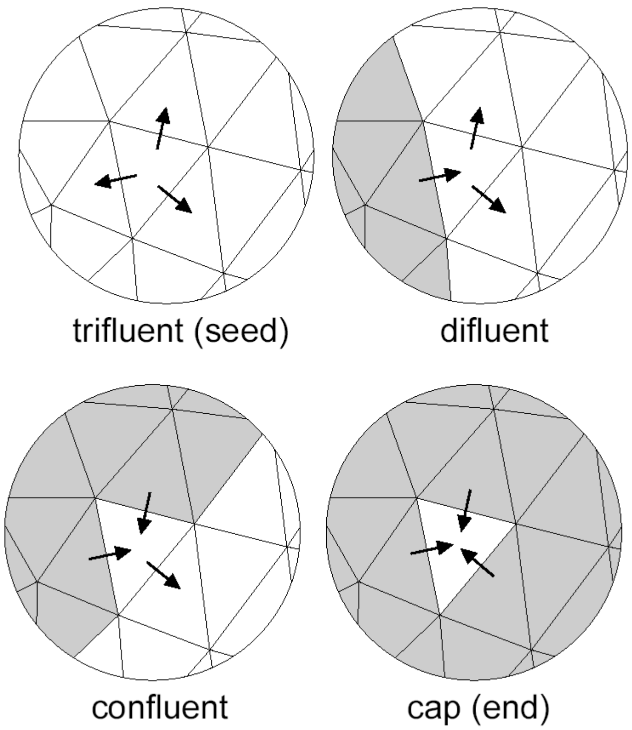
Image

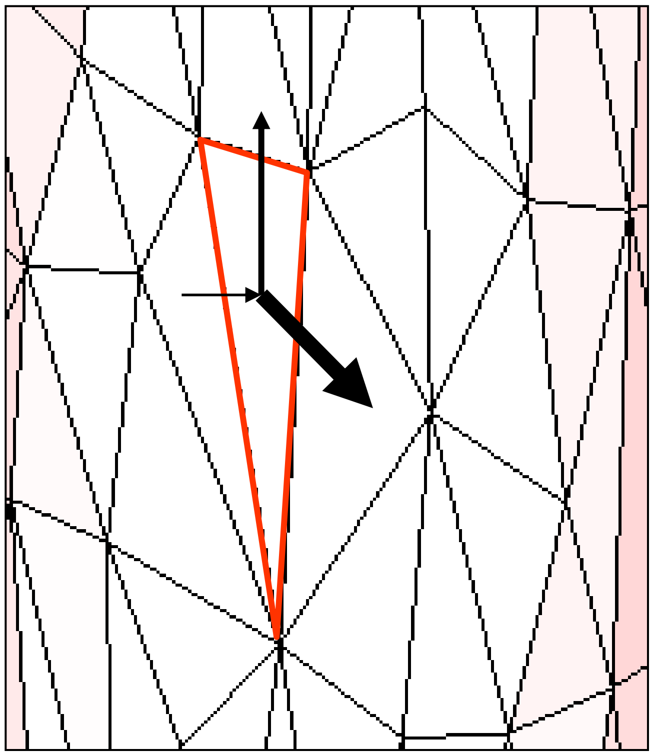


Model: triangle mesh + feature skeleton

2D Error diffusion

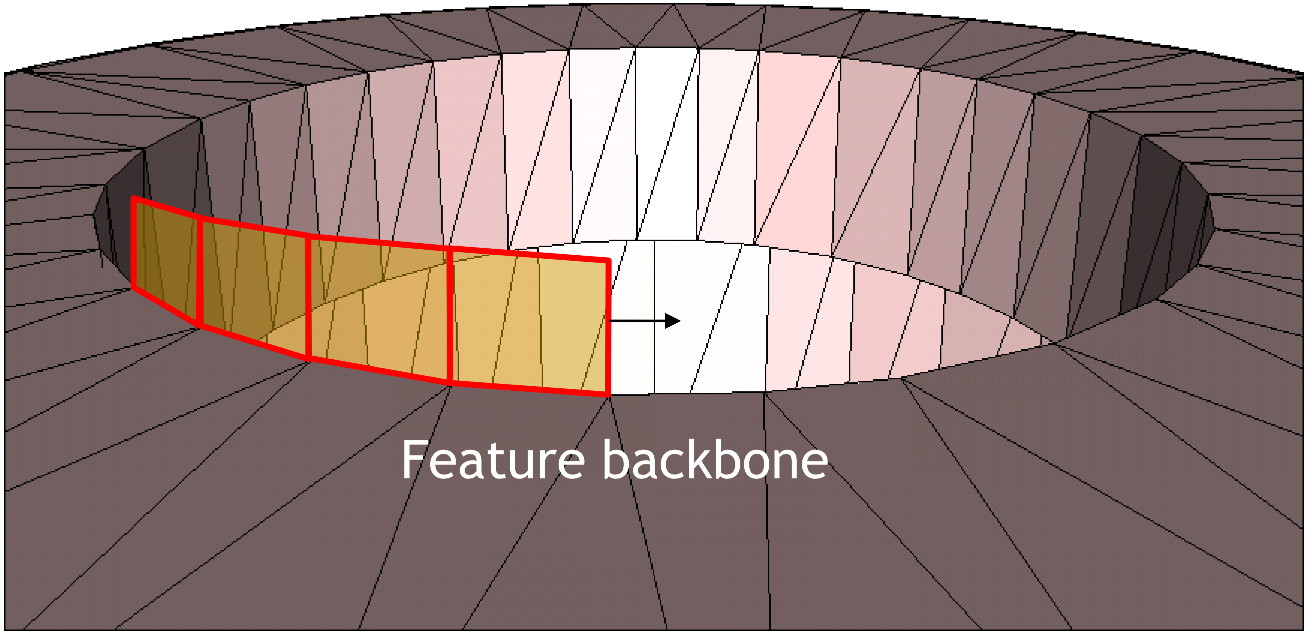
- Compute processing path on triangles
- Coefficients for error diffusion: organize fluency on triangles (edge lengths)

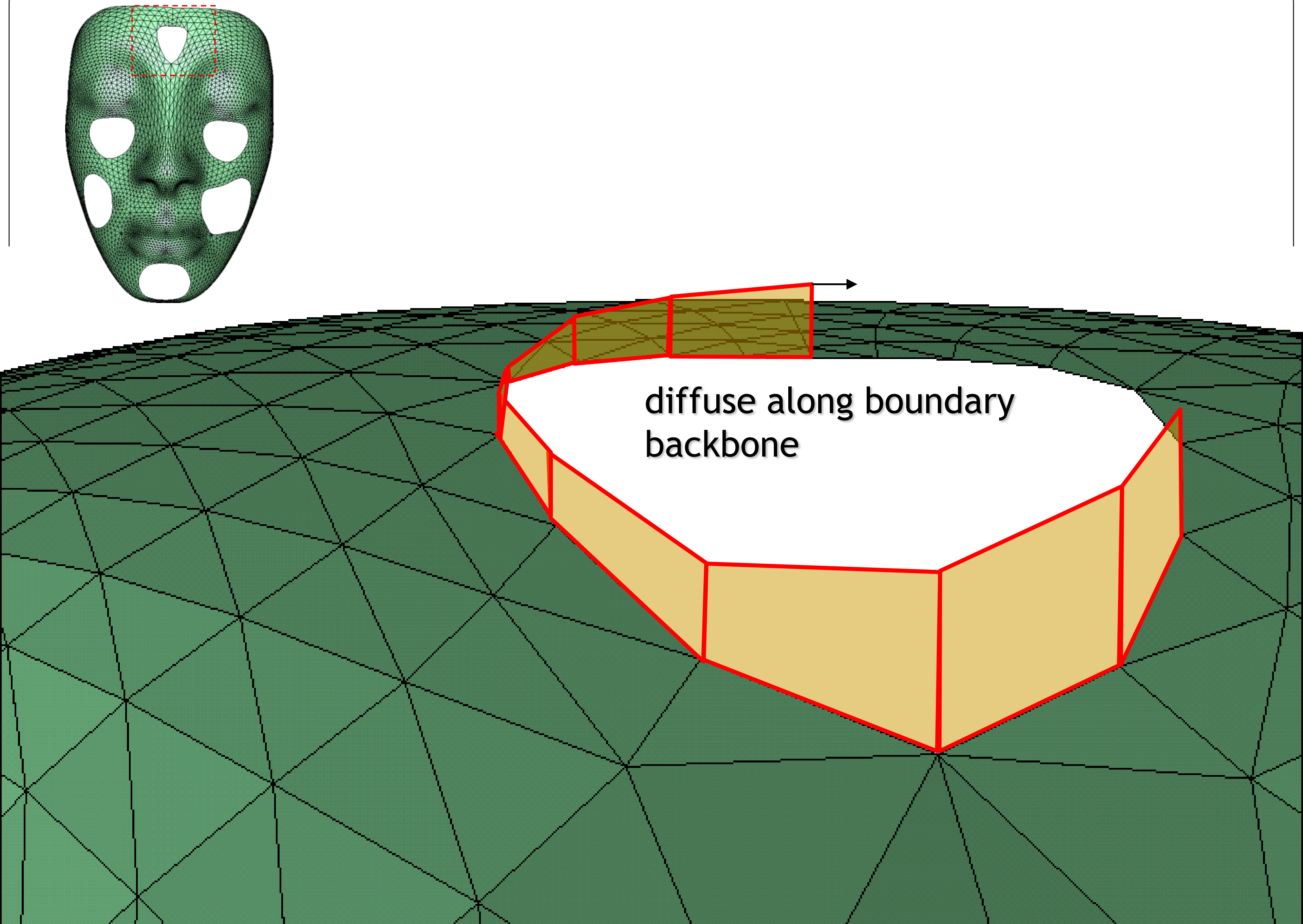




1D Error diffusion

- Given processing path on backbones
  - fluency on edges




diffuse along boundary backbone

Motivation

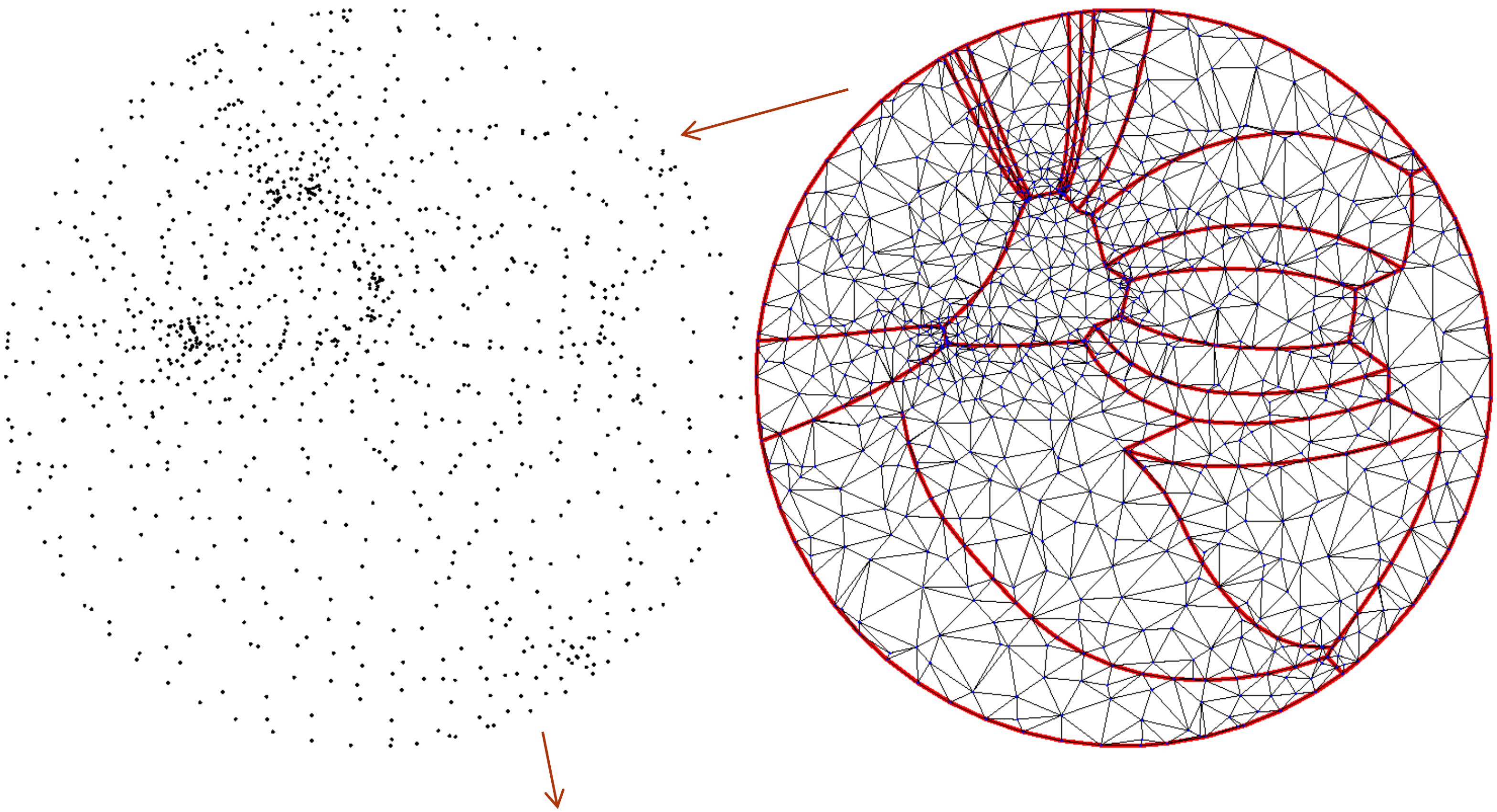
Previous work

Contributions

Algorithm

- parameterization *conformal*
- sample repartition *error diffusion*
- meshing
- sample placement

Meshing



Constrained Delaunay triangulation
in parameter space
[CGAL] -> solves robustness issues

Motivation

Previous work

Contributions

Algorithm

- sample repartition *error diffusion*
- parameterization *conformal*
- meshing *Delaunay*
- sample placement

Sample placement

Given a bounded domain and a density function,

sampling

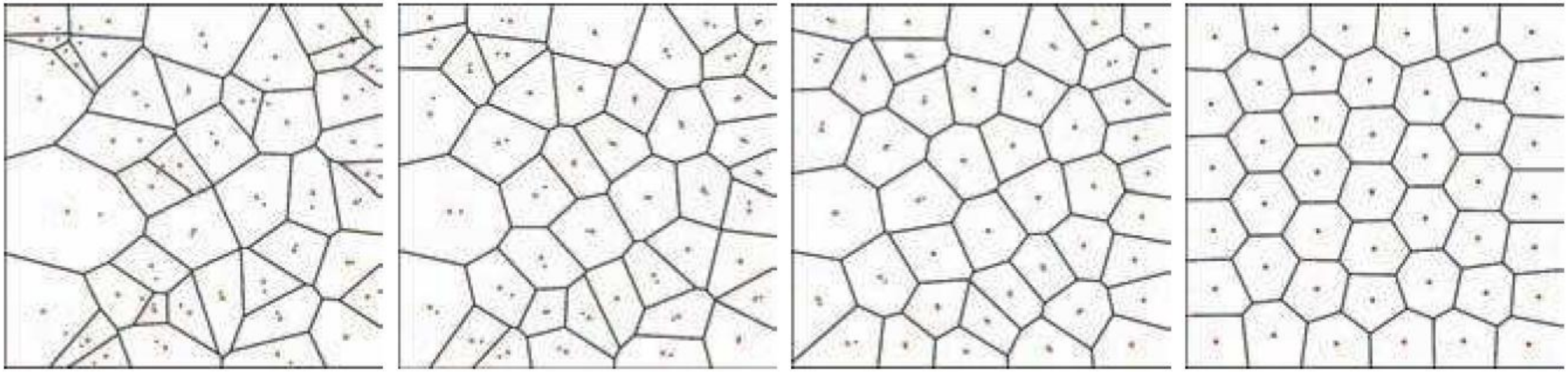
=

- partitioning the domain
- repartitioning the density function among a set of samples

Delaunay Triangulation and Voronoi Diagram

- Delaunay Triangulation
- Voronoi Diagram
- Centroidal Voronoi Diagram
  - Centroids: $$\mathbf{C}_i = \frac{\int_A \mathbf{x}\rho(\mathbf{x})dA}{\int_A \rho(\mathbf{x})dA}$$
  - To minimize: $\int_A \rho(\mathbf{x})|\mathbf{C}_i - \mathbf{x}|^2$
  - Can be computed by Lloyd's algorithm iteratively

Centroidal Voronoi diagram



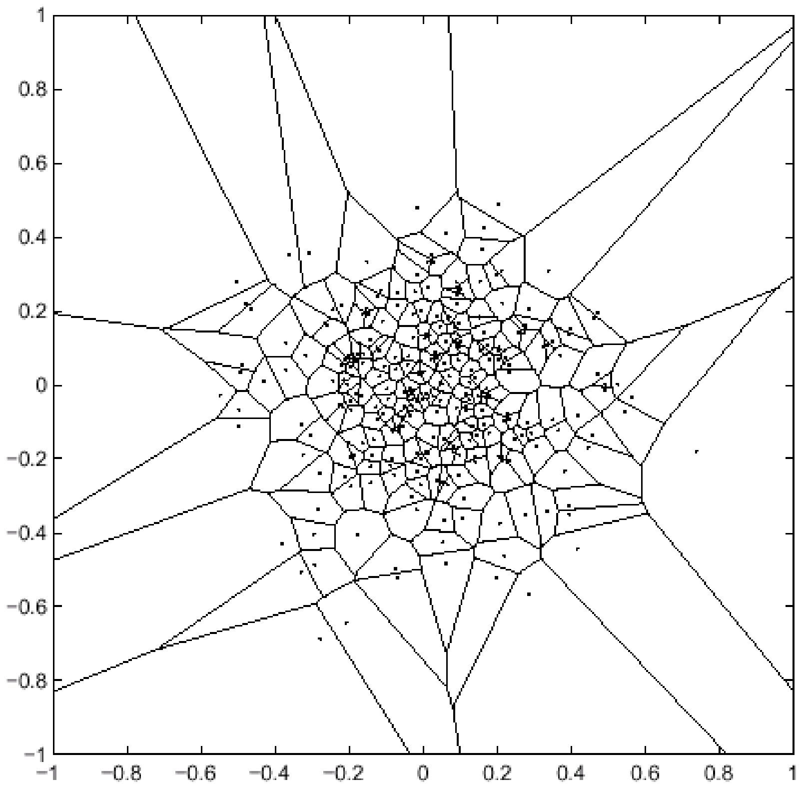
Ordinary Voronoi diagram

Centroidal Voronoi diagram

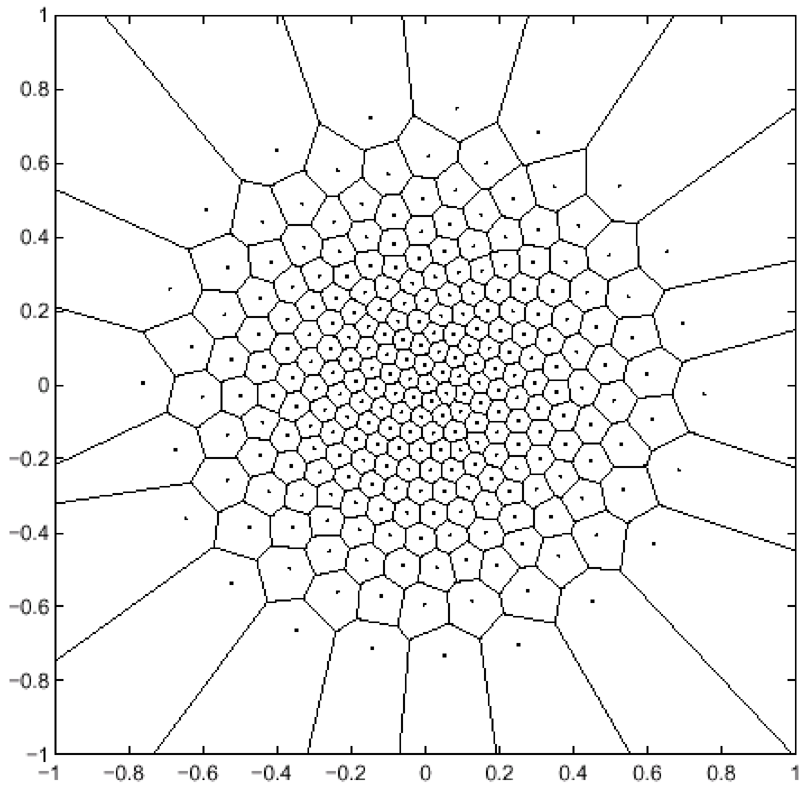
- sites
- centroids

Sites coincide with centroids
(center of mass)

Weighted Centroidal Voronoi diagram



Monte-Carlo



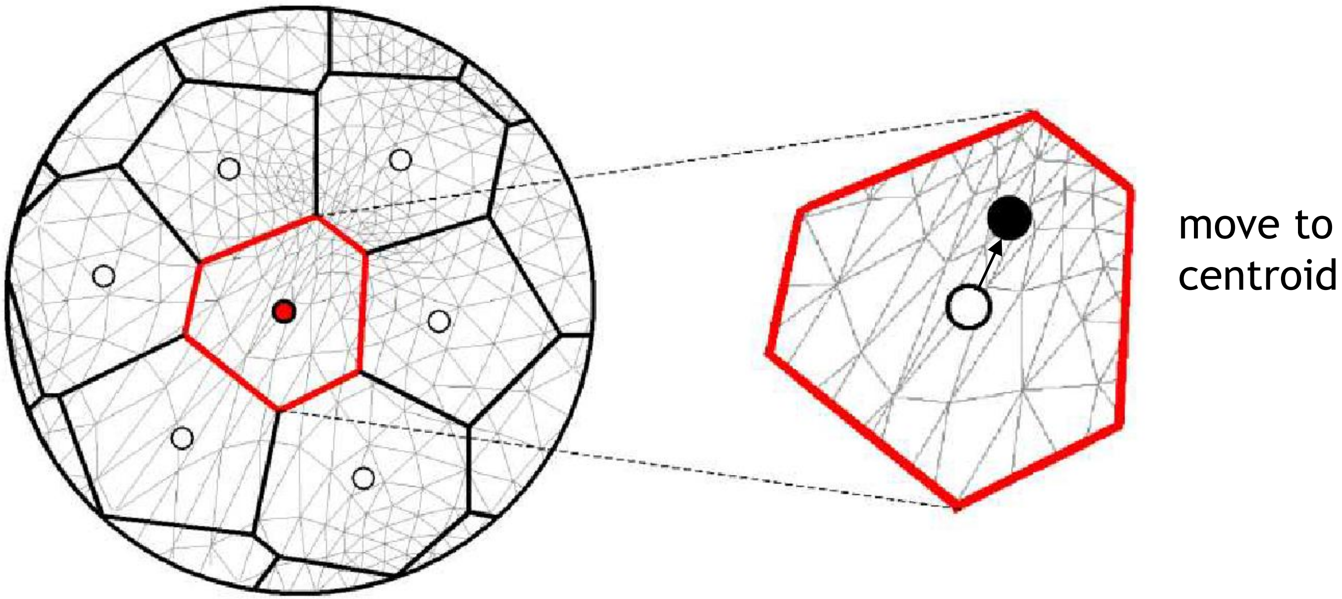
WCVD

Non-uniform density

Sample placement

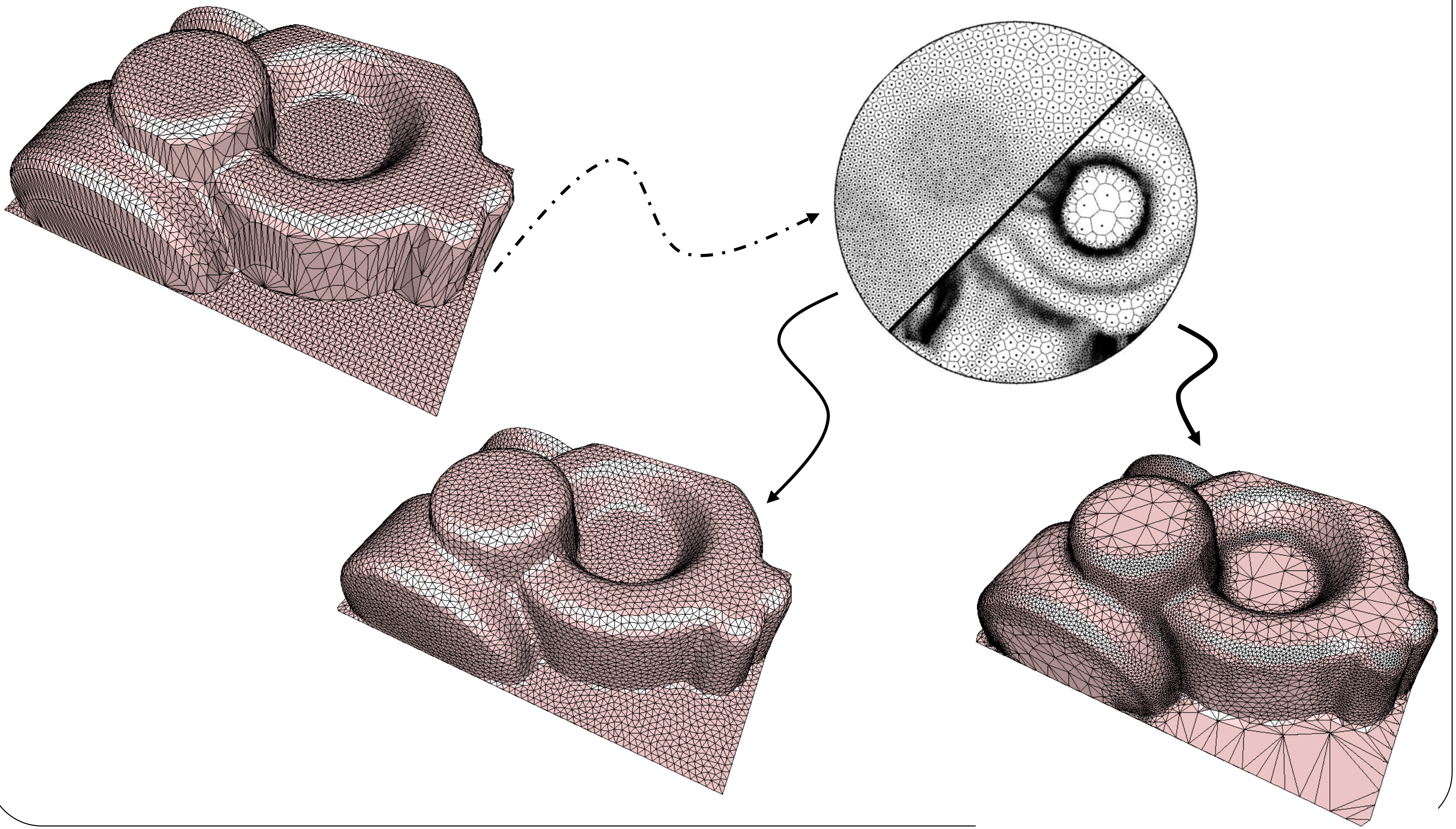
Two process sorted by increasing degrees of freedom:

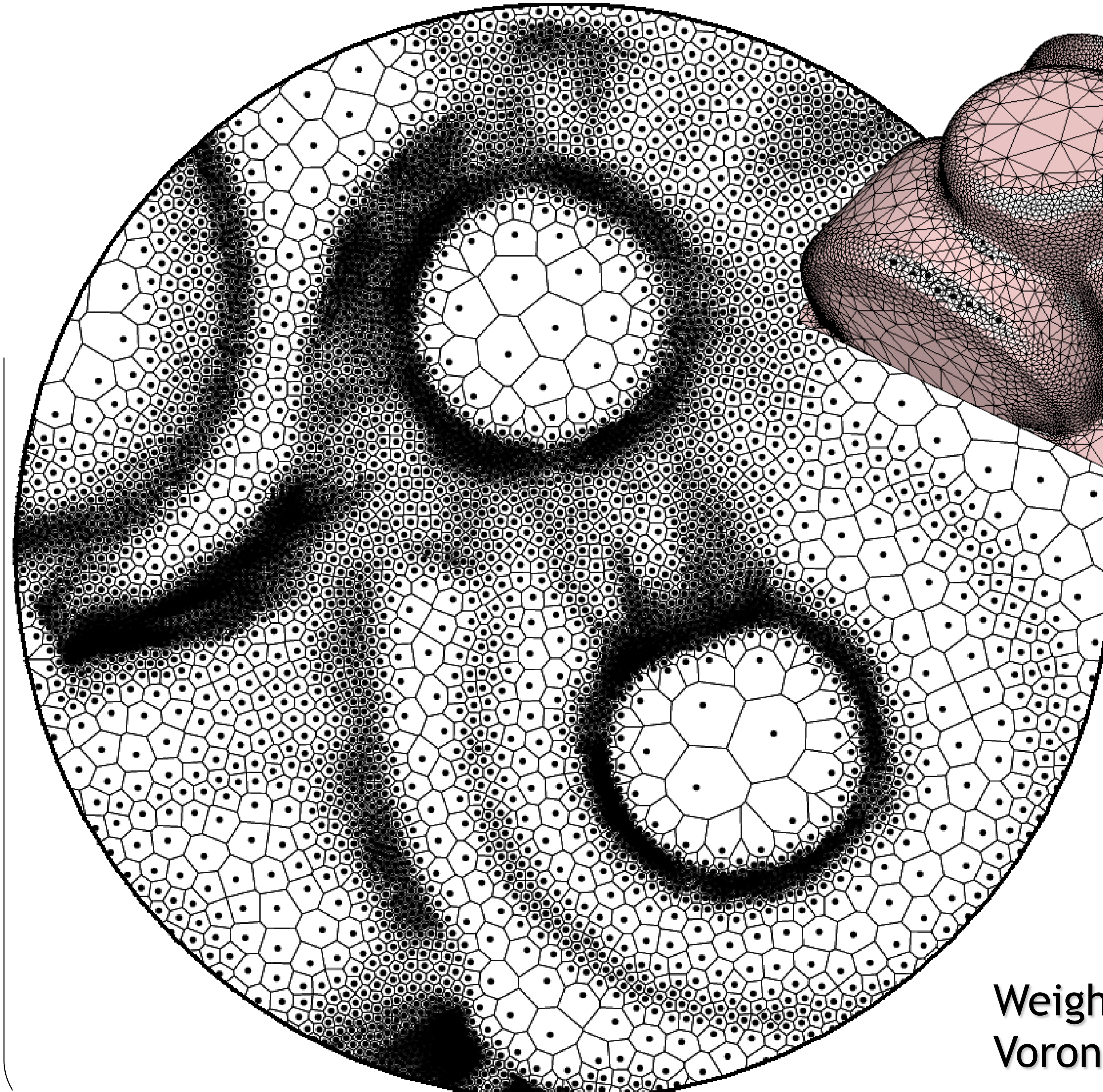
1. build 1D WCVD
2. build 2D WCVD via Lloyd relaxation

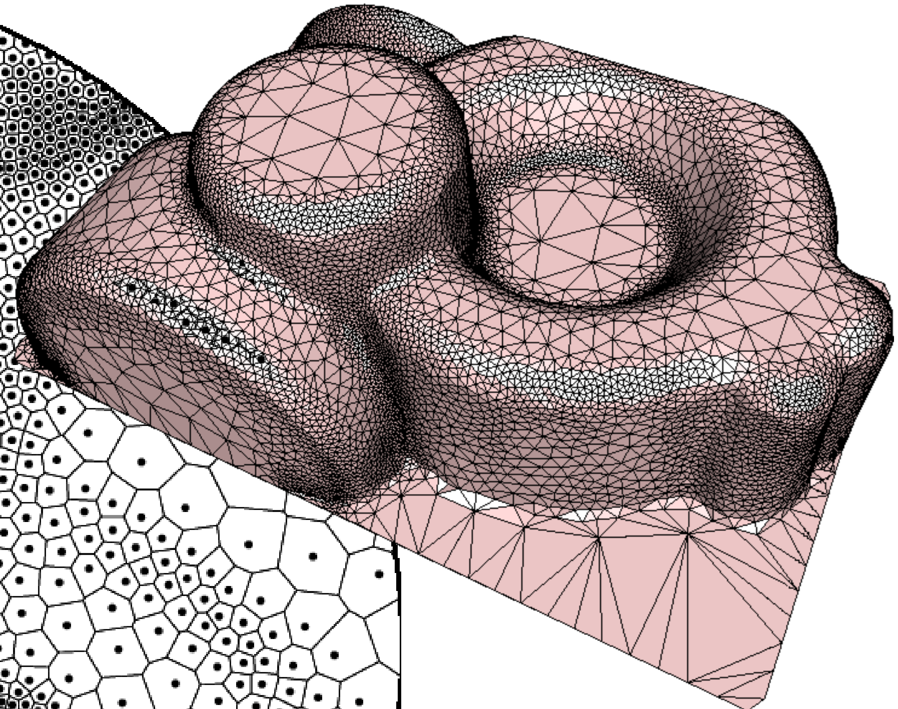


Motivation
Previous work
Contributions
Algorithm
Results
Limitations
Conclusions
Future Work

Uniform vs curvature-adapted

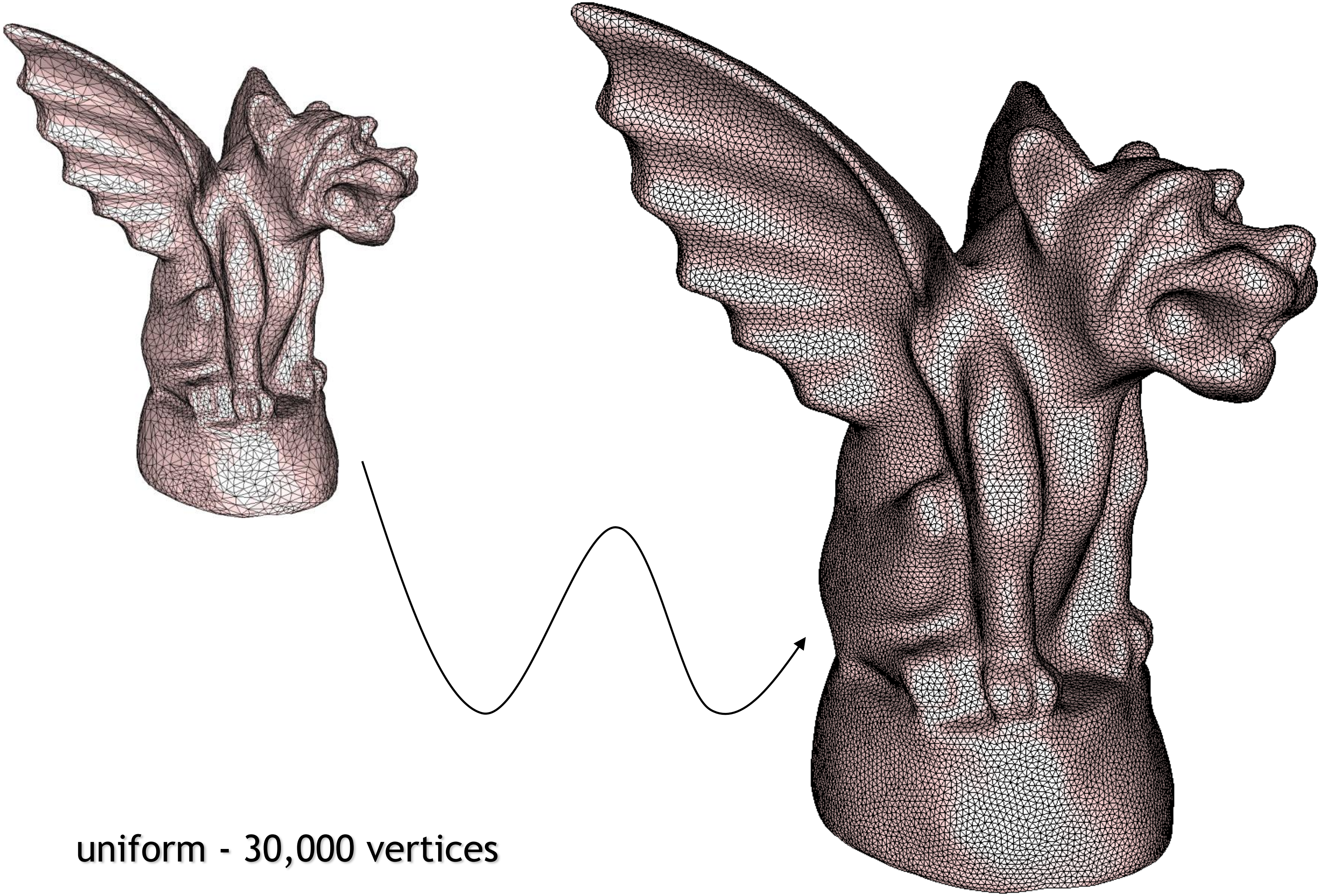




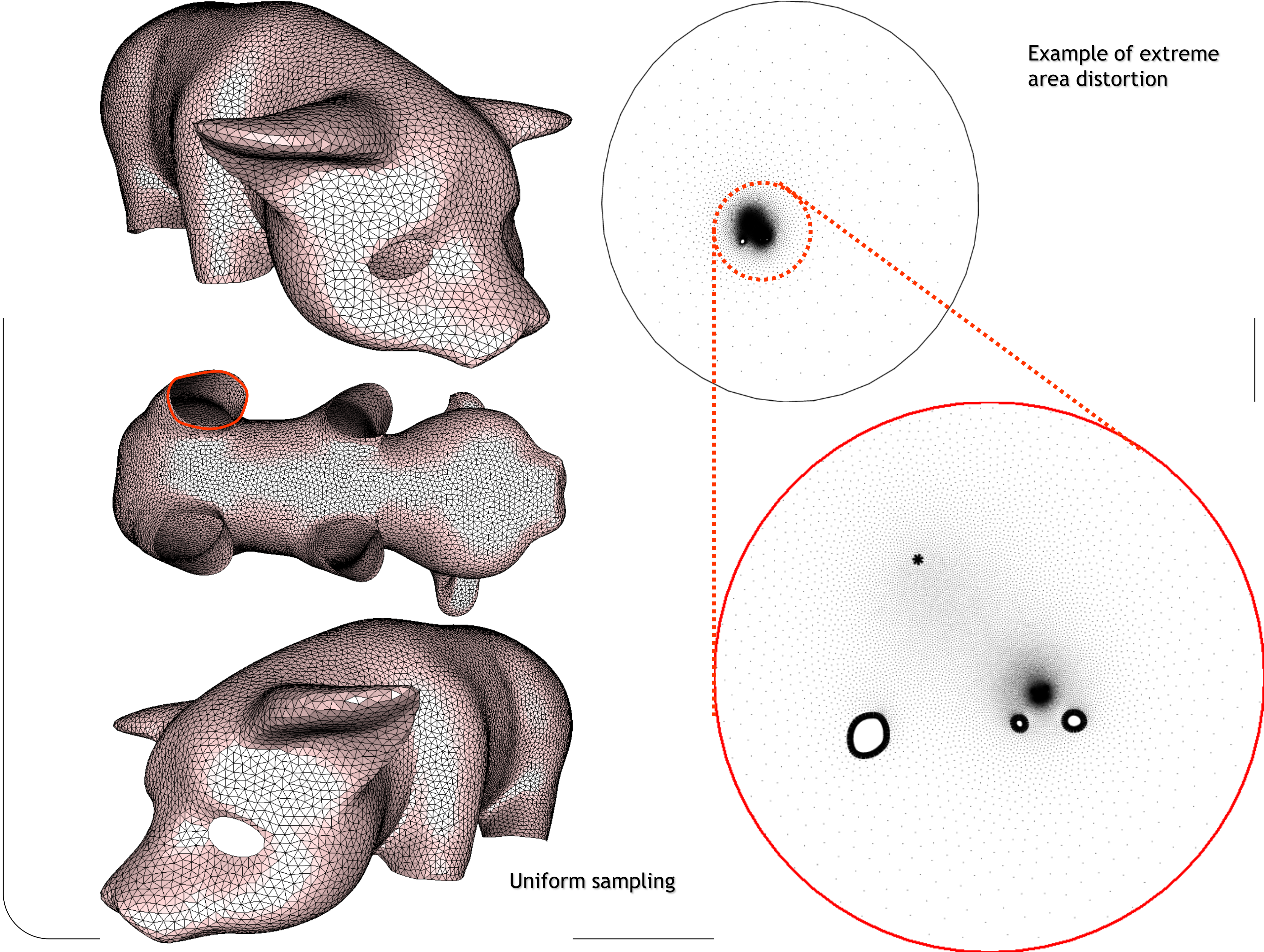


Weighted centroidal
Voronoi diagram

Gargoyle



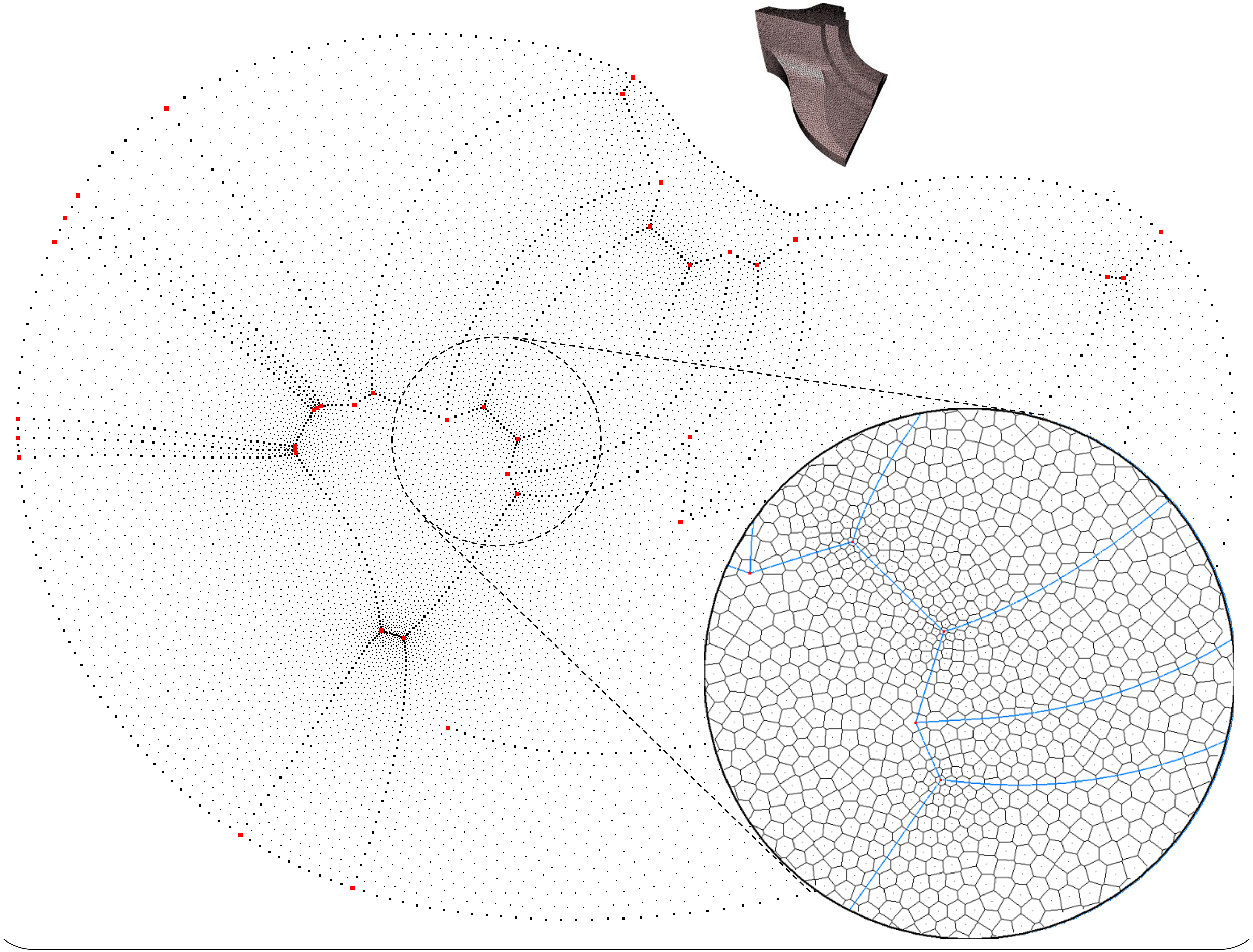
uniform - 30,000 vertices

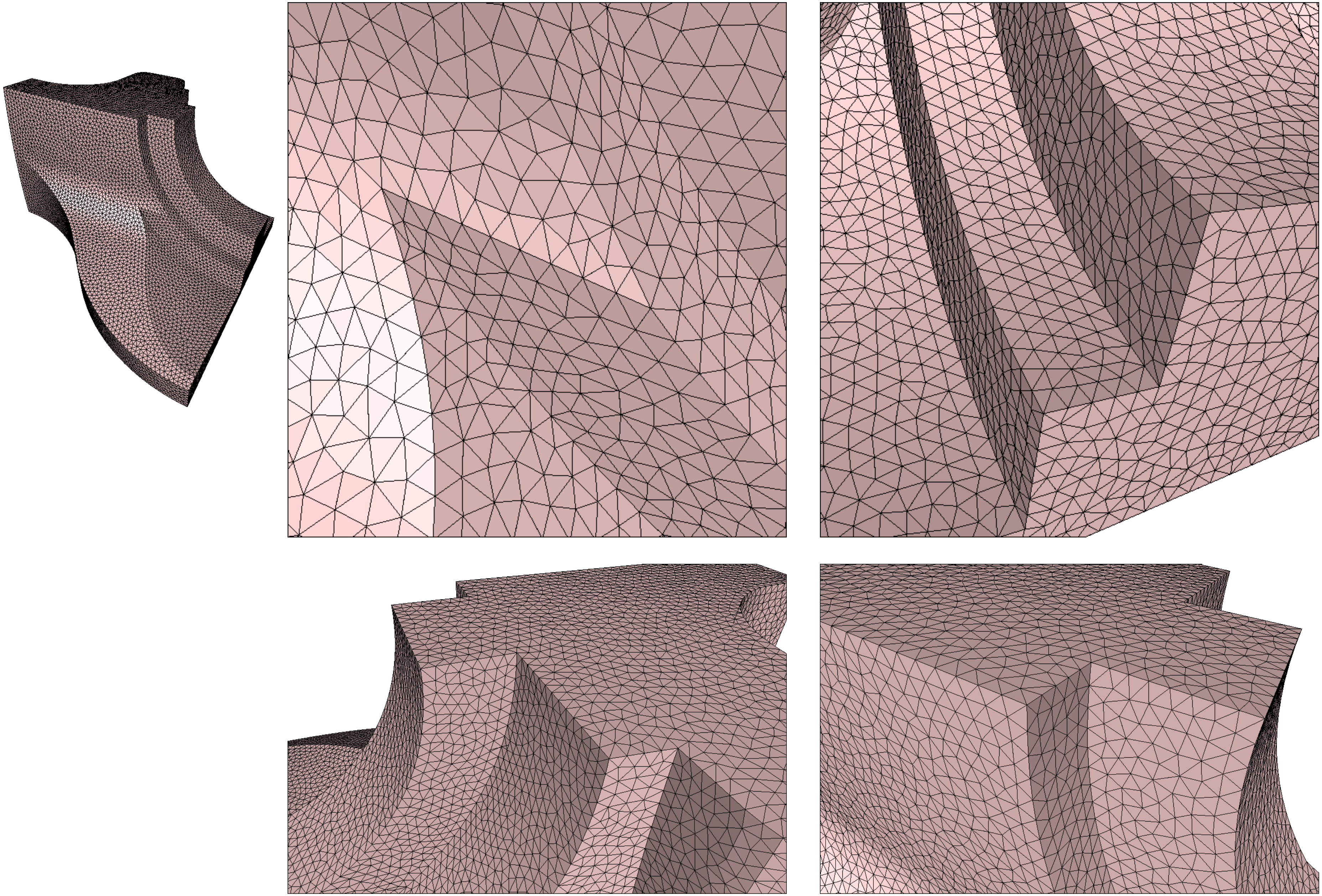

Example of extreme
area distortion
Uniform sampling

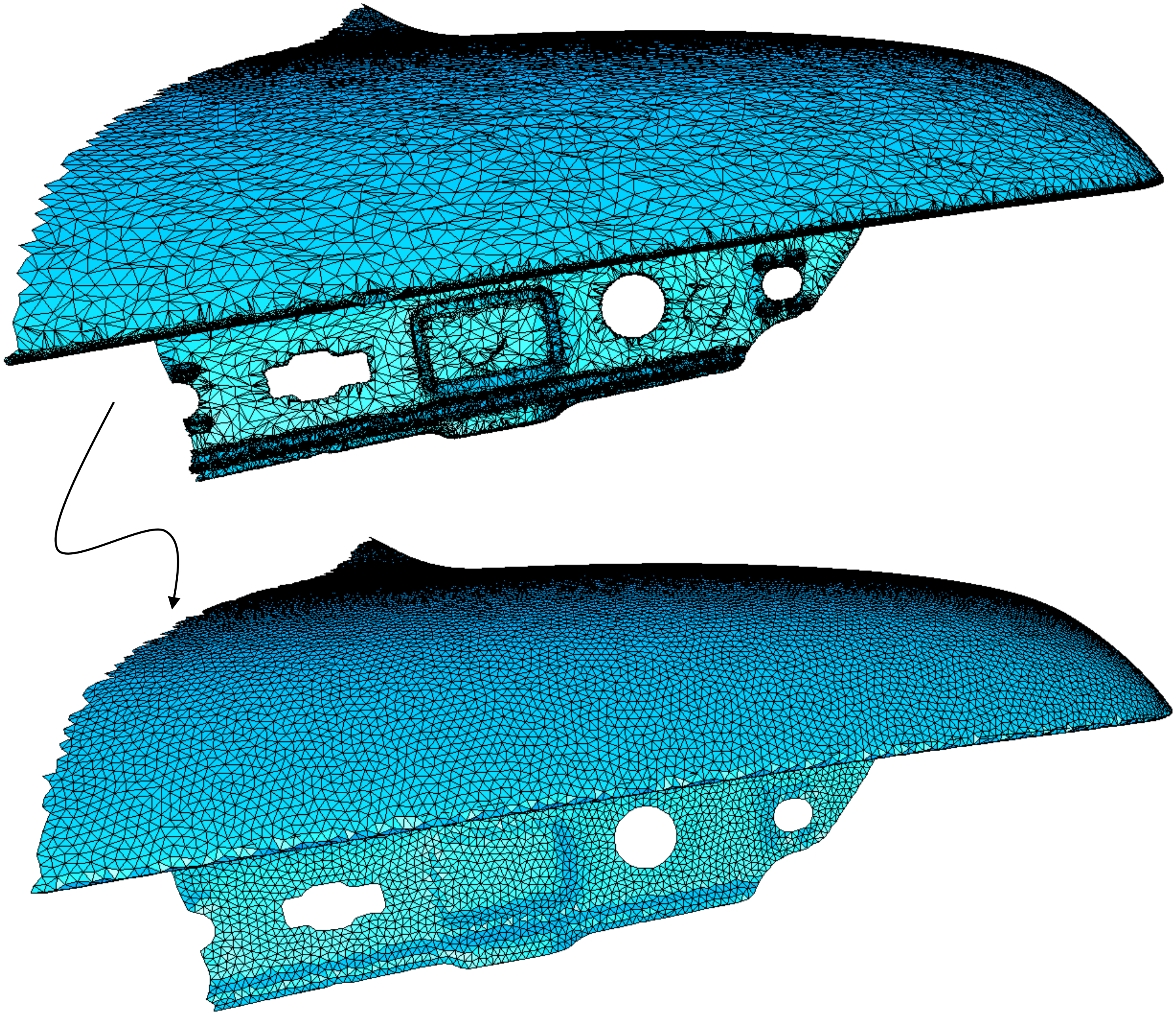
CAD models

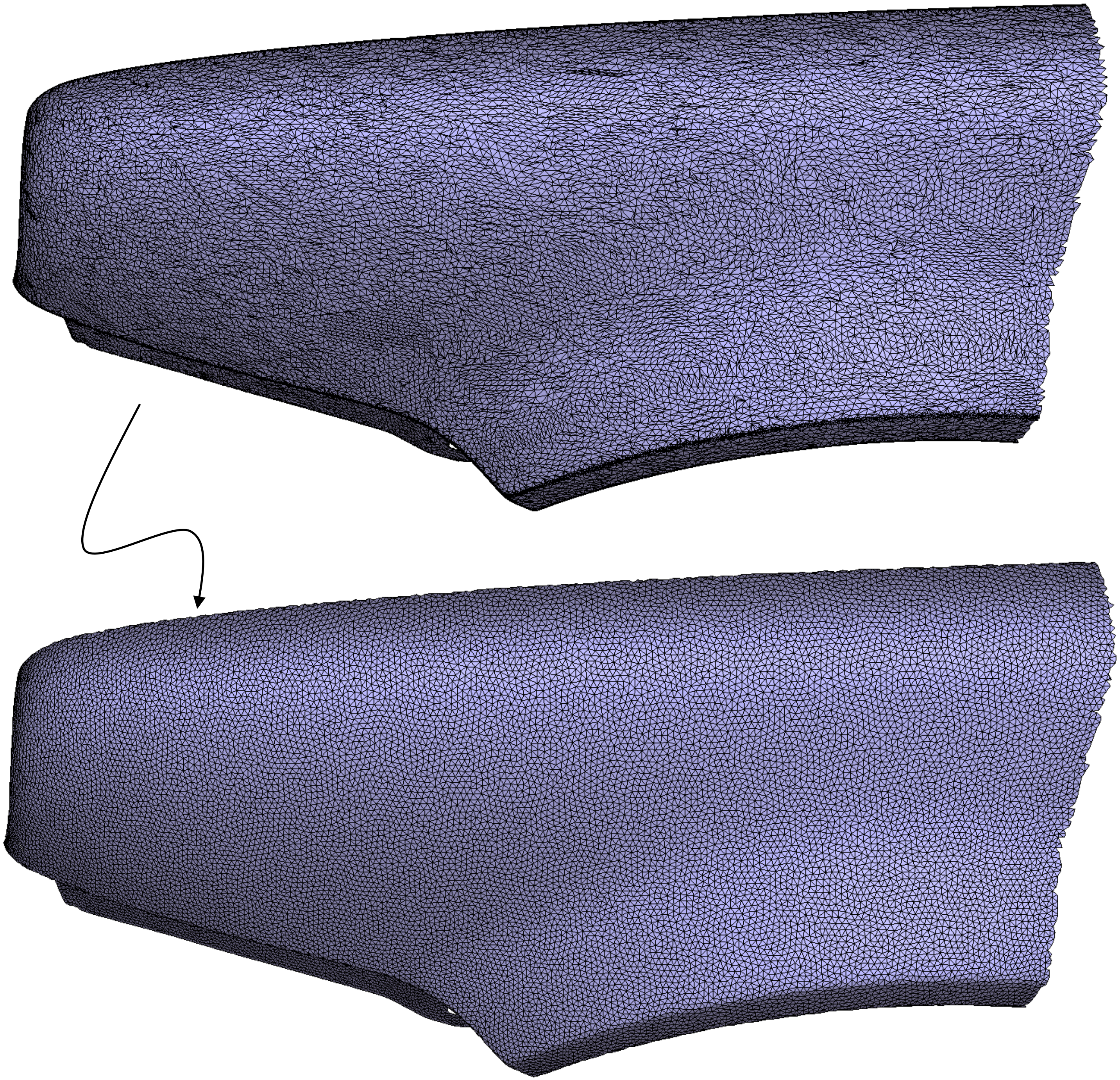
Feature backbones:

- 1D error diffusion
- arc-length parameterization of backbones
- 1D WCVD



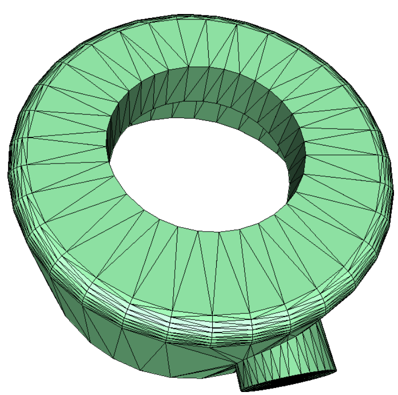


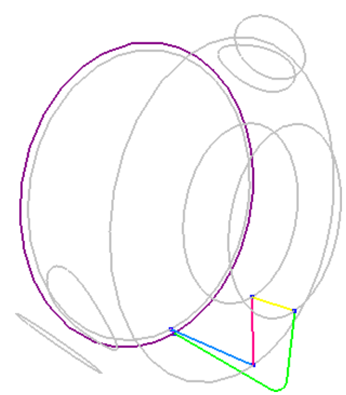


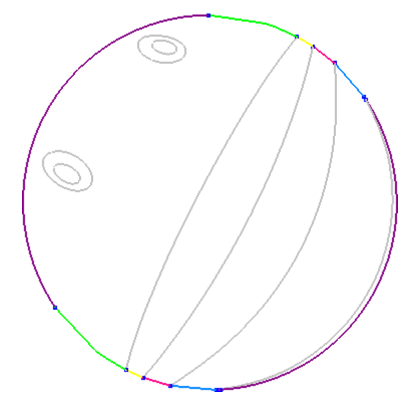


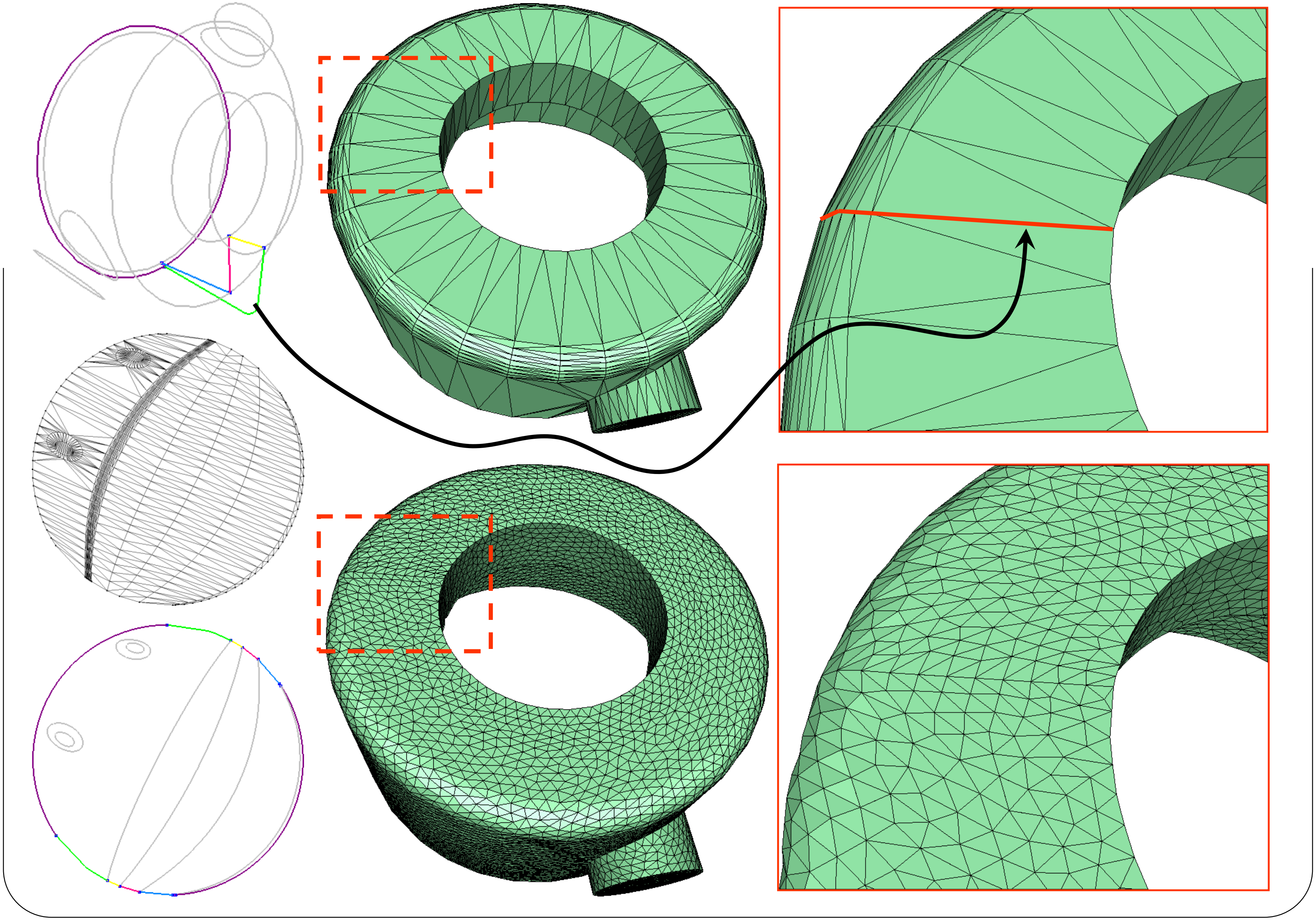
Genus>0 model

- cutting -> cut graph
- add cut graph to feature skeleton
  -> twin backbones associated pairwise
- synchronize sampling along twin backbones to guarantee stitching

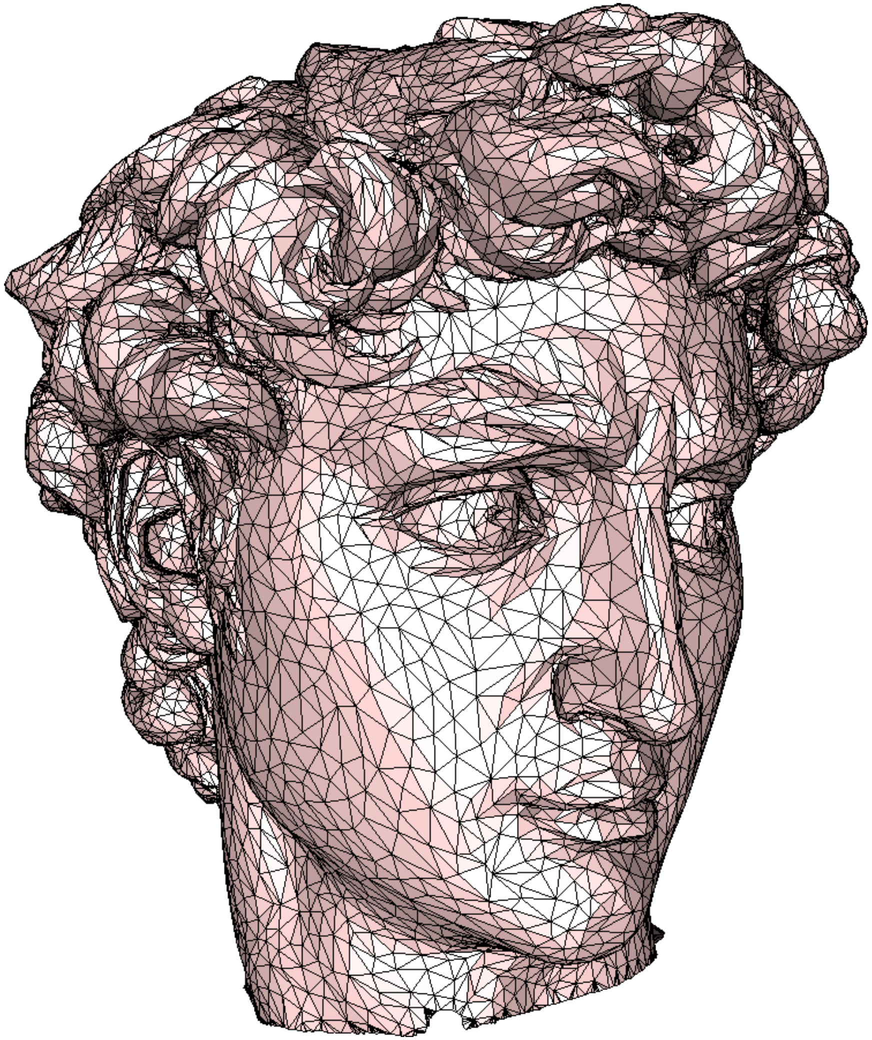

genus 1

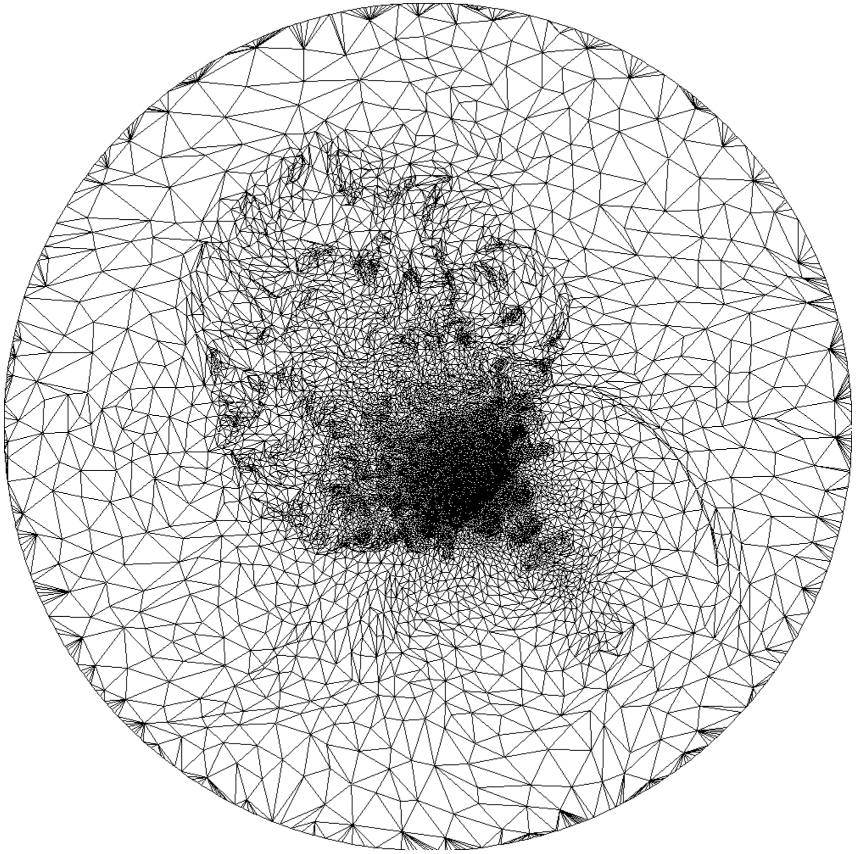

cut graph

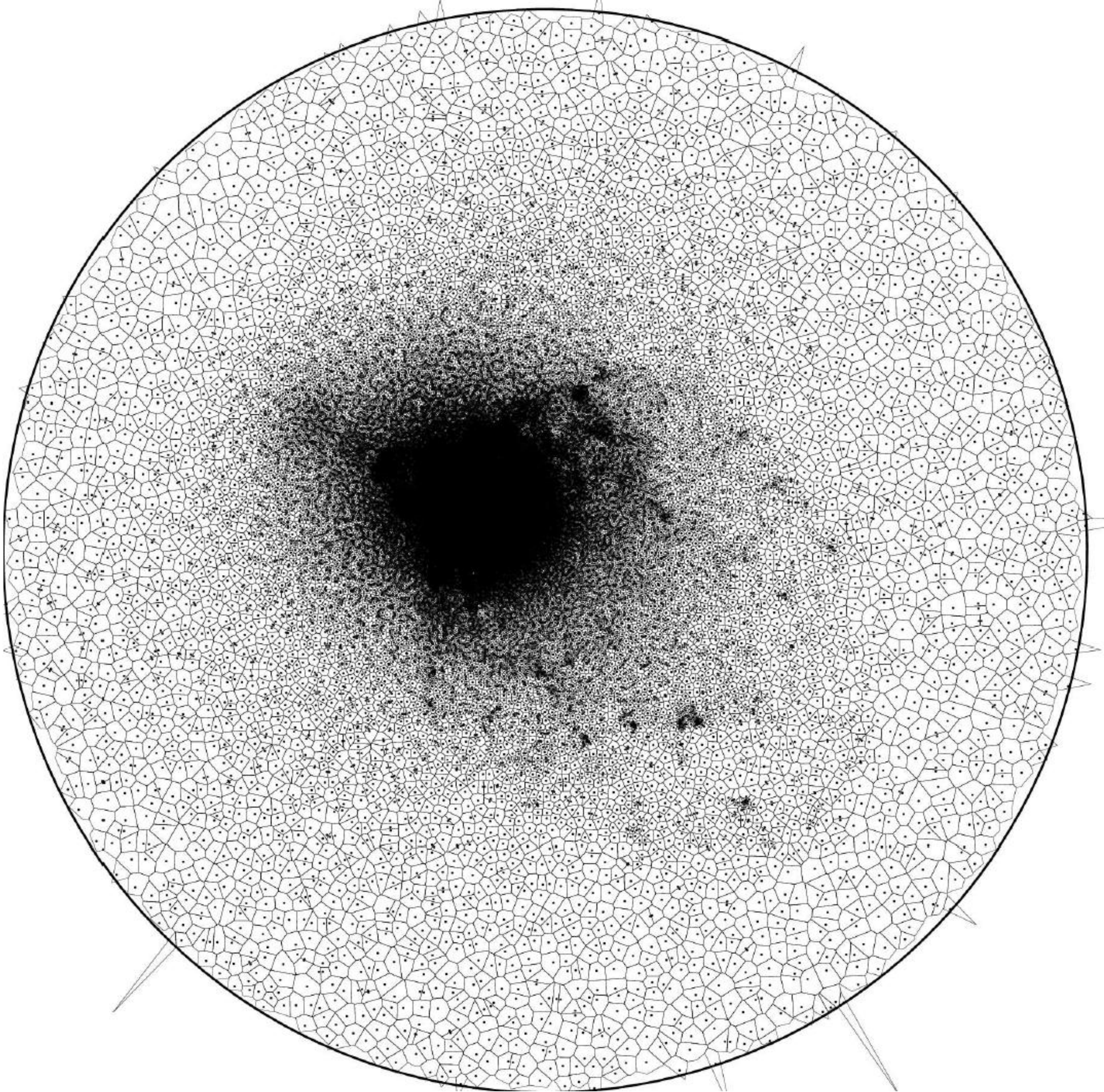

parameter space



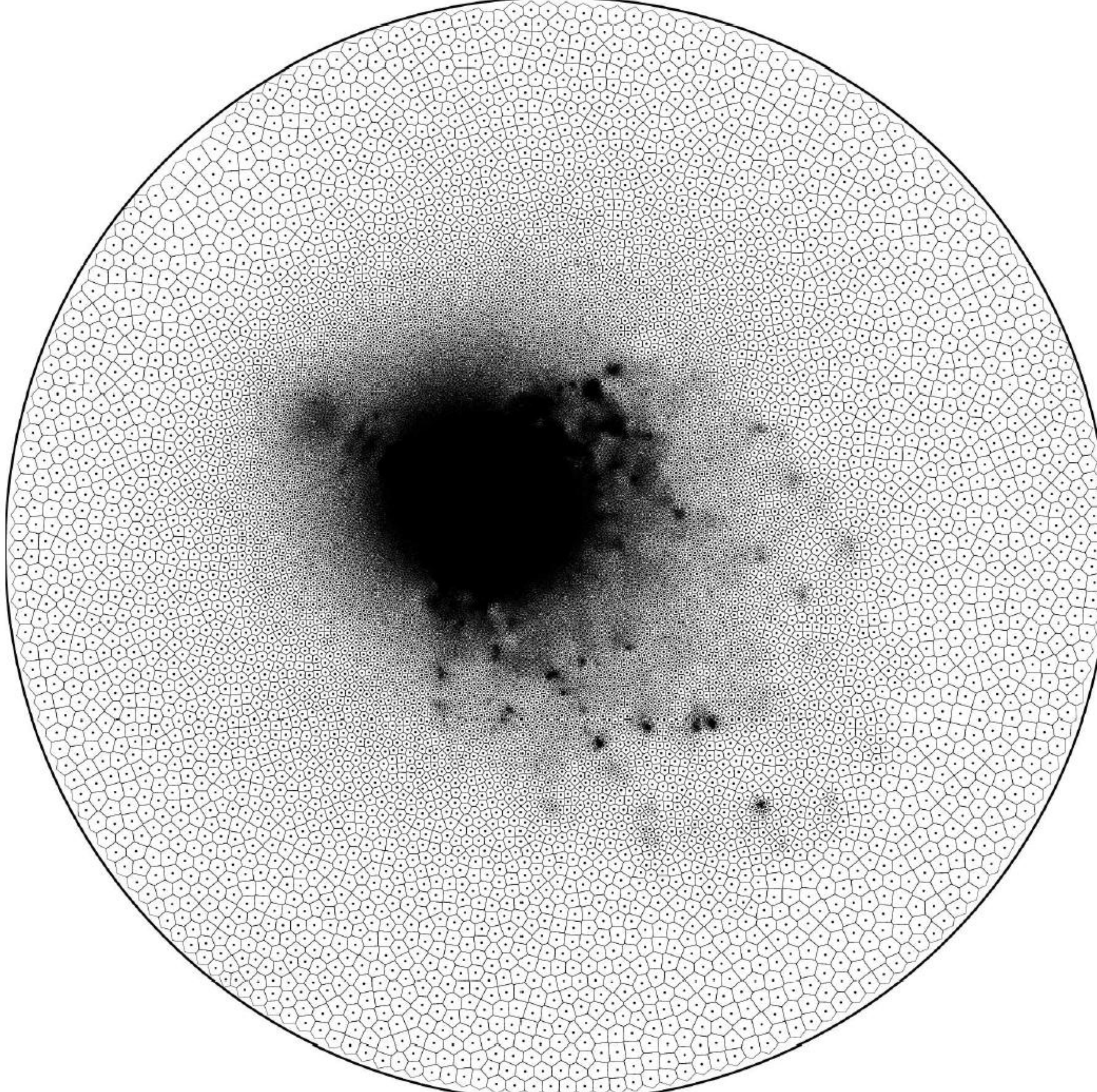
David model [Digital Michelangelo]





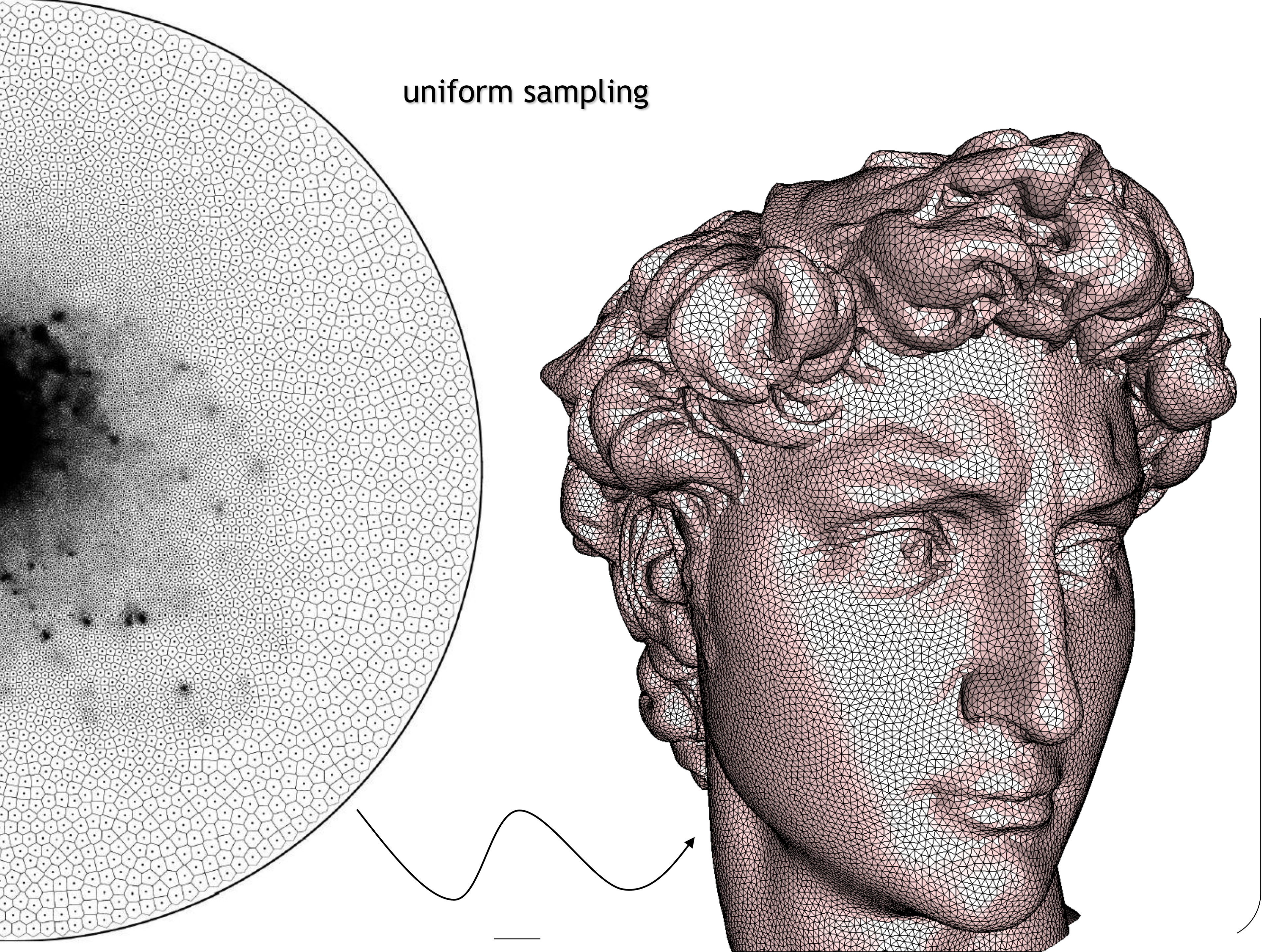


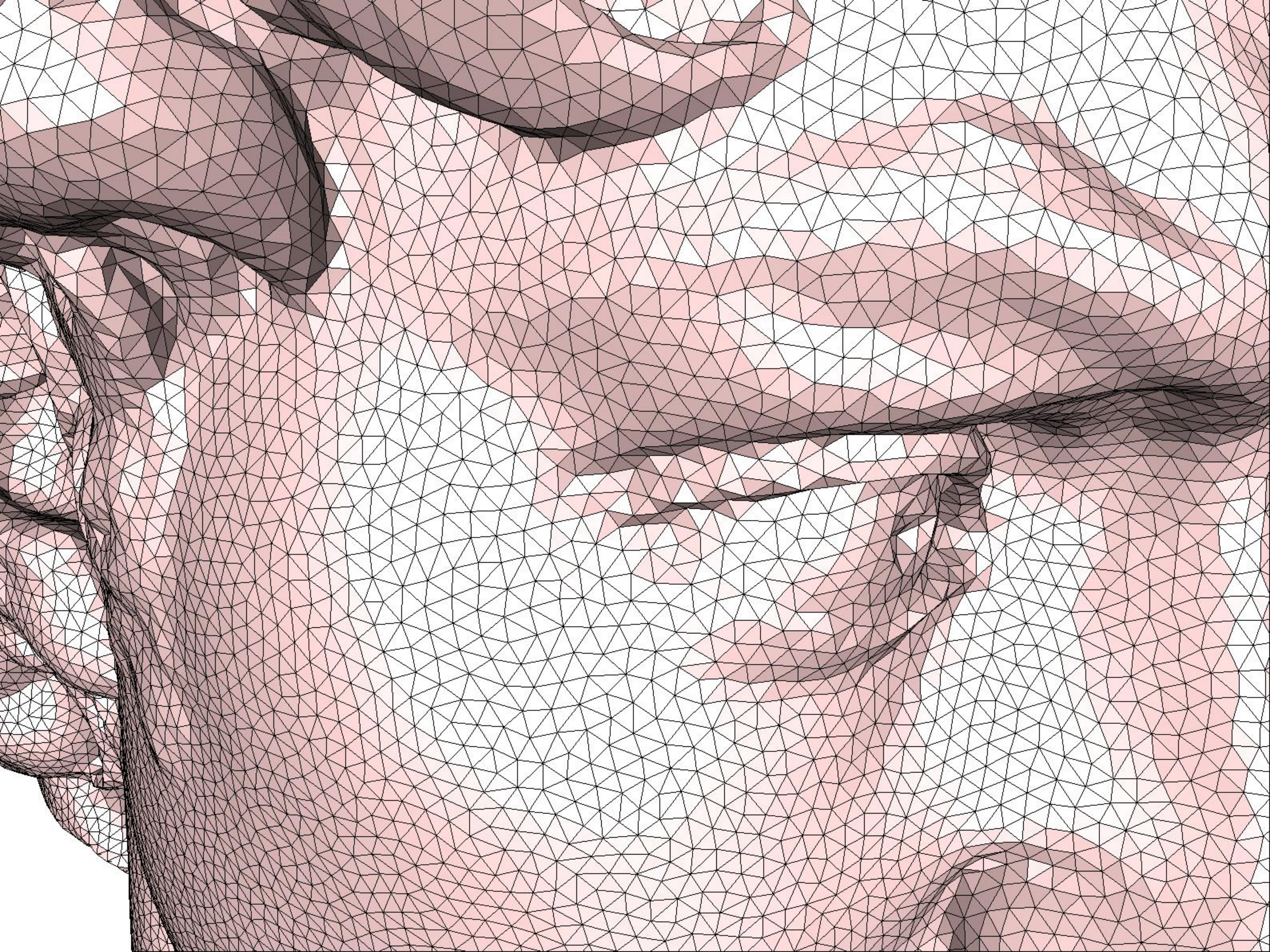
Sample repartition
by error diffusion
(50,000 vertices)

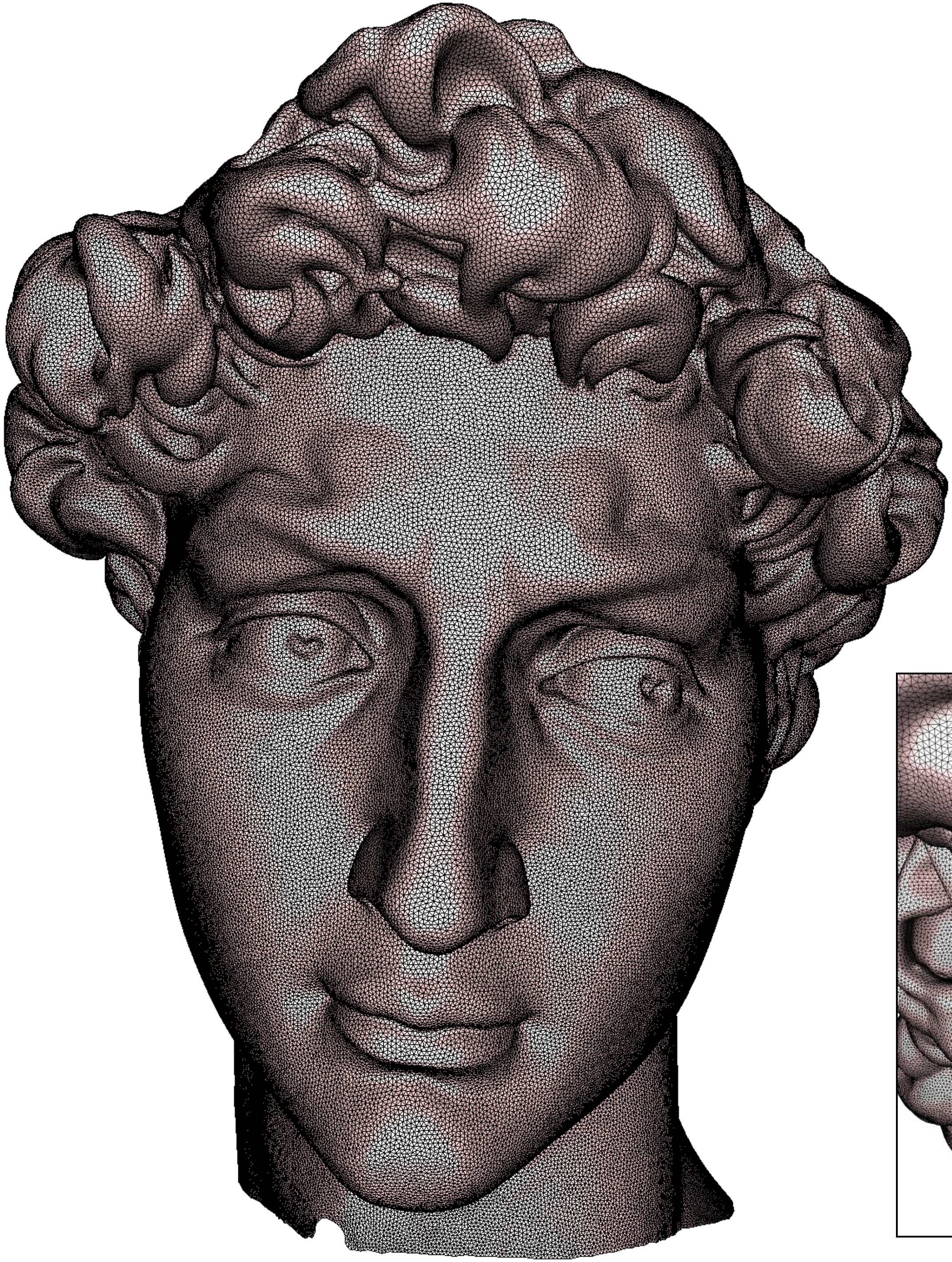


Sample placement by WCVD

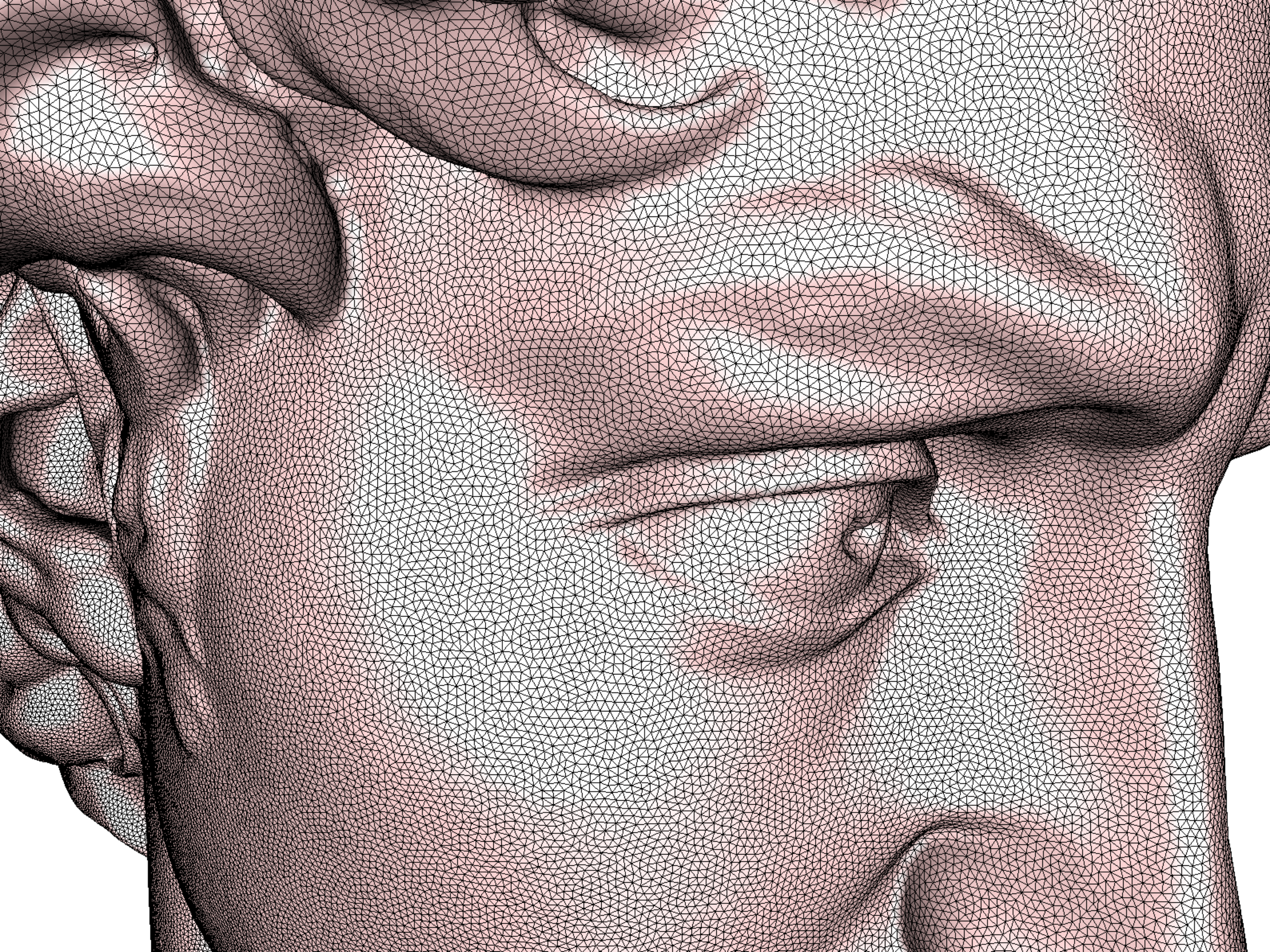
uniform sampling



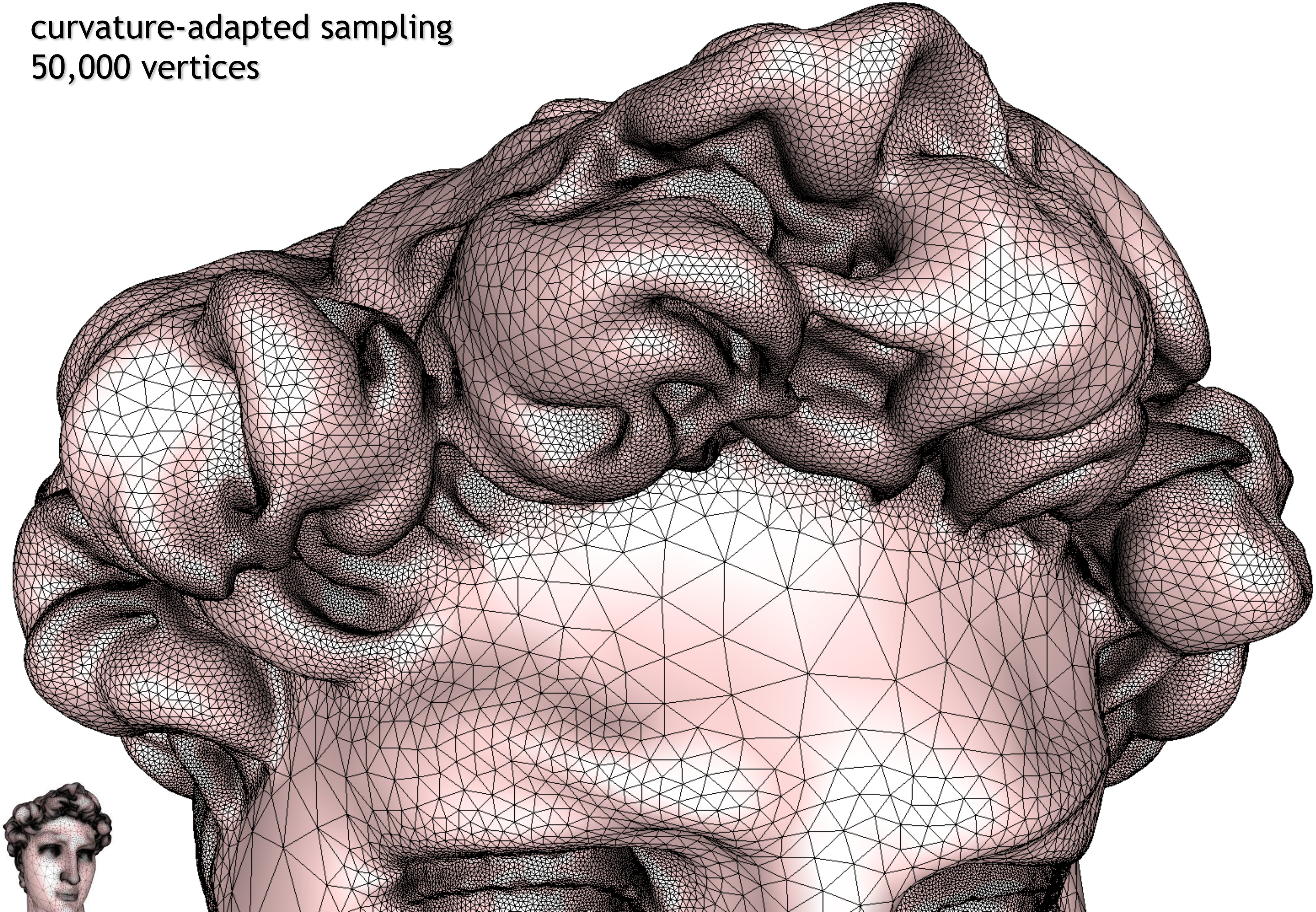


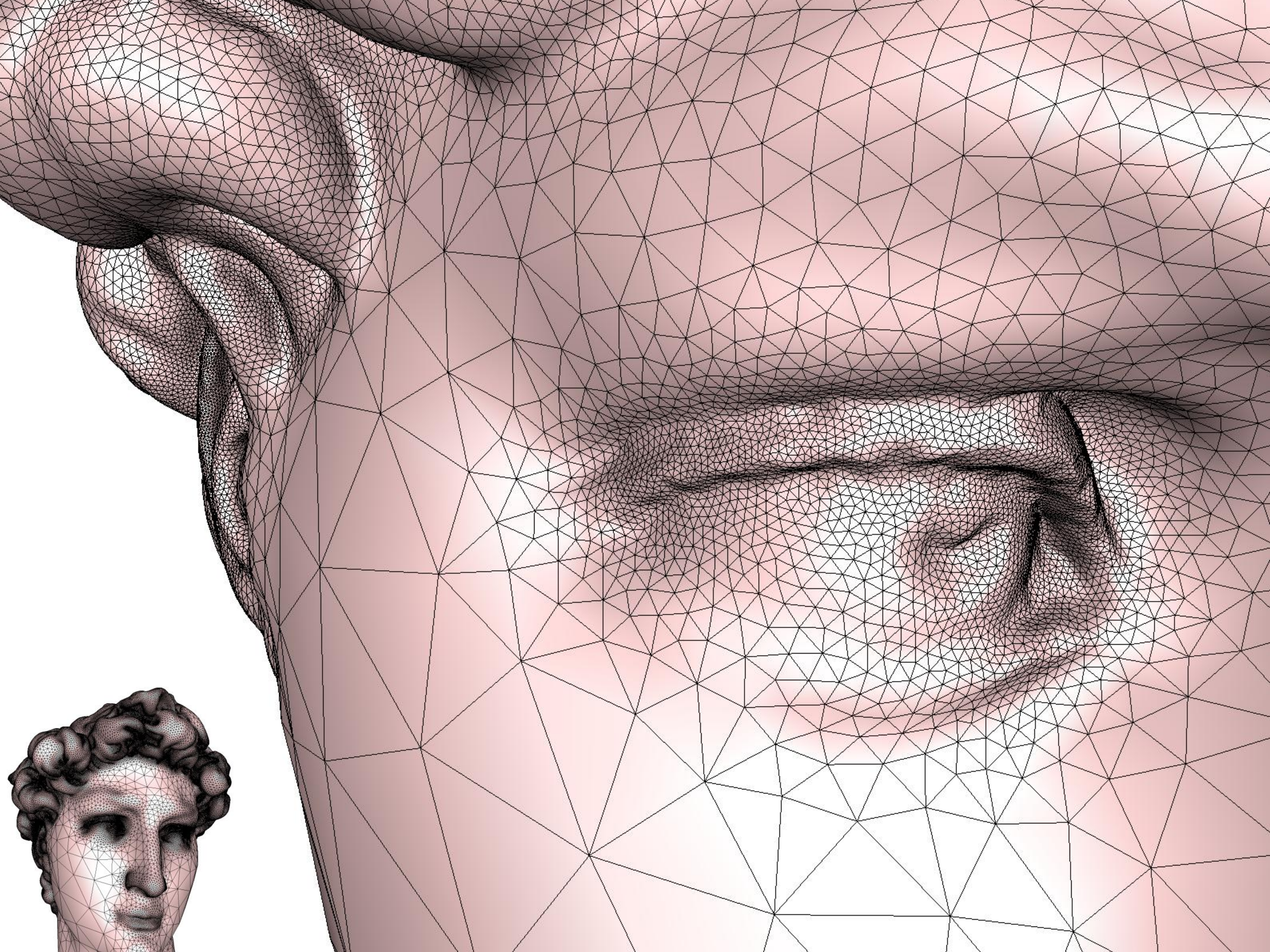


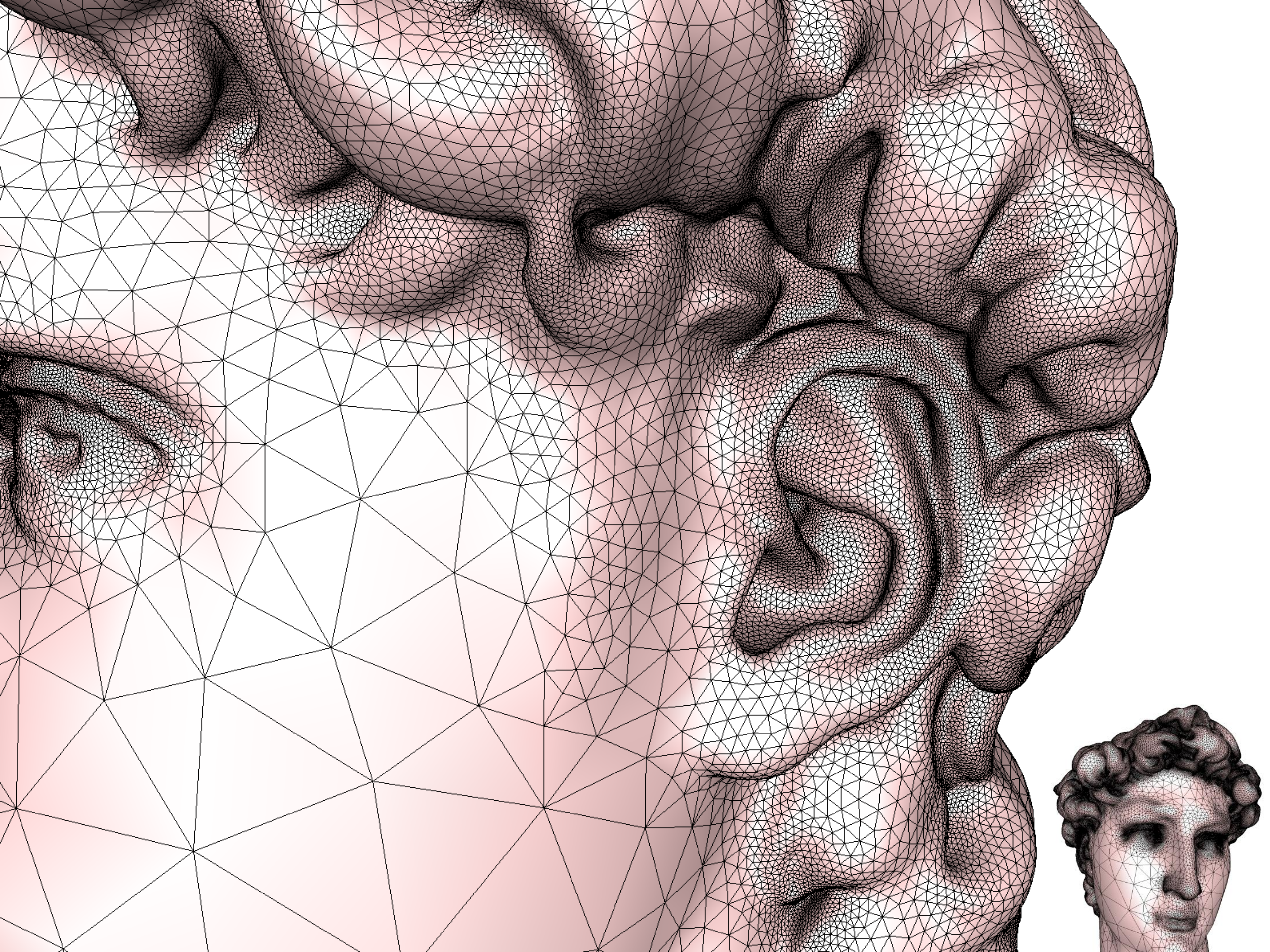
uniform sampling
300,000 vertices



curvature-adapted sampling
50,000 vertices







Motivation
Previous work
Contributions
Algorithm
Results
Limitations
Conclusions
Future Work

Limitations

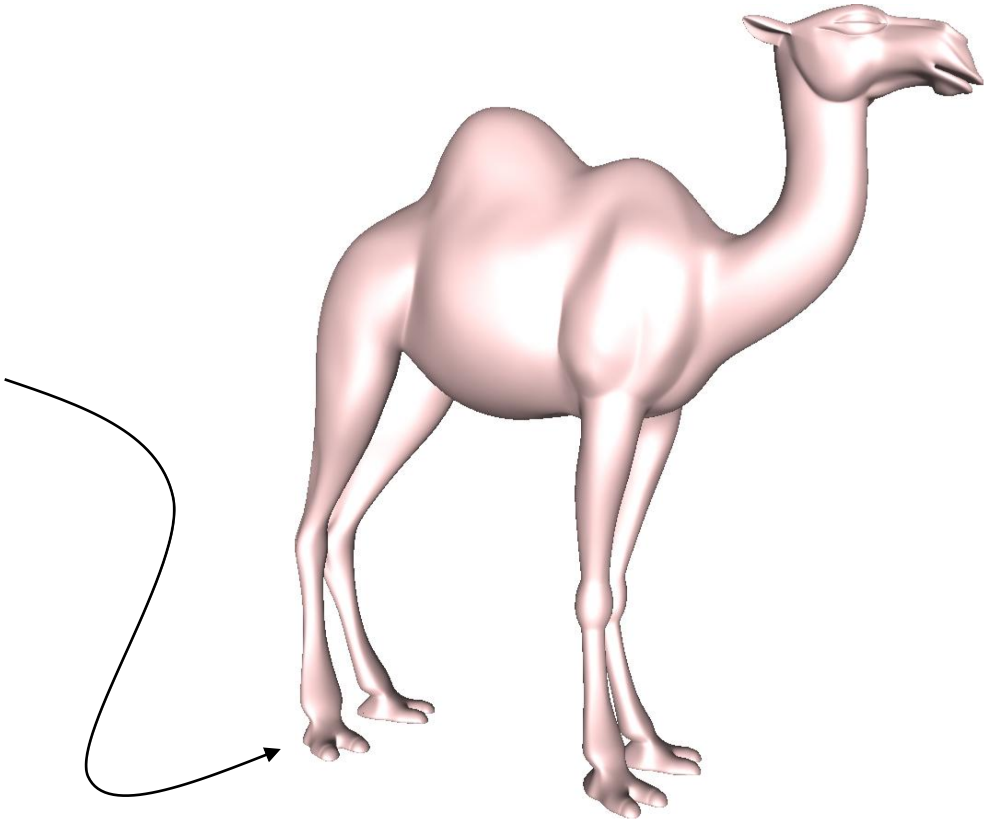
- Parameterization
  - quality of sampling is *very dependent* on the quality of the parameterization
  - still some numerical issues for huge models

- Complex genus or closed surface
  - requires surface cutting (difficult task)
  - process "curve sampling " along the cut graph
  - makes the implementation trickier (seaming backbones, twin samples to synchronize for stitching, branching vertices, etc.)

Limitations

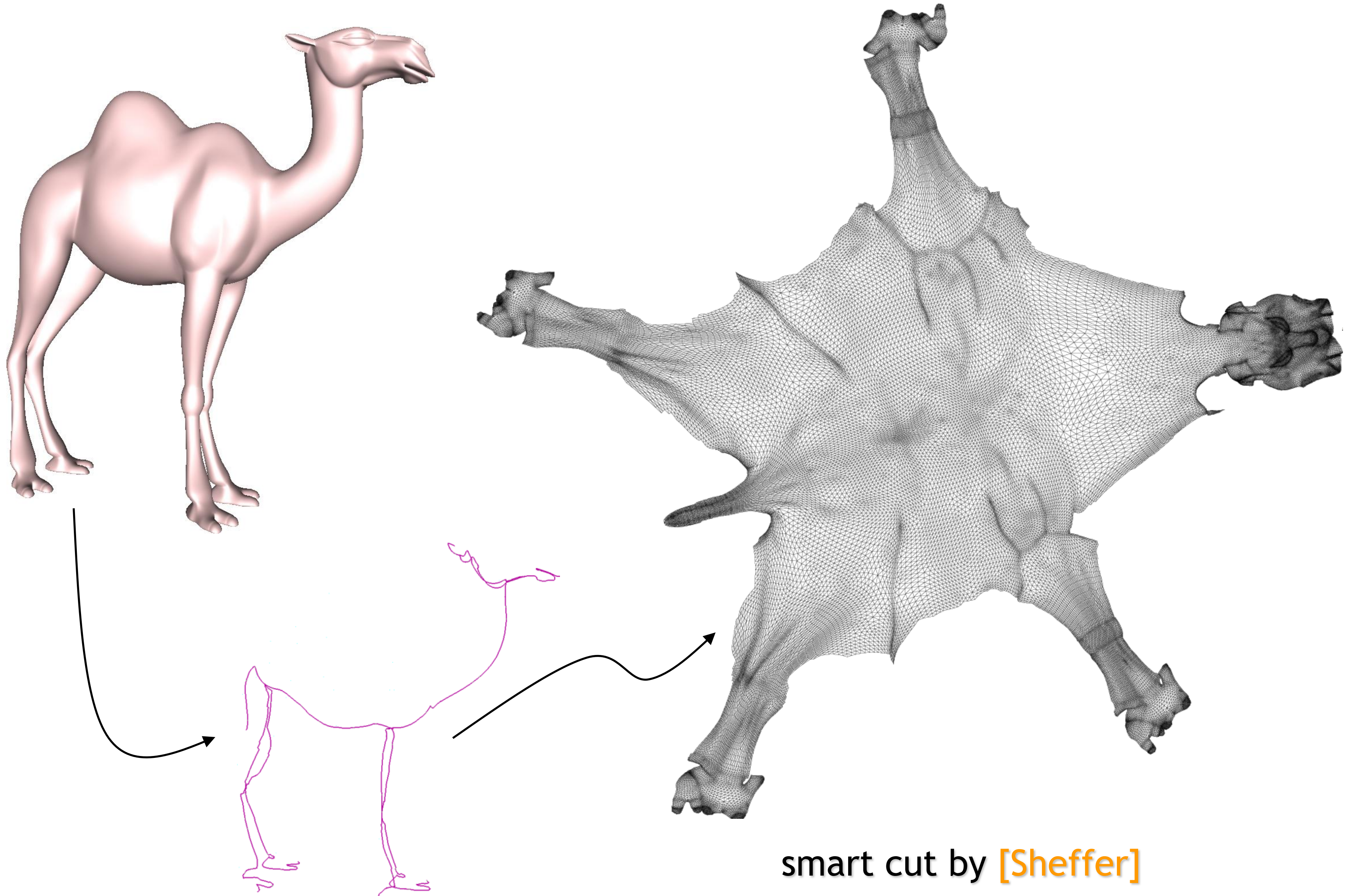
The Camel

- Closed
- Genus 0
- Sock-like shapes



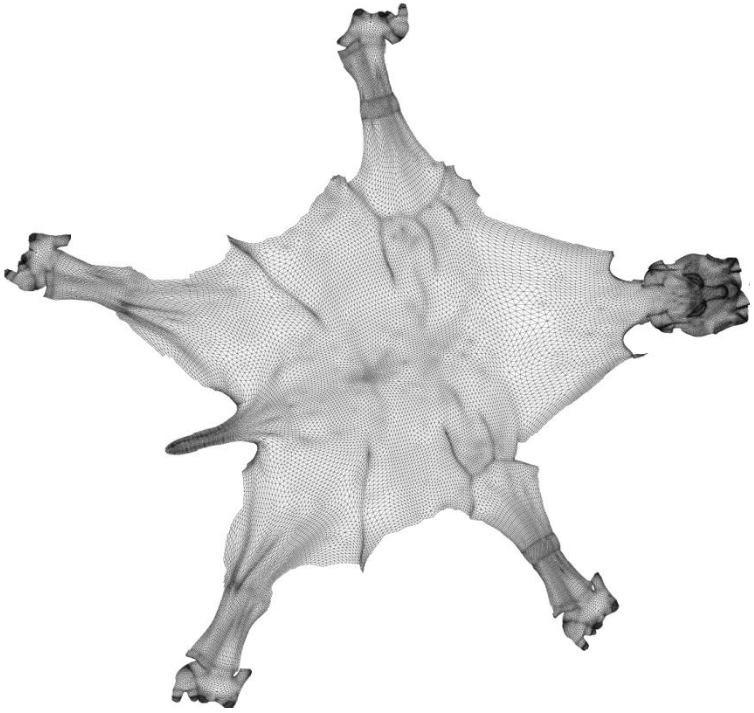
how to cut it? ->

Camel pelting

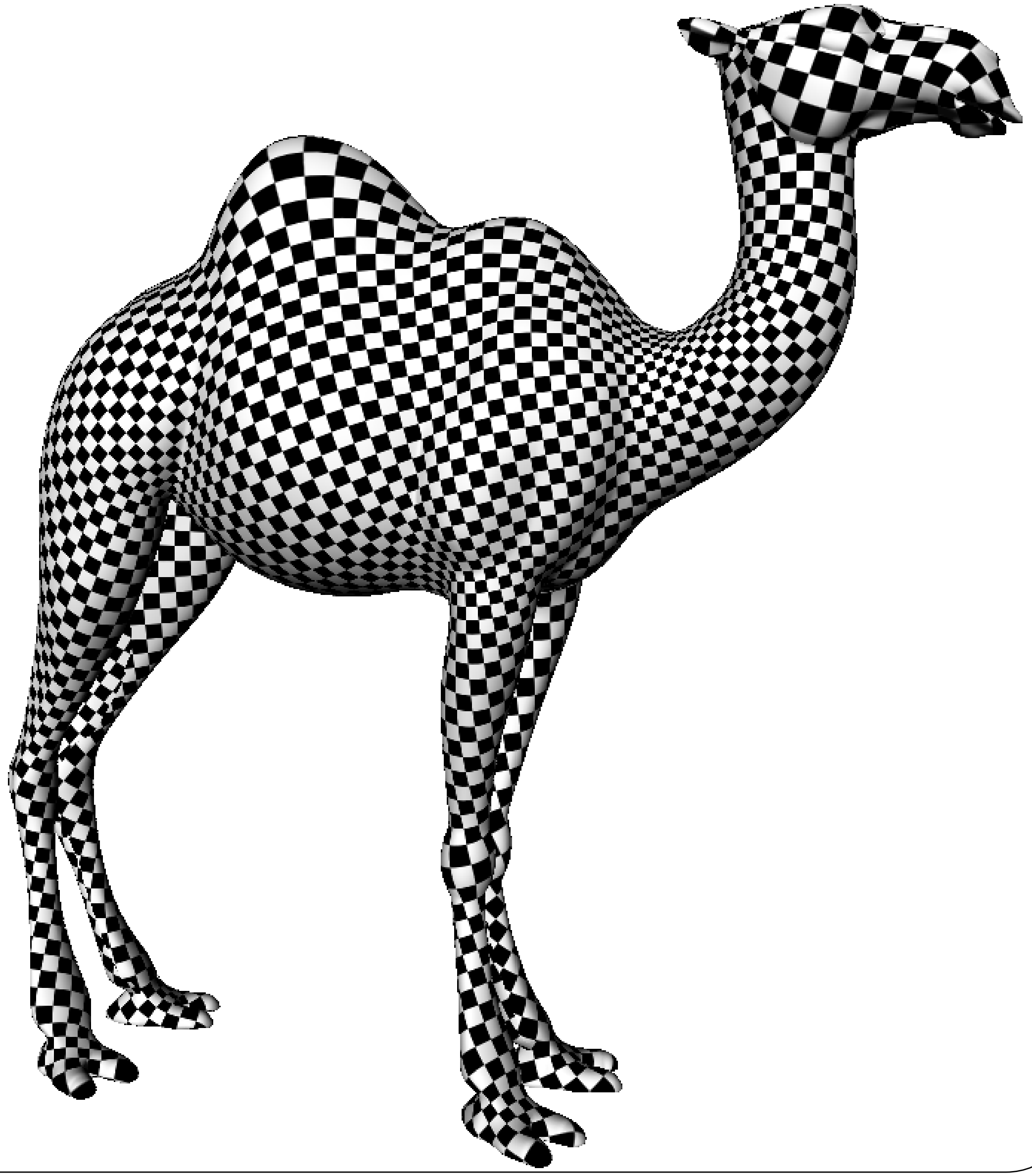


smart cut by [Sheffer]

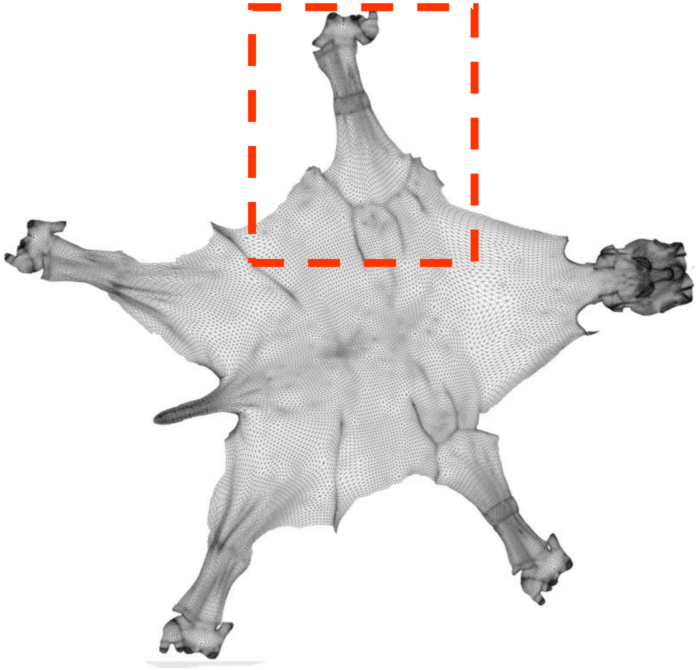
Smart pelting & Parameterization

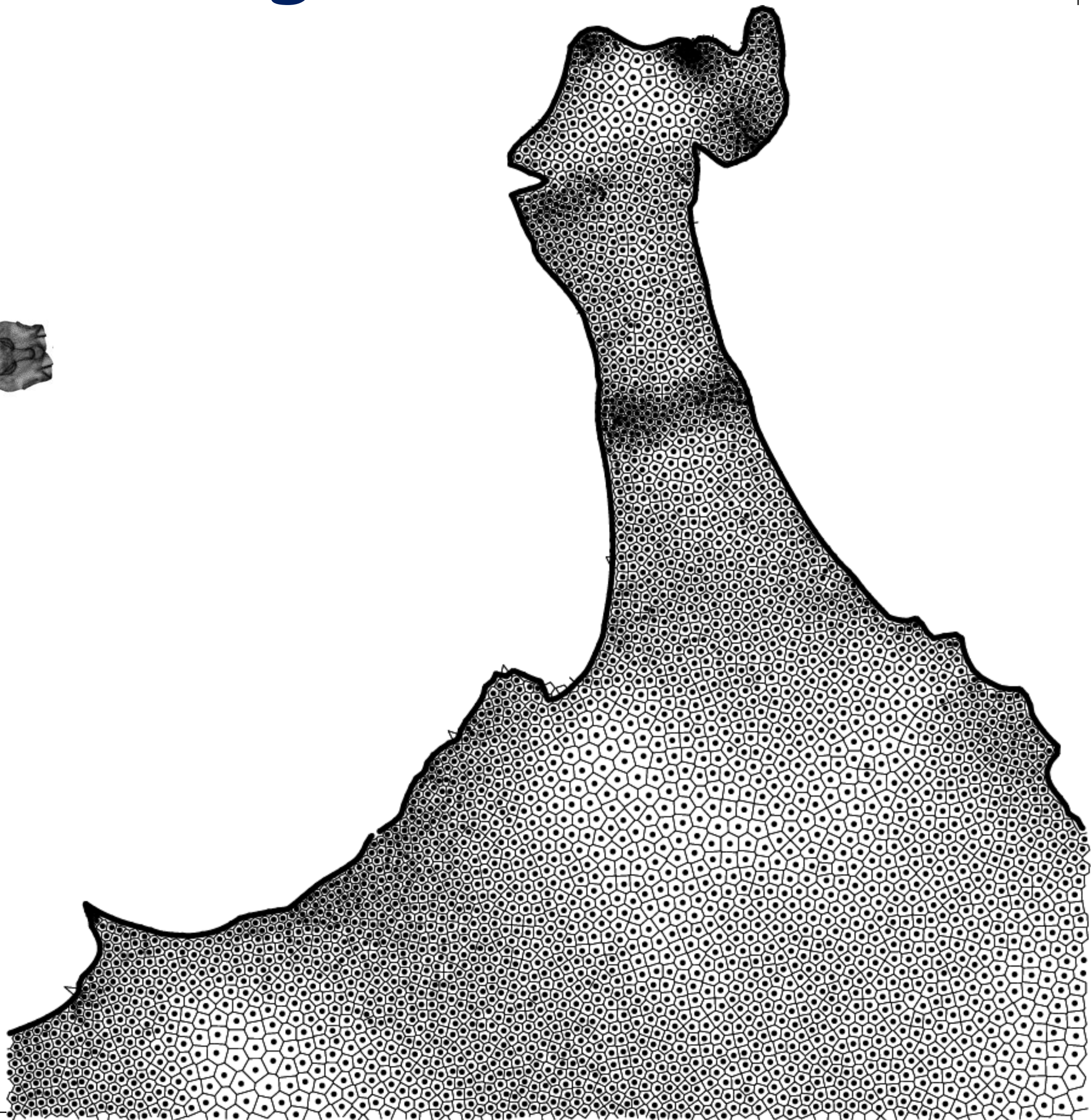


[Sheffer]

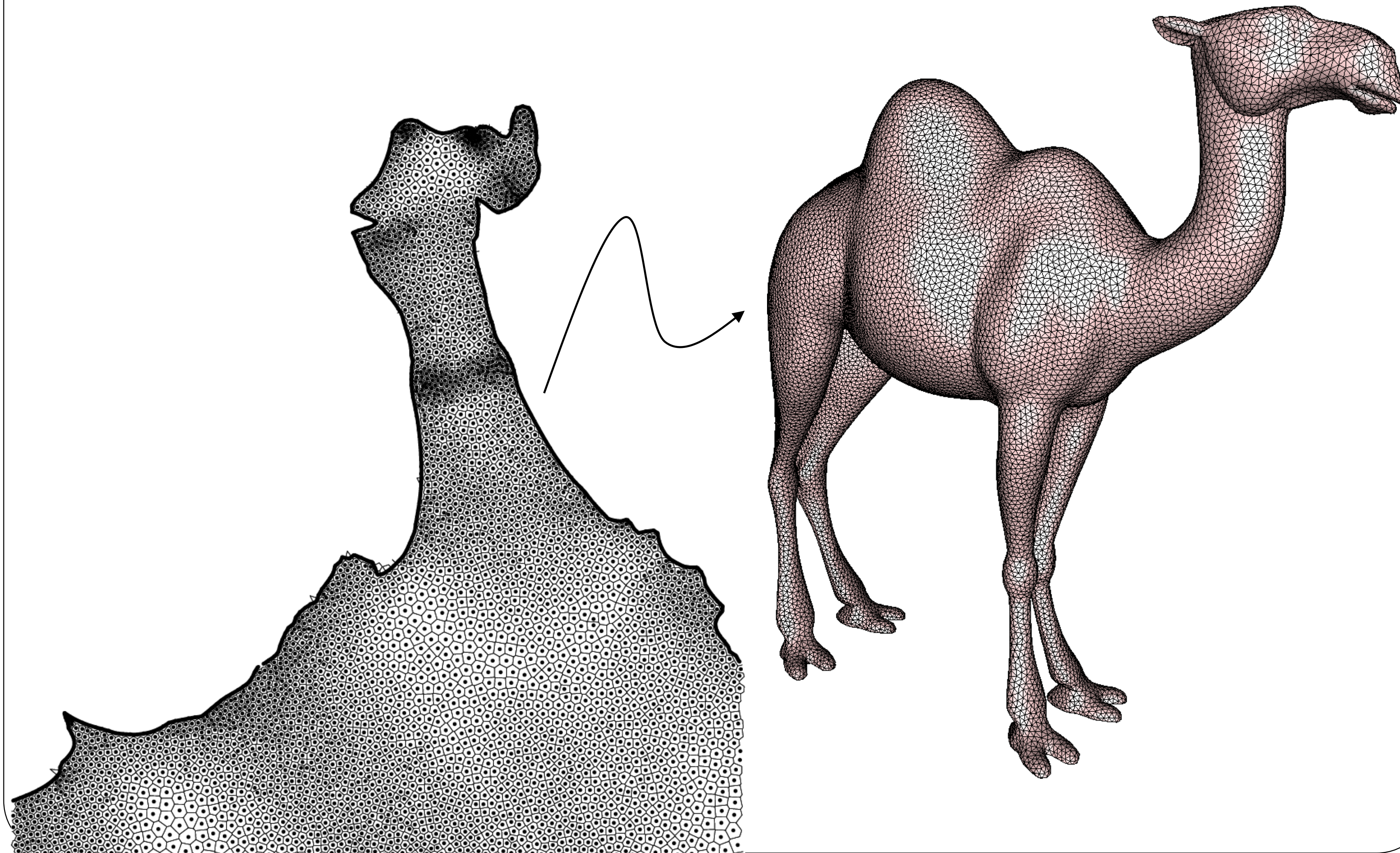


Uniform remeshing

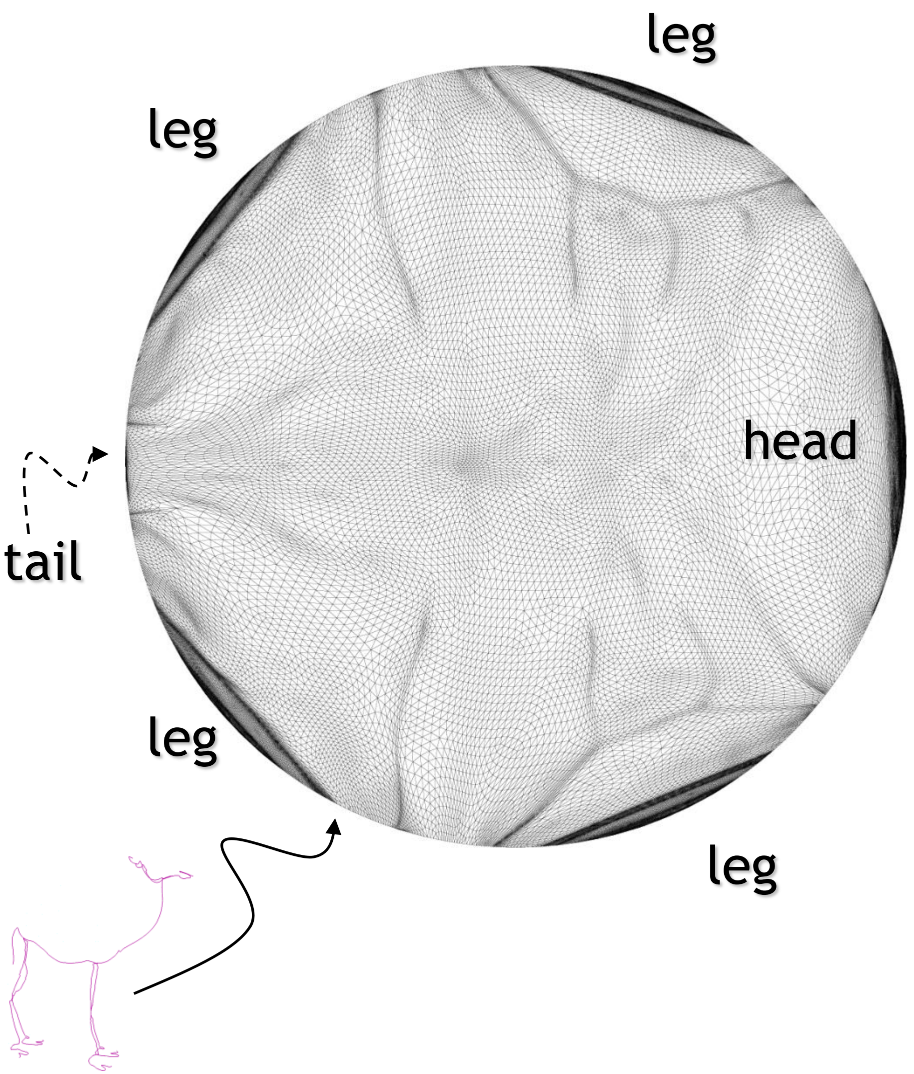


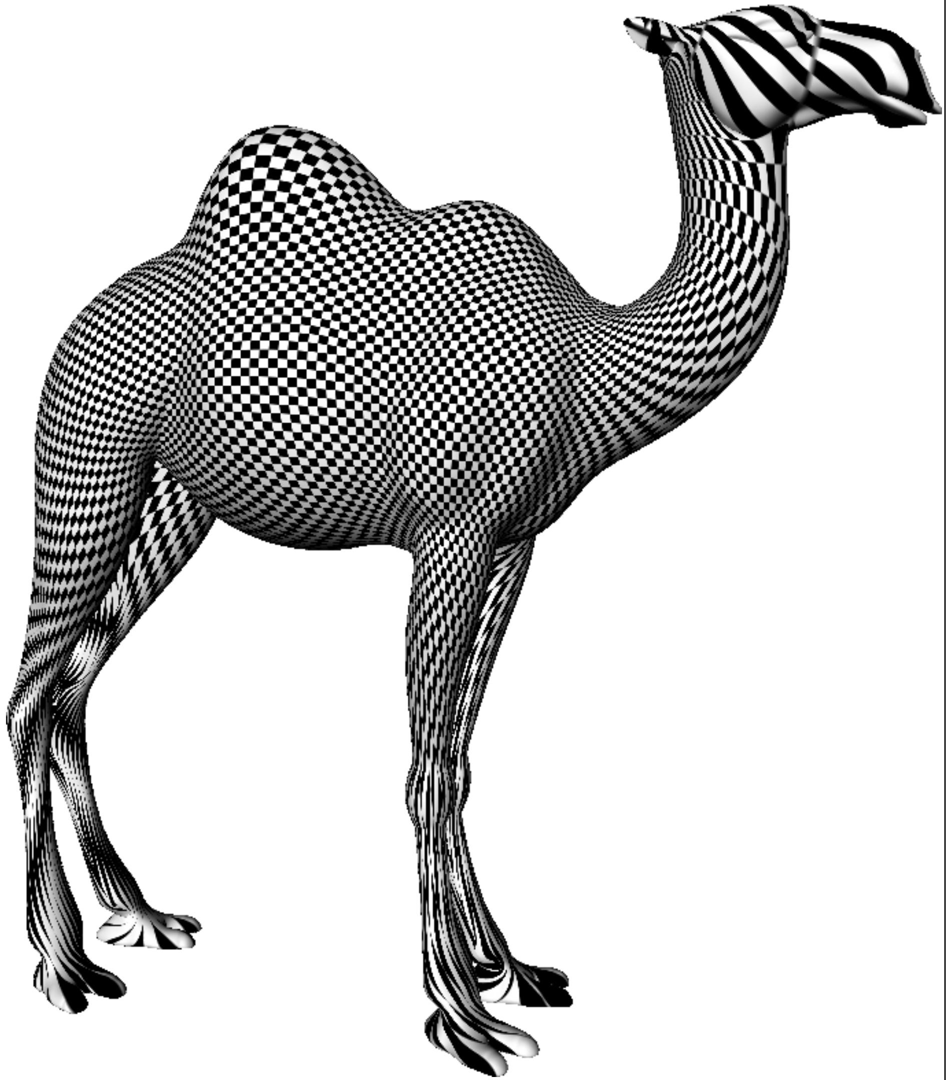


Uniform remeshing

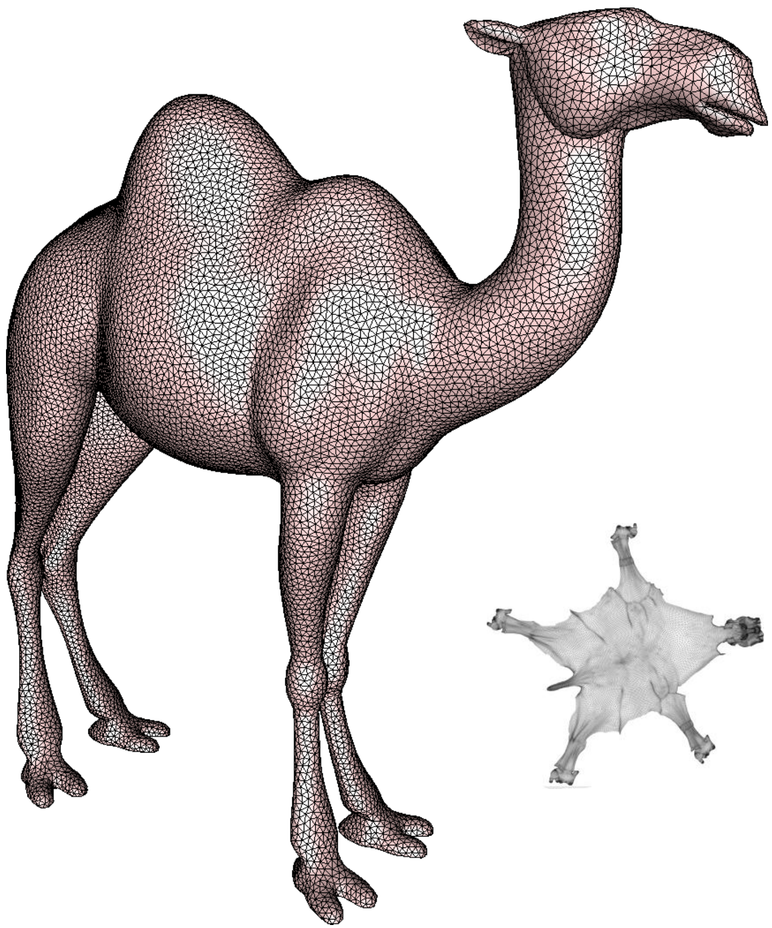


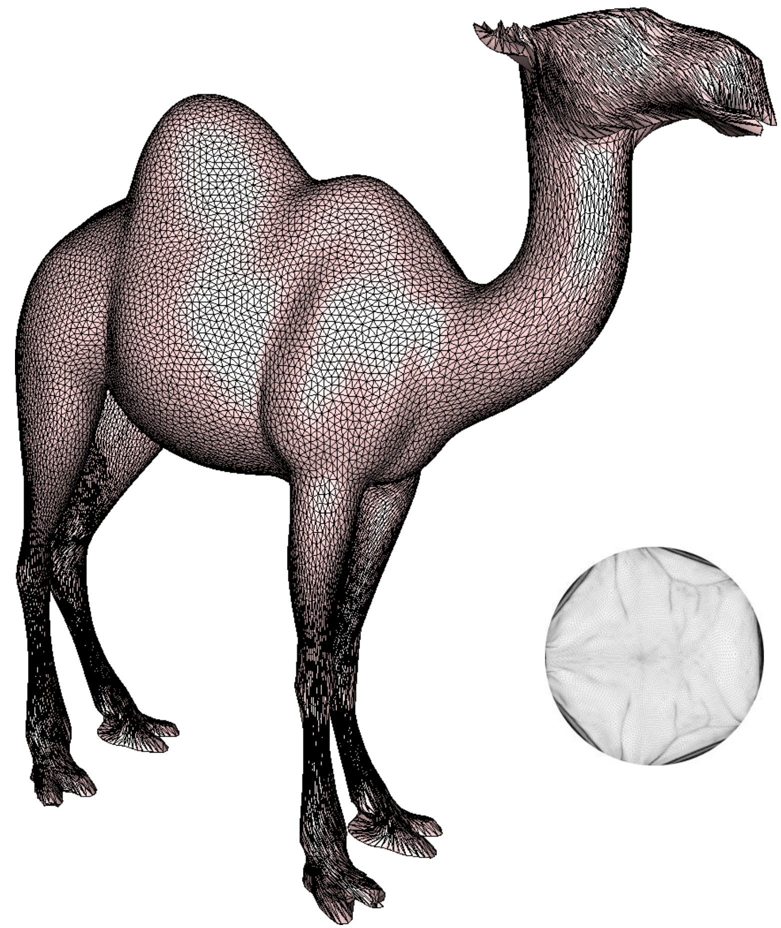
Same cut, fixed boundary ?



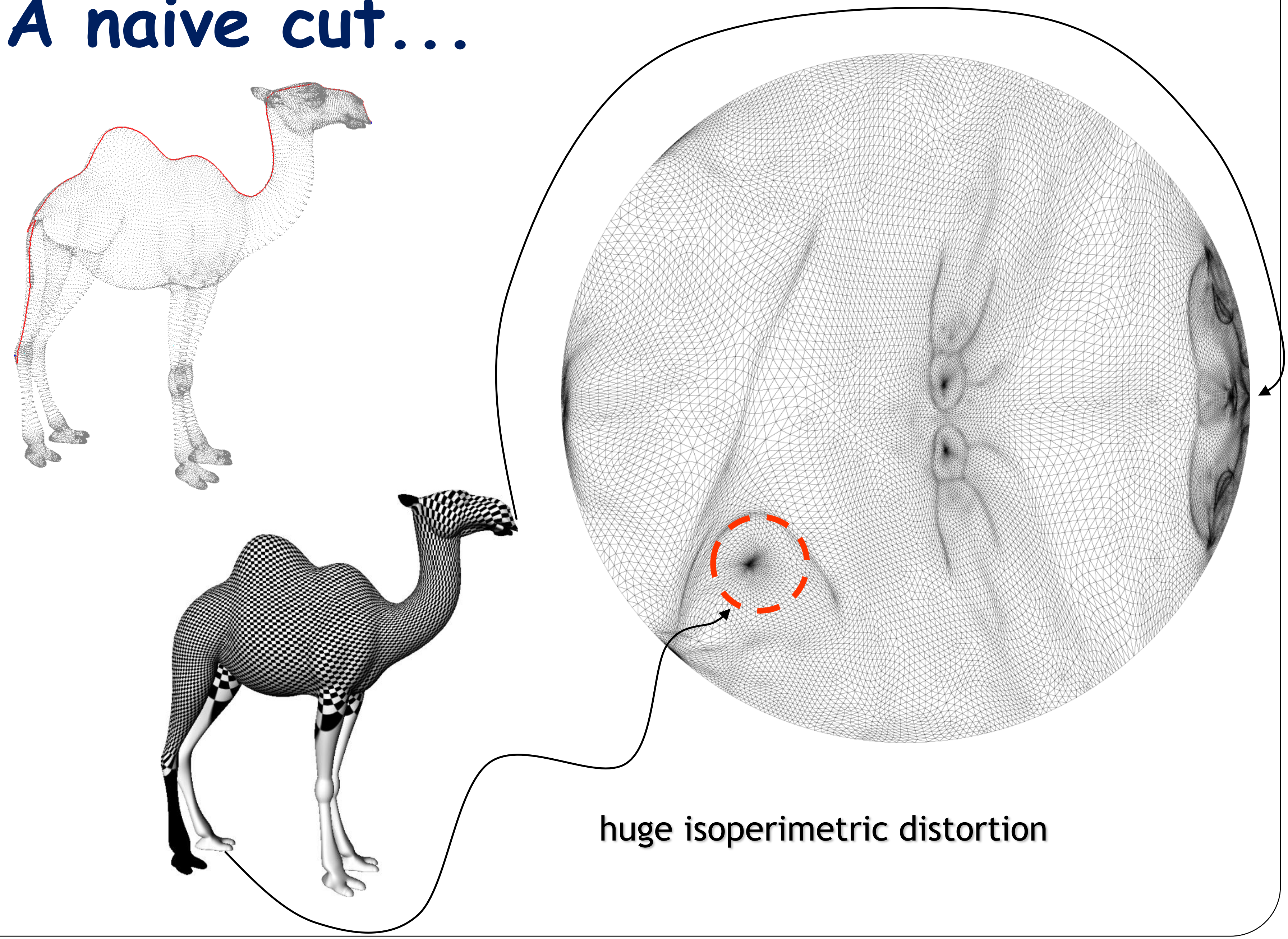


Remeshing with Free vs fixed boundary

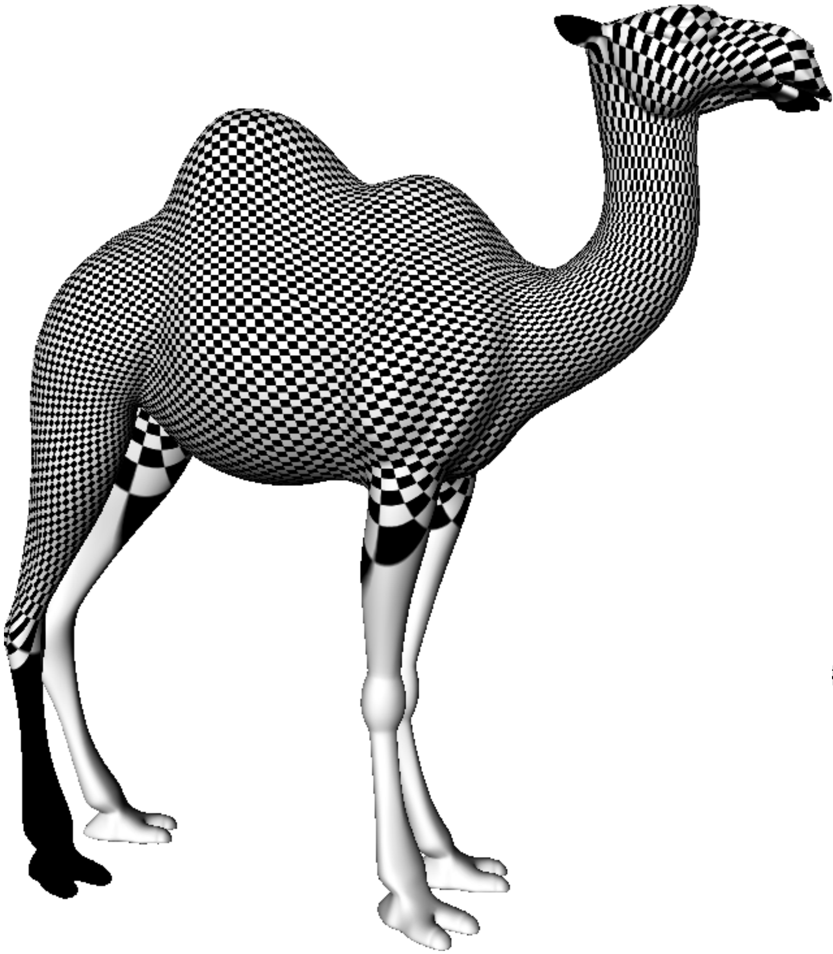


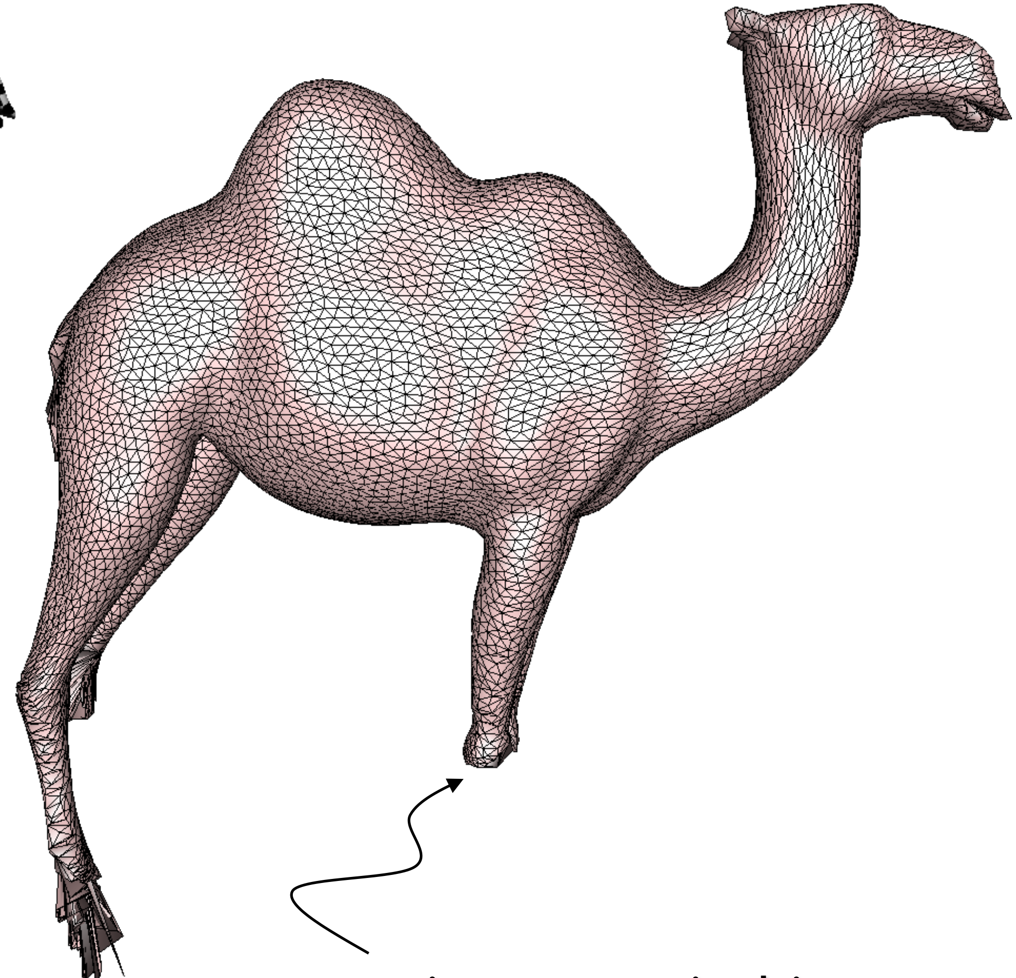


A naive cut...



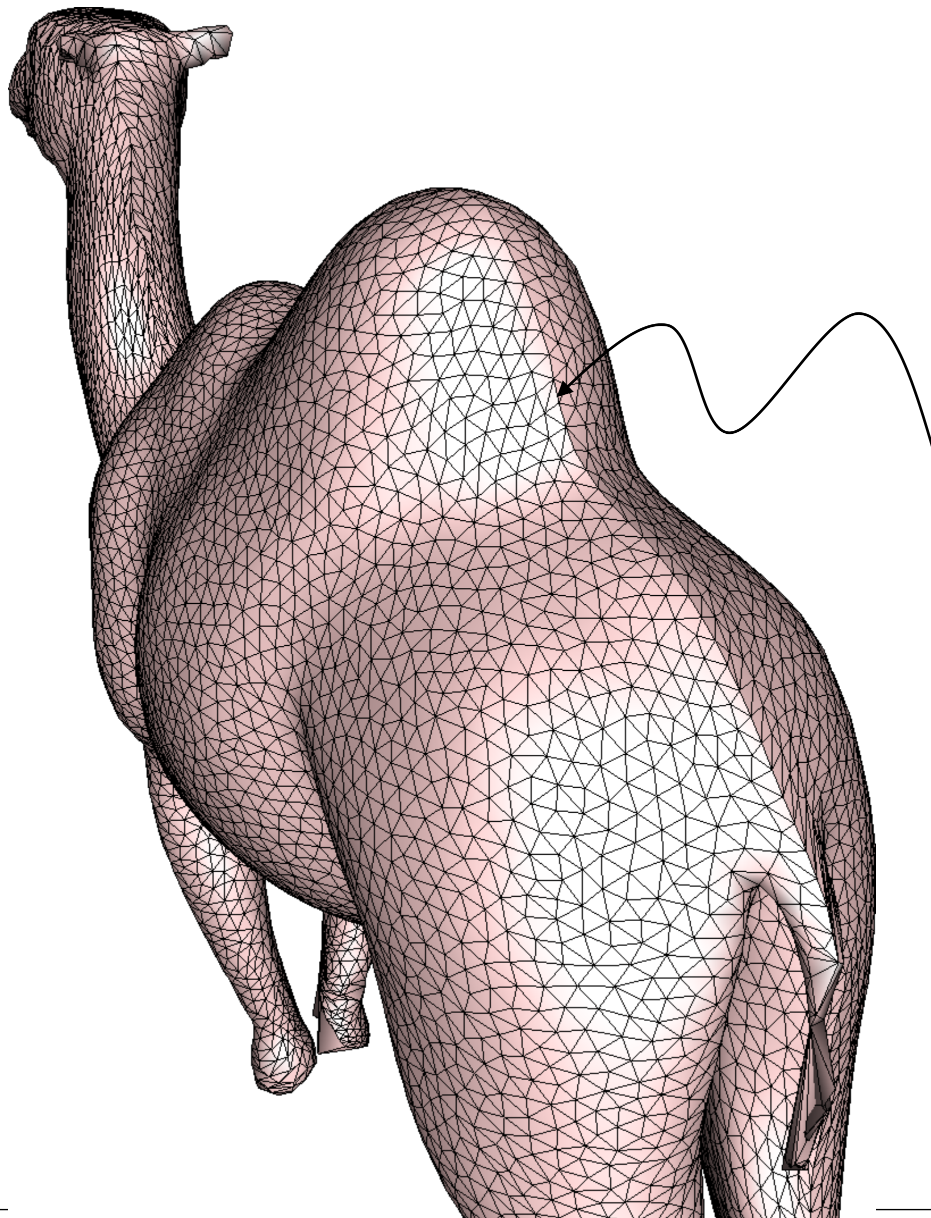
Numerical issues





serious numerical issues
with sock-like shapes

Moreover...



visible seam (cut graph has been sampled like a set of curves)

Motivation
Previous work
Contributions
Algorithm
Results
Limitations
Conclusions
Future Work

Conclusion

- Guarantee: vertex budget
- Centroidal Voronoi diagram: captures the essence of isotropic sampling
- Flexible design through density
- Handle features
- Handle important area distortion
- Still some limitations

Anisotropic Remeshing