Meshing

Xin (Shane) Li

Motivations

Digital Mesh Processing:

- Skinny triangles → harmonic weights lead to flip over
- Preprocess to handle arbitrary irregular / non uniform meshes.
- Most scanned surfaces need to undergo complete remeshing before further processing.

Classifications

Most remeshing techniques fall into :

- simplification / refinement
  - Demo: Using our available tools “Meshsimplify.exe + Filtermesh.exe”
- optimization
- resampling (point sampling)

Control over:

- vertex density
- shape of elements
- etc.

Isotropic Surface Meshing

"Isotropic Surface Remeshing"

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

IEEE International Conference on Shape Modeling and Applications, 2003

Previous work

Applications in Two different fields:

Finite Element community:

High-quality meshes

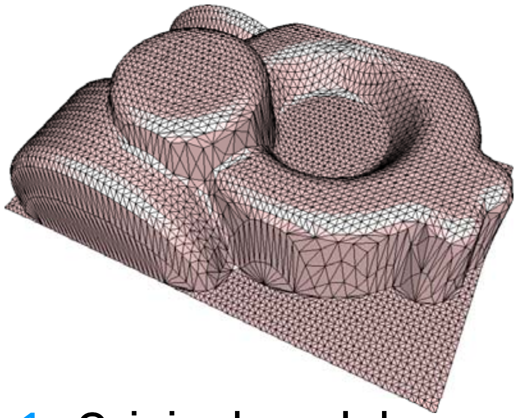
for simulation

Computer Graphics community:

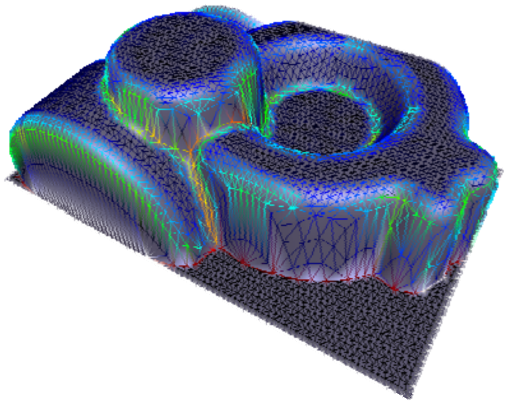
Geometric modeling

for effective processing and fast display

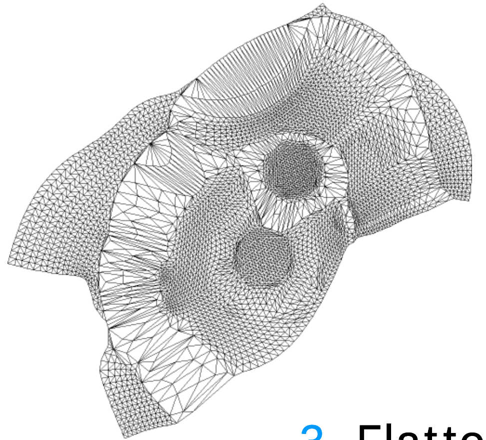
Remeshing Pipeline in this Paper



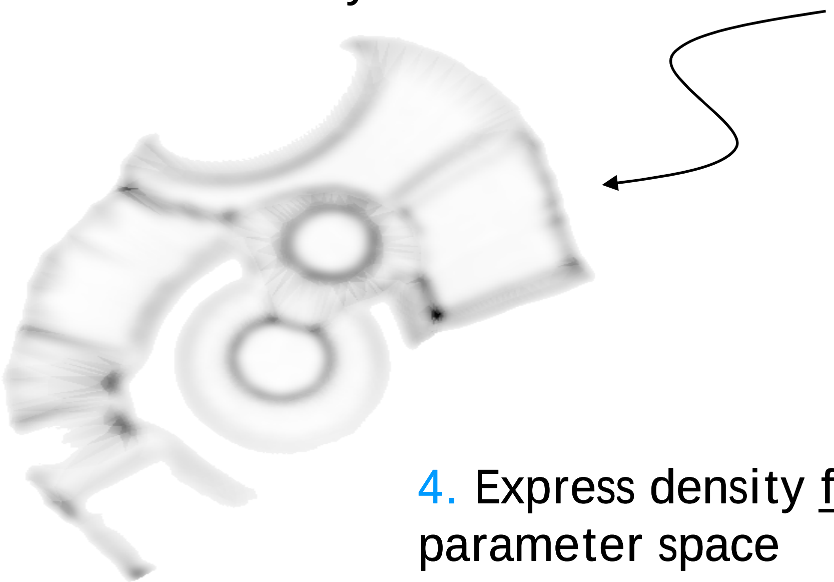
1. Original model



2. Measure density



3. Flatten it



4. Express density <u>function</u> in parameter space
5. Resample this function

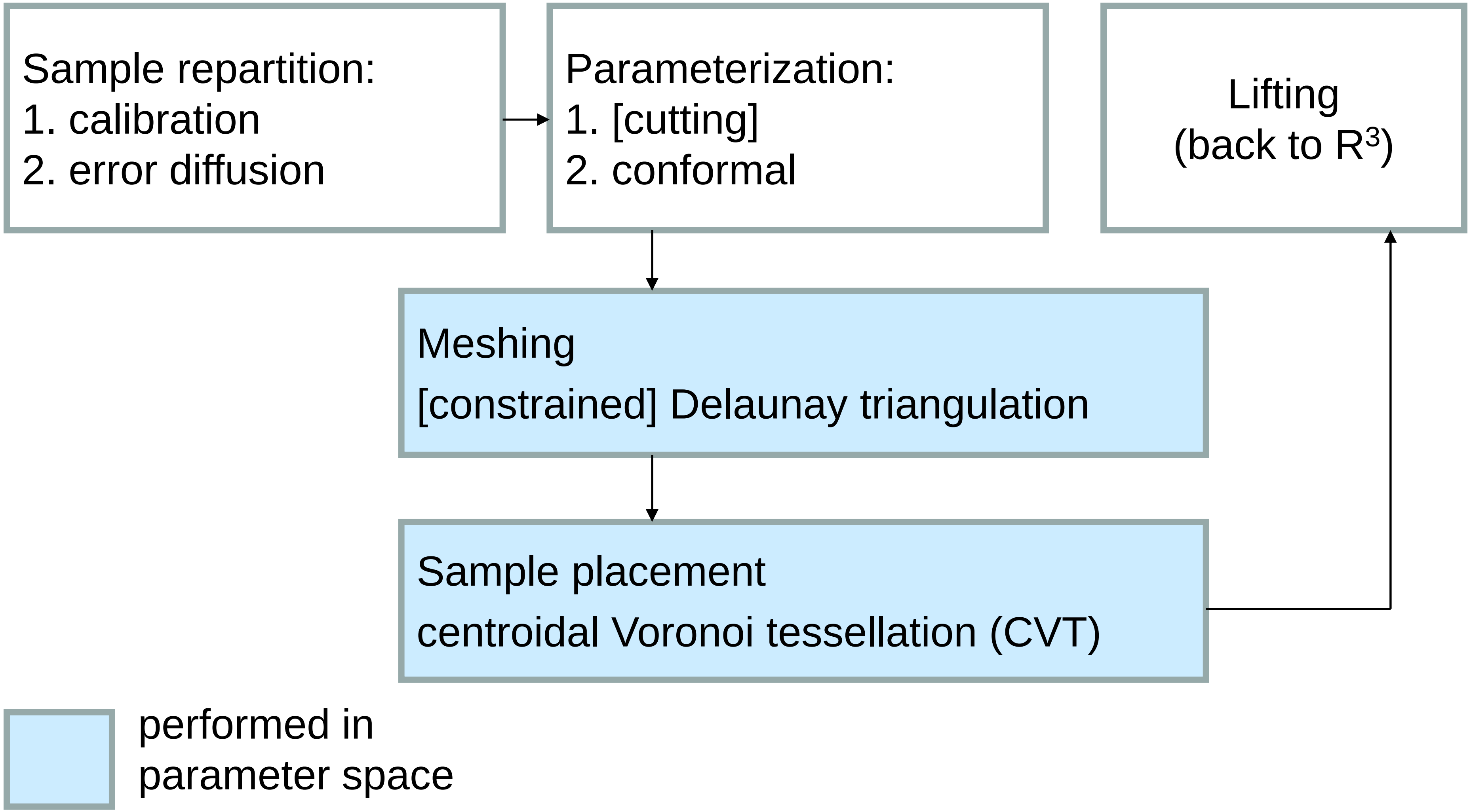
More precisely...

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

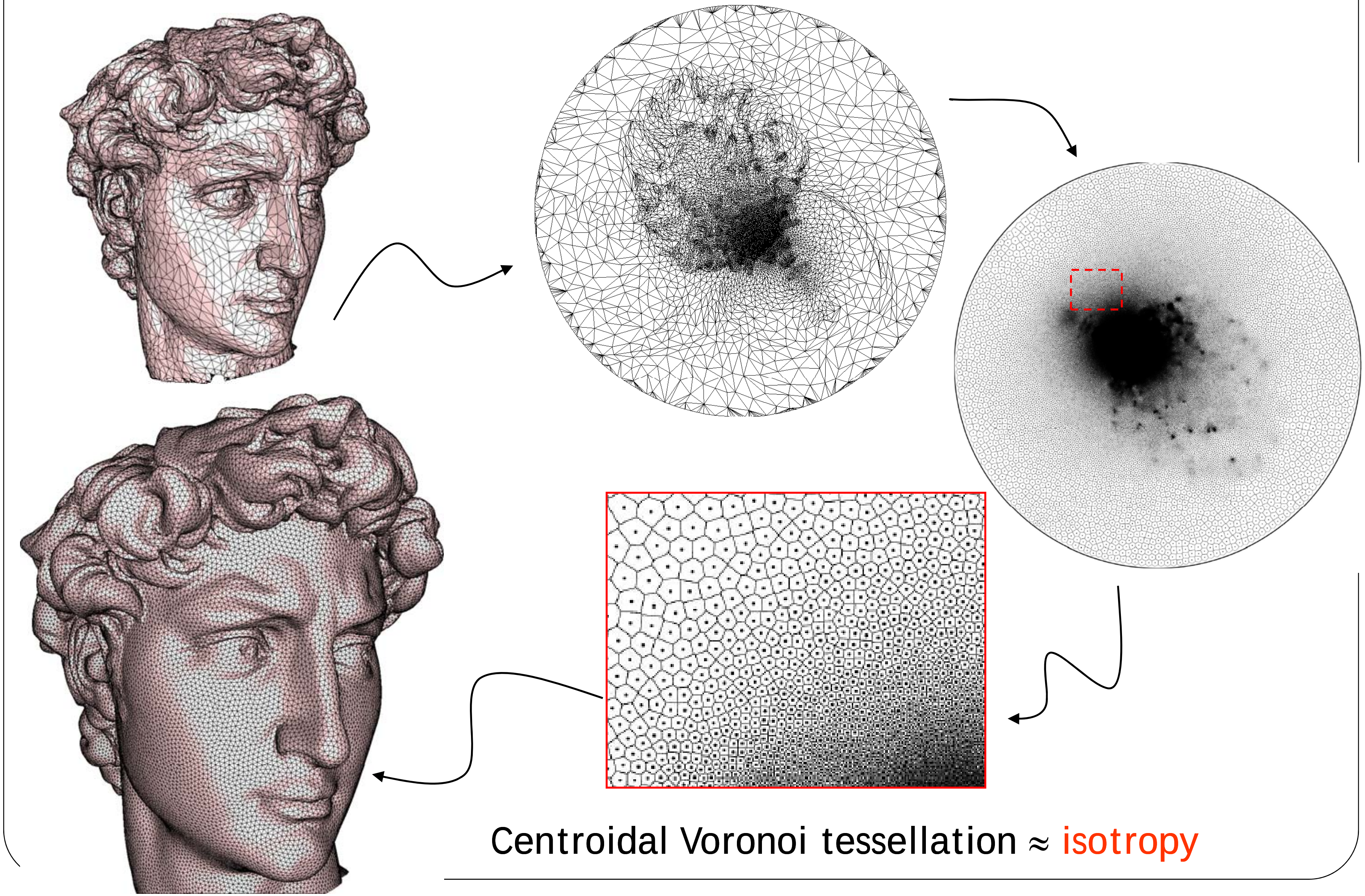
Solution

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

Remeshing Pipeline



Key idea



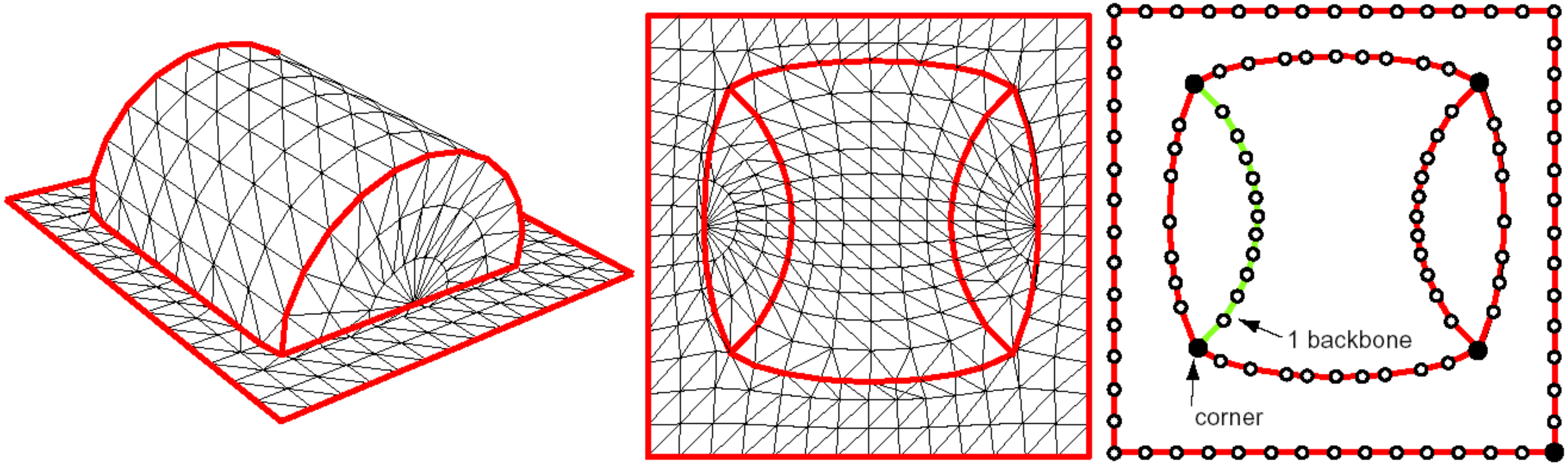
Centroidal Voronoi tessellation ≈ isotropy

Preliminaries

Input

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

Feature skeleton

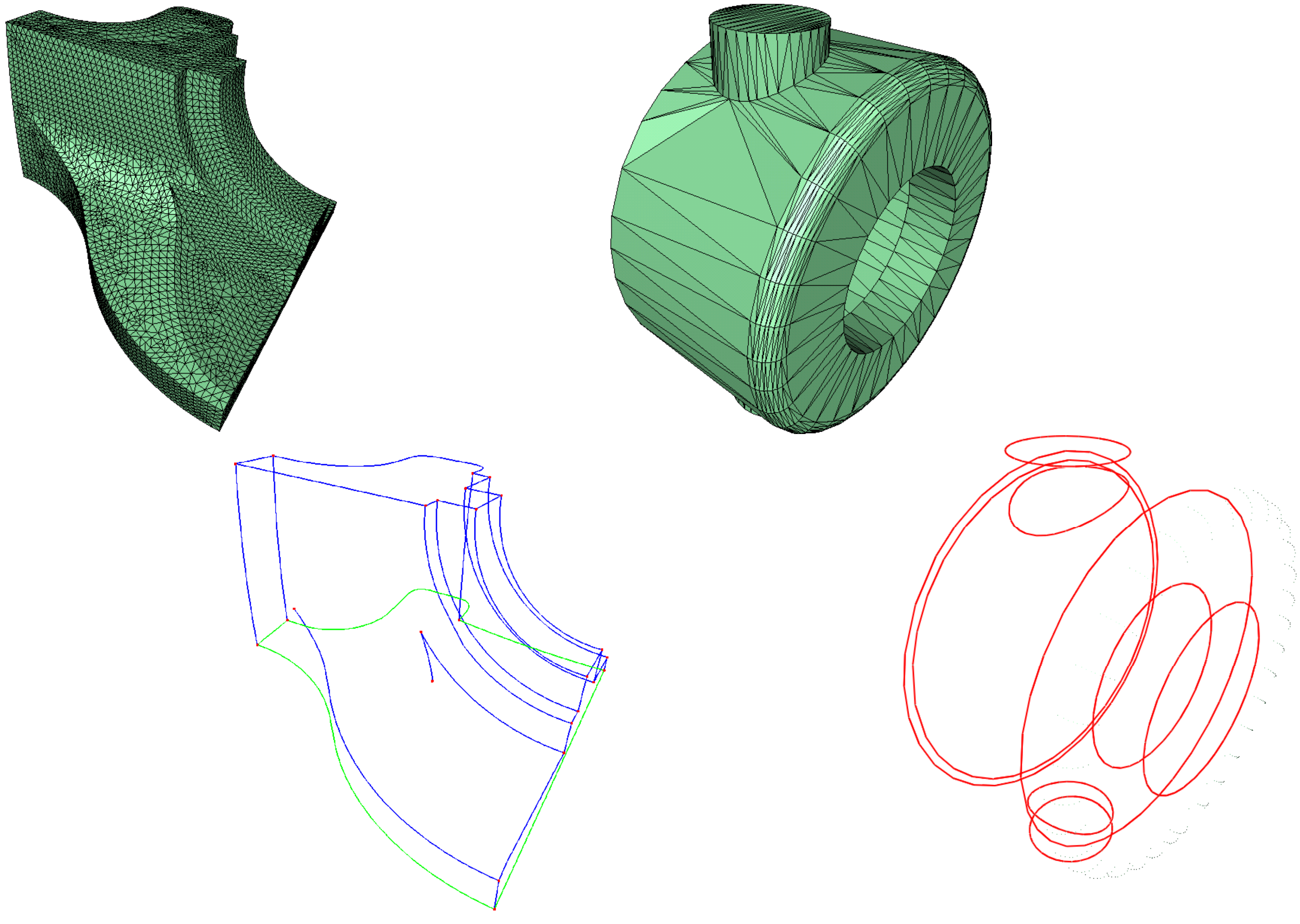


Original model Parameterization and tagged edges Feature skeleton

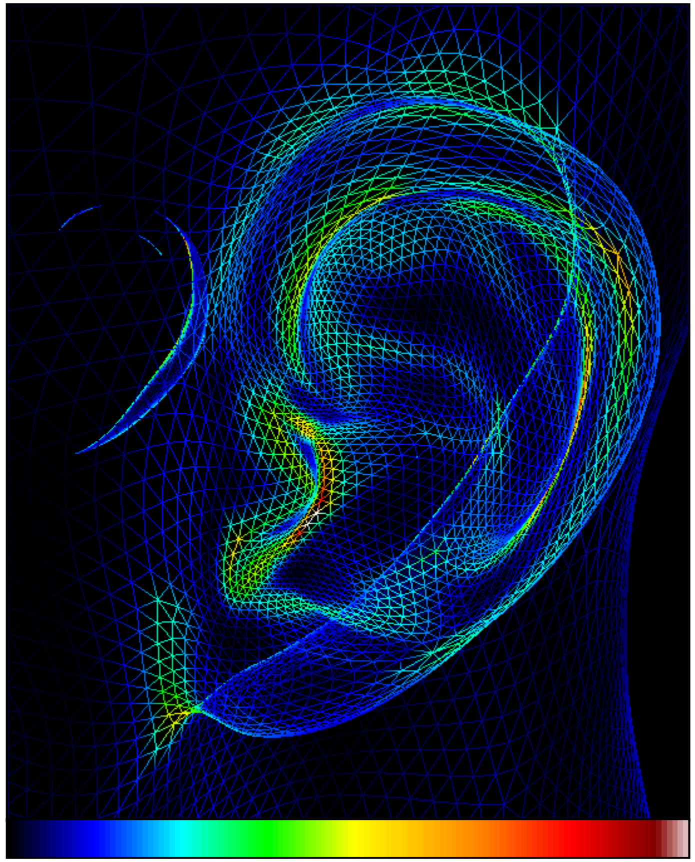
0-manifolds: corners
1-manifolds: backbone: feature edges chained together

feature skeleton: corners + backbones

More feature skeletons



Example of density function



Curvature related density function

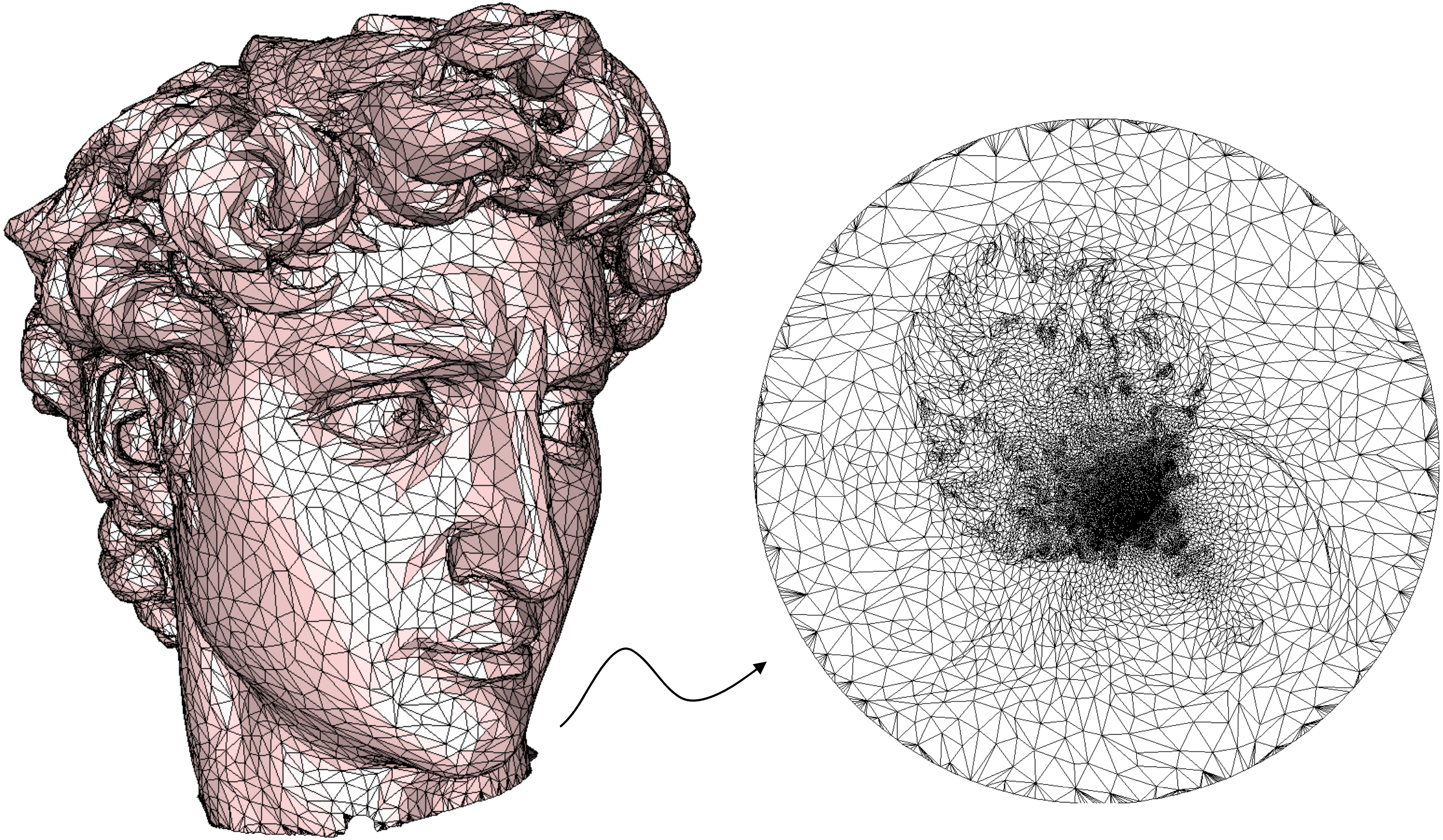
Discrete Differential-Geometry Operators

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

Algorithm

- parameterization
- meshing
- sample placement

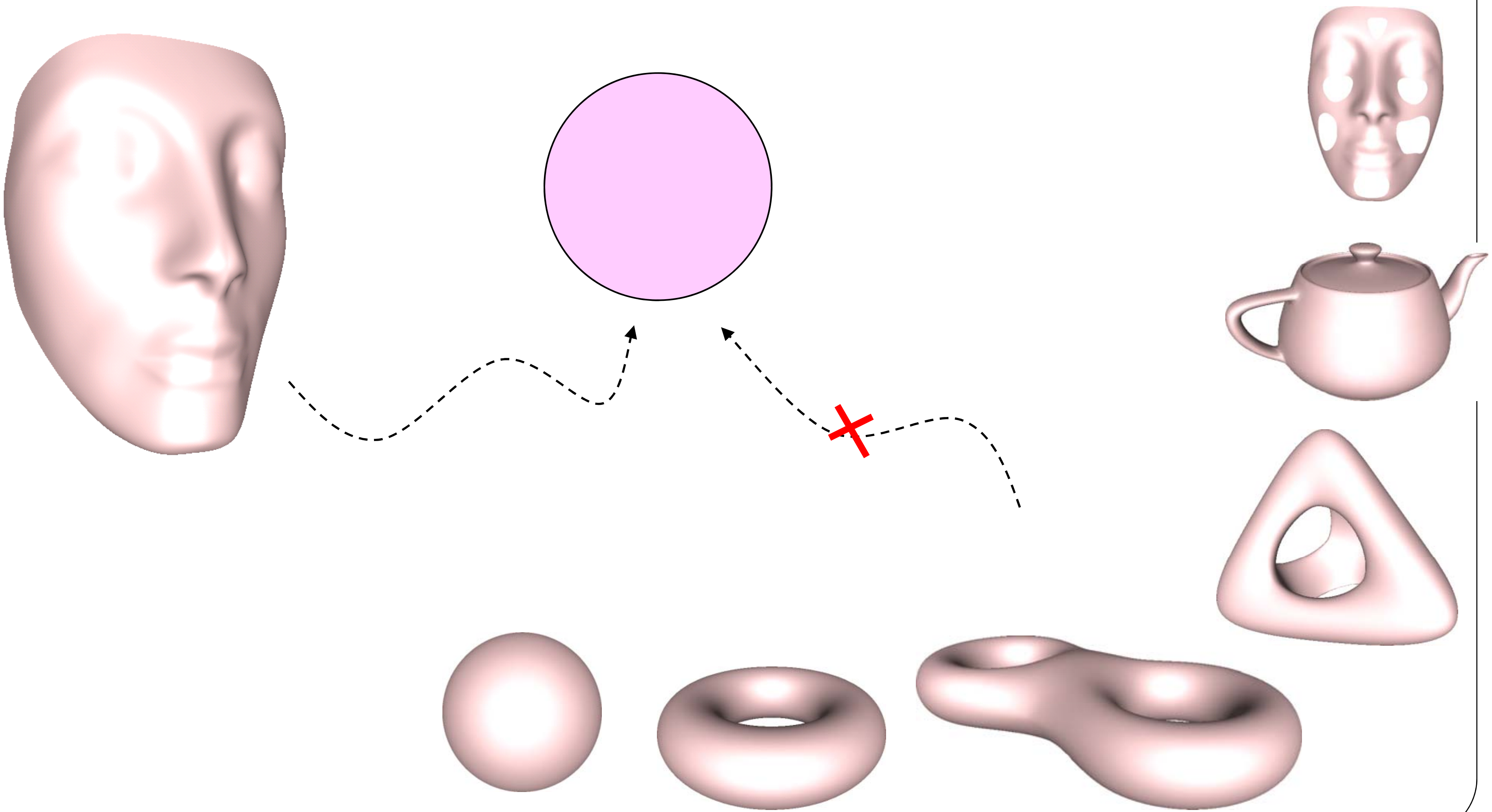
Parameterization



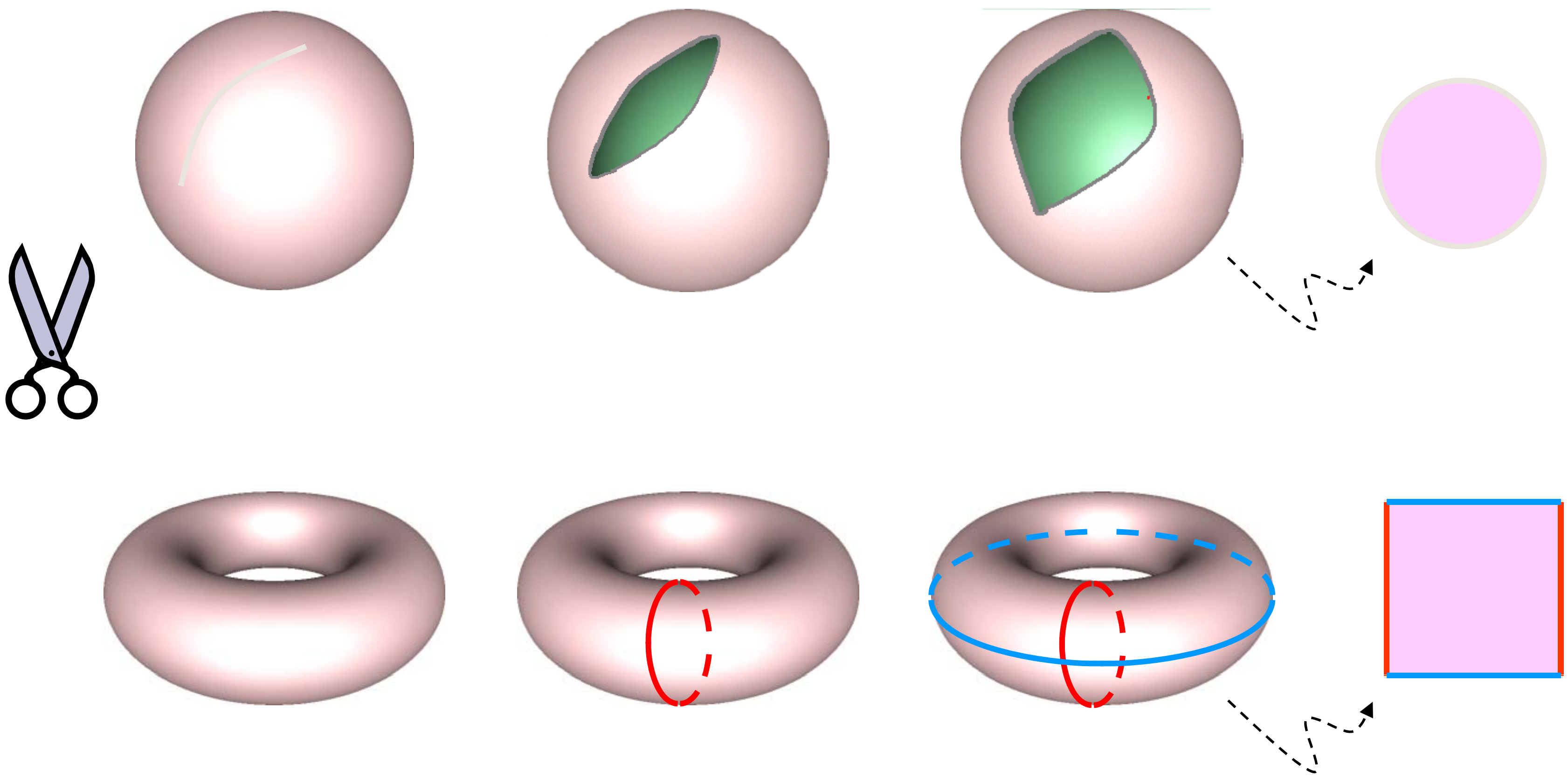
~remove embedding (back to manifold)

Parameterization

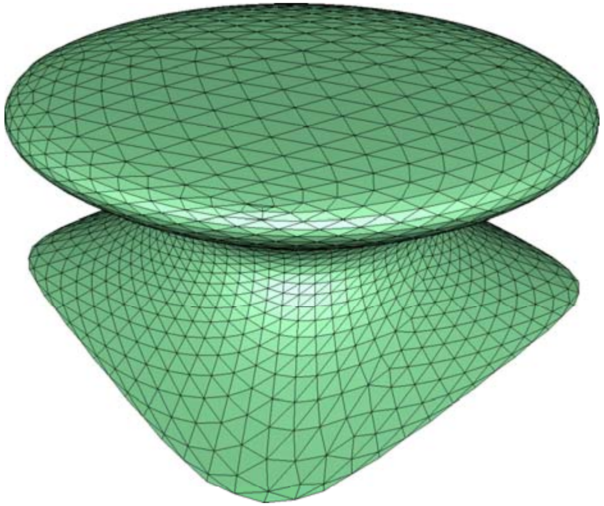
-> simple domain

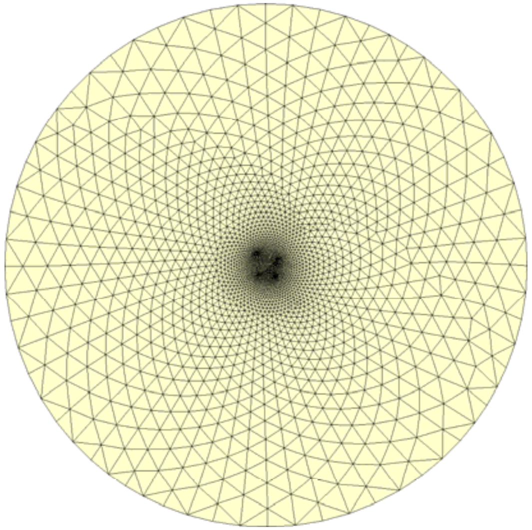


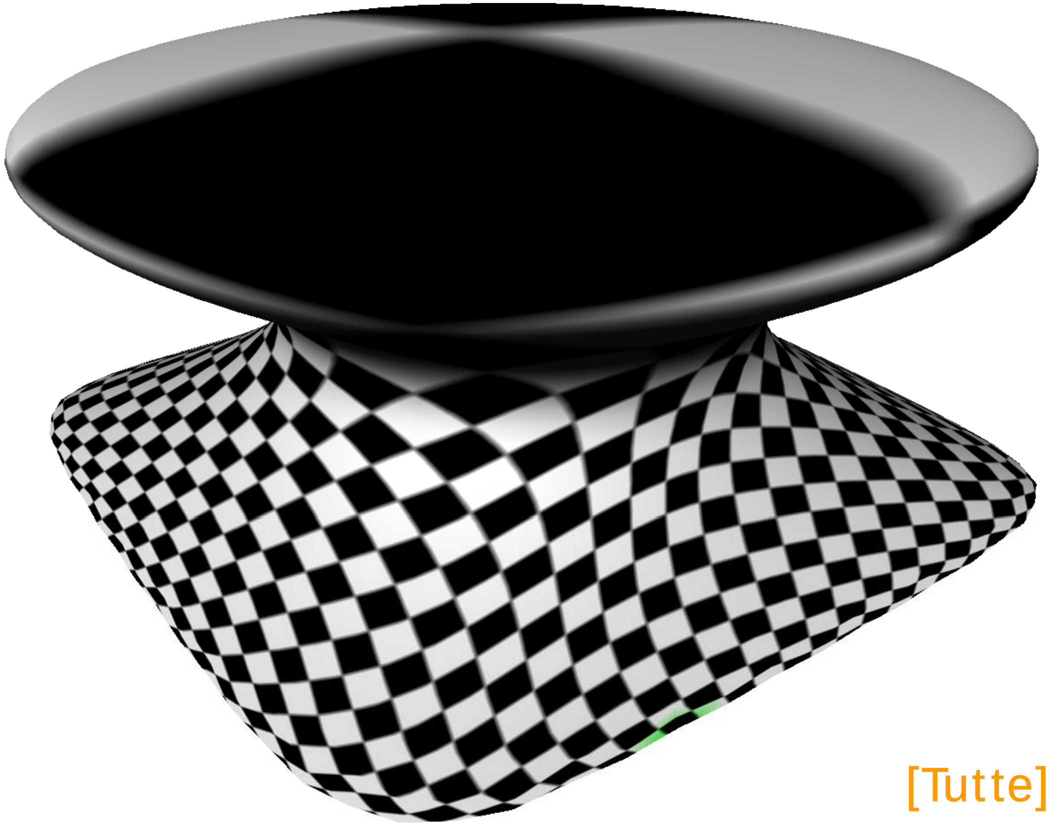
Surface cutting



Convex embedding

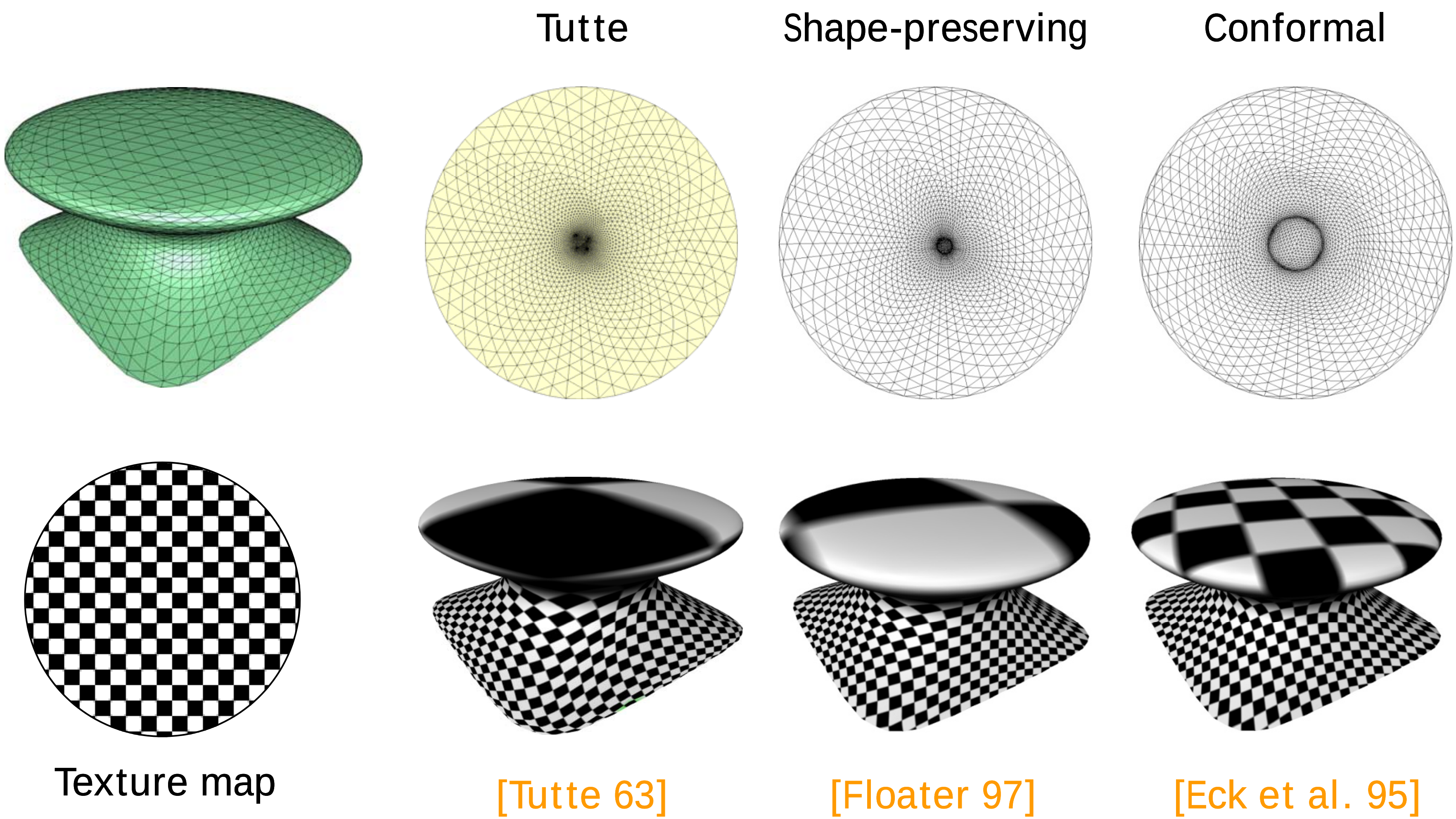




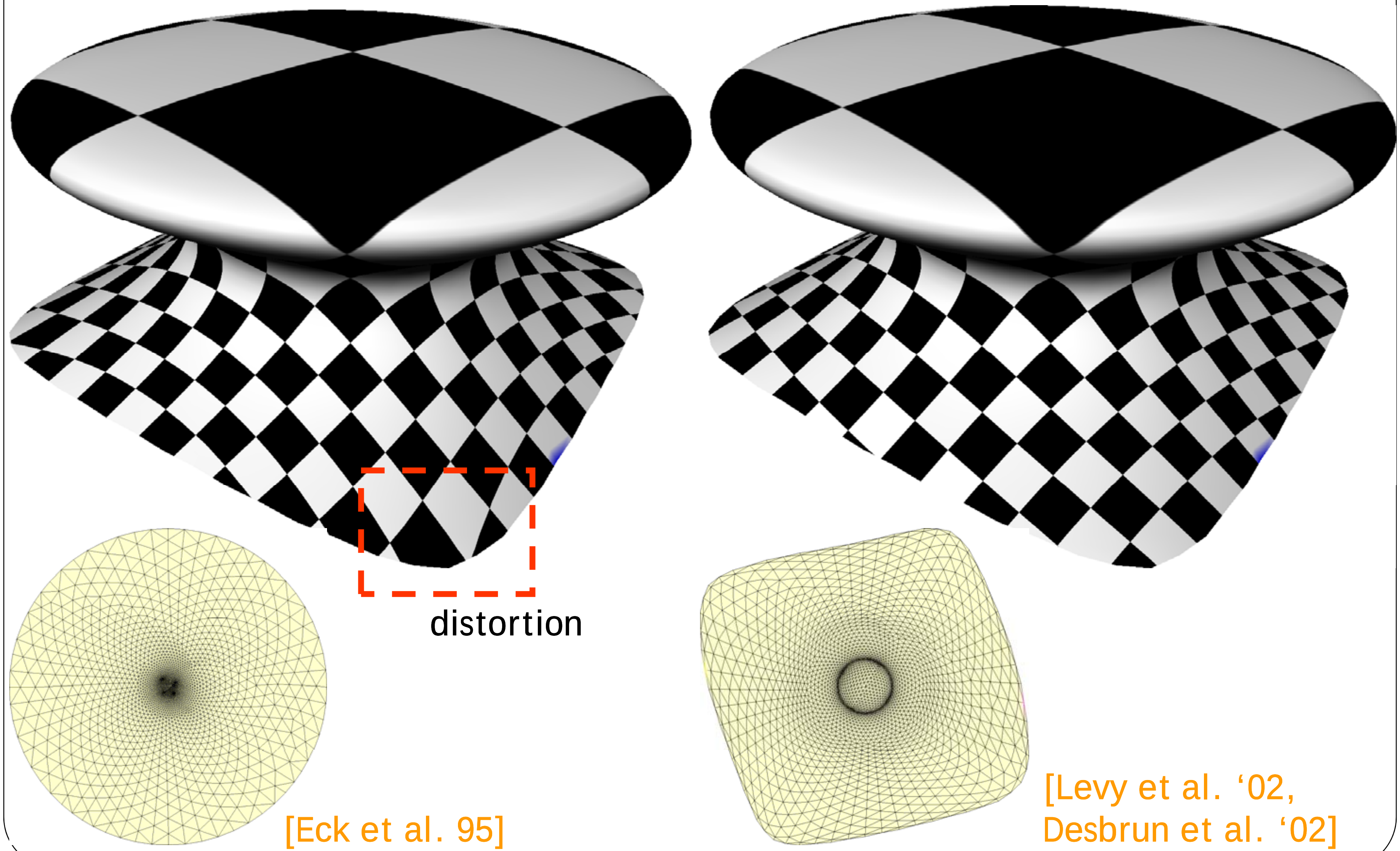


[Tutte]

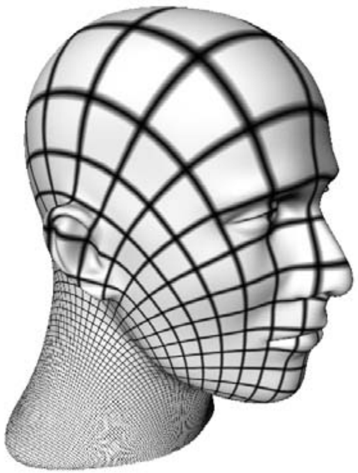
More parameterizations



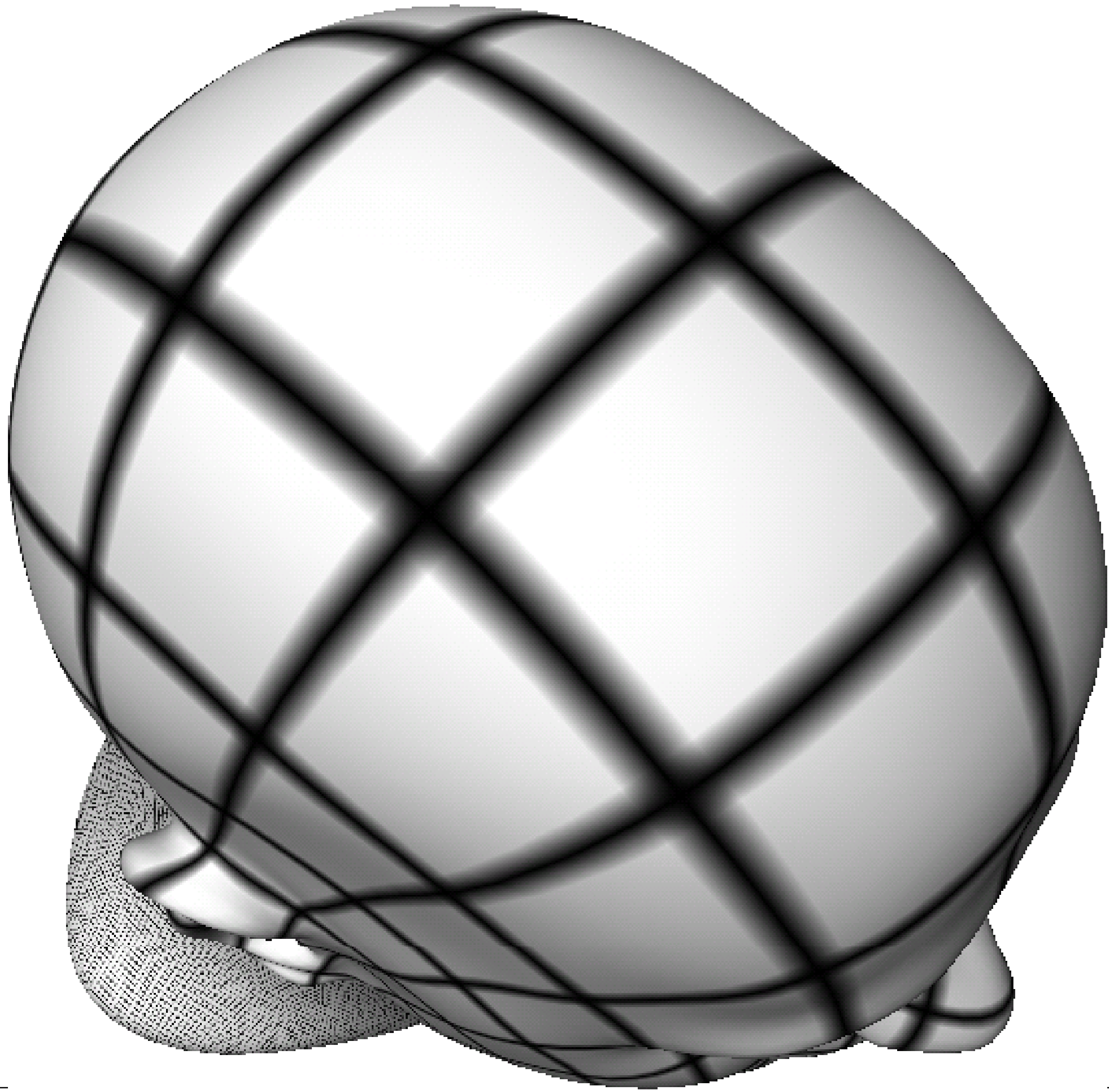
Conformal: fixed vs free boundary



Preservation of angles

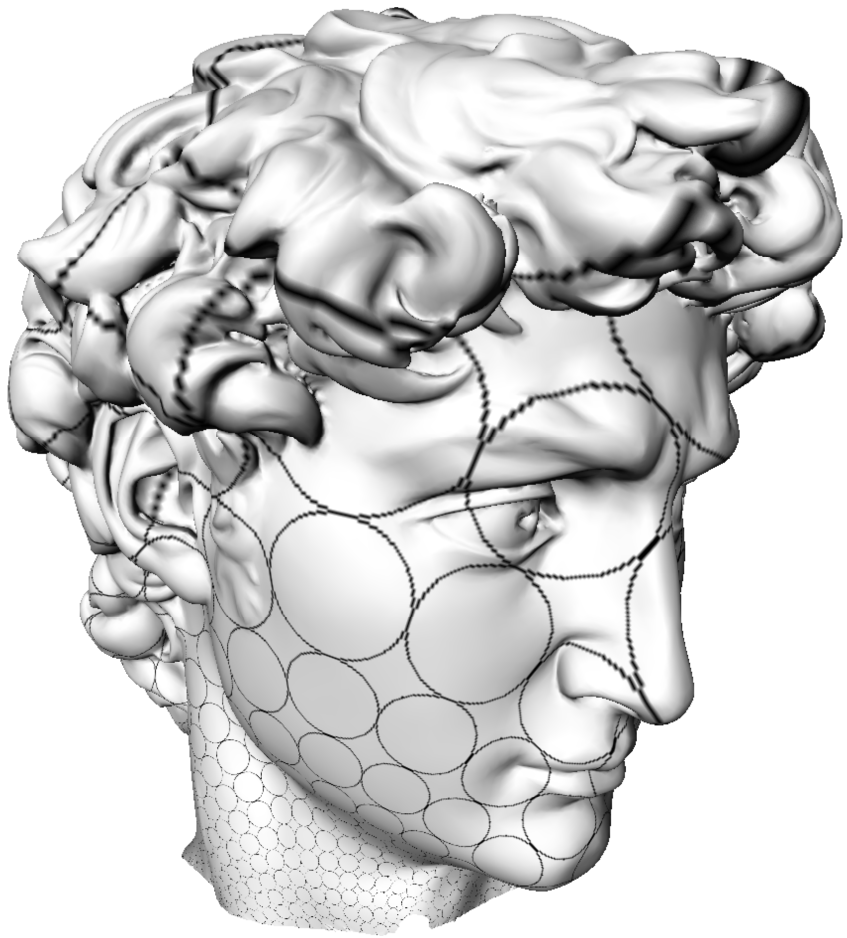


Isoparametric lines

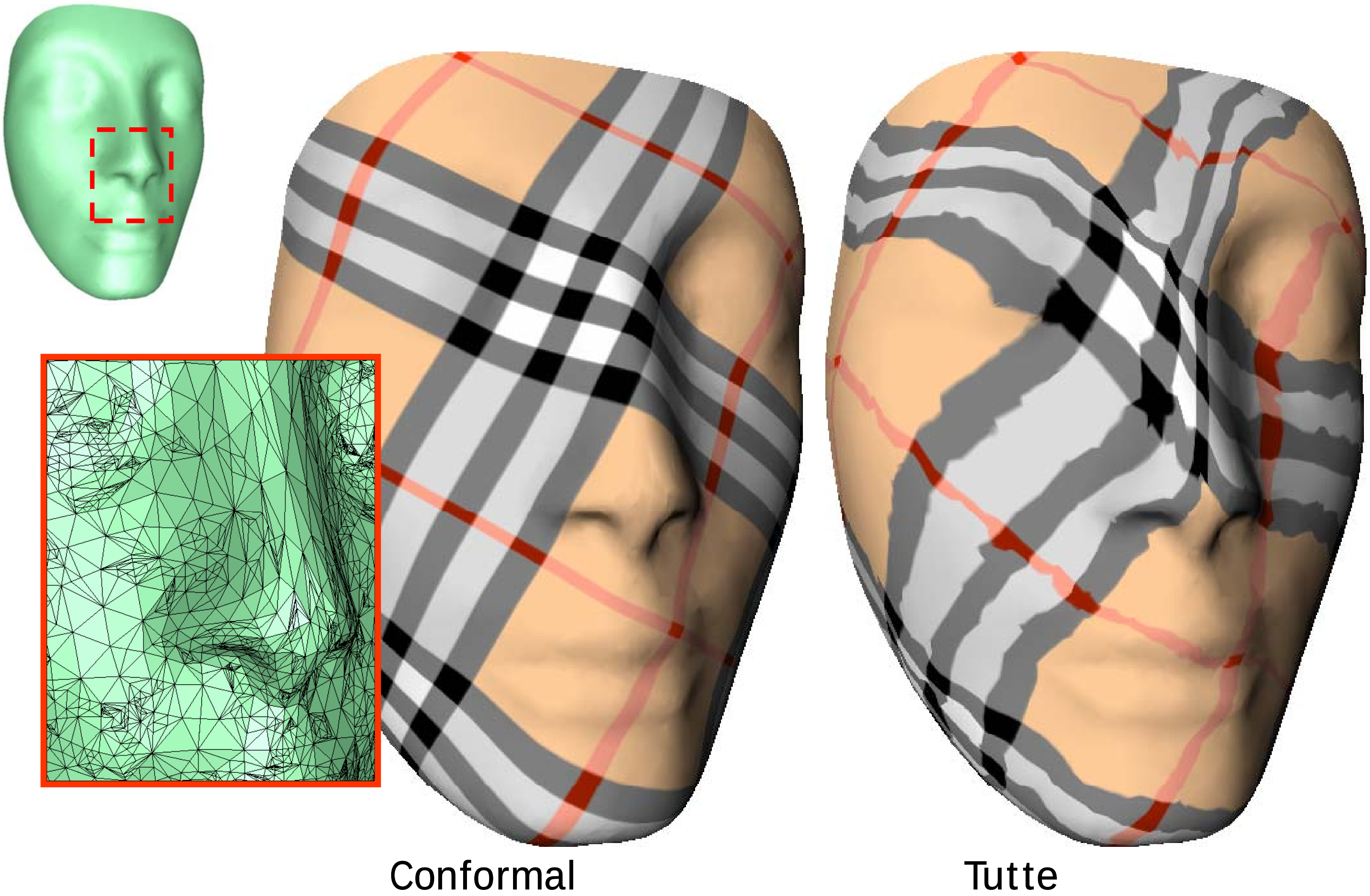


Conformality

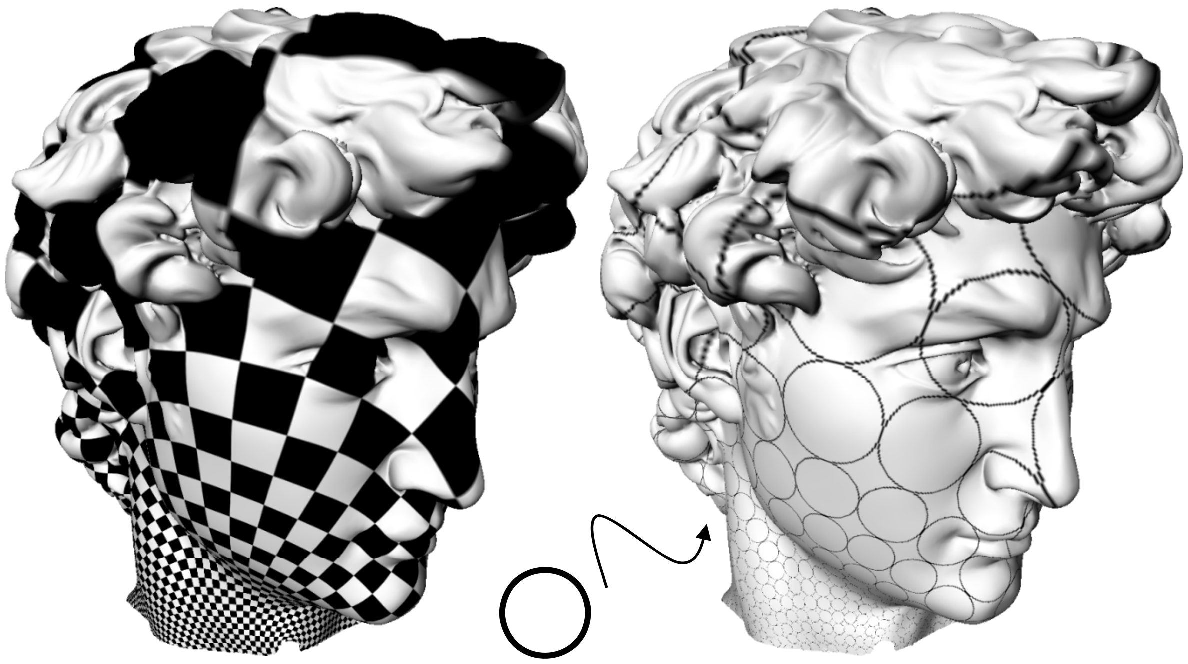
Angle-preserving + locally isotropic



Behavior w.r.t. sampling

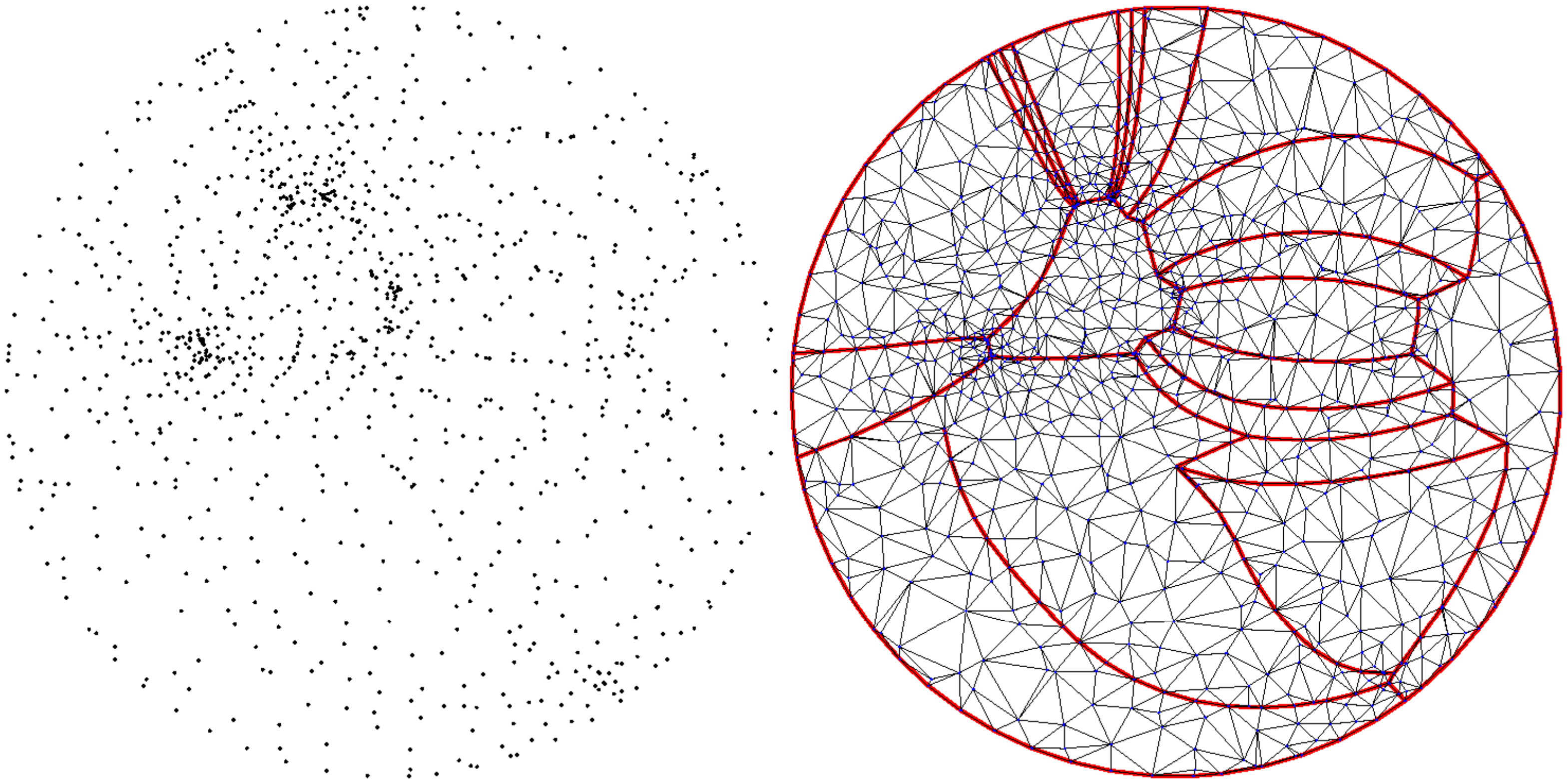


"A well-shaped element in parameter space will not be deformed too much once lifted in embedding space"



- Motivation
- Previous work
- Contributions
- Algorithm
  - sample repartition *error diffusion*
  - parameterization *conformal*
  - meshing
  - sample placement

Meshing



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

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

Sample placement

Given a bounded domain and a density function,

sampling

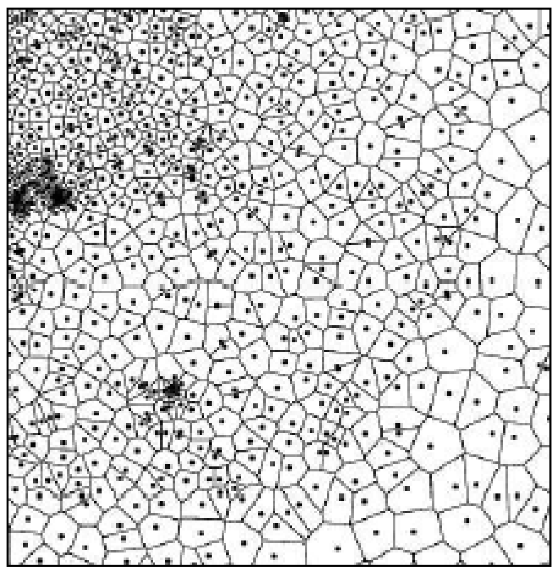
=

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

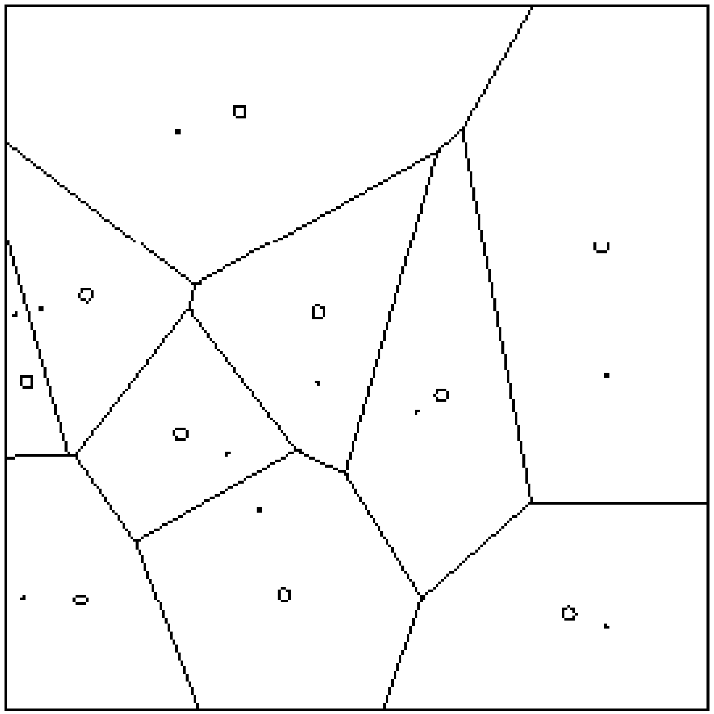
Sample placement

- partitioning the domain
  -> Voronoi tessellation

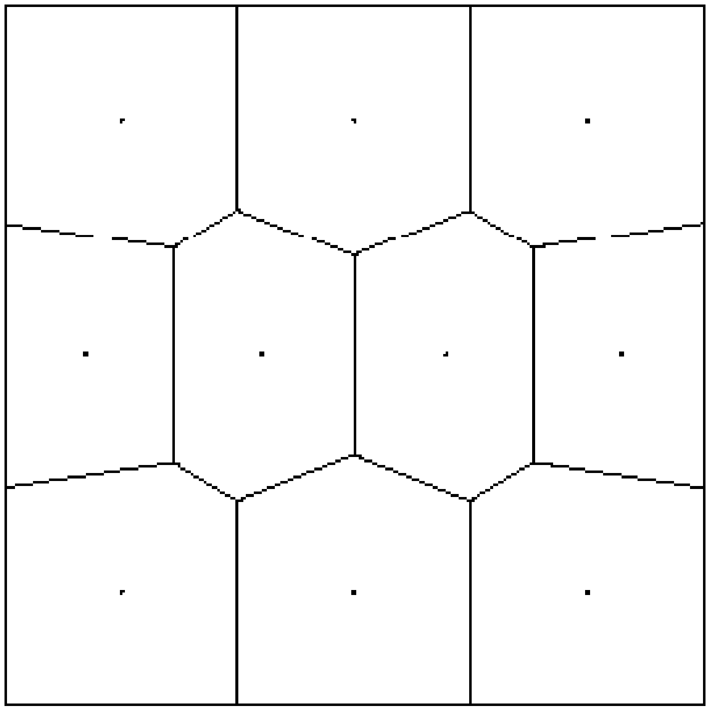
- repartitioning the density function among a set of samples
  = Equal-mass enclosing



Centroidal Voronoi diagram



Ordinary Voronoi diagram

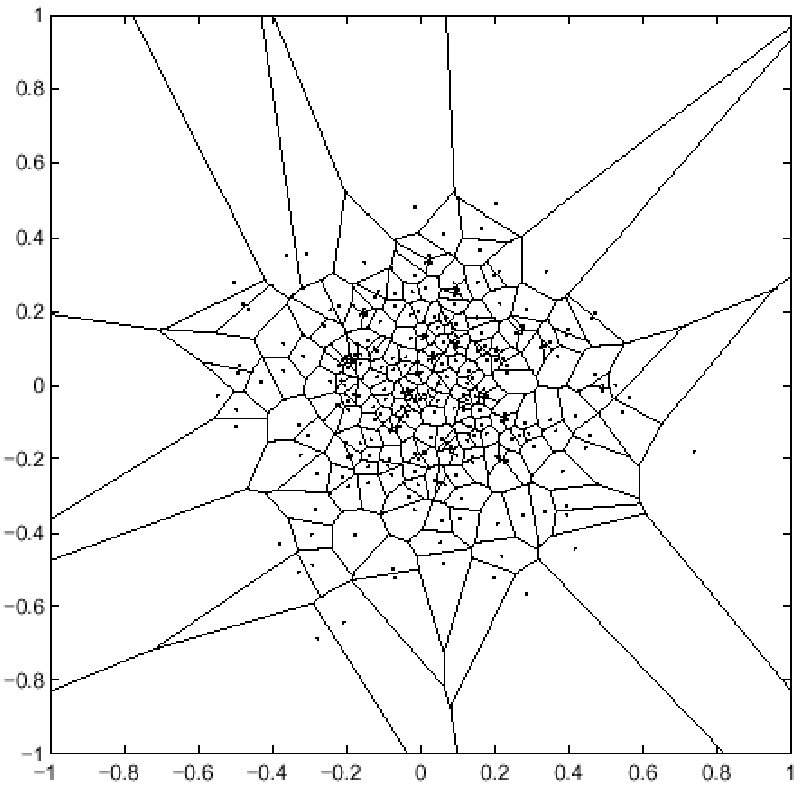


Centroidal Voronoi diagram

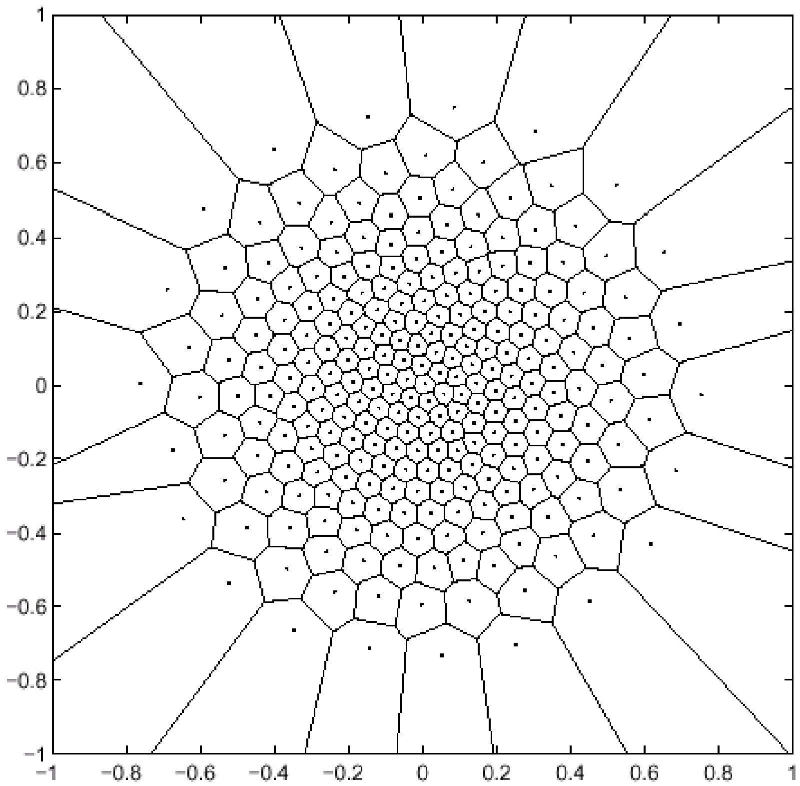
- ● sites
- ○ centroids

Sites coincide with centroids (center of mass)

Weighted Centroidal Voronoi diagram



Monte-Carlo



WCVD

Non-uniform density

Centroidal Voronoi diagram

Used for:

- optimal clustering
- optimal repartition of resources
- quantization
- tiling, etc. [Du *et al.* 01]

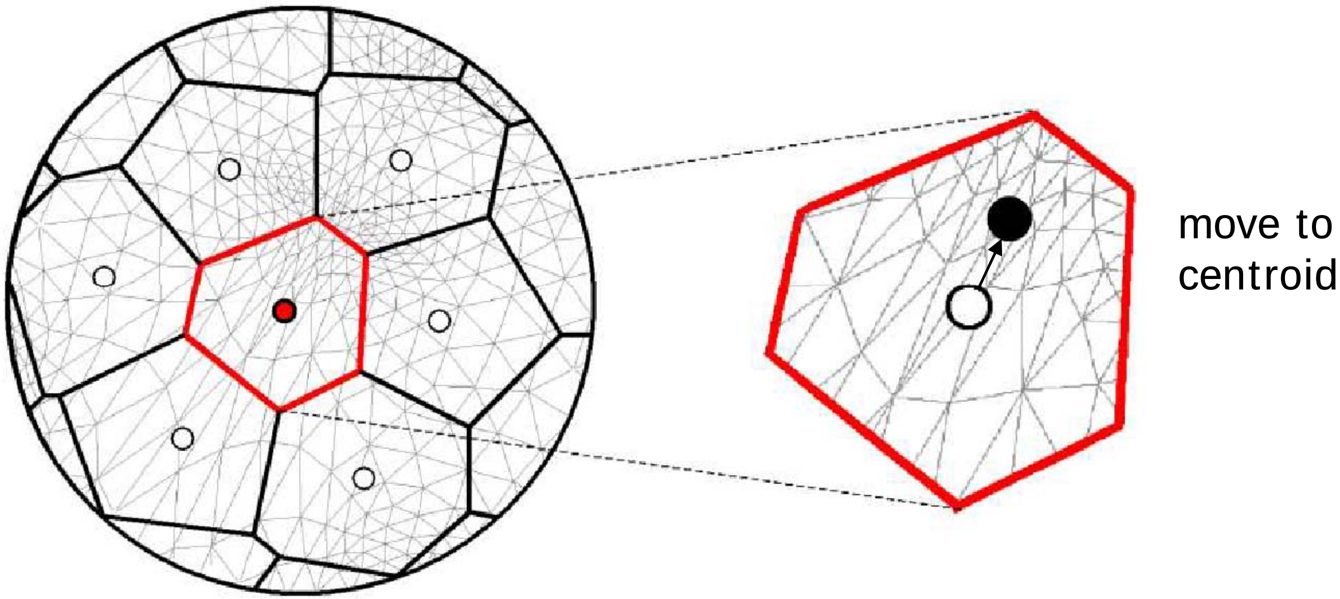
Note:

- special configuration, not algorithm
- several algorithms: Lloyd, k-means, etc.
- works in nD

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

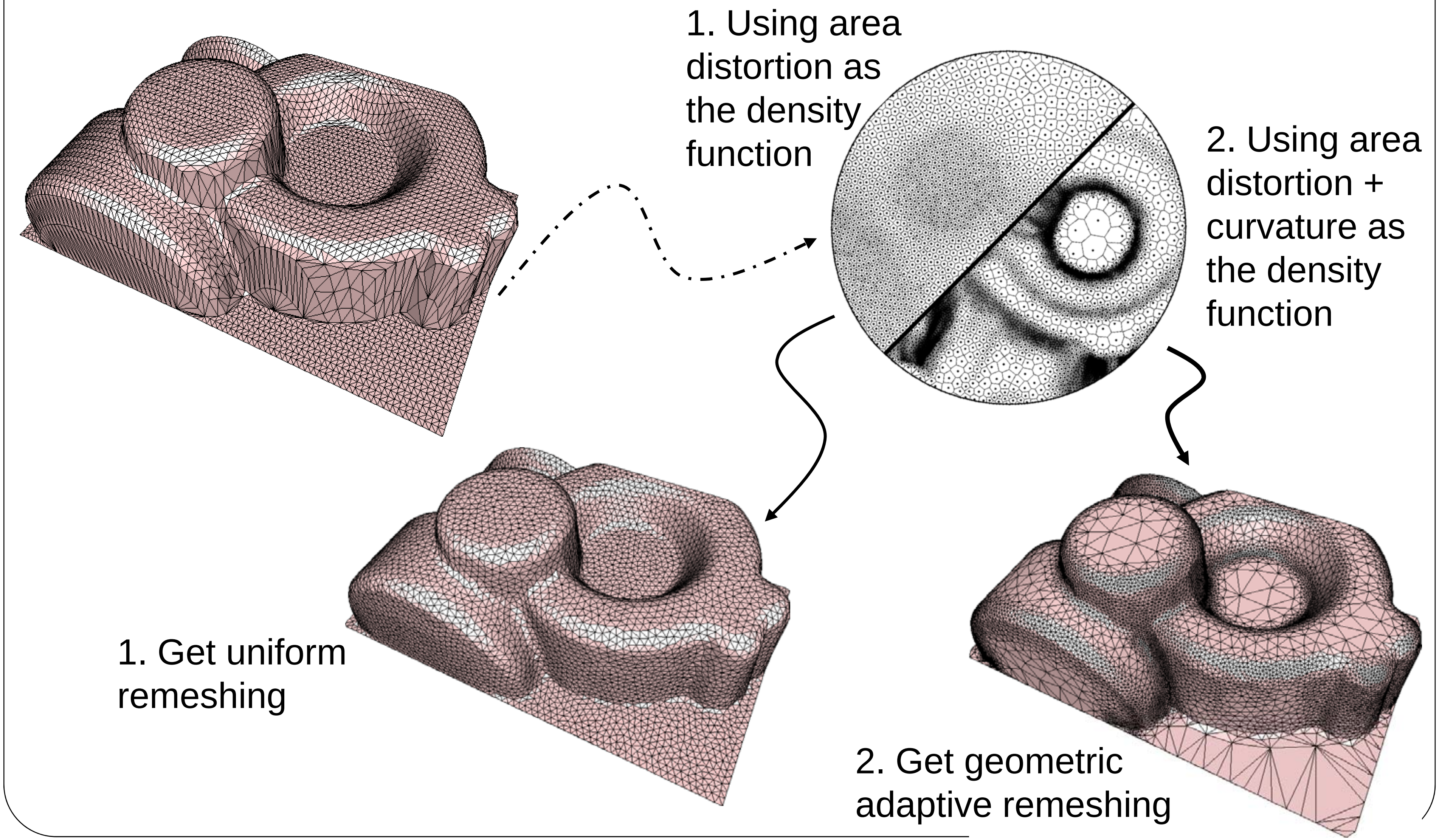
Results

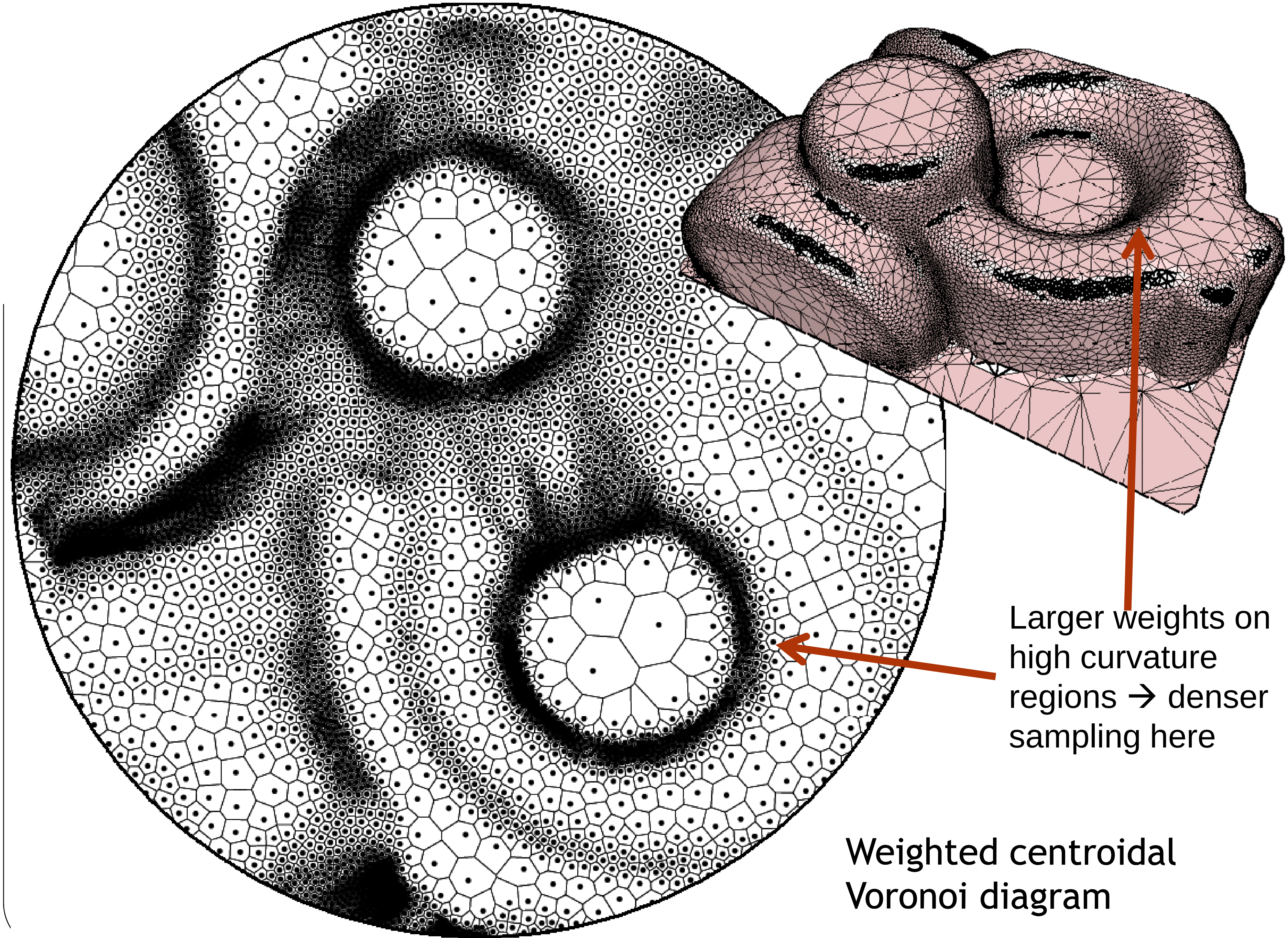
Limitations

Conclusions

Future Work

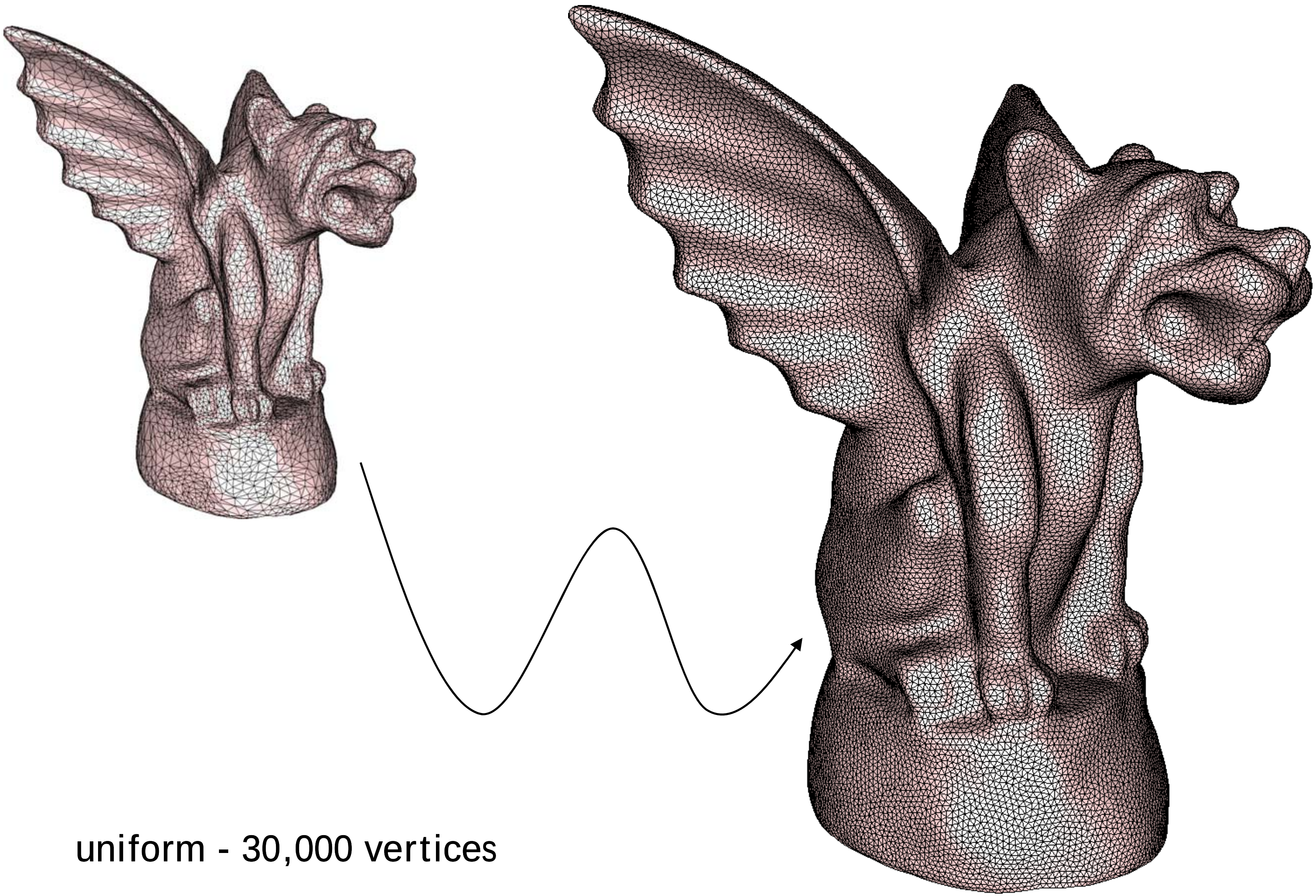
Uniform vs curvature-adapted



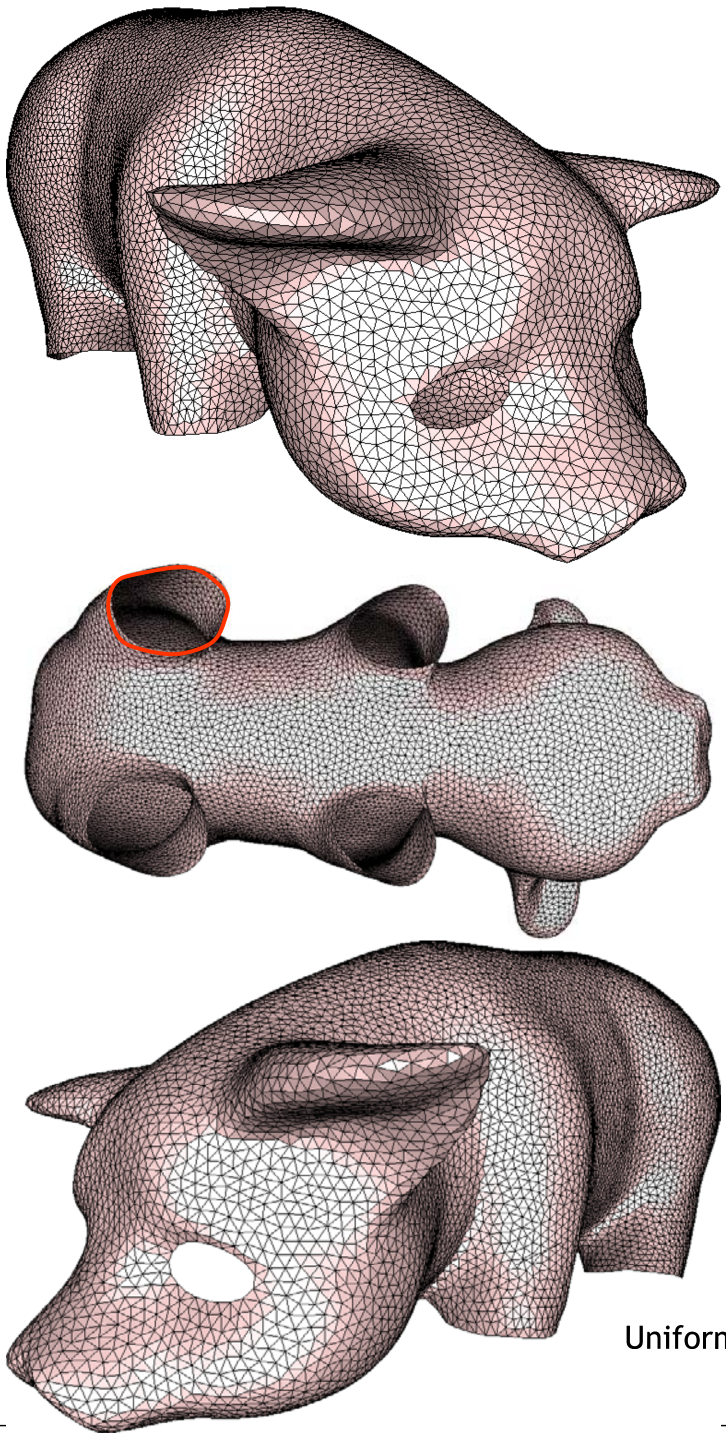


Weighted centroidal Voronoi diagram

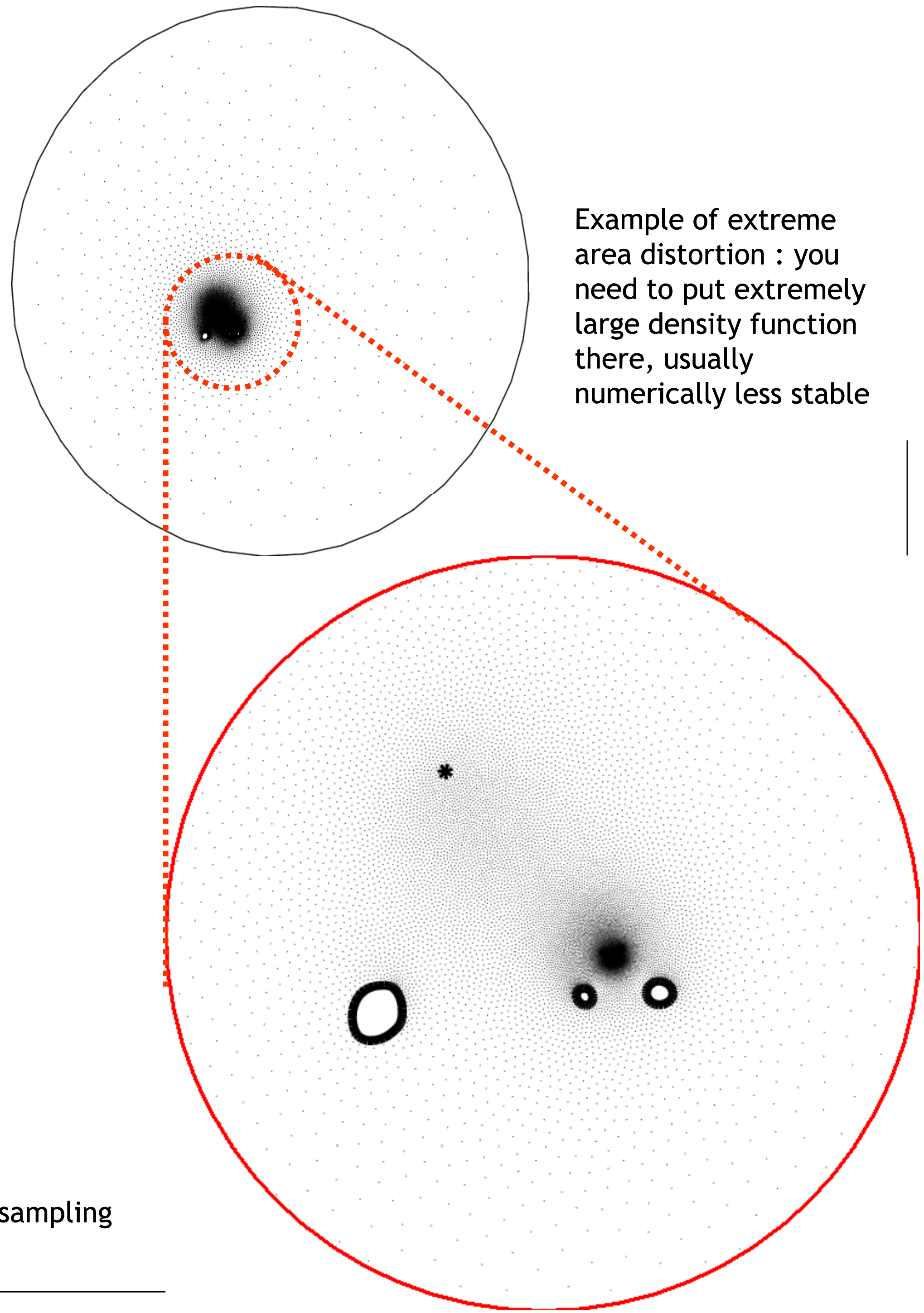
Gargoyle



uniform - 30,000 vertices



Uniform sampling

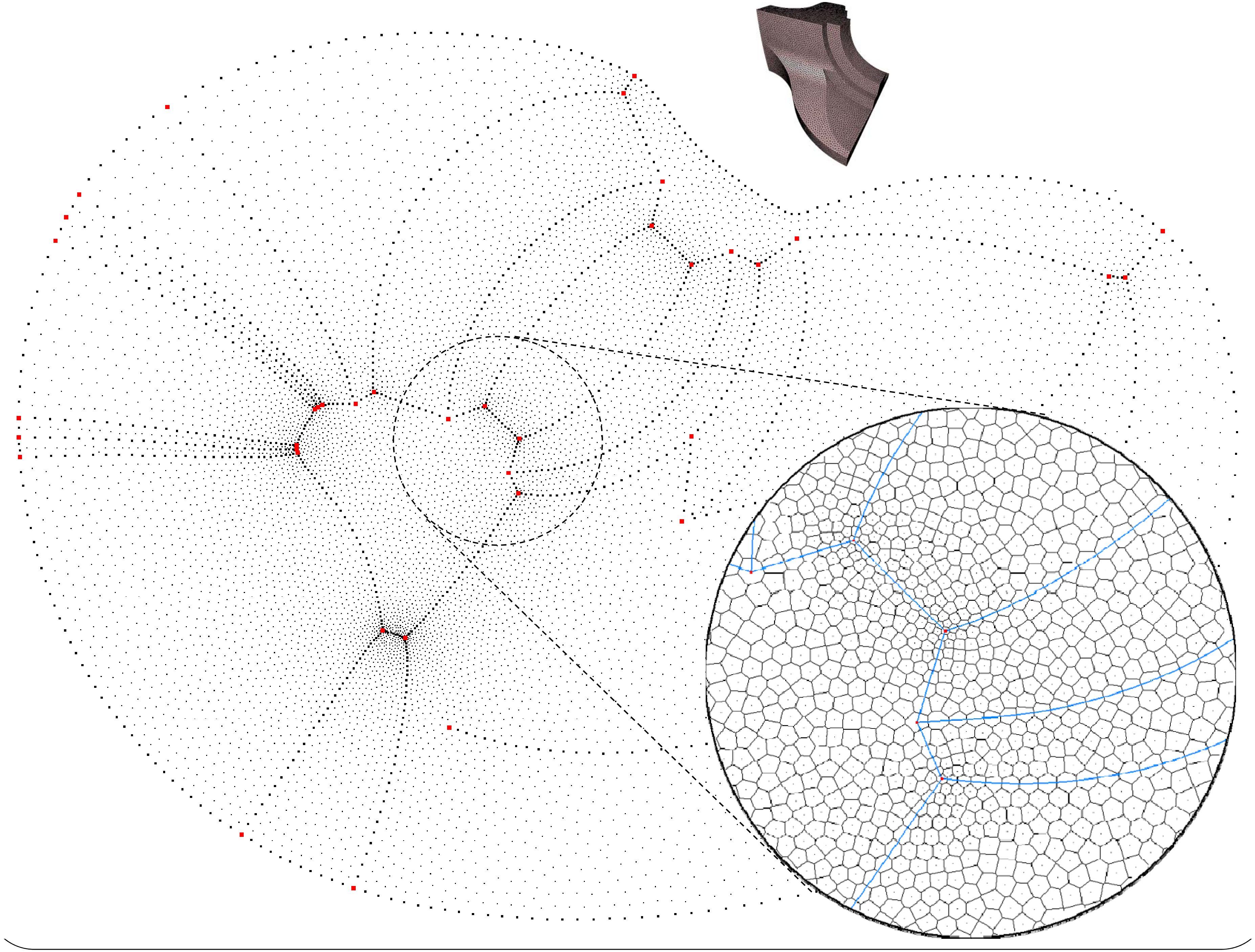


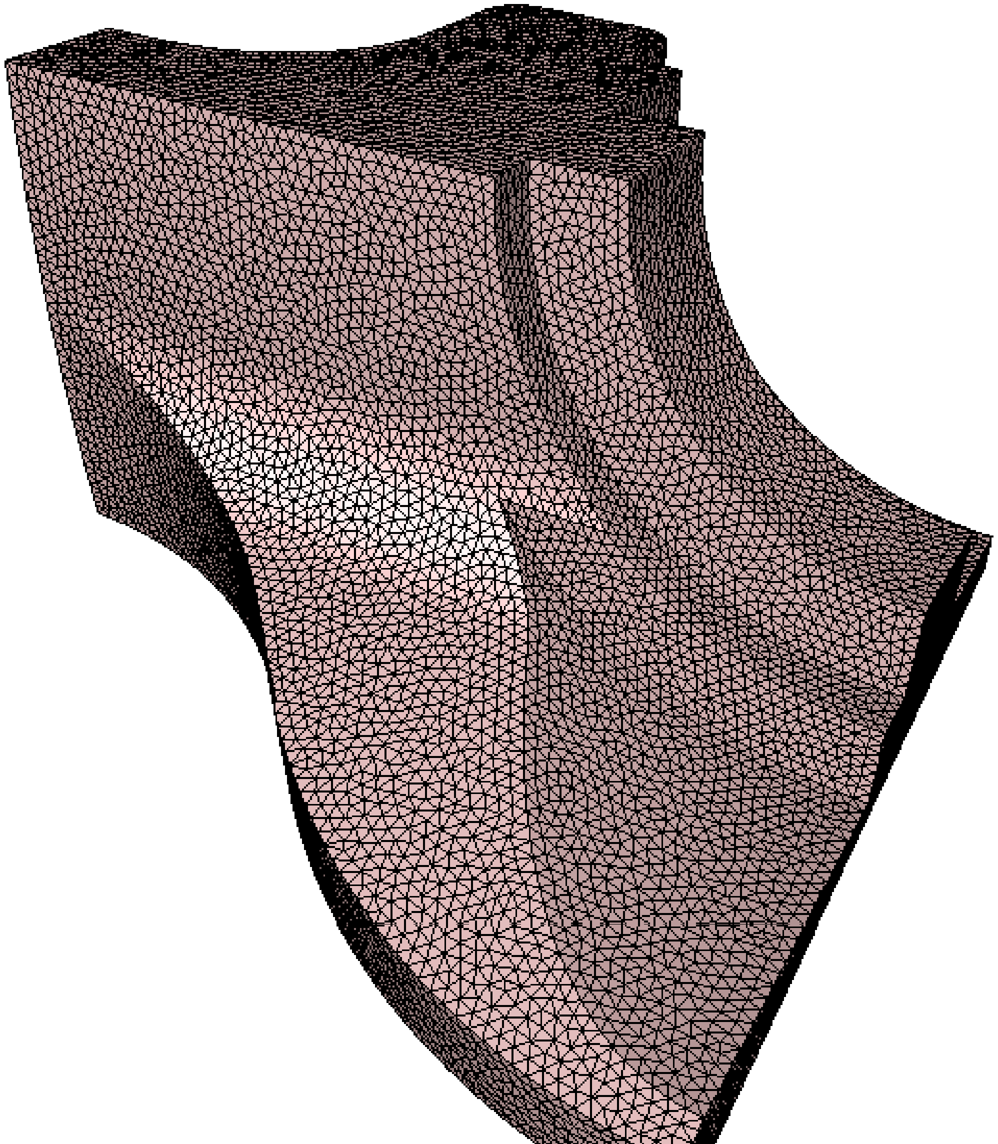
Example of extreme area distortion : you need to put extremely large density function there, usually numerically less stable

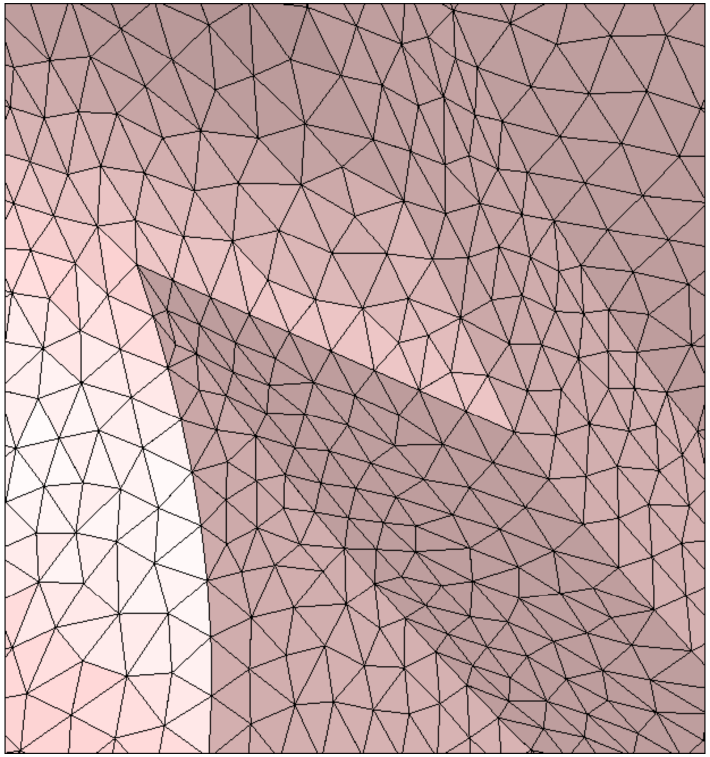
CAD models

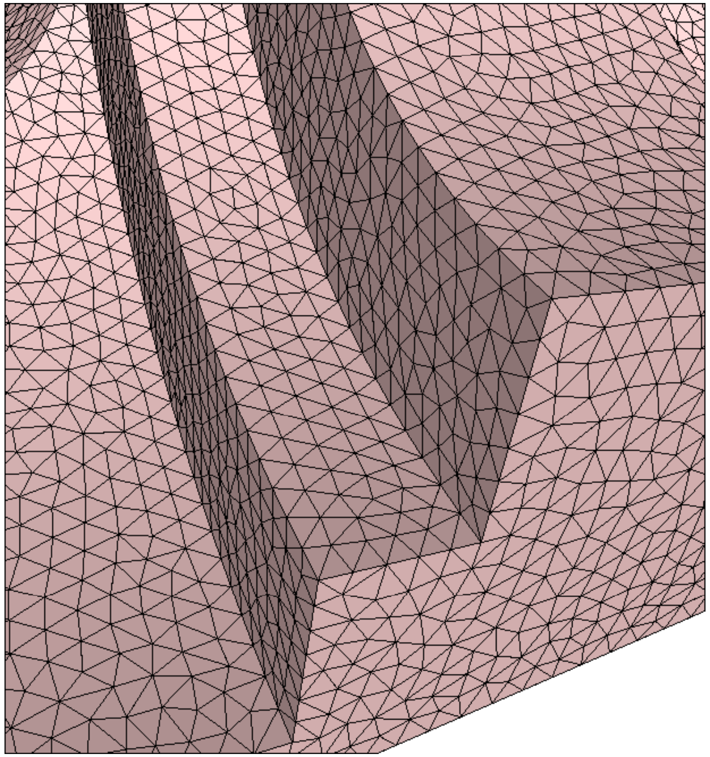
Feature backbones:

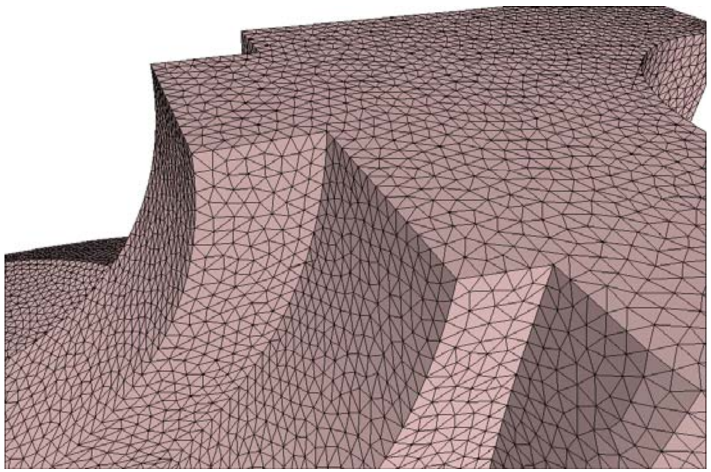
- 1D error diffusion
- arc-length parameterization of backbones
- 1D WCVD

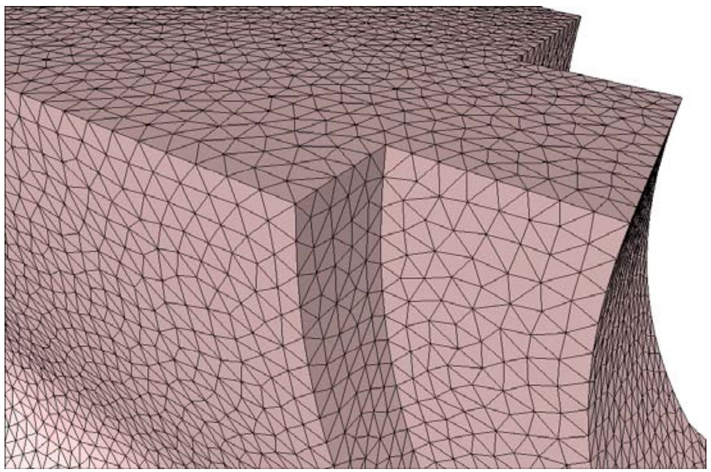


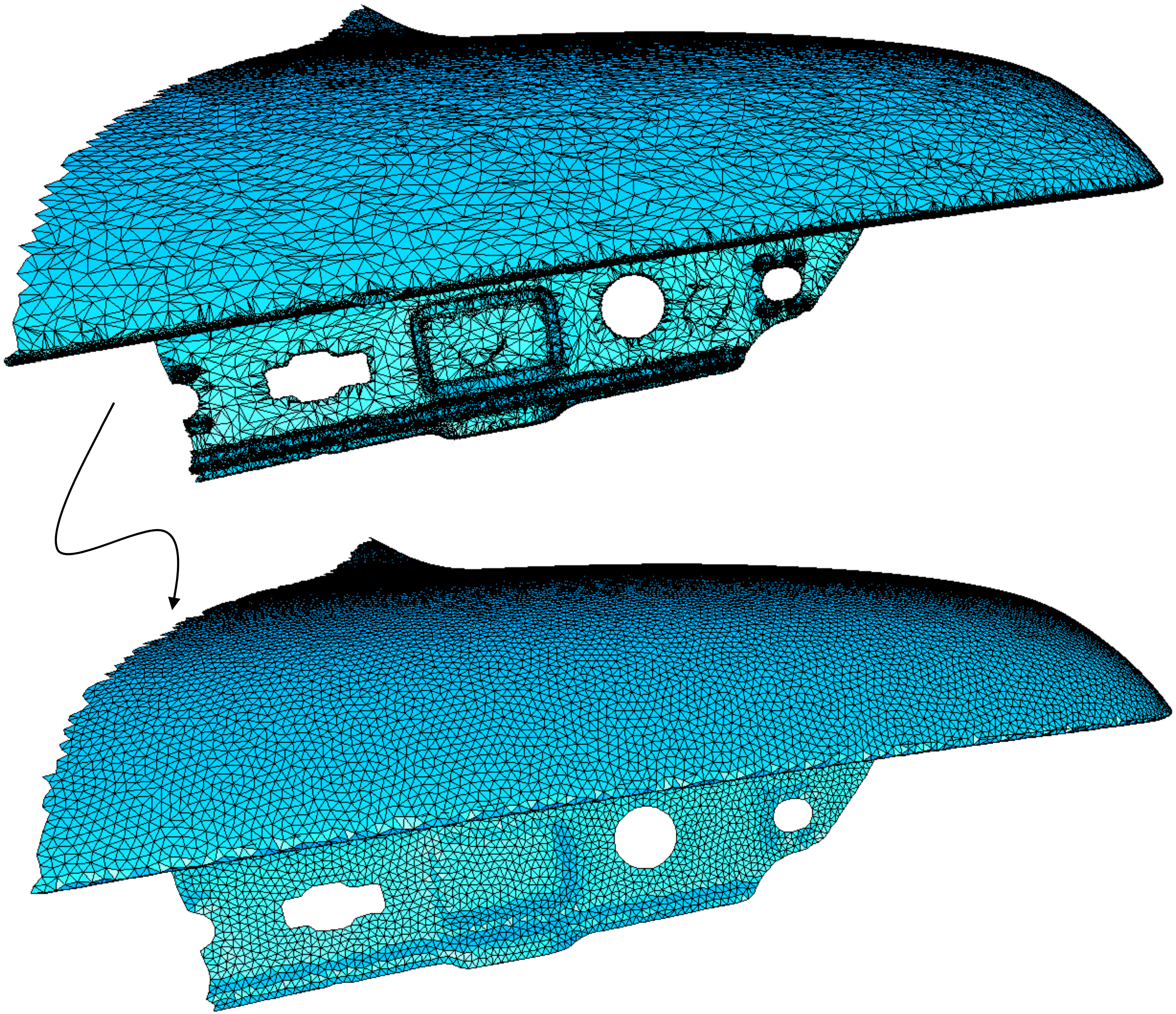


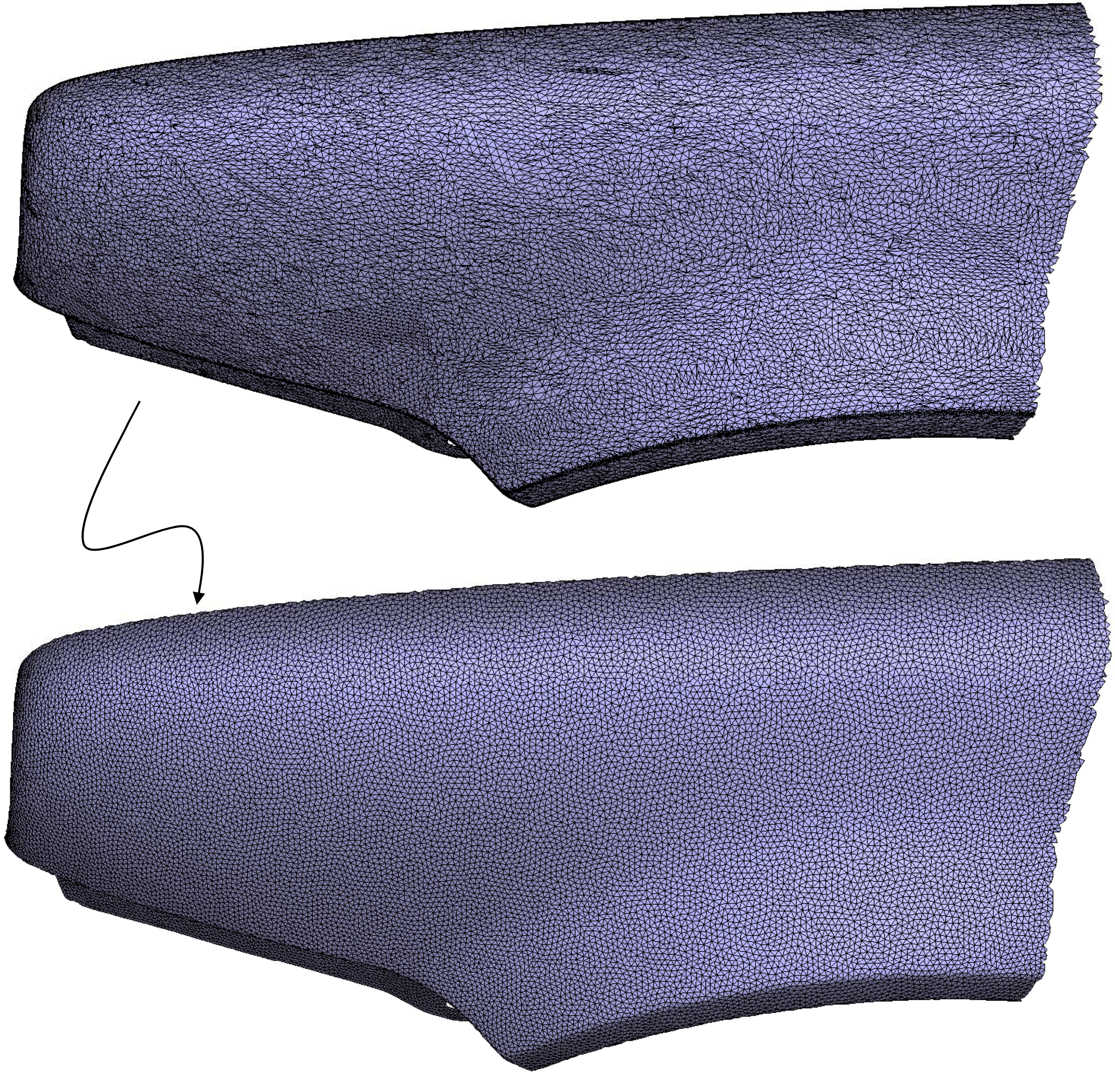






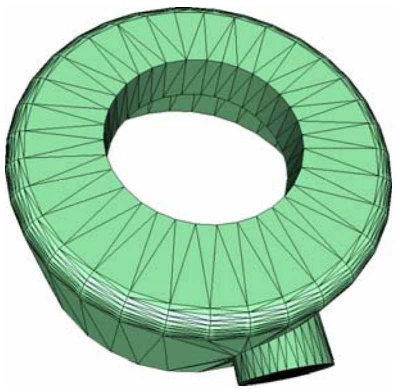




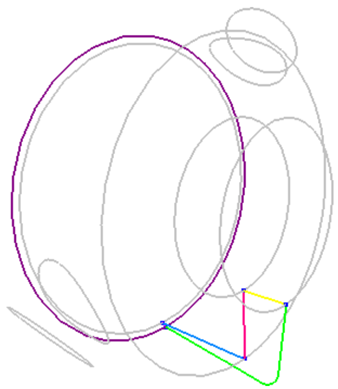


Genus>0 model

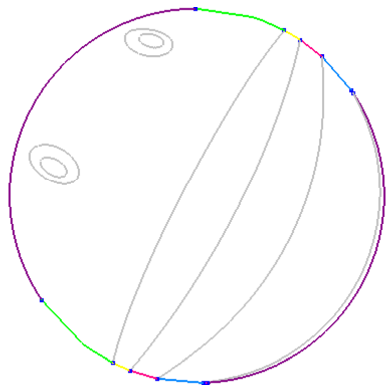
- cutting -> cut graph
- add cut graph to feature skeleton
  -> twin backbones associated pairwise
- synchronize sampling along twin backbones to guarantee stitching



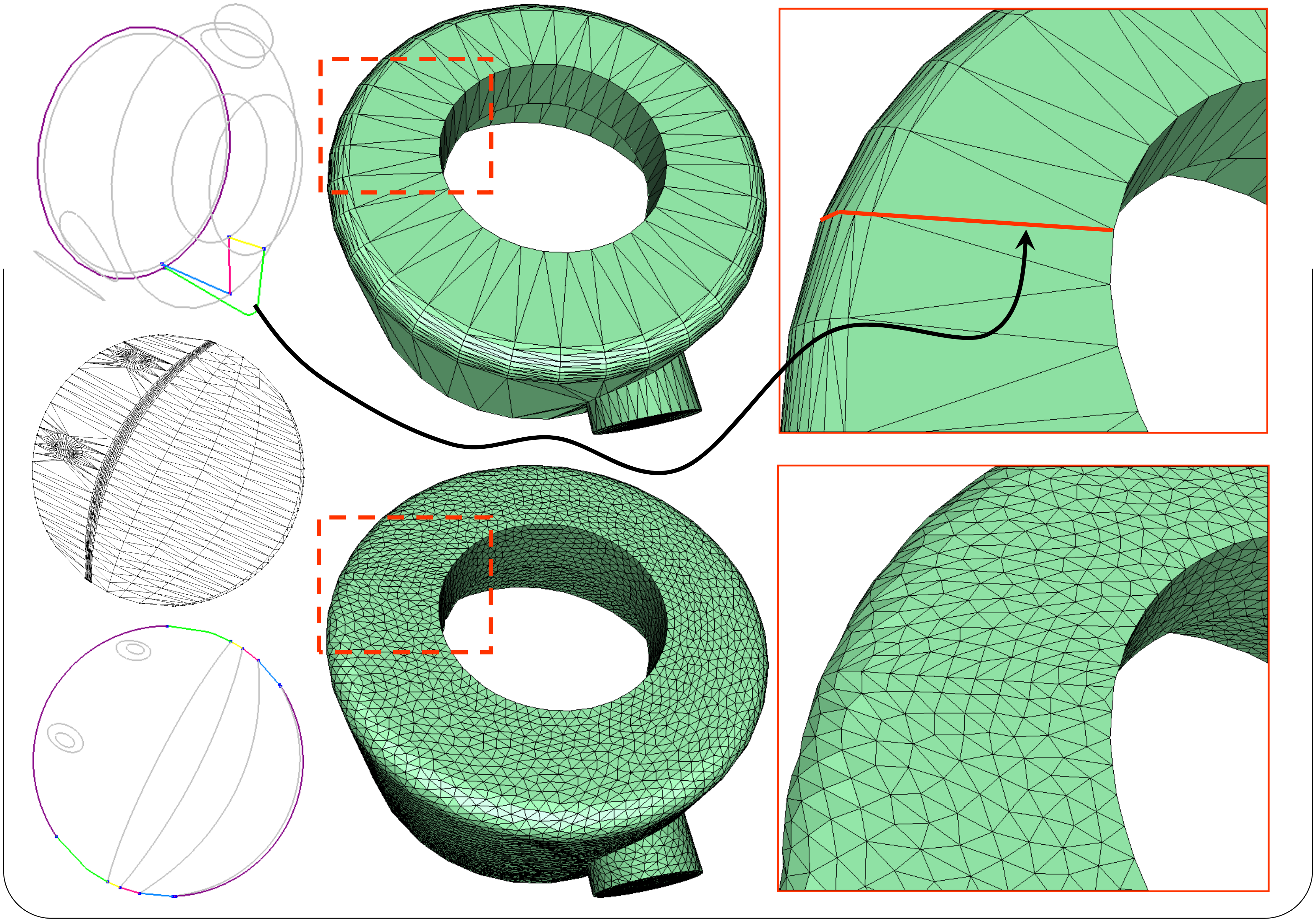
genus 1



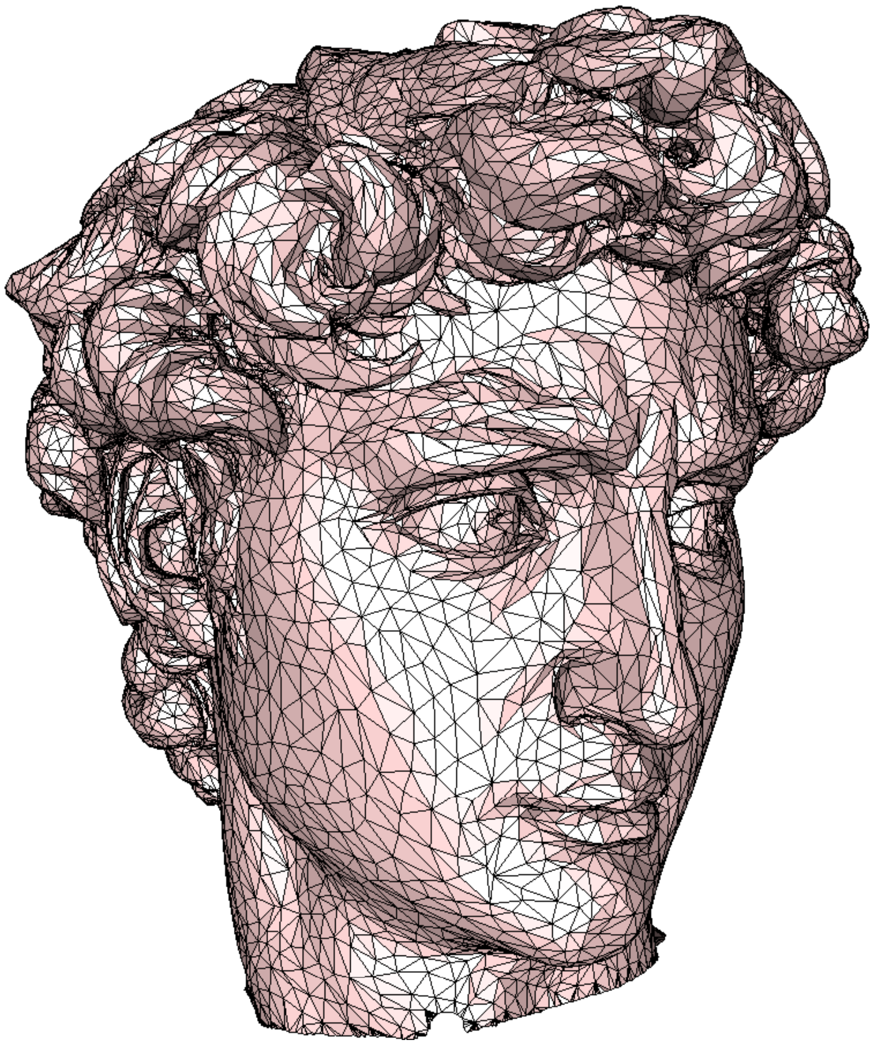
cut graph

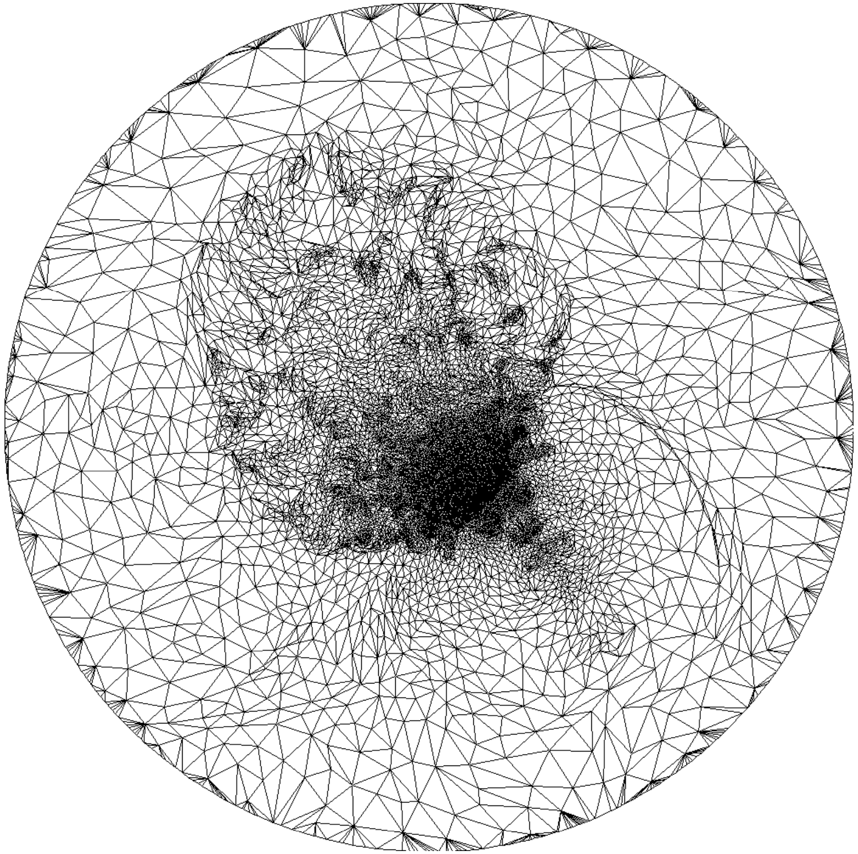


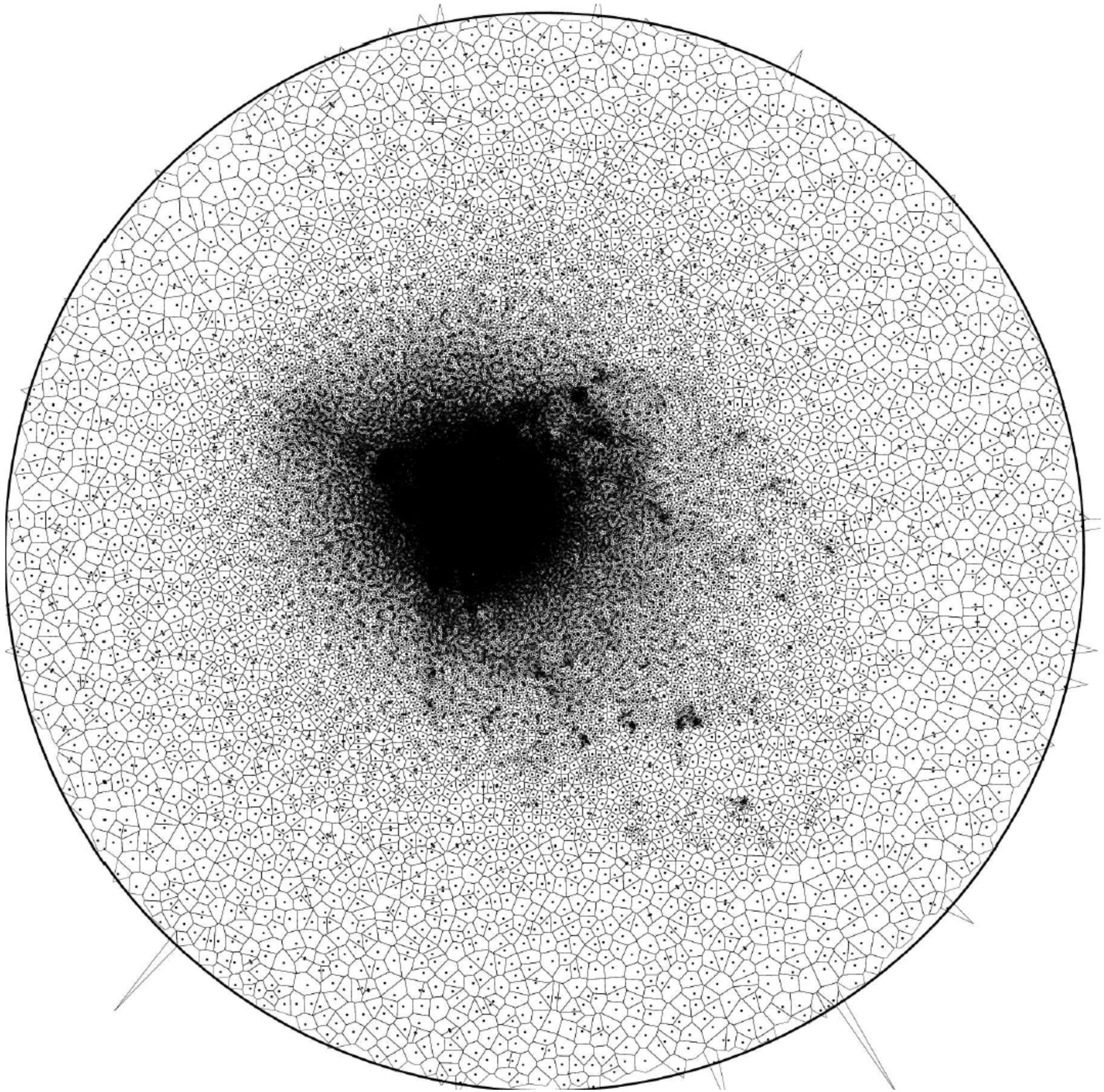
parameter space



David model [Digital Michelangelo]

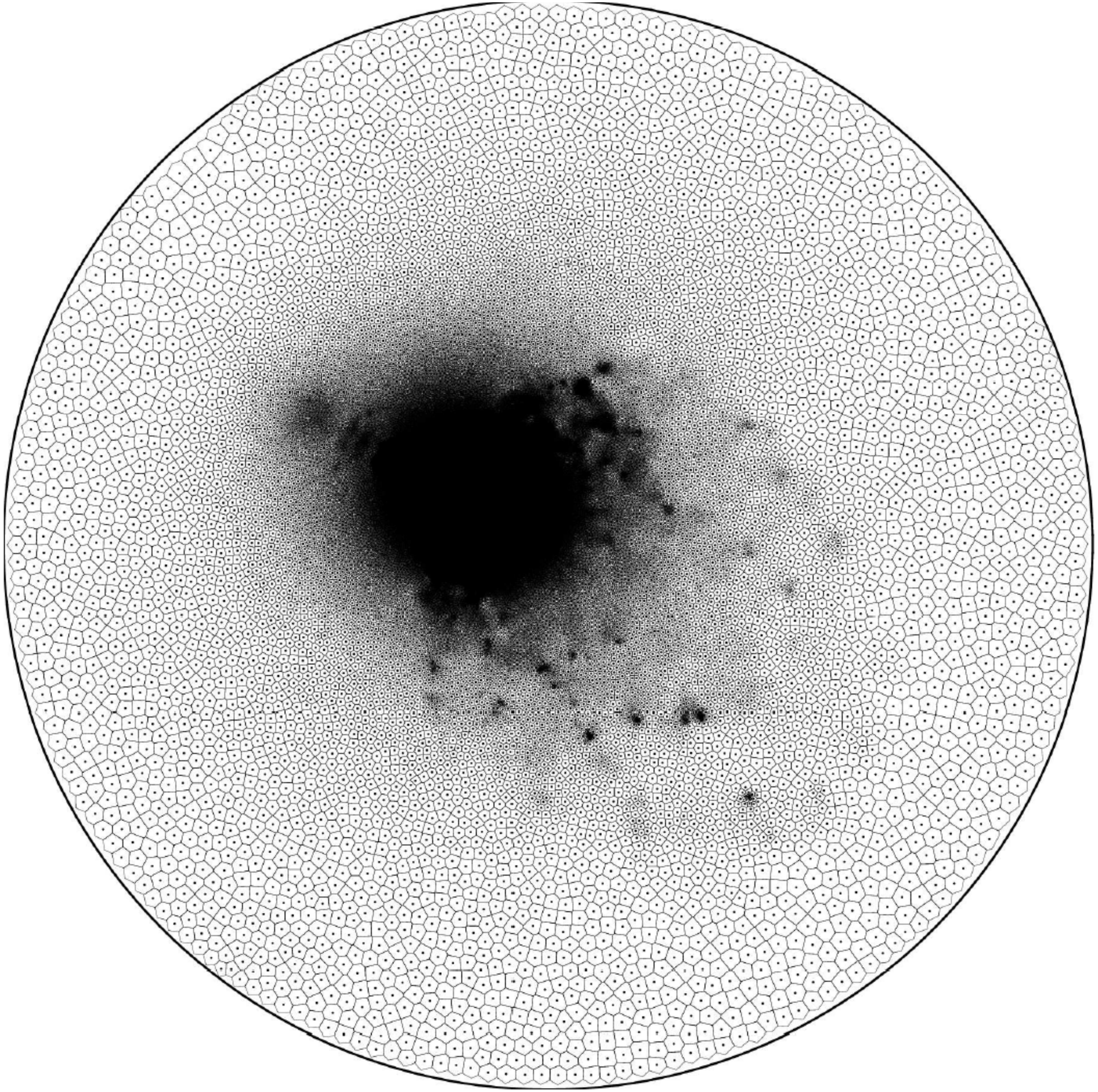


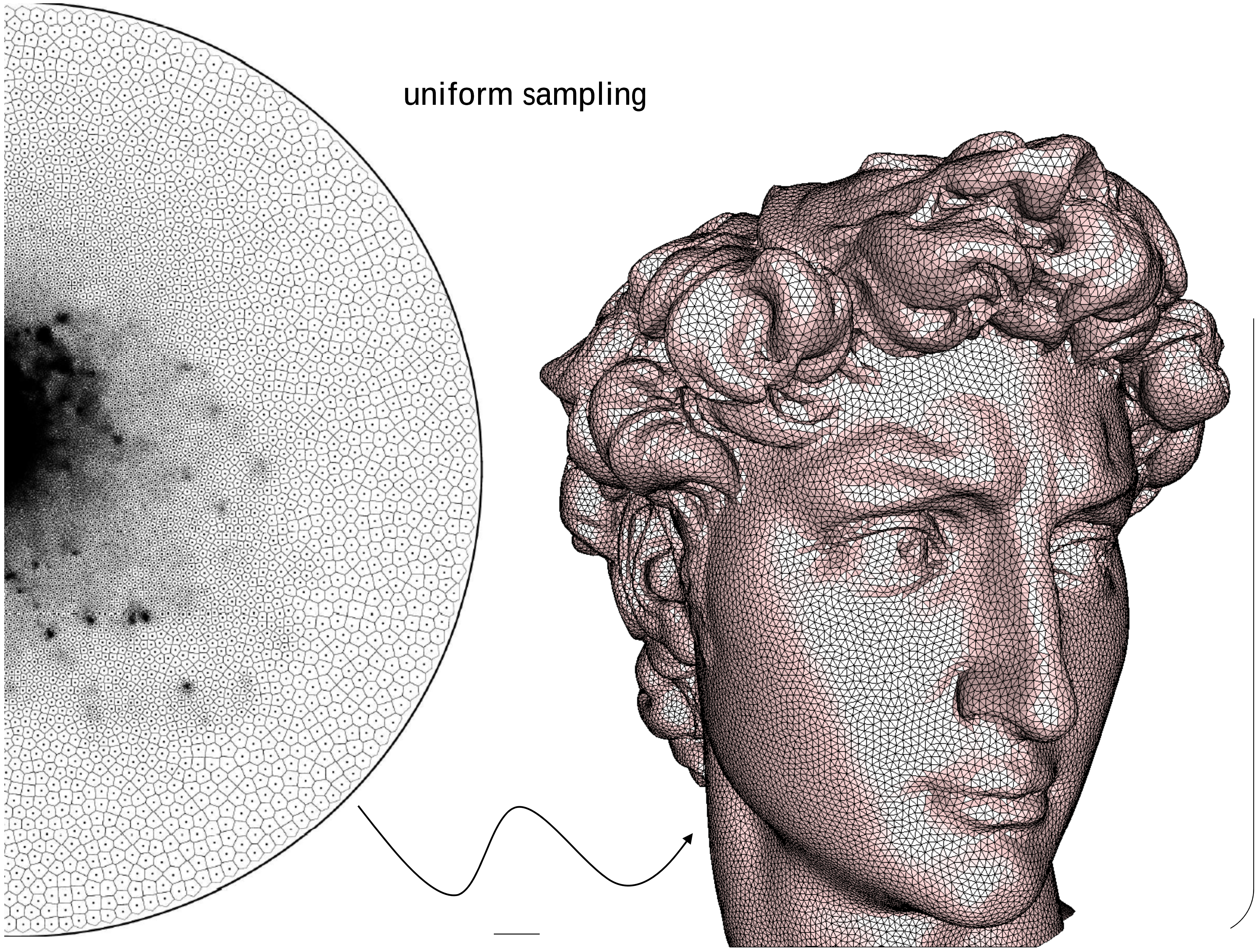


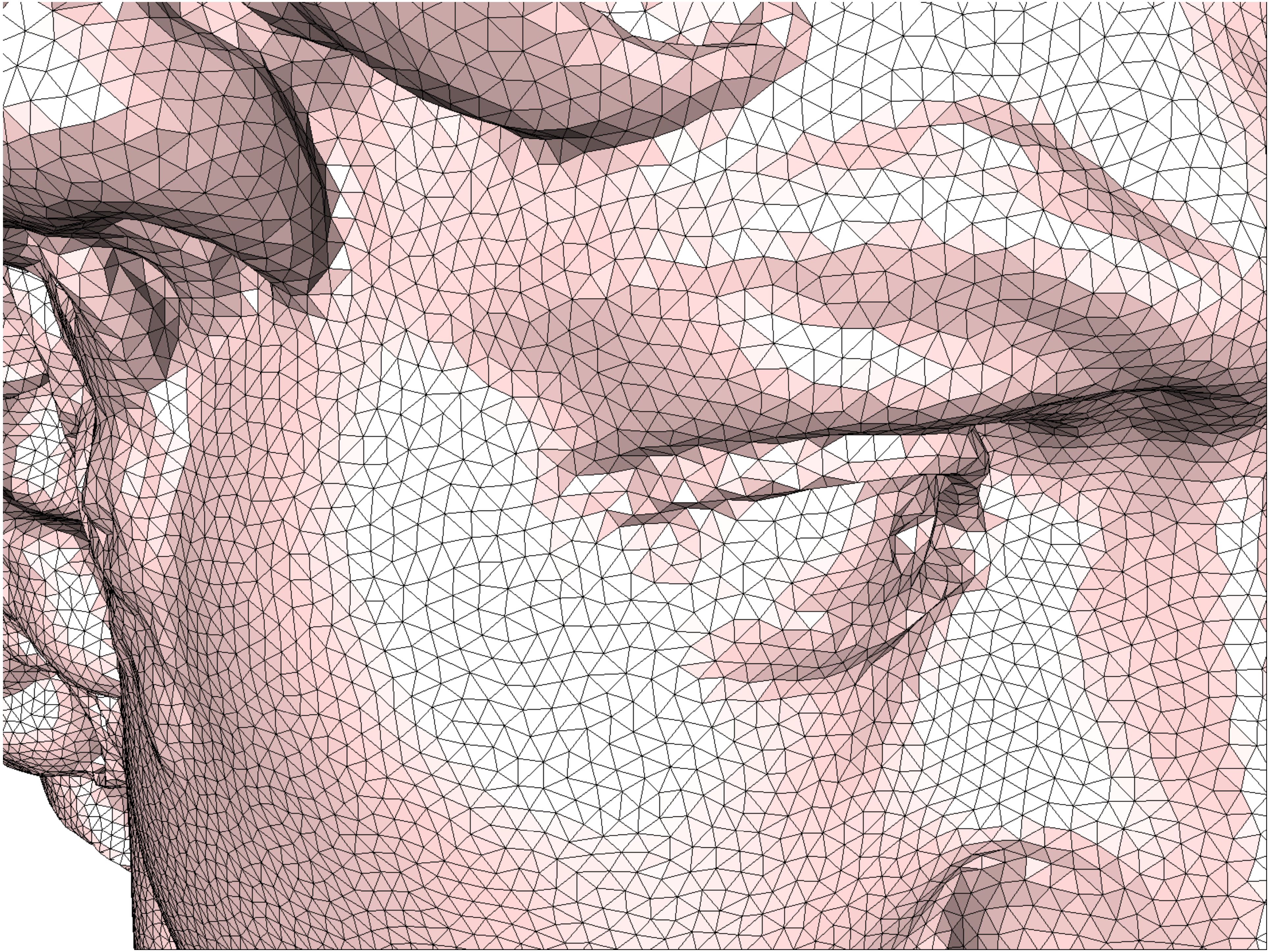


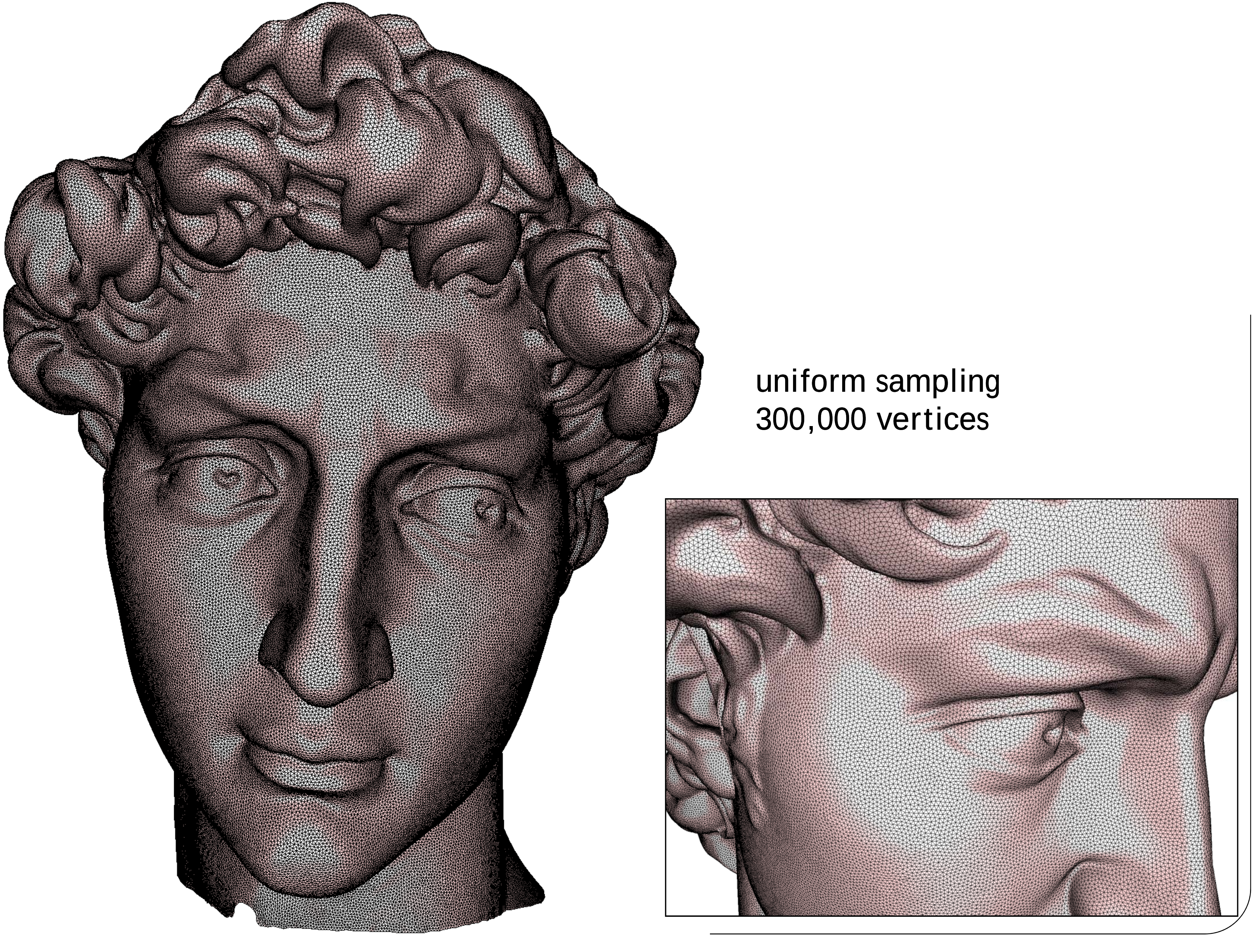
Sample repartition
by error diffusion
(50,000 vertices)

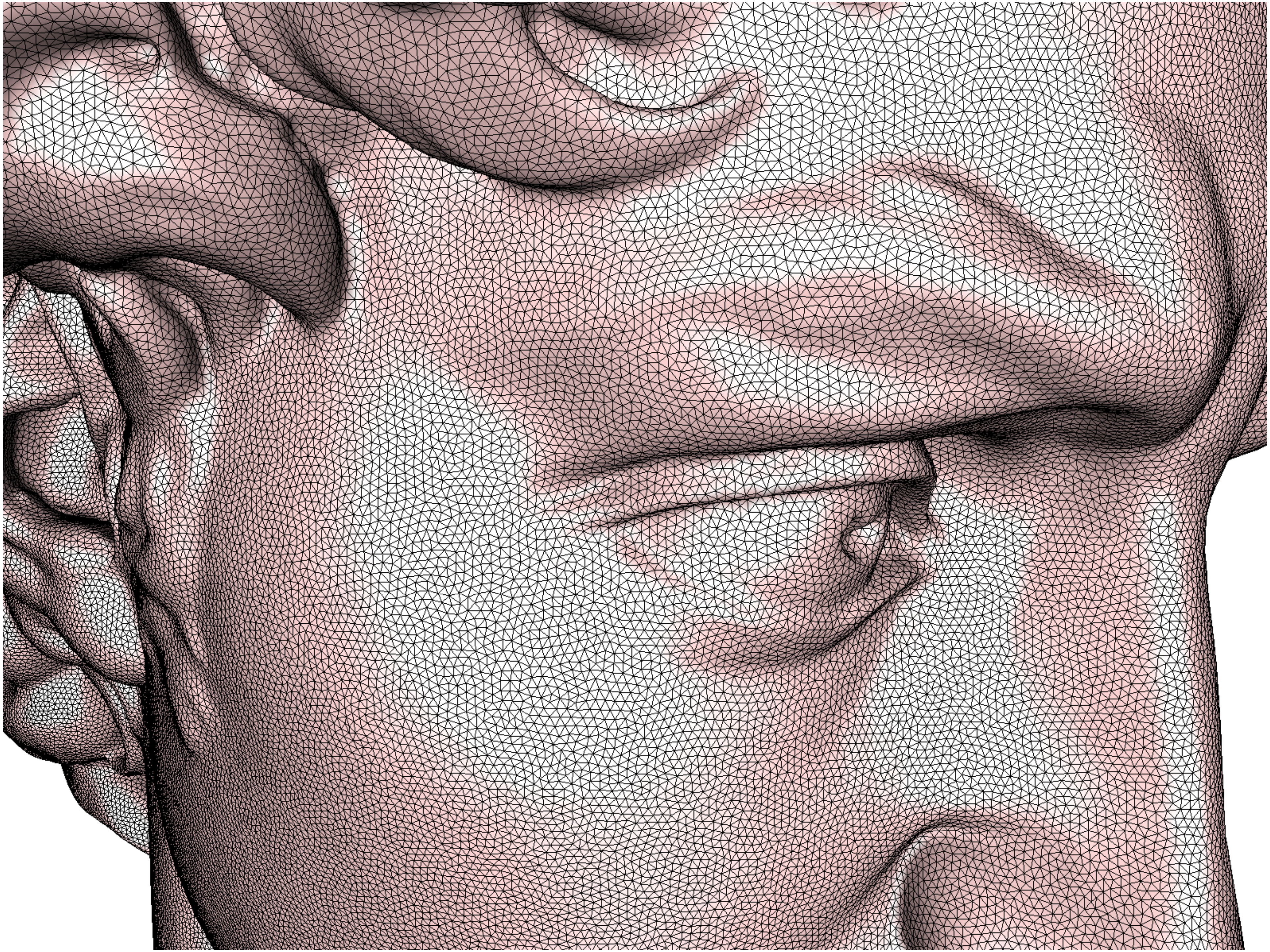
Sample placement by WCVD



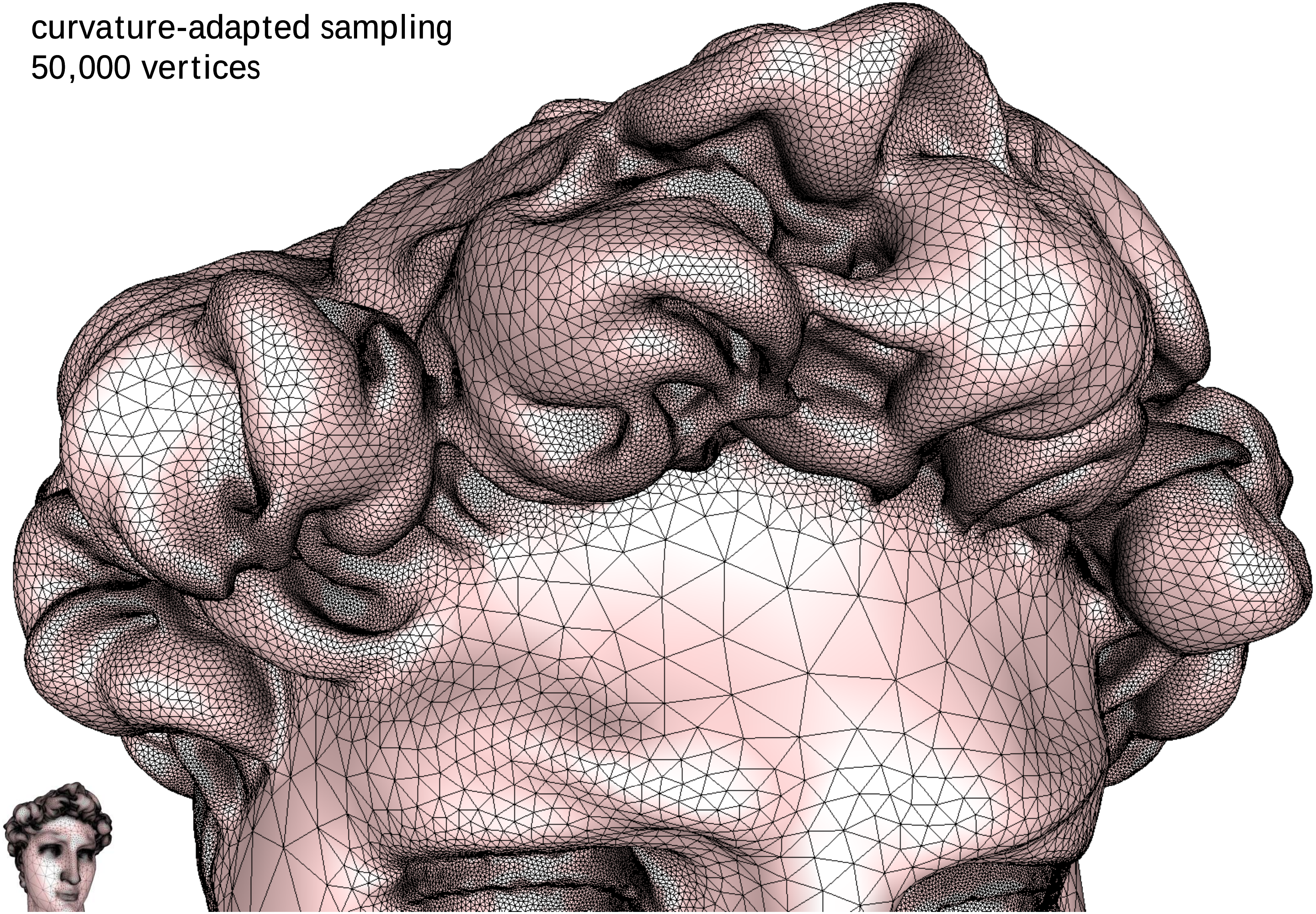

uniform sampling

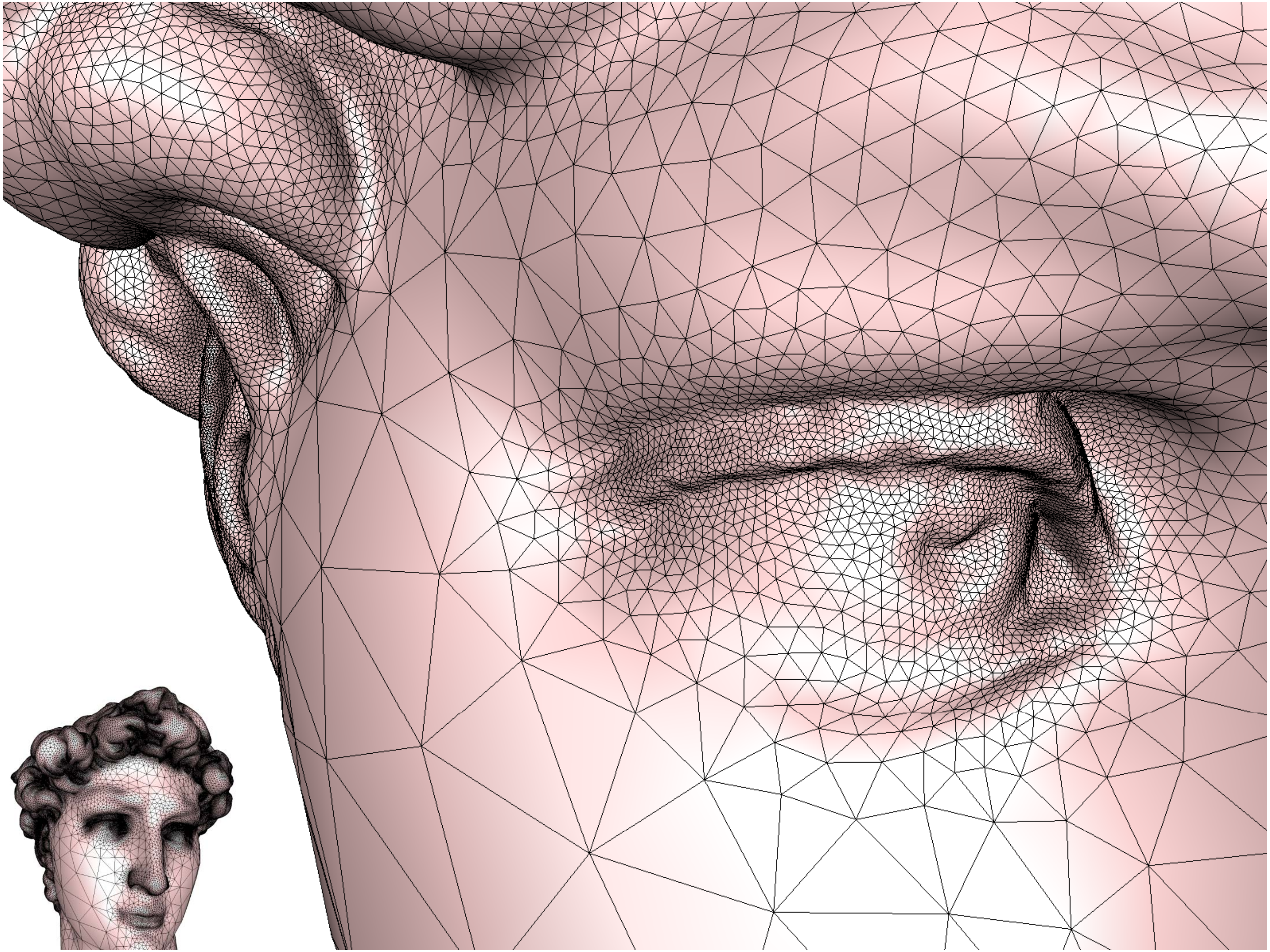


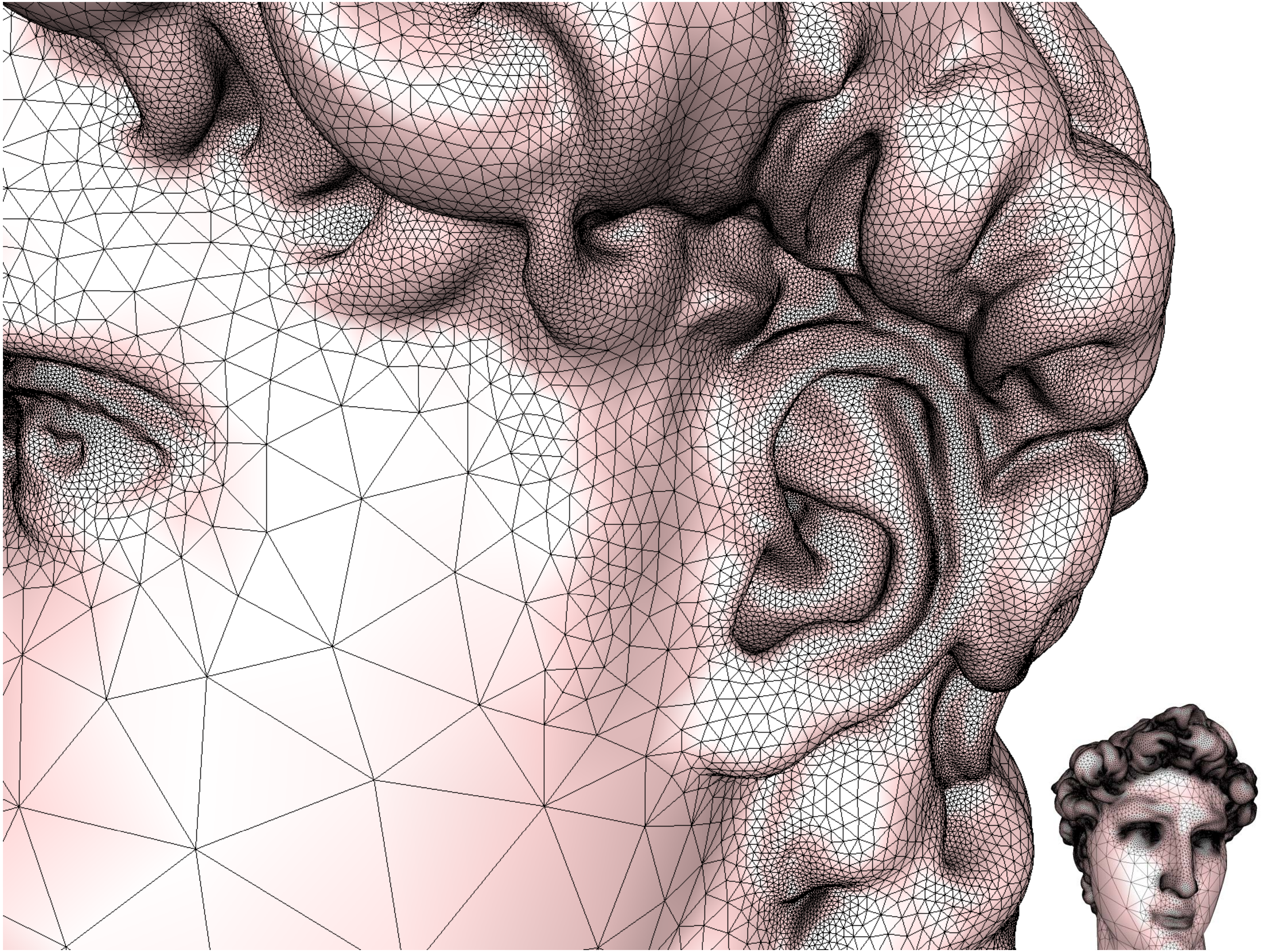

uniform sampling
300,000 vertices



curvature-adapted sampling
50,000 vertices







Motivation
Previous work
Contributions
Algorithm
Results
Limitations
Conclusions
Future Work

Limitations

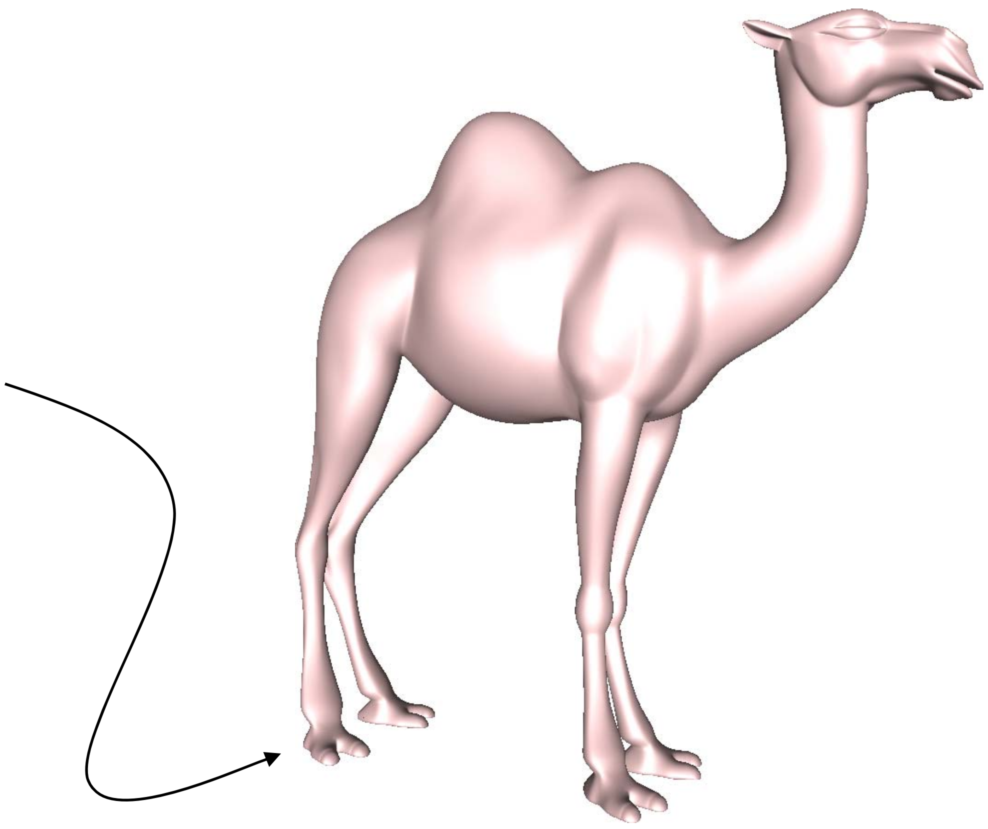
- Parameterization
  - still some numerical issues for huge models
  - quality of sampling is *very dependent* on the quality of the parameterization

- 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

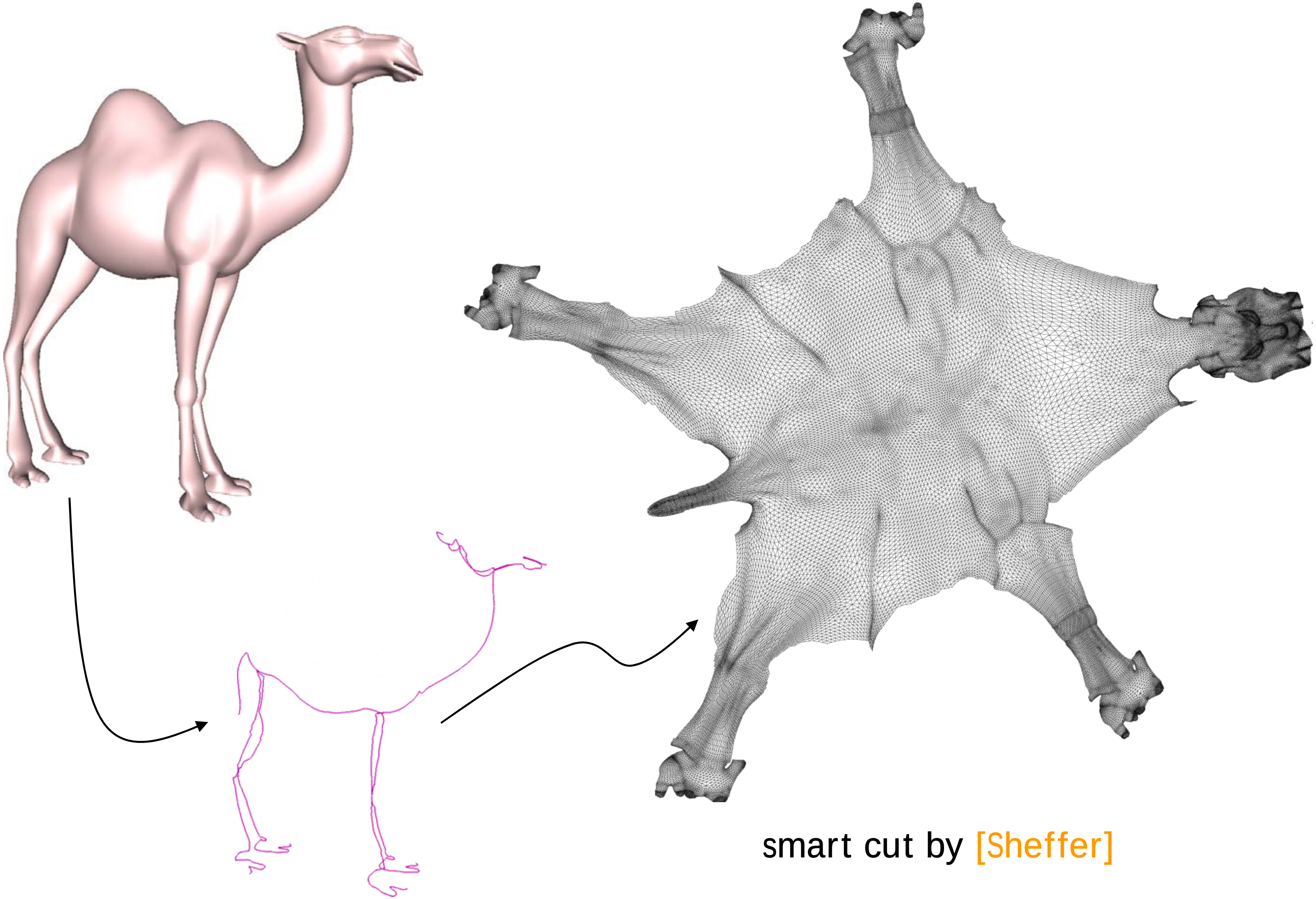
The Camel

- Closed
- Genus 0
- Sock-like shapes



how to cut it? ->

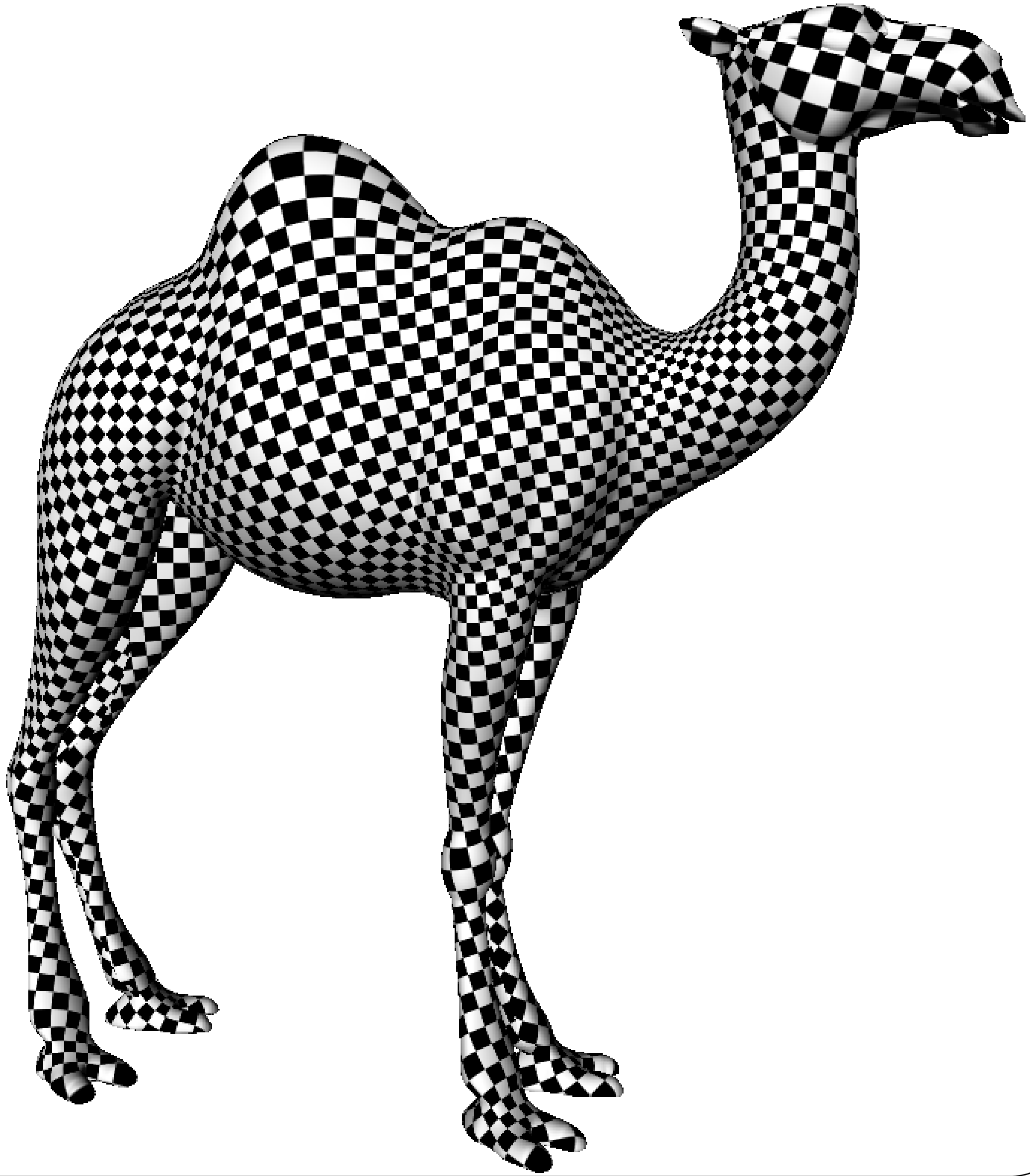
Camel pelting



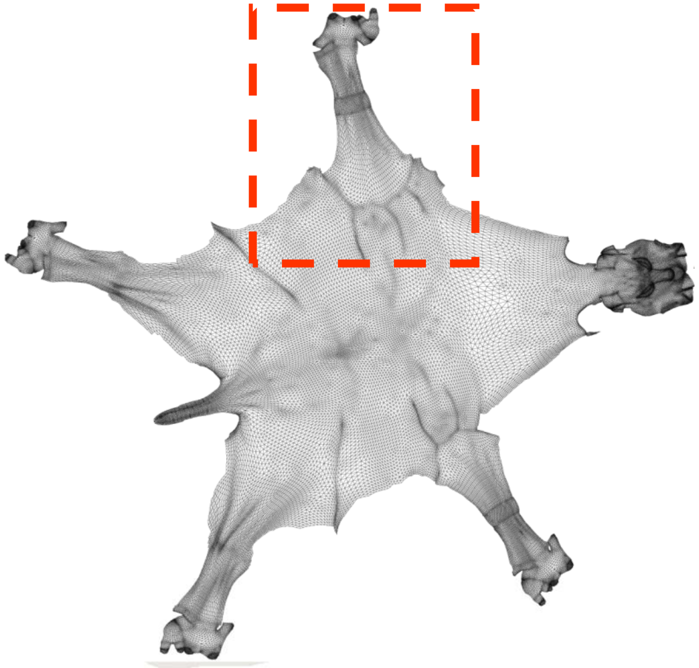
smart cut by [Sheffer]

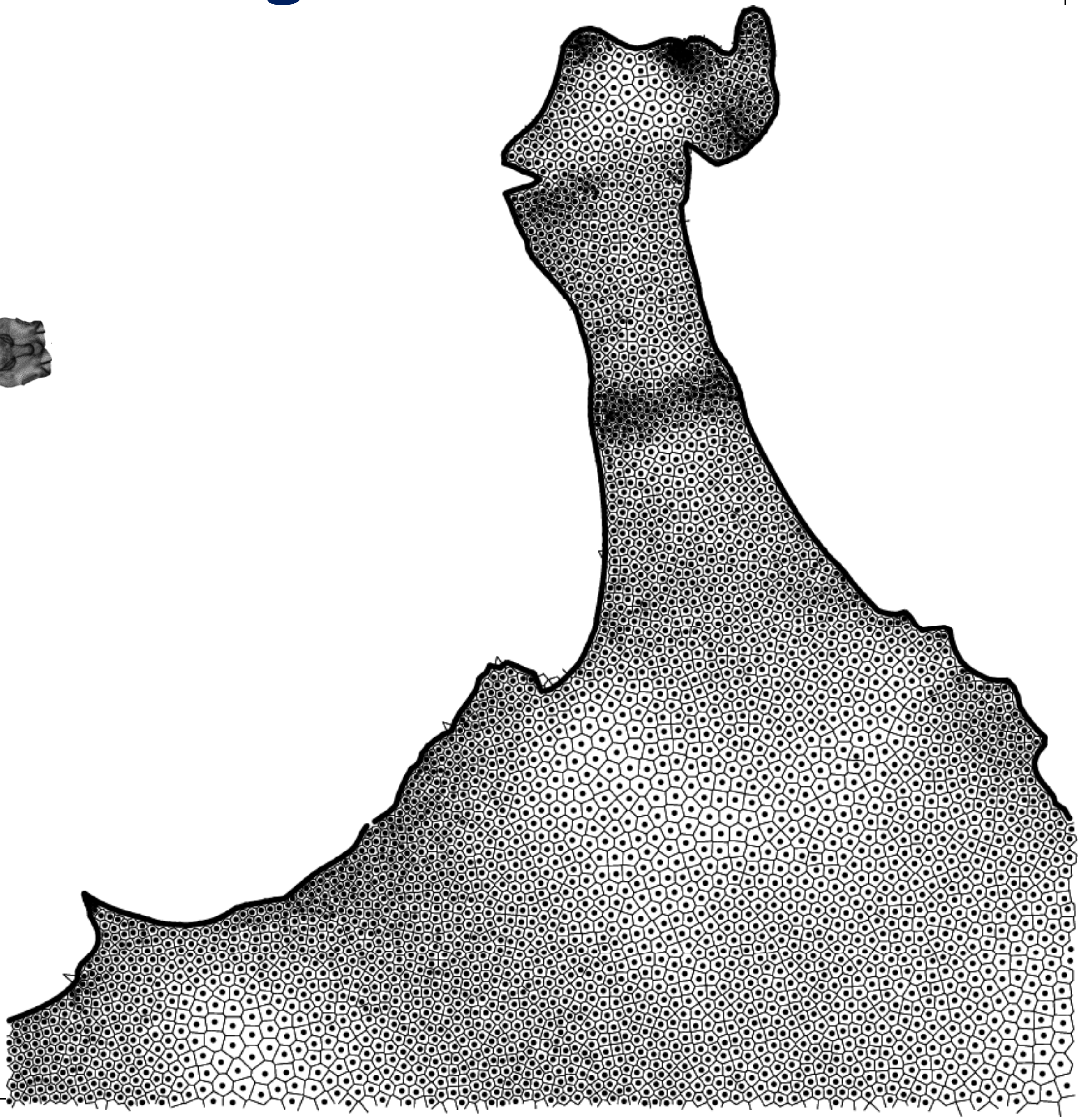
Smart pelting & Parameterization

[Sheffer]

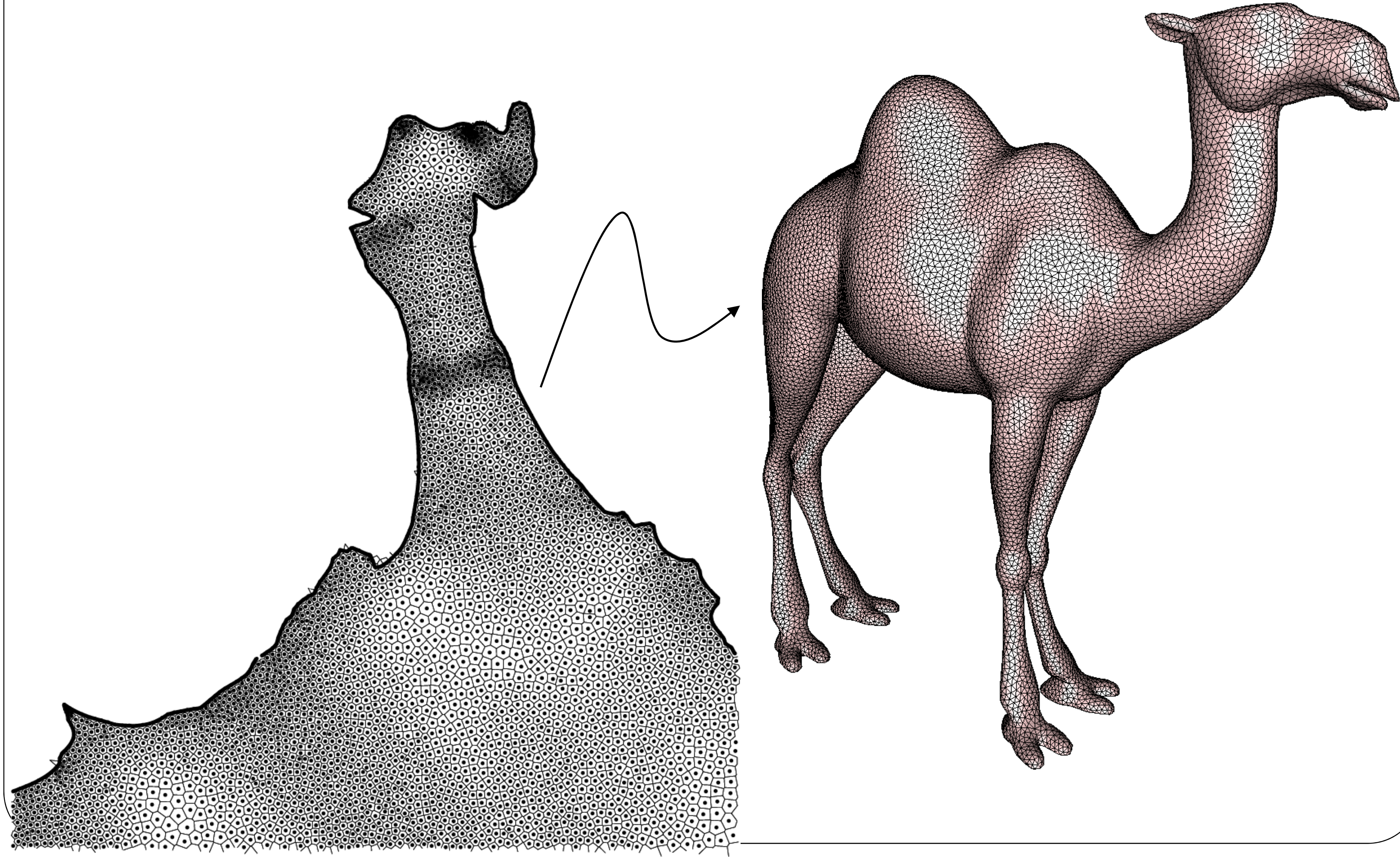


Uniform remeshing

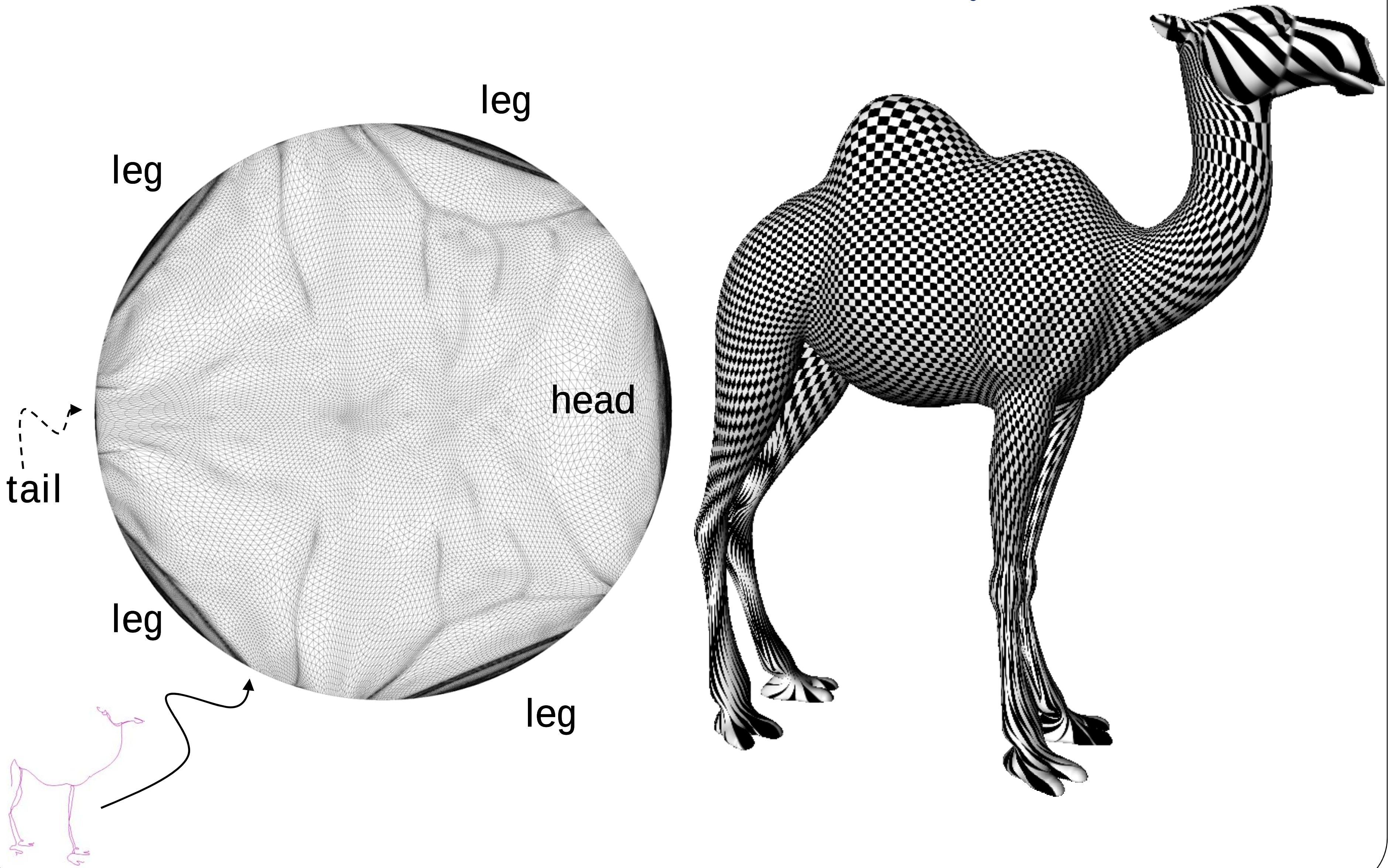




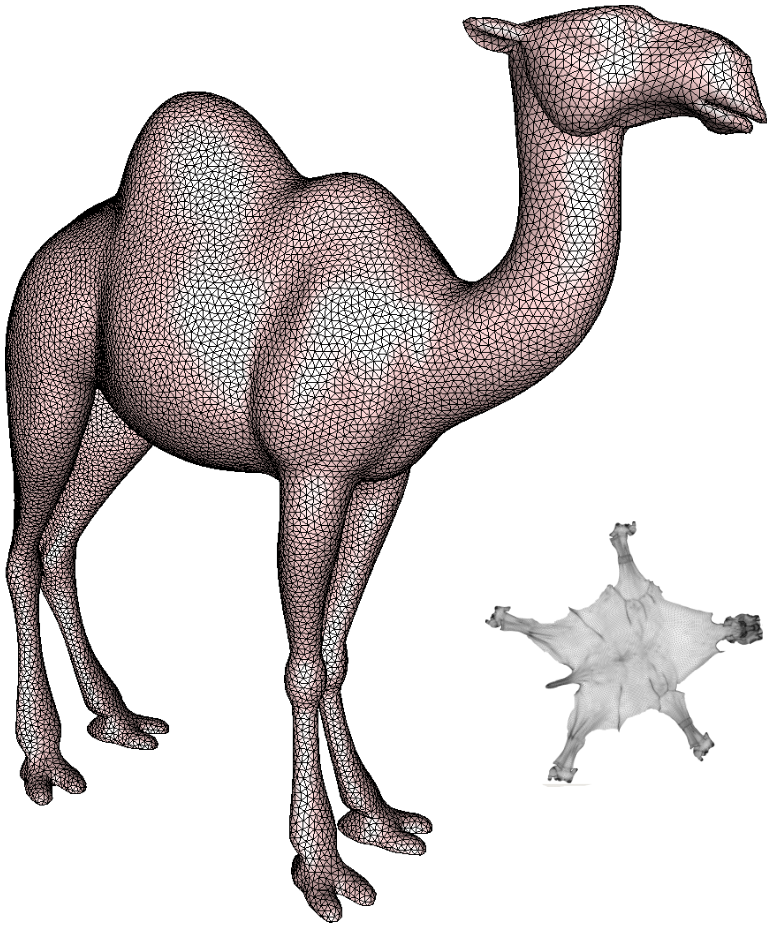
Uniform remeshing

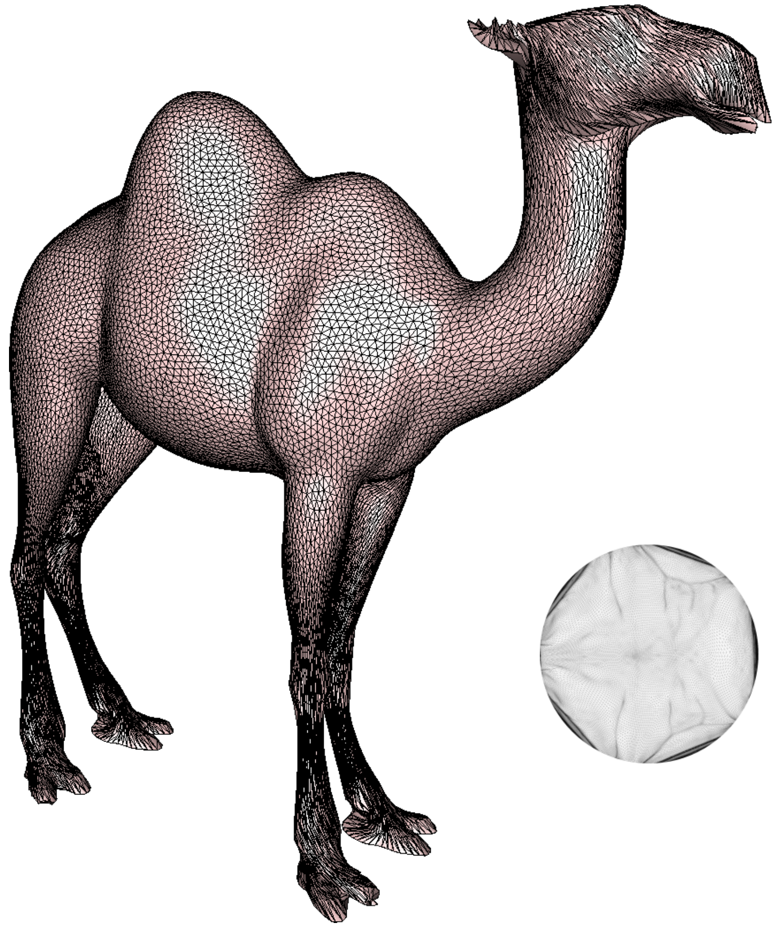


Same cut, fixed boundary ?

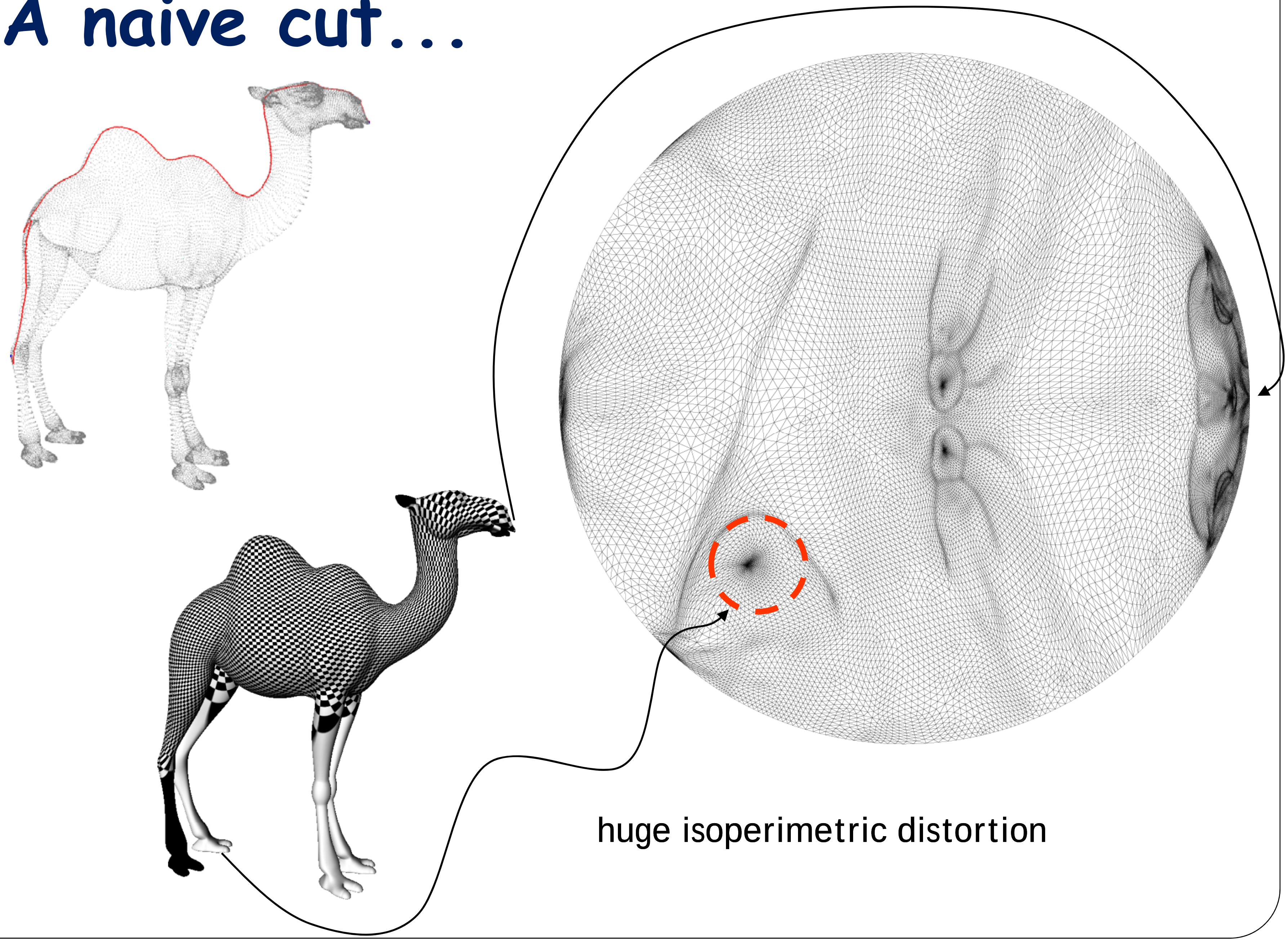


Remeshing with Free vs fixed boundary

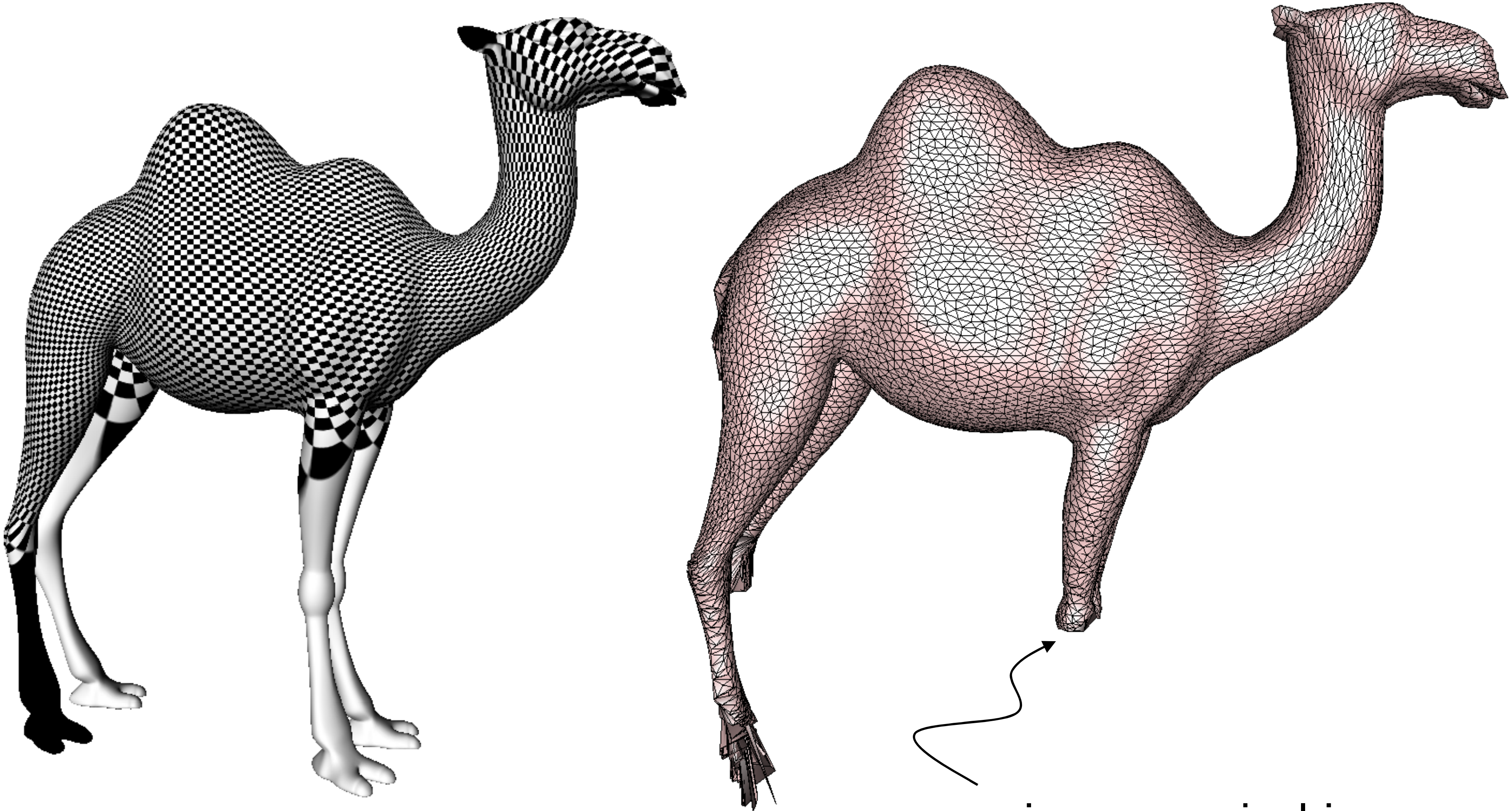




A naive cut...

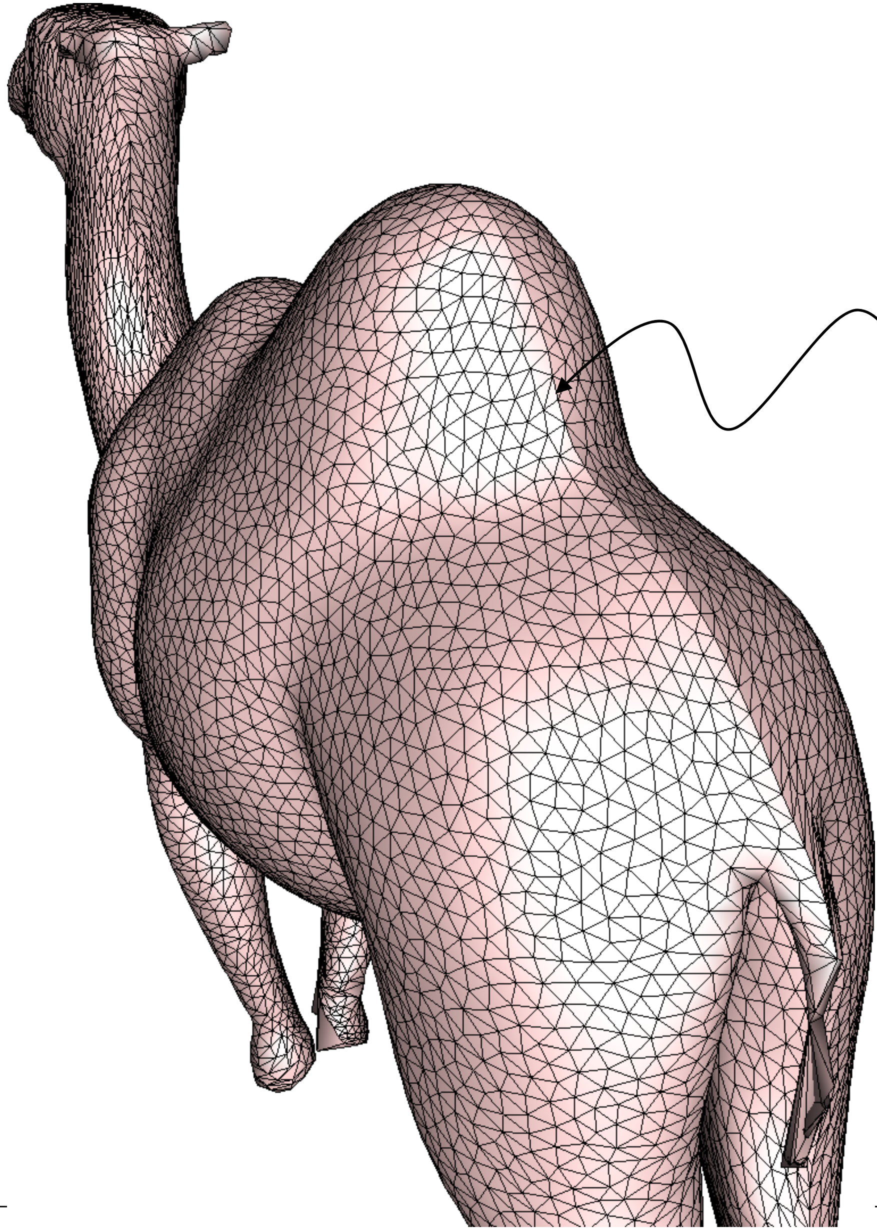


Numerical issues



serious numerical issues
with sock-like shapes

Moreover...



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

Conclusions

- Guarantee: vertex budget
- Delaunay triangulation → given the sampling, get the optimal planar triangulation
- Centroidal Voronoi diagram → improve the sampling to capture the essence of isotropic sampling
- Density → reflects area distortion and other geometric properties (e.g. cuvatures)
- Handle features

Anisotropic Remeshing