Remeshing

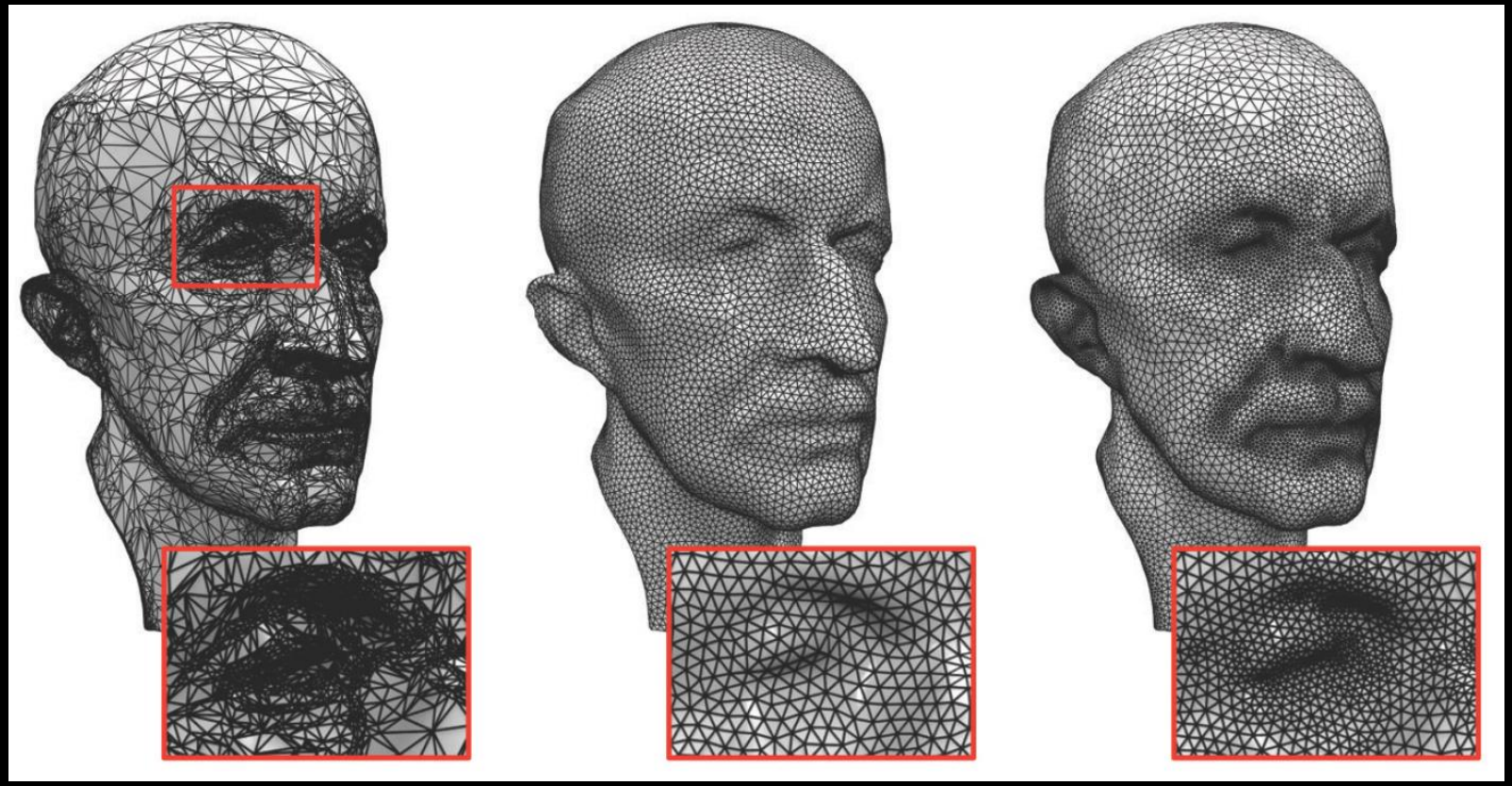
Xiao-Ming Fu

Outlines

- Isotropic remeshing
  - Error-bounded method
    - Error-Bounded and Feature Preserving Surface Remeshing with Minimal Angle Improvement
  - Improve small and large angles
  - Optimal Delaunay Triangulation (ODT)
  - Centroidal Patch Triangulation (CPT)
- Anisotropic remeshing
  - Introduction
  - ODT with general convex function
  - Local convex triangulation
  - Partial-based method
  - High-dim embedding

Remeshing

- Given a 3D mesh, compute another mesh, whose elements satisfy some quality requirements, while approximating the input acceptably.



Remeshing

- A key technique for mesh quality improvement
- Goal
  - 1. Reduce the complexity of an input surface mesh
    - subject to certain quality criteria
  - 2. Improve the quality of a mesh
    - Different applications imply different quality criteria and requirements.

- **Given a 3D mesh, compute another mesh, whose elements satisfy some quality requirements, while approximating the input acceptably.**

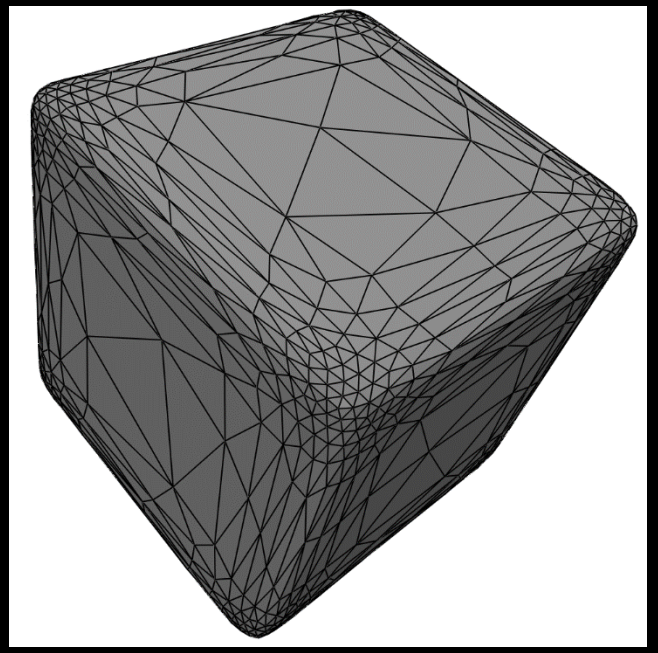
Discussion

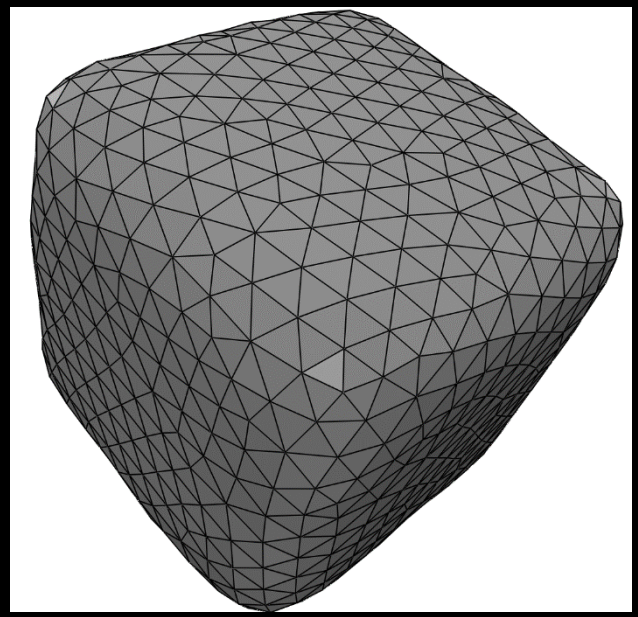
- Input: a manifold triangle mesh or part of it.
- Mesh quality
  - Sampling density
  - Regularity
  - Size
  - Orientation
  - Alignment
  - Shape of the mesh elements.
  - Non-topological issues (mesh repair)

Local Structure

- Element shape
  - Isometric
  - Anisotropic

anisotropic

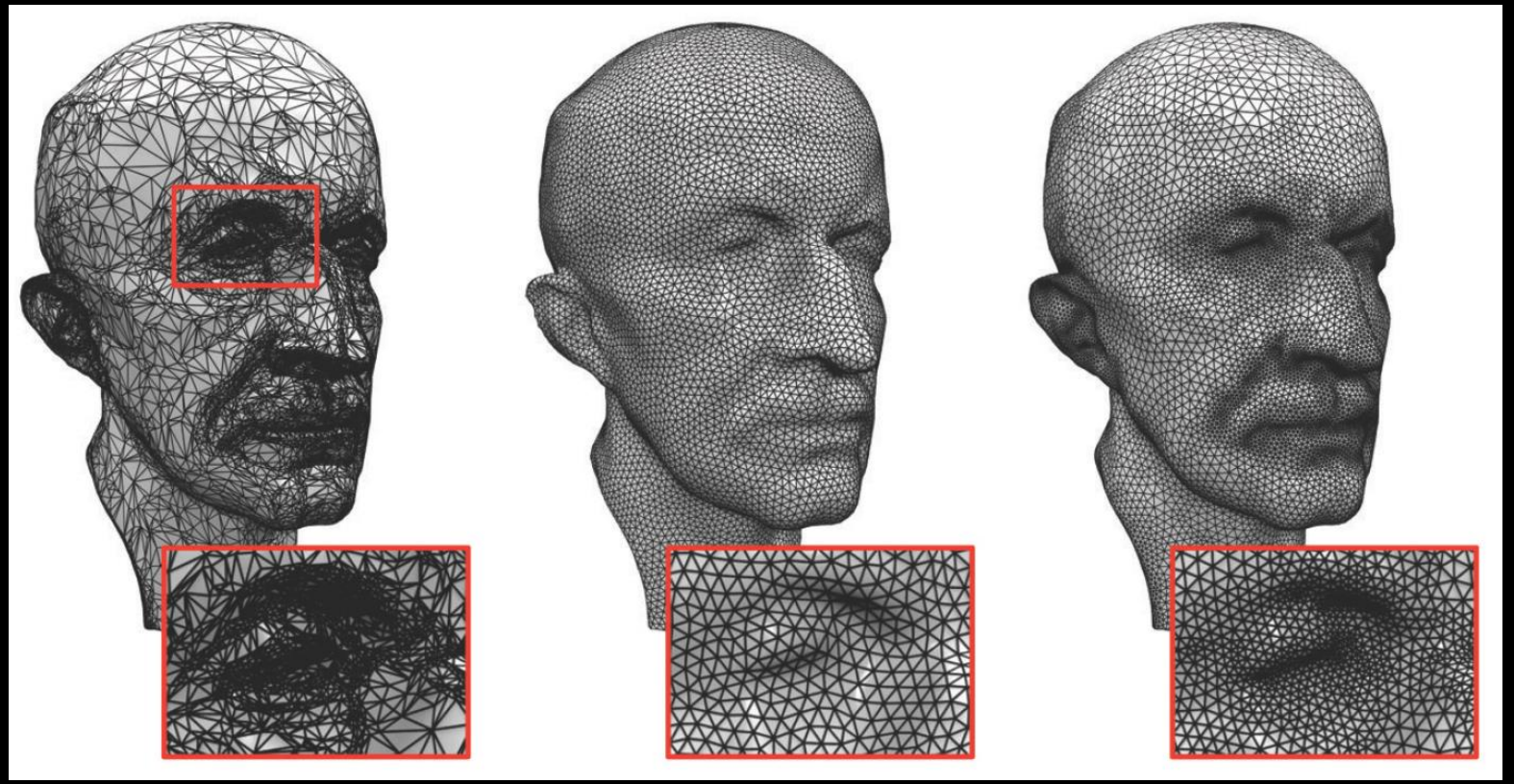




isotropic

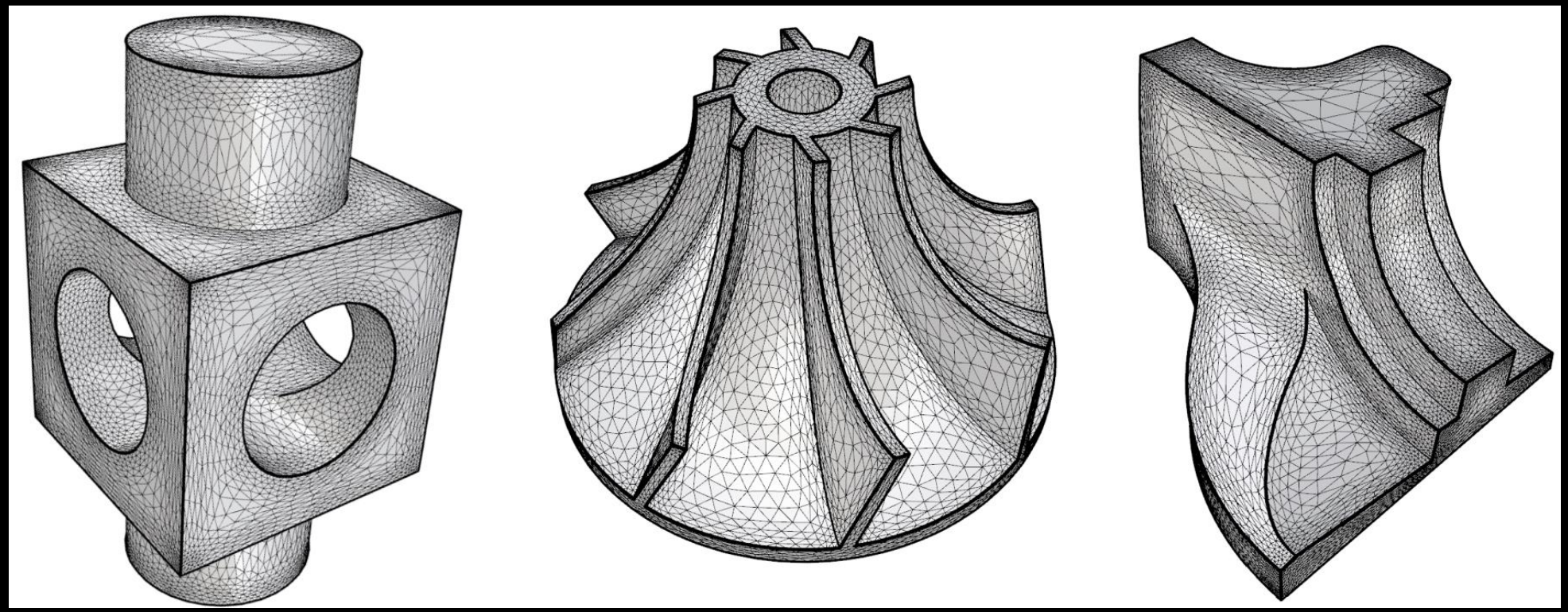
Local Structure

- Element density
  - uniform VS. nonuniform or adaptive



Local Structure

- Element alignment and orientation
  - elements should align to sharp features
  - orientation of anisotropic elements



Global structure

- Vertex
  - Regular
    - Valence = 6 for triangle mesh
    - Valence = 4 for quad mesh
  - Irregular (singular)
- Global
  - Irregular
  - Semiregular
    - regular subdivision of a coarse initial mesh
  - Highly regular
    - most vertices are regular
  - Regular
    - all vertices are regular

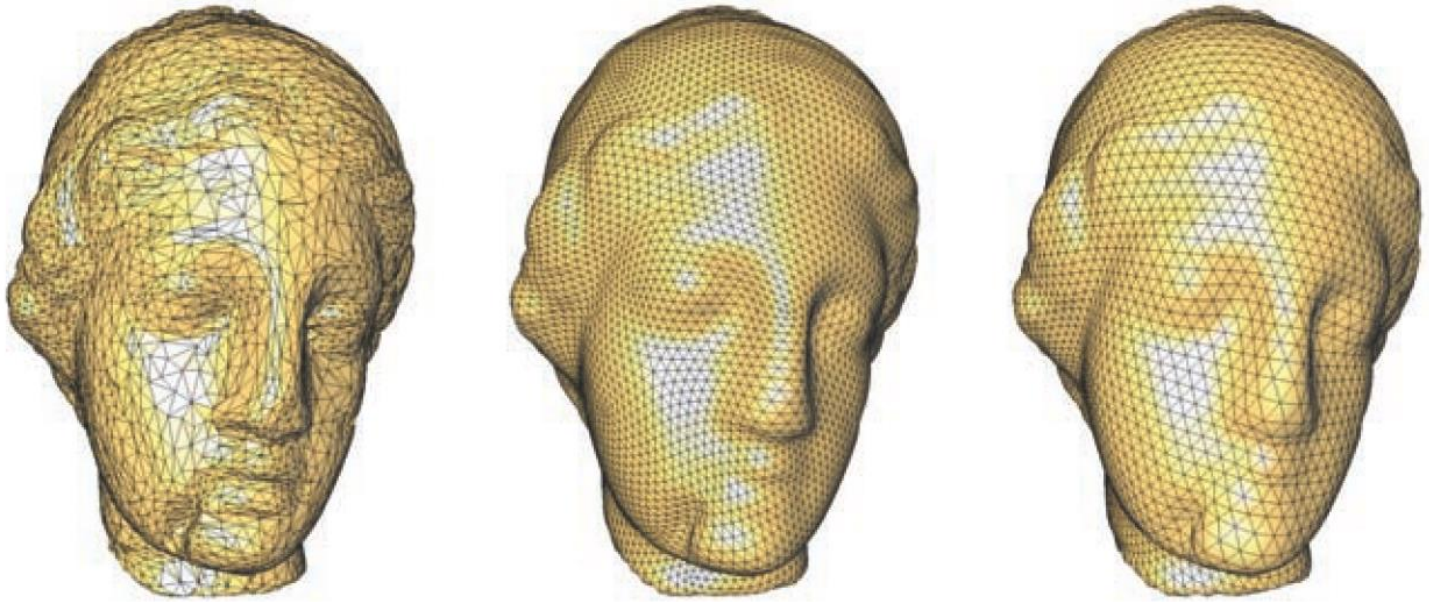
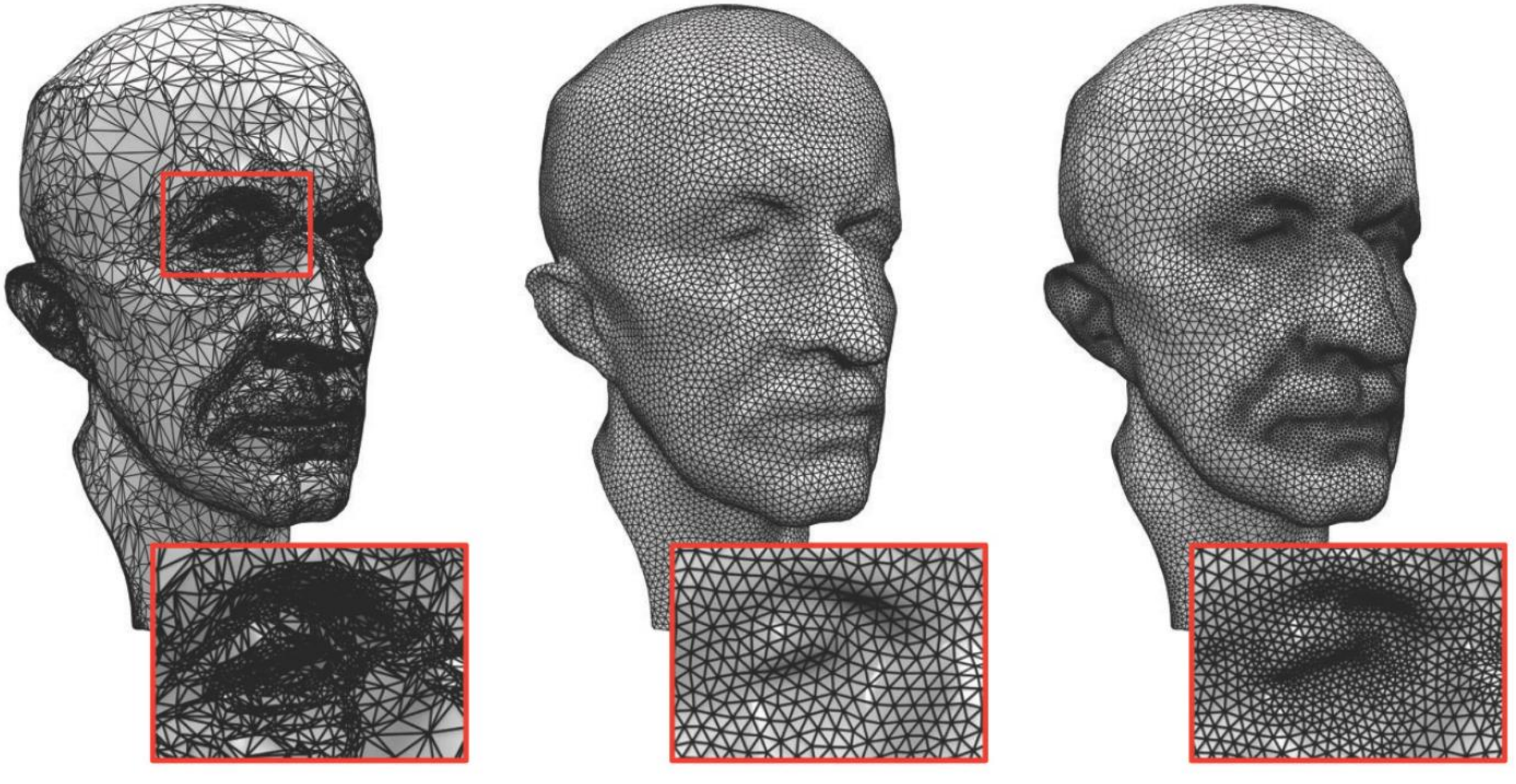


Figure 6.2. Meshes: Irregular (left), semiregular (center), and regular (right). (Model courtesy of Cyberware.)

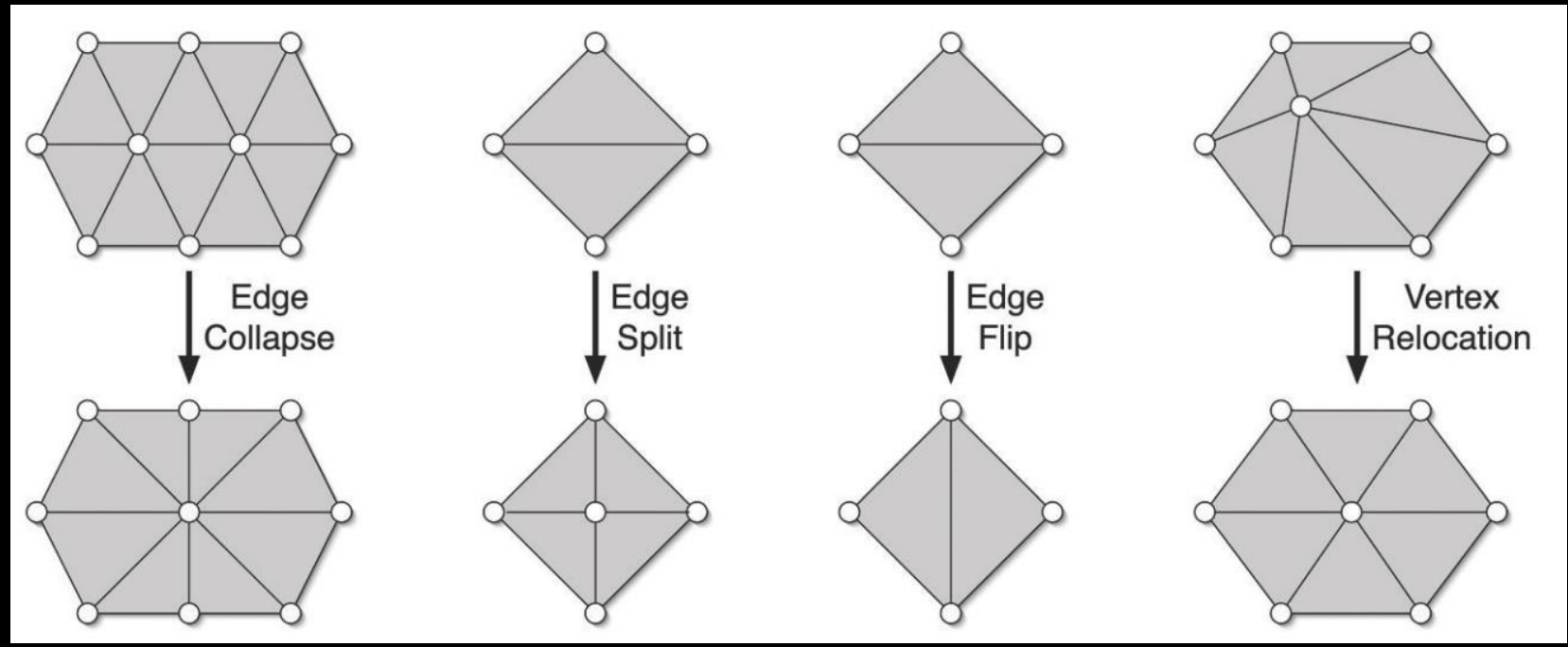
Outlines

- Isotropic remeshing
  - Error-bounded method
    - Error-Bounded and Feature Preserving Surface Remeshing with Minimal Angle Improvement
  - Improve small and large angles
  - Optimal Delaunay Triangulation (ODT)
  - Centroidal Patch Triangulation (CPT)
- Anisotropic remeshing
  - Introduction
  - ODT with general convex function
  - Local convex triangulation
  - Partial-based method
  - High-dim embedding



Incremental Remeshing

- Input: triangle mesh and a target edge length
- Method
  - **Split long edges**
  - **Collapse short edges**
  - **Relocate vertices**



Pseudo-code

Remesh(target edge length)

Low_e = $\frac{4}{5}$ * target edge length

High_e = $\frac{4}{3}$ * target edge length

for i = 0 to 10 do

    split long edges(high_e)

    collapse short edges(low_e, high_e)

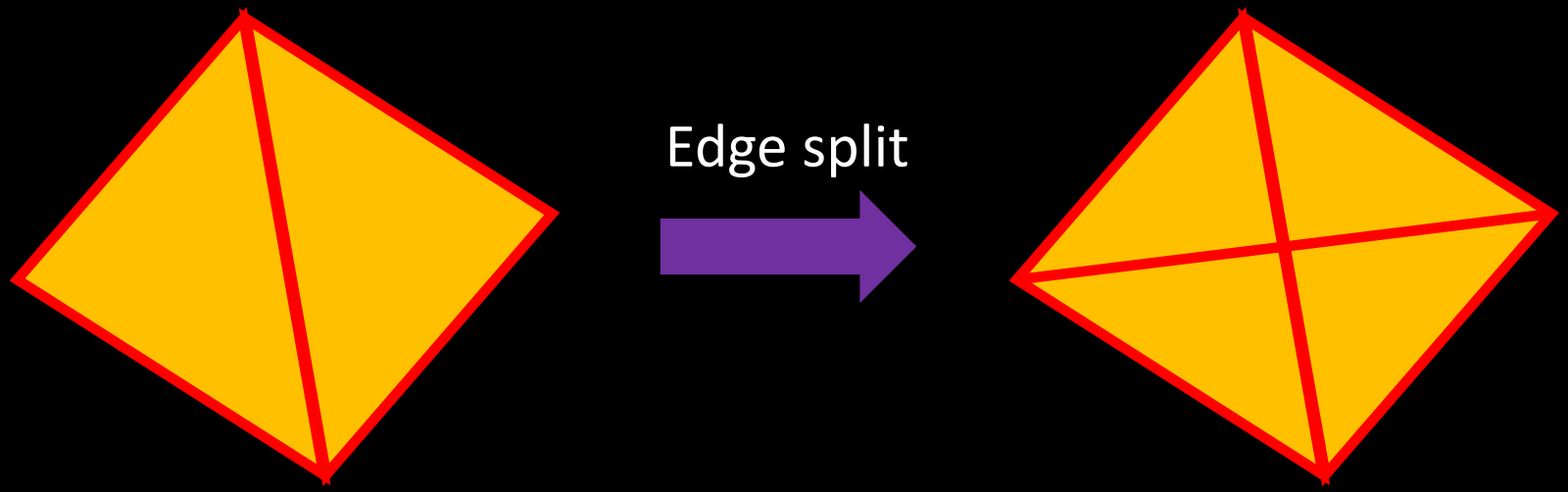
    equalize valences()

    tangential relaxation()

    project to surface()

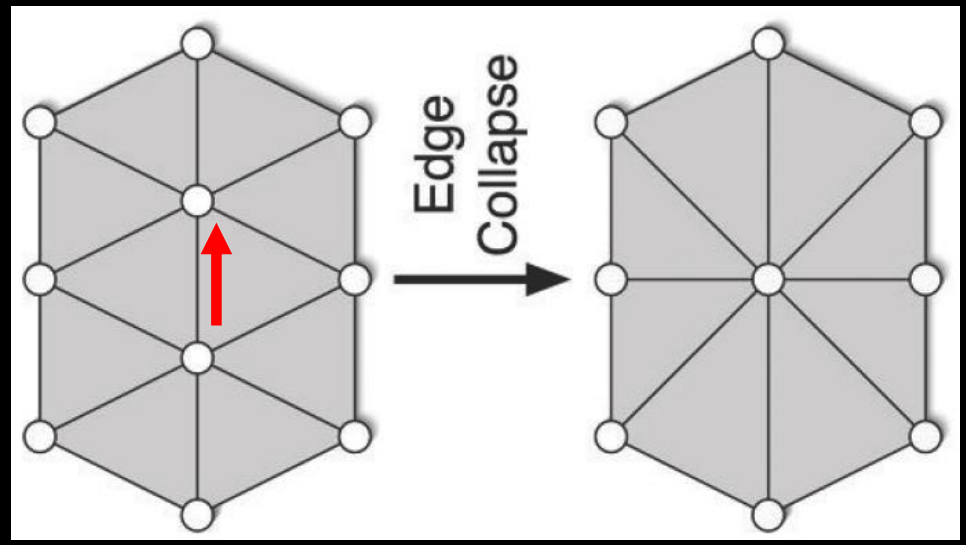
Discussion

- Proper threshold $\frac{4}{5}$ and $\frac{4}{3}$ are essential.
- split long edges(high_e)
  - visits all edges of the current mesh
  - If an edge is longer than the given threshold high_e, the edge is split at its midpoint and the two adjacent triangles are bisected



Discussion

- collapse short edges(low_e, high_e)
  - collapses and thus removes all edges that are shorter than a threshold low_e.
- Issue
  - by collapsing along chains of short edges, the algorithm may create new edges that are arbitrarily long.
  - This issue is resolved by testing before each collapse whether the collapse would produce an edge that is longer than high.
  - If so, the collapse is not executed.



Discussion

- equalize valences()
  - equalizes the vertex valences by flipping edges.
  - The algorithm tentatively flips each edge and checks whether the deviation to the target valences decreases.
- The tangential relaxation()
  - an iterative smoothing filter to the mesh
  - the vertex movement has to be constrained to the vertex tangent plane in order to stabilize the following projection operator.

Discussion

- The tangential relaxation(): uniform Laplacian weights

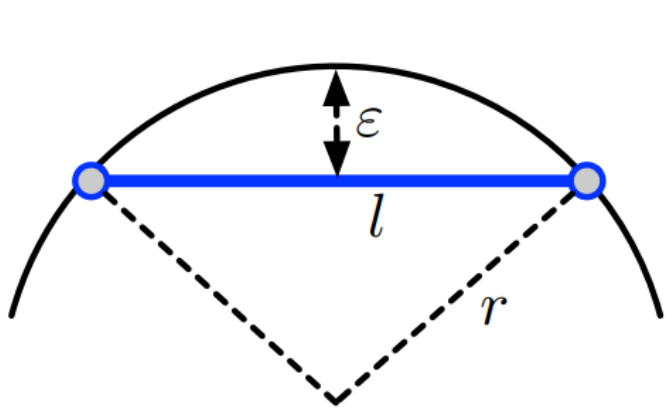
$$q = \frac{1}{N_p} \sum_{p_j \in \Omega(p)} p_j$$

projecting q onto p's tangent plane:

$$p' = q + nn^t(p - q)$$

- **Project to surface()**
  - maps the vertices back to the surface.
  - **CGAL: AABB tree**

Adaptive edge length



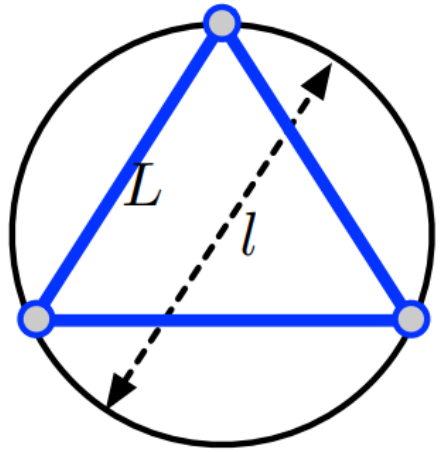


Figure 1: Left: *How to determine the edge length l for approximating a circular arc up to an error tolerance* ε. Right: *For surfaces the edge length l has to be scaled to yield the target edge length L for an equilateral triangle. In the surface case, the left and right figures can be considered a cross-section or a top view of each other.*

- $l = 2\sqrt{2r\varepsilon - \varepsilon^2}$
- $L(x_i) = \sqrt{\frac{6\varepsilon}{k_i} - 3\varepsilon^2}$

k_i: maximum absolute curvature at x_i

Discussion

- Feature preservation.
  - feature edges and vertices have already been marked in the input model.
  - 1. Corner vertices with **more than two or exactly one incident feature edge** have to be preserved and are excluded from all topological and geometric operations.
  - 2. Feature vertices may only be collapsed along their incident feature edges.
  - 3. Splitting a feature edge creates two new feature edges and a feature vertex.
  - 4. Feature edges are never flipped.
  - 5. Tangential smoothing of feature vertices is restricted to univariate smoothing along the corresponding feature lines. (How to do???)

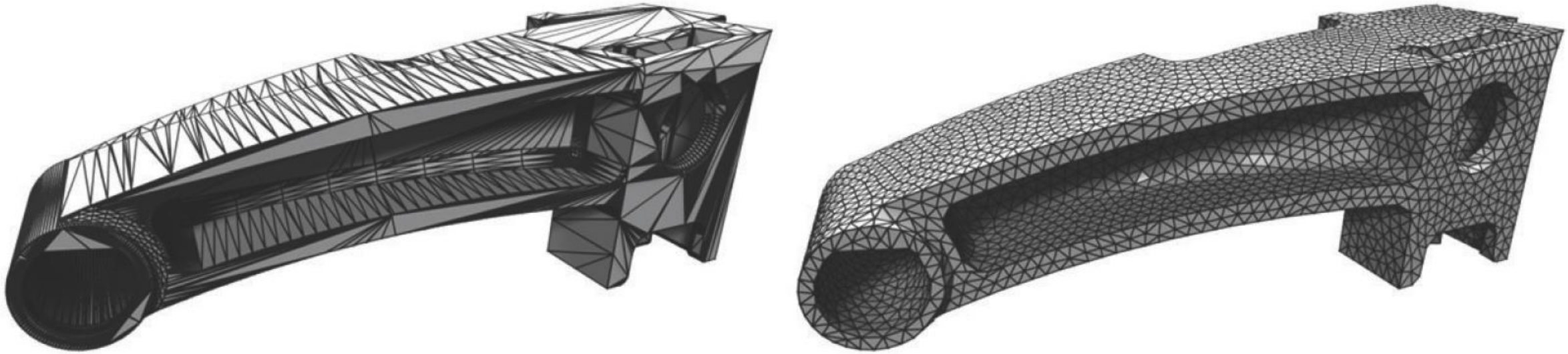


Figure 6.13. Isotropic, feature-sensitive remeshing (right) of a CAD model (left).

Parameterization-based method

- All remeshing algorithms compute point locations on or near the original surface.
- Global parameterization.
  - The input model is globally parameterized onto a 2D domain.
  - Sample points can then be easily distributed and relocated in the 2D domain.
  - Be lifted to three dimensions
  - Disadvantages:
    - Parametric distortion
    - Discontinuities when the mesh needs to be cut into a topological disk.

Global parameterization

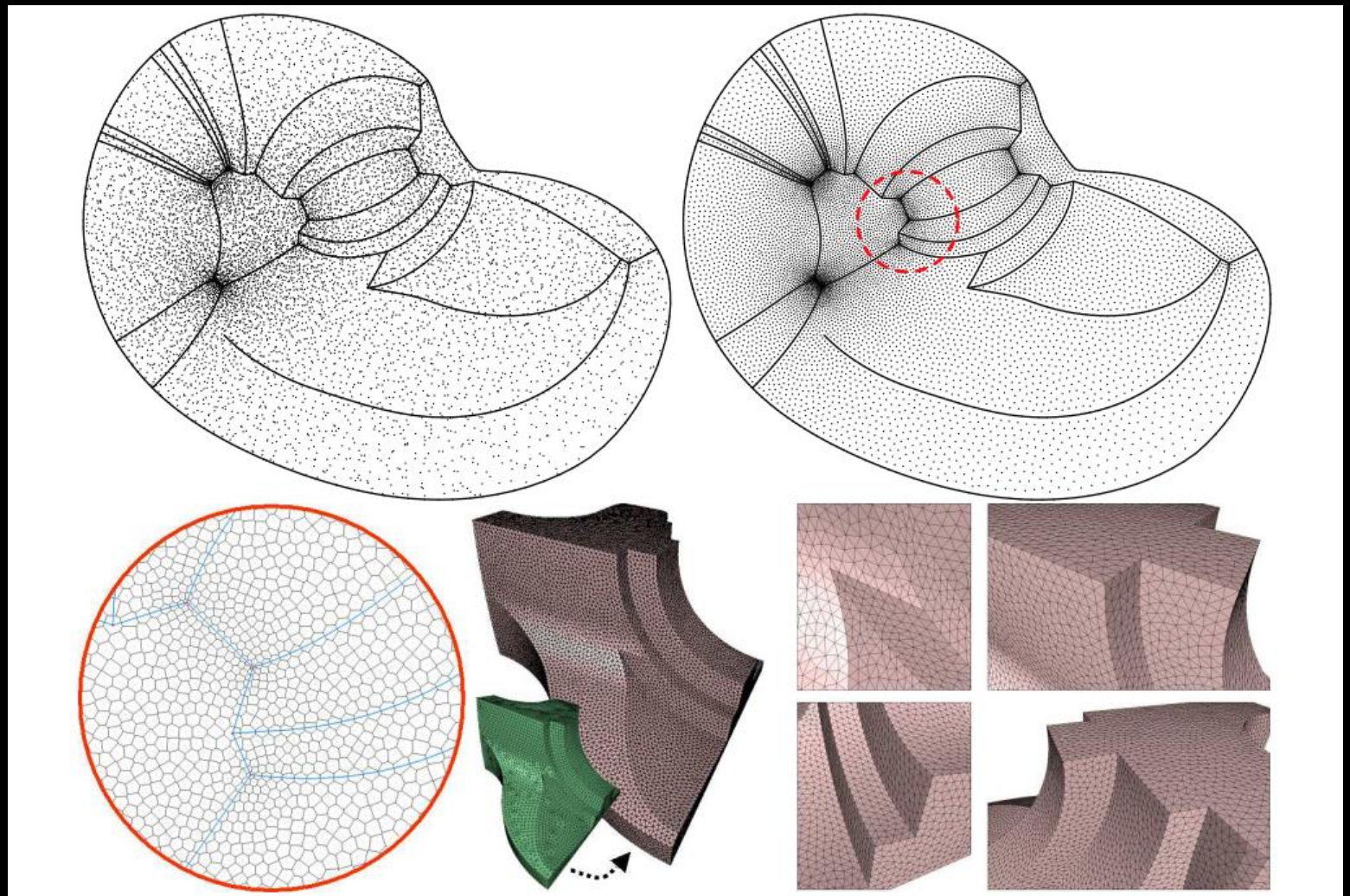


Figure 15: *Uniform remeshing of the fandisk model with 10k vertices. Top row: Initial sampling computed by direct error diffusion, and sampling obtained after 20 iterations of Lloyd relaxation. Bottom row: closeup over the Voronoi tessellation after Lloyd convergence, global view of the remeshed model, and several closeups nearby the features.*

Closed surface

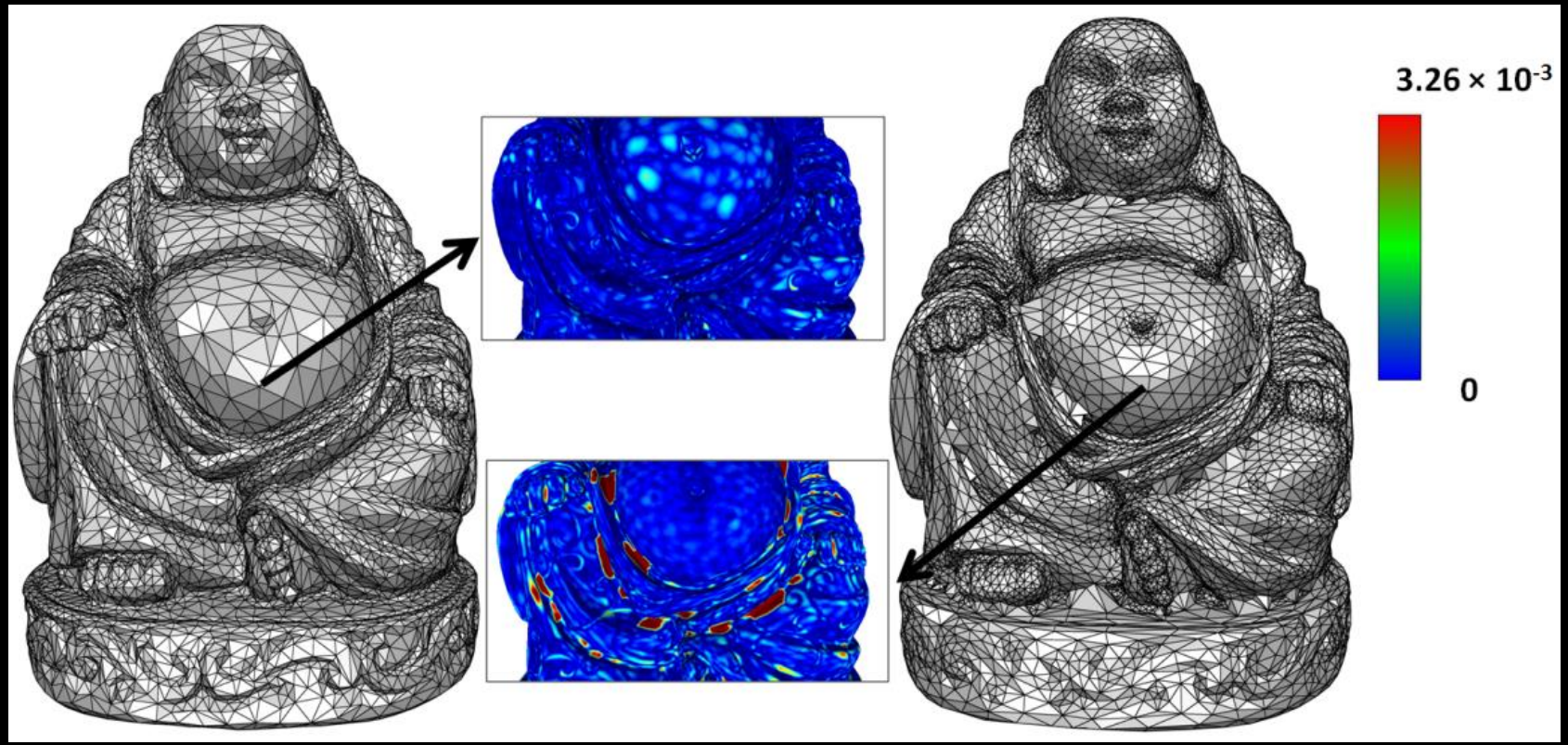
- Parameterization-based method requires cut paths.
- How to generate the cut paths:
  - Distortion, D-Charts
  - Visit at least twice

Outlines

- Isotropic remeshing
  - Error-bounded method
    - Error-Bounded and Feature Preserving Surface Remeshing with Minimal Angle Improvement
  - Improve small and large angles
  - Optimal Delaunay Triangulation (ODT)
  - Centroidal Patch Triangulation (CPT)
- Anisotropic remeshing
  - Introduction
  - ODT with general convex function
  - Local convex triangulation
  - Partial-based method
  - High-dim embedding

Approximation

- Definition:
  - Given a 3D mesh, compute another mesh, whose elements satisfy some quality requirements, while approximating the input acceptably.



Hausdorff distance

https://en.wikipedia.org/wiki/Hausdorff_distance

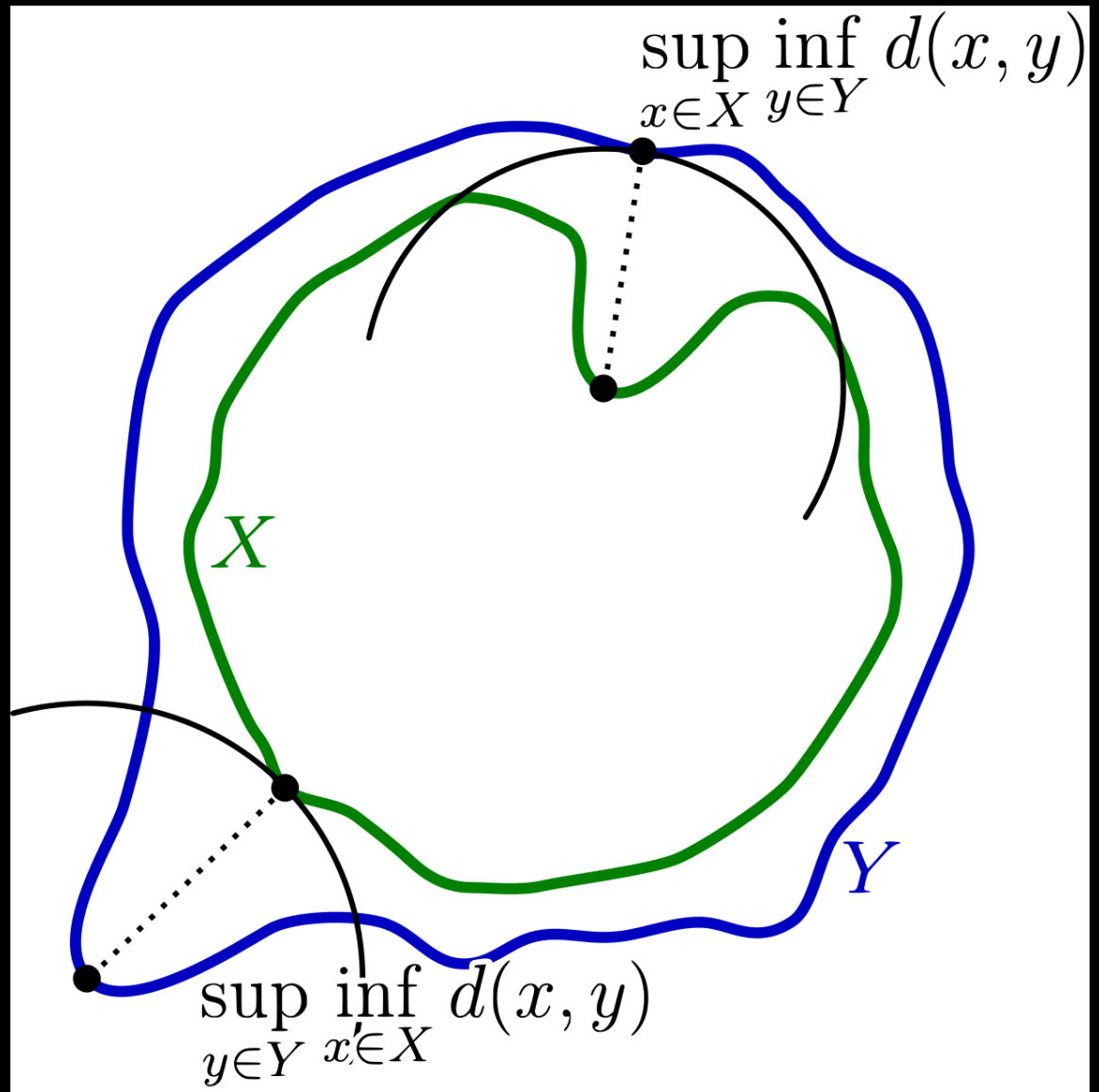
- **Hausdorff distance** measures how far two subsets of a metric space are from each other.

- Let X and Y be two non-empty subsets of a metric space (M, d). We define their Hausdorff distance $d_H(X, Y)$ by

$$d_H(X, Y) = \max\left\{\sup_{x \in X} \inf_{y \in Y} d(x, y), \sup_{y \in Y} \inf_{x \in X} d(x, y)\right\}$$

where sup represents the supremum and inf the infimum.

Hausdorff distance



In general:

$$\sup_{x \in X} \inf_{y \in Y} d(x, y) \neq \sup_{y \in Y} \inf_{x \in X} d(x, y)$$

Hausdorff distance on triangular mesh

- Hausdorff distance between M_1 and M_2:

$$d_H(M_1, M_2) = \max\{d_h(M_1, M_2), d_h(M_2, M_1)\}$$

where

$$d_h(S, T) = \max_{p \in S} \min_{q \in T} d(p, q)$$

Note: $d(p, T) = \min_{q \in T} d(p, q)$ is the distance from p to T.

The distance of a point p to a surface T is defined as the shortest distance between p and any point of T.

Approximating $d_H(M_1, M_2)$

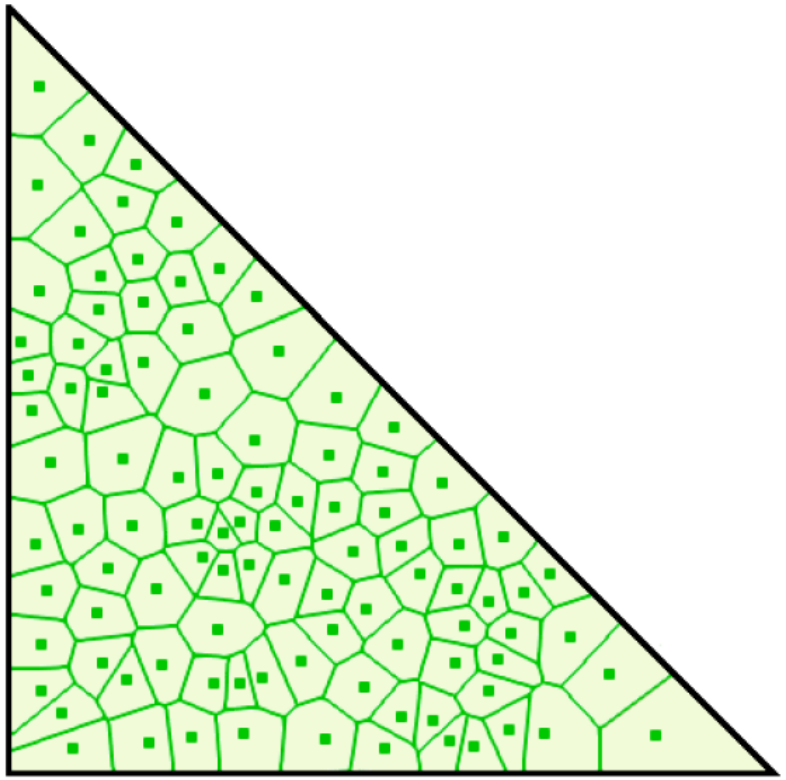
- Assume that M_1 is sampled by a point set $S_1 \subset M_1$ and M_2 is sampled by a point set $S_2 \subset M_2$
- $d_h(M_1, M_2)$ can be approximated by

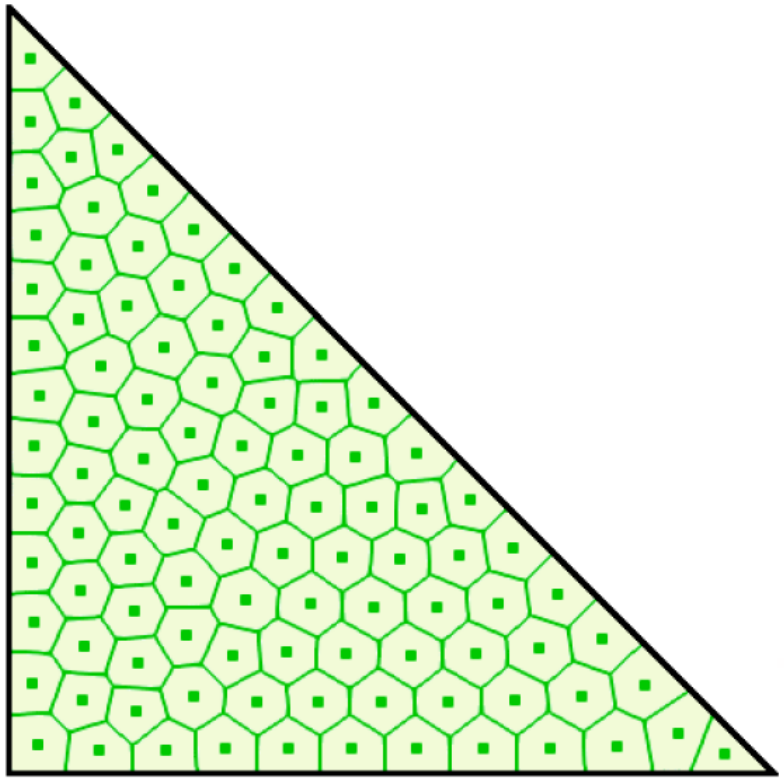
$$d_h(M_1, M_2) \approx \max_{a \in S_1} d(a, M_2)$$

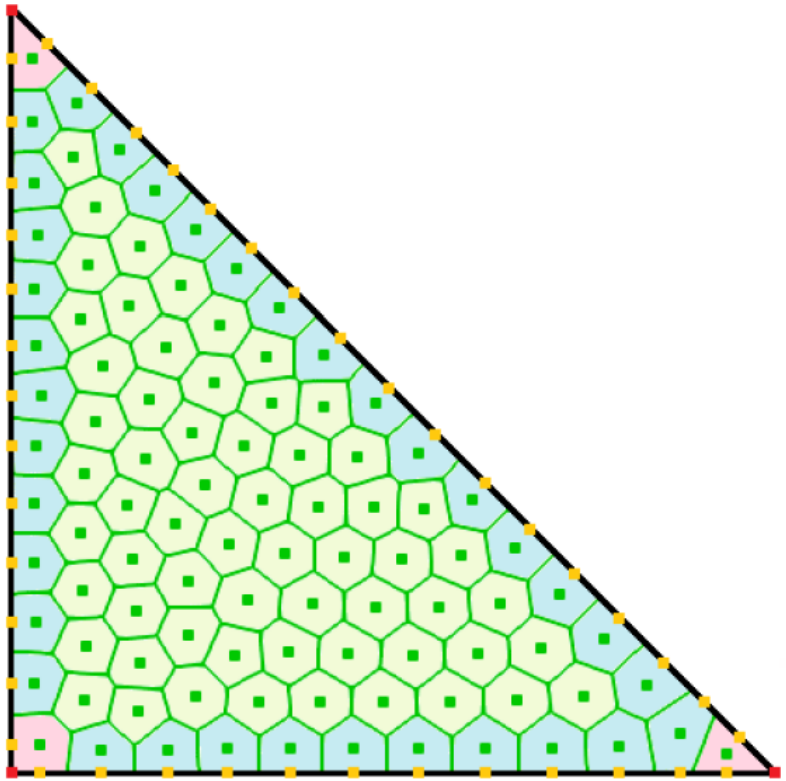
- $\Longrightarrow d_H(M_1, M_2) = \max\left\{\max_{a \in S_1} d(a, M_2), \max_{b \in S_2} d(b, M_1)\right\}$

- It provides significantly higher accuracy than a point cloud distance $d_H(S_1, S_2)$.

Sampling







Stratified sampling process

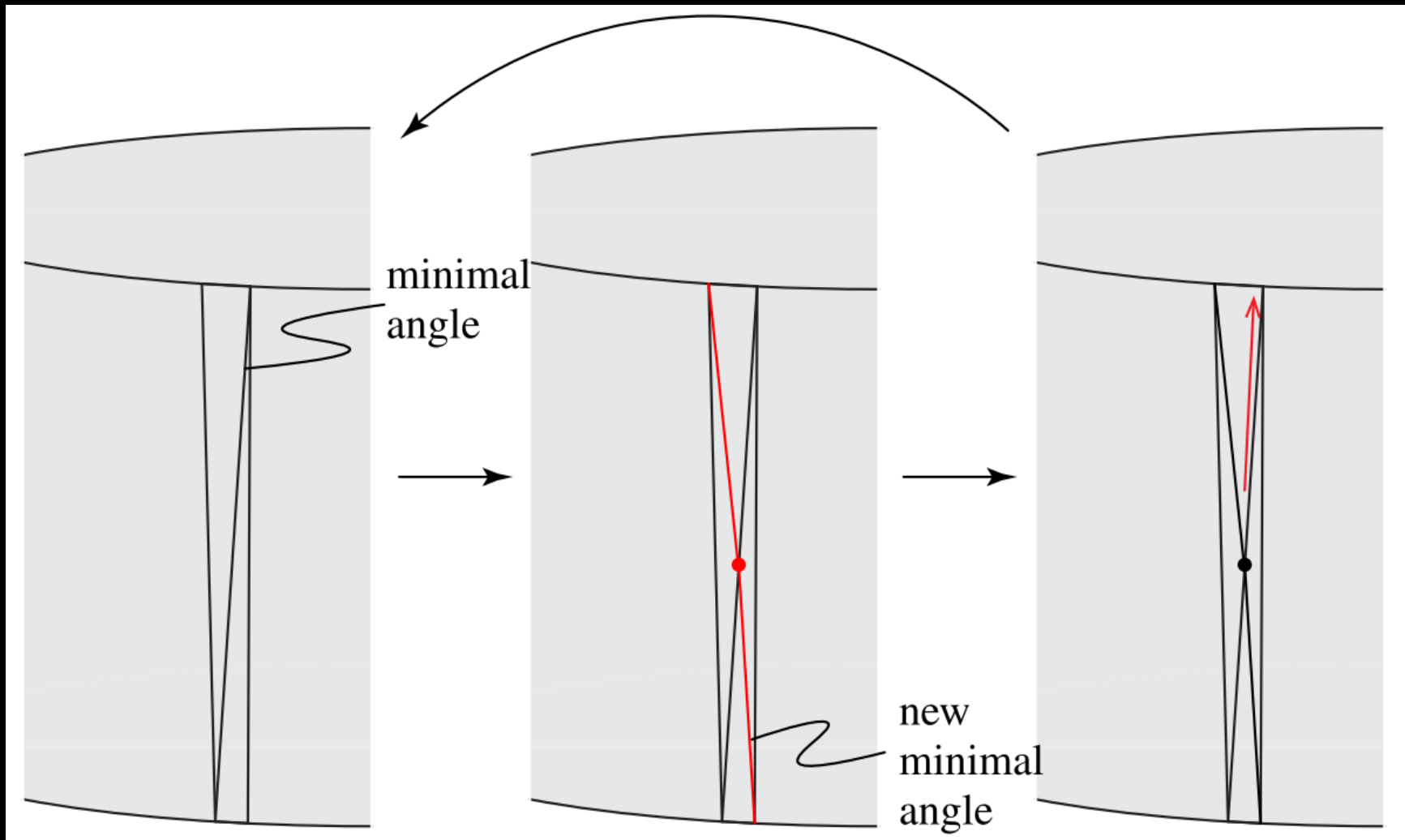
Error-bounded method

- A local operator is only executed if it respects the approximation error bound.
  - never leaving the feasible set of meshes with approximation error

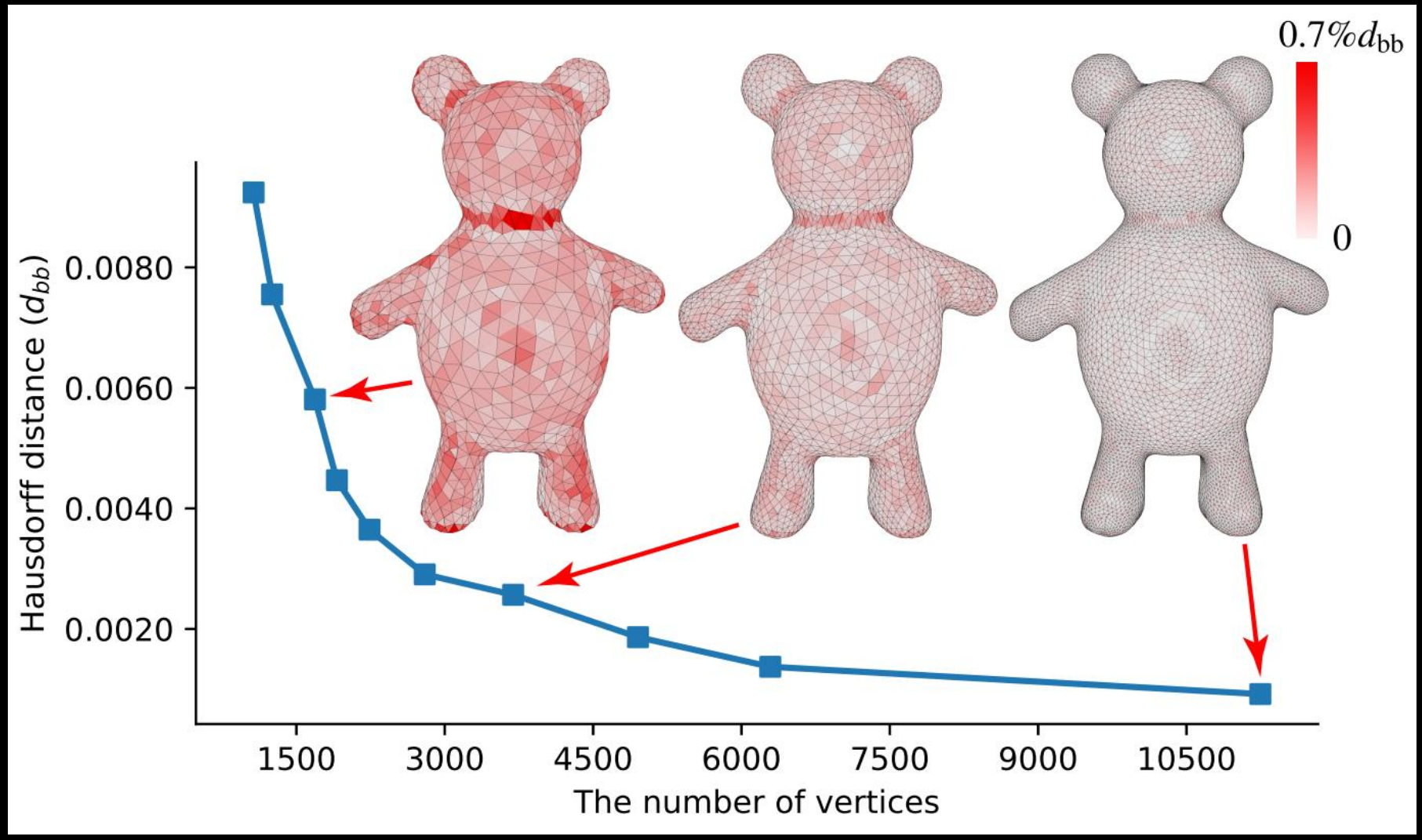
- A local operator
  - Edge collapse
  - Edge split
  - Edge flip
  - Smoothing / Relocation
  - ……

Main limitations

- 1. Time consuming
- 2. Infinite loop



Thinking from a different view

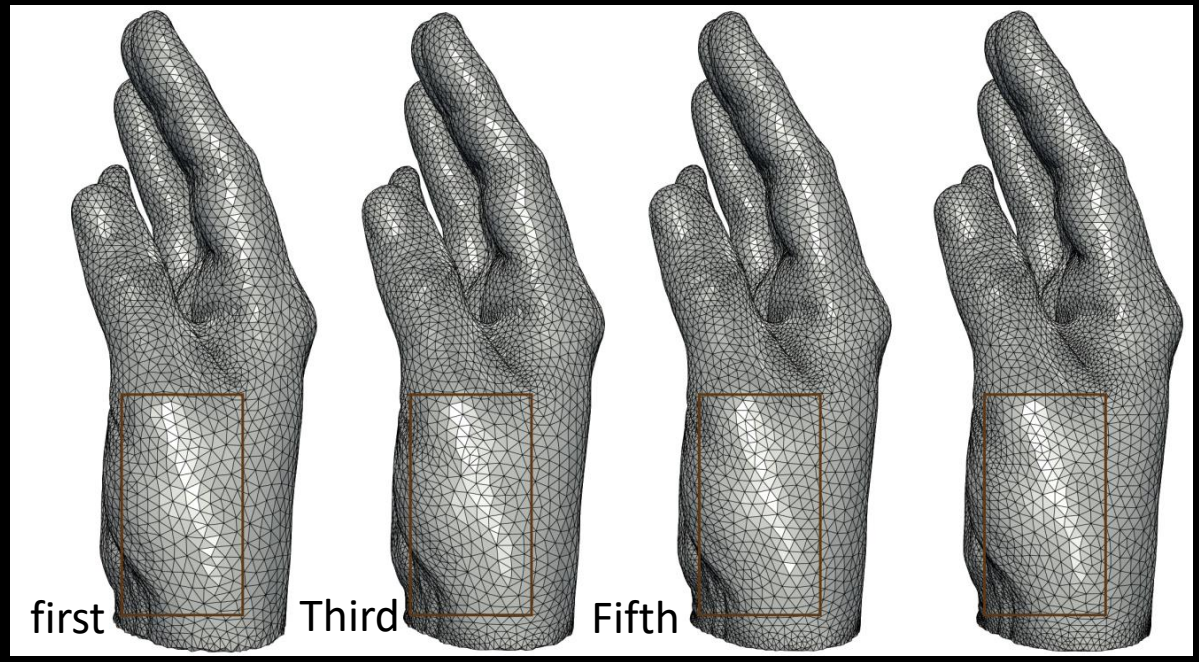


Thinking from a different view

- A key observation: more uniformly distributed vertices are used for remeshing, the Hausdorff distance between the remeshed and input meshes is usually easily bounded.
- During the remeshing process, intermediate meshes are allowed to violate the error-bounded constraint, and attempt to reduce the Hausdorff distance by adding vertices into the mesh.

Algorithm

- 1. Initialize a target edge length field $L(x)$
- 2. Perform edge-based remeshing using $L(x)$
- If $d_h(M_1, M_2) \leq \delta$, stop the algorithm; otherwise, adaptively adjust $L(x)$, and then go to step (2).



Outlines

- Isotropic remeshing
  - Error-bounded method
    - Error-Bounded and Feature Preserving Surface Remeshing with Minimal Angle Improvement
  - Improve small and large angles
  - Optimal Delaunay Triangulation (ODT)
  - Centroidal Patch Triangulation (CPT)
- Anisotropic remeshing
  - Introduction
  - ODT with general convex function
  - Local convex triangulation
  - Partial-based method
  - High-dim embedding

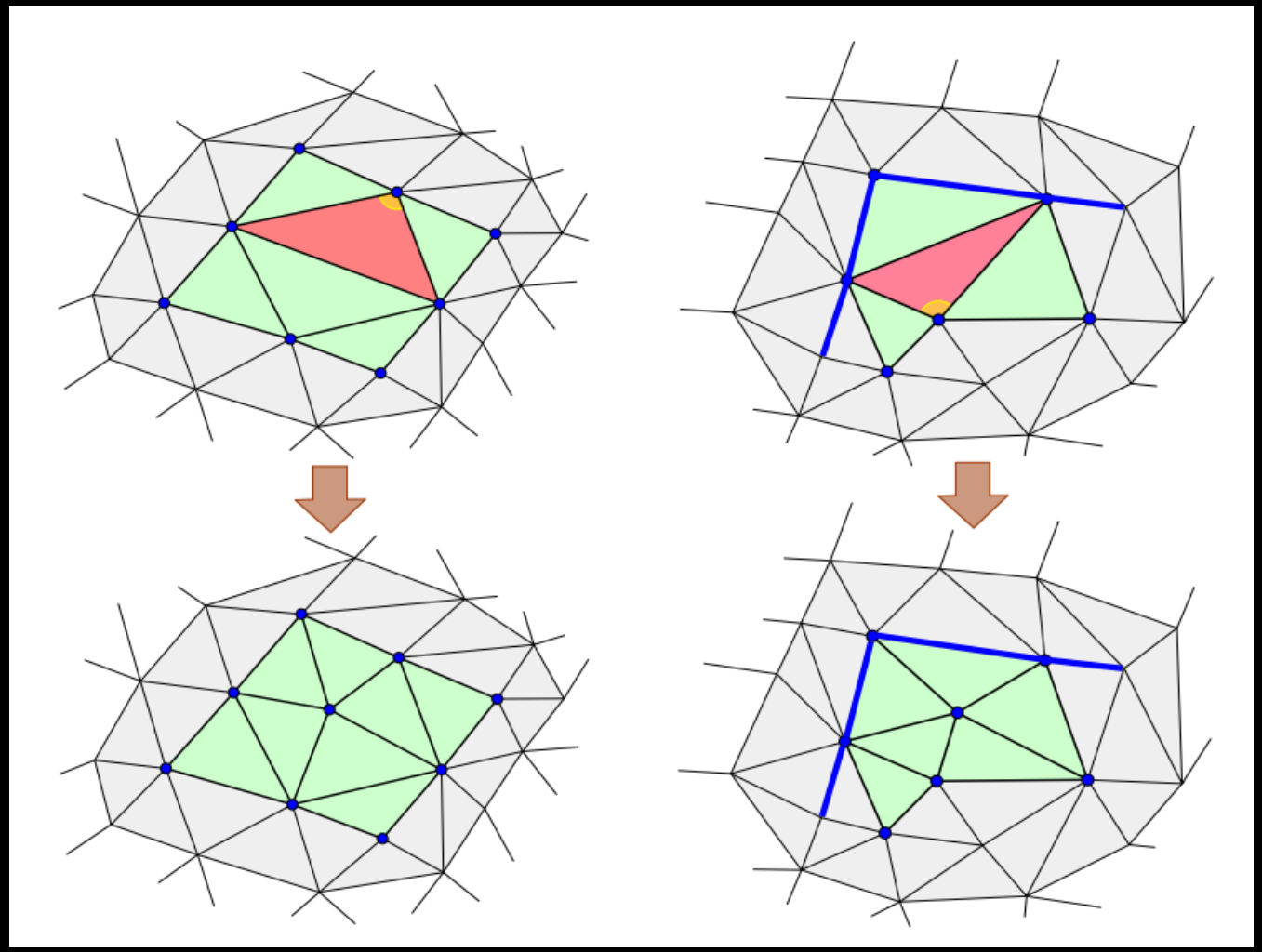
Angles

Isotropic Surface Remeshing without Large and Small Angles

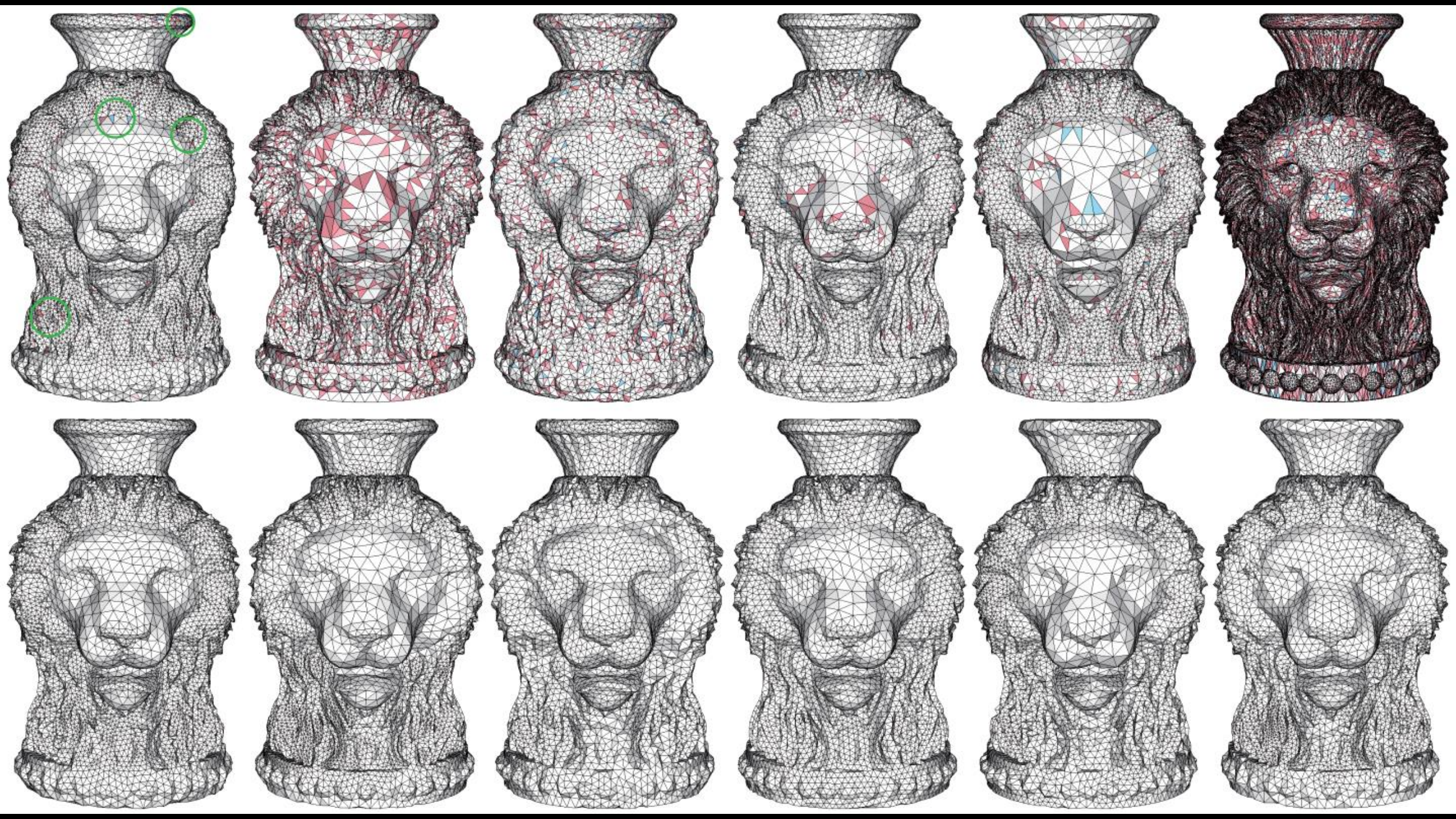
- All the triangles with small or large angles outside the desired bounds $[\beta_{\min}, \beta_{\max}]$ are processed.

- Large angle removal: edge splitting
- Small angle improvement: edge collapsing

Large angle removal



1. Split
2. Flip

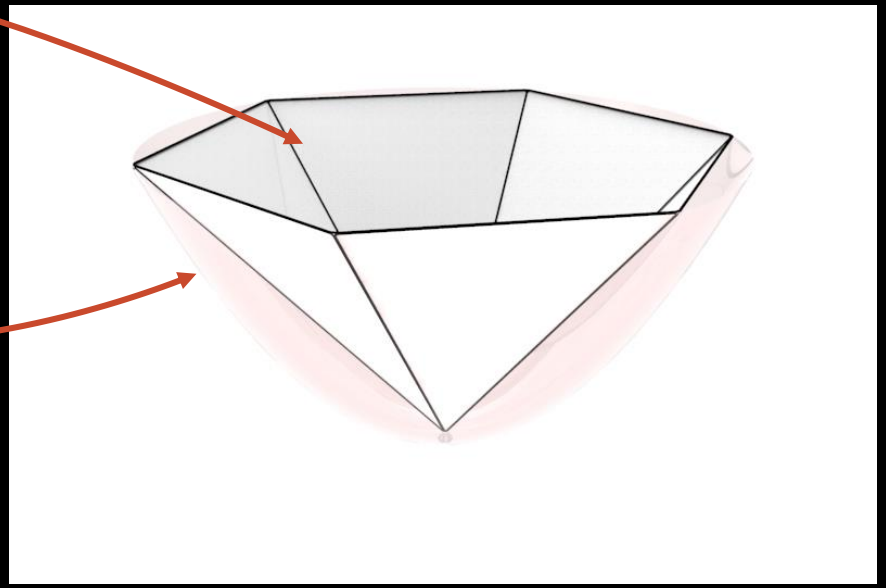


Outlines

- Isotropic remeshing
  - Error-bounded method
    - Error-Bounded and Feature Preserving Surface Remeshing with Minimal Angle Improvement
  - Improve small and large angles
  - Optimal Delaunay Triangulation (ODT)
  - Centroidal Patch Triangulation (CPT)
- Anisotropic remeshing
  - Introduction
  - ODT with general convex function
  - Local convex triangulation
  - Partial-based method
  - High-dim embedding

Optimal Delaunay Triangulation (ODT)

$$E = \sum_{T \in \mathcal{T}} \int_T |\hat{u}(x) - u(x)| \, dx$$



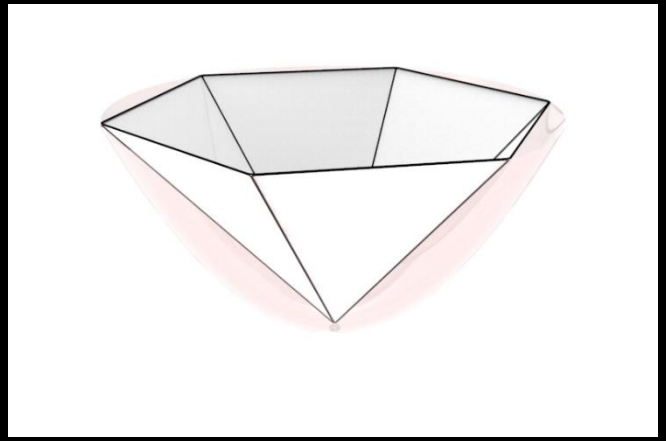
$u(x)$: $z = x^2 + y^2$

$\hat{u}(x)$: piecewise linear interpolation of u

$\mathcal{T}$: a triangulation

Fix positions of vertices, Delaunay triangulation is optimal.

Update of vertices' positions



- Fix the triangulation, update the vertices.

$$E = \sum_{T\in\mathcal{T}} \int_T |\hat{u}(x) - u(x)|\, d\boldsymbol{x} = \sum_{T\in\mathcal{T}} \int_T \hat{u}(x)\, d\boldsymbol{x} + C$$

$$= \sum_{T\in\mathcal{T}} \frac{|T|}{3}\left(u(p_i) + u\left(p_j\right) + u(p_k)\right) + C$$

$$\nabla E_{p_i} = \sum_{T\in\Omega(i)} \frac{\nabla|T|}{3}\left(u(p_i) + u\left(p_j\right) + u(p_k)\right) + \frac{|\Omega|}{3}\nabla u(p_i) = 0$$

Because $\sum_{T\in\Omega(i)} \frac{\nabla|T|}{3} u(p_i) = 0$

$$\nabla u(p_i) = -\frac{1}{|\Omega|}\sum_{T\in\Omega(i)} \frac{\nabla|T|}{3}\left(u\left(p_j\right) + u(p_k)\right)$$

Optimal position

- Since $u = x^2 + y^2$

$$p_i^\star = -\frac{1}{|\Omega|}\sum_{T\in\Omega(i)}\frac{\nabla|T|}{6}\left(\|p_j\|^2 + \|p_k\|^2\right)$$

If $u = x^2 + y^2 = \|x\|^2 \rightarrow u = \|x - p_i\|^2$, it does not change the interpolation error, leading to the same optimal position.

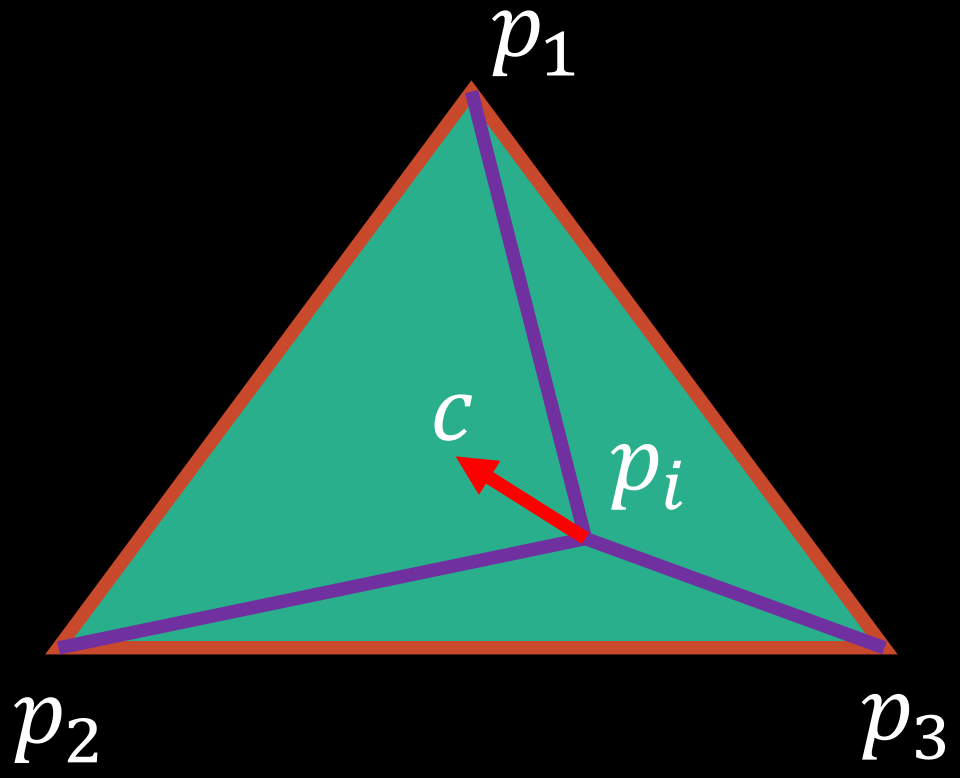
$$p_i^\star = p_i - \frac{1}{|\Omega|}\sum_{T\in\Omega(i)}\frac{\nabla|T|}{6}\left(\|p_j - p_i\|^2 + \|p_k - p_i\|^2\right)$$

Weighted circumcenter

- Since $\sum_{T\in\Omega(i)} \frac{\nabla|T|}{6} = 0$, if $\|p_j - p_i\|^2$ are equal, then $p_i^\star = p_i$.

- For the right case, the optimal position of p_i is the circumcenter c of $\Delta p_1 p_2 p_3$.

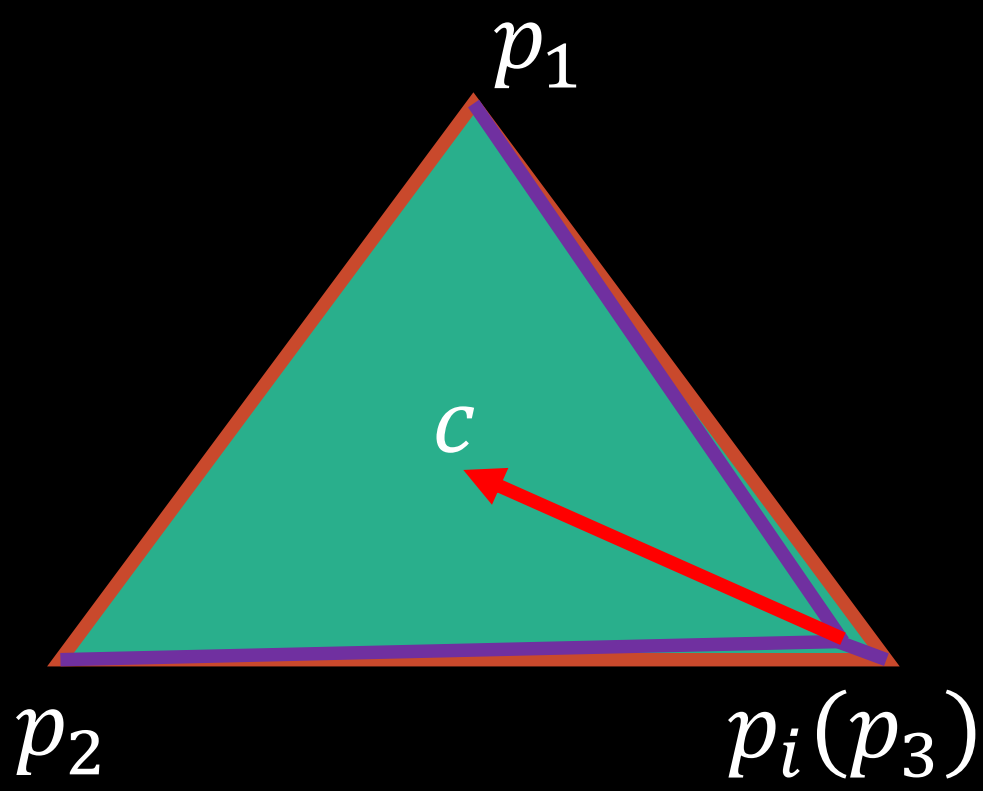
$$
\begin{aligned}
& c \\
& = p_i \\
& - \frac{1}{|\Omega|}\left(\frac{\nabla|\Delta p_1 p_i p_3|}{6}(\|p_1 - p_i\|^2 + \|p_3 - p_i\|^2)\right. \\
& \left. + \frac{\nabla|\Delta p_1 p_2 p_i|}{6}(\|p_1 - p_i\|^2 + \|p_2 - p_i\|^2)\right)
\end{aligned}
$$



Weighted circumcenter

- A special case: $p_i = p_3$

$$c = p_3 - \frac{1}{|\Delta p_1 p_2 p_3|}\left(\frac{\nabla|\Delta p_1 p_i p_3|}{6}(\|p_1 - p_3\|^2) + \frac{\nabla|\Delta p_1 p_2 p_3|}{6}(\|p_1 - p_3\|^2 + \|p_2 - p_3\|^2)\right)$$



Weighted circumcenter

- Taking the one-ring of p_3

$$\sum_{T_j \in \Omega(p_3)} |T_j| c_j$$

$$= \sum_{T_j \in \Omega(p_3)} |T_j| p_3$$

$$- \sum_{T_j \in \Omega(p_3)} \left(\frac{\nabla |\Delta p_1 p_i p_3|}{6} (\|p_1 - p_3\|^2) + \frac{\nabla |\Delta p_1 p_2 p_3|}{6} (\|p_1 - p_3\|^2 + \|p_2 - p_3\|^2) \right.$$

Centroidal Voronoi tessellations (CVT)

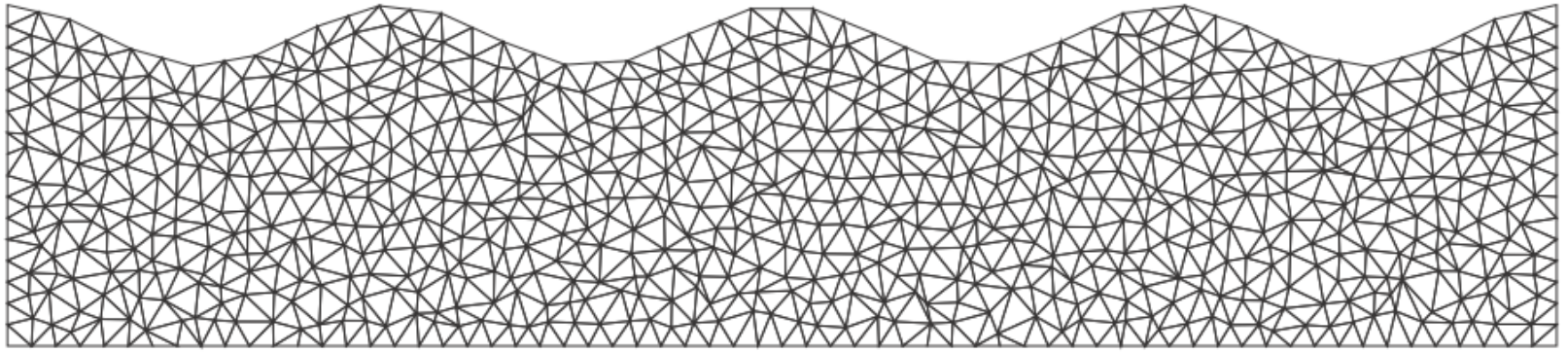
$$E = \sum_{i=1}^{n} \int_{V_i} \|x - p_i\|^2 \, dx$$ CVT

$$E = \sum_{i=1}^{n} \int_{\Omega(p_i)} \|x - p_i\|^2 \, dx$$ CPT

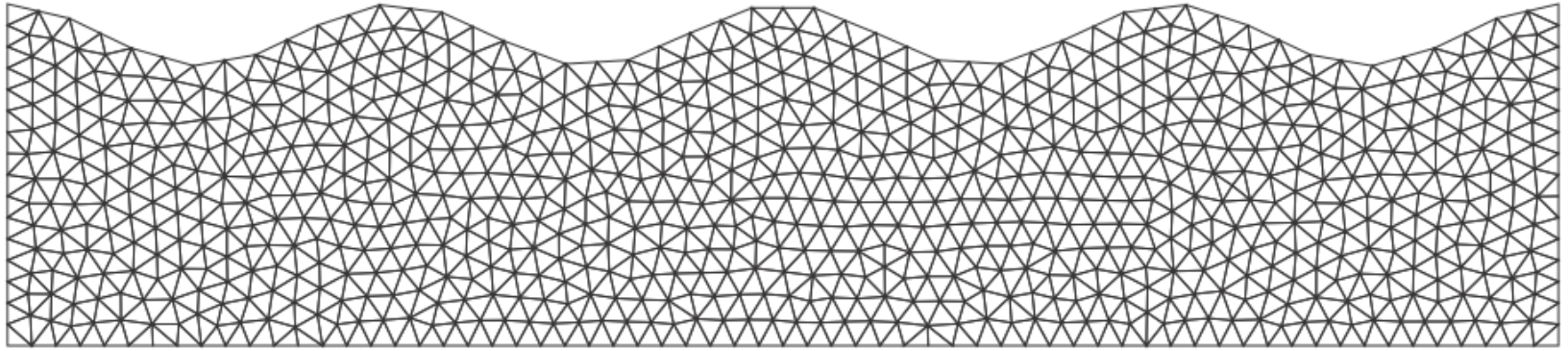
Centroidal Patch Triangulation (CPT)

- Delaunay triangulation
- Moving p_i to the centroid c_i of the corresponding patch.

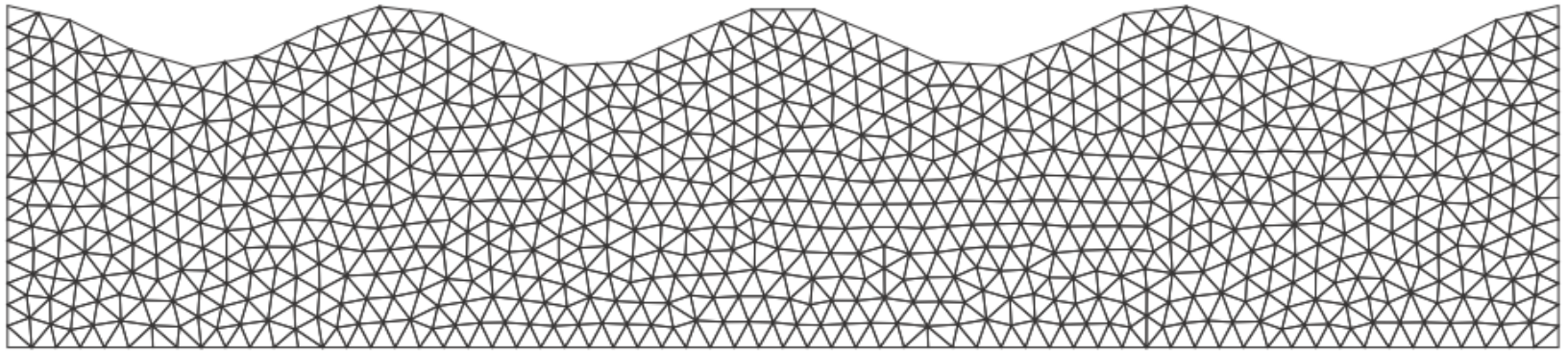
$$E = \sum_{i=1}^{n} \int_{\Omega(p_i)} \|x - p_i\|^2 \, dx$$



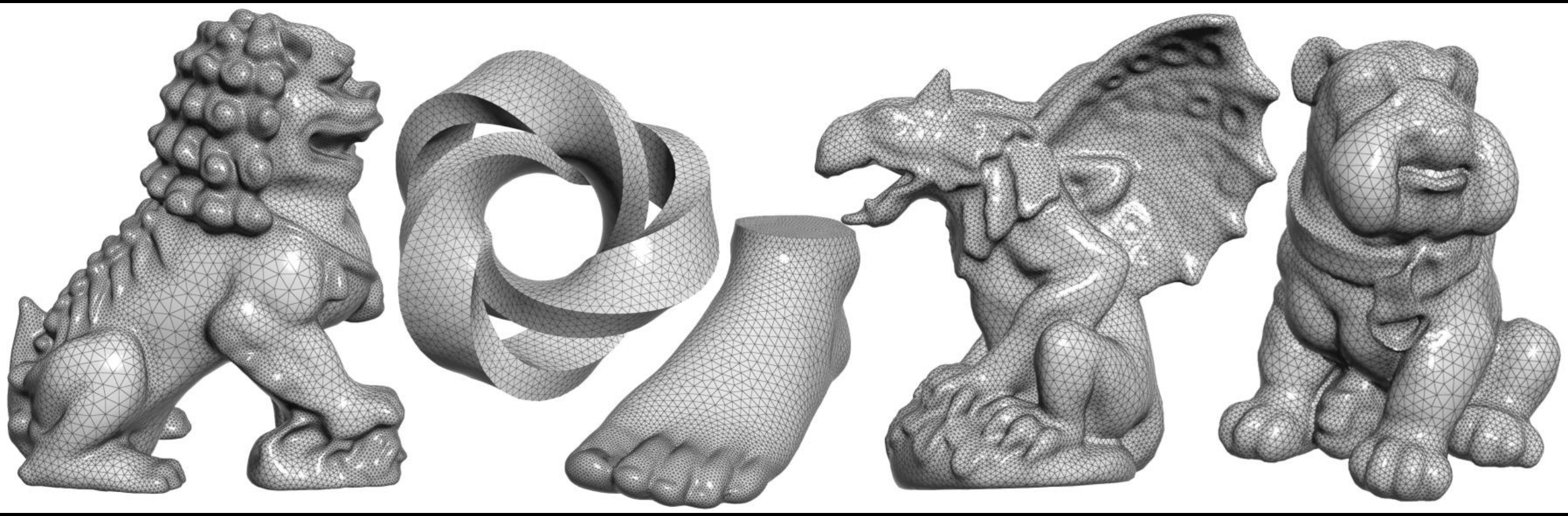
(a) Original mesh: min(q) = 0.141, mean(q) = 0.875, uniformity = 15.62%



(b) ODT smoothing: min(q) = 0.709, mean(q) = 0.964, uniformity = 5.16%



(c) CPT smoothing: min(q) = 0.708, mean(q) = 0.967, uniformity = 6.15%



Simulated annealing method with the operation “perturb-optimize”

Outlines

- Isotropic remeshing
  - Error-bounded method
    - Error-Bounded and Feature Preserving Surface Remeshing with Minimal Angle Improvement
  - Improve small and large angles
  - Optimal Delaunay Triangulation (ODT)
  - Centroidal Patch Triangulation (CPT)
- Anisotropic remeshing
  - Introduction
  - ODT with general convex function
  - Local convex triangulation
  - Partial-based method
  - High-dim embedding

Metric

- A metric on a set X is a function (called the distance function or simply distance)

$$d: X \times X \to [0, \infty)$$

where $[0, \infty)$ is the set of non-negative real numbers.

- For all $x, y, z \in X$, the following conditions are satisfied:
  - ✓ Non-negativity or separation axiom
    - ✓ $d(x, y) \geq 0$
  - ✓ Identity of indiscernibles
    - ✓ $d(x, y) = 0 \Leftrightarrow x = y$
  - ✓ Symmetry
    - ✓ $d(x, y) = d(y, x)$
  - ✓ Subadditivity or triangle inequality
    - ✓ $d(x, z) \leq d(x, y) + d(y, z)$

Metric

- Conditions 1 and 2 together define a positive-definite function.
- The first condition is implied by the others.

- In practice, the metric can be represented by a positive-definite symmetric $m \times m$ matrix $M(x)$.
  - $M(x) = Q(x)^T Q(x)$.

- Given a $M(x)$, its decomposition to $Q(x)$ is non-unique.

Length

- Given the metric field $M(x)$ and an open curve $C \subset \Omega$, the length of C is defined as the integration of the length of tangent vector along the curve C with metric $M(x)$

- The anisotropic distance $d_M(x, y)$ between two points x and y can be defined as the length of the (possibly non-unique) shortest curve (assuming line segment) that connects x and y.

$$\int_0^1 \sqrt{(x-y)^T M(tx+(1-t)y)(x-y)}dt$$

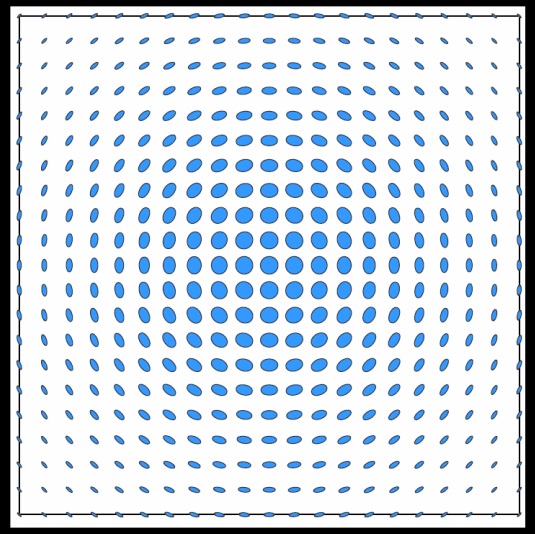
Anisotropic remeshing

- Input:
  - Domain: $\Omega \in \mathbb{R}^d$
  - Metric field: $\boldsymbol{M}(\boldsymbol{x}),\ \boldsymbol{x} \in \Omega$
    - $d \times d$ positive-definite matrix

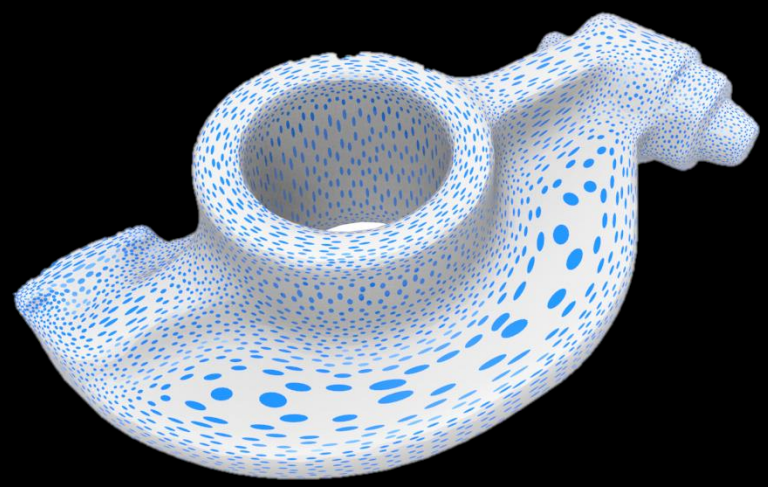
- Isotropic remeshing
  - All edge lengths are as equal as possible.

- Anisotropic remeshing
  - All edge lengths with metric are as equal as possible.

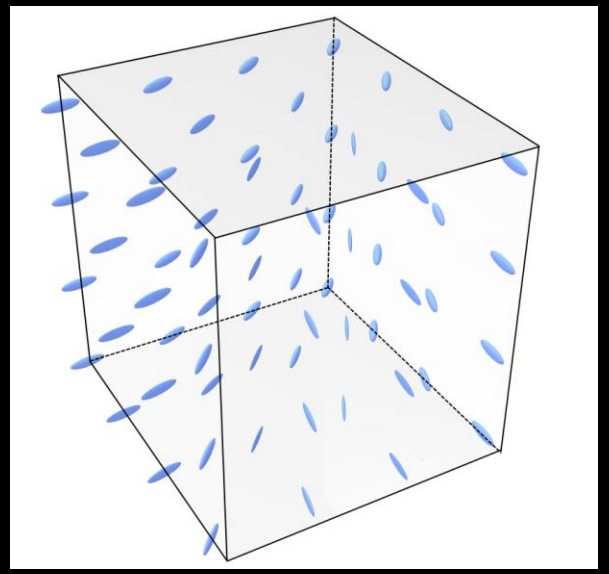
Input examples



Ω = 2D square,
$M(p)$ = Hessian of given u



Ω = 3D surface,
$M(p)$ = mesh curvature



Ω = 3D cube,
$M(p)$ = given tensor field

Anisotropic remeshing

- Eigen-decomposition: $M(x) = U(x)\Lambda U(x)^T$
- Transformation $\Phi = \Lambda^{1/2} U(x)^T$
- The quality metrics are measured in the transformed space.

transforms simplex to isotropic space

ODT properties *[Chen 2004, Liu et al. 2013, Desbrun et al. 2013]*

$$\min_T E_{ODT} \triangleq \sum_{\tau \in T} \int_{\tau} |\hat{u}(\boldsymbol{x}) - u(\boldsymbol{x})| \, d\boldsymbol{x}$$

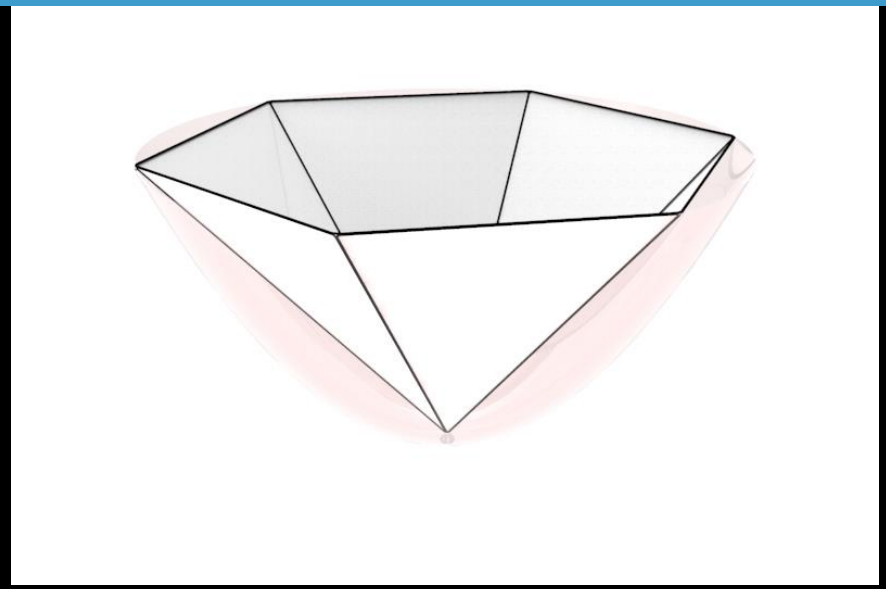
- $u(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^2 \Rightarrow$

  T_{ODT} = Delaunay triangulation

- $u(\boldsymbol{x})$ convex $\Rightarrow$

  T_{ODT} = regular triangulation

  power weight: $w(\boldsymbol{x}) = \boldsymbol{x}^2 - 2\,u(\boldsymbol{x})$



ODT method

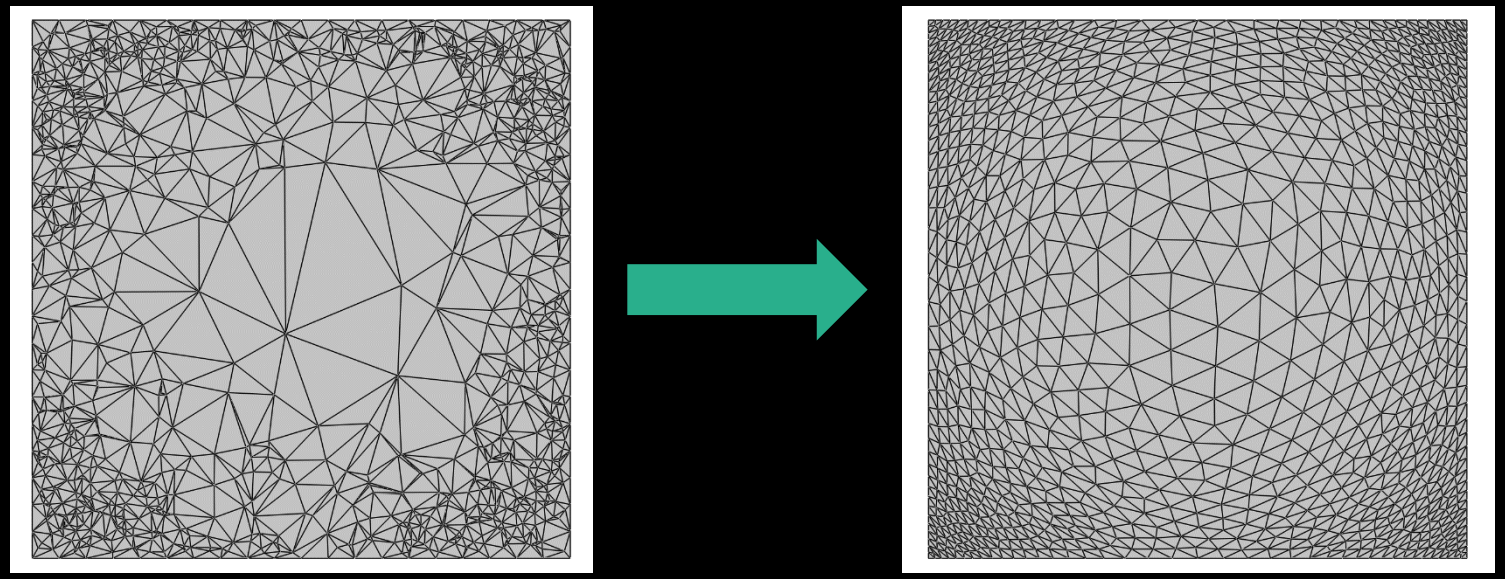
power weight:
$$w(\boldsymbol{x}) = \boldsymbol{x}^2 - 2\,u(\boldsymbol{x})$$

Iterative mesh optimization:

- update vertices: *move each vertex to its power-weighted circumcenter*
- update connectivity: *compute constrained regular triangulation*

$$u(x,y) = \mathrm{e}^{\frac{(x^2+y^2)}{10}}$$

$$\Omega = [-5,5]^2$$



Thinking from optimization

$$\min_{T} E_{ODT} \triangleq \sum_{\tau \in \mathrm{T}} \int_{\tau} |\hat{u}(\boldsymbol{x}) - u(\boldsymbol{x})| \, d\boldsymbol{x}$$

1. Update connectivity
2. Update vertex positions

ODT limitations

- $\boldsymbol{M}$ must be Hessian of a global u
- u must be convex

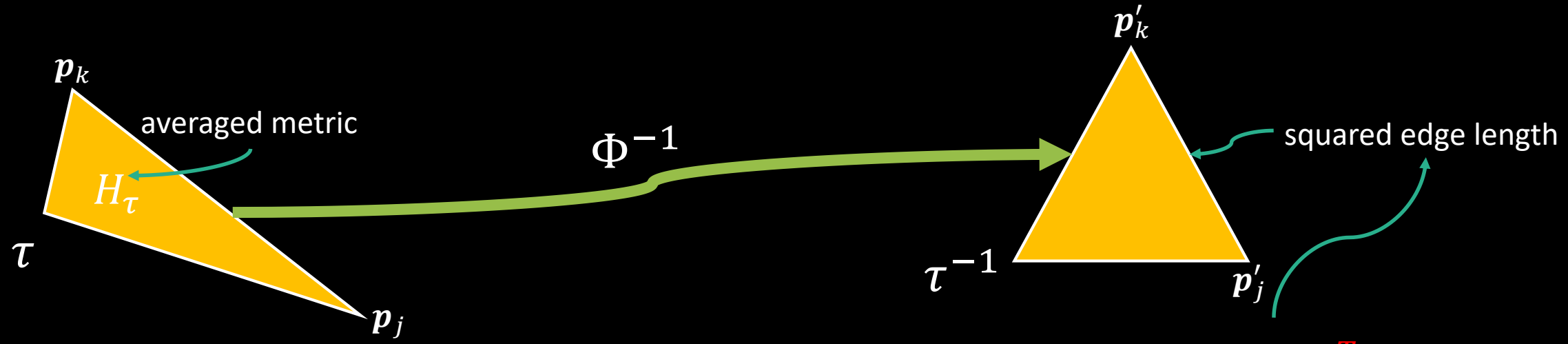
$\Rightarrow$ neither true for general $\boldsymbol{M}$!

Locally Convex Triangulation (LCT)

Anisotropic Simplicial Meshing Using Local Convex Functions

Anisotropy approximated ***locally*** by per-simplex convex function:

$u(\boldsymbol{x}) \rightarrow u_\tau(\boldsymbol{x})$.



$$u_\tau(\boldsymbol{x}) \triangleq \frac{1}{2}\boldsymbol{x}^T H_\tau \boldsymbol{x} \longrightarrow E_\tau \triangleq \int_\tau |\hat{u}_\tau - u_\tau| \, d\boldsymbol{x} \longrightarrow E_\tau = \frac{|\tau| \sum_{j<k} (\boldsymbol{p}_j - \boldsymbol{p}_k)^T H_\tau (\boldsymbol{p}_j - \boldsymbol{p}_k)}{2(d+1)(d+2)}$$

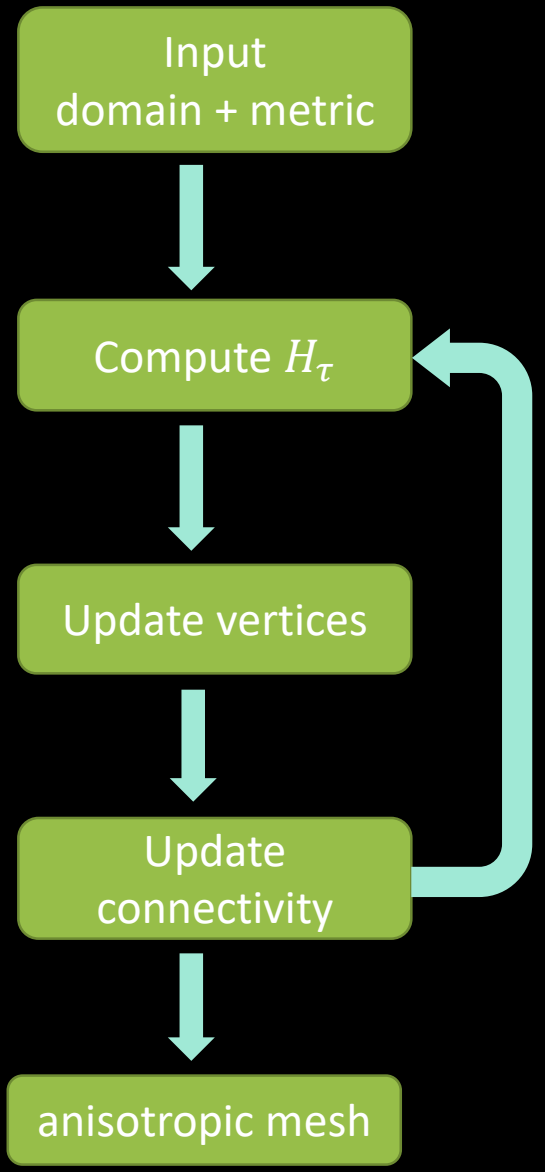
LCT Optimization

$$\min_{T} E_{LCT} \triangleq \sum_{\tau \in \mathrm{T}} E_\tau = \sum_{\tau \in \mathrm{T}} \int_\tau |\hat{u}_\tau(x) - u_\tau(x)| \, d\boldsymbol{x}$$

Step 1: compute H_τ on each simplex

Step 2: fix connectivity, update vertices

Step 3: fix vertices, update connectivity

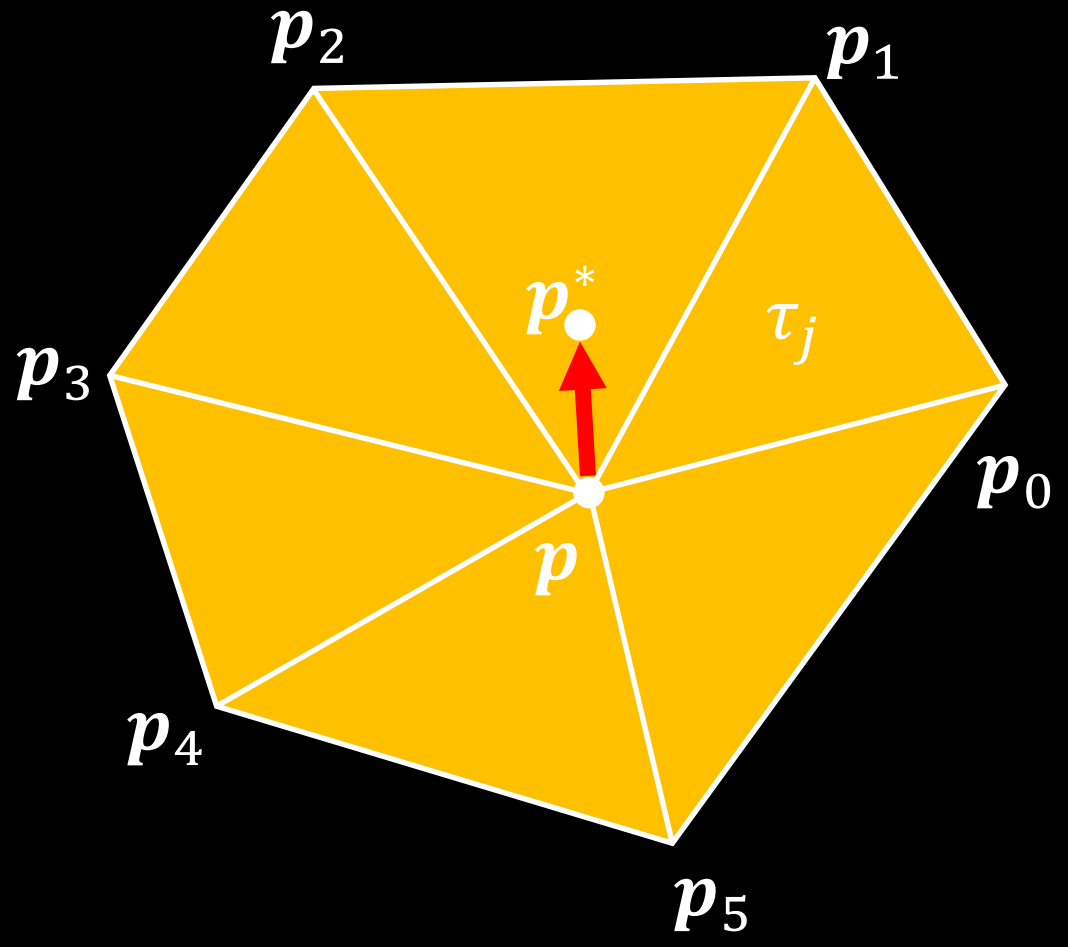


Vertex update

Update each vertex $\boldsymbol{p}$ in sequence:

- one-ring $\tau_j \in \Omega(\boldsymbol{p})$
- energy sum $E_{\boldsymbol{p}} = \sum_j E_{\tau_j}$
- gradient $\boldsymbol{g} = \partial E_p / \partial \boldsymbol{p}$
- Hessian $\boldsymbol{h} = \mathrm{PD}\left(\partial^2 E_p / \partial \boldsymbol{p}^2\right)$
- $\boldsymbol{p}^* \coloneqq \boldsymbol{p} - \alpha\, \boldsymbol{h}^{-1}\, \boldsymbol{g}$

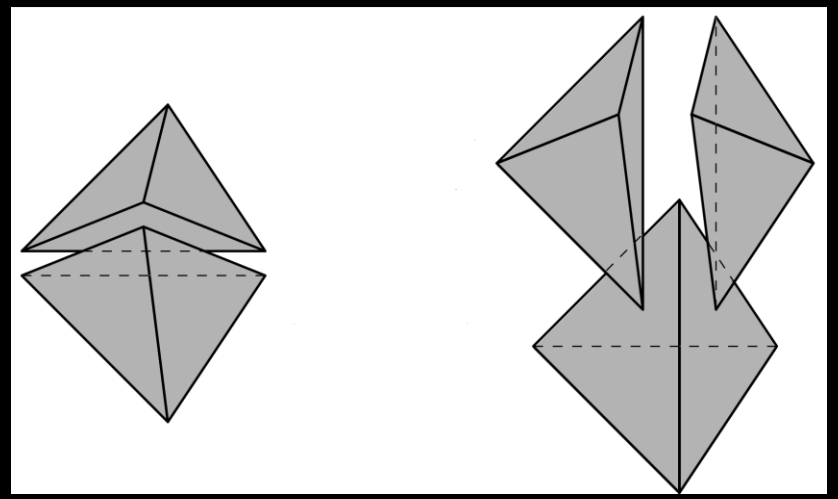
restrict boundary/feature vertices



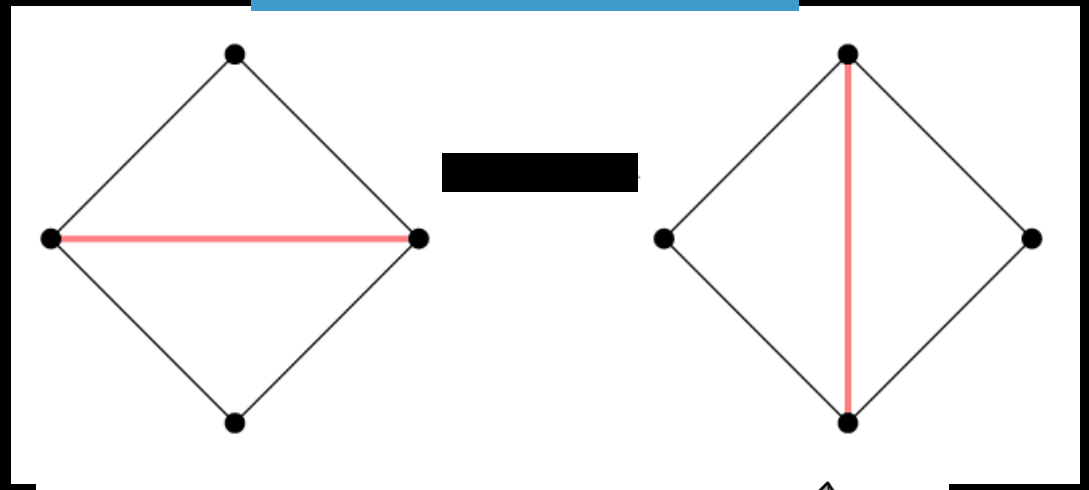
Connectivity update

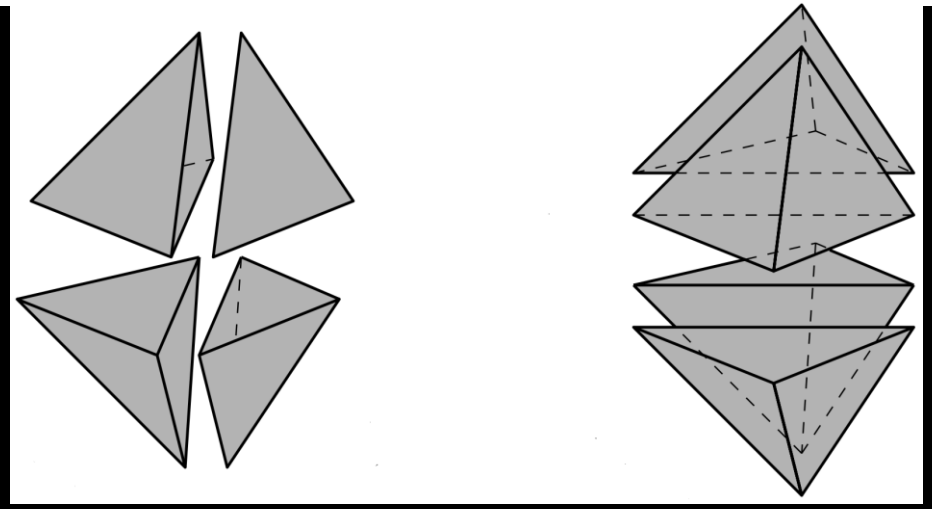
triangle mesh ⇒ edge flip

tetrahedral mesh ⇒ 2-3,
3-2, 4-4, 2-2 flips

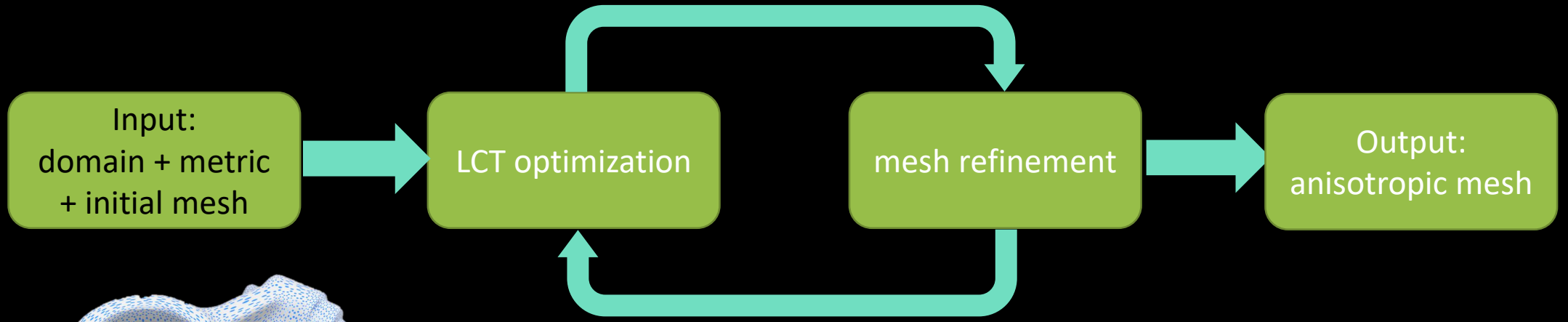


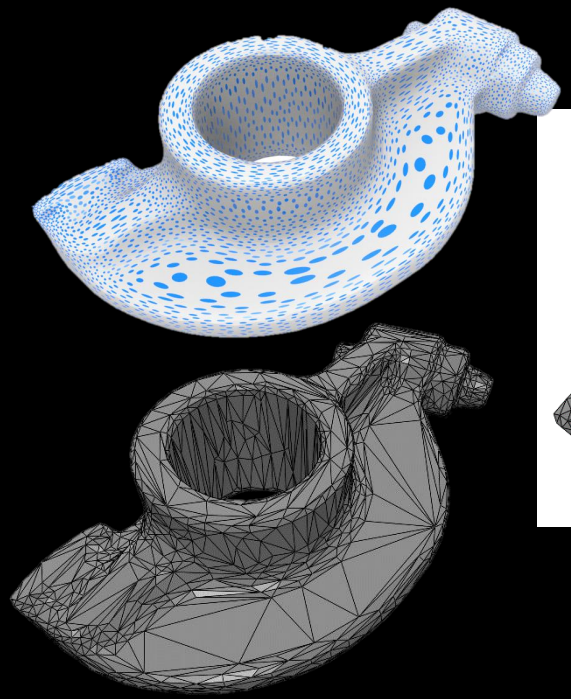
$E_{\text{before}} > E_{\text{after}}$

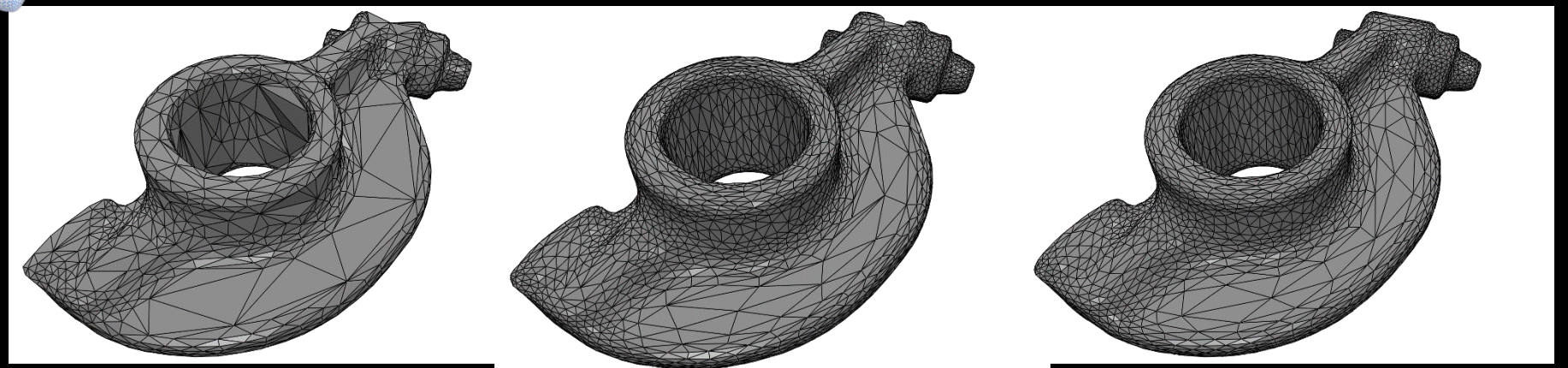




Anisotropic meshing pipeline



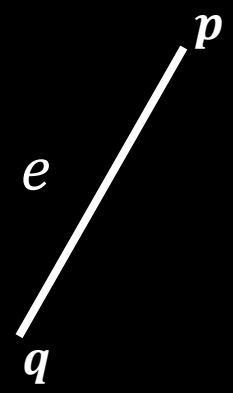




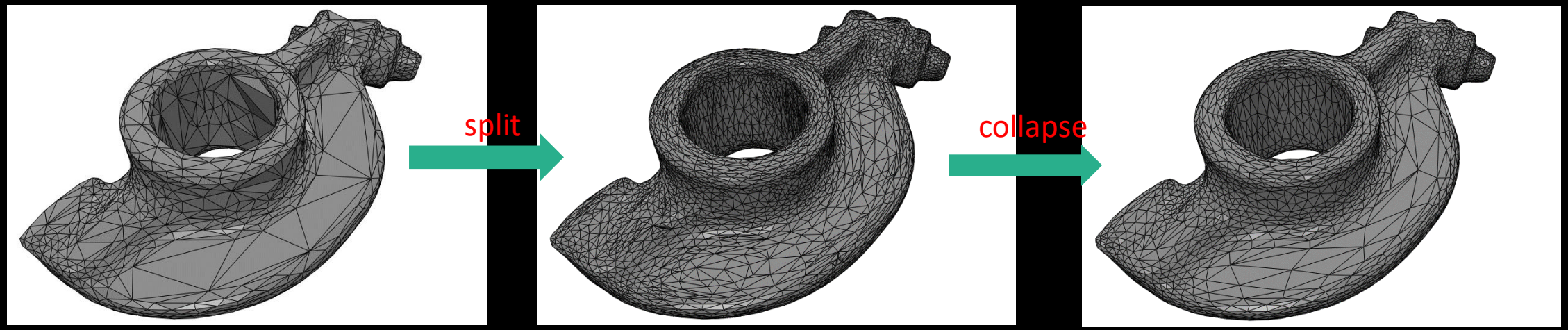
Edge refinement

$$L/\beta \leq |e^{-1}| \leq \beta L$$

- split if $|e^{-1}| > \beta L$
- collapse if $|e^{-1}| < L/\beta$

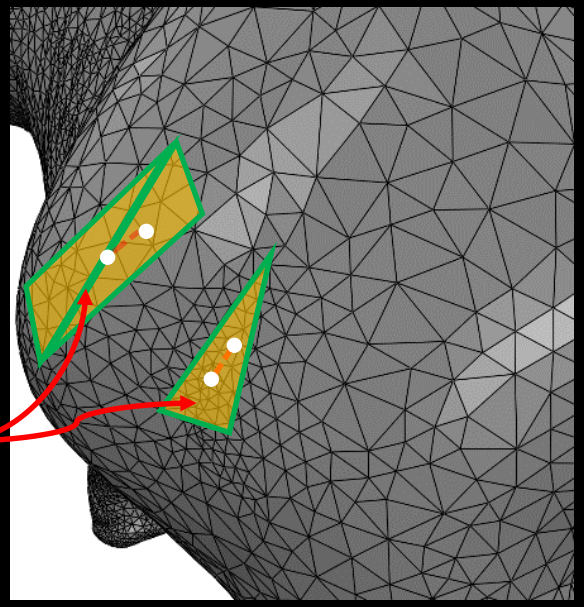


$$|e^{-1}| \approx \sqrt{(\boldsymbol{p}-\boldsymbol{q})^T \frac{\boldsymbol{M}(\boldsymbol{p})+\boldsymbol{M}(\boldsymbol{q})}{2}(\boldsymbol{p}-\boldsymbol{q})}$$

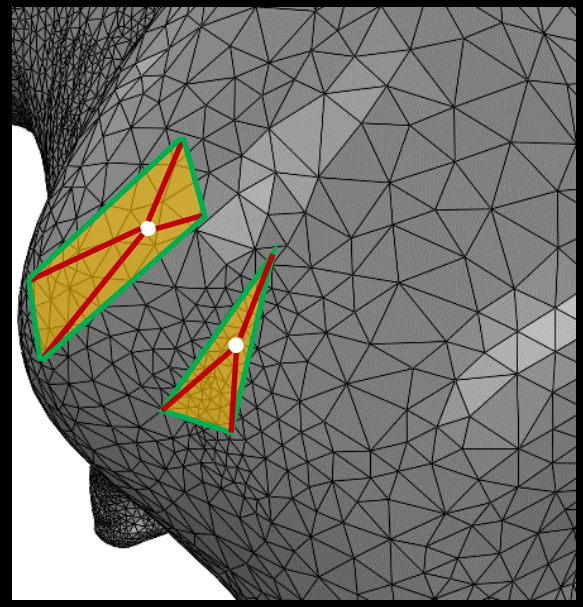


Geometric refinement

query distance

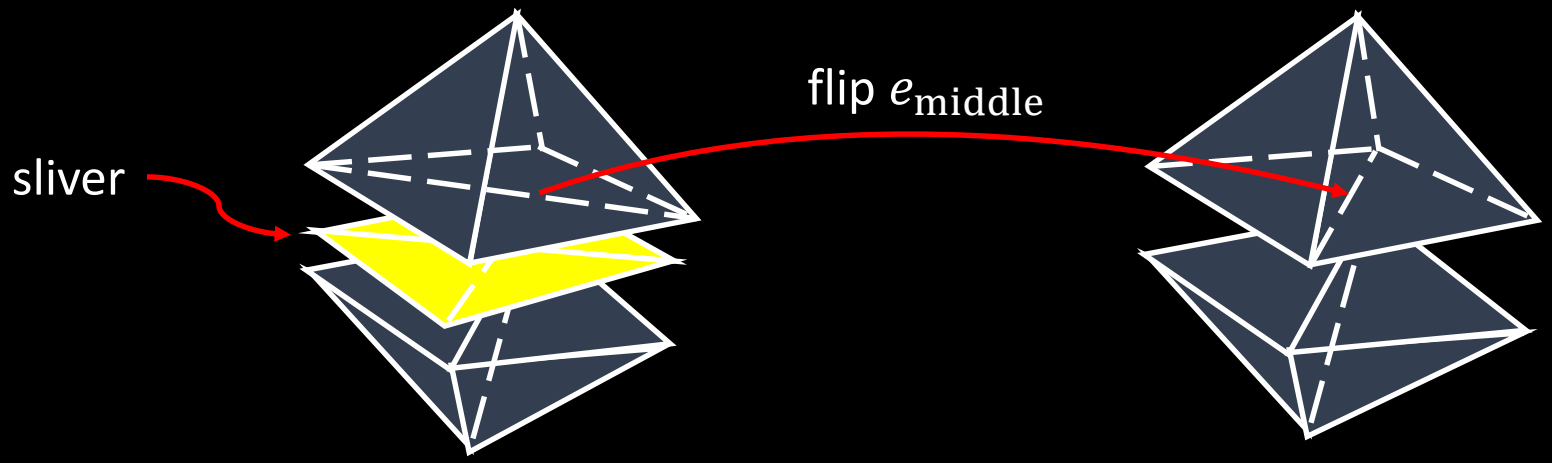


project & split

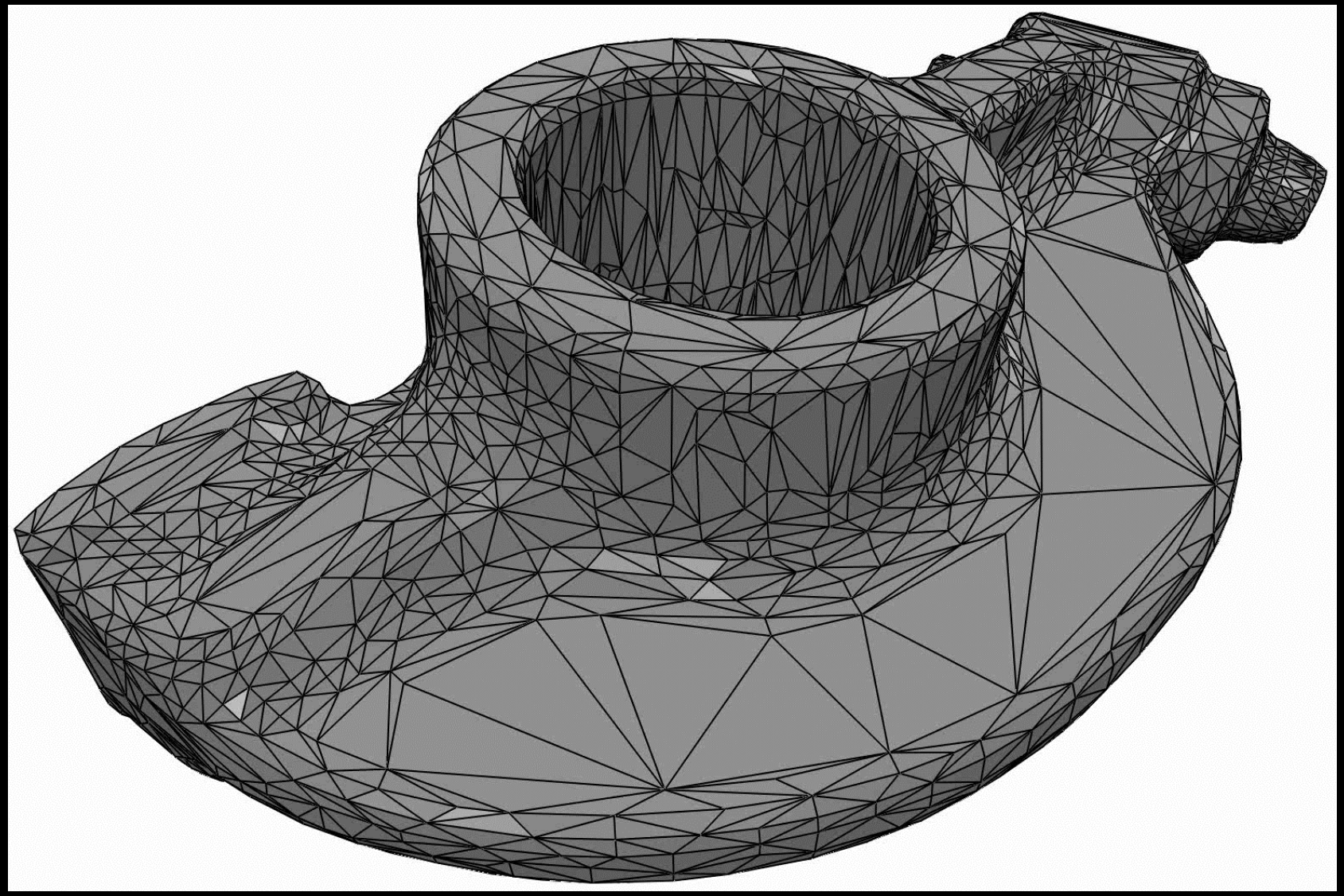


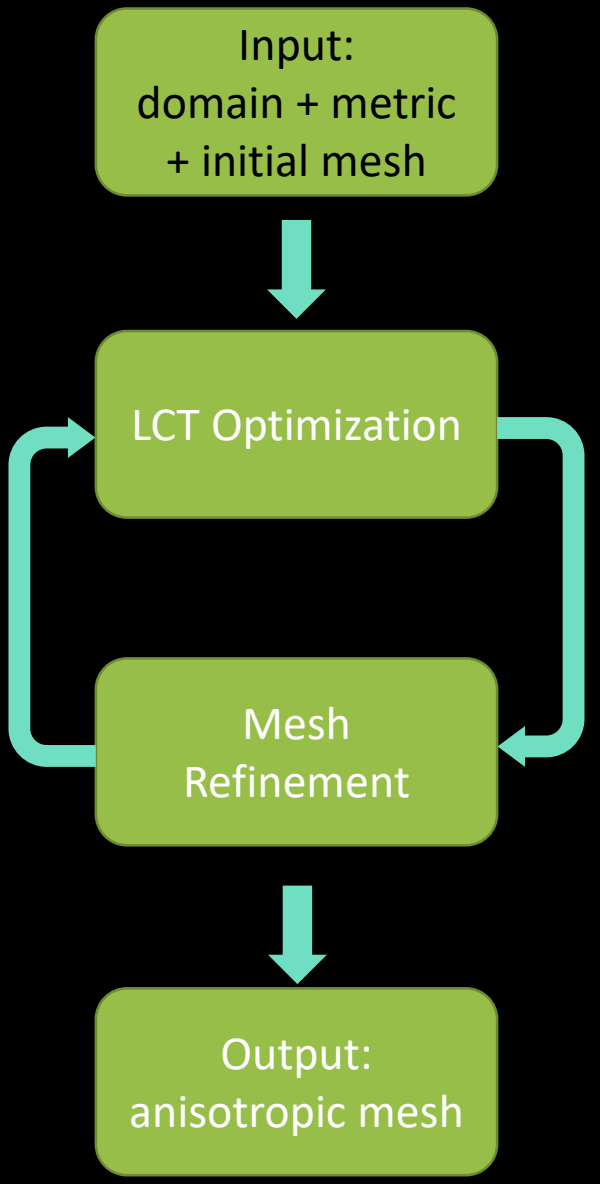
Sliver elimination (tetrahedral mesh)

- perform 5-4 flips
- perturb vertices and perform flips to eliminate small dihedral angles
  [Tournois et al. 2009]

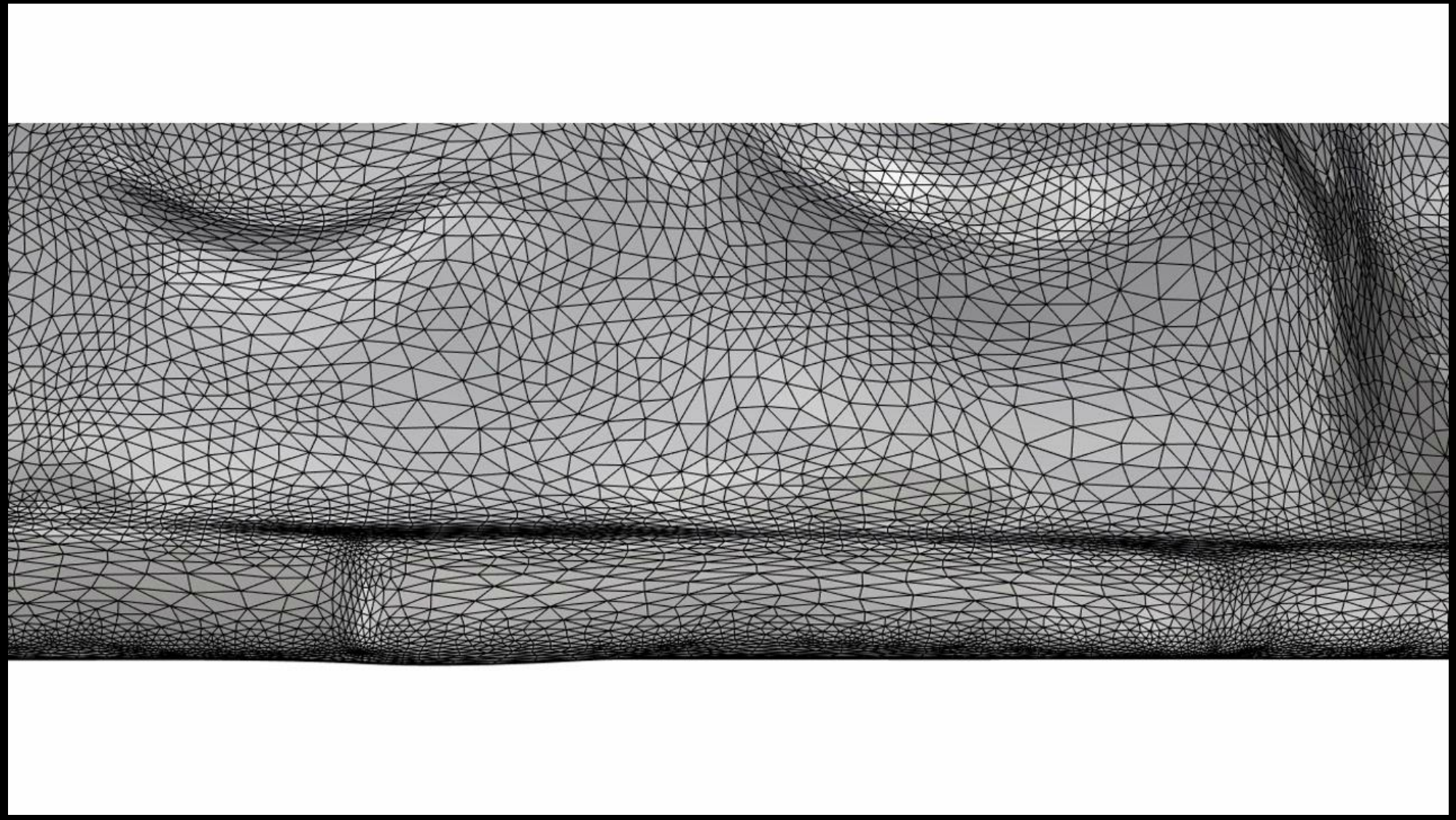


Anisotropic meshing pipeline

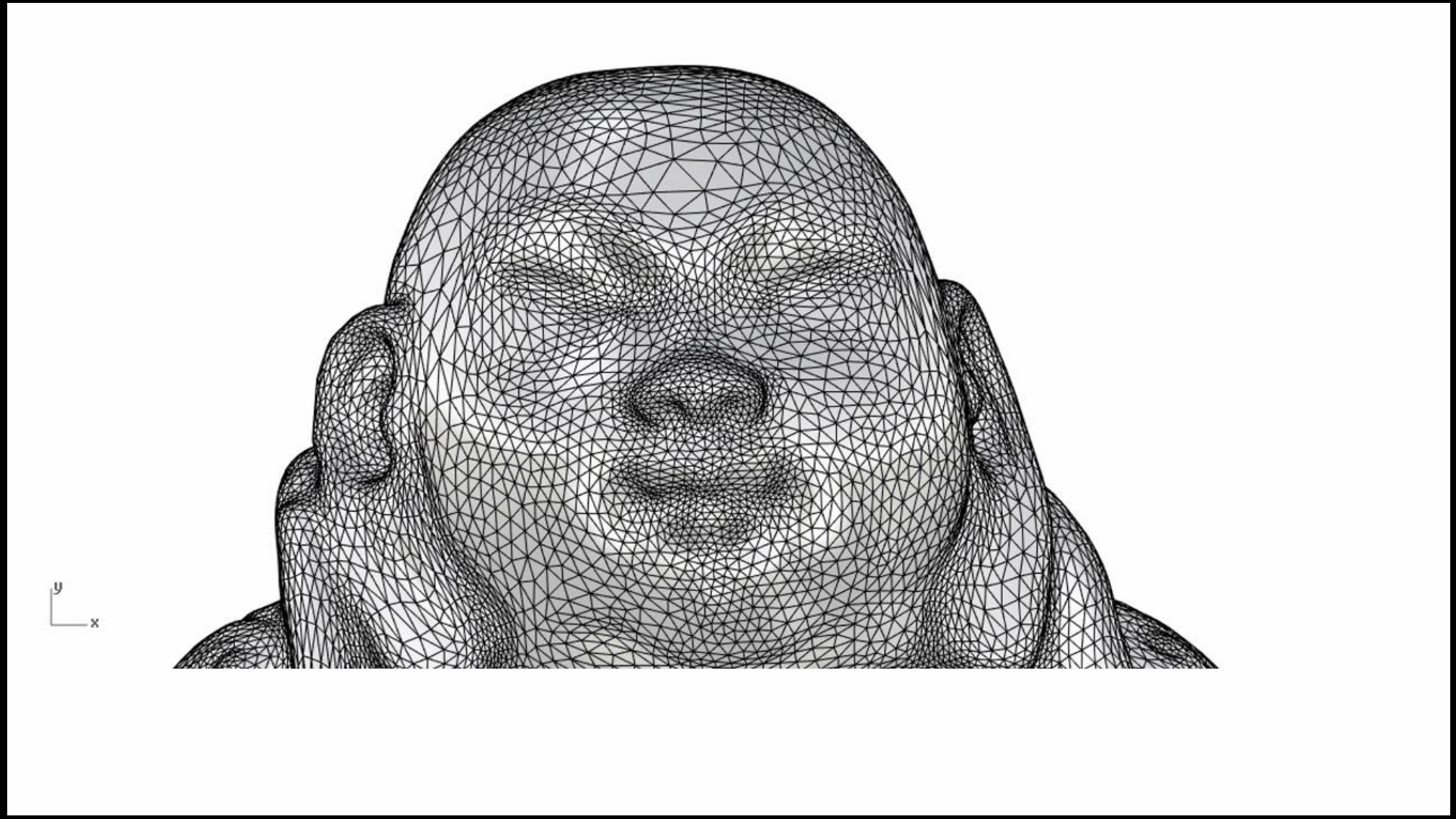




Lucy



Buddha



Partial-based method

Particle-Based Anisotropic Surface Meshing

- Considering each vertex as a particle, the potential energy between the particles determines the inter-particle forces. When the forces applied on each particle become equilibrium, the particles reach the optimal balanced state with uniform distribution.

- Energy:

$$E = \sum_{i=1}^{n} \sum_{j \neq i}^{n} E_{ij}$$

$$E_{ij} = \exp\left(-\left(x_i - x_j\right)^T M_{ij} \left(x_i - x_j\right) / 4\sigma^2\right)$$

Data: a surface Ω with metric $\mathbf{M}$, and the desired number of vertices n
Result: an anisotropic sampling $\mathbf{X}$ of Ω
Initialize particle locations $\mathbf{X}$;
while *stopping condition not satisfied* **do**
    Update the ANN data structure for the current sampling $\mathbf{X}$;
    for *each particle* i **do**
        Get particle i's neighbor $N(i)$ from ANN;
        for *each particle* $j \in N(i)$ **do**
            Compute $\overline{E}^{ij}$ using Eq. (12);
            Compute $\widetilde{\mathbf{F}}^{ij}$ using Eq. (21);
        end
        Sum the total force $\widetilde{\mathbf{F}}^{i}$ using Eq. (22);
        Project $\widetilde{\mathbf{F}}^{i}$ to the surface tangent using Eq. (27);
    end
    Sum the total energy $\overline{E}$ in Eq. (13);
    Run L-BFGS with $\overline{E}$ and $\{\widetilde{\mathbf{F}}^{i}\}$, to get updated locations $\mathbf{X}$;
    Project $\mathbf{X}$ onto the surface;
end

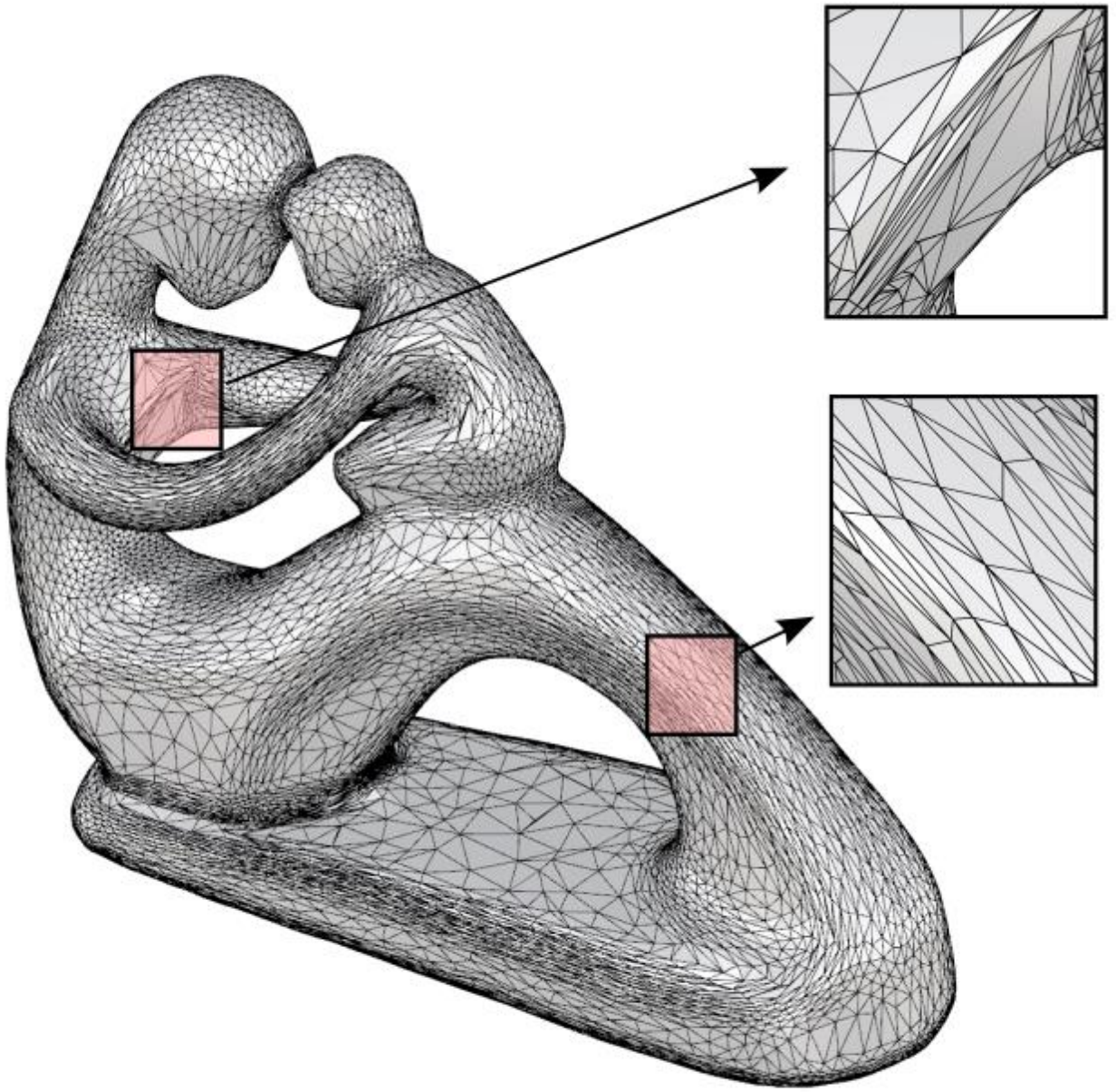
Algorithm 1: Anisotropic Particle Optimization with Metric $\mathbf{M}$.

$$\frac{Q_{ij}(x_i - x_j)}{2\sigma^2} \exp\left(-\frac{(x_i - x_j)^T M_{ij}(x_i - x_j)}{4\sigma^2}\right)$$

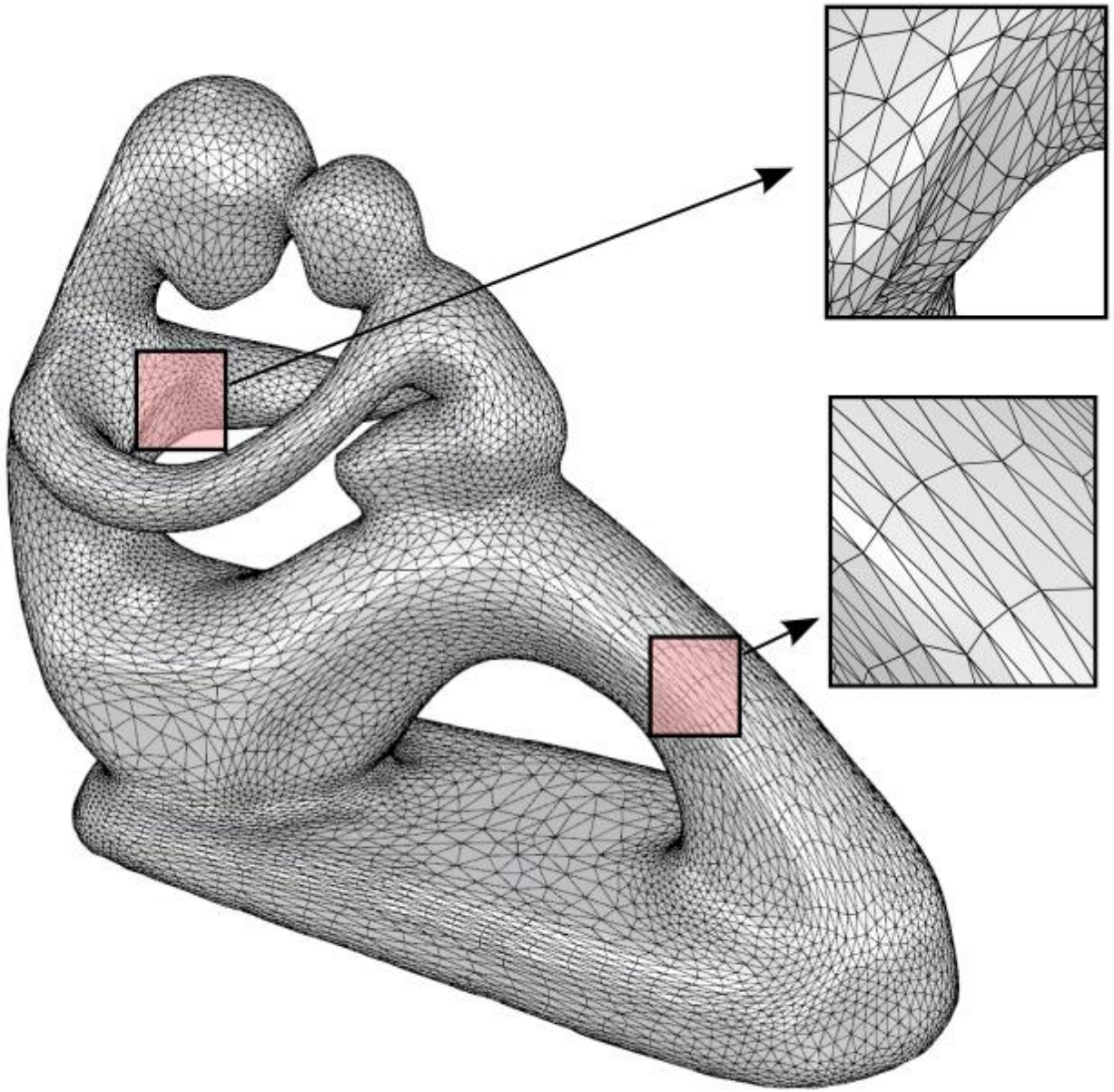
Particle-Based Anisotropic Surface Meshing

Zichun Zhong[1] Xiaohu Guo[1] Wenping Wang[2] Bruno Lévy[3]
Feng Sun[2] Yang Liu[4] Weihua Mao[5]

1 The University of Texas at Dallas
2 The University of Hong Kong
3 INRIA Nancy - Grand Est
4 NVIDIA Corporation
5 UT Southwestern Medical Center at Dallas



particle



LCT