# (Re)Meshing

Nov. 17, 19, 2009

# Review – Surface Mapping

- A fundamental geometry problem with long history
- A ubiquitous tool for many applications in broad areas
  - This class → Remeshing
- Study the mapping problem from aspects of
  - Spring network
  - Distortion measurement

# Remeshing Overview

- A fundamental geometry problem with long history
- A ubiquitous tool for many applications in broad areas
  - This week → Remeshing
- Study the mapping problem from aspects of
  - Spring network
  - Distortion measurement

# Motivations

Digital Mesh Processing:

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

Control over:

- vertex density
- shape of elements
- etc.

# 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:

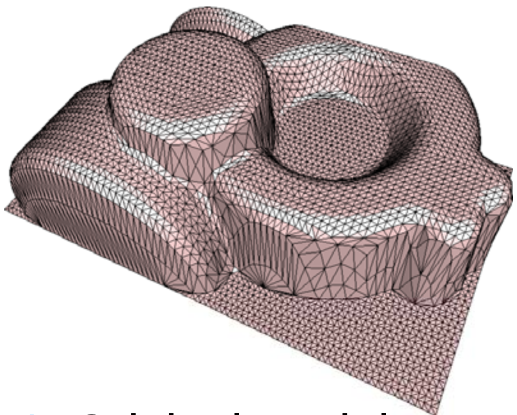
Geometric modeling

for effective processing and fast display

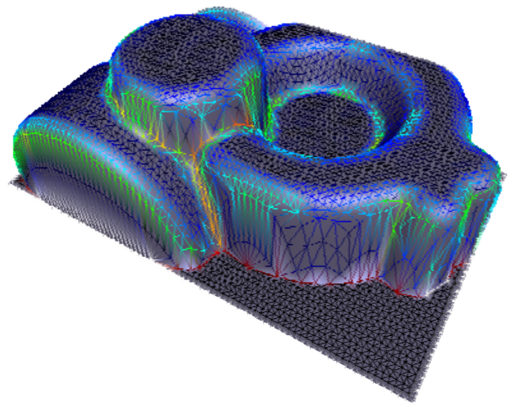
# Contributions

- New algorithm for *isotropic* surface remeshing
- Resampling: Error diffusion for sample *repartition*
- Optimization: Centroidal Voronoi tessellation for sample *placement*

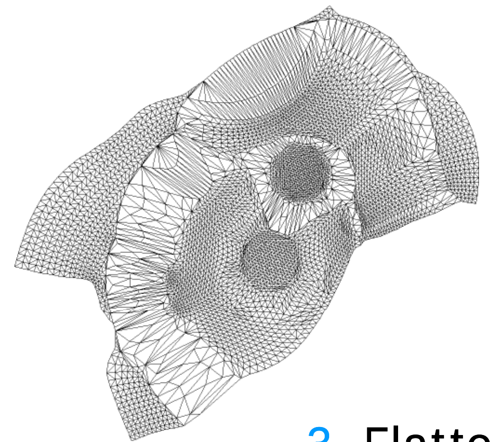
# Proposed technique



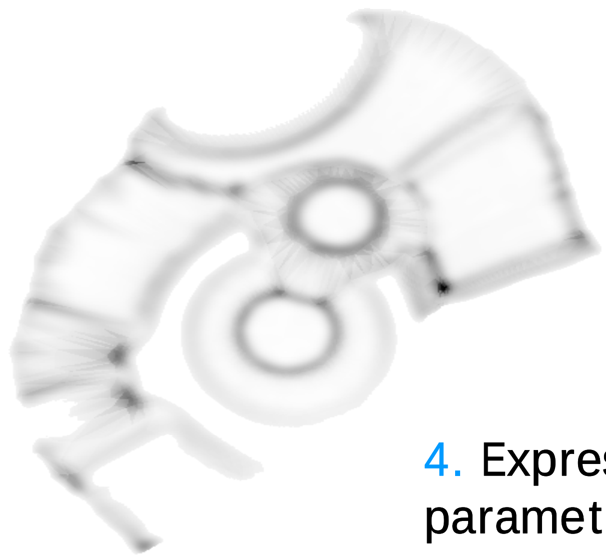
1. Original model



2. Measure density



3. Flatten it



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

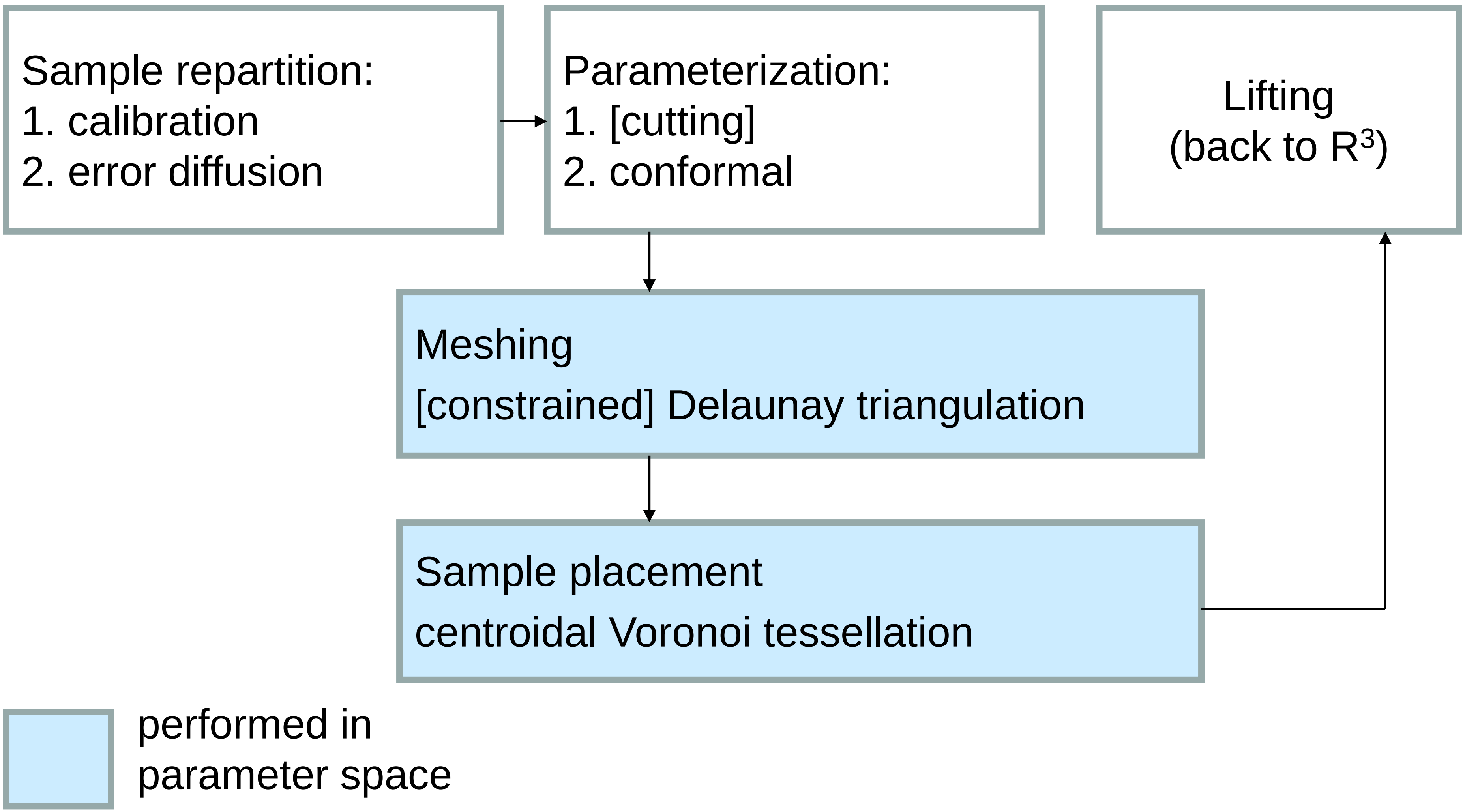
# More precisely...

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

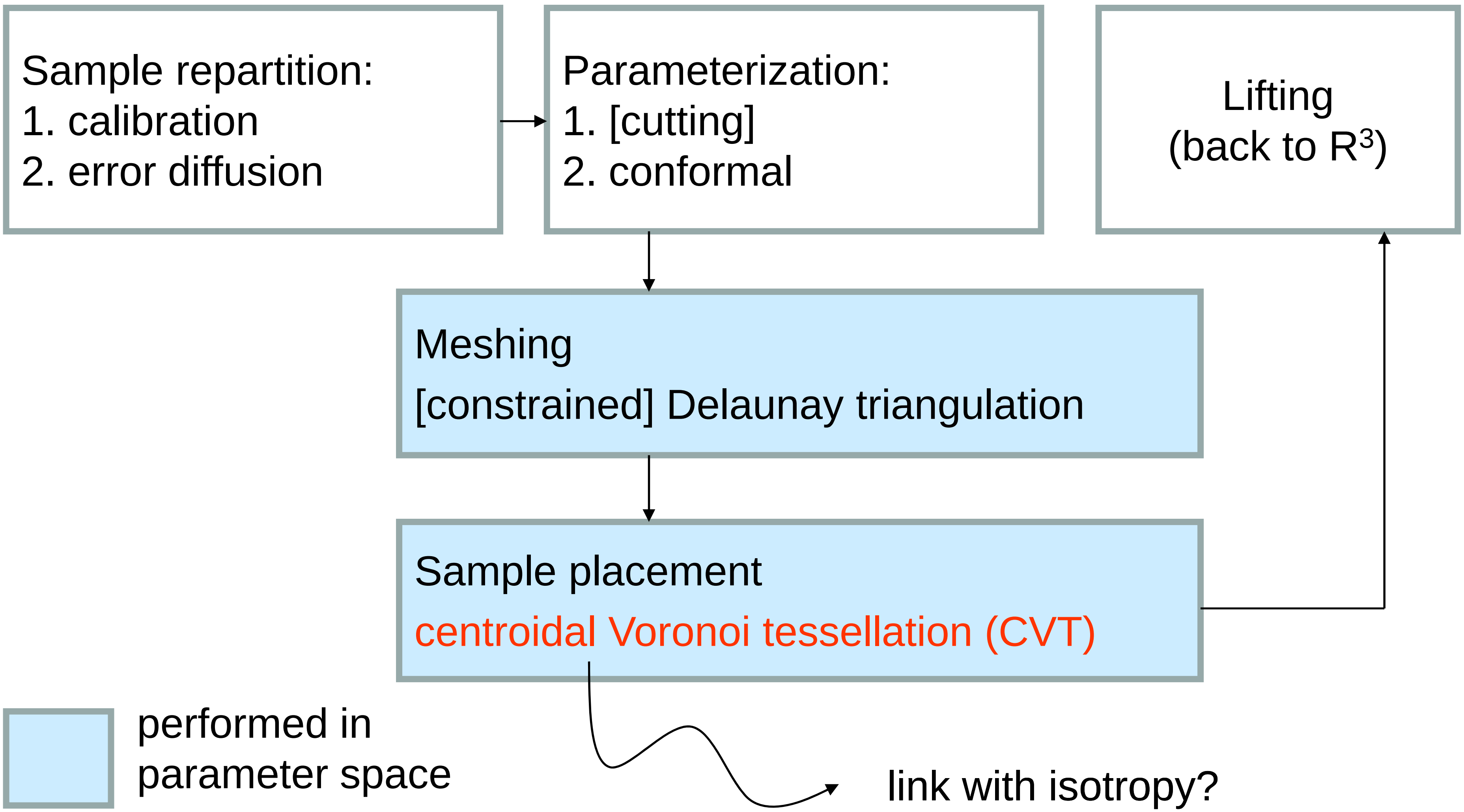
Solution

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

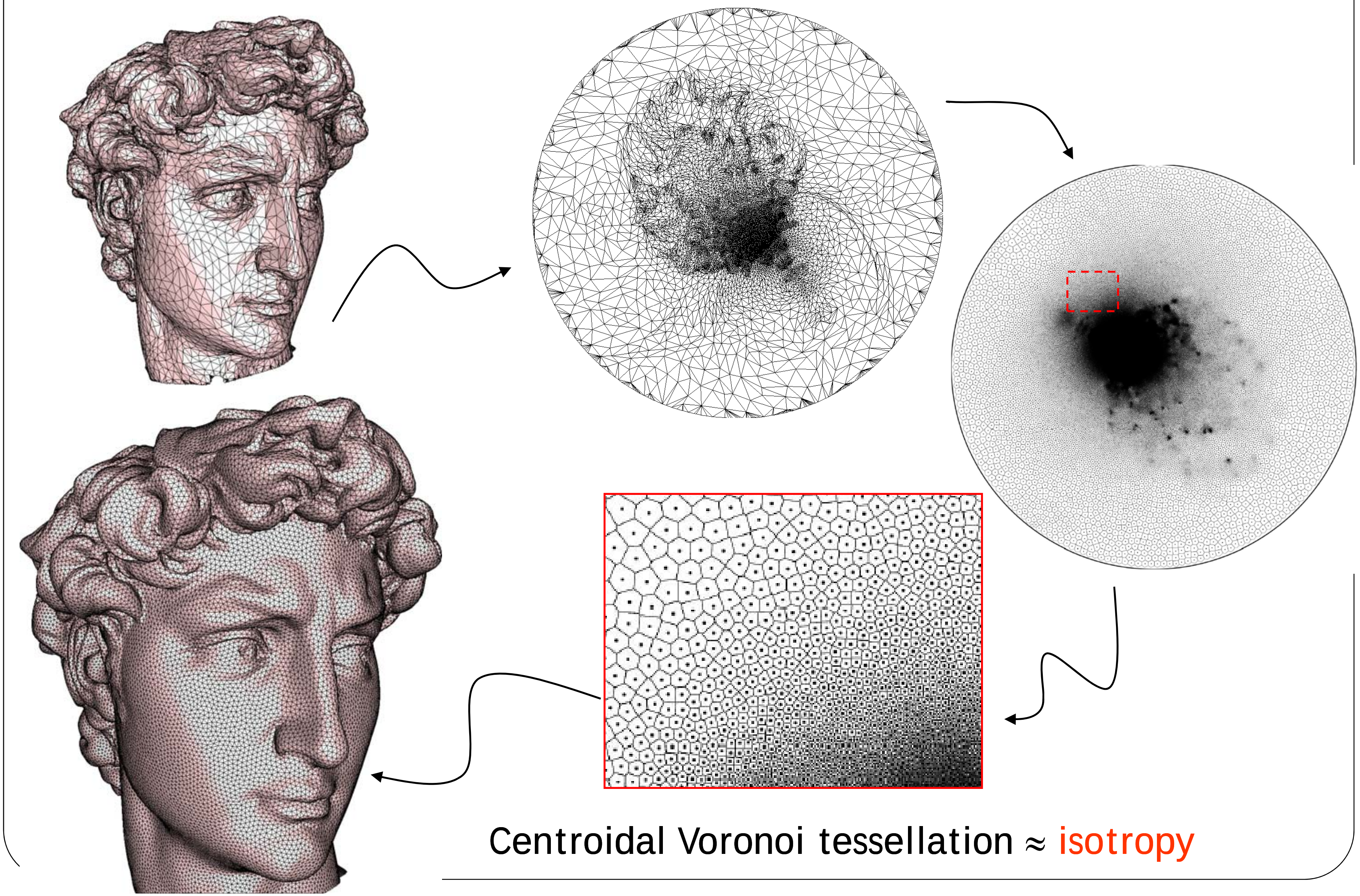
# Remeshing Pipeline



# Remeshing Pipeline



# Key idea



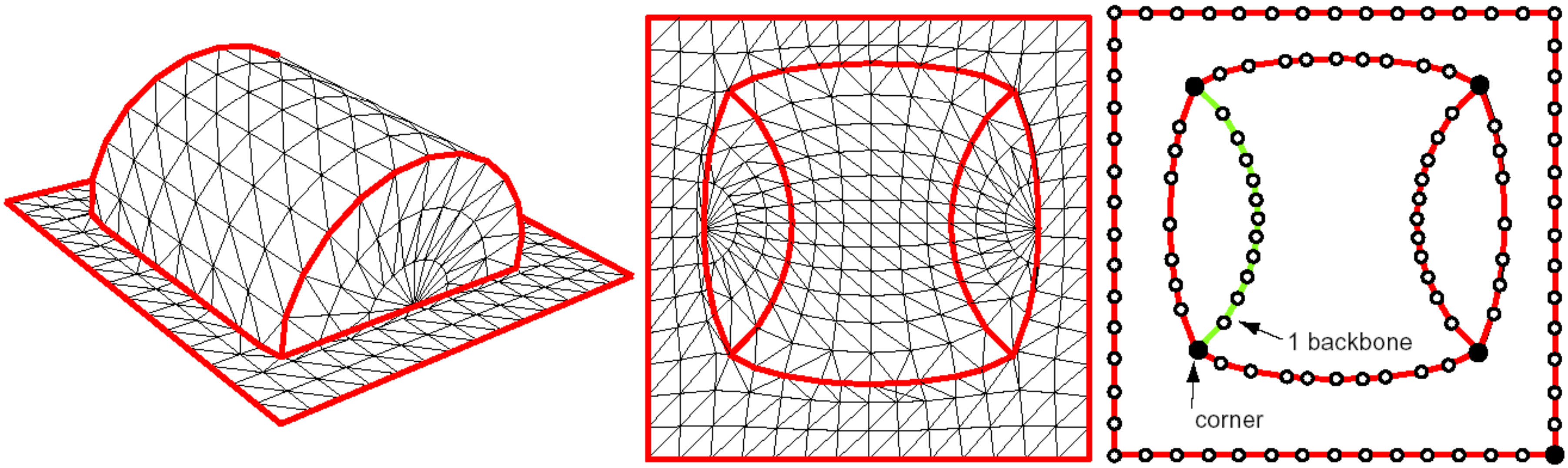
Centroidal Voronoi tessellation ≈ isotropy

# 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

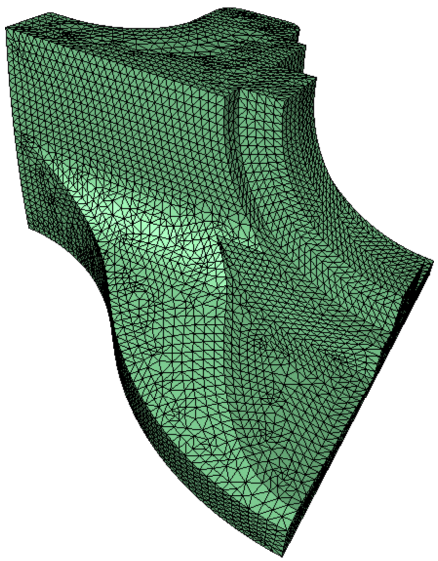


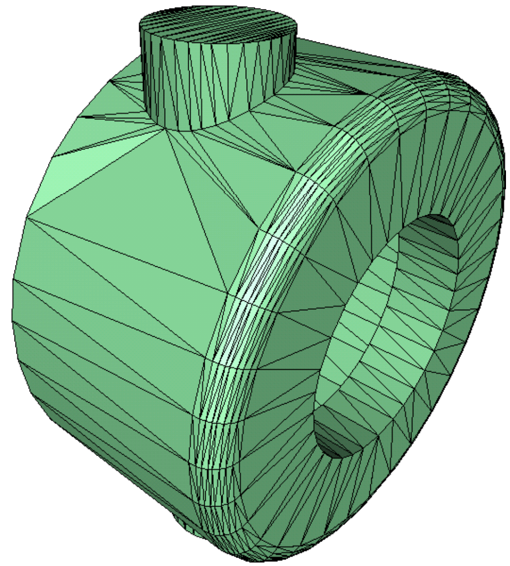

Original model    Parameterization and tagged edges    Feature skeleton

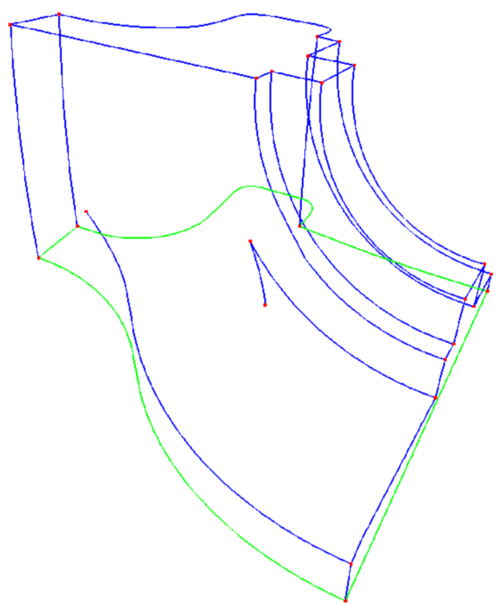
0-manifolds: corners
1-manifolds: backbone: feature edges chained together

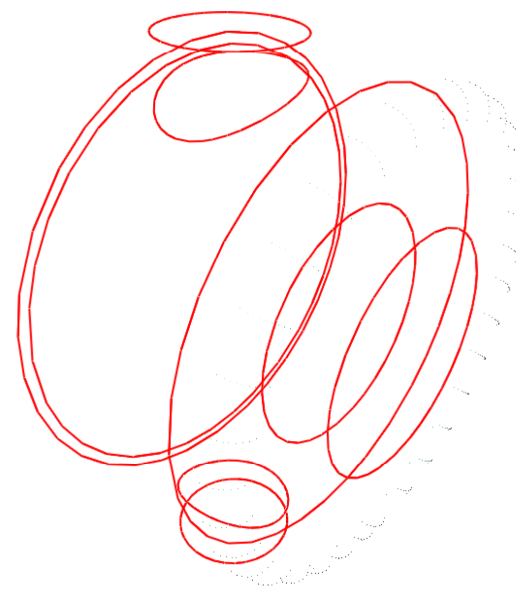
feature skeleton: corners + backbones

# More feature skeletons

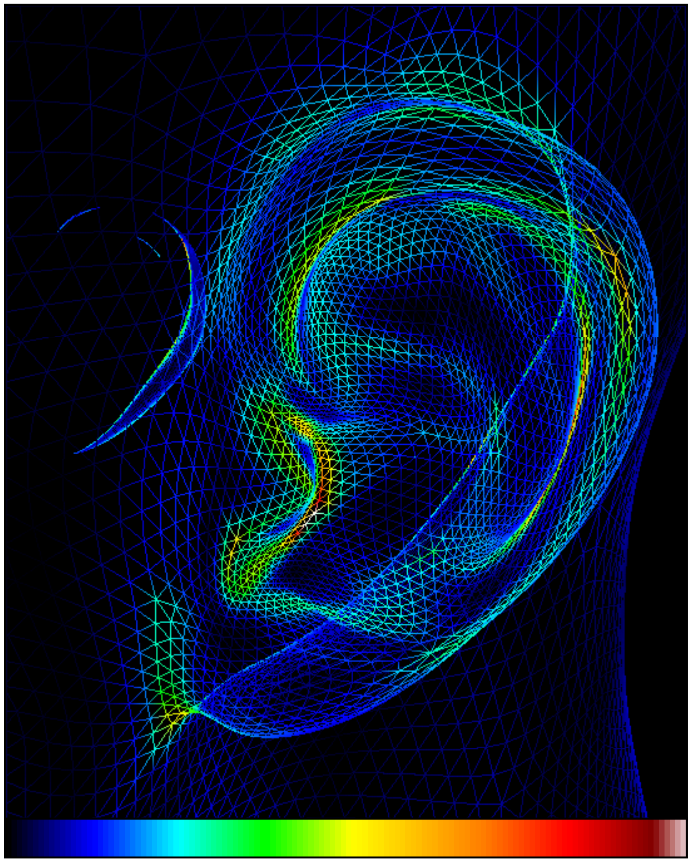








# Example of density function



1. Curvature related density function

Discrete Differential-Geometry Operators

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

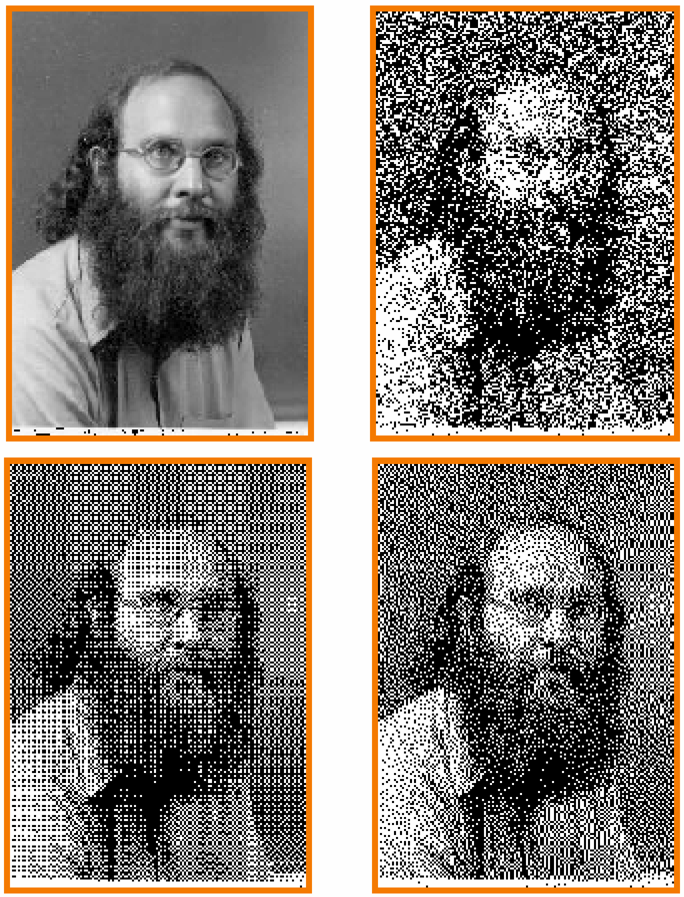
## Algorithm

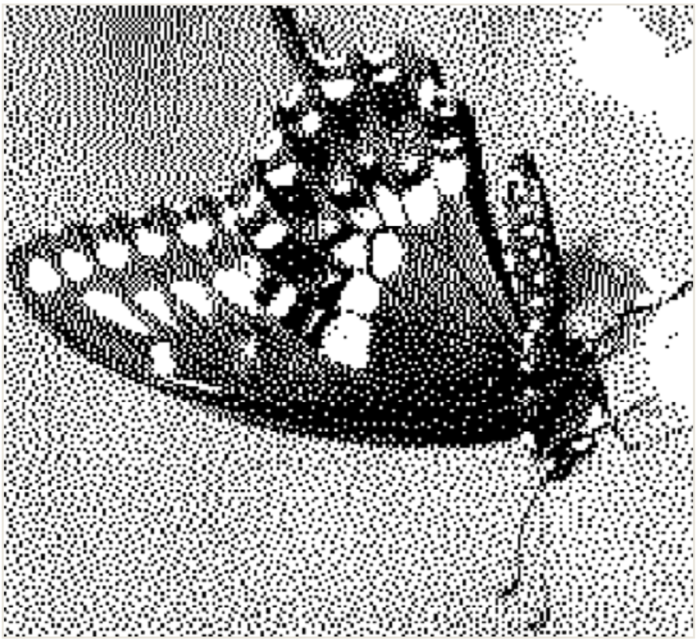
1. sample repartition
   -> fast (initial guess)
2. parameterization
3. meshing
4. sample placement
   -> slow, precise

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

# Sample repartition

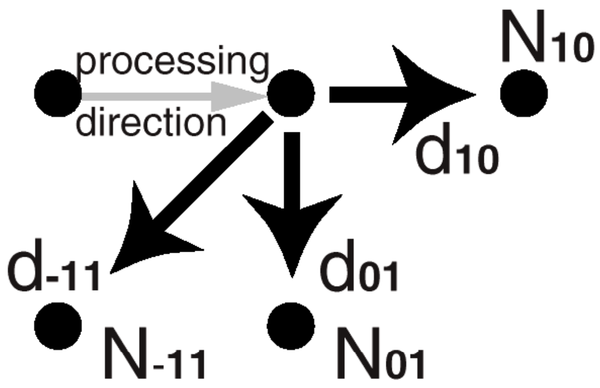
Use core principle of error diffusion

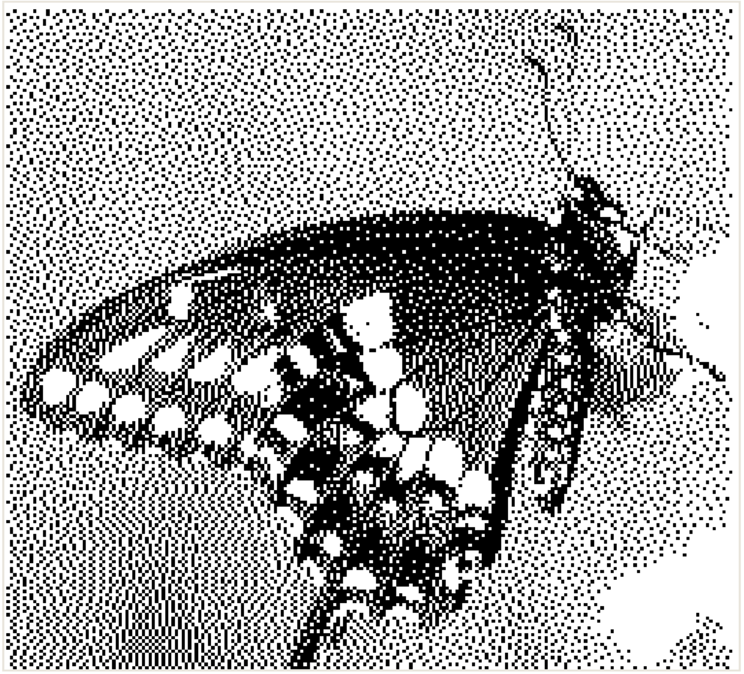




[Ostromoukhov '01]

# Error diffusion



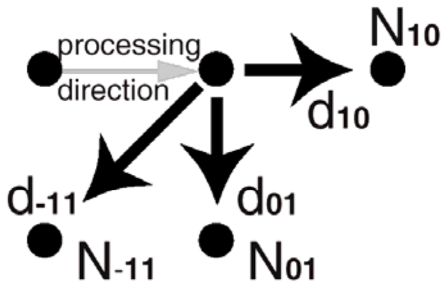



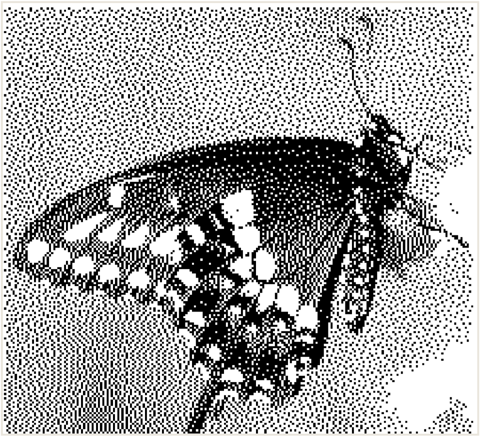
Two main components:

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

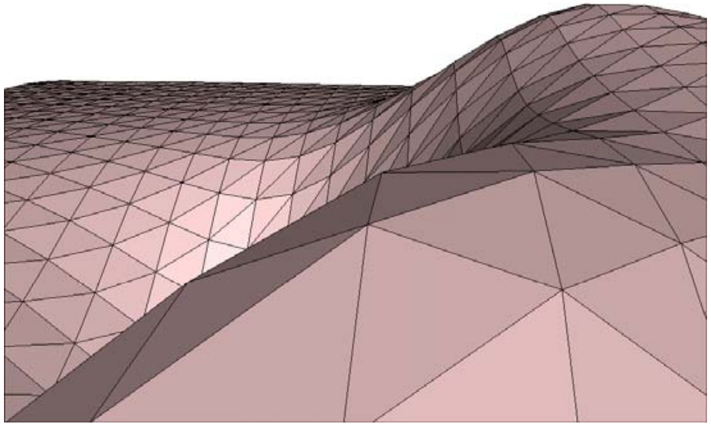
# Sample repartition

Idea: generalize error diffusion over discrete surfaces



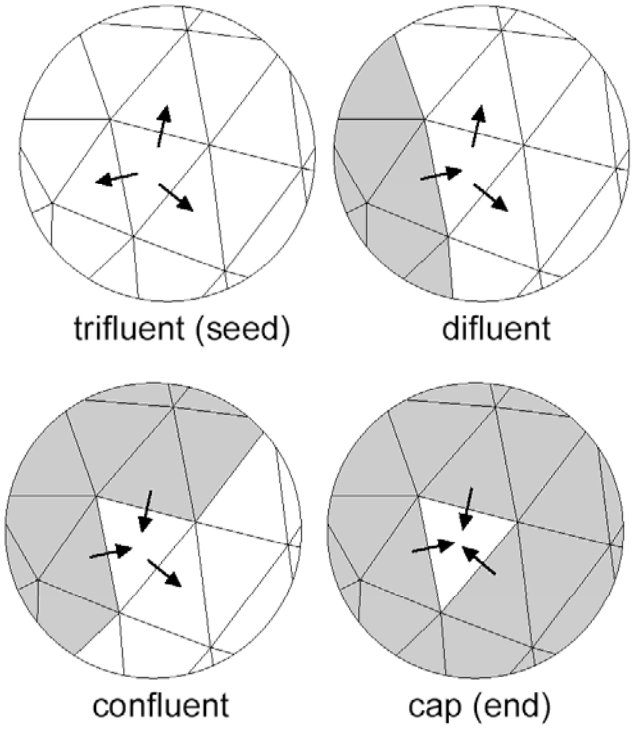
Image

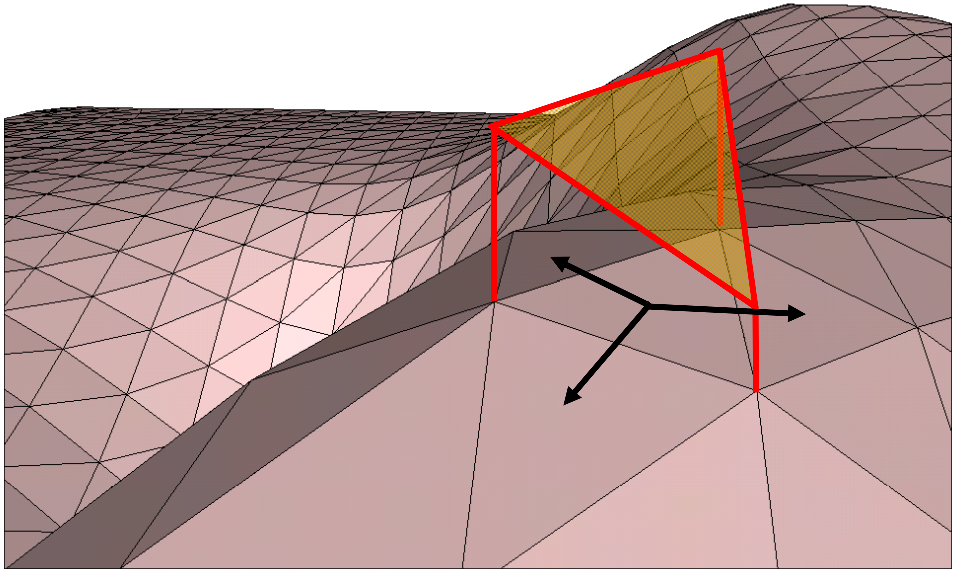


Model: triangle mesh + feature skeleton

# 2D Error diffusion

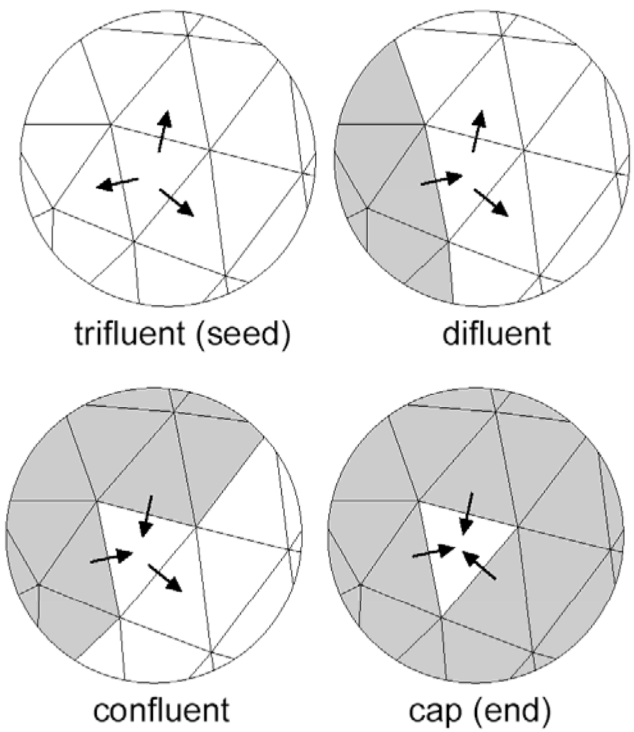
- Compute processing path on triangles
  - organize fluency on triangles

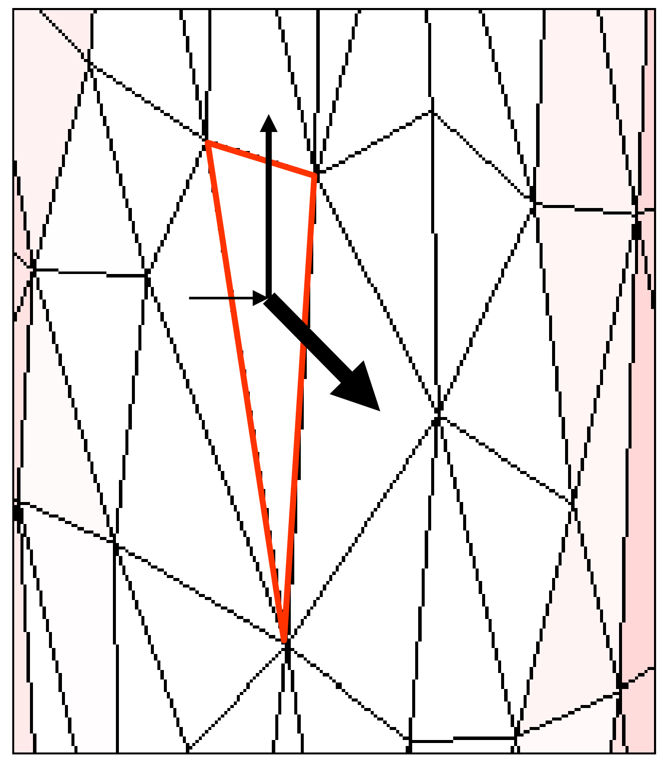


# 2D Error diffusion

- Coefficients for error diffusion
  - edge length: simulate "aperture"



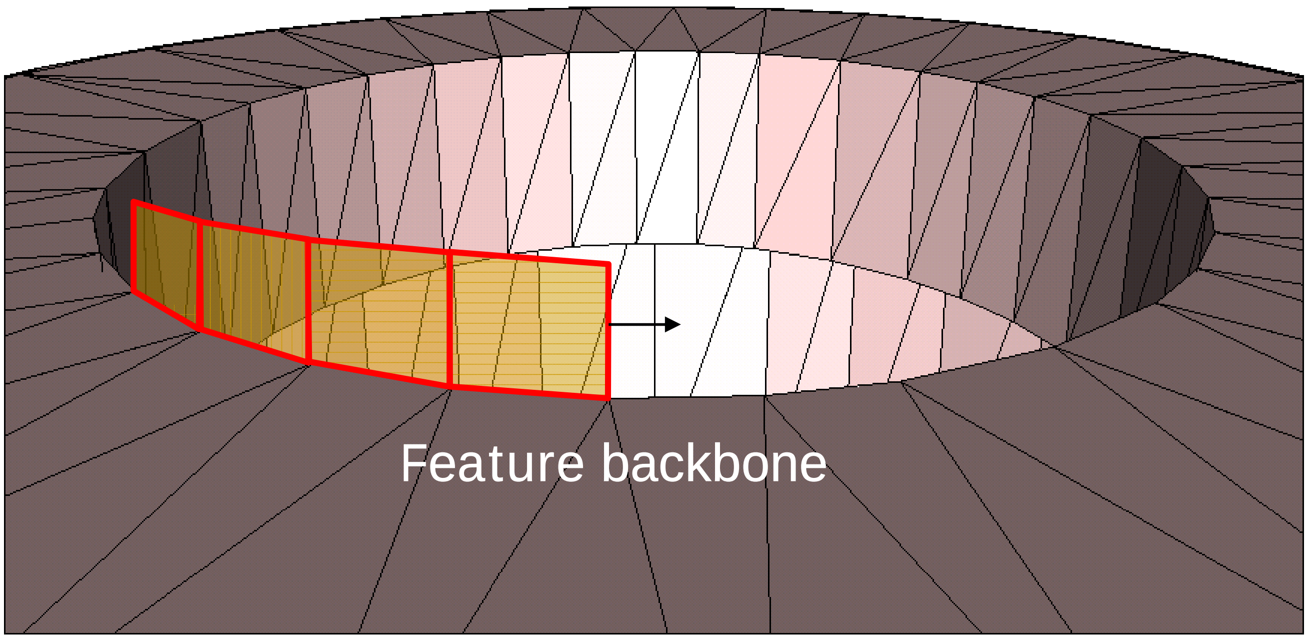
# 2D Error diffusion

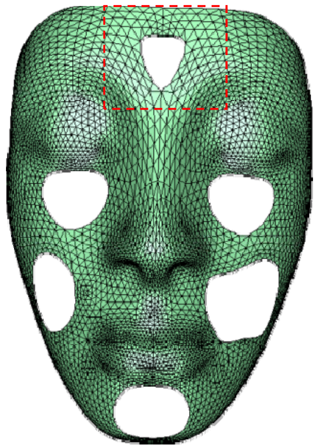
Algorithm:

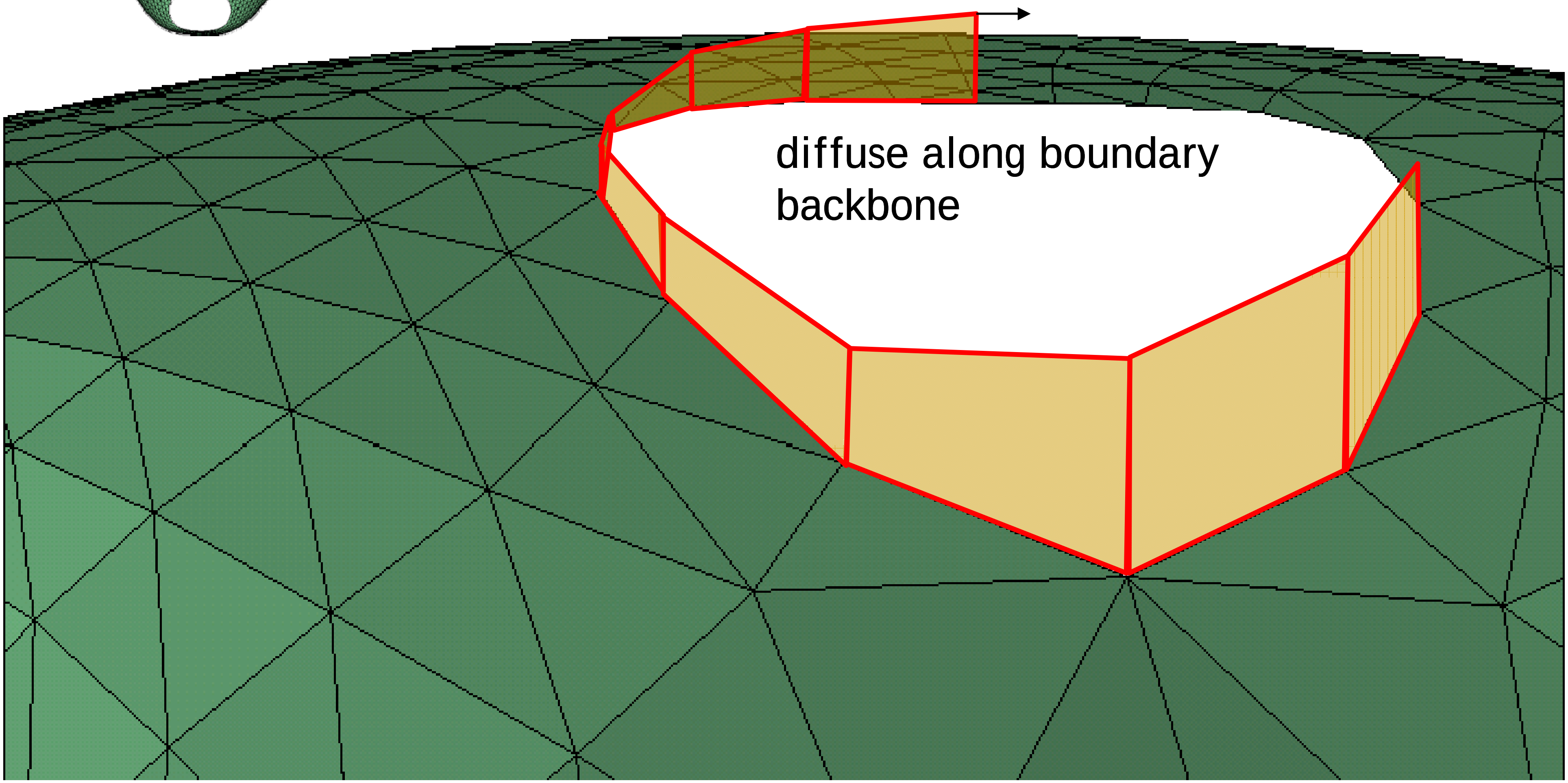
1. read the total amount of density on $f$;
2. from the sampling rate ==> deduce the number of samples to distribute on the current face (the distribution inside a triangle is described later). This number is rounded to the nearest integer value, such a rounding generating a signed quantization error $e$, translated in amount of density, to diffuse on incident unprocessed faces;
3. the error $e$ is diffused to the unprocessed faces incident to $f$ through the corresponding edges and proportionally to their length. This heuristic mimics the notion of "geometric aperture" for diffusion;
4. flag $f$ as processed;
5. pick the next face to proceed and restart from step 1 if any, stop otherwise.

# 1D Error diffusion

- Given processing path on backbones
  - fluency on edges

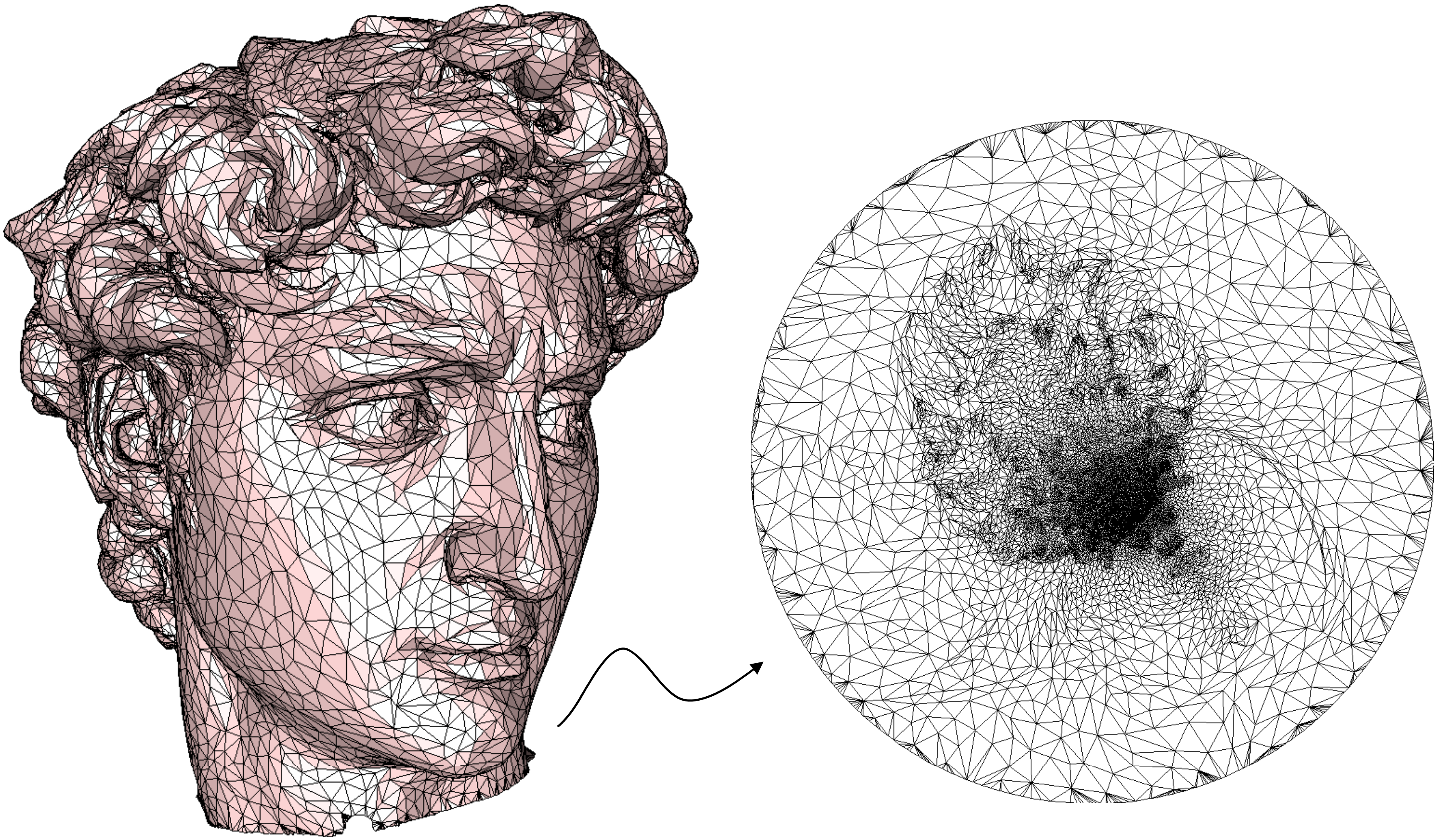





diffuse along boundary
backbone

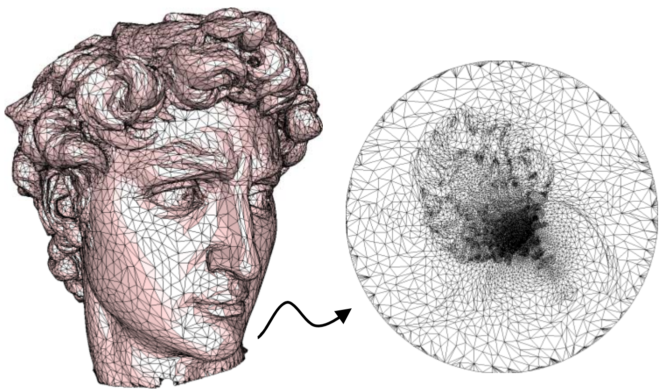
- Motivation
- Previous work
- Contributions
- Algorithm
  - sample repartition *error diffusion*
  - parameterization
  - meshing
  - sample placement

# Parameterization



~remove embedding (back to manifold)

# Parameterization

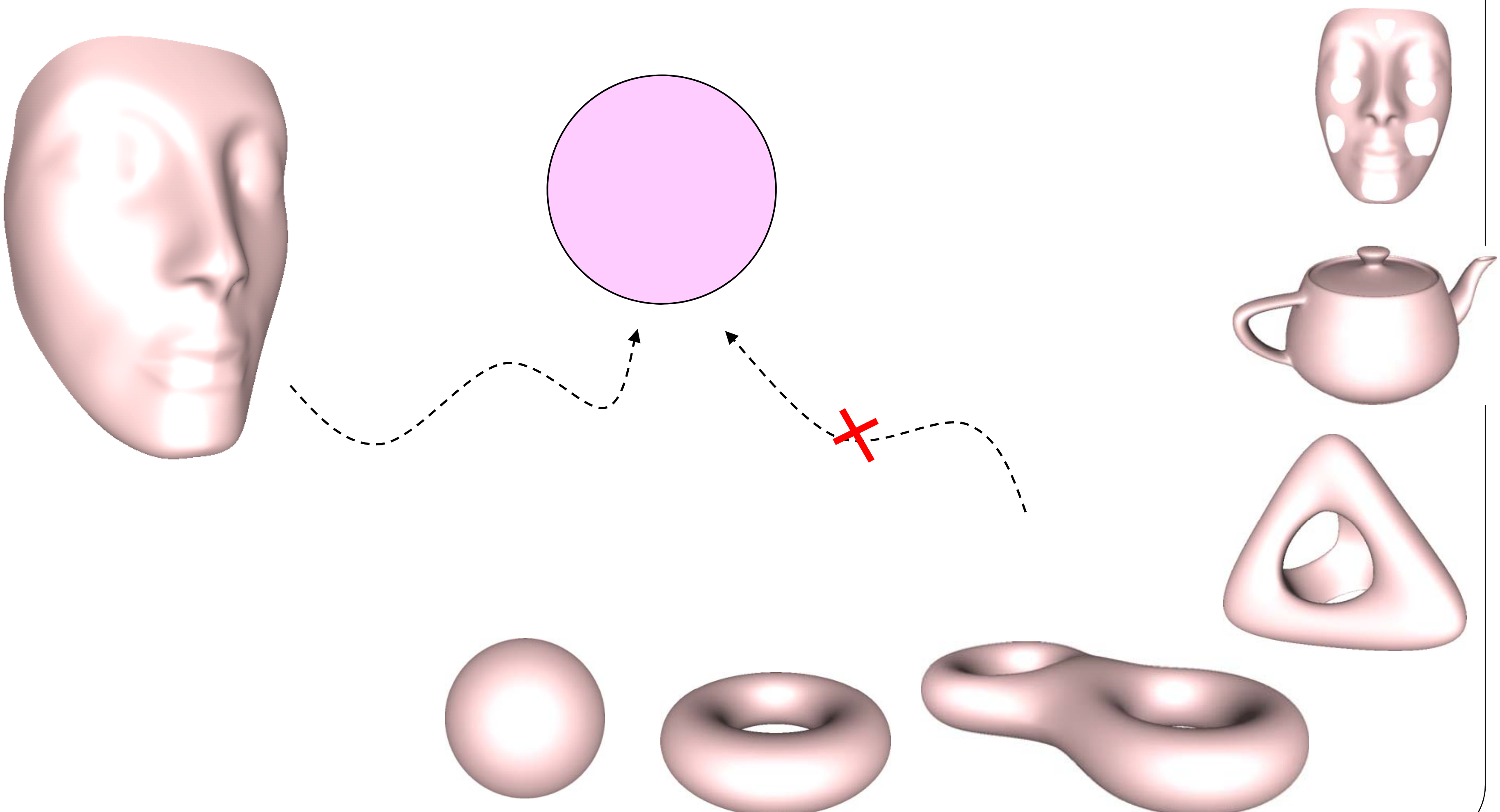


Motivations:

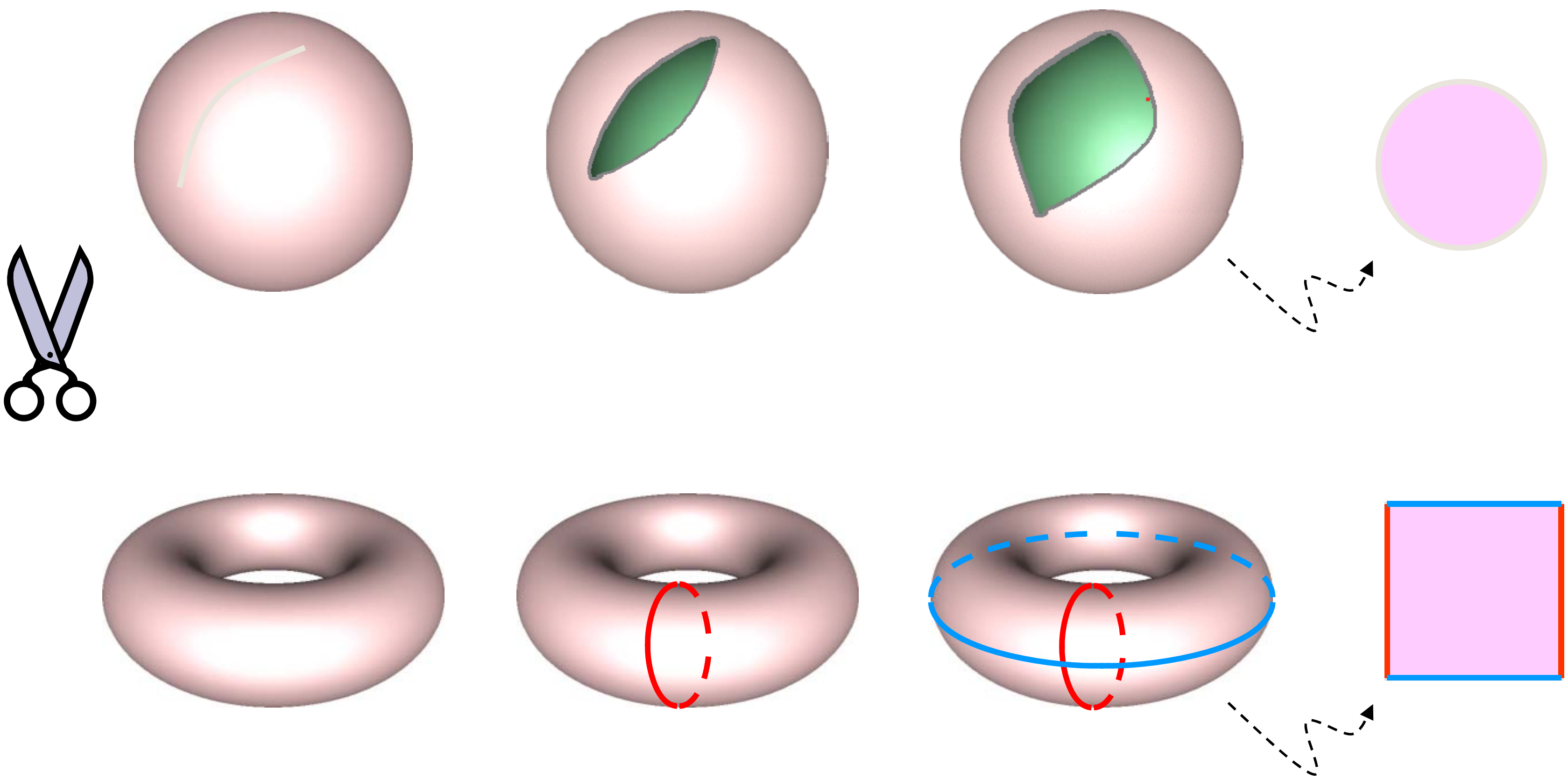
- Map signals
  - modulation
  - enrich geometry (GPU)
- Algorithms in flatland
  - analysis
  - sampling
  - meshing
  - etc.

# Parameterization

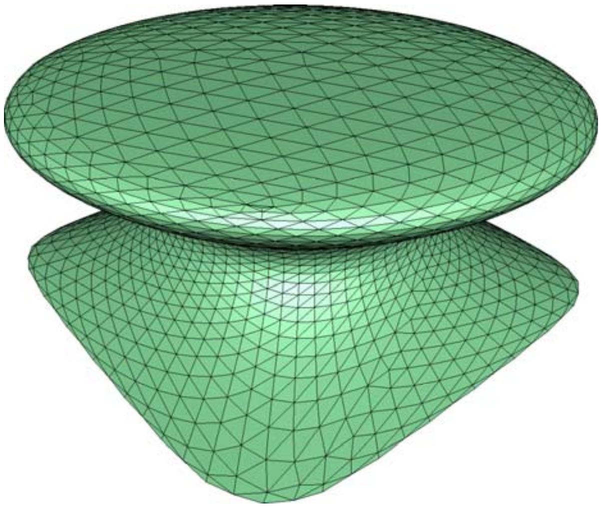
-> simple domain

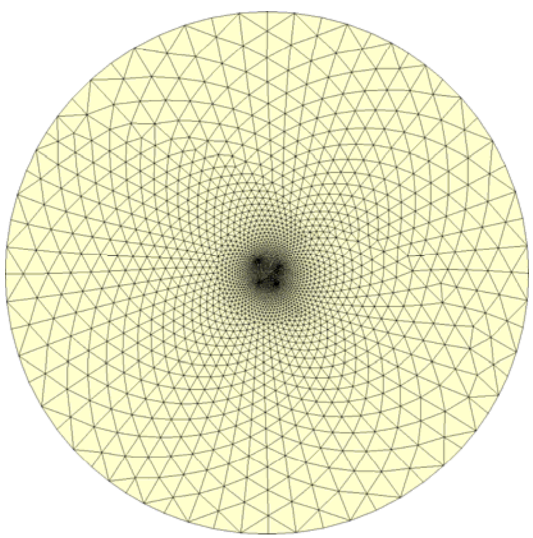


# Surface cutting



# Convex embedding





[Tutte]

# More parameterizations