Compact Course for IT Professionals

March 10th, 2006

Efficient Geometric Modeling with Polygonal Meshes

Dr. Mario Botsch

Prof. Dr. Mark Pauly

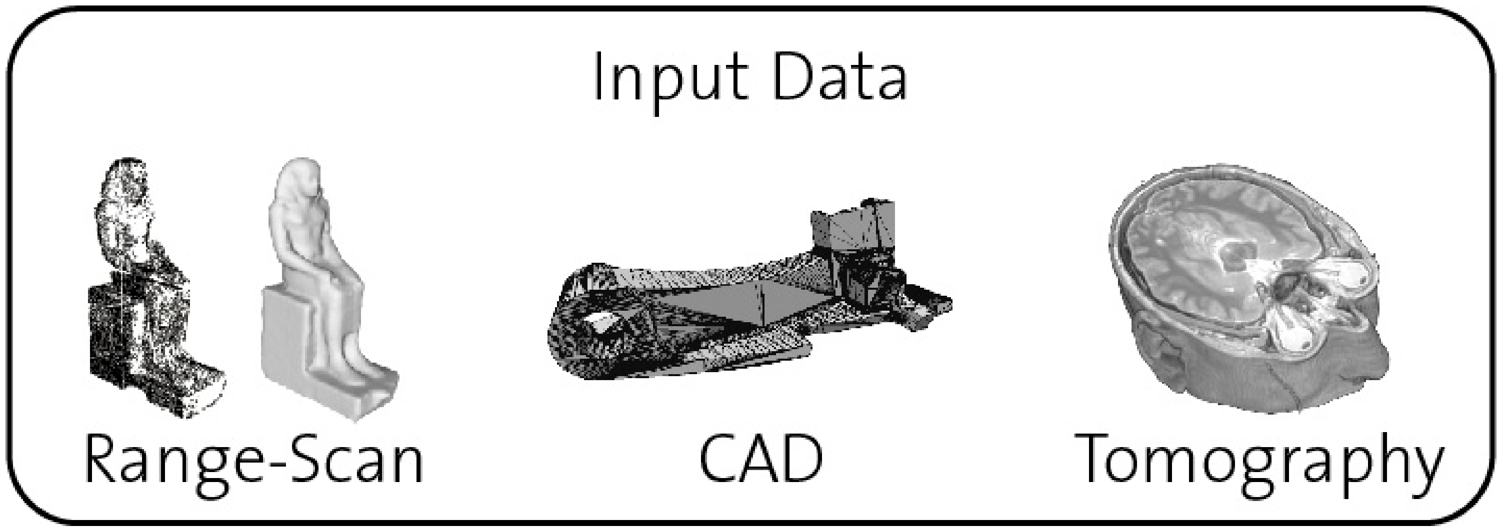
ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

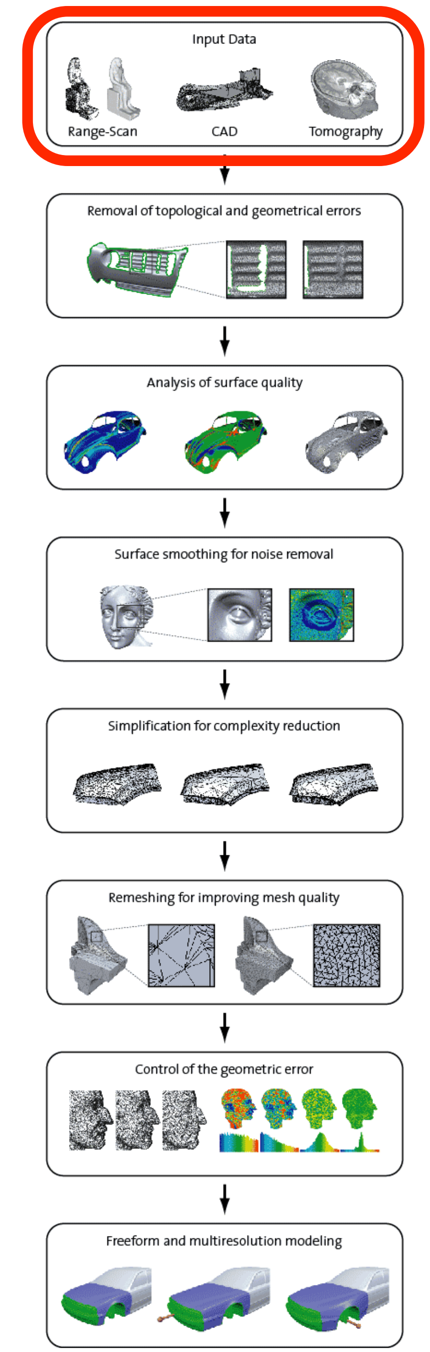
inf | Informatik
Computer Science

Goals

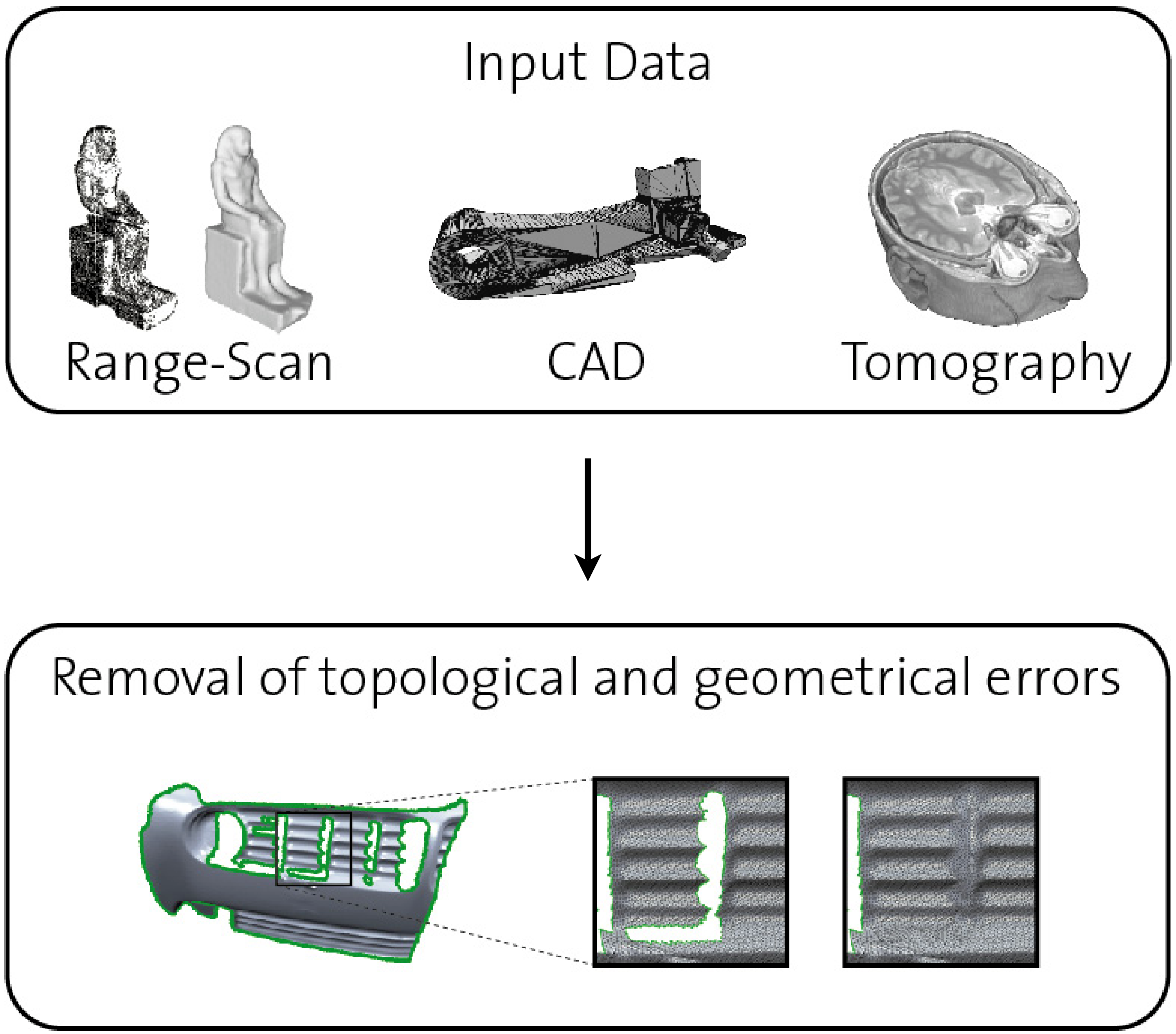
- Present the complete geometry processing pipeline based on triangle meshes
- Focus on fundamental concepts and recent developments
- Provide pointers to relevant source code and literature
- Stimulate new ideas

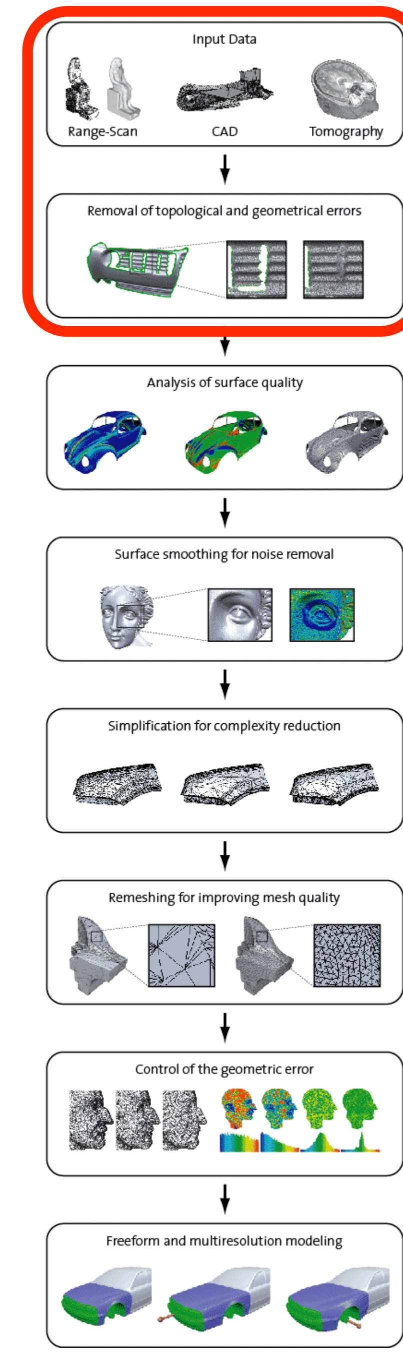
Processing Pipeline





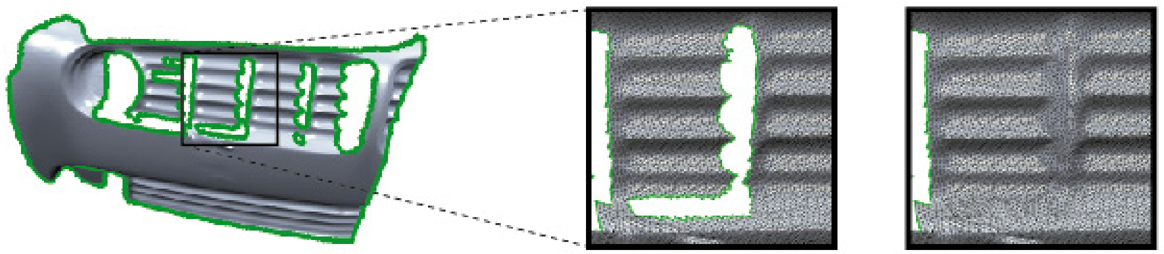
Processing Pipeline





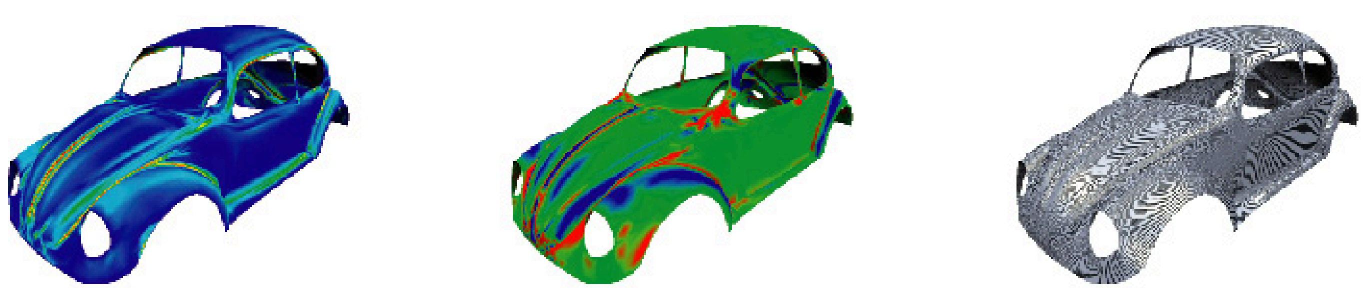
Processing Pipeline

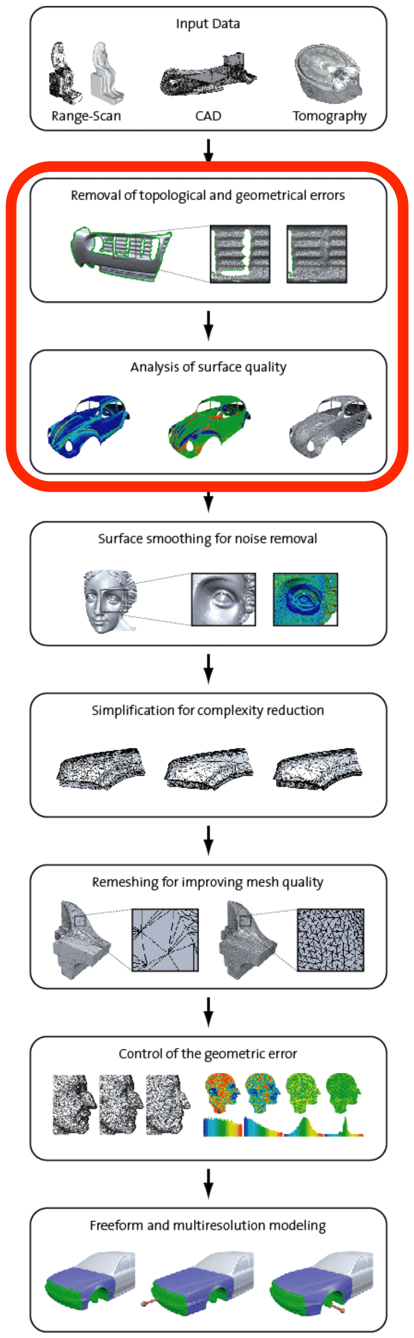
Removal of topological and geometrical errors



↓

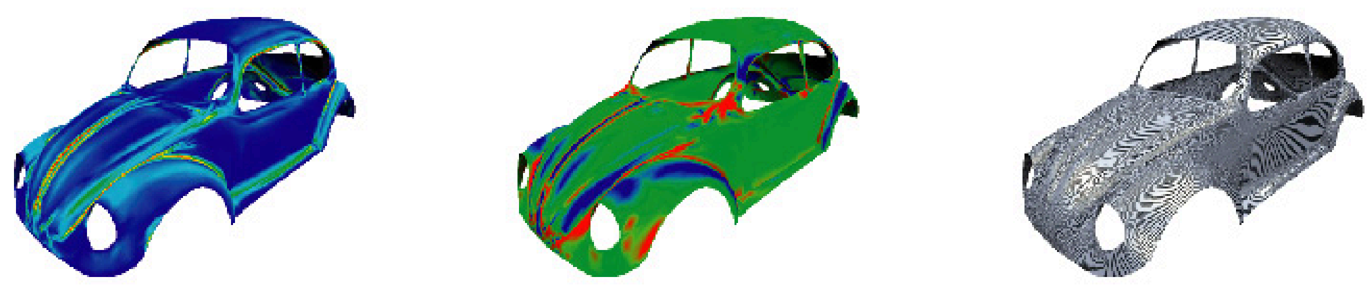
Analysis of surface quality



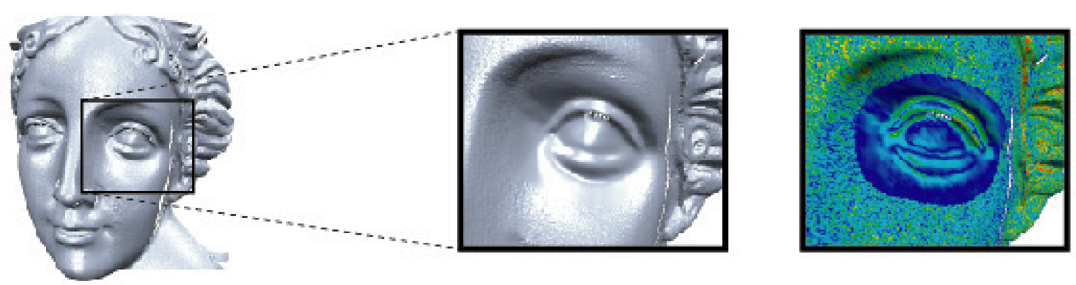


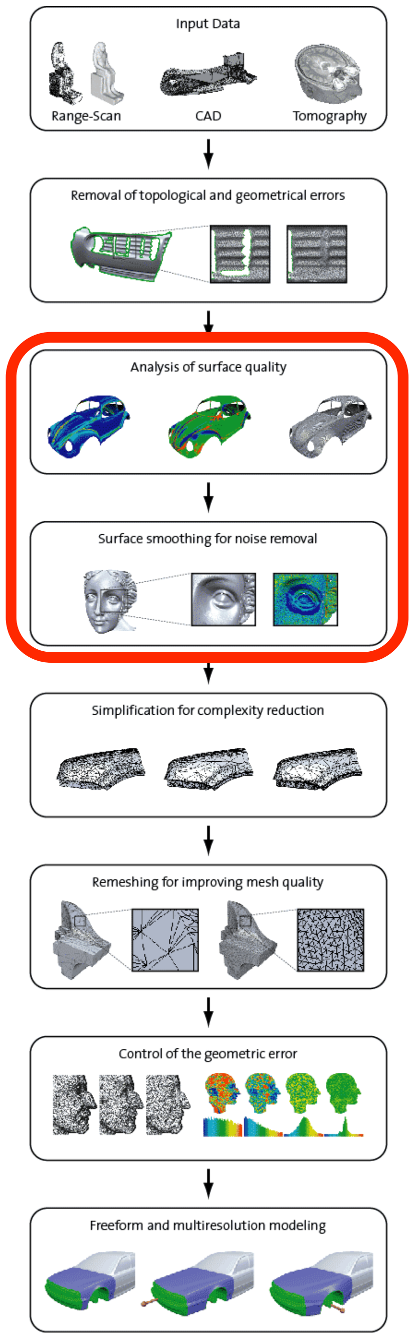
Processing Pipeline

Analysis of surface quality



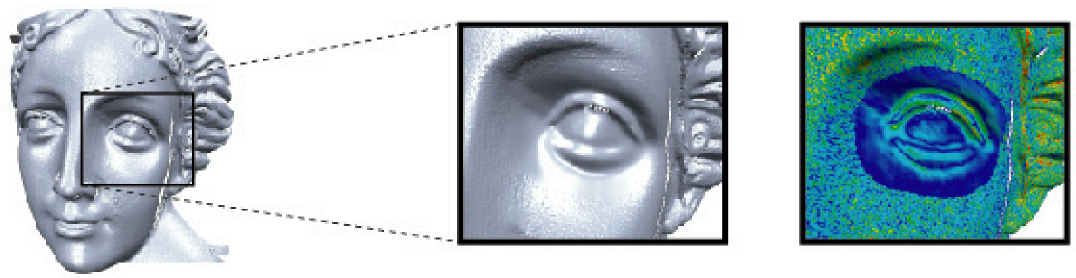
Surface smoothing for noise removal





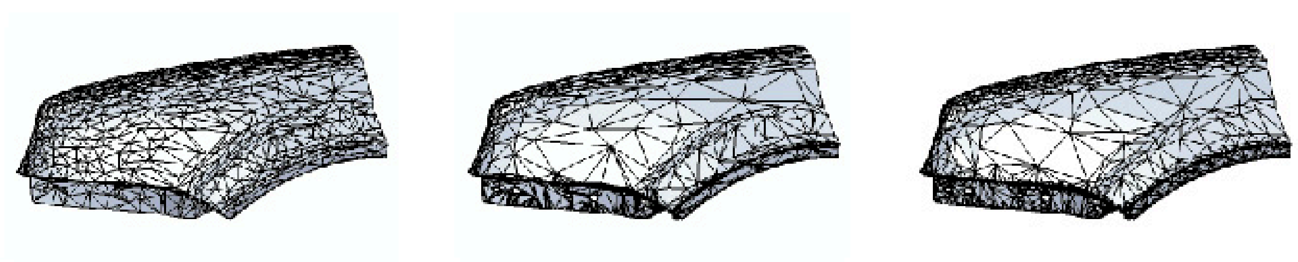
Processing Pipeline

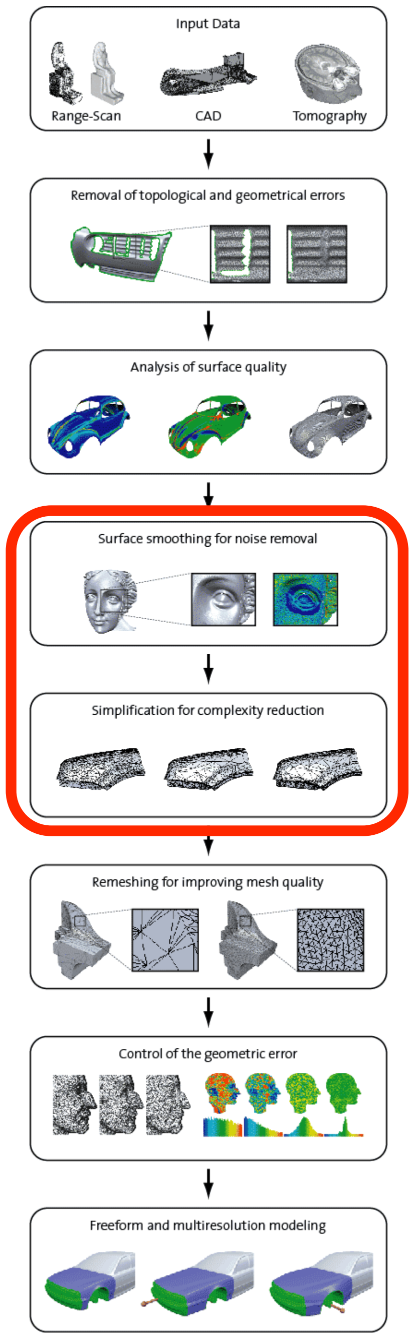
Surface smoothing for noise removal



↓

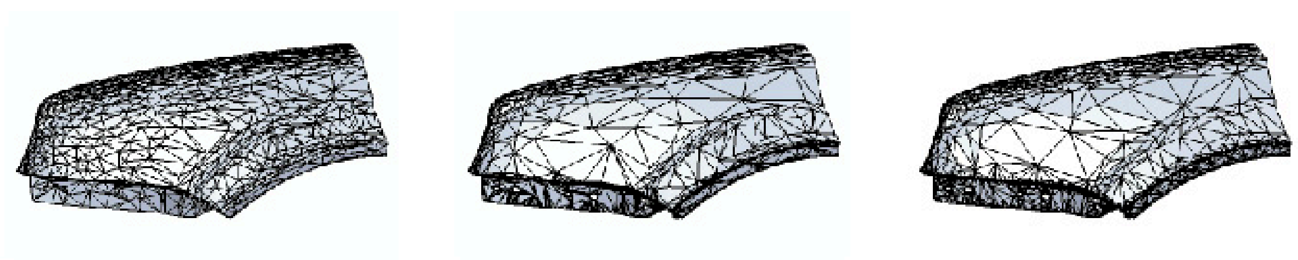
Simplification for complexity reduction



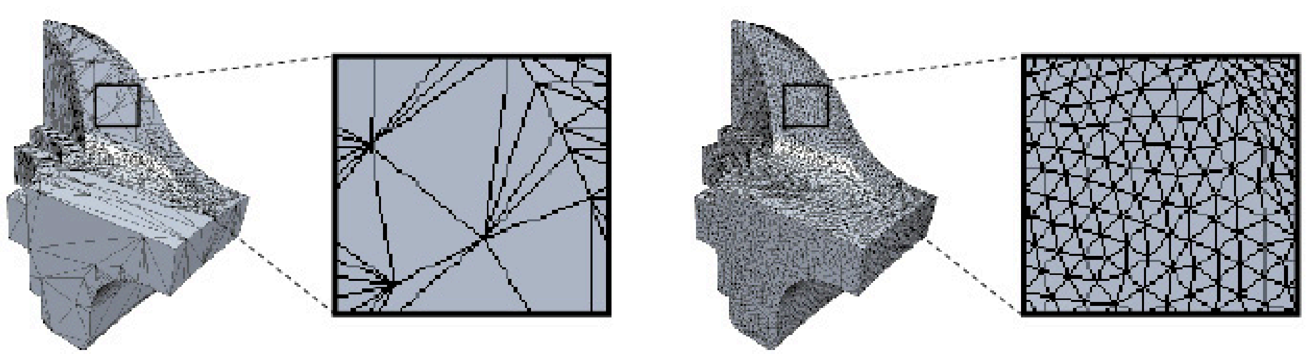


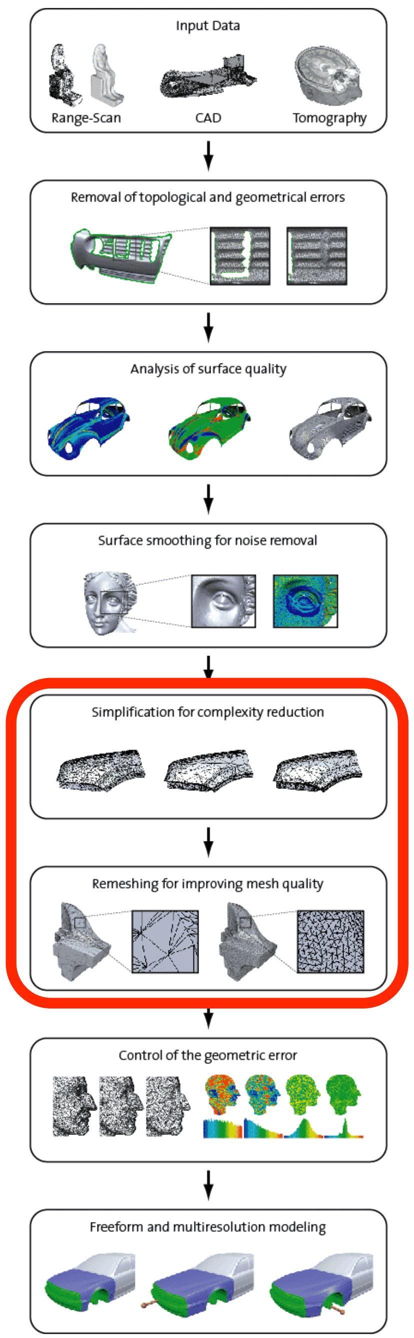
Processing Pipeline

Simplification for complexity reduction



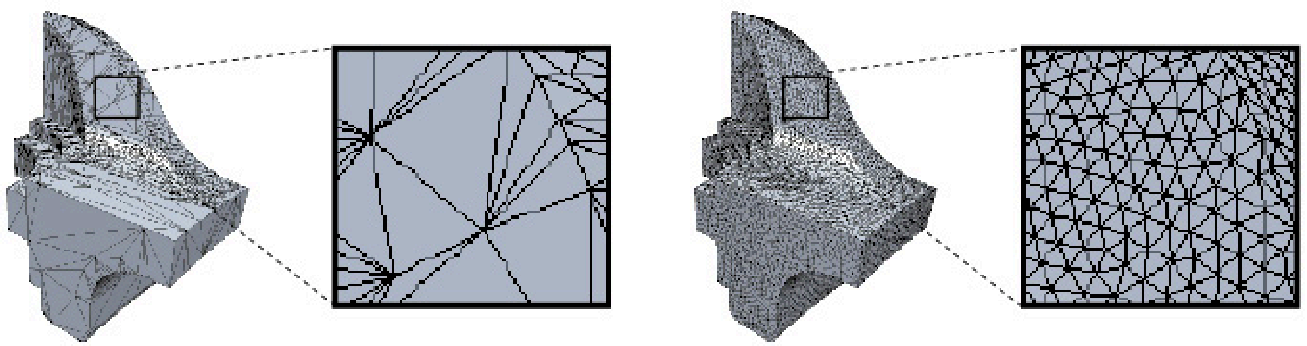
Remeshing for improving mesh quality



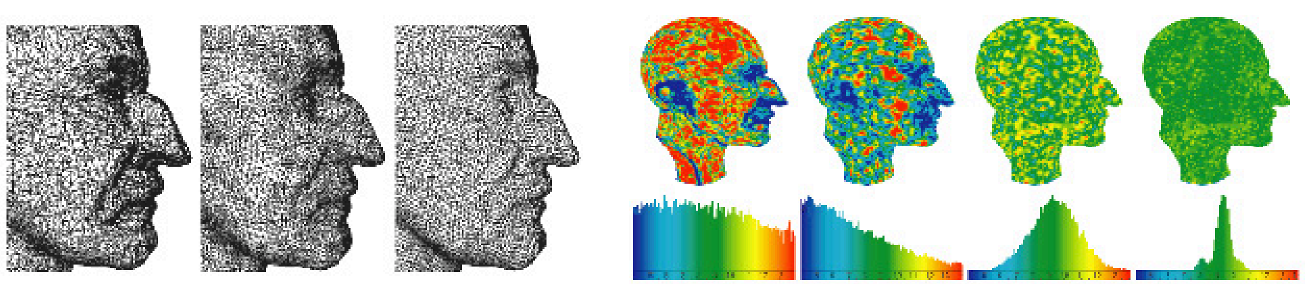


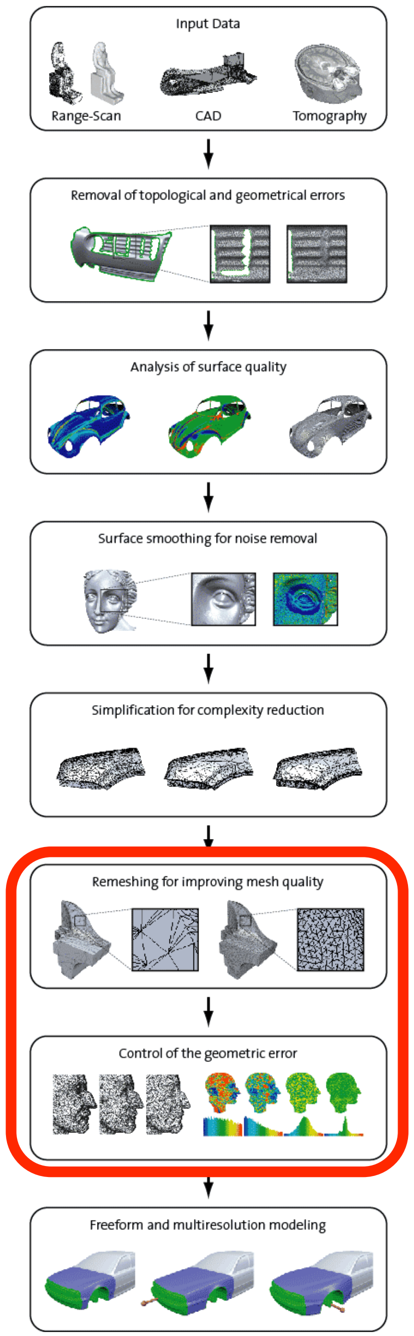
Processing Pipeline

Remeshing for improving mesh quality

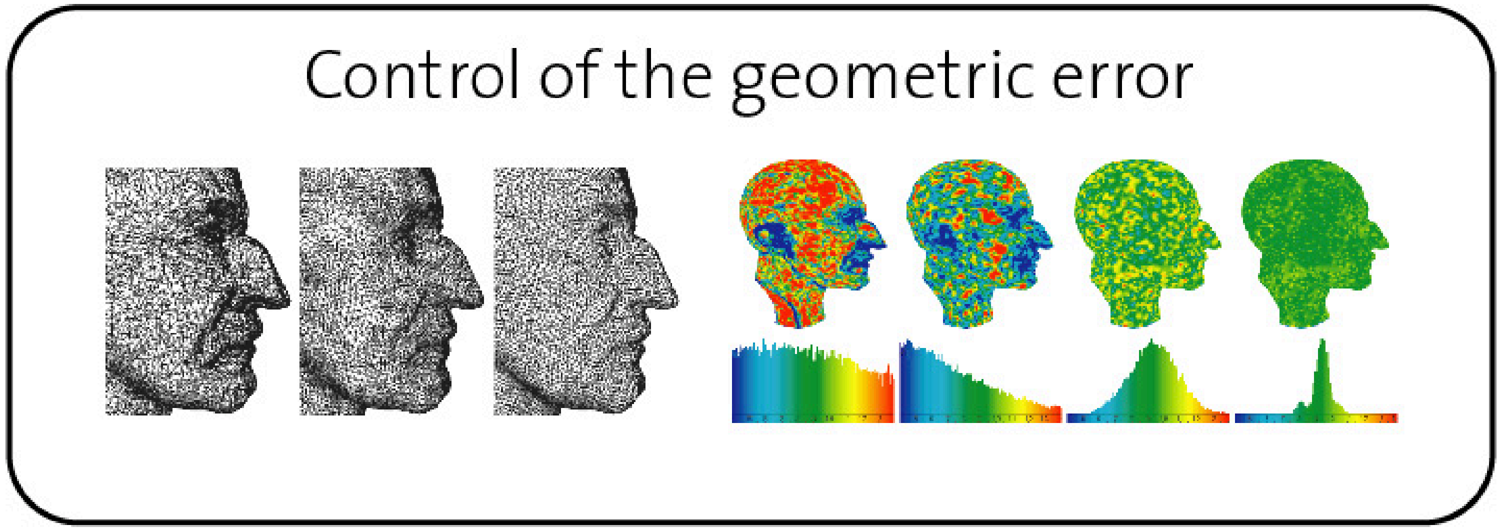


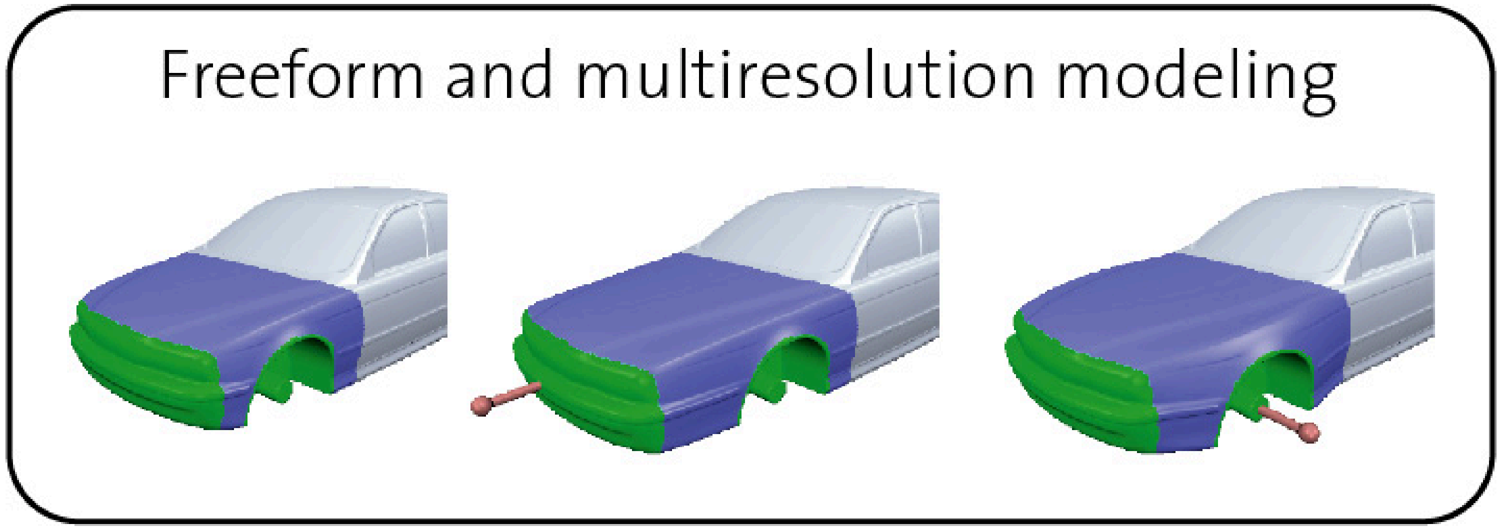
Control of the geometric error

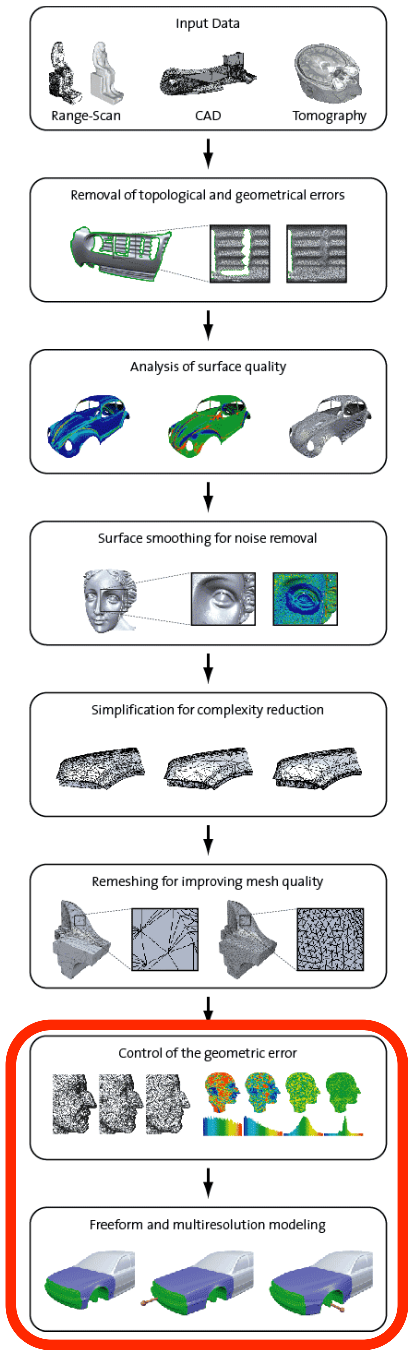




Processing Pipeline







Outline

- Part I 9:00 - 10:30
  - Introduction
  - Data Acquisition
  - Surface Representations
  - Conversions

- Coffee Break 10:30 - 11:00

Outline

- Part II 11:00 - 12:30
  - Surface Quality Analysis
  - Mesh Repair
  - Discrete Curvatures
  - Mesh Smoothing

- Lunch Break 12:30 - 13:45

Outline

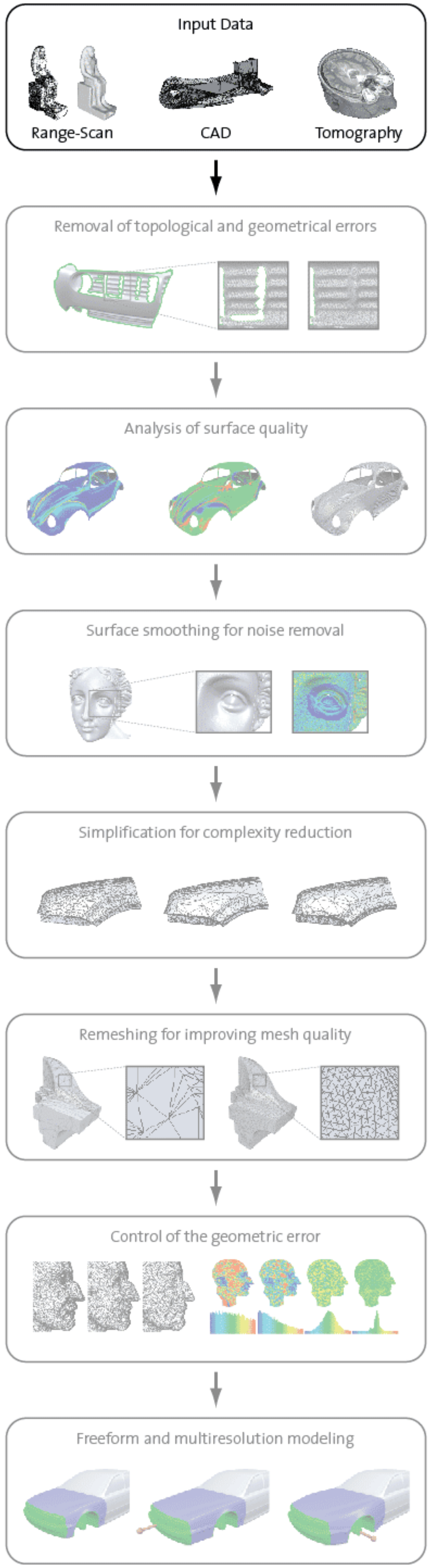
- Part III 13:45 - 15:15
  - Mesh Decimation
  - Isotropic Remeshing
  - Global Error Control

- Coffee Break 15:15 - 15:45

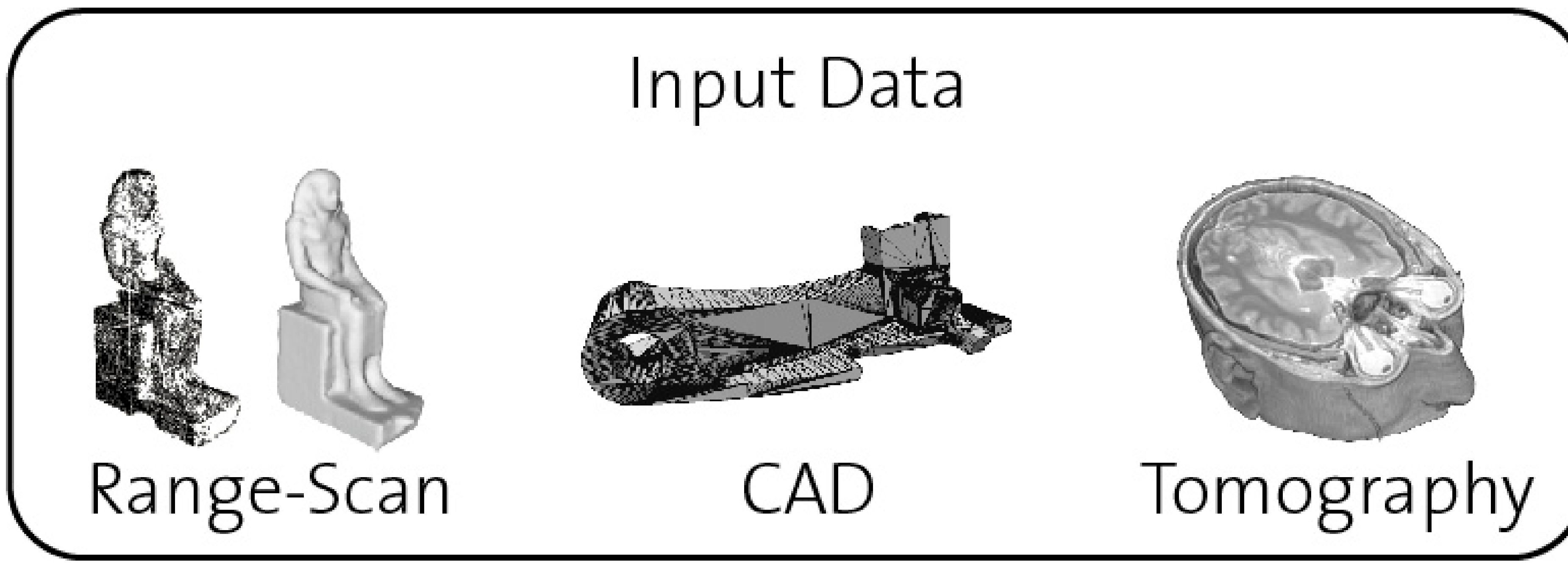
Outline

- Part IV 15:45 - 16:30
  - Surface-Based Deformation
  - Space Deformation
  - Multiresolution Modeling

- Discussion 16:30 - 17:00


Input Data
Range-Scan
CAD
Tomography
Removal of topological and geometrical errors
Analysis of surface quality
Surface smoothing for noise removal
Simplification for complexity reduction
Remeshing for improving mesh quality
Control of the geometric error
Freeform and multiresolution modeling

Input Data

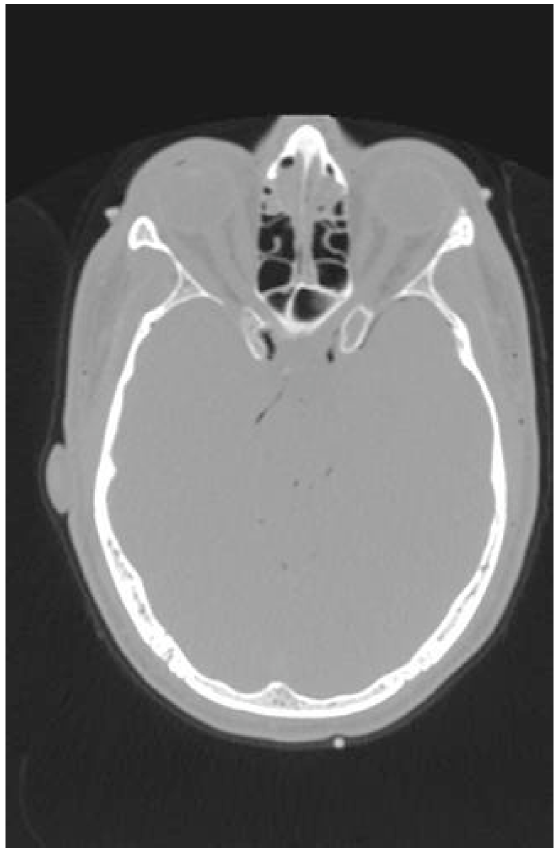

Input Data
Range-Scan
CAD
Tomography

Outline

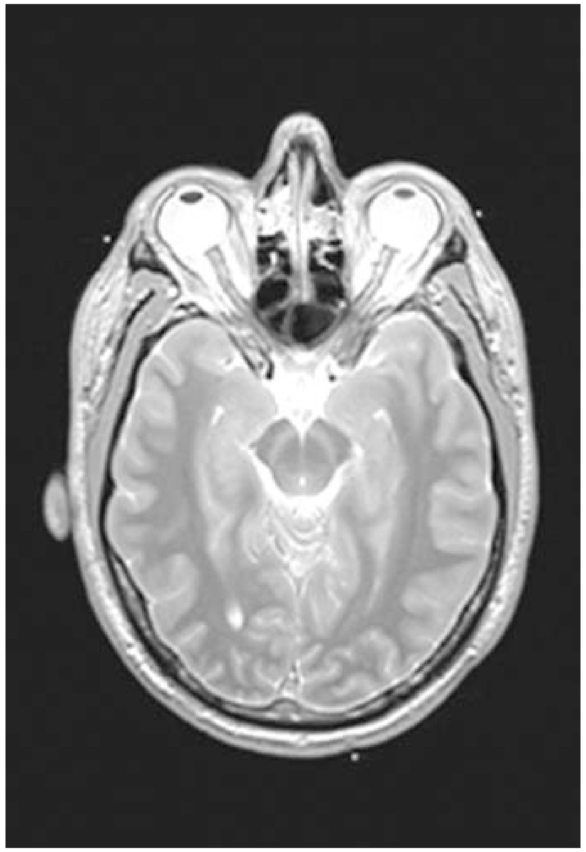
- Overview of different acquisition systems
  - volumetric scanning
  - photogrammetry
  - range scanning
- Surface Representations

Volume Scanning

- Build voxel structure by scanning slices



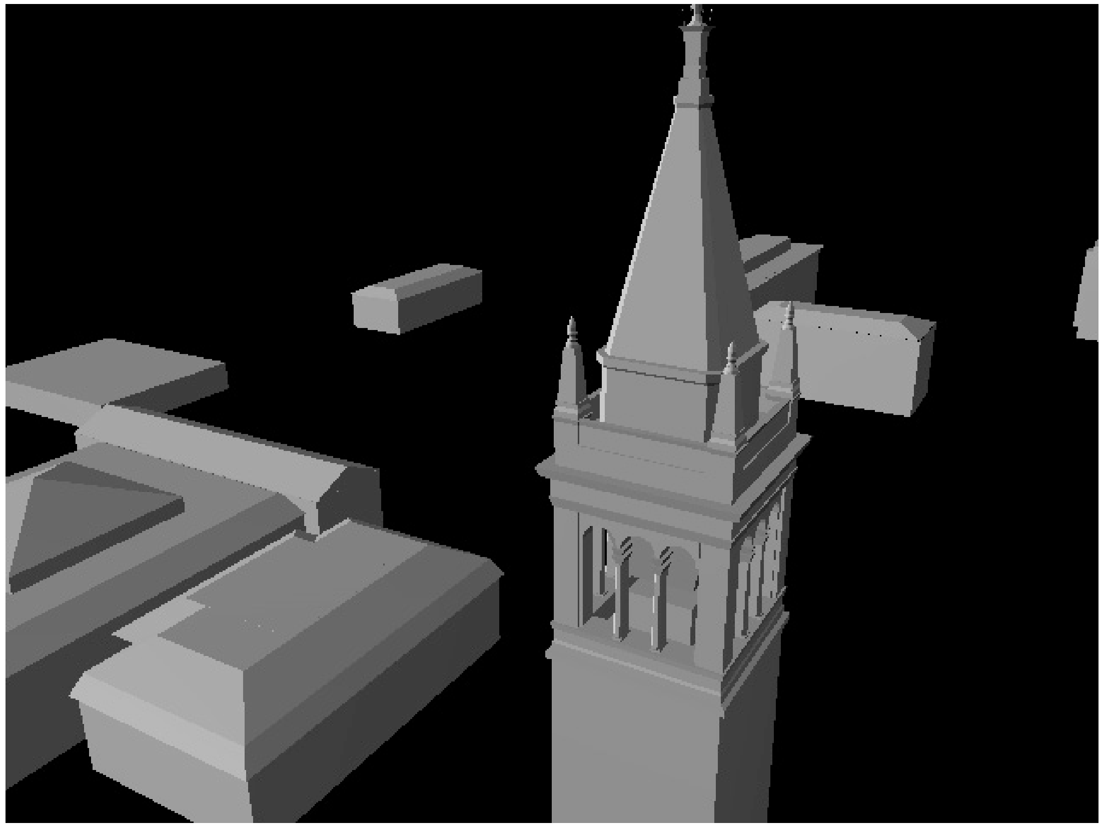
CT



MRI

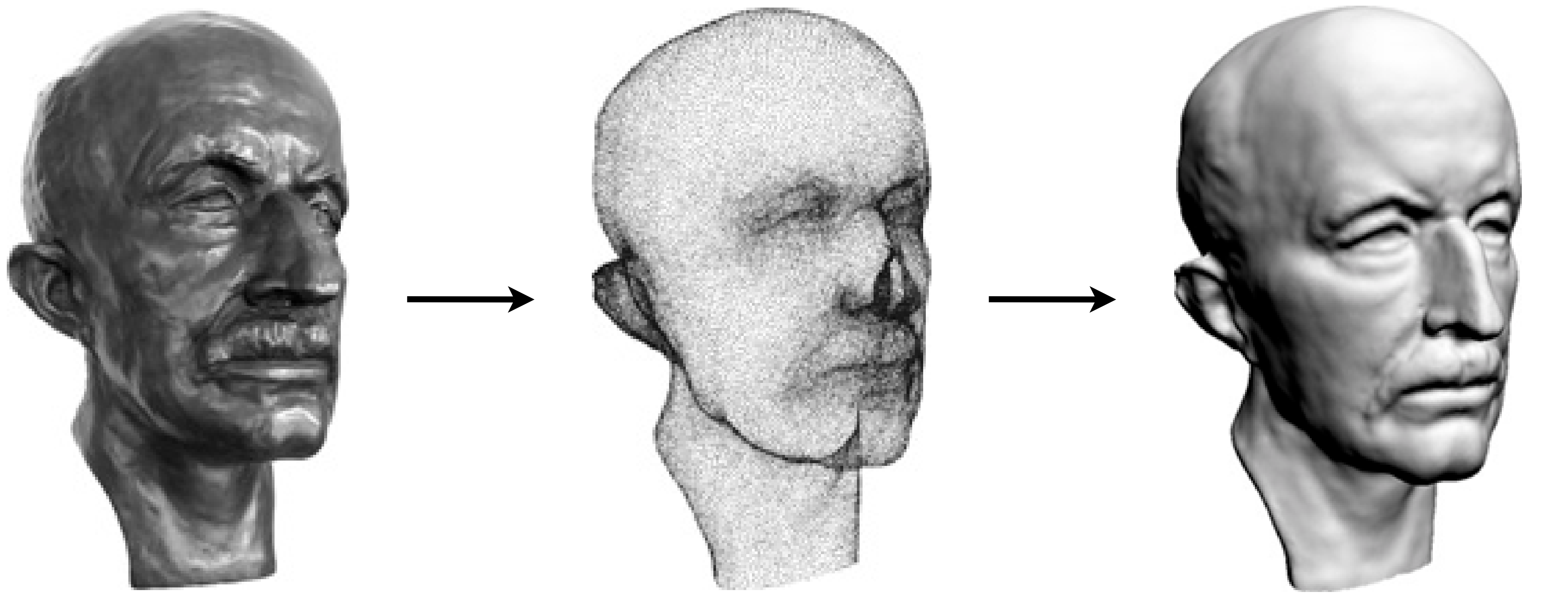
Photogrammetry

- Reconstruction from photographs

http://www.debevec.org/campanile

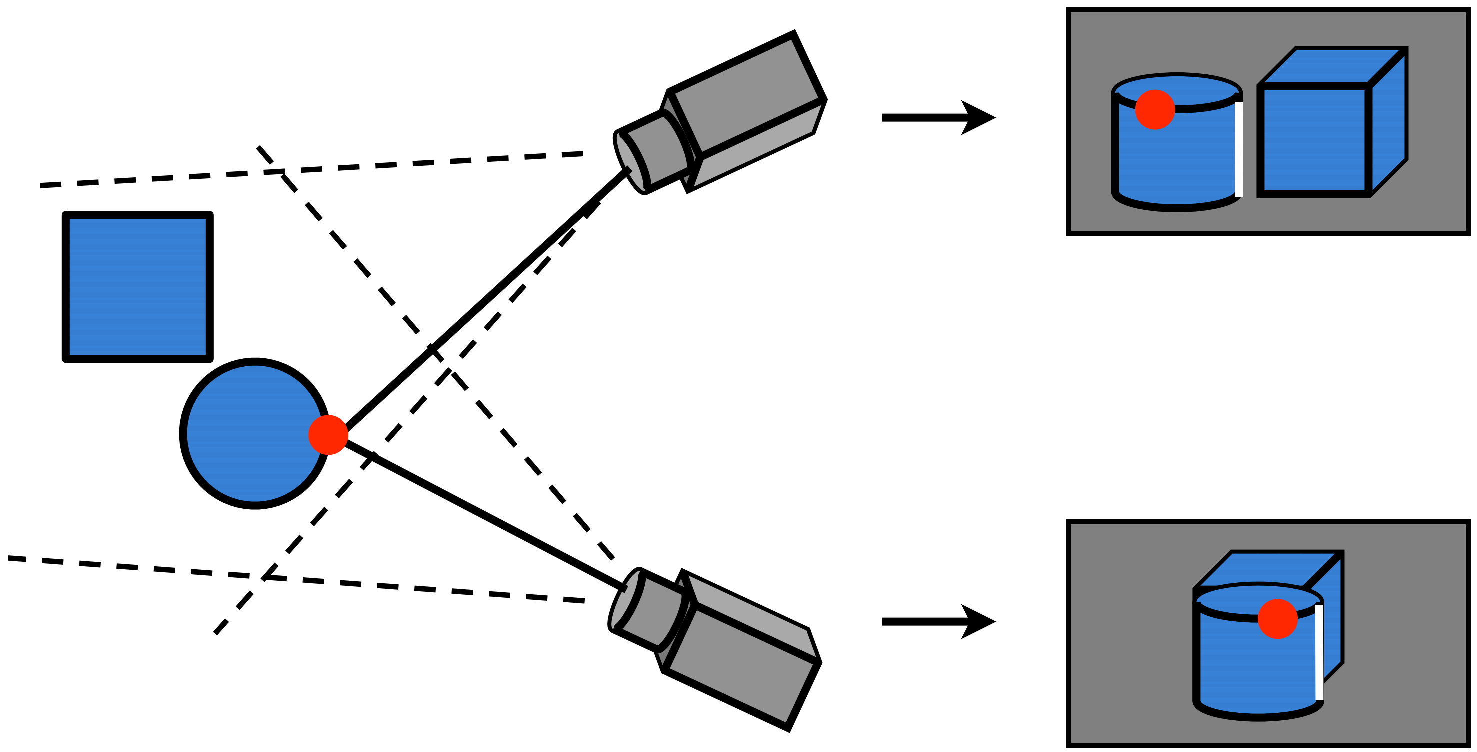
Range Scanning



physical model — acquired point cloud — reconstructed model

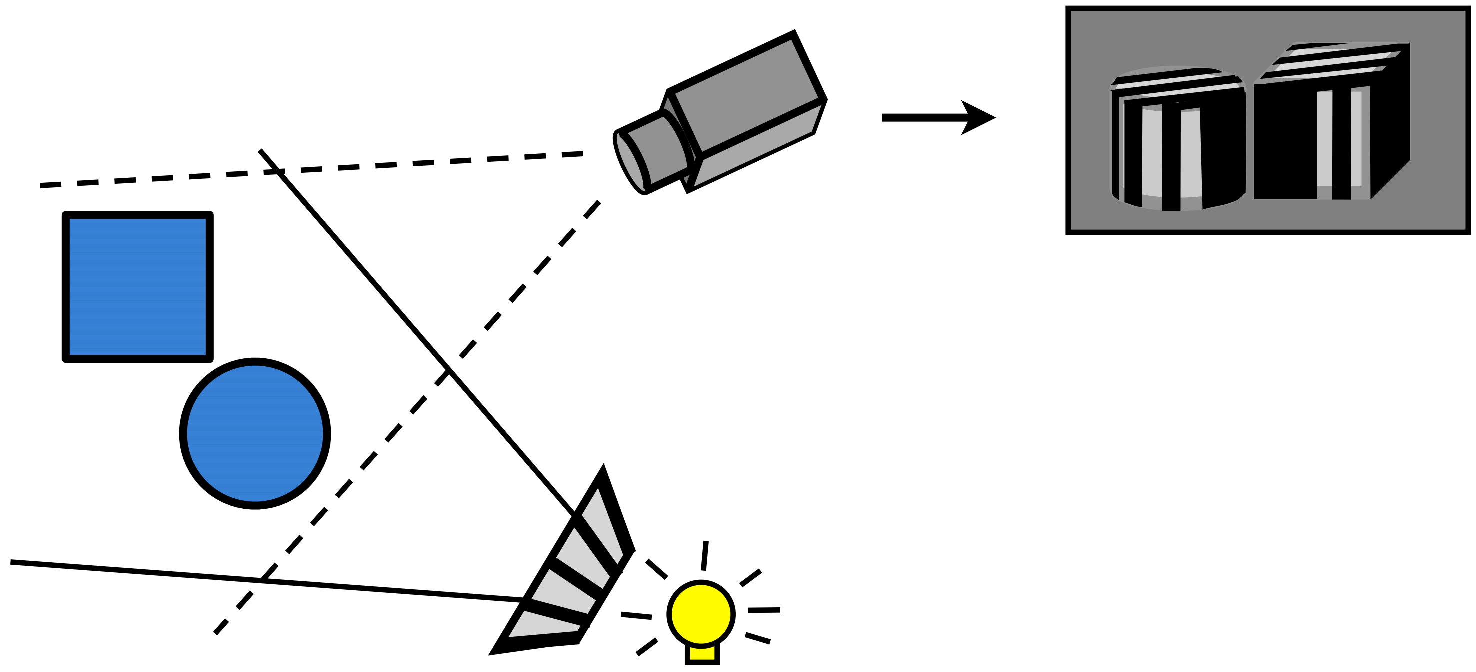
Range Scanning Systems

- Passive: Stereo Matching



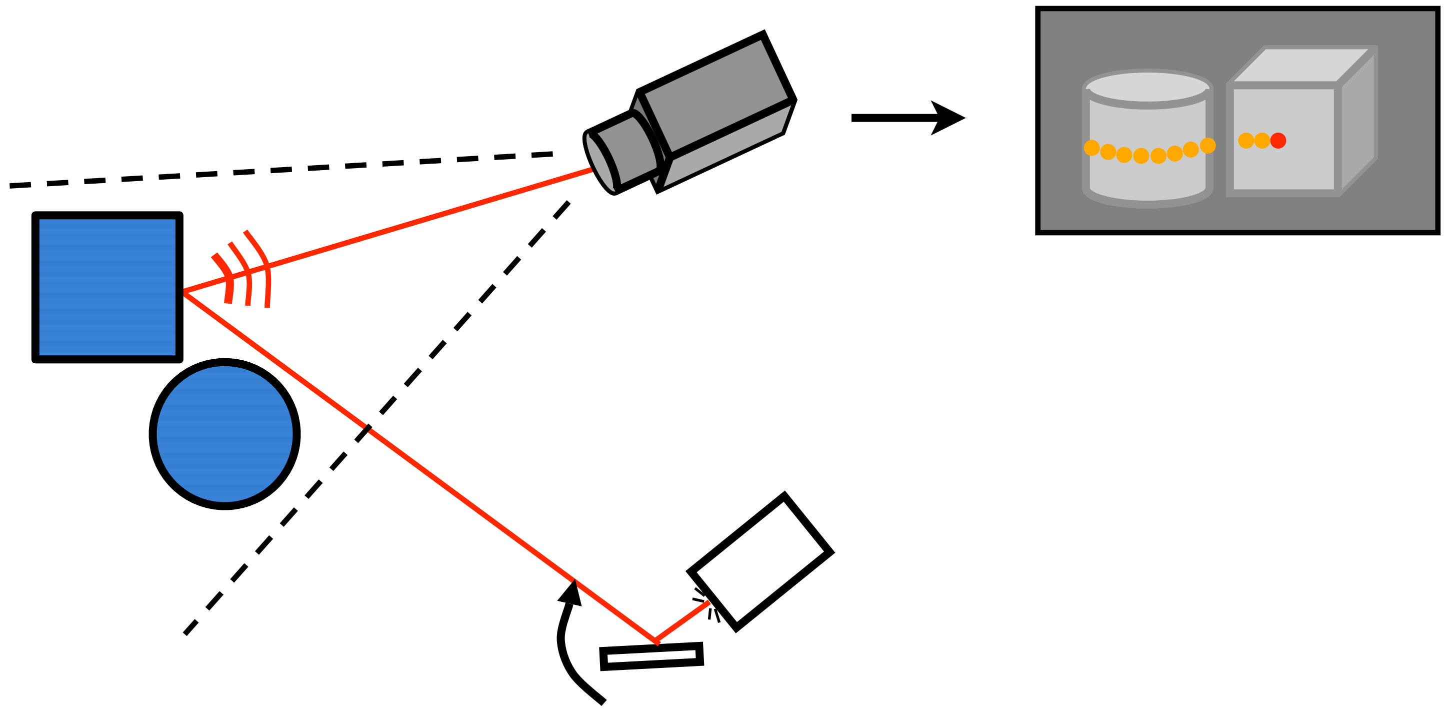
Range Scanning Systems

- Active: Structured Light Acquisition



Range Scanning Systems

- Active: Laser Scanning

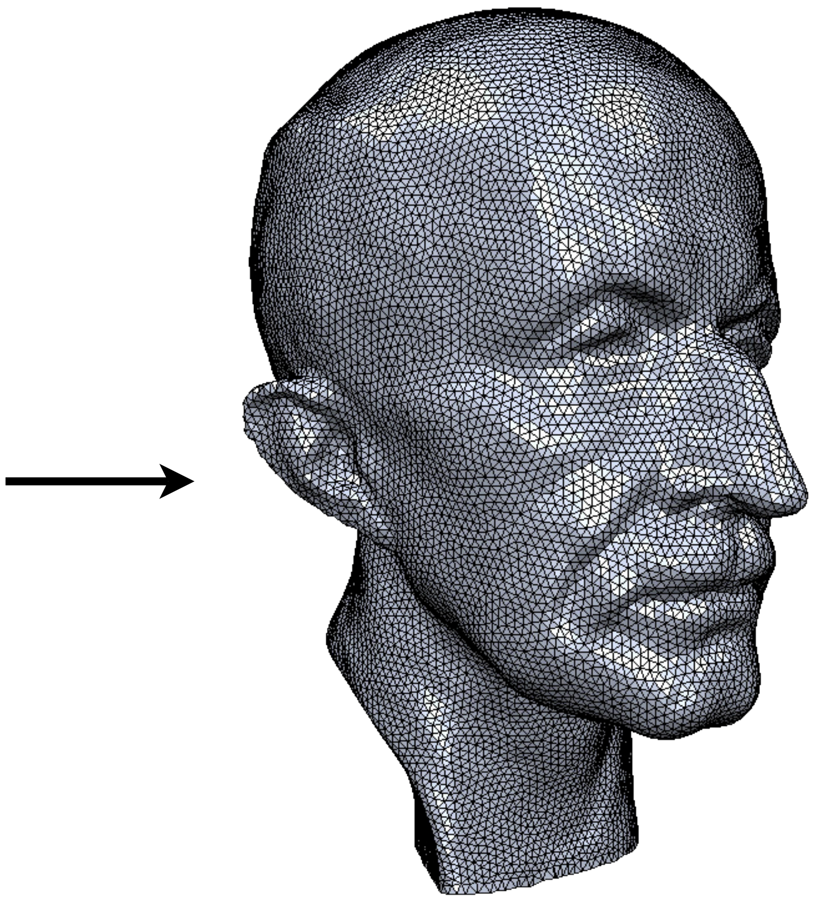


Range Scanning

- Active systems are superior
- Accurate calibration is crucial
- Multiple scans required for complex objects
  - scan path planing
  - scan registration
- Scans are incomplete and noisy
  - model repair, hole filling
  - smoothing for noise removal

Goal

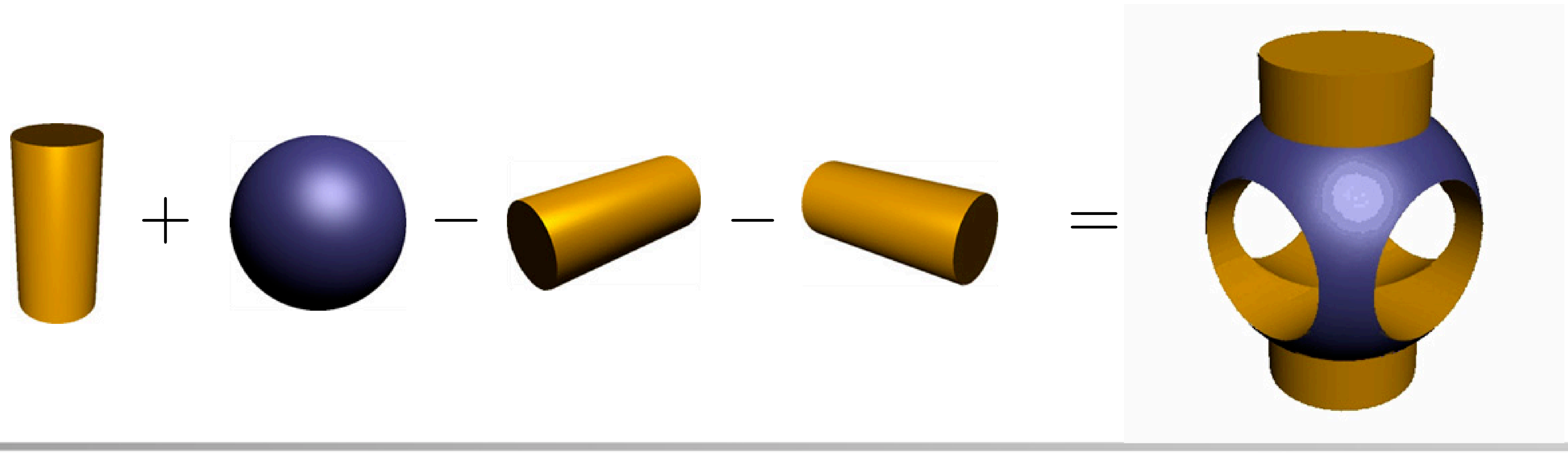
set of raw scans



reconstructed model

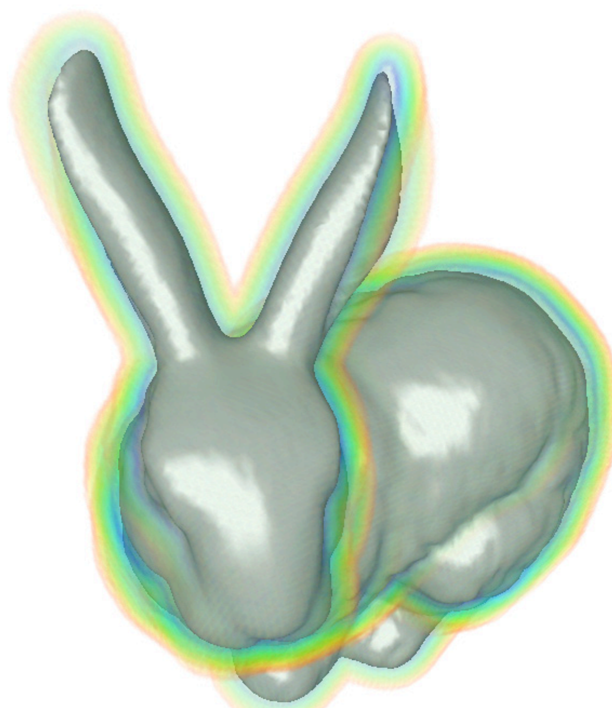
Representing Surfaces

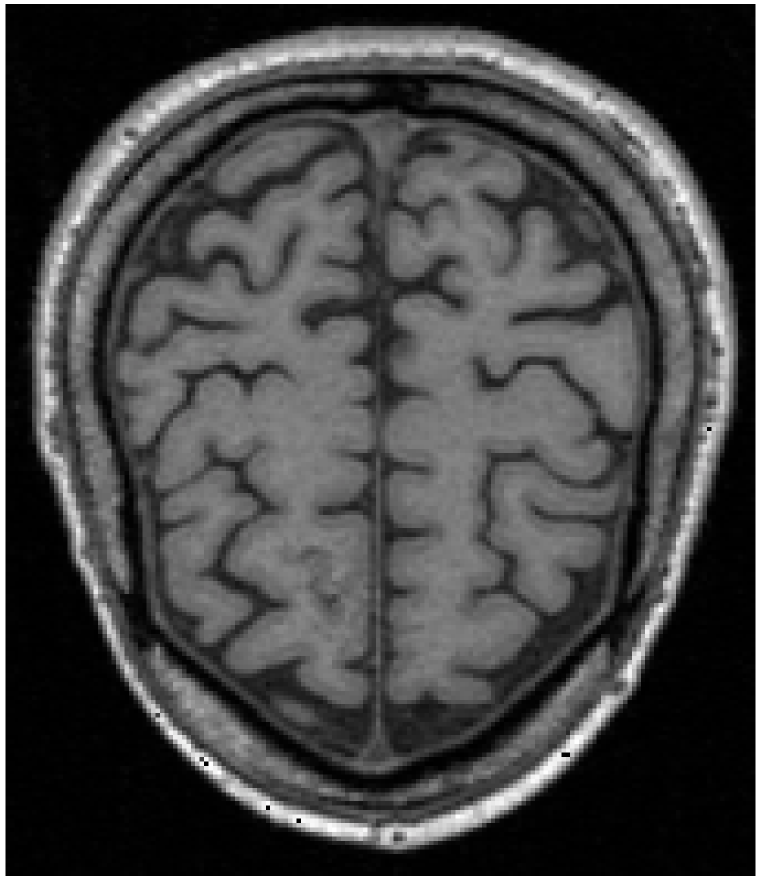
- Constructive $((A \cup B) \cap C) \cap D$



Representing Surfaces

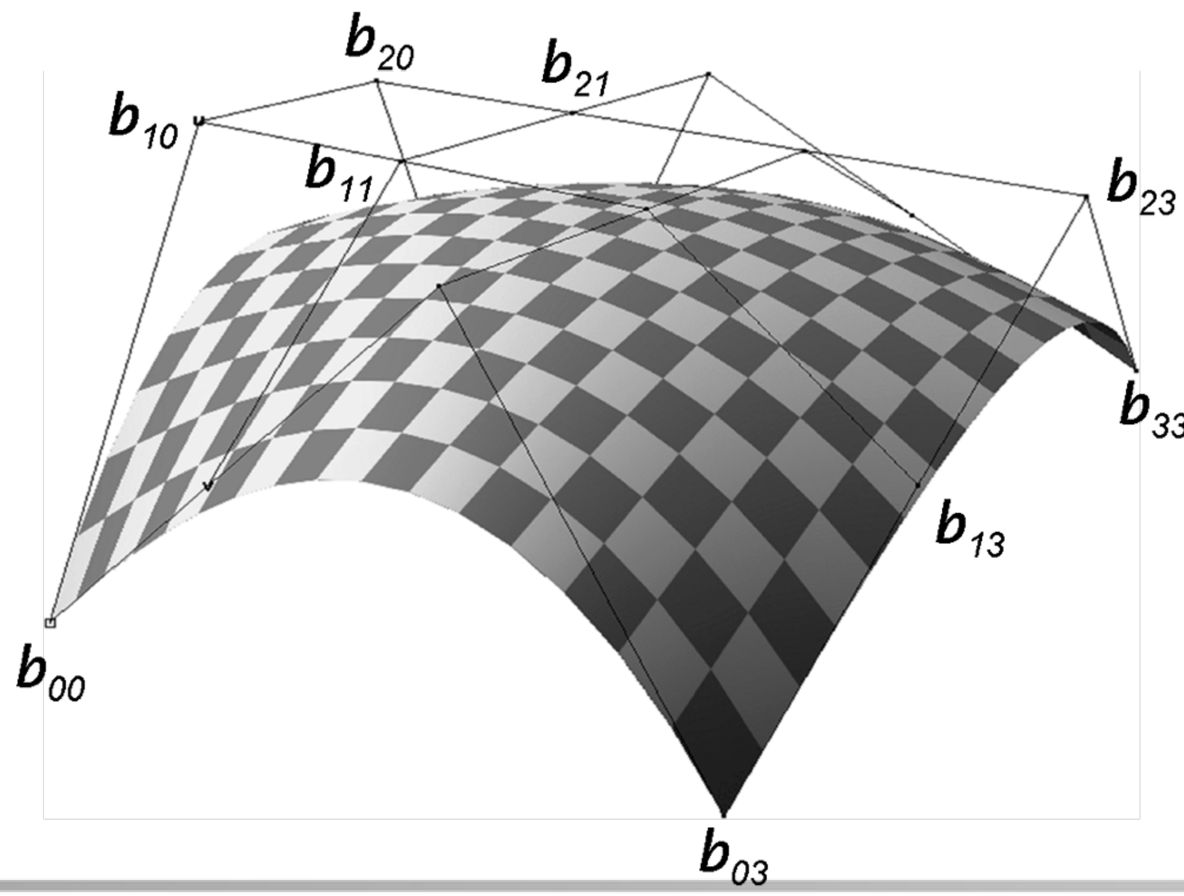
- Constructive $((A \cup B) \cap C) \cap D$
- Implicit $f(x, y, z) = 0$





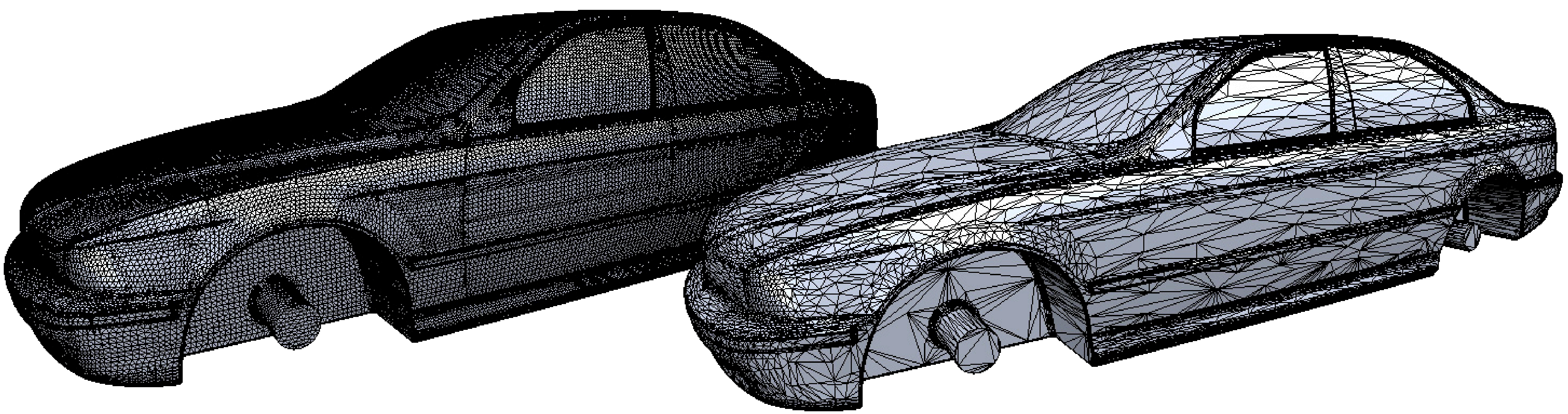
Representing Surfaces

- Constructive $((A \cup B) \cap C) \cap D$
- Implicit $f(x, y, z) = 0$
- Parametric $f(u, v) = [x, y, z]^T$



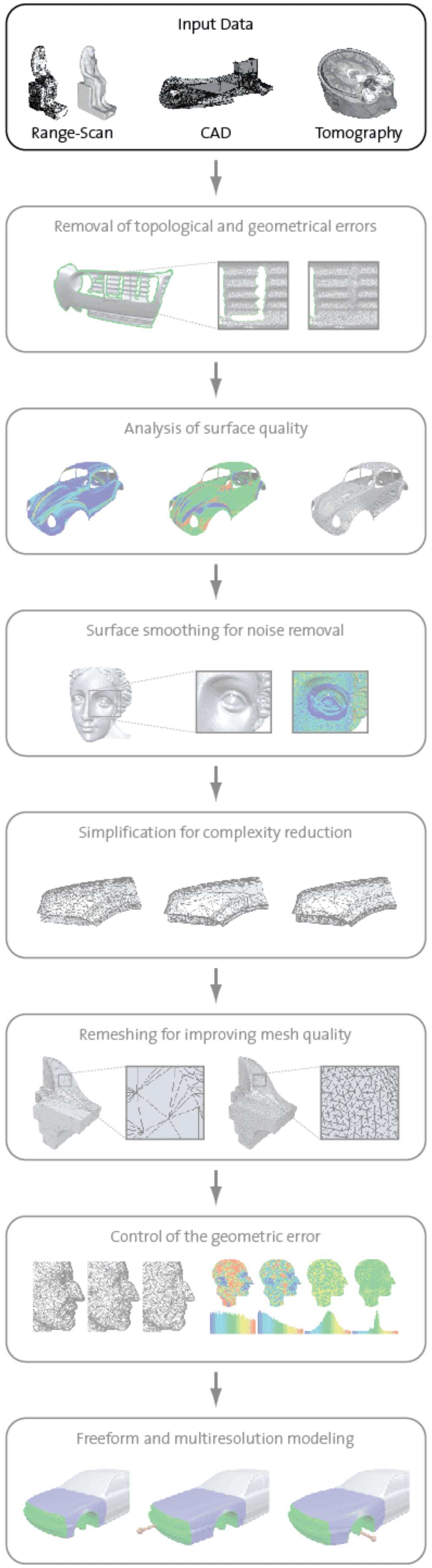
Representing Surfaces

- Constructive $((A \cup B) \cap C) \cap D$
- Implicit $f(x, y, z) = 0$
- Parametric $f(u, v) = [x, y, z]^T$
- Explicit $(\{\mathbf{v}_0, \ldots, \mathbf{v}_n\}, \{[i_0, j_0, k_0], \ldots, [i_m, j_m, k_m]\})$

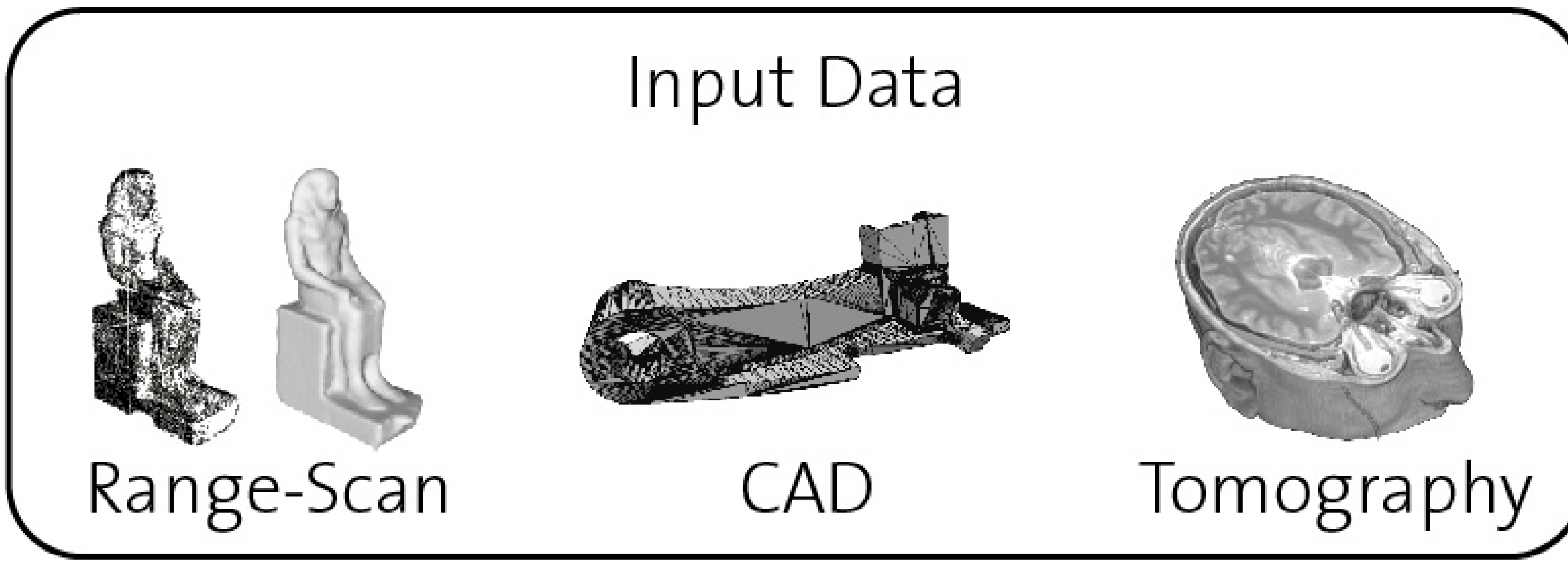


Links & Literature

- ICCV 2005 Short Course: *3D Scan Matching and Registration*
  - http://www.cs.princeton.edu/~bjbrown/iccv05_course/
- Scanalyze: a system for aligning and merging range data
  - http://graphics.stanford.edu/software/scanalyze/
- Davis, Nehab, Ramamoorthi, Rusinkiewicz: *Spacetime Stereo: A Unifying Framework for Depth from Triangulation*. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 27(2), 2005.
- Weyrich, Pauly, Keiser, Heinzle, Scandella, Gross: *Post-processing of Scanned 3D Surface Data*.. Symposium on Point-Based Graphics 2004



Surface Representations



Outline

- Surface Representations
  - Explicit vs. Implicit
- Explicit Representation
  - Triangle Meshes
- Implicit Representations
  - Signed Distance Functions
- Conversions
  - Implicit $\leftrightarrow$ Explicit

Outline

- Surface Representations
  - Explicit vs. Implicit
- Explicit Representation
  - Triangle Meshes
- Implicit Representations
  - Signed Distance Functions
- Conversions
  - Implicit ↔ Explicit

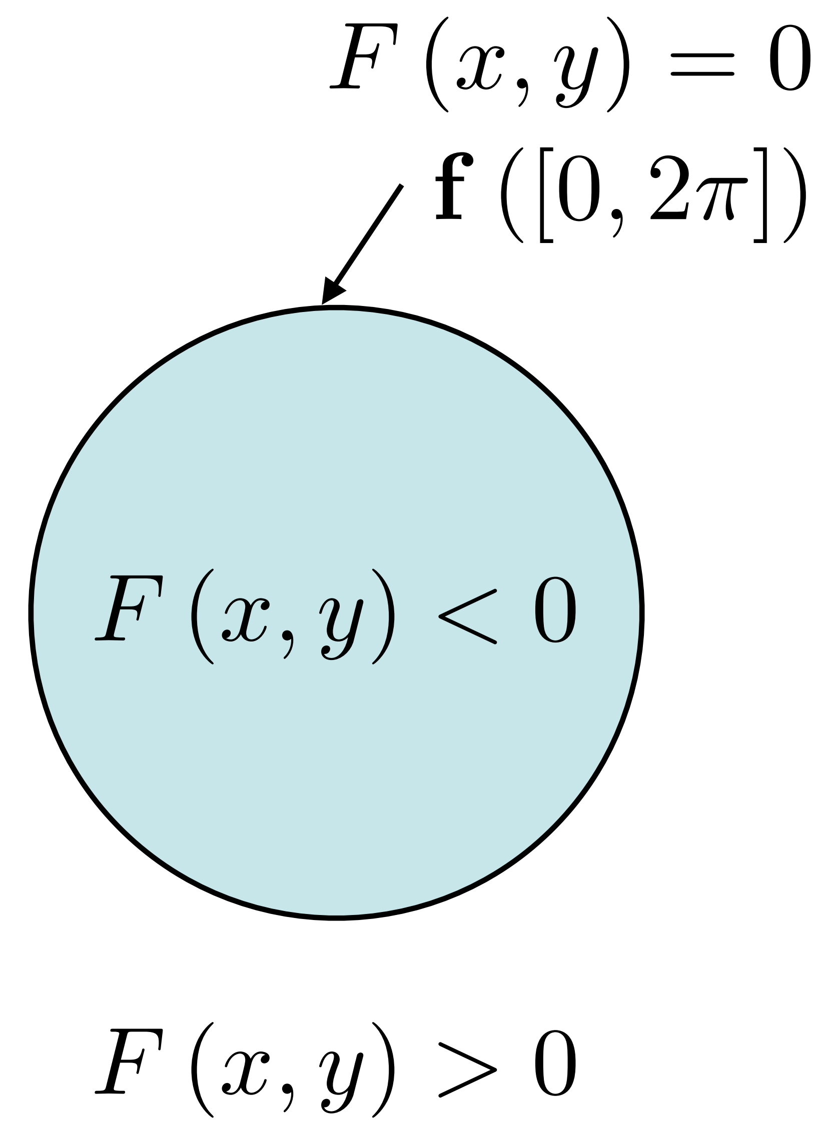
Explicit / Implicit

- Explicit representation
  - Image of parametrization

$$\mathbf{f}(x) = \begin{pmatrix} r \cdot cos(x) \\ r \cdot sin(x) \end{pmatrix}$$

- Implicit representation
  - Kernel of distance function

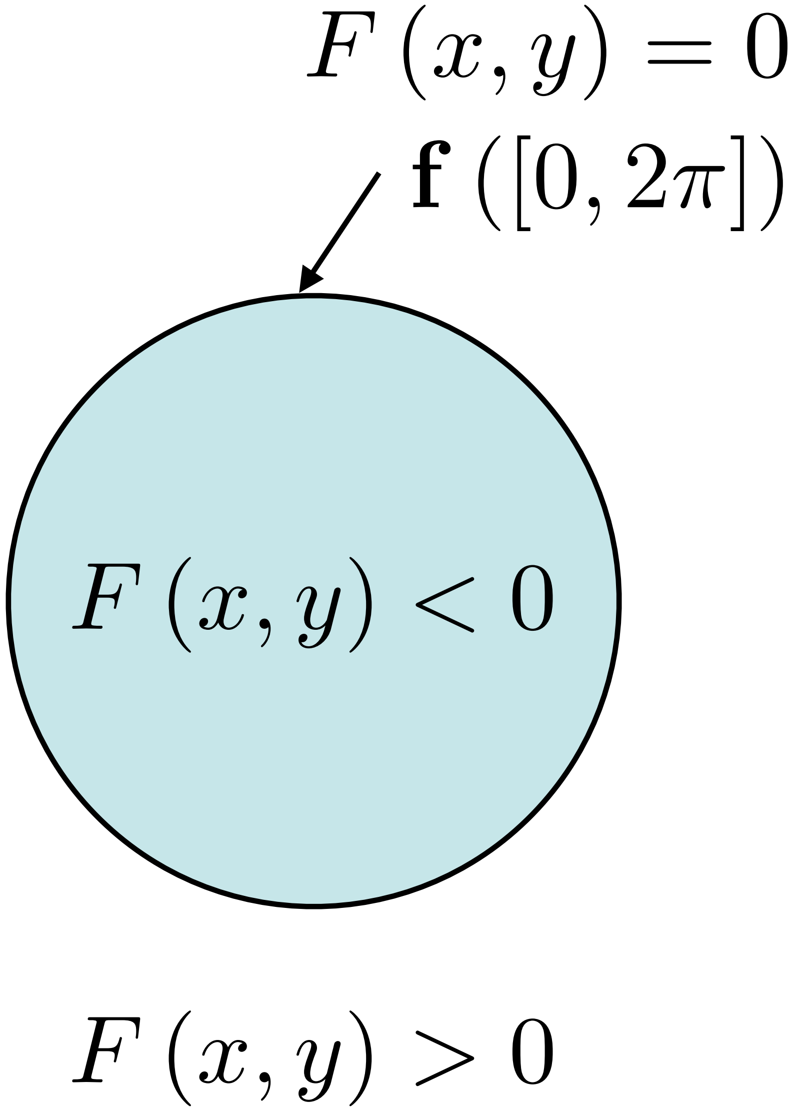
$$F(x,y) = \sqrt{x^2 + y^2} - r$$



Explicit / Implicit

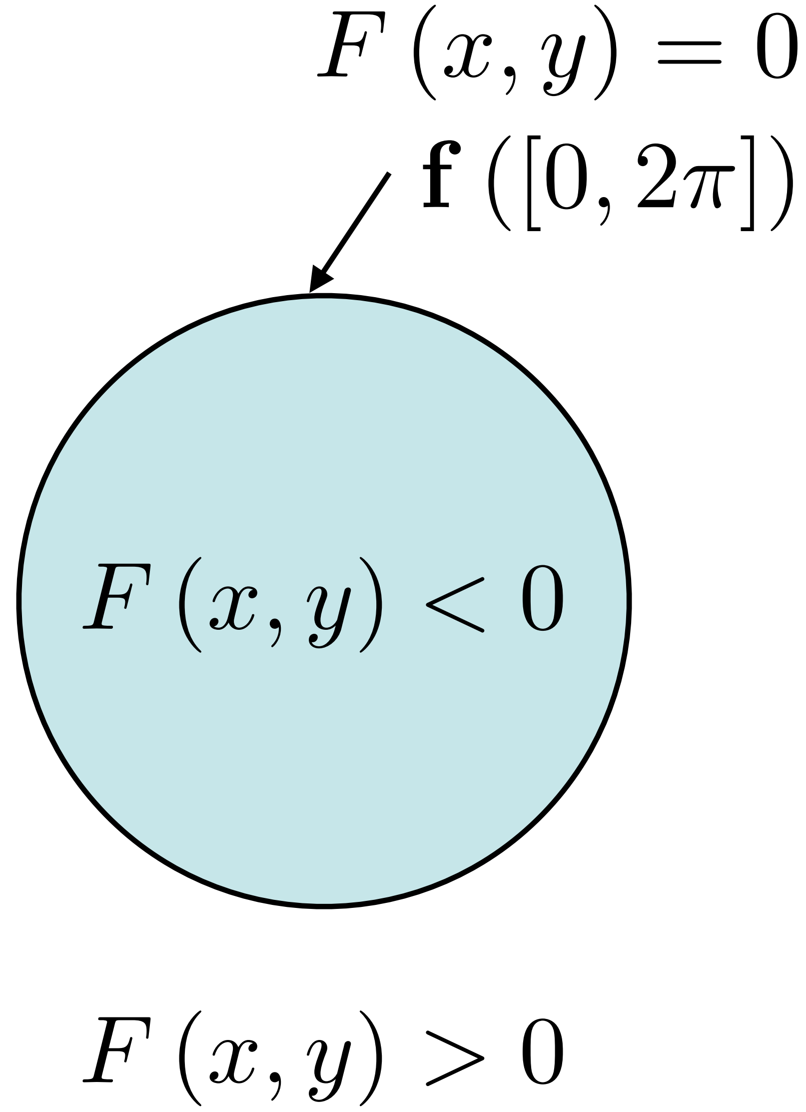
- Explicit representation
  - Image of parametrization
  - Easy enumeration

- Implicit representation
  - Kernel of distance function
  - Easy in/out/distance test



Explicit / Implicit

- Explicit representation
  - Image of parametrization
  - Easy enumeration
  - NURBS, triangle mesh
- Implicit representation
  - Kernel of distance function
  - Easy in/out/distance test
  - Scalar-valued 3D grid



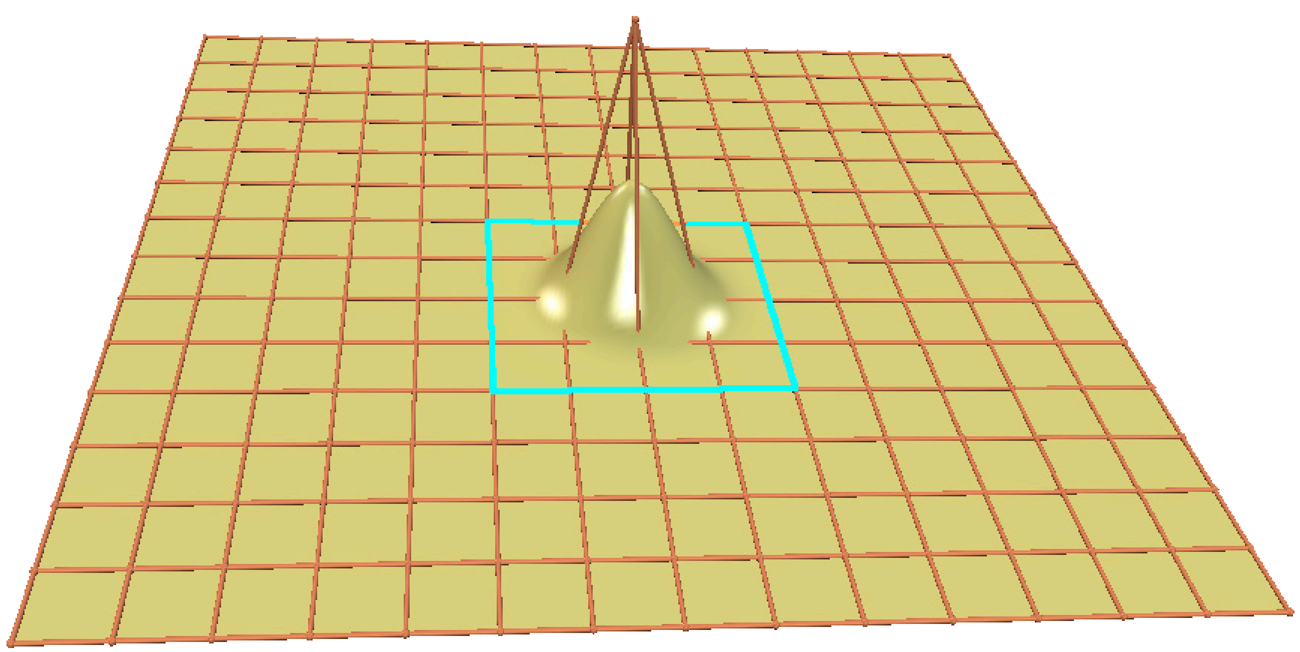
Outline

- Surface Representations
  - Explicit vs. Implicit
- Explicit Representation
  - Triangle Meshes
- Implicit Representations
  - Signed Distance Functions
- Conversions
  - Implicit ↔ Explicit

Spline Surfaces

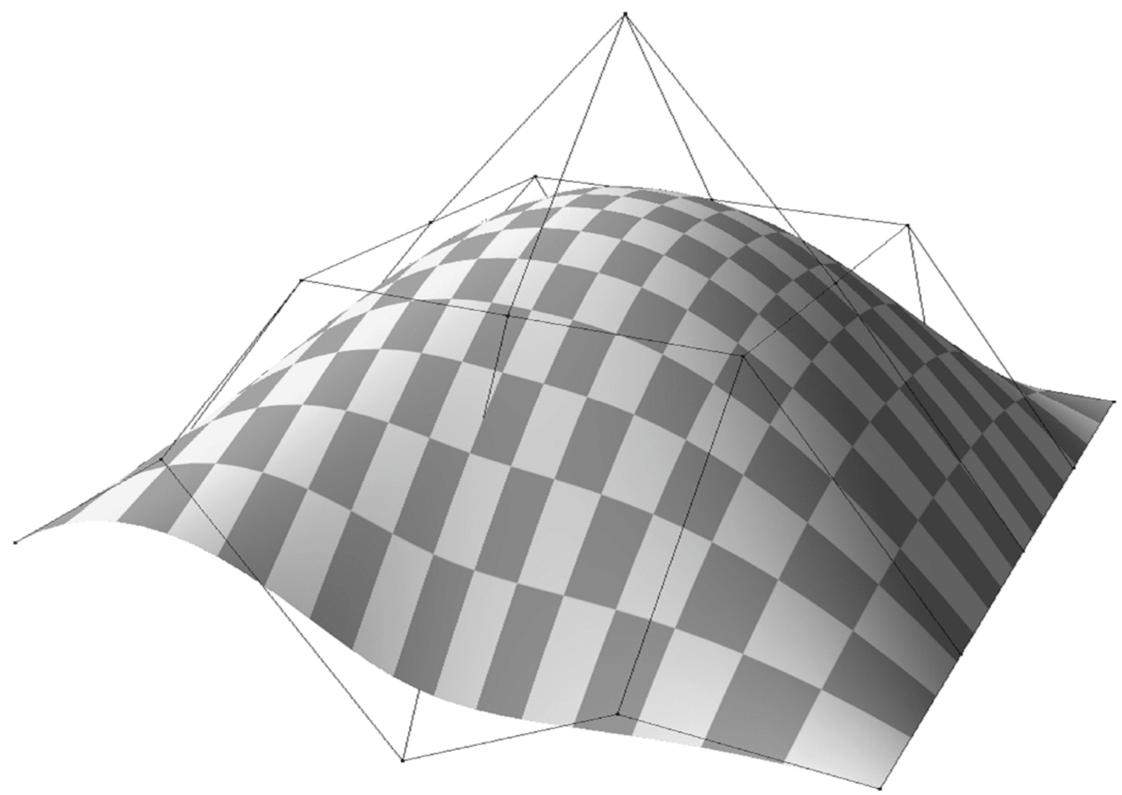
- Piecewise polynomial approximation

$$\mathbf{f}(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} \mathbf{c}_{ij} N_i^n(u) N_j^m(v)$$



Spline Surfaces

- Piecewise polynomial approximation



- Topological constraints
  - Rectangular patches
  - Regular control mesh
- Geometric constraints
  - Continuity between patches
  - Trimming

Triangle Meshes

- Topology: vertices, edges, triangles

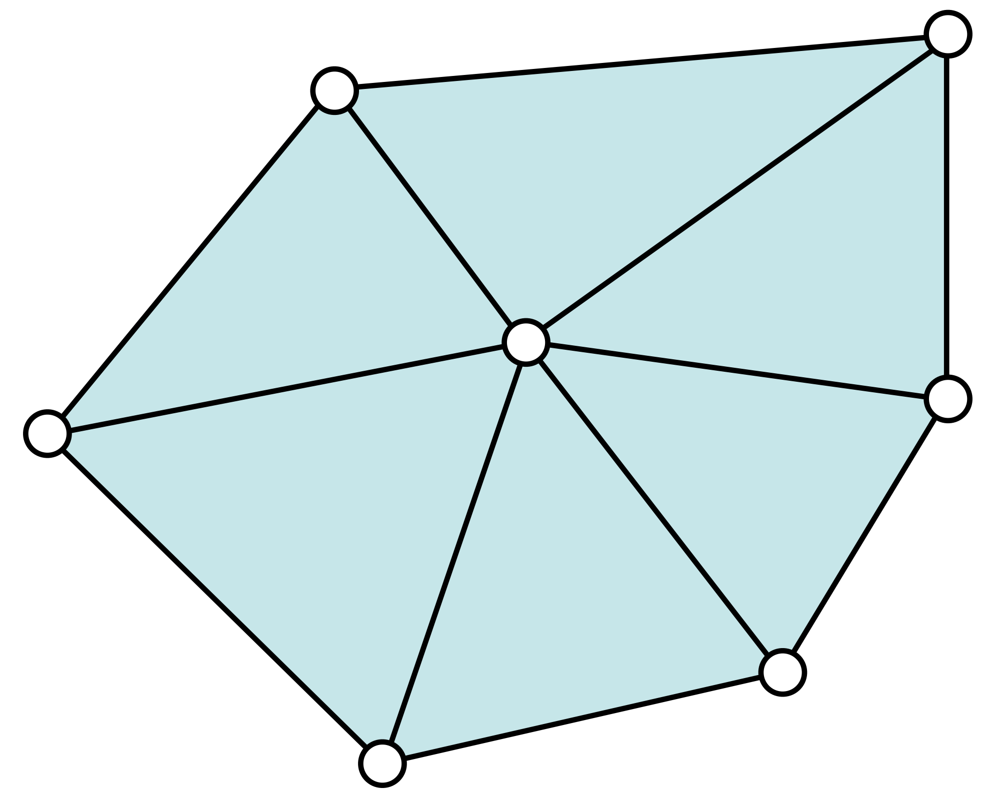
$$\mathcal{V} = \{v_1, \ldots, v_n\}$$

$$\mathcal{E} = \{e_1, \ldots, e_k\} \ , \quad e_i \in \mathcal{V} \times \mathcal{V}$$

$$\mathcal{F} = \{f_1, \ldots, f_m\} \ , \quad f_i \in \mathcal{V} \times \mathcal{V} \times \mathcal{V}$$

- Geometry: vertex positions

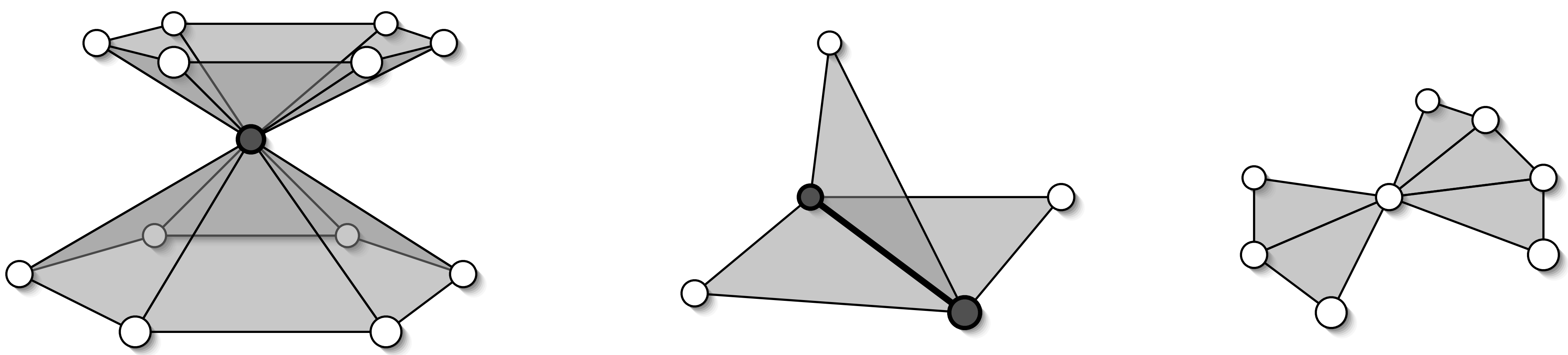
$$\mathcal{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_n\} \ , \quad \mathbf{p}_i \in \mathbb{R}^3$$



Triangle Meshes

- Consistency
  - 2-manifolds
  - Locally homeomorphic to disk

- Non-manifold examples

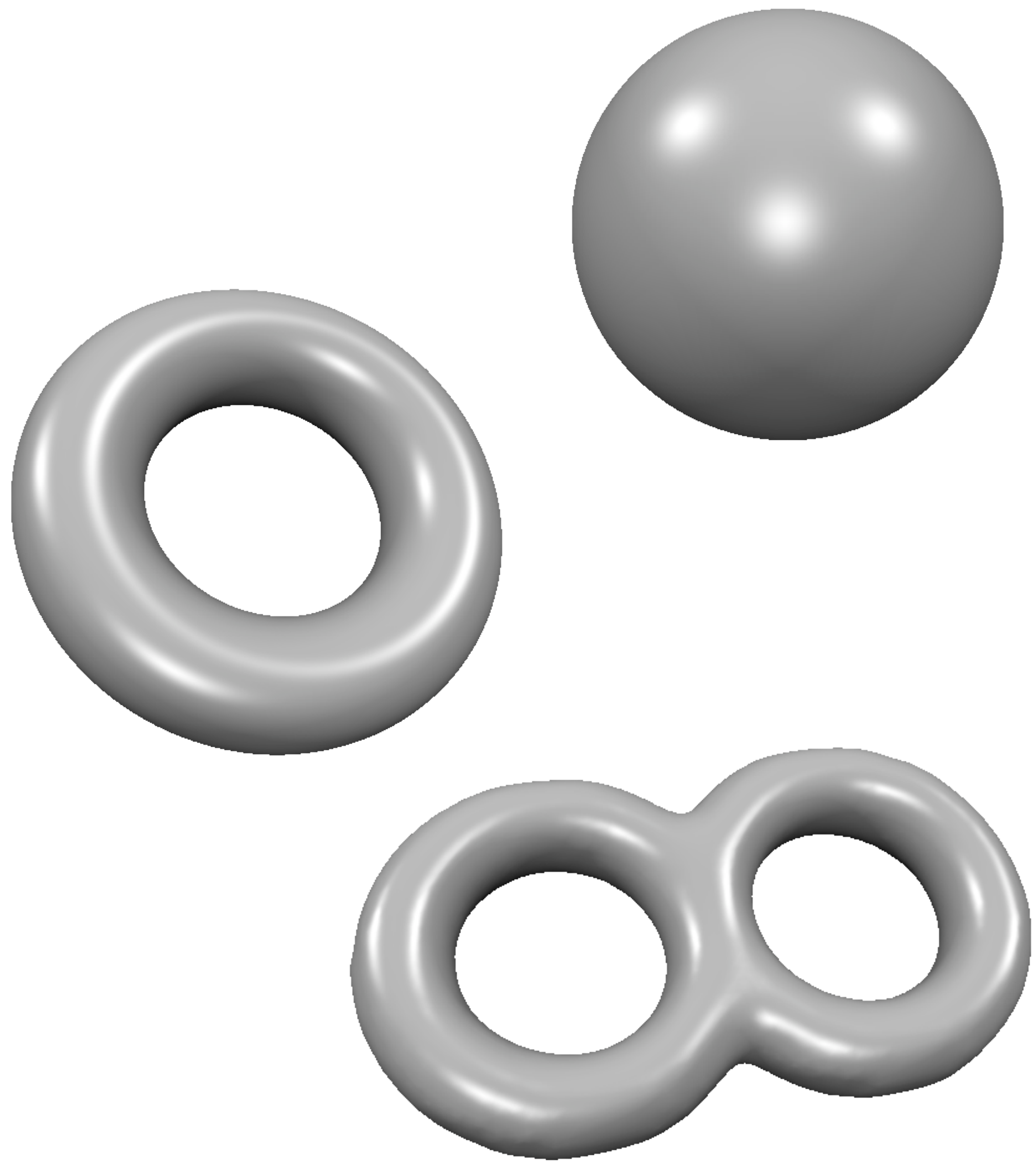


Triangle Meshes

- Euler formula

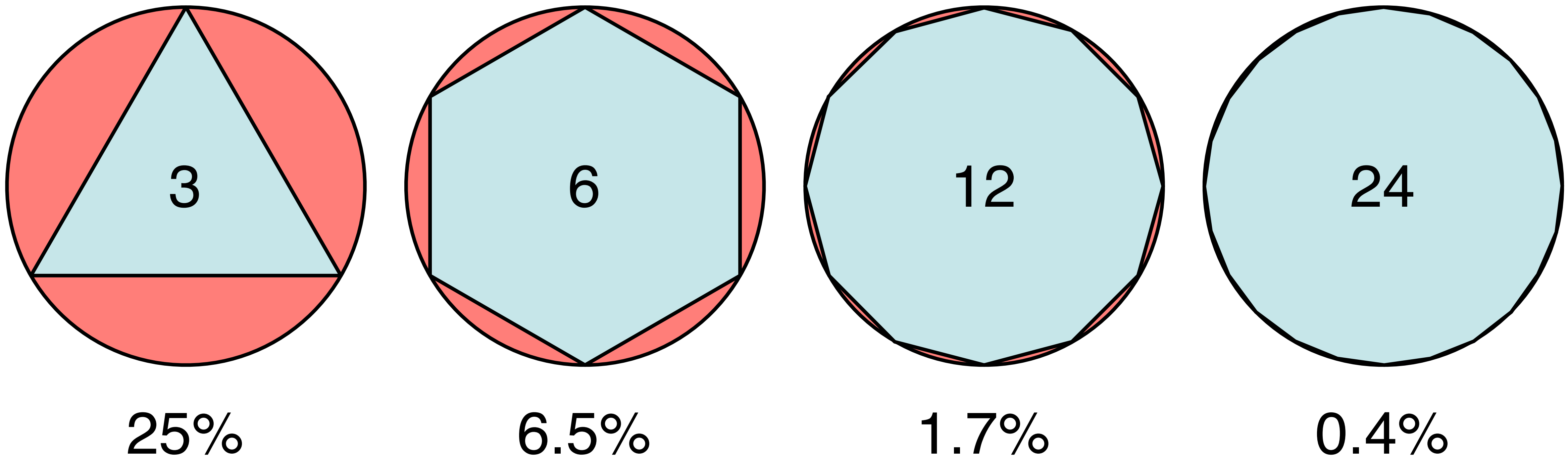
$$|\mathcal{V}| - |\mathcal{E}| + |\mathcal{F}| = 2(1-g)$$

- Mesh statistics
  - $|\mathcal{F}| \approx 2 \cdot |\mathcal{V}|$
  - $|\mathcal{E}| \approx 3 \cdot |\mathcal{V}|$
  - Avg. valence ≈ 6



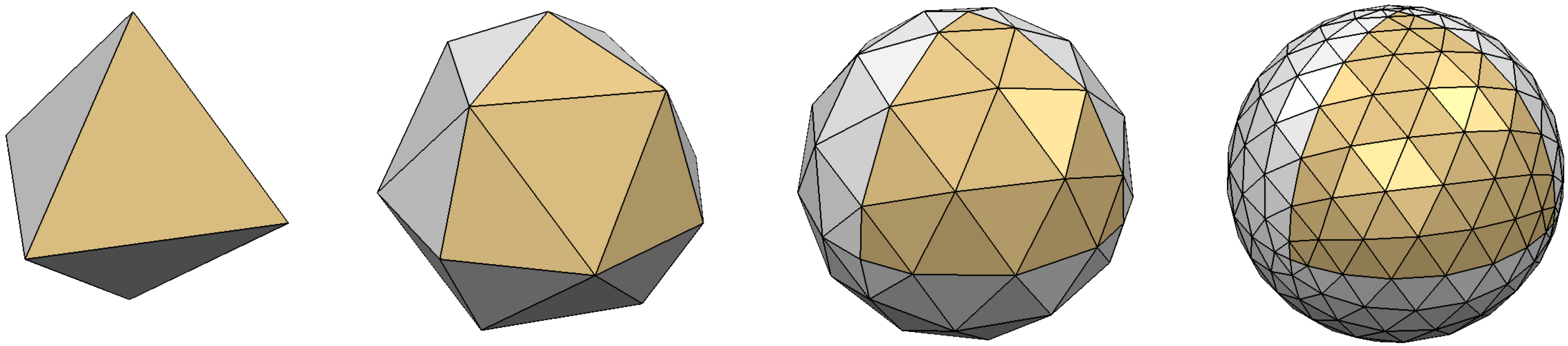
Triangle Meshes

- Piecewise linear approximation
  - Error is $O(h^{-2})$



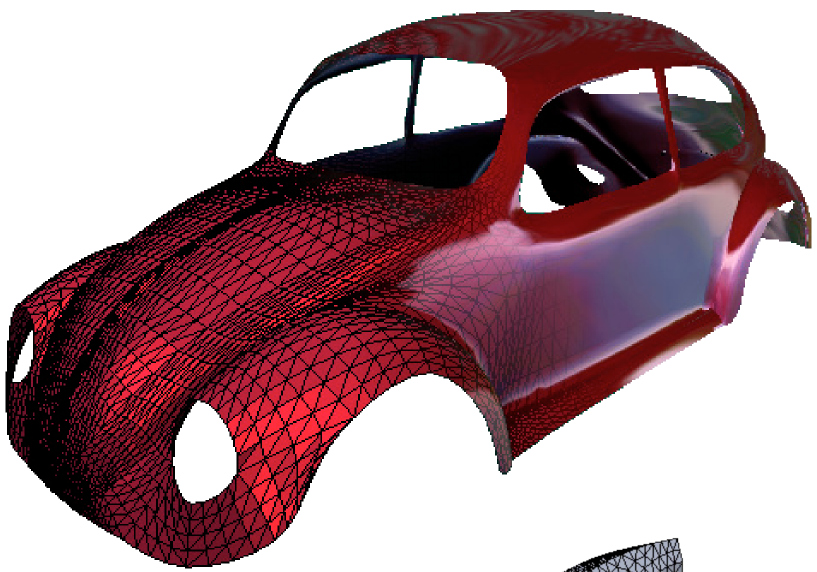
Triangle Meshes

- Piecewise linear approximation
  - Error is $O(h^{-2})$
  - |V| inversely proportional to error

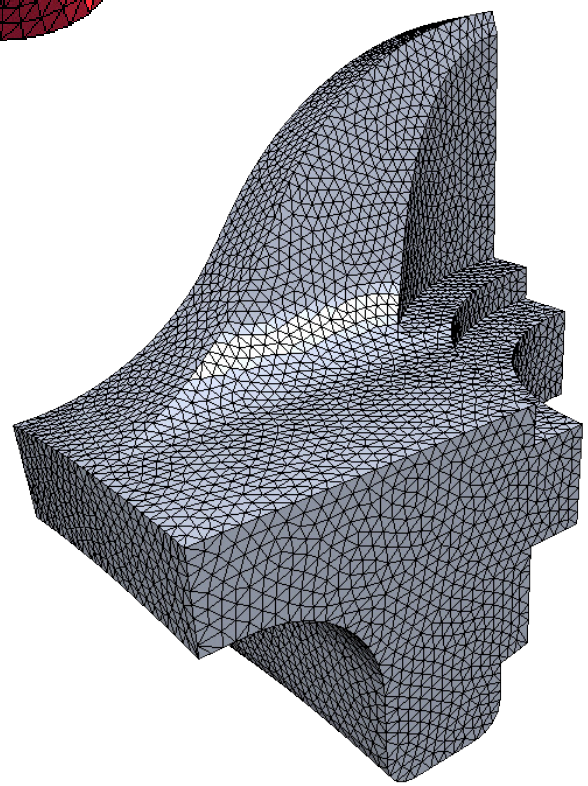


Triangle Meshes

- Highly flexible
  - Arbitrary surface topology
  - Smooth surfaces, sharp features



- Highly efficient
  - Simplest surface primitive
  - GPU accelerated rendering



Mesh Data Structures

- How to store geometry & connectivity?
- Compact storage
  - File formats
- Efficient algorithms on meshes
  - Identify time-critical operations
  - All vertices/edges of a face
  - All incident vertices/edges/faces of a vertex

Face Set (STL)

- Face:
  - 3 positions

Triangles		
x_{11} y_{11} z_{11}	x_{12} y_{12} z_{12}	x_{13} y_{13} z_{13}
x_{21} y_{21} z_{21}	x_{22} y_{22} z_{22}	x_{23} y_{23} z_{23}
...	...	...
x_{F1} y_{F1} z_{F1}	x_{F2} y_{F2} z_{F2}	x_{F3} y_{F3} z_{F3}

36 B/f = 72 B/v

No connectivity!

Shared Vertex (OBJ, OFF)

- Vertex:
  - Position
- Face:
  - Vertices

Vertices
x_1 y_1 z_1
...
x_V y_V z_V

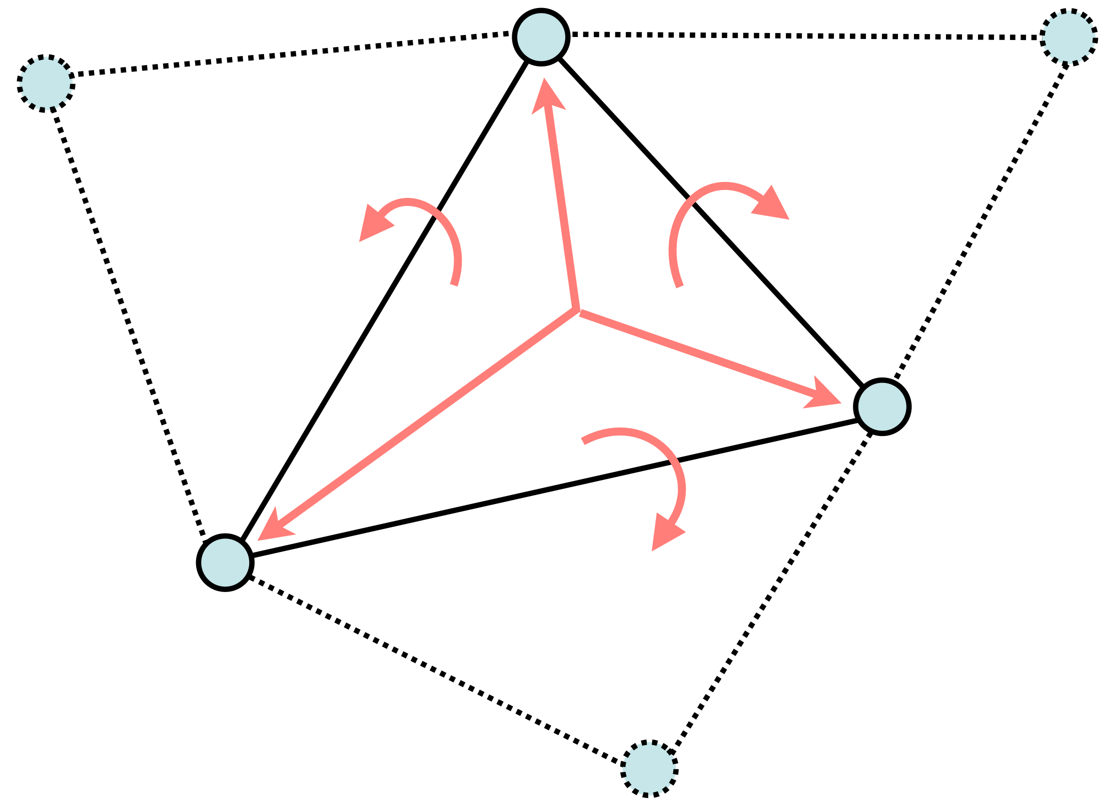
Triangles
v_{11} v_{12} v_{13}
...
...
...
...
v_{F1} v_{F2} v_{F3}

12 B/v + 12 B/f = 36 B/v

No neighborhood info

Face-Based Connectivity

- Vertex:
  - Position
  - 1 face
- Face:
  - 3 vertices
  - 3 face neighbors

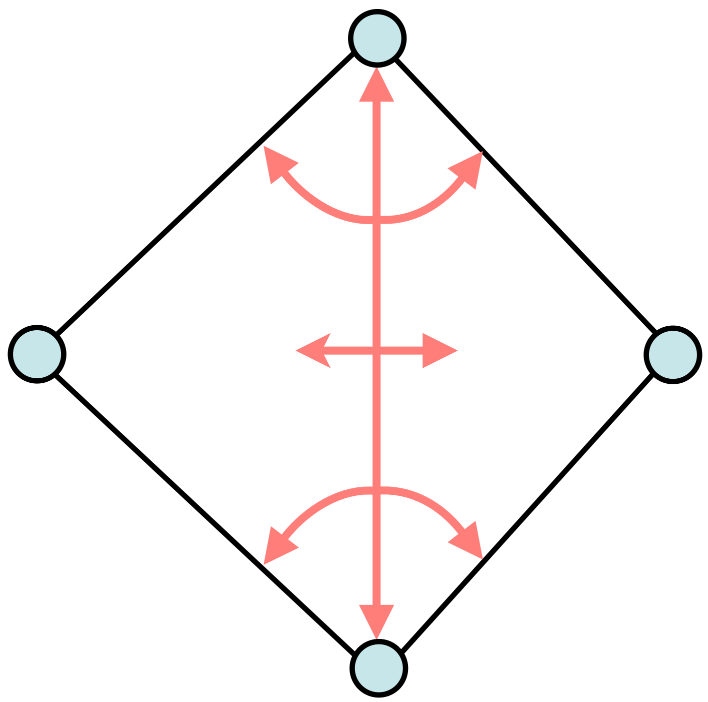


64 B/v

No edges!

Edge-Based Connectivity

- Vertex
  - Position
  - 1 edge
- Edge
  - 2 vertices
  - 2 faces
  - 4 edges
- Face
  - 1 edge

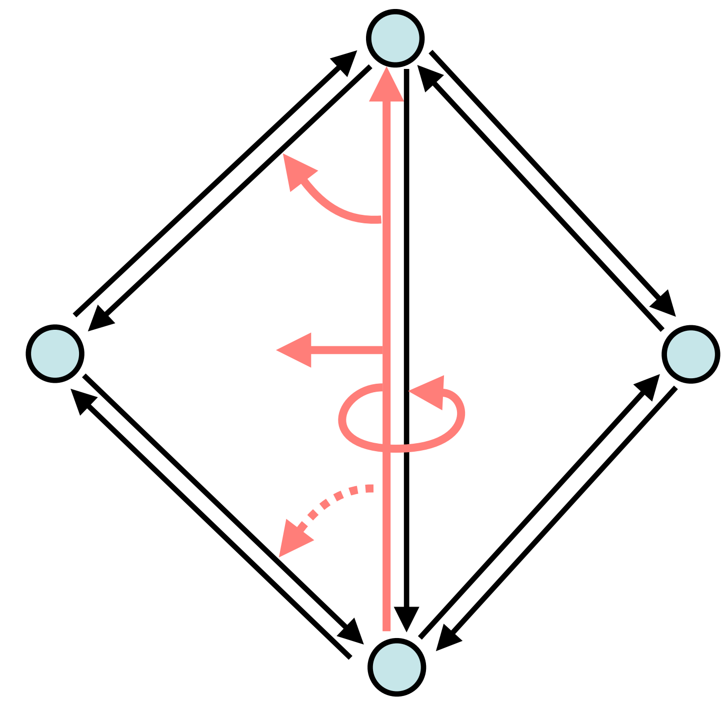


120 B/v

Edge orientation?

Halfedge-Based Connectivity

- Vertex
  - Position
  - 1 halfedge
- Halfedge
  - 1 vertex
  - 1 face
  - 2 or 3 halfedges
- Face
  - 1 halfedge

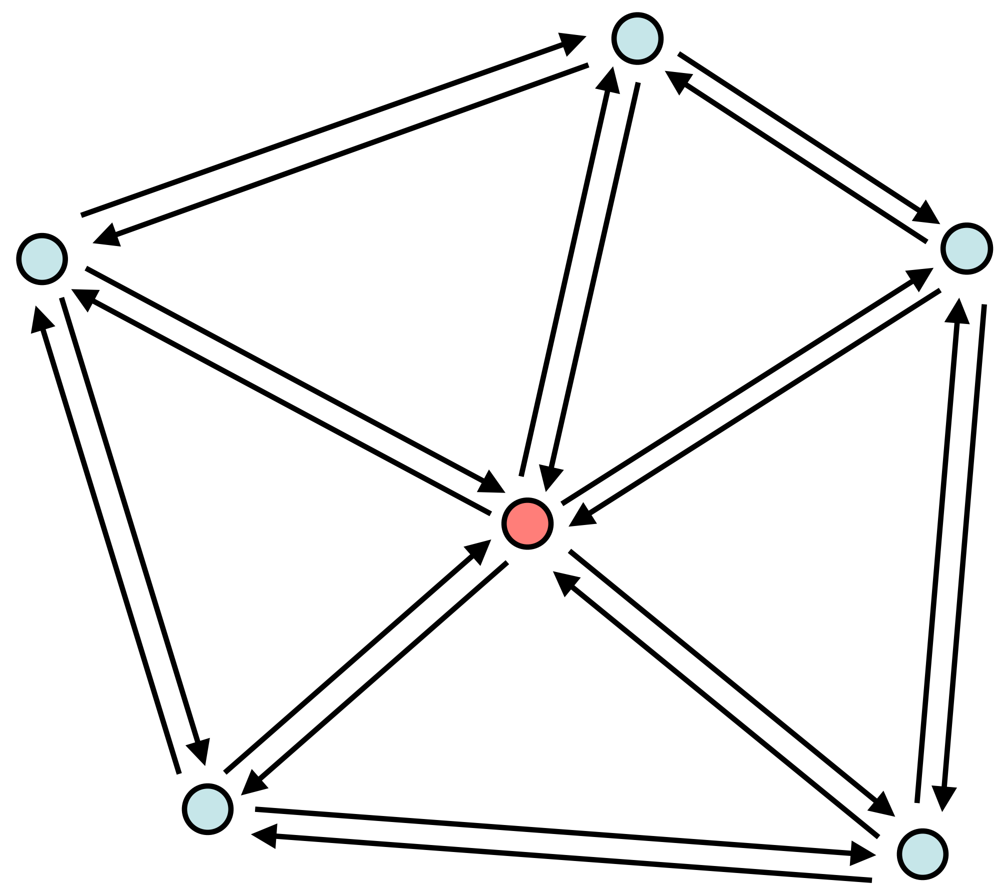


120 B/v

No case distinctions during traversal

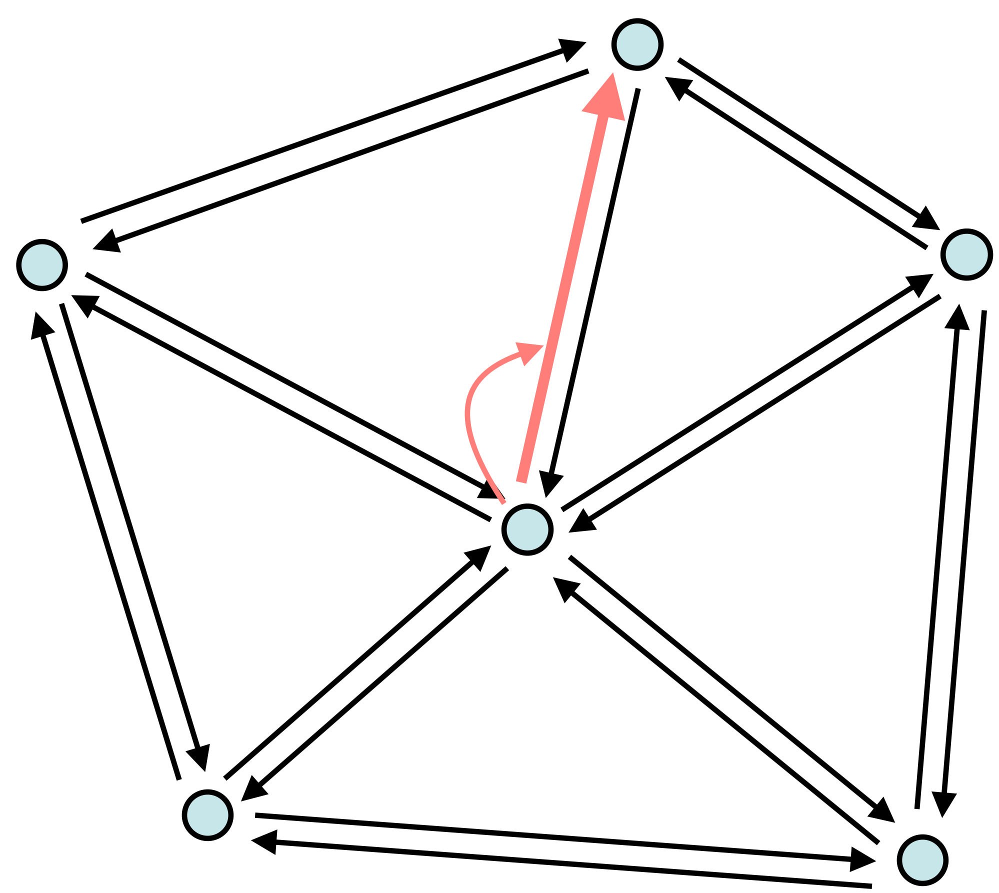
One-Ring Traversal

1. Start at vertex



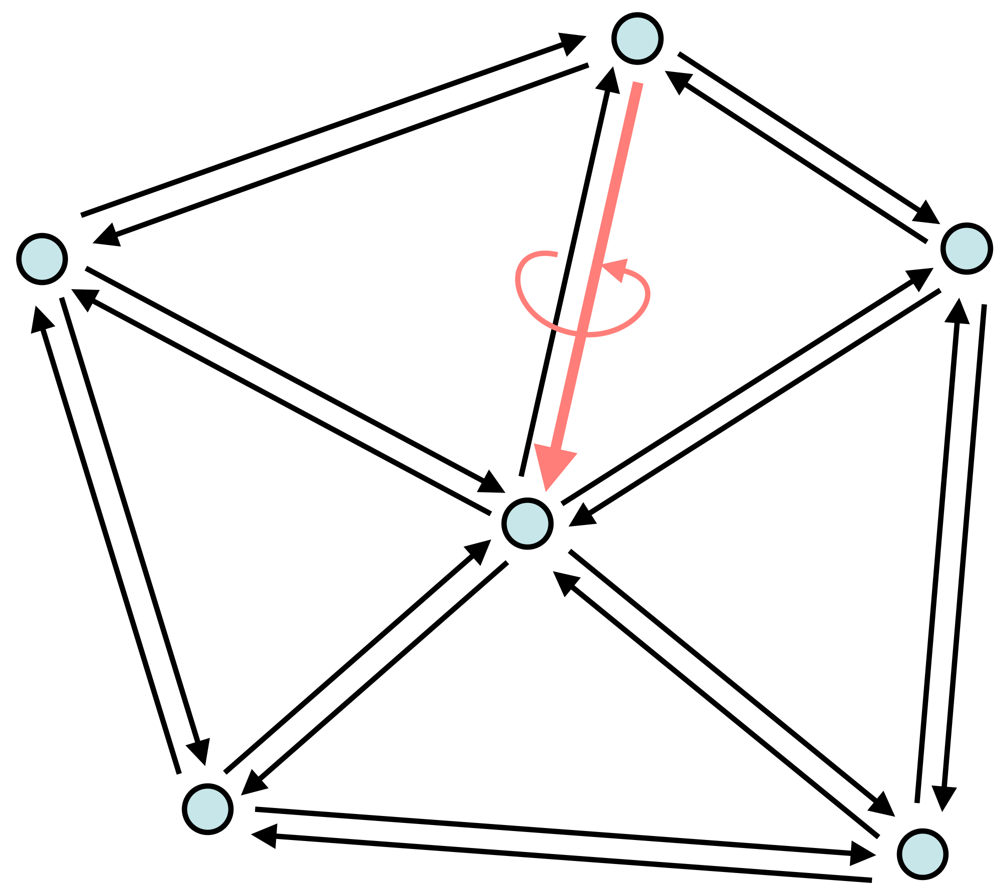
One-Ring Traversal

1. Start at vertex
2. Outgoing halfedge



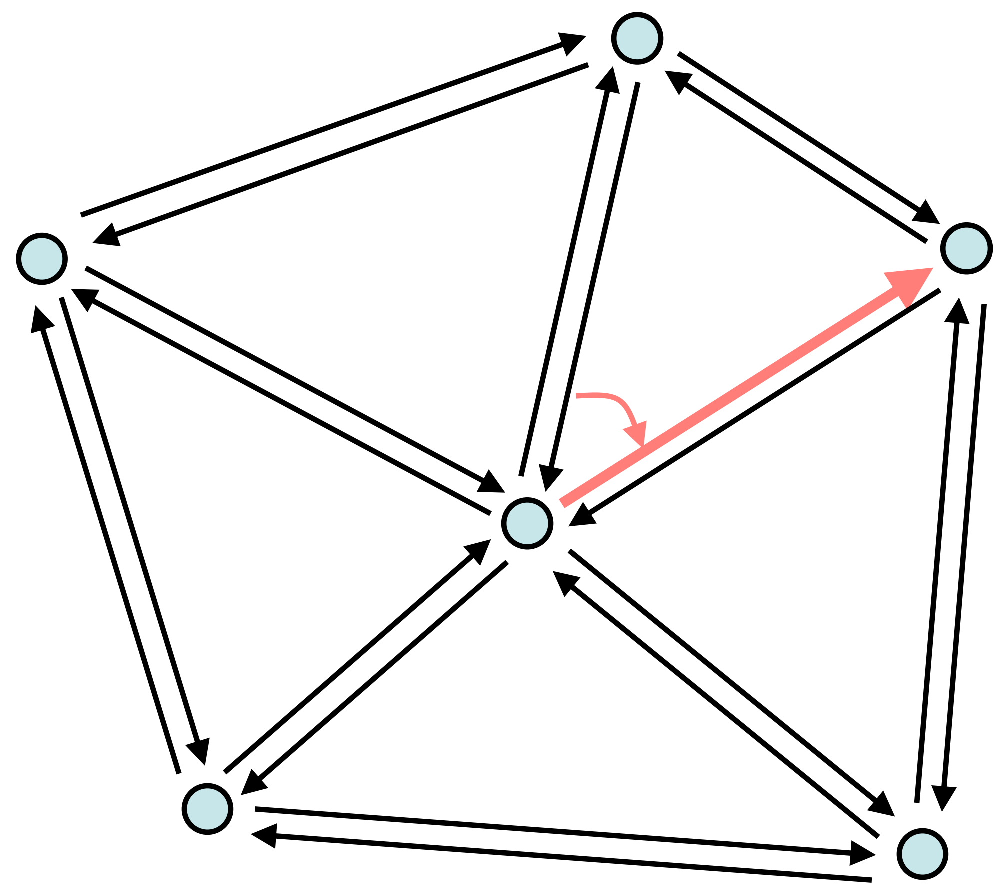
One-Ring Traversal

1. Start at vertex
2. Outgoing halfedge
3. Opposite halfedge



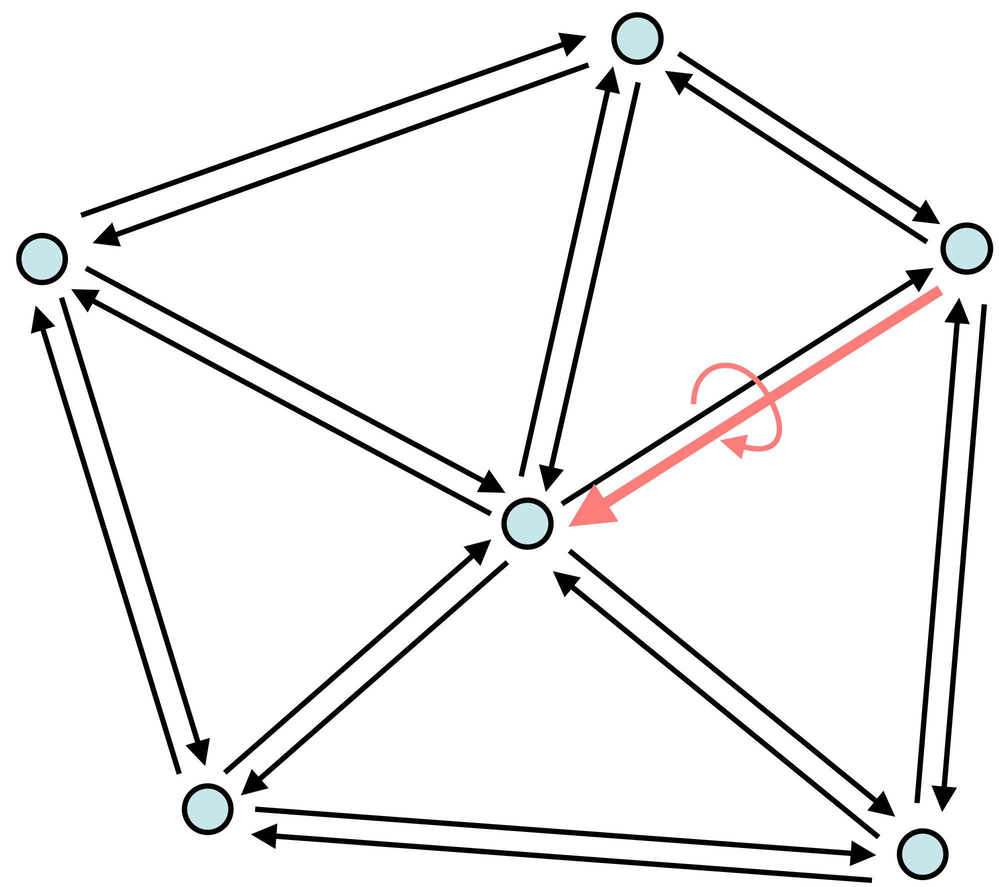
One-Ring Traversal

1. Start at vertex
2. Outgoing halfedge
3. Opposite halfedge
4. Next halfedge



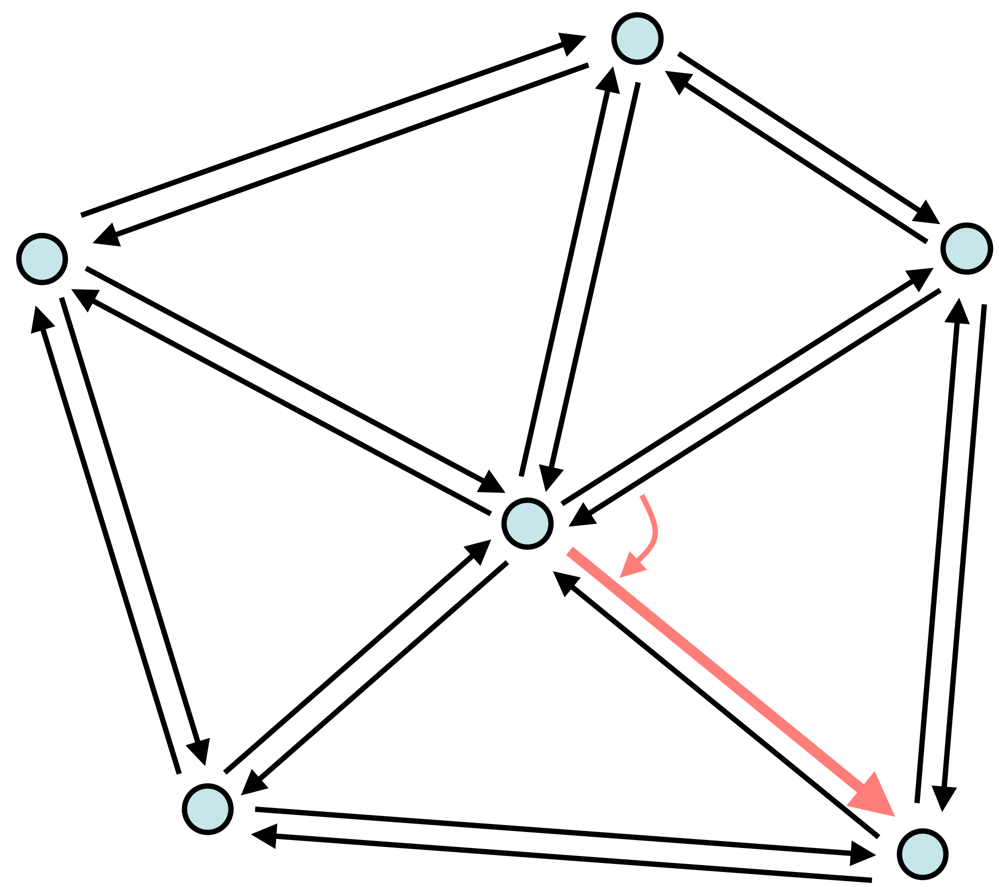
One-Ring Traversal

1. Start at vertex
2. Outgoing halfedge
3. Opposite halfedge
4. Next halfedge
5. Opposite



One-Ring Traversal

1. Start at vertex
2. Outgoing halfedge
3. Opposite halfedge
4. Next halfedge
5. Opposite
6. Next
7. ...



Halfedge-Based Libraries

- CGAL
  - `www.cgal.org`
  - Computational geometry
  - Free for non-commercial use

- OpenMesh
  - `www.openmesh.org`
  - Mesh processing
  - Free, LGPL licence

Literature

- Kettner, *Using generic programming for designing a data structure for polyhedral surfaces*, Symp. on Comp. Geom., 1998
- Campagna et al, *Directed Edges - A Scalable Representation for Triangle Meshes*, Journal of Graphics Tools 4(3), 1998
- Botsch et al, *OpenMesh - A generic and efficient polygon mesh data structure*, OpenSG Symp. 2002

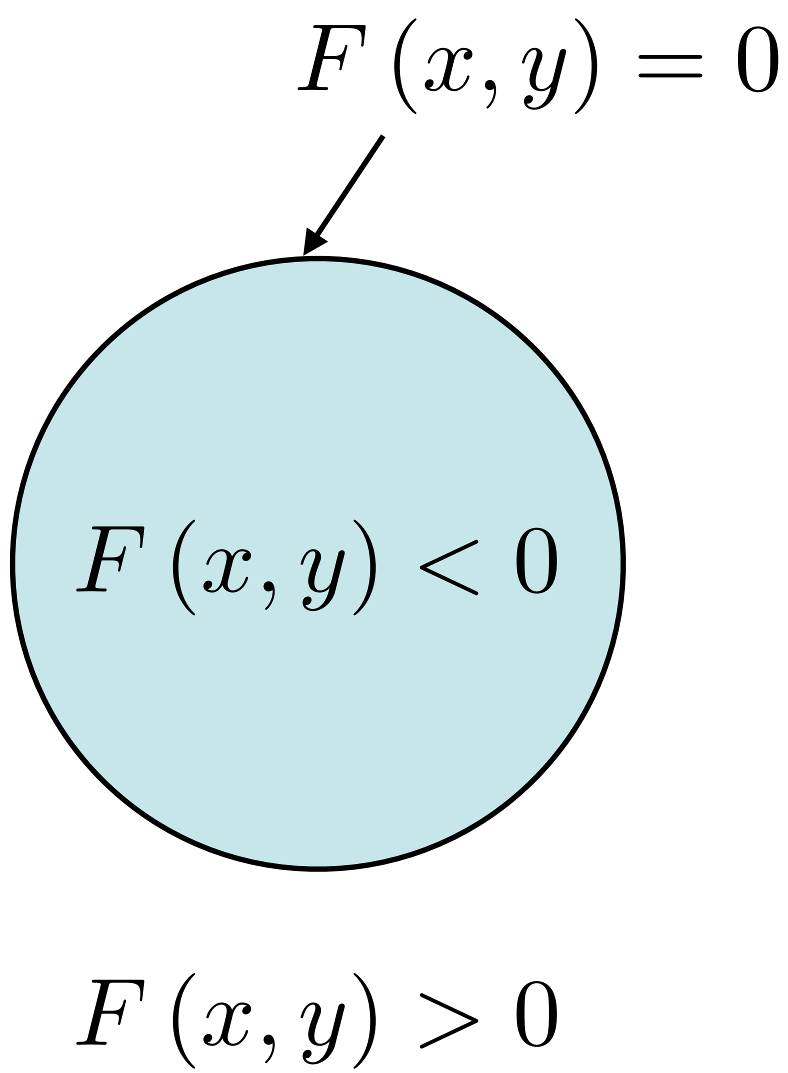
Outline

- Surface Representations
  - Explicit vs. Implicit
- Explicit Representation
  - Triangle Meshes
- Implicit Representations
  - Signed Distance Functions
- Conversions
  - Implicit ↔ Explicit

Implicit Representations

- General implicit function:
  - Interior: F(x,y,z) < 0
  - Exterior: F(x,y,z) > 0
  - Surface: F(x,y,z) = 0

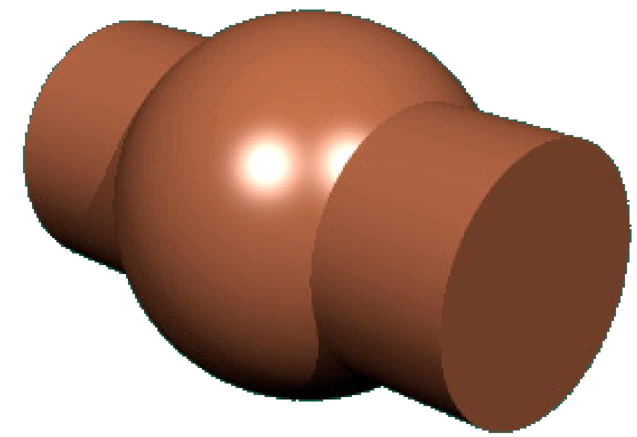
- Special case
  - Signed distance function (SDF)



Constructive Solid Geometry

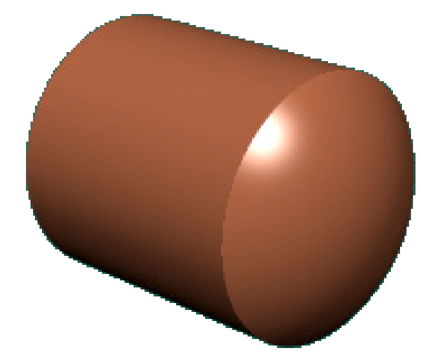
- Union

$$F_{C\cup S}(\cdot) = \min\left\{F_C(\cdot)\,, F_S(\cdot)\right\}$$



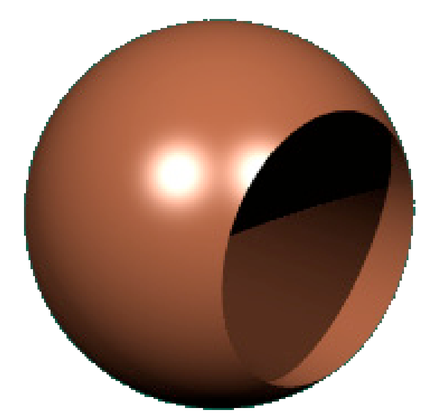
- Intersection

$$F_{C\cap S}(\cdot) = \max\left\{F_C(\cdot)\,, F_S(\cdot)\right\}$$

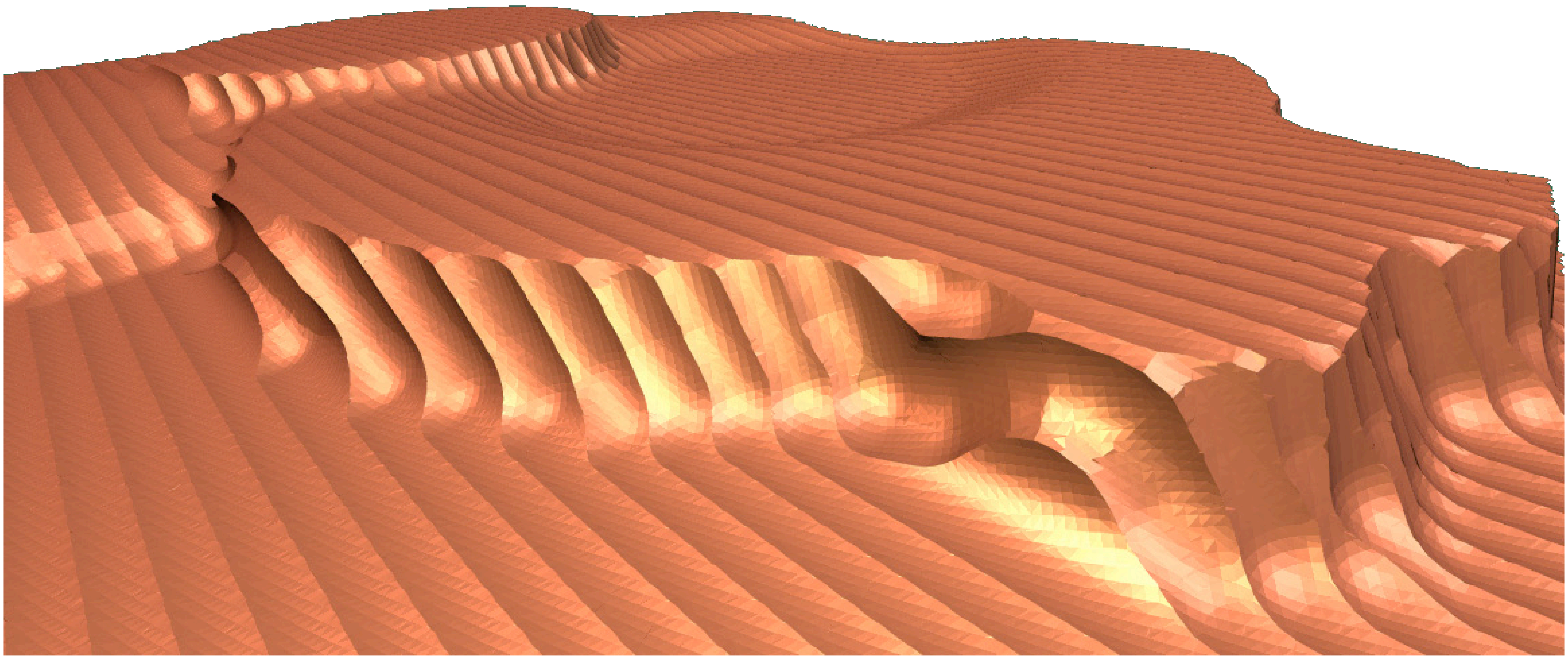


- Difference

$$F_{C\setminus S}(\cdot) = \max\left\{F_C(\cdot)\,, -F_S(\cdot)\right\}$$

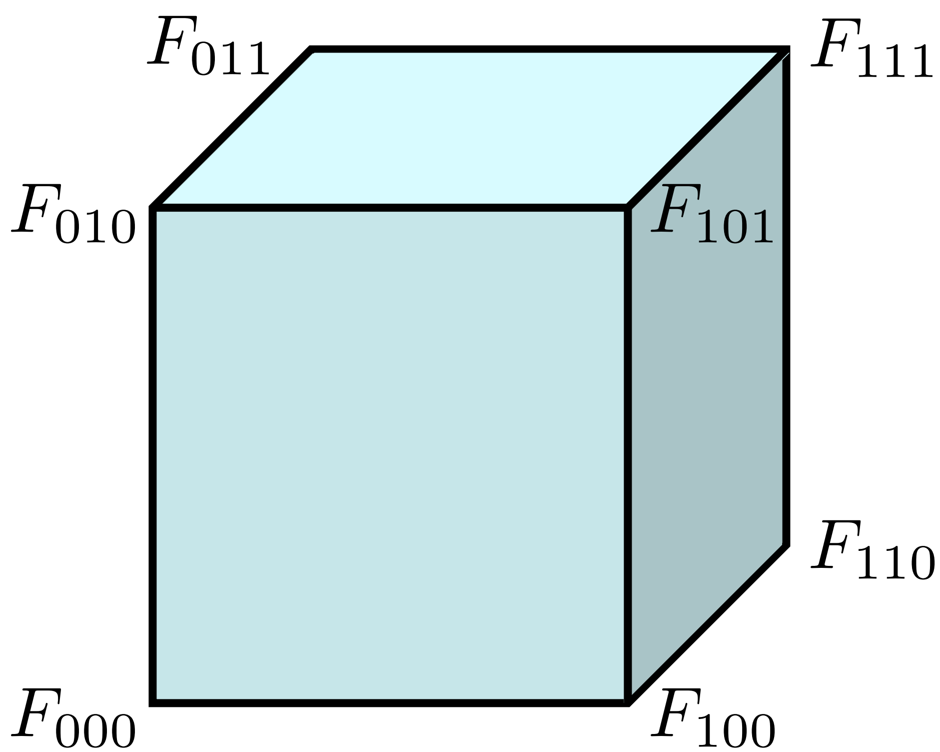


CSG Example: Milling



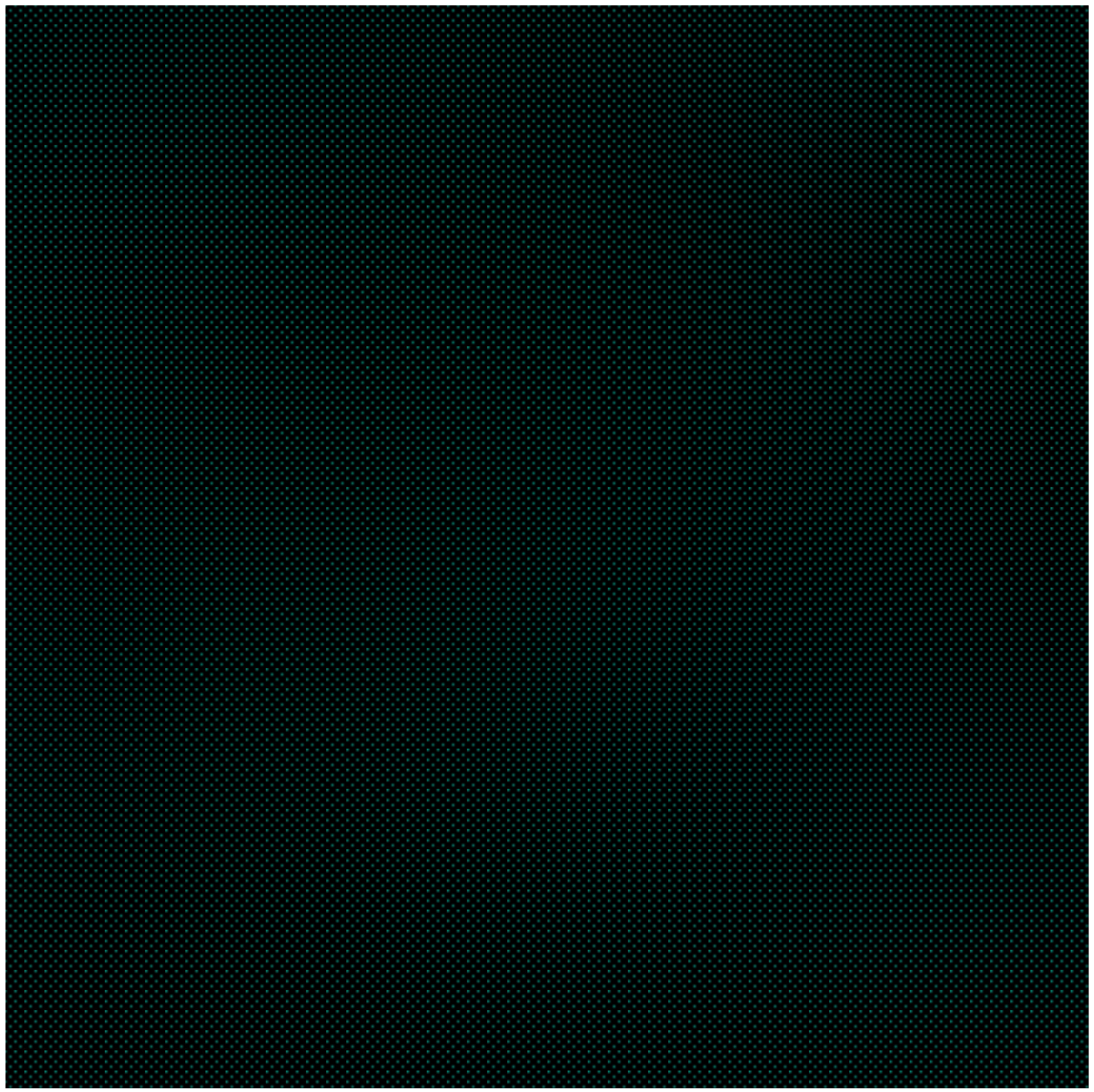
SDF Discretization

- Regular cartesian 3D grid
  - Compute signed distance at nodes
  - Tri-linear interpolation within cells

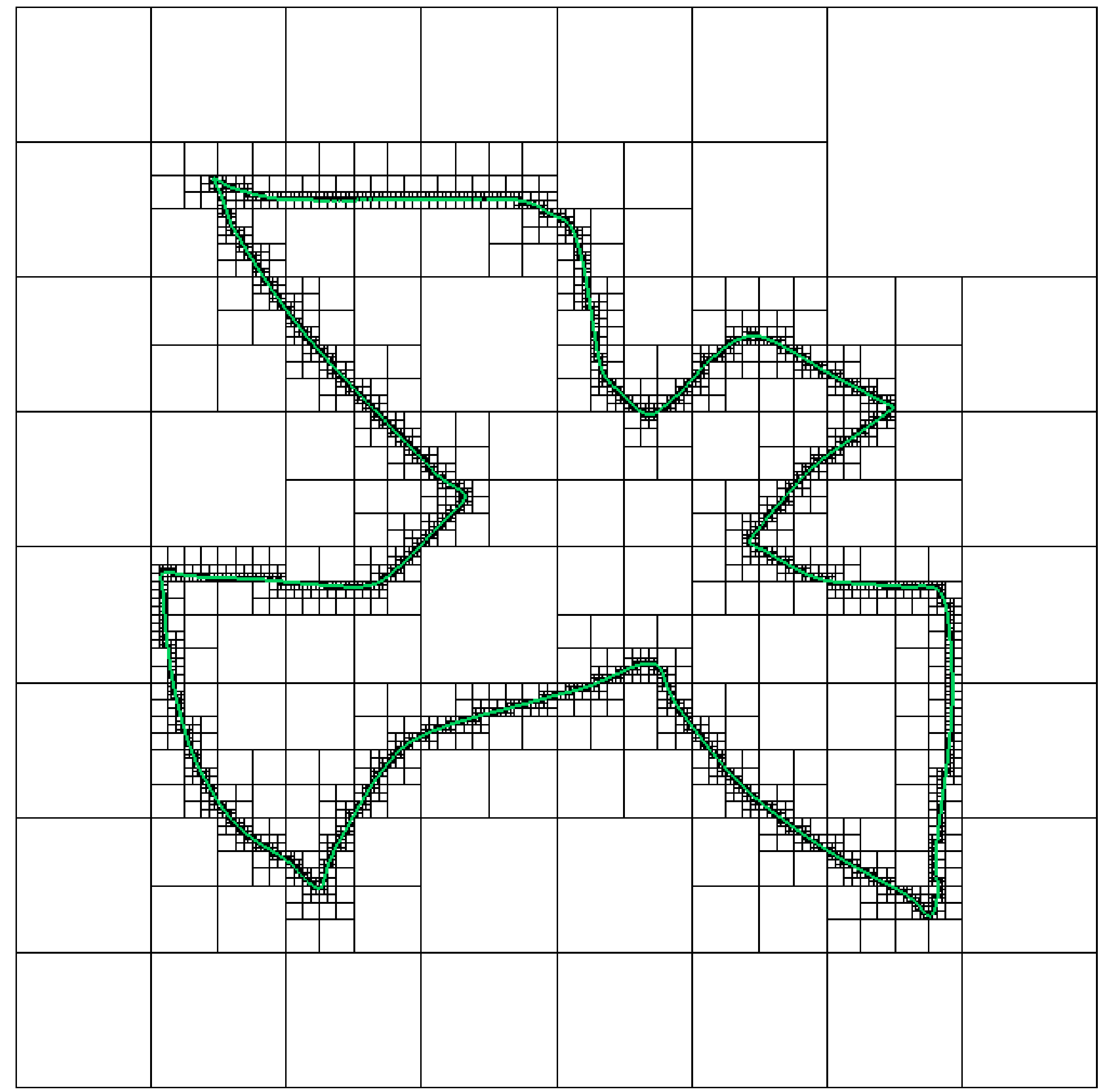


$$
\begin{array}{lrrrl}
F_{000} & (1-u) & (1-v) & (1-w) & + \\
F_{100} & u & (1-v) & (1-w) & + \\
F_{010} & (1-u) & v & (1-w) & + \\
F_{001} & (1-u) & (1-v) & w & + \\
\vdots & & & & \\
F_{111} & u & v & w &
\end{array}
$$

3-Color Octree



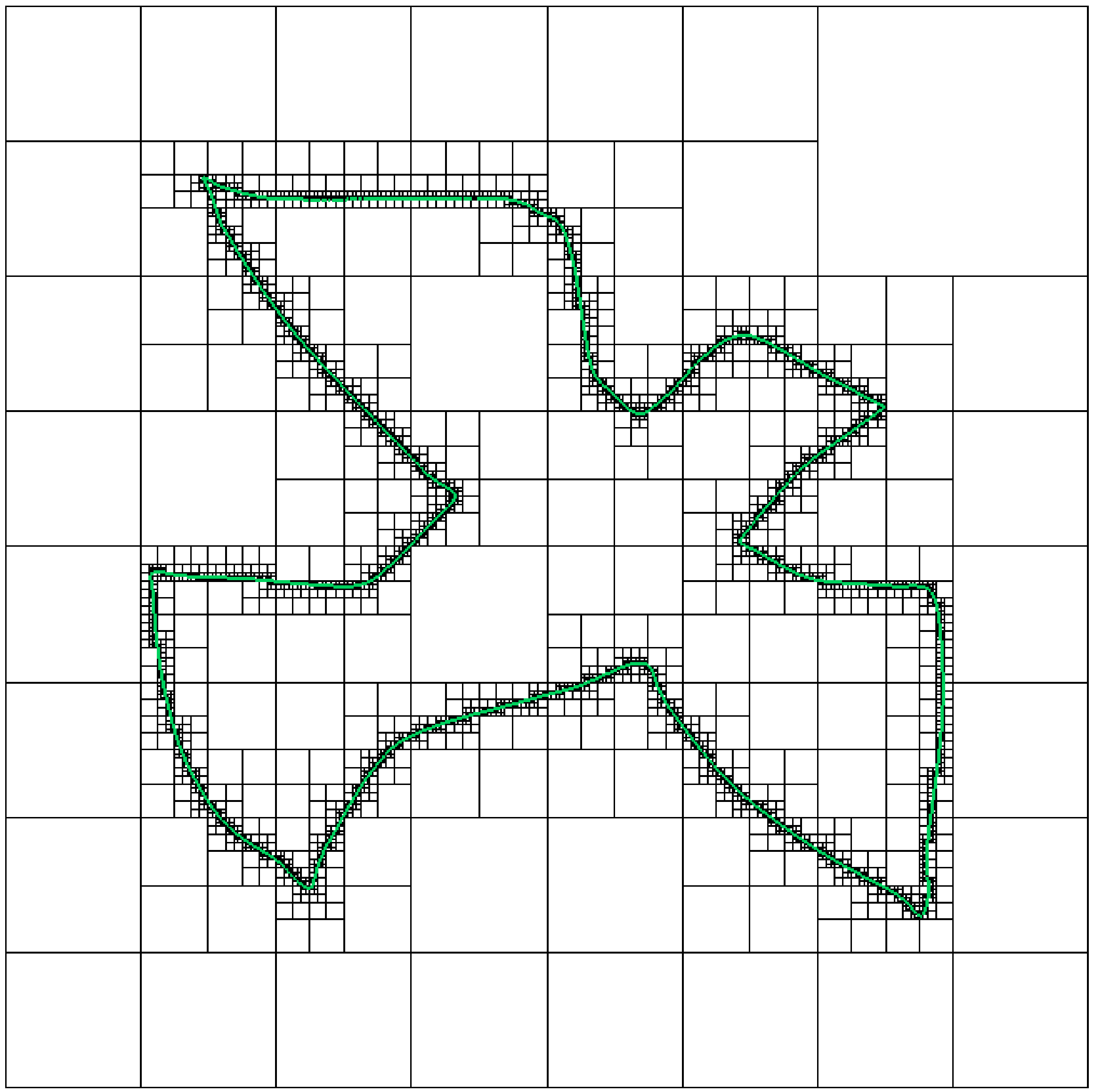
1048576 cells



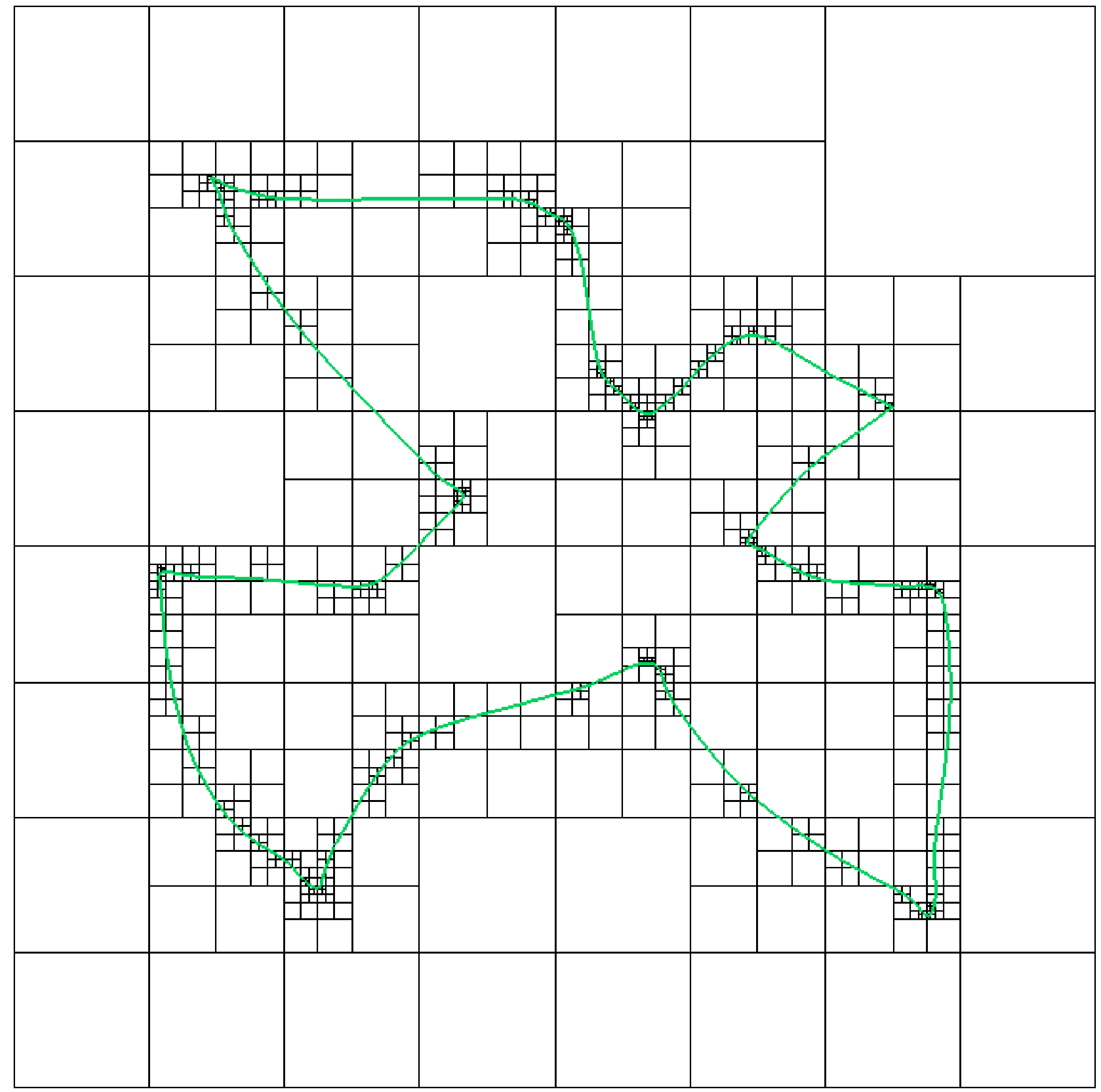
12040 cells

[Wu, Kobbelt, VMV 2003]

Adaptively Sampled Dist. Fields



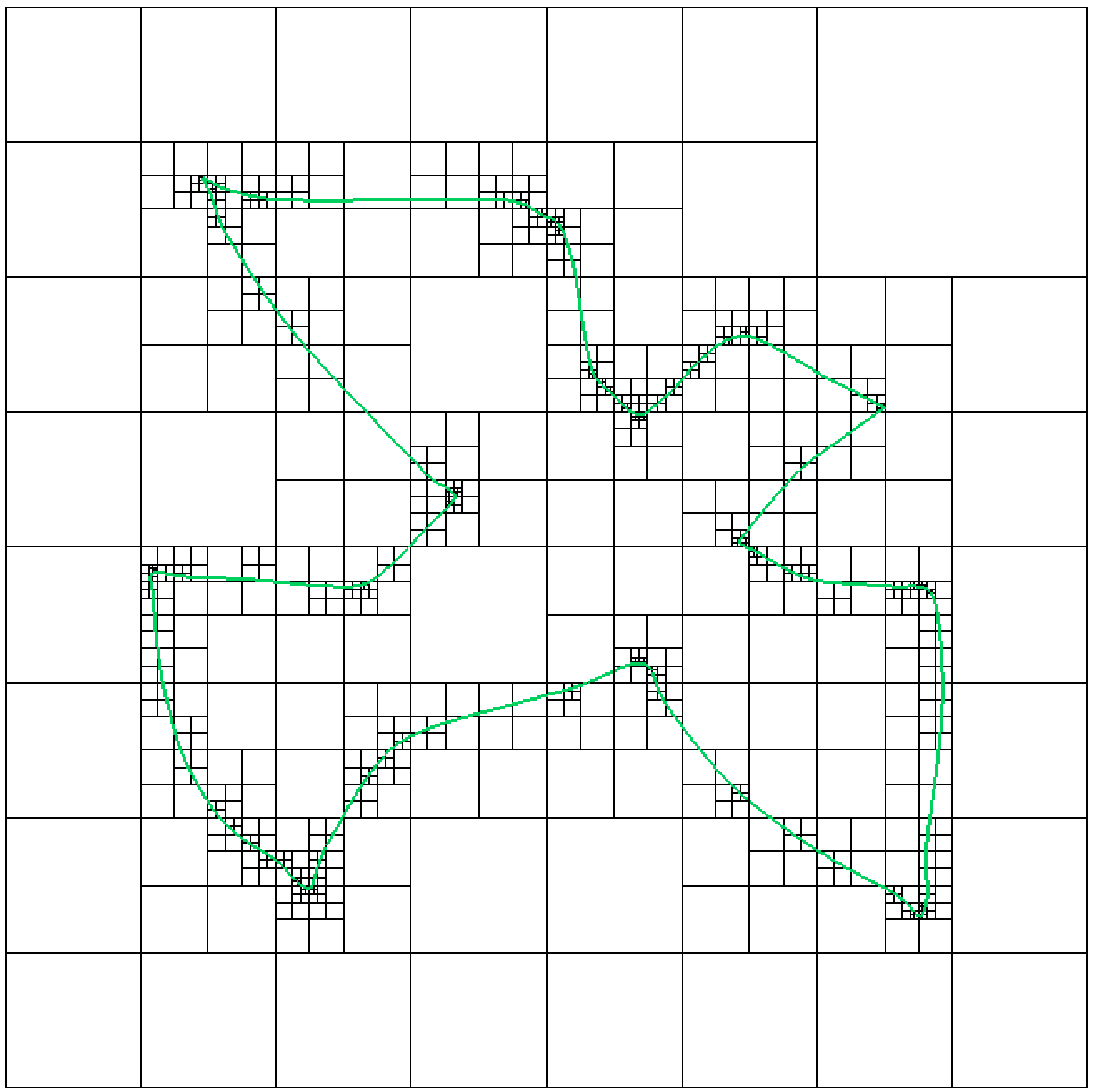
12040 cells



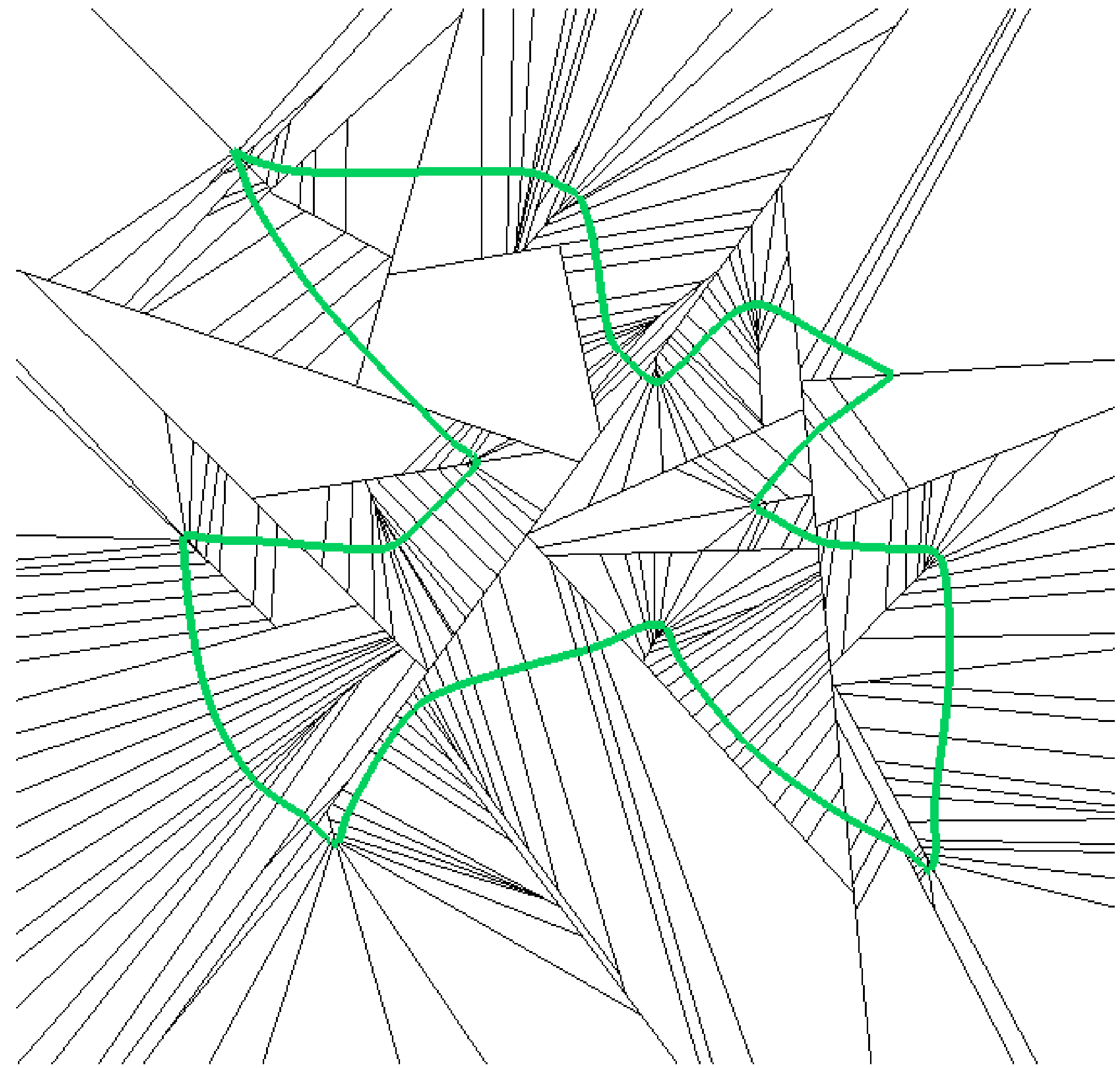
895 cells

[Wu, Kobbelt, VMV 2003]

Binary Space Partitions



895 cells



254 cells

[Wu, Kobbelt, VMV 2003]

Literature

- Frisken et al, *"Adaptively Sampled Distance Fields: A general representation of shape for computer graphics"*, SIGGRAPH 2000
- Wu & Kobbelt, *"Piecewise Linear Approximation of Signed Distance Fields"*, VMV 2003

Outline

- Surface Representations
  - Explicit vs. Implicit
- Explicit Representation
  - Triangle Meshes
- Implicit Representations
  - Signed Distance Functions
- Conversions
  - Implicit ↔ Explicit

Conversions

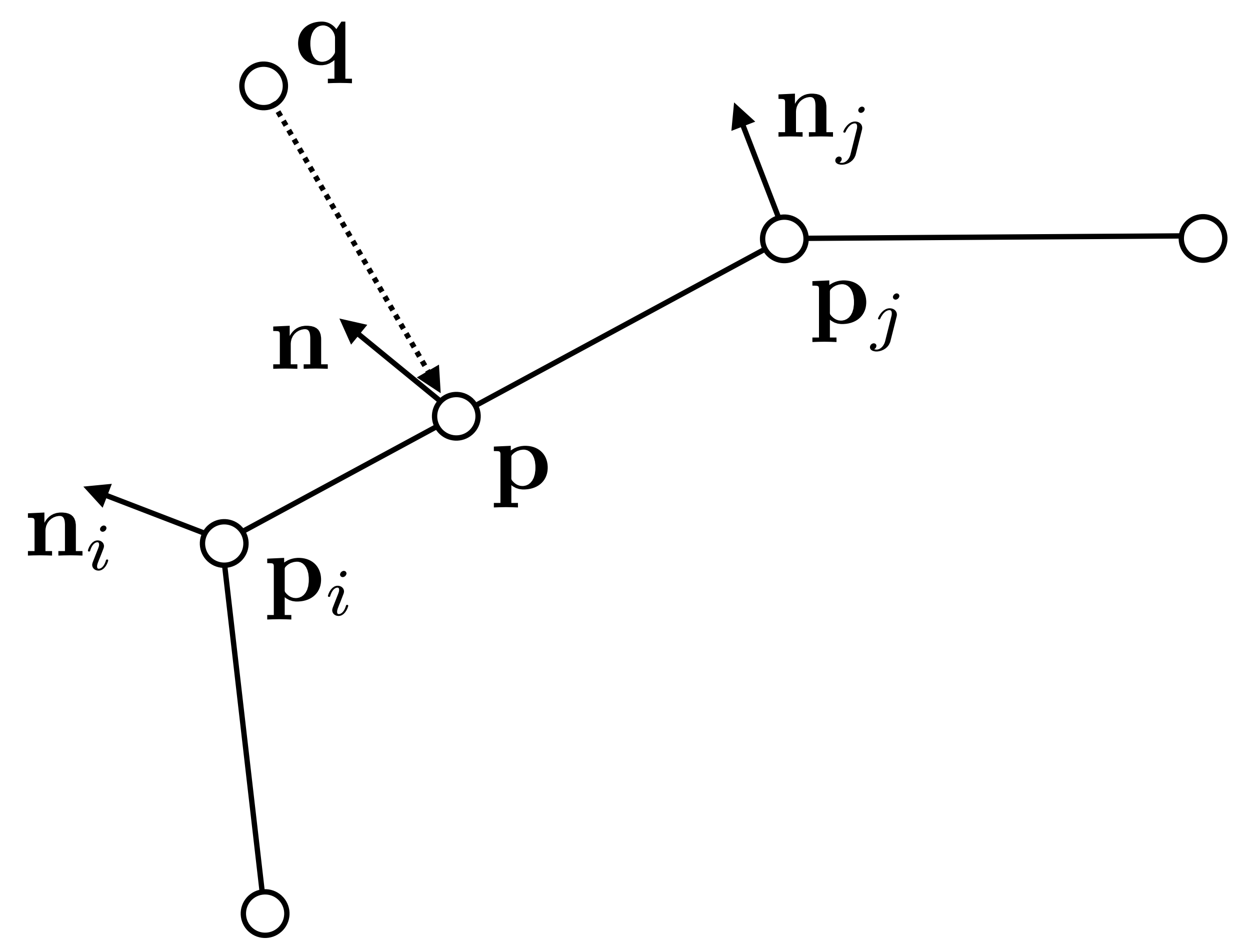
- Explicit to Implicit
  - Compute signed distance at grid points
  - Compute distance point-mesh
  - Fast marching
- Implicit to Explicit
  - Extract zero-level iso-surface $F(x,y,z)=0$
  - Other iso-surfaces $F(x,y,z)=C$
  - Medical imaging, simulations, measurements, ...

Signed Distance Computation

- Find closest mesh triangle
  - Use spatial hierarchies (octree, BSP tree)
- Distance Point-Triangle
  - Distance to plane/edge/vertex?
- Inside or outside?
  - Based on interpolated surface normals

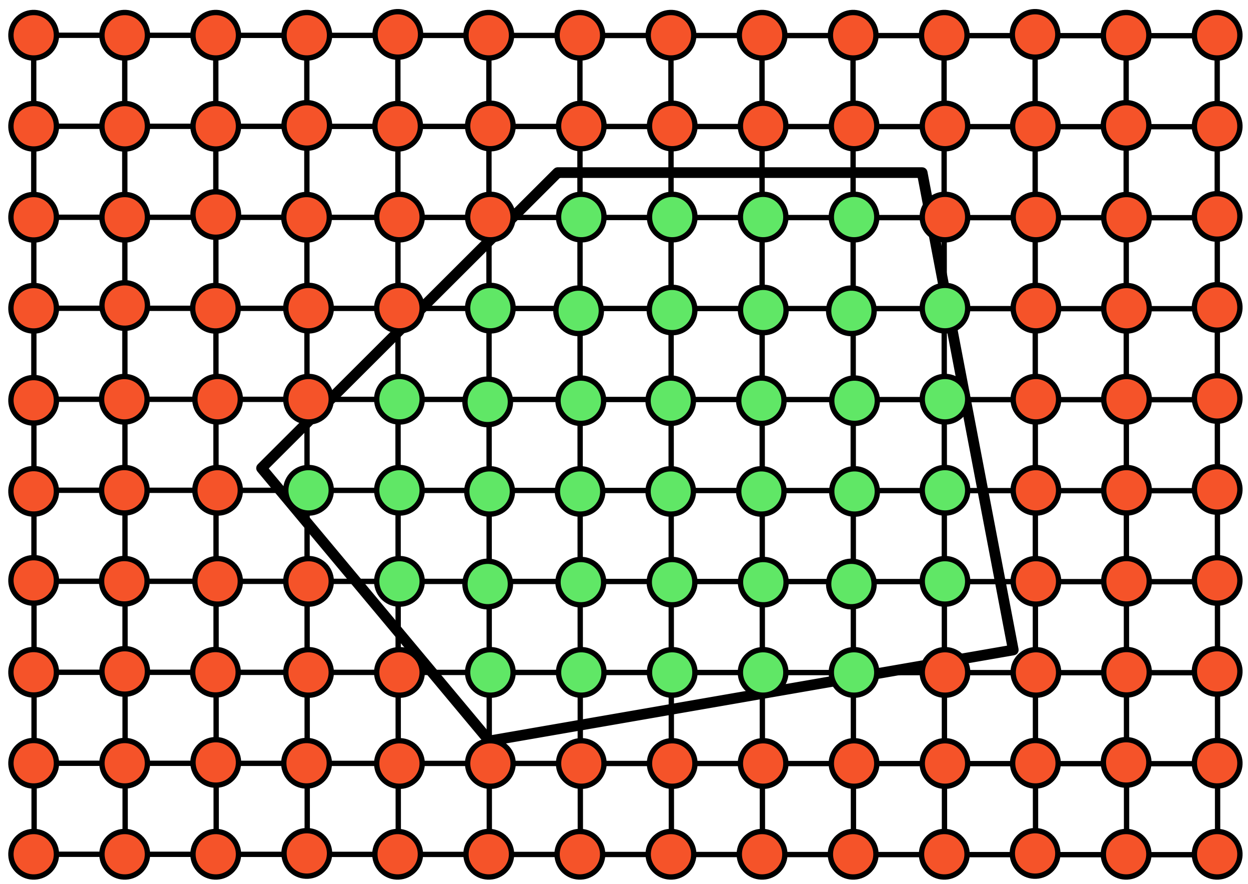
Signed Distance Computation

1. Closest point $\mathbf{p} = \alpha \mathbf{p}_i + (1-\alpha)\mathbf{p}_j$
2. Interpolated normal $\mathbf{n} = \alpha \mathbf{n}_i + (1-\alpha)\mathbf{n}_j$
3. Inside if $(\mathbf{q}-\mathbf{p})^T \mathbf{n} < 0$



Fast Marching Techniques

1. Initialize with exact distance in mesh's vicinity
2. Fast-march outwards
3. Fast-march inwards



Literature

- Schneider, Eberly, *"Geometric Tools for Computer Graphics"*, Morgan Kaufmann, 2002
- Sethian, *"Level Set and Fast Marching Methods"*, Cambridge University Press, 1999

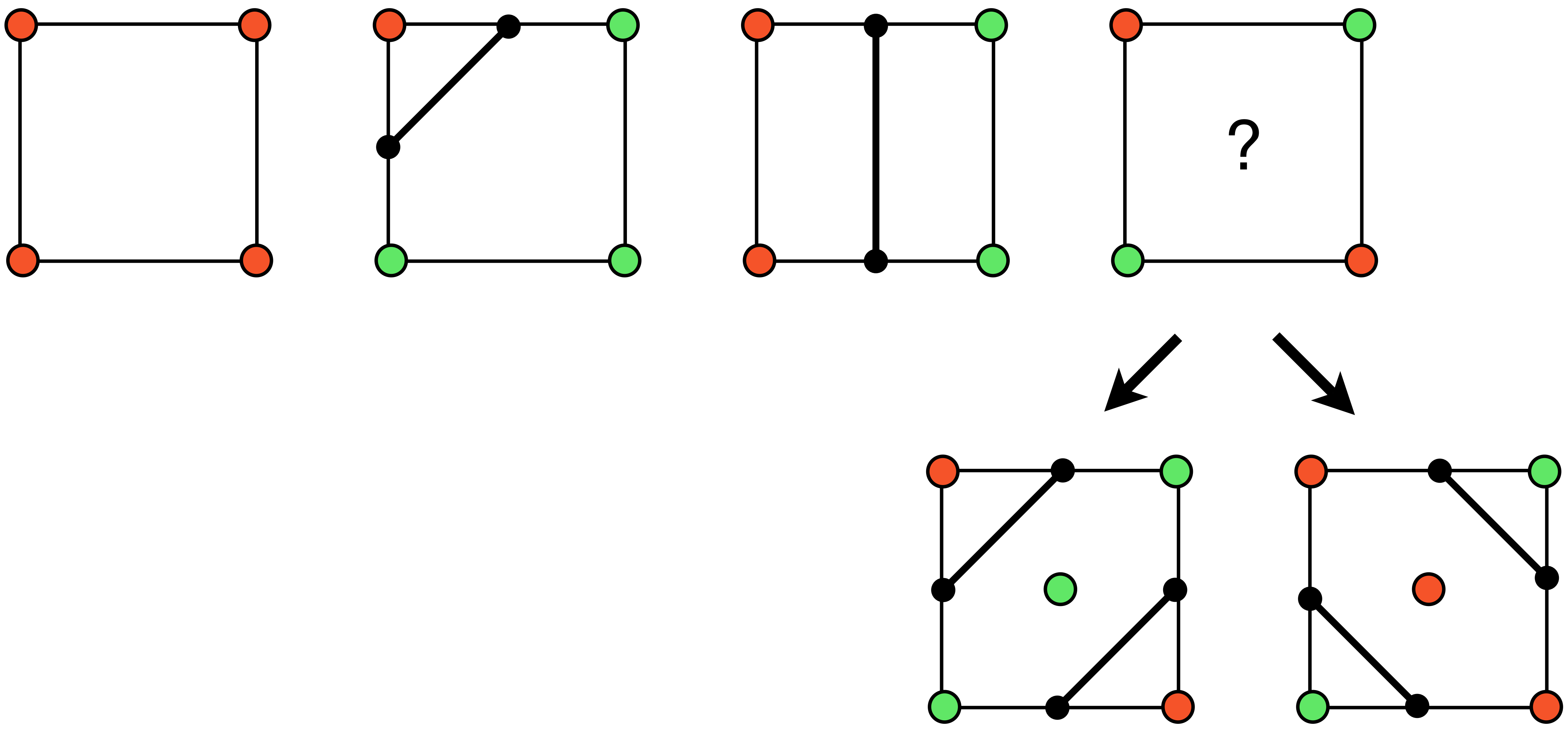
Conversions

- Explicit to Implicit
  - Compute signed distance at grid points
  - Compute distance point-mesh
  - Fast marching
- Implicit to Explicit
  - Extract zero-level iso-surface $F(x,y,z)=0$
  - Other iso-surfaces $F(x,y,z)=C$
  - Medical imaging, simulations, measurements, ...

2D: Marching Squares

- Classify grid nodes as inside/outside
- Classify cell: 16 configurations
- Linear interpolation along edges
- Look-up table for edge configuration

2D: Marching Squares

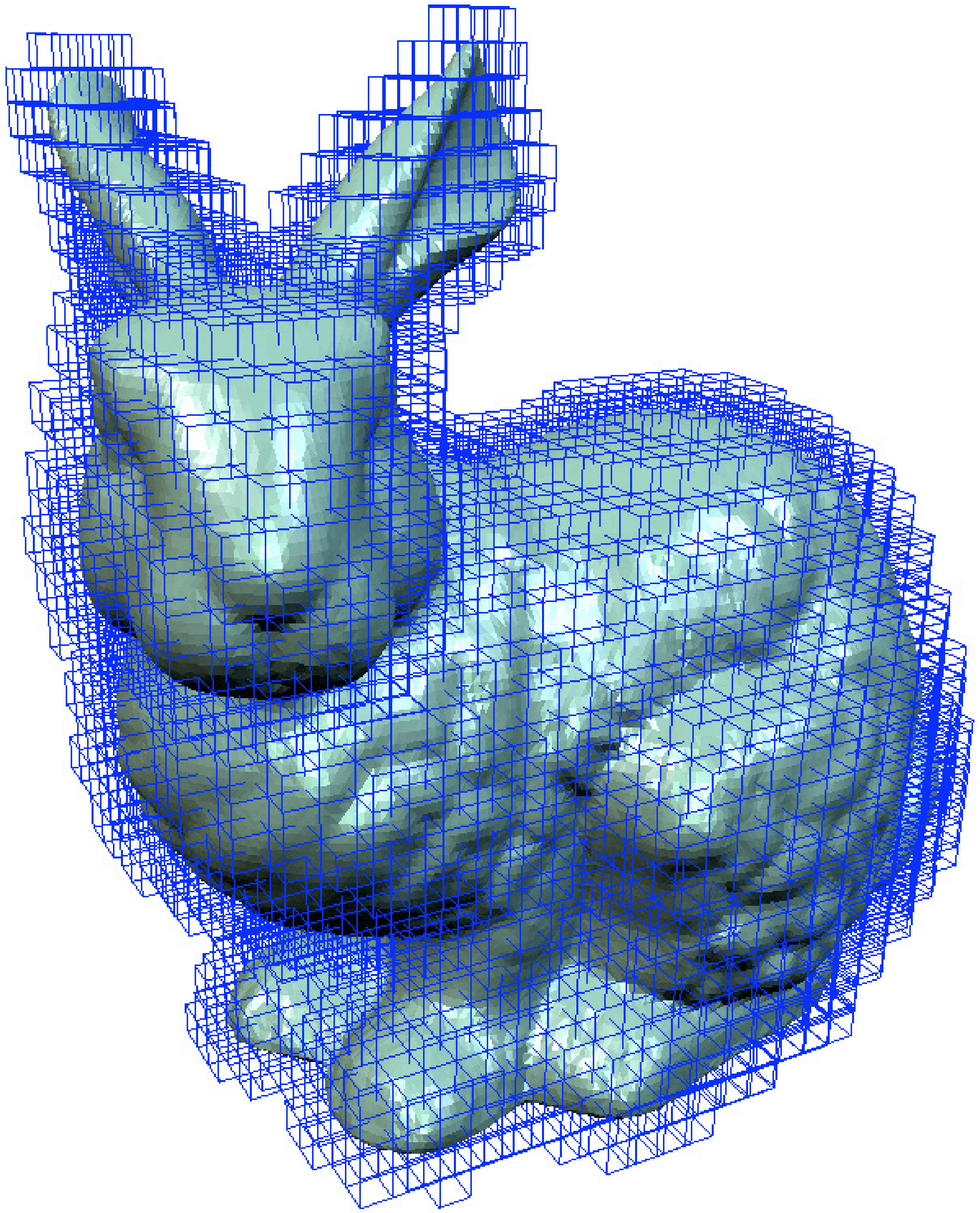


3D: Marching Cubes

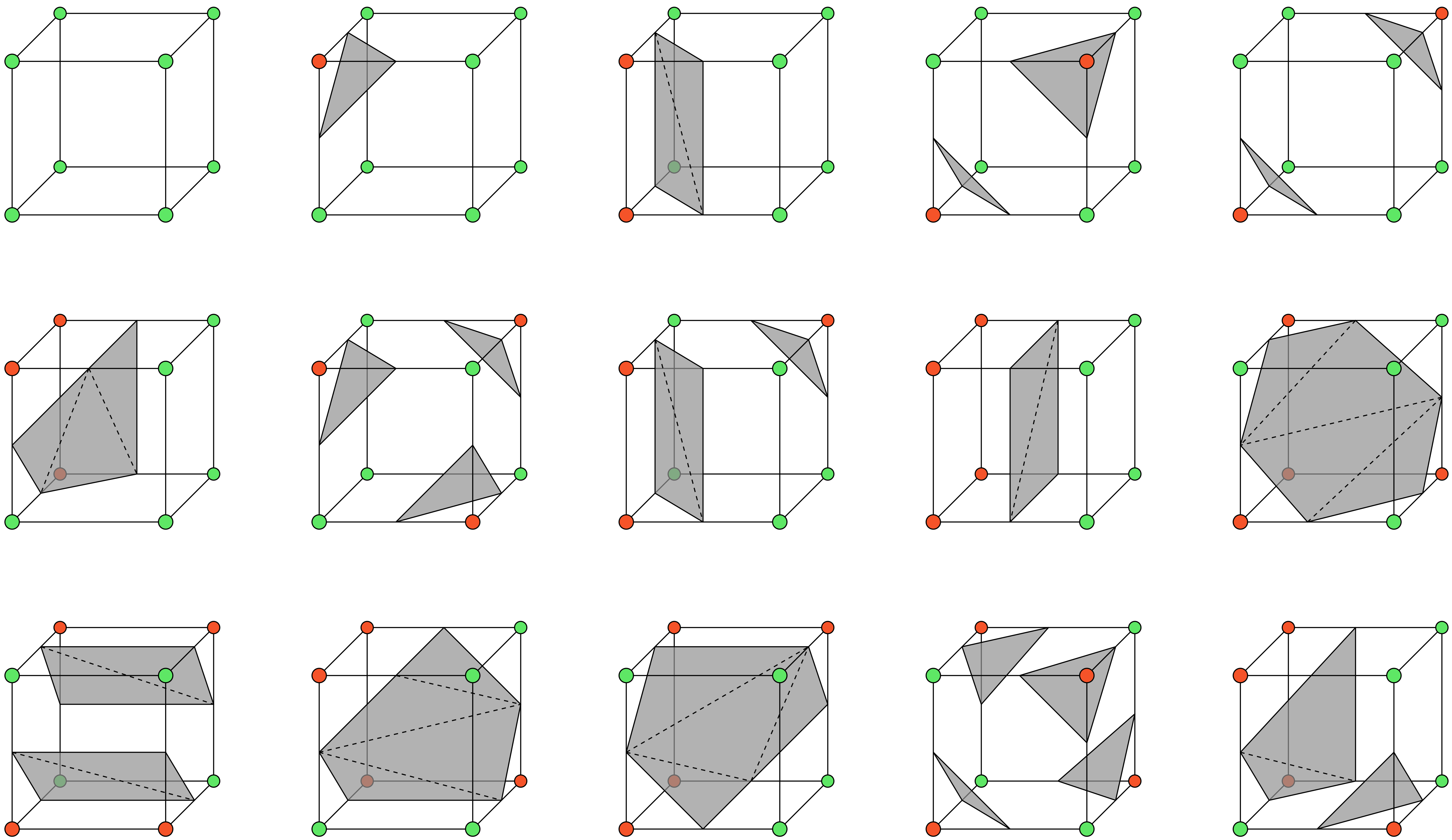
- Classify grid nodes as inside/outside
- Classify cell: 2^8 configurations
- Linear interpolation along edges
- Look-up table for patch configuration
  - Disambiguation more complicated

Marching Cubes

- Cell classification:
  - Inside
  - Outside
  - Intersecting

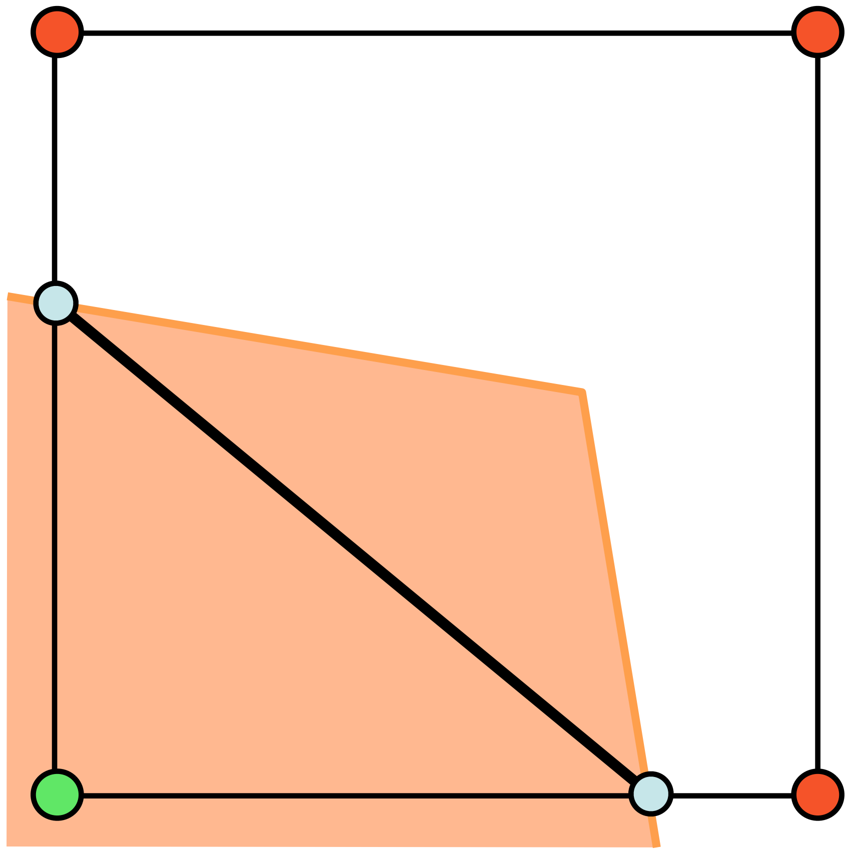


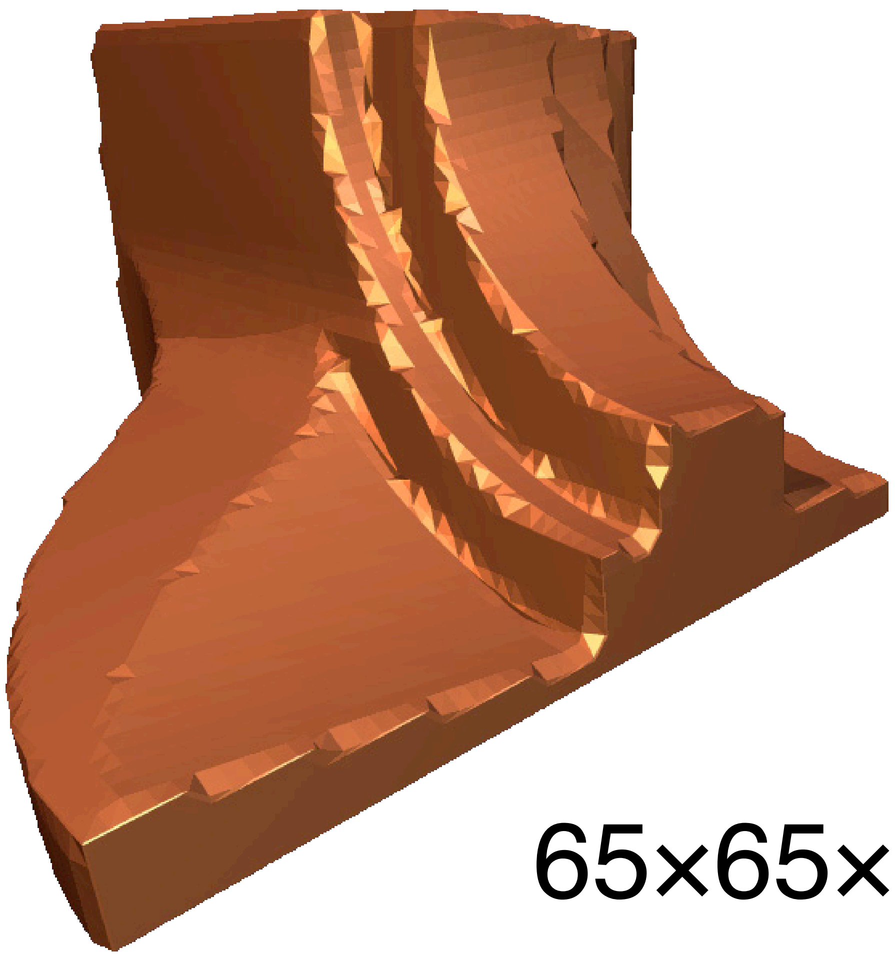
Marching Cubes



Marching Cubes

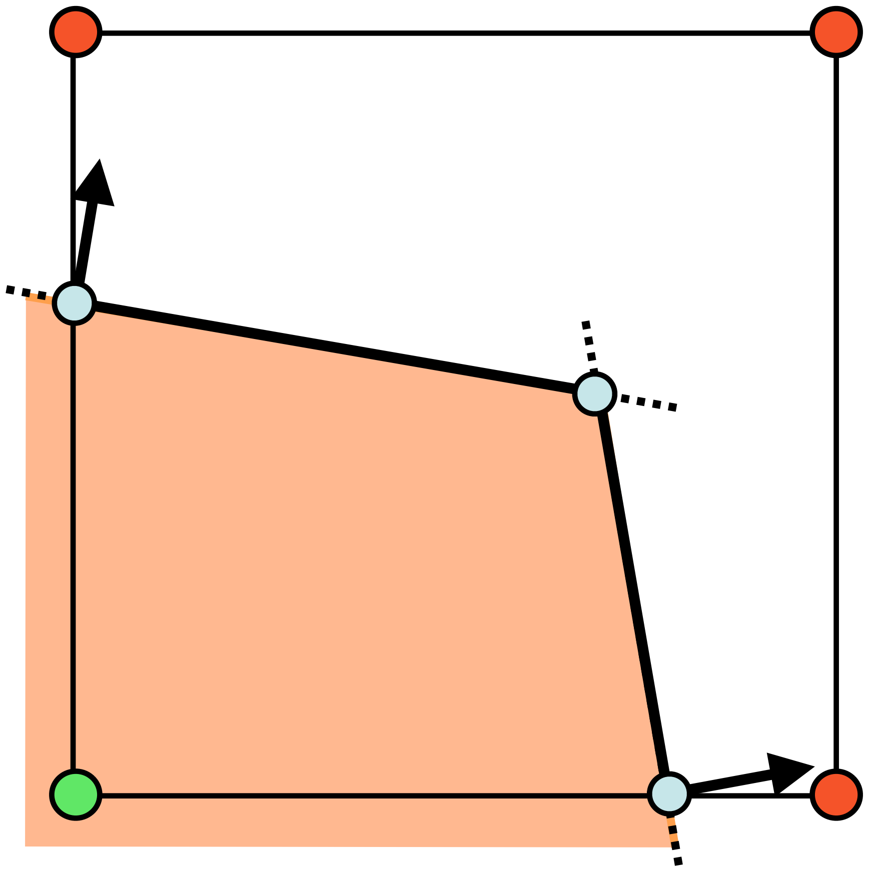
- Sample points restricted to edges of regular grid
- Alias artifacts at sharp features

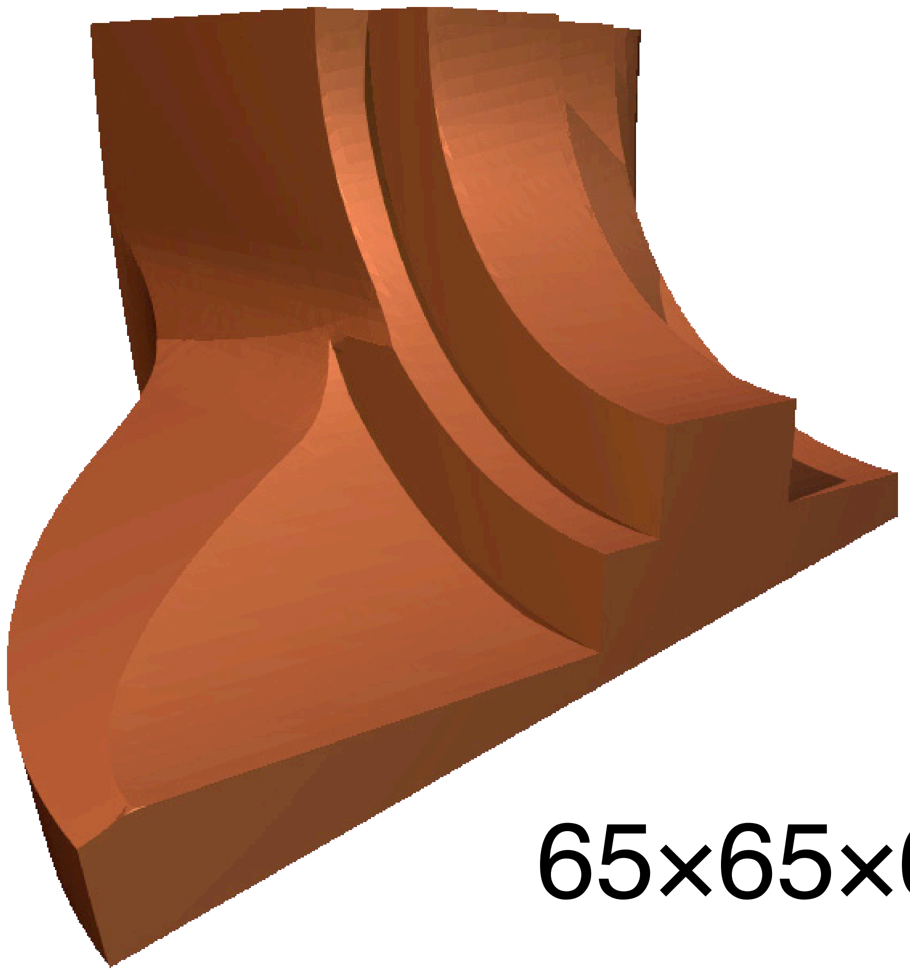




Extended Marching Cubes

- Locally extrapolate distance gradient
- Place samples on estimated feature

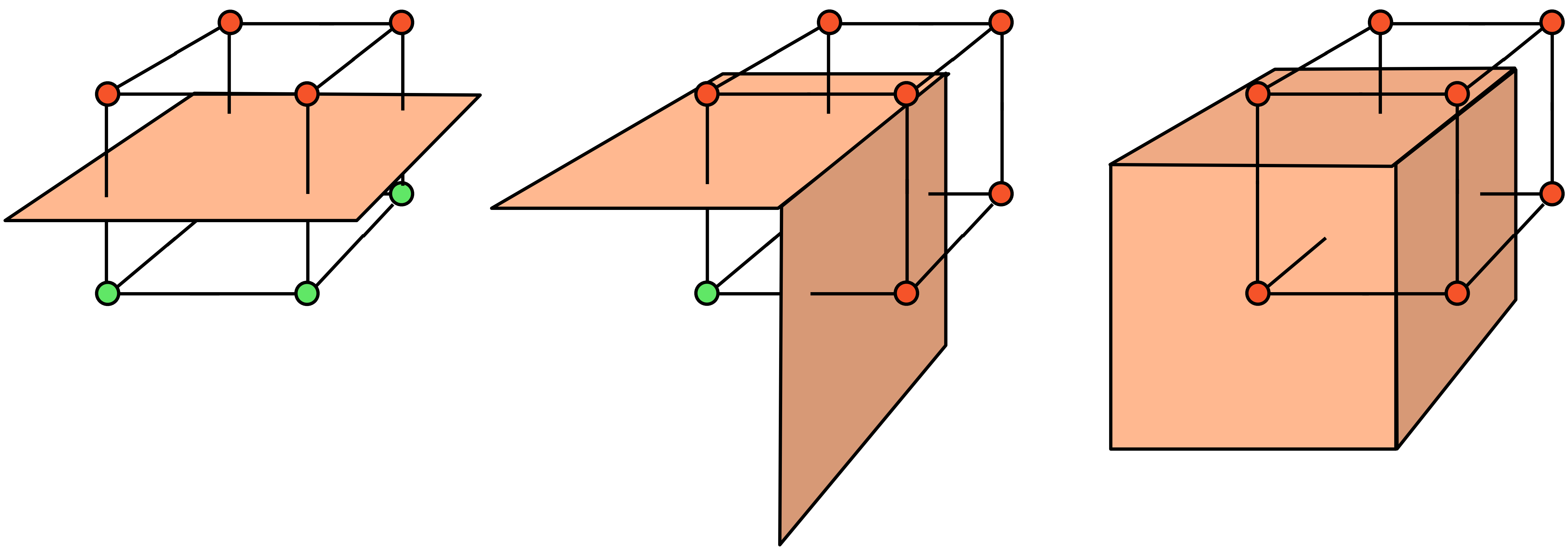




65×65×65

Extended Marching Cubes

- Feature detection
  - Based on angle between normals $\mathbf{n}_i$
  - Classify into edges / corners



Extended Marching Cubes

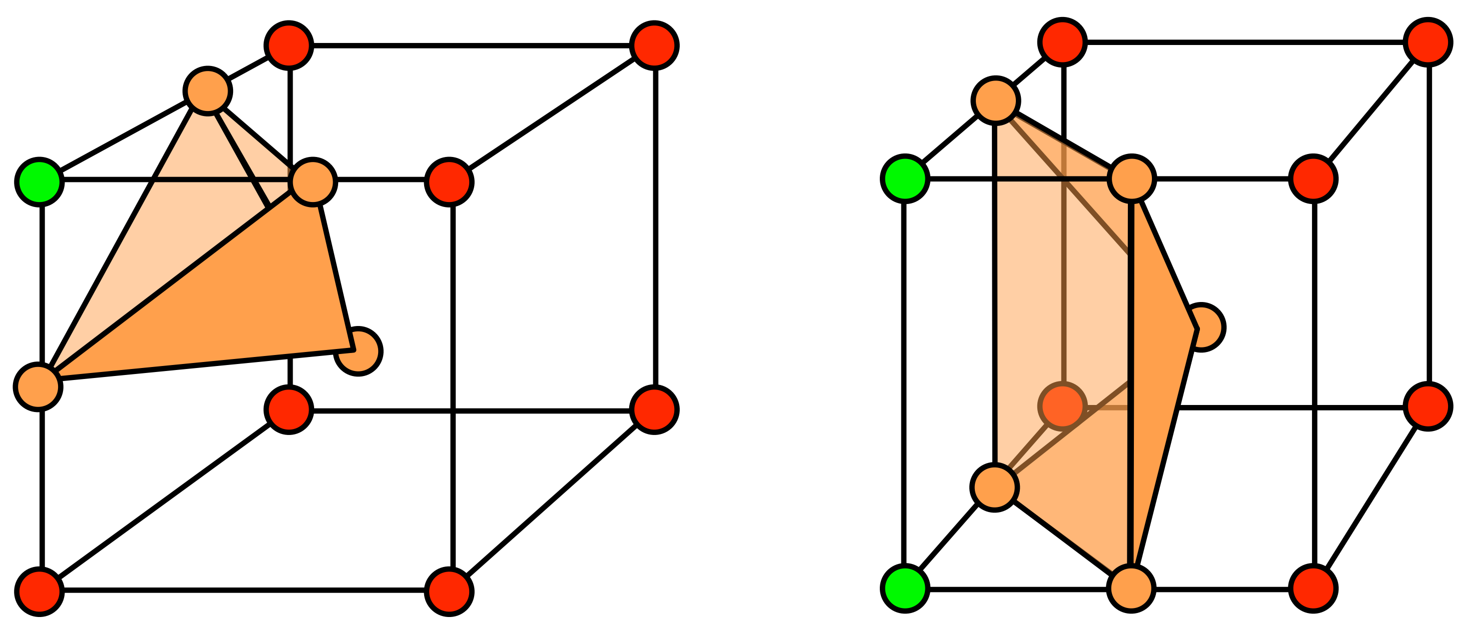
- Feature sampling
  - Intersect tangent planes $(\mathbf{s}_i, \mathbf{n}_i)$

$$\begin{pmatrix} \vdots \\ \mathbf{n}_i \\ \vdots \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \vdots \\ \mathbf{n}_i^T \mathbf{s}_i \\ \vdots \end{pmatrix}$$

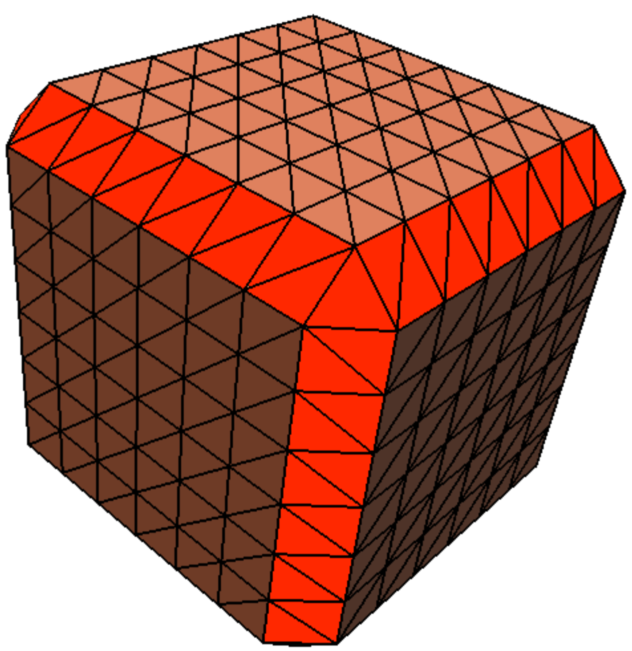
  - Over- or under-determined system
  - Solve by SVD pseudo-inverse

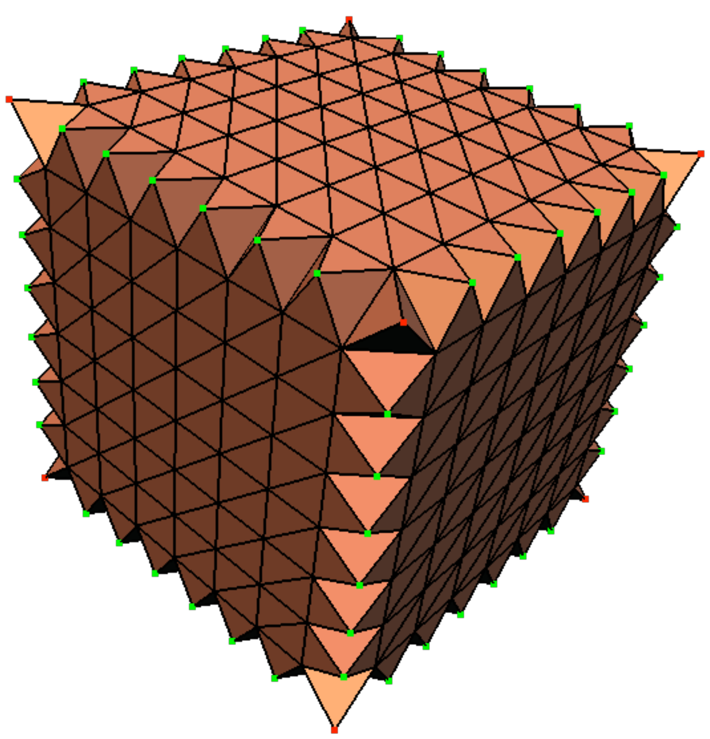
Extended Marching Cubes

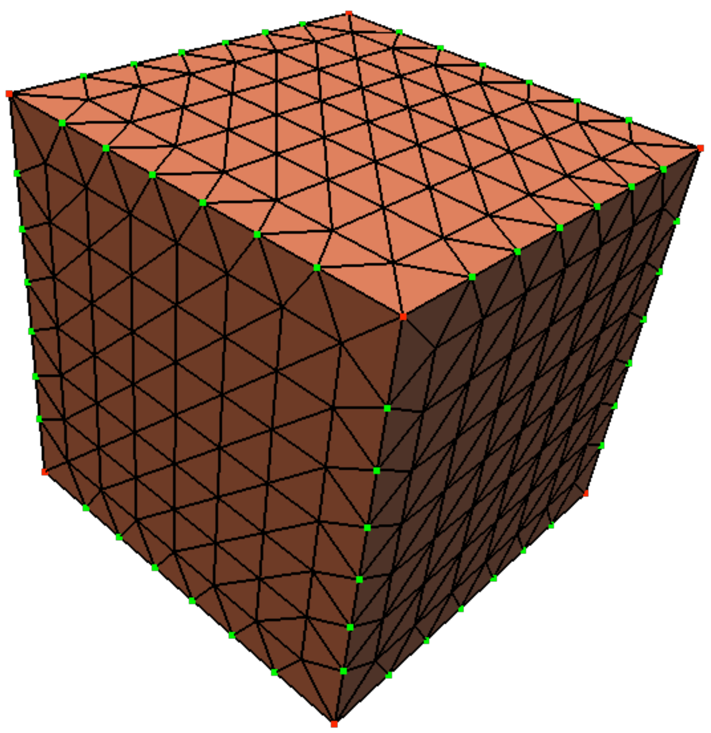
- Feature sampling
  - Intersect tangent planes $(\mathbf{s}_i, \mathbf{n}_i)$
  - Triangle fans centered at feature point



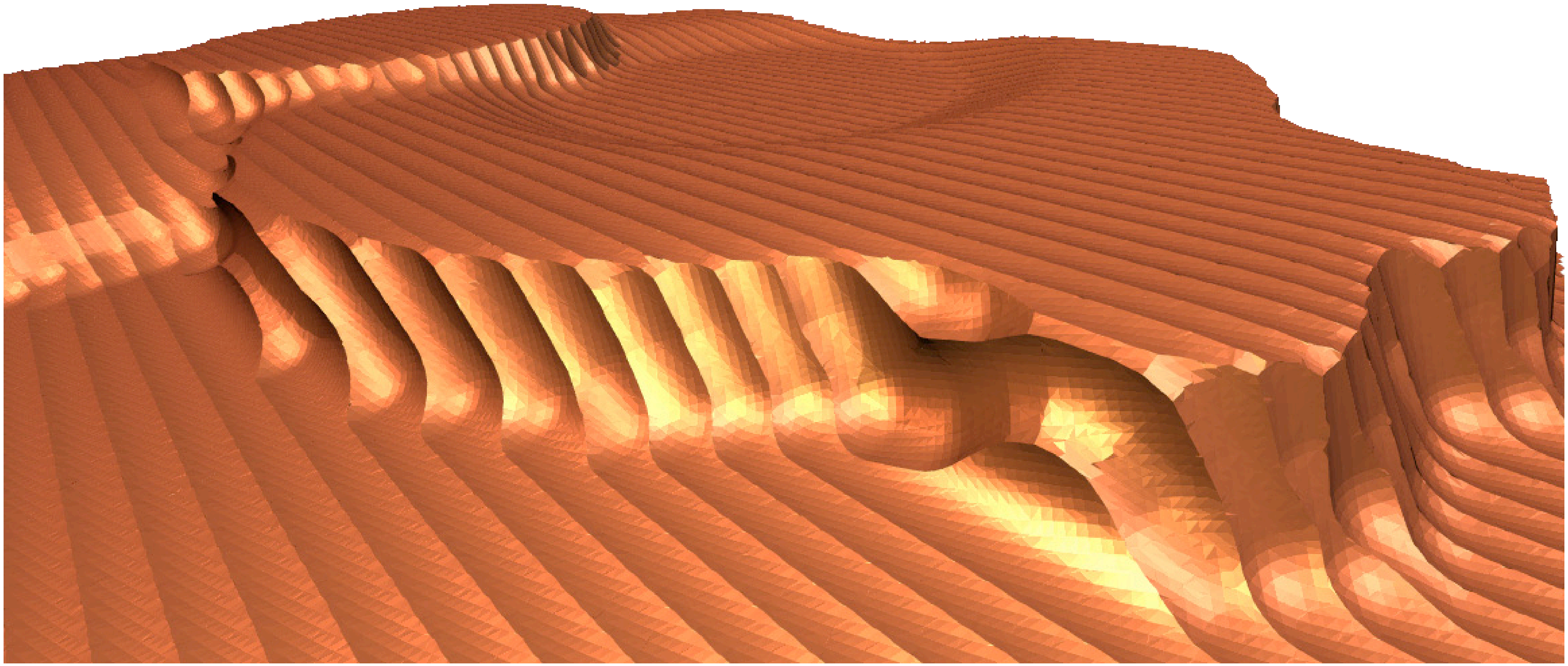
Extended Marching Cubes





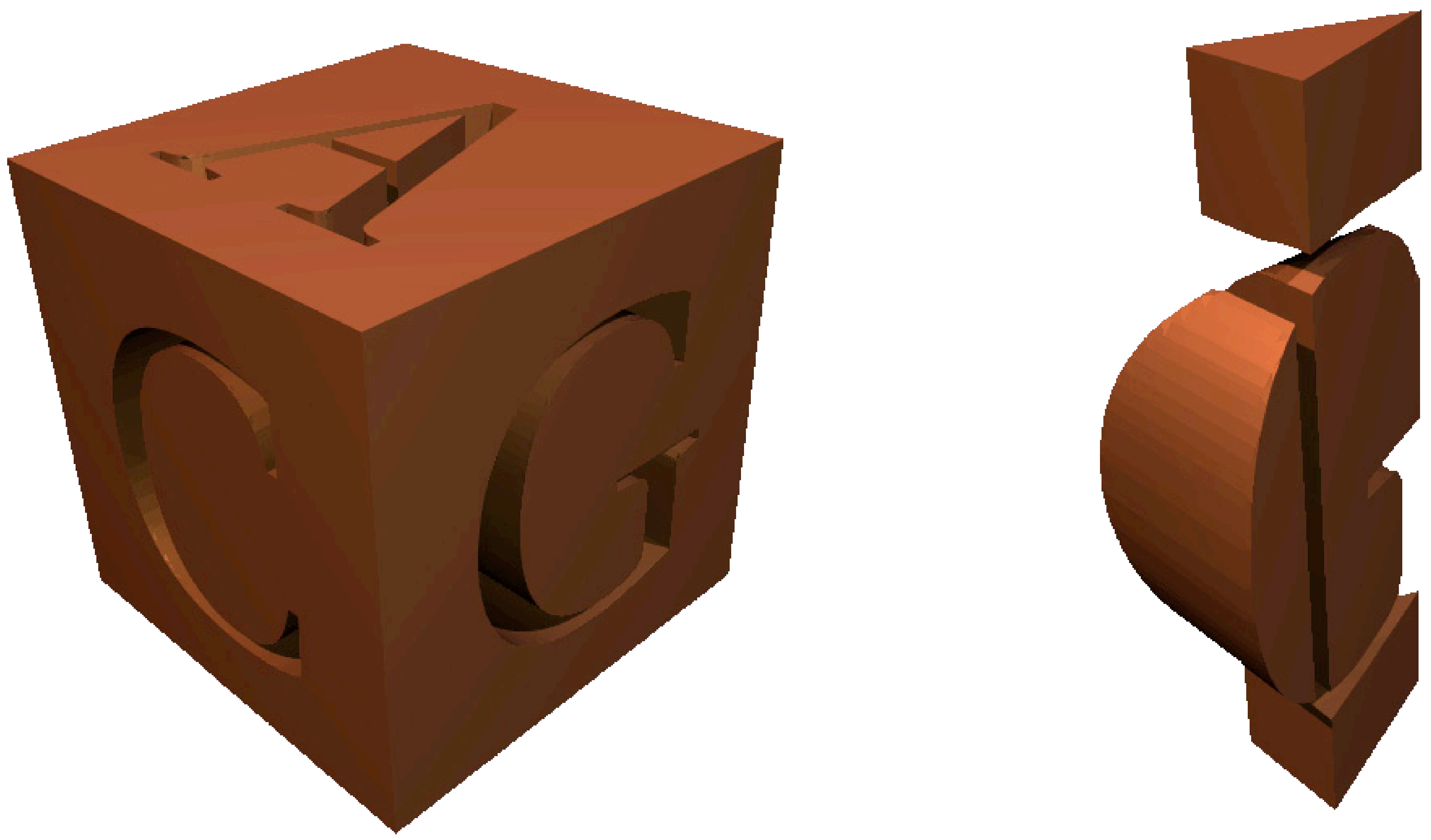


Milling Simulation



257×257×257

CSG Modeling



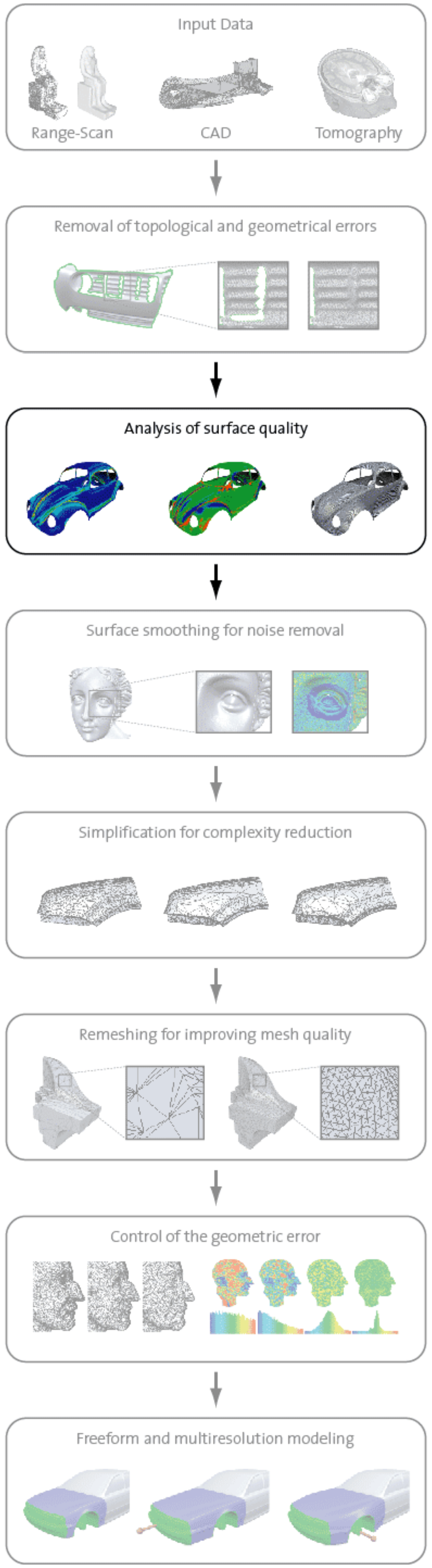
65×65×65

Literature

- Lorensen & Cline, *“Marching Cubes: a High Resolution 3D Surface Construction Algorithm”*, SIGGRAPH 1987
- Montani et al, *“A modified look-up table for implicit disambiguation of Marching Cubes”*, Visual Computer 1994
- Kobbelt et al, *“Feature Sensitive Surface Extraction from Volume Data”*, SIGGRAPH 2001

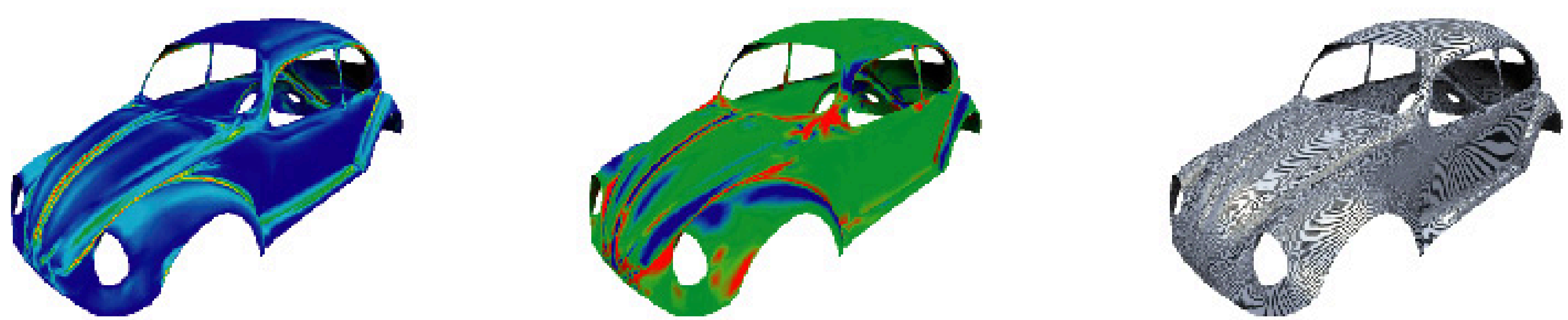
Outline

- Surface Representations
  - Explicit vs. Implicit
- Explicit Representation
  - Triangle Meshes
- Implicit Representations
  - Signed Distance Functions
- Conversions
  - Implicit ↔ Explicit



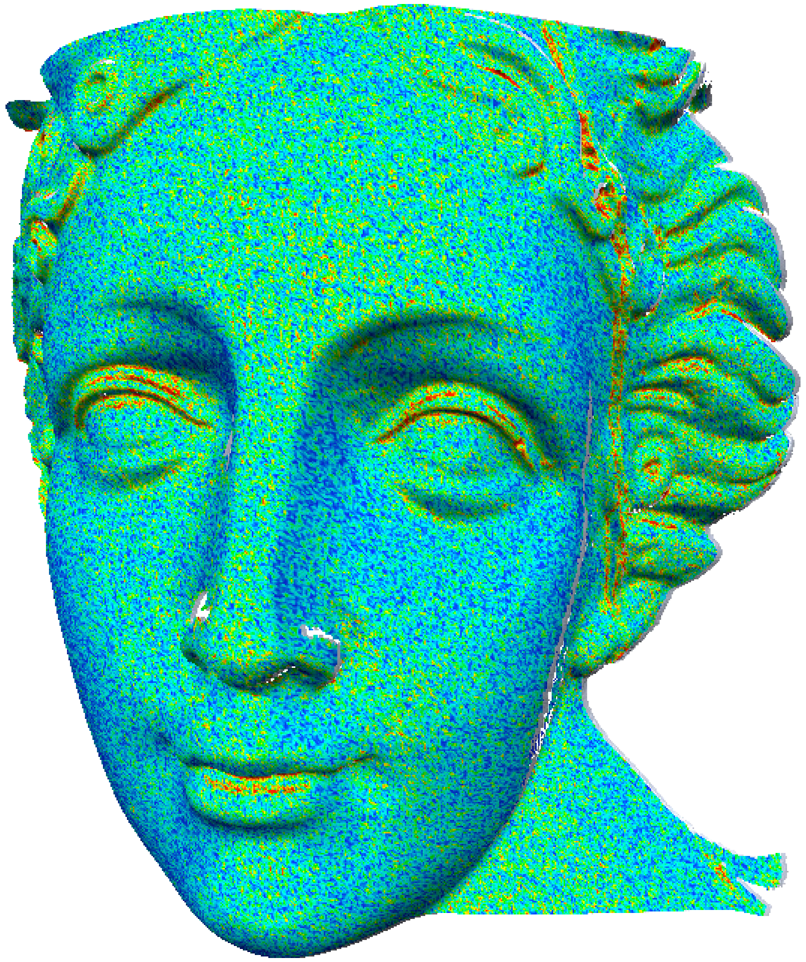
Analysis of Surface Quality

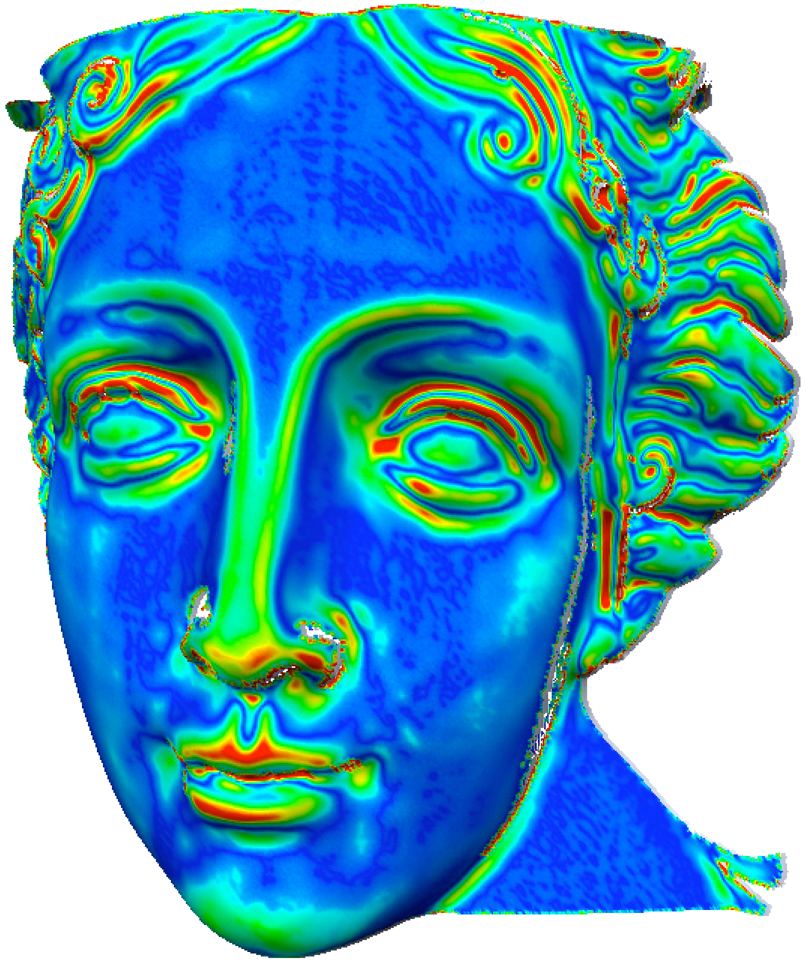
Analysis of surface quality



Mesh Quality Criteria

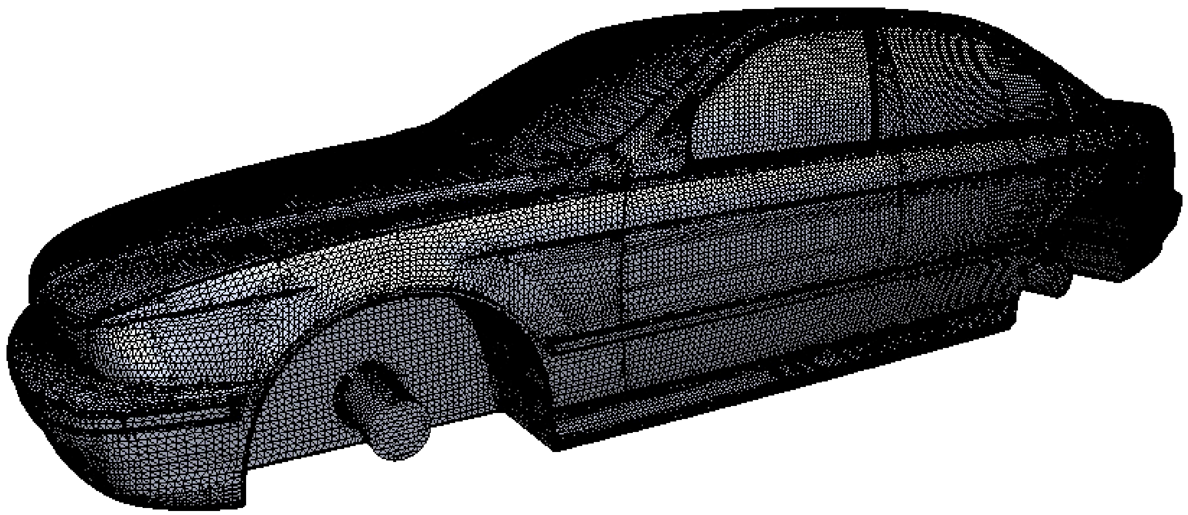
- Smoothness
  - Low geometric noise

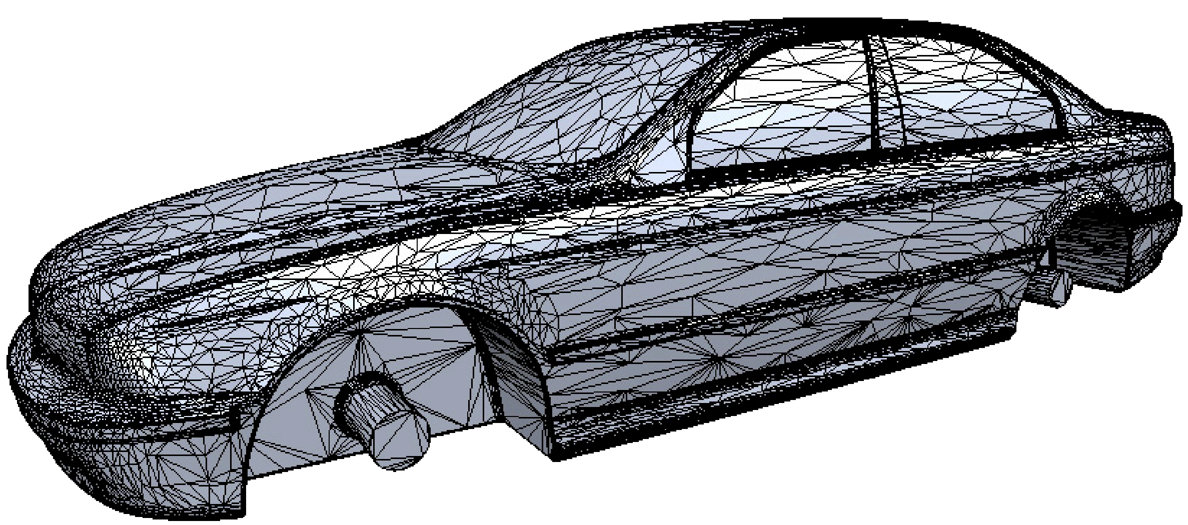




Mesh Quality Criteria

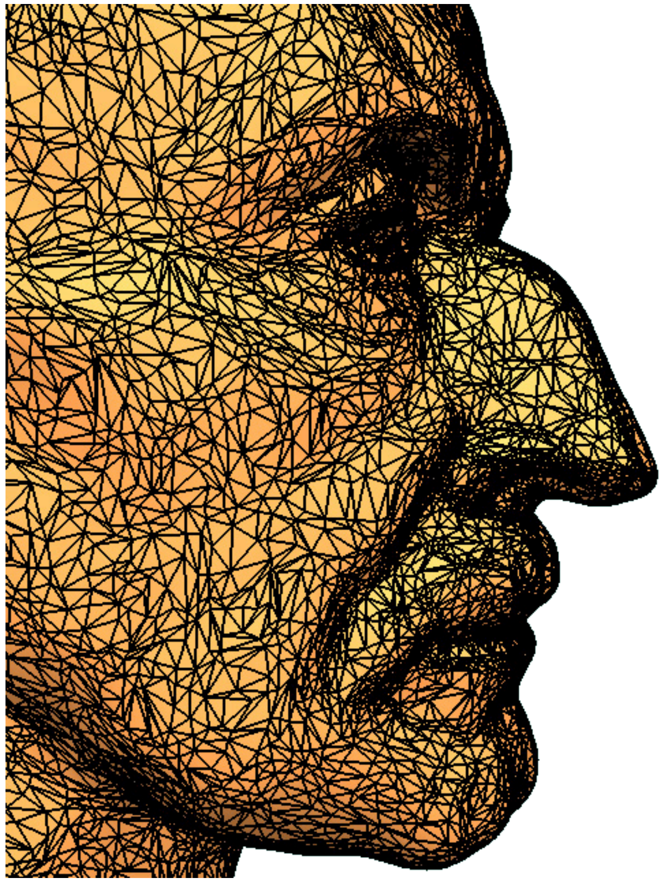
- Smoothness
  - Low geometric noise
- Adaptive tessellation
  - Low complexity

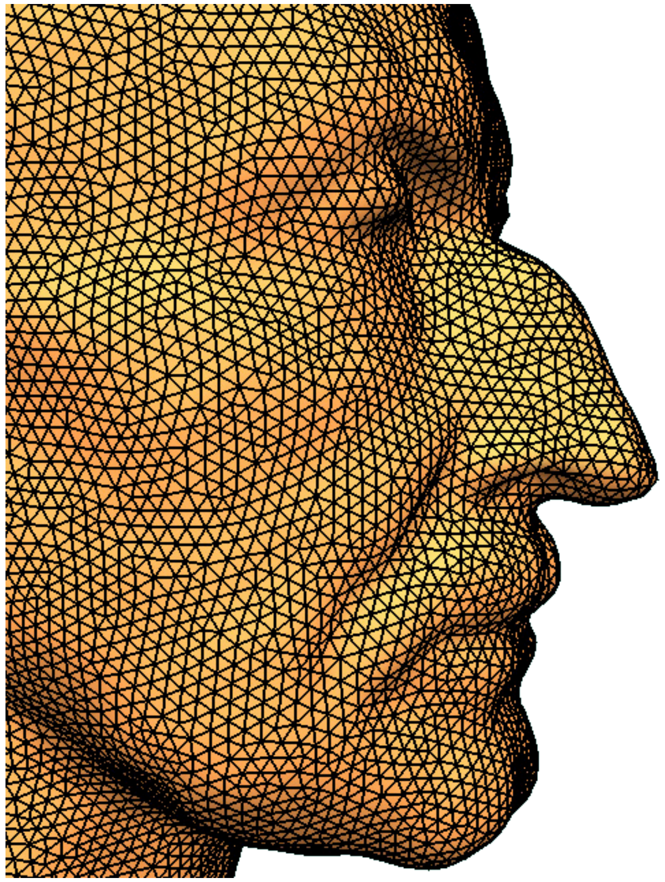




Mesh Quality Criteria

- Smoothness
  - Low geometric noise
- Adaptive tessellation
  - Low complexity
- Triangle shape
  - Numerical robustness



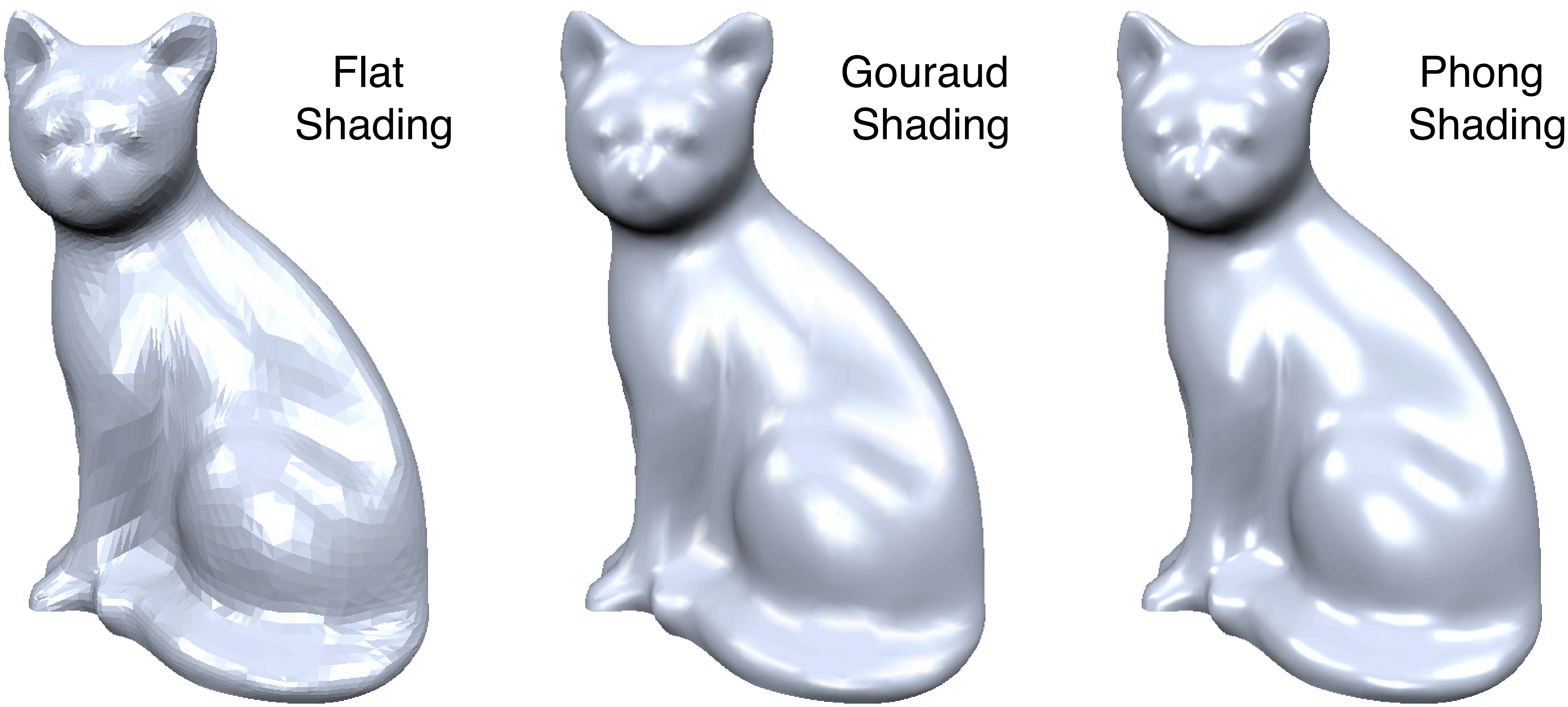


Mesh Quality Criteria

- Smoothness
  - Low geometric noise
- Adaptive tessellation
  - Low complexity
- Triangle shape
  - Numerical robustness
- Feature preservation
  - Low normal noise

Smoothness Analysis

- Visual inspection of "sensitive" attributes
  - Specular shading

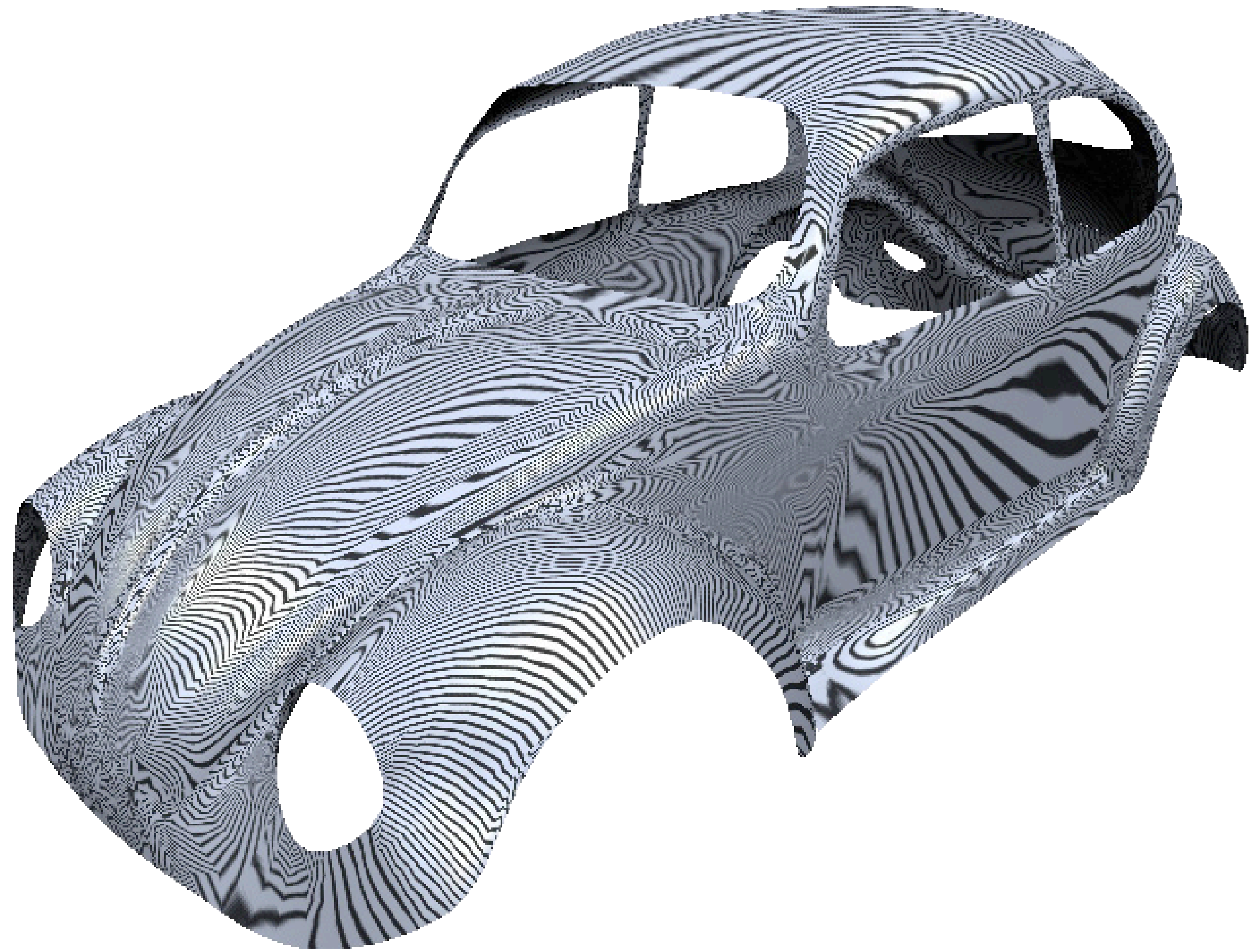


Smoothness Analysis

- Visual inspection of “sensitive” attributes
  - Specular shading

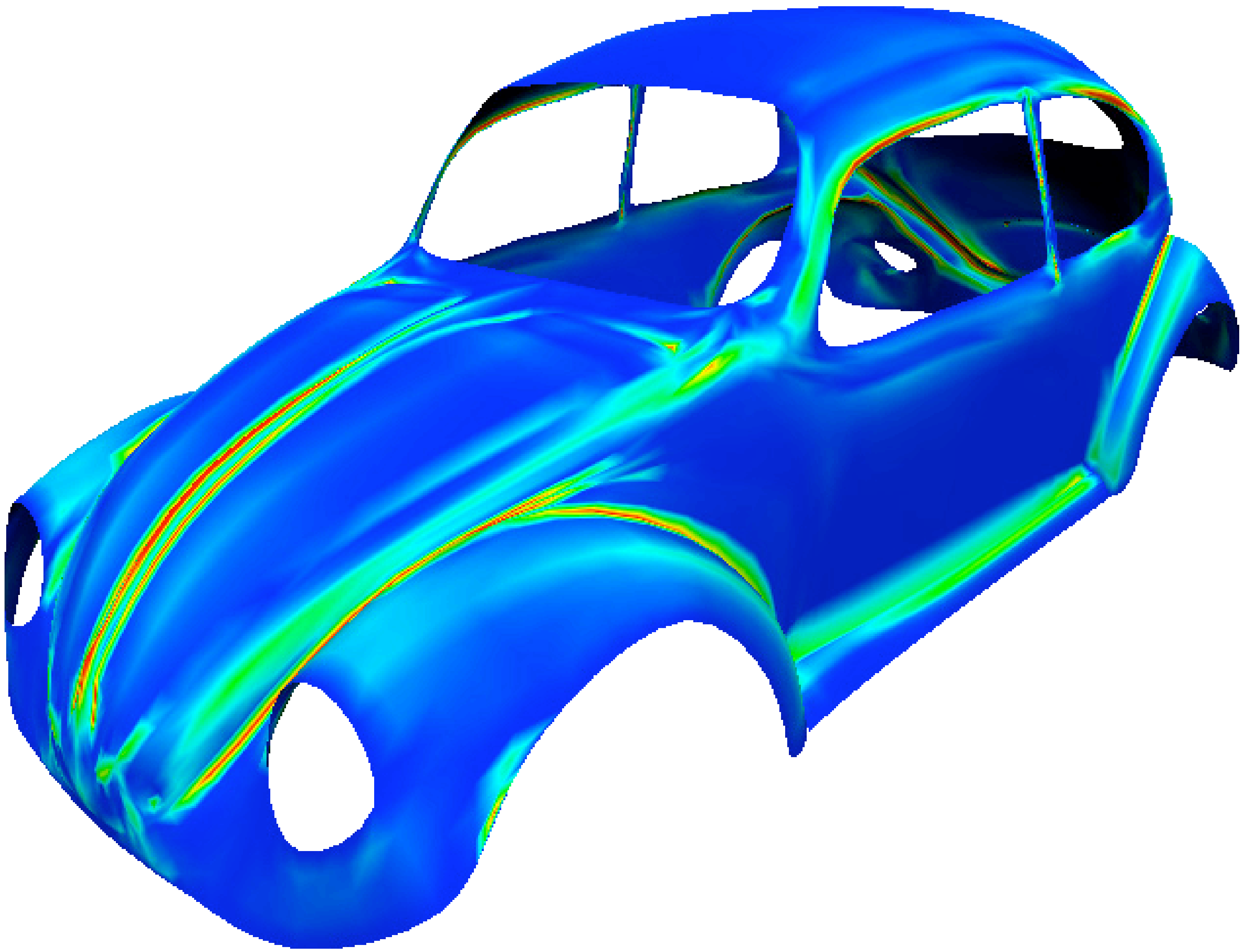
Smoothness Analysis

- Visual inspection of “sensitive” attributes
  - Specular shading
  - Reflection lines



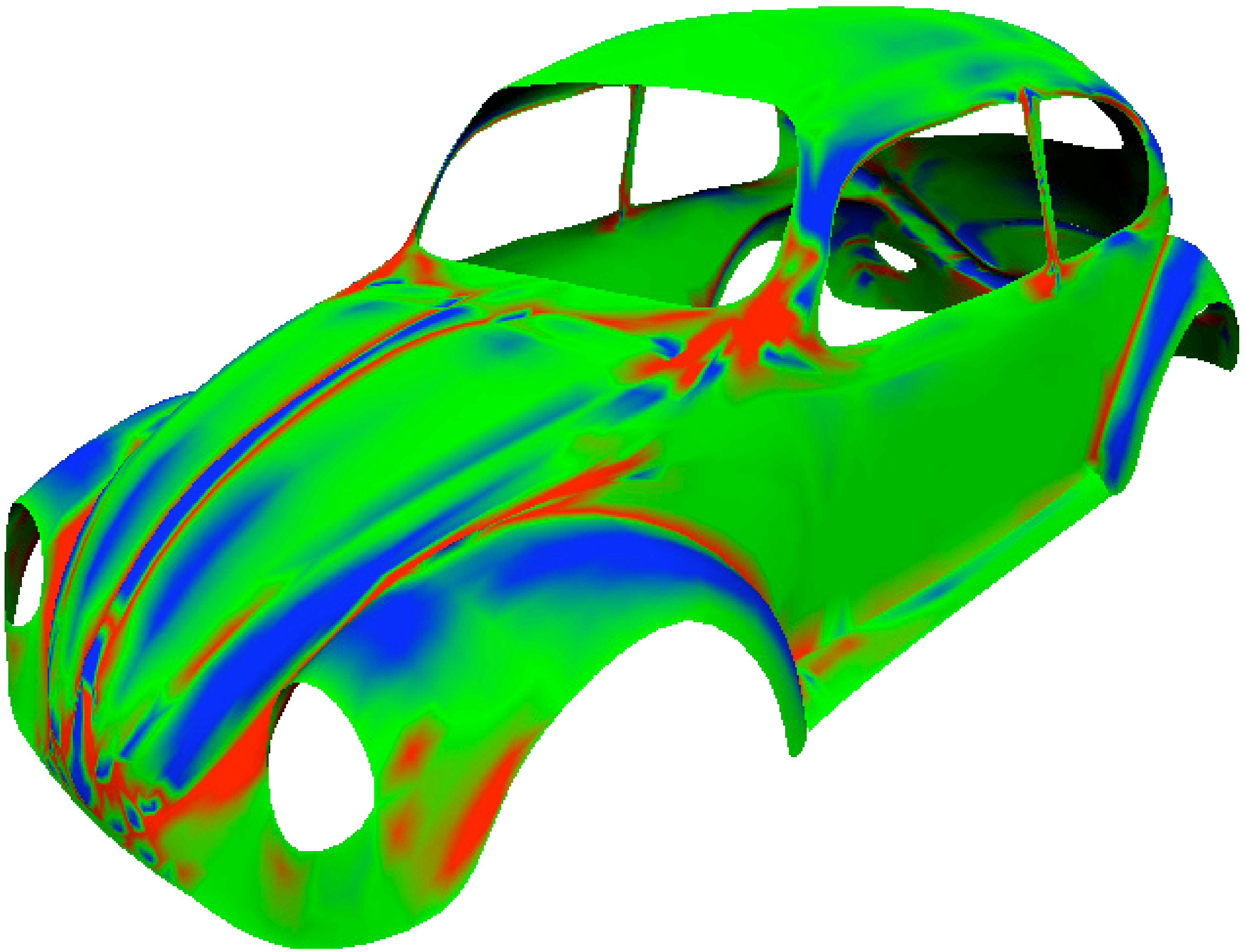
Smoothness Analysis

- Visual inspection of “sensitive” attributes
  - Specular shading
  - Reflection lines
  - Curvature
    - Mean curvature



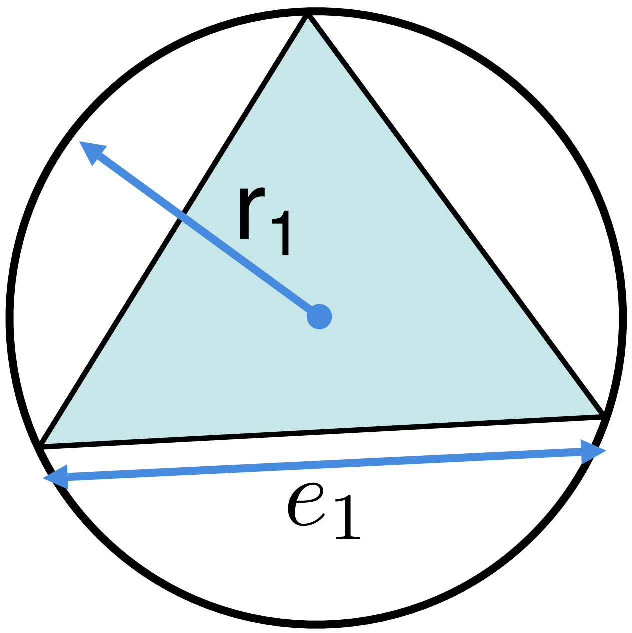
Smoothness Analysis

- Visual inspection of “sensitive” attributes
  - Specular shading
  - Reflection lines
  - Curvature
    - Mean curvature
    - Gauss curvature

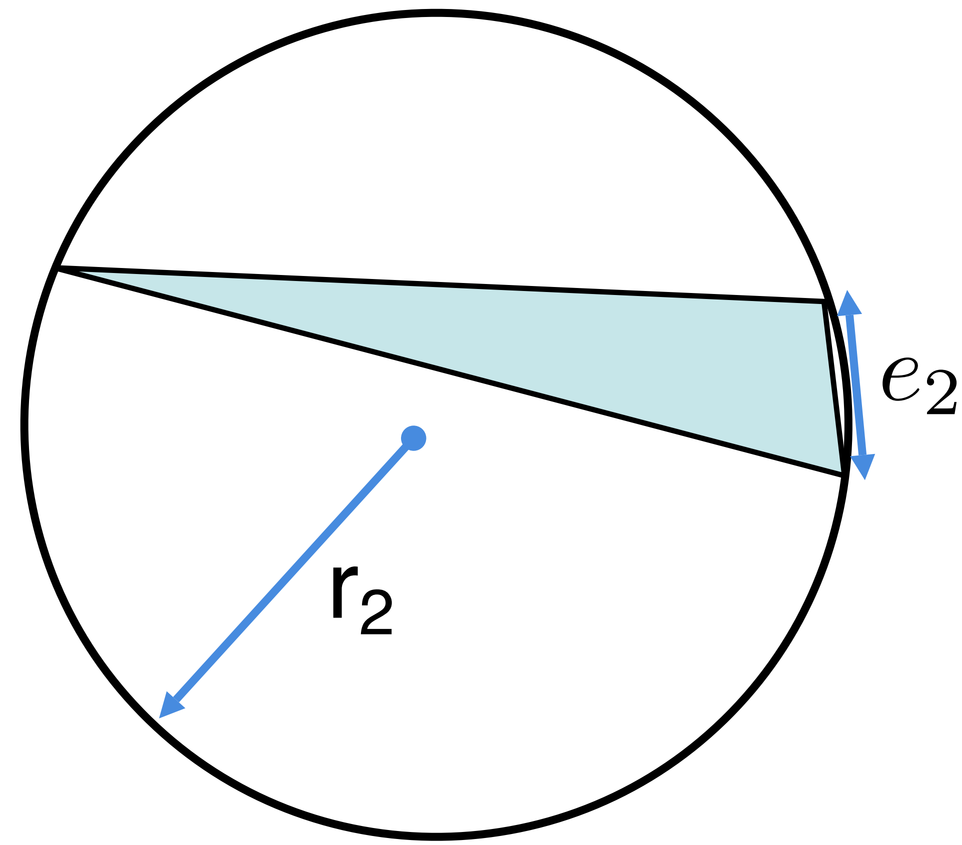


Triangle Shape Analysis

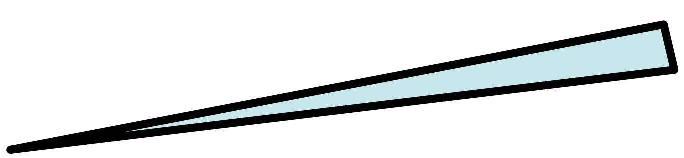
- Circum radius / shortest edge



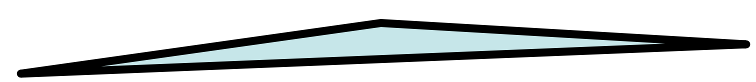
$$\frac{r_1}{e_1} < \frac{r_2}{e_2}$$



- Needles and caps

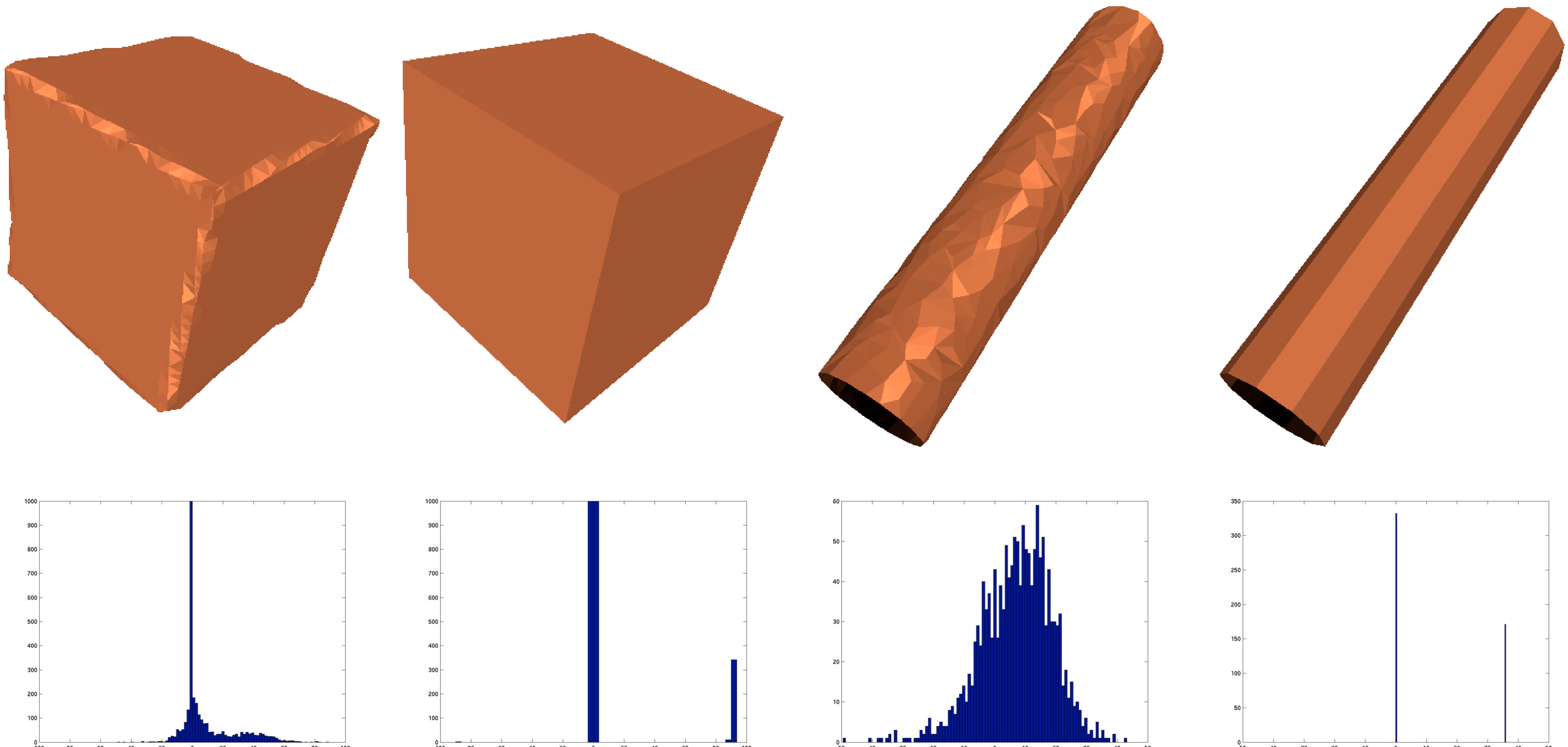


Needle



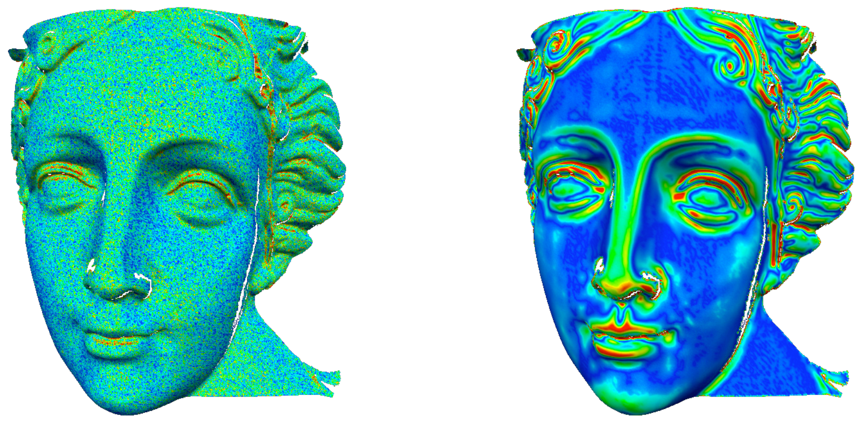
Cap

Normal Noise Analysis

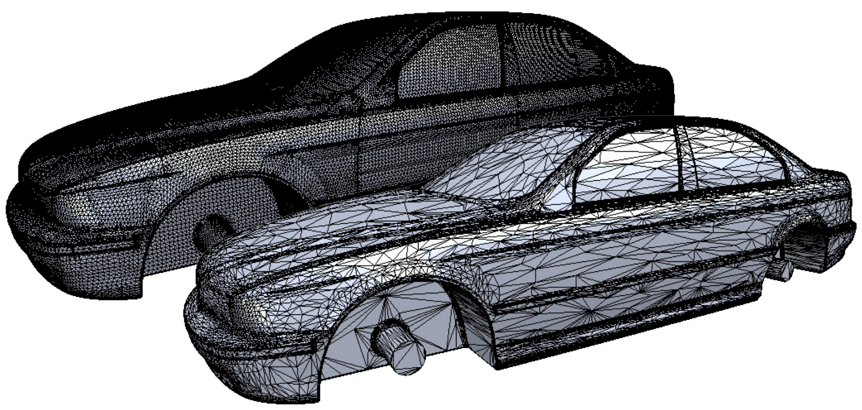


Mesh Optimization

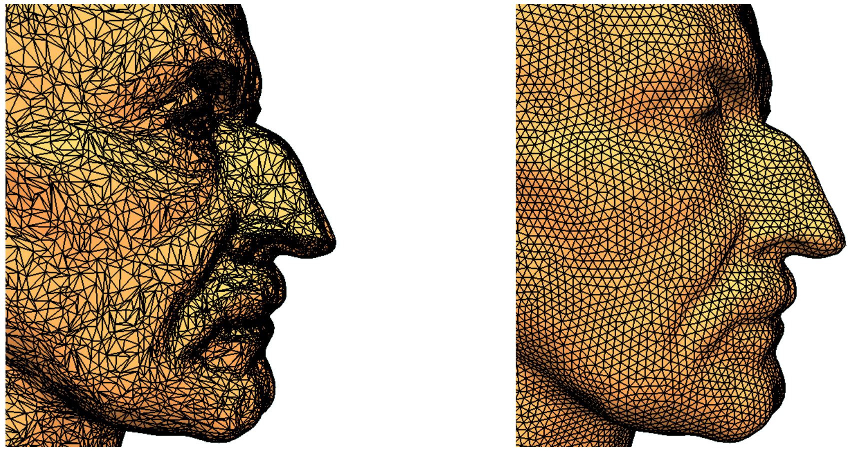
- Smoothness
  - ➡ Mesh smoothing

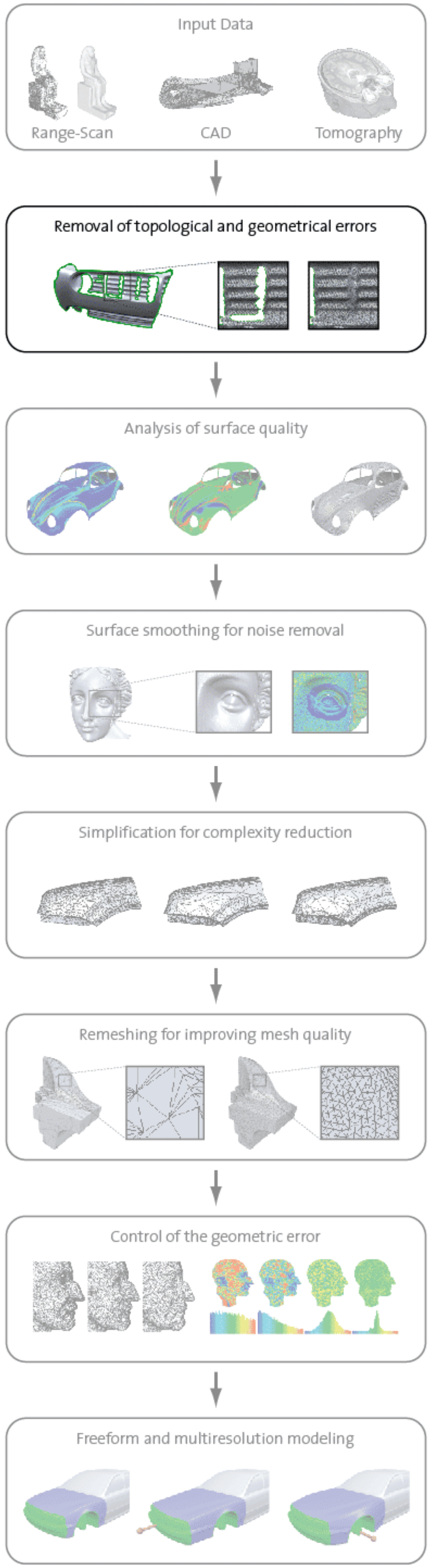


- Adaptive tessellation
  - ➡ Mesh decimation

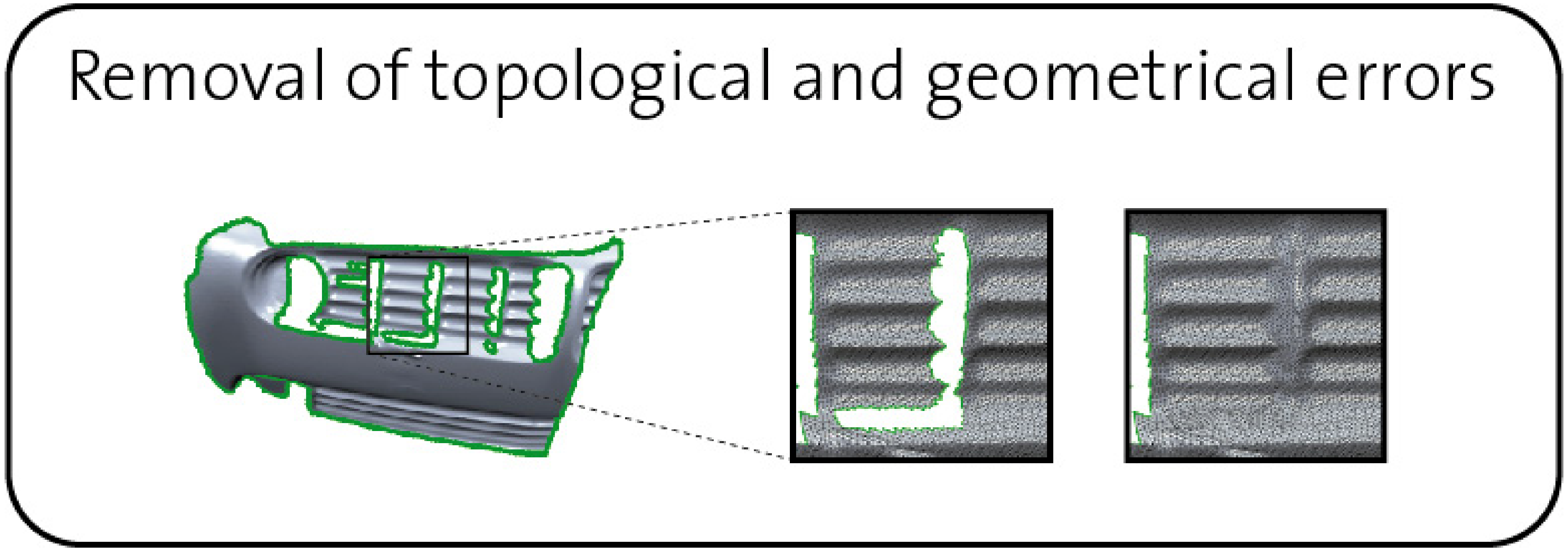


- Triangle shape
  - ➡ Repair, remeshing



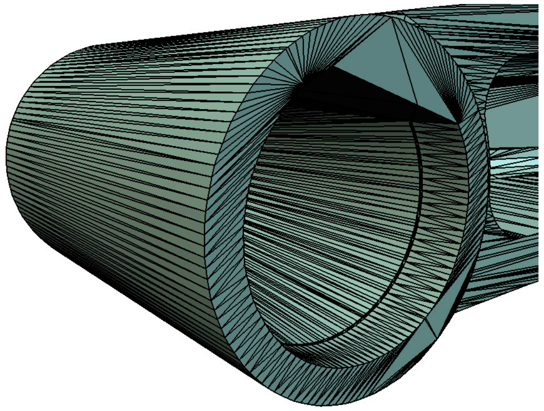

Input Data
Range-Scan
CAD
Tomography
Removal of topological and geometrical errors
Analysis of surface quality
Surface smoothing for noise removal
Simplification for complexity reduction
Remeshing for improving mesh quality
Control of the geometric error
Freeform and multiresolution modeling

Mesh Repair

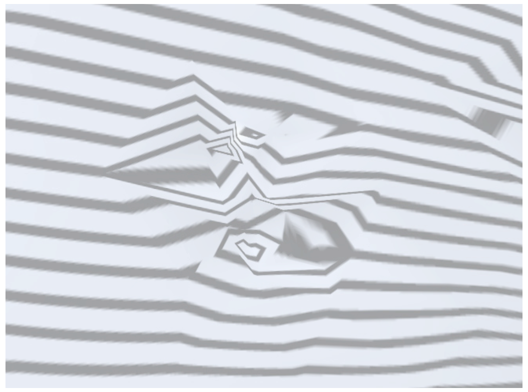

Removal of topological and geometrical errors

Mesh Degeneracies

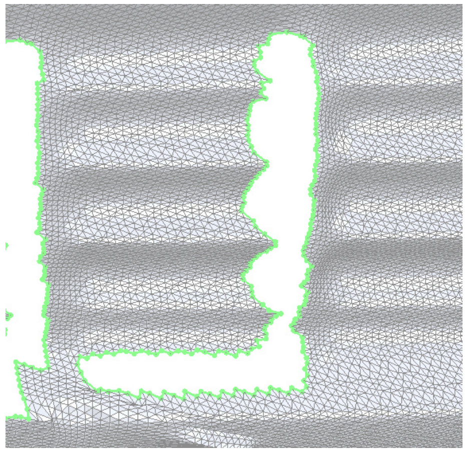
- Degenerate triangles
  - Needles, caps



- Scanning artifacts
  - Noise

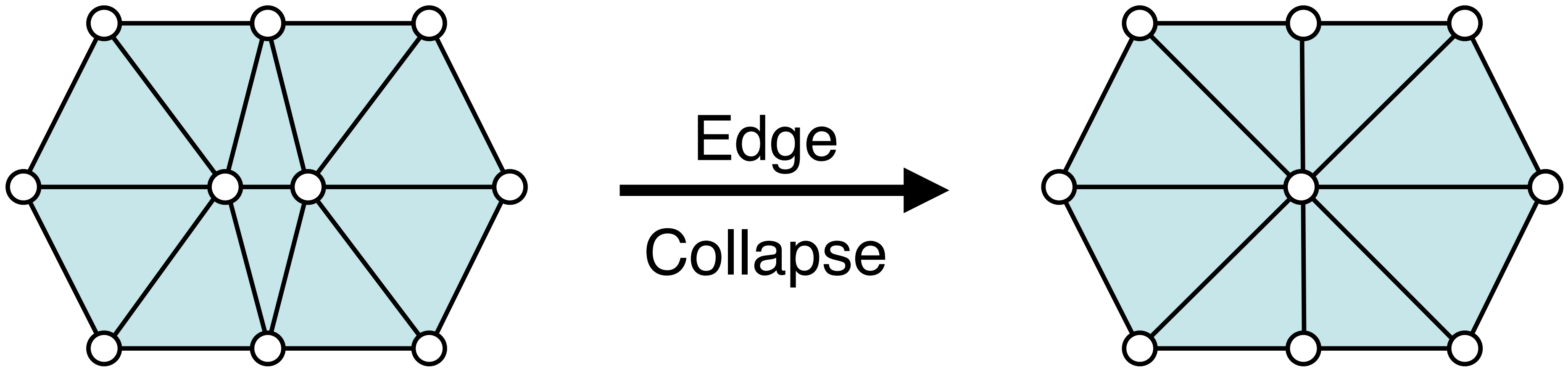


- Holes
  - Occlusion during scanning

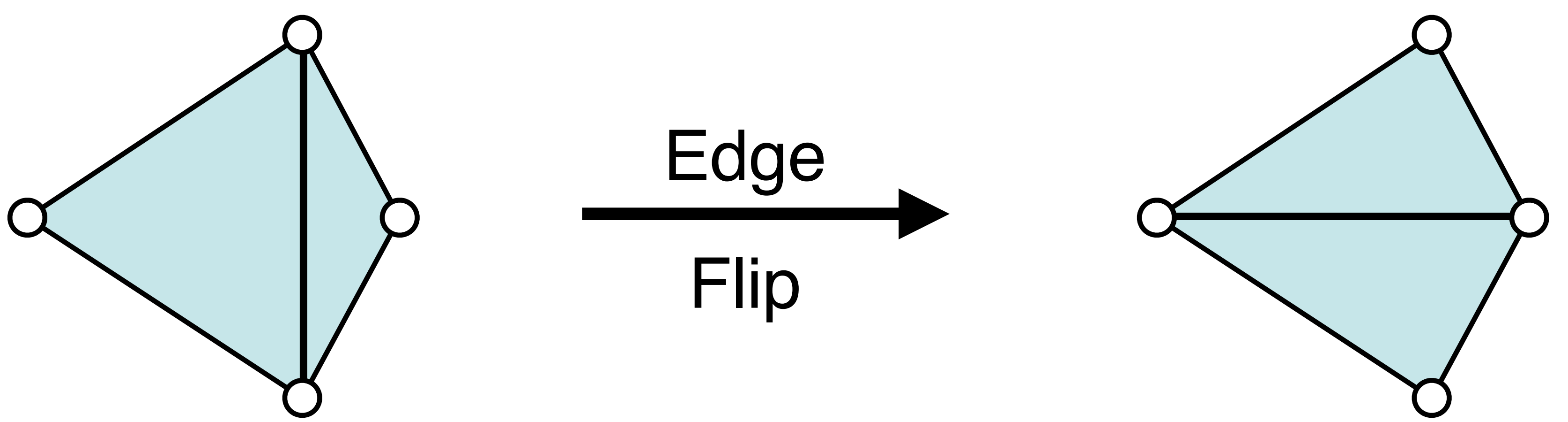


Degenerate Triangles

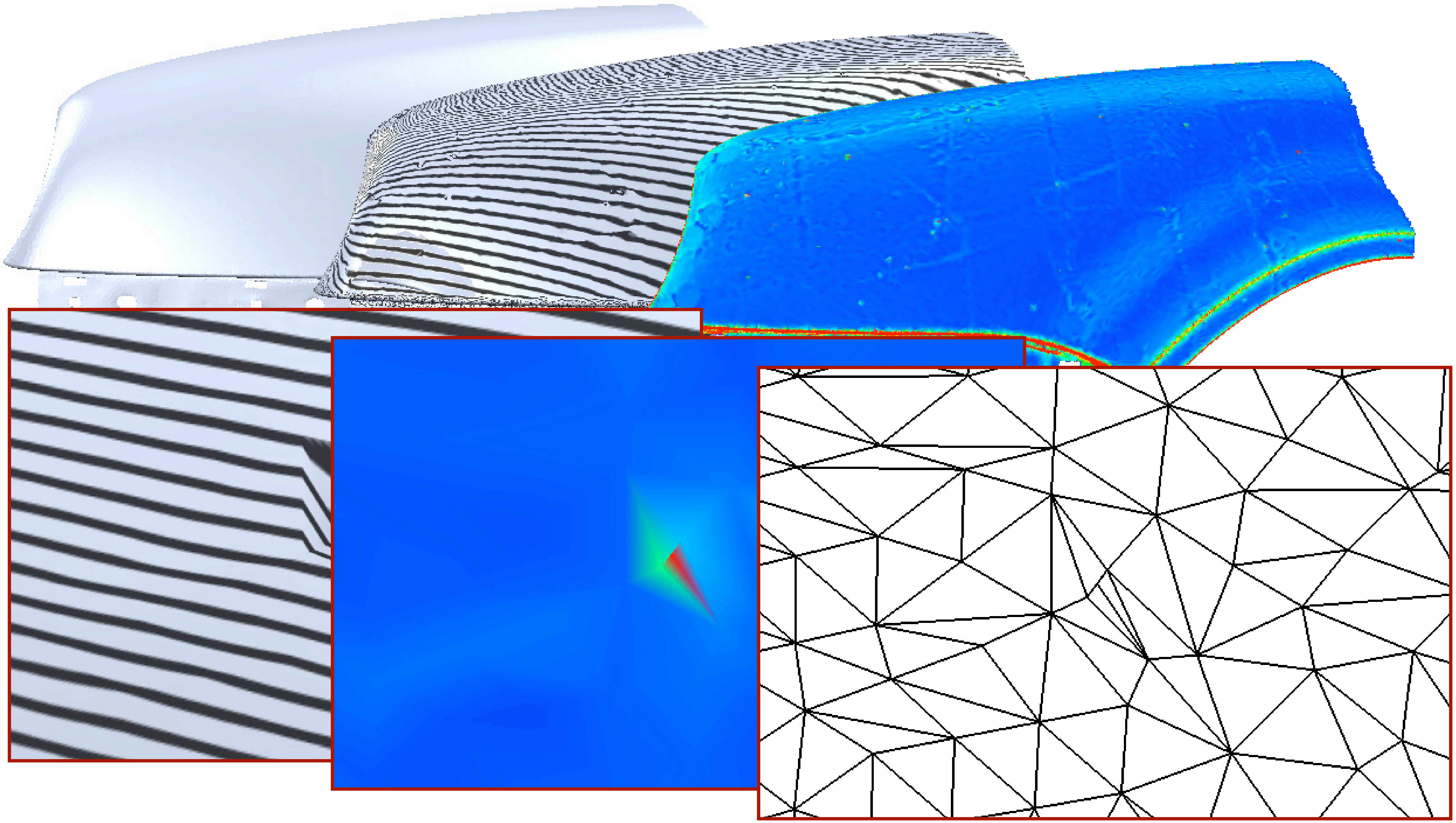
- Remove needles by edge collapses



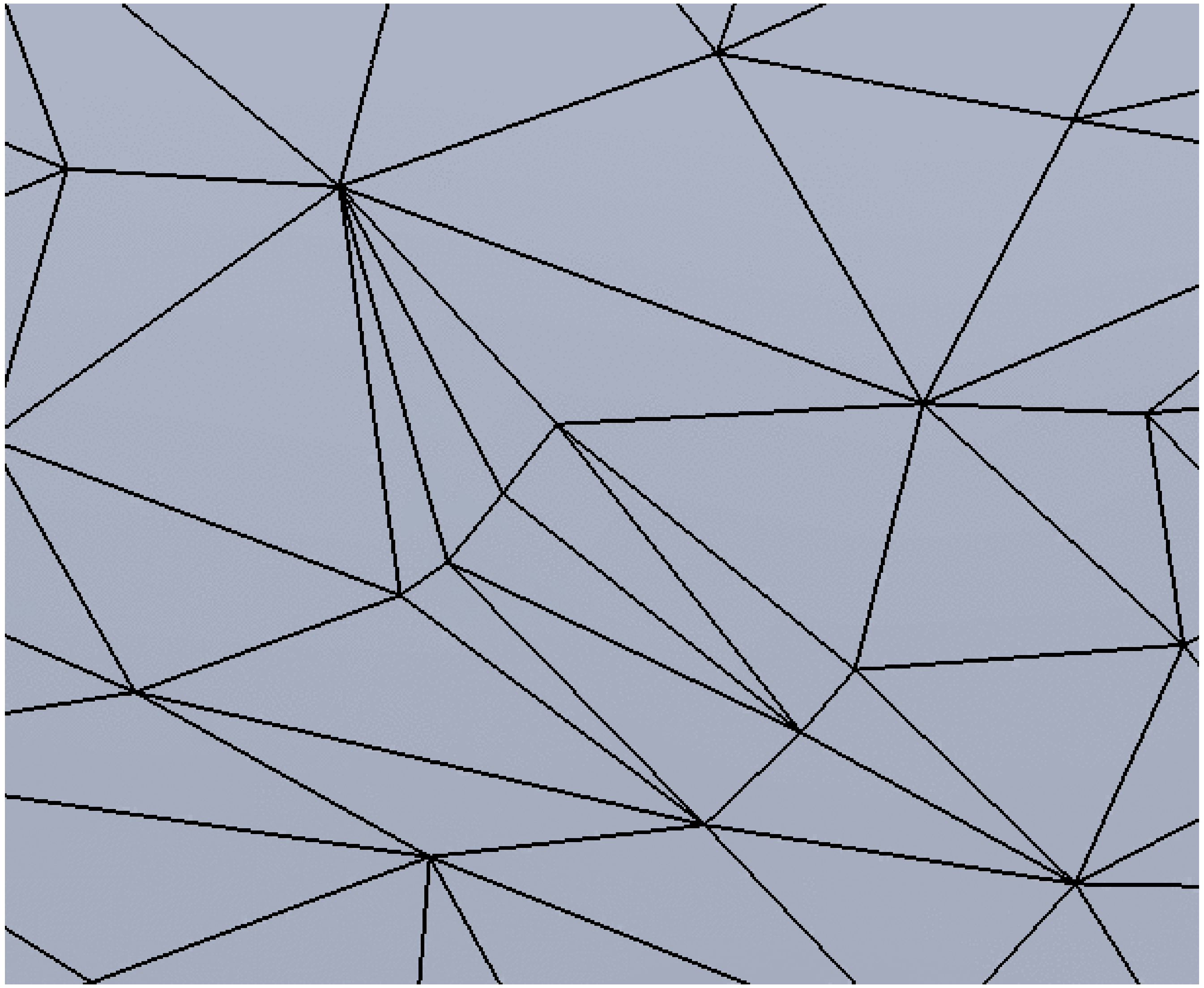
- Remove isolated caps by edge flips



Degenerate Triangles



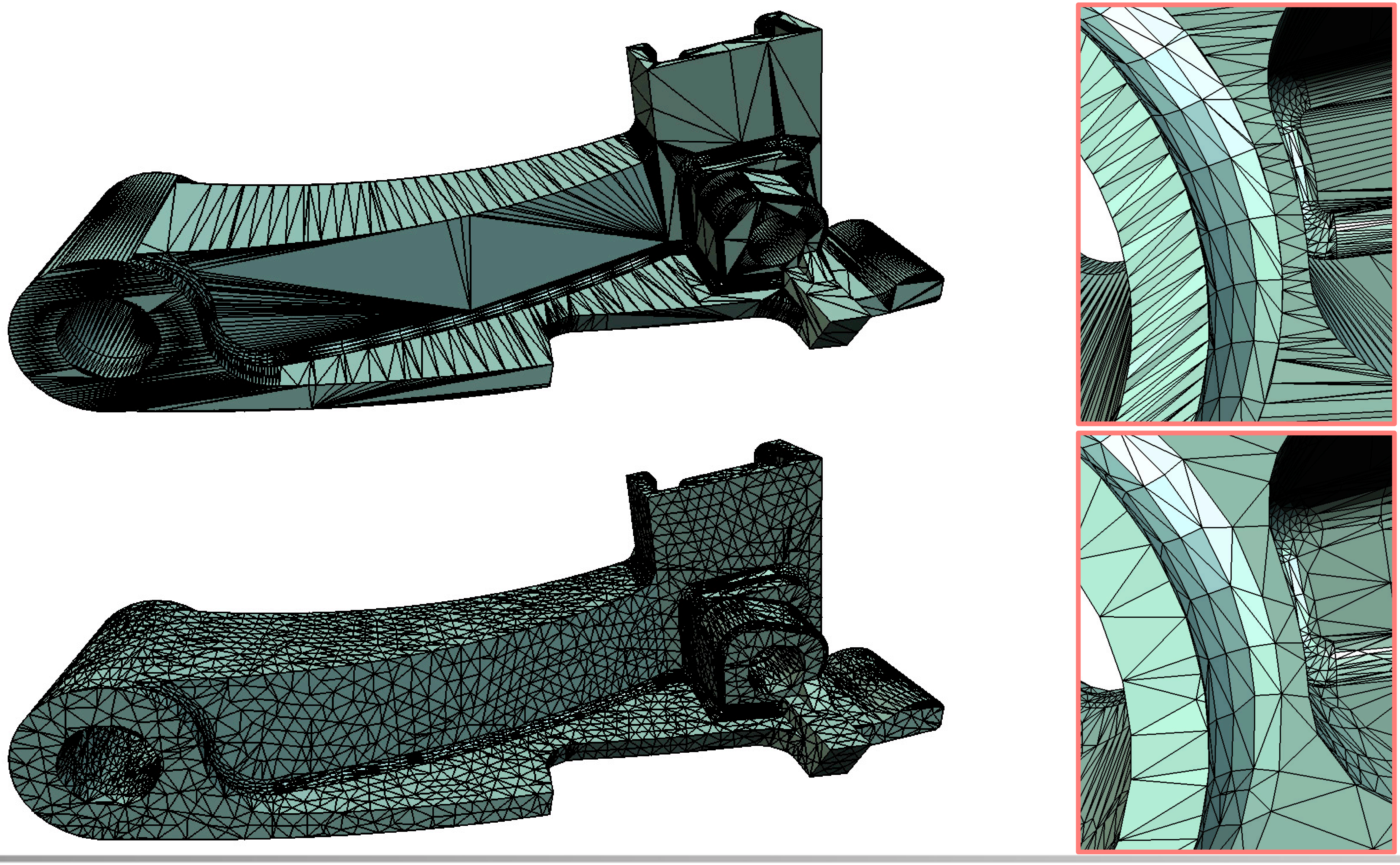
Degenerate Triangles



Degenerate Triangles

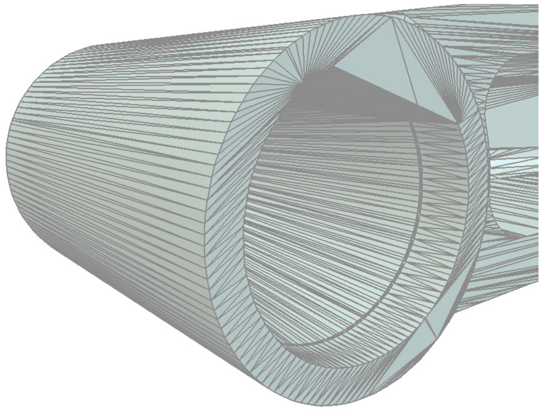
- Remove needles by edge collapses
- Remove isolated caps by edge flips
- Remove groups of caps by mesh slicing
  - Intersect mesh with stacks of parallel planes
  - Turns caps into needles
  - Use mesh decimation to remove needles

Mesh Slicing



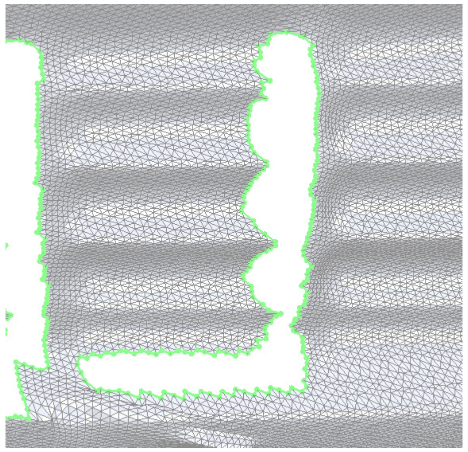
Mesh Degeneracies

- Degenerate triangles
  - Needles, caps



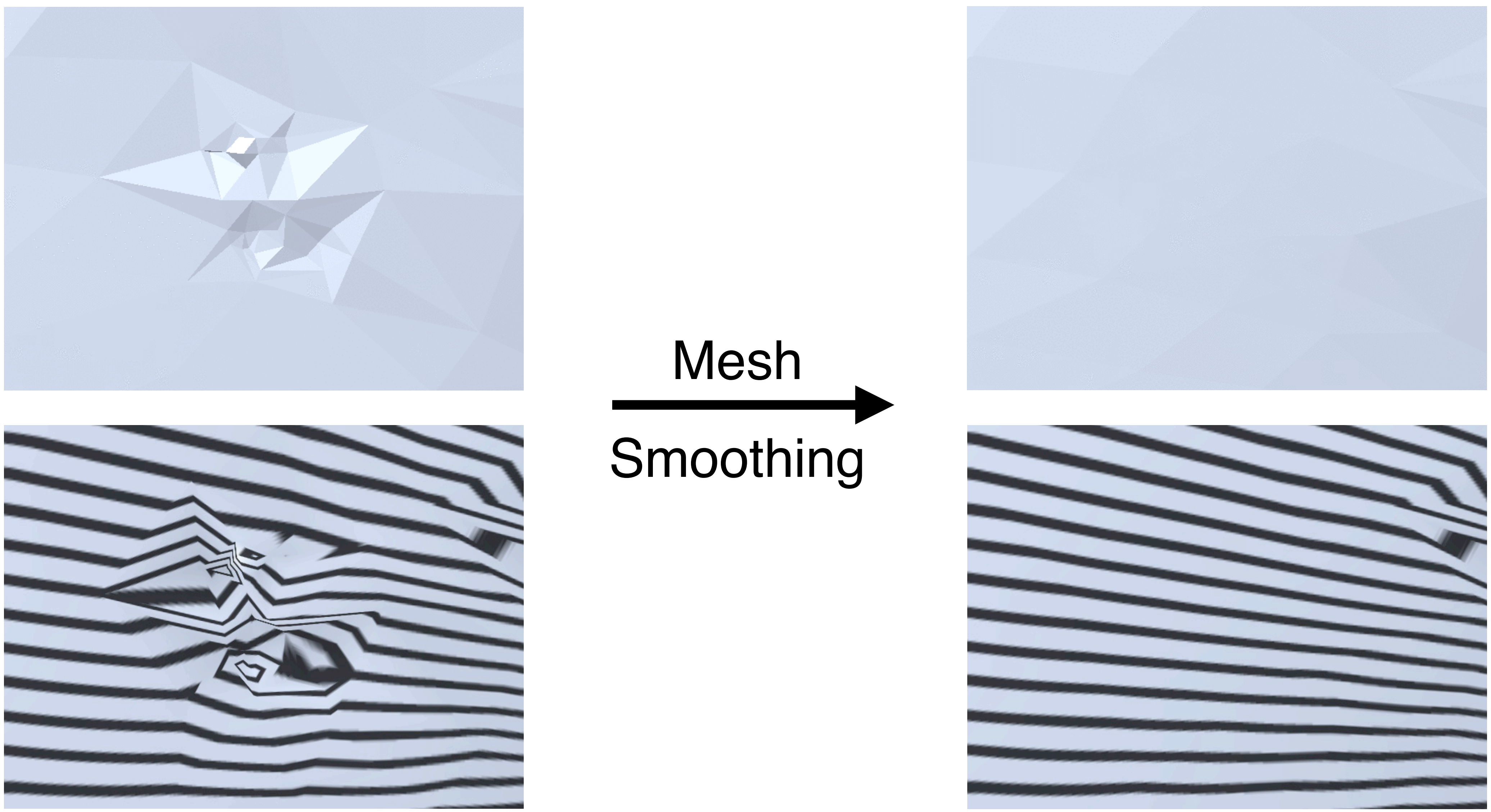
- Scanning artifacts
  - Noise

- Holes
  - Occlusion during scanning



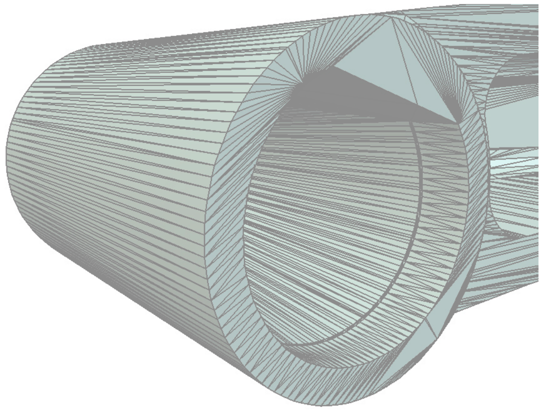
Measurement Noise

- Later: mesh smoothing

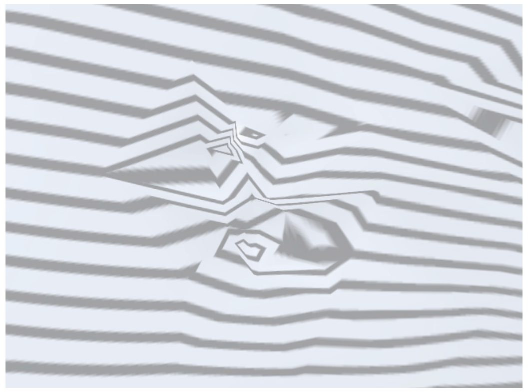


Mesh Degeneracies

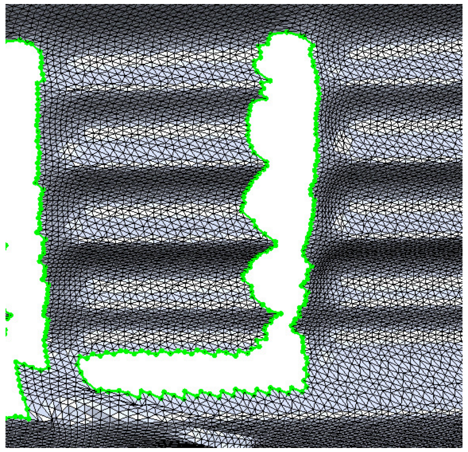
- Degenerate triangles
  - Needles, caps



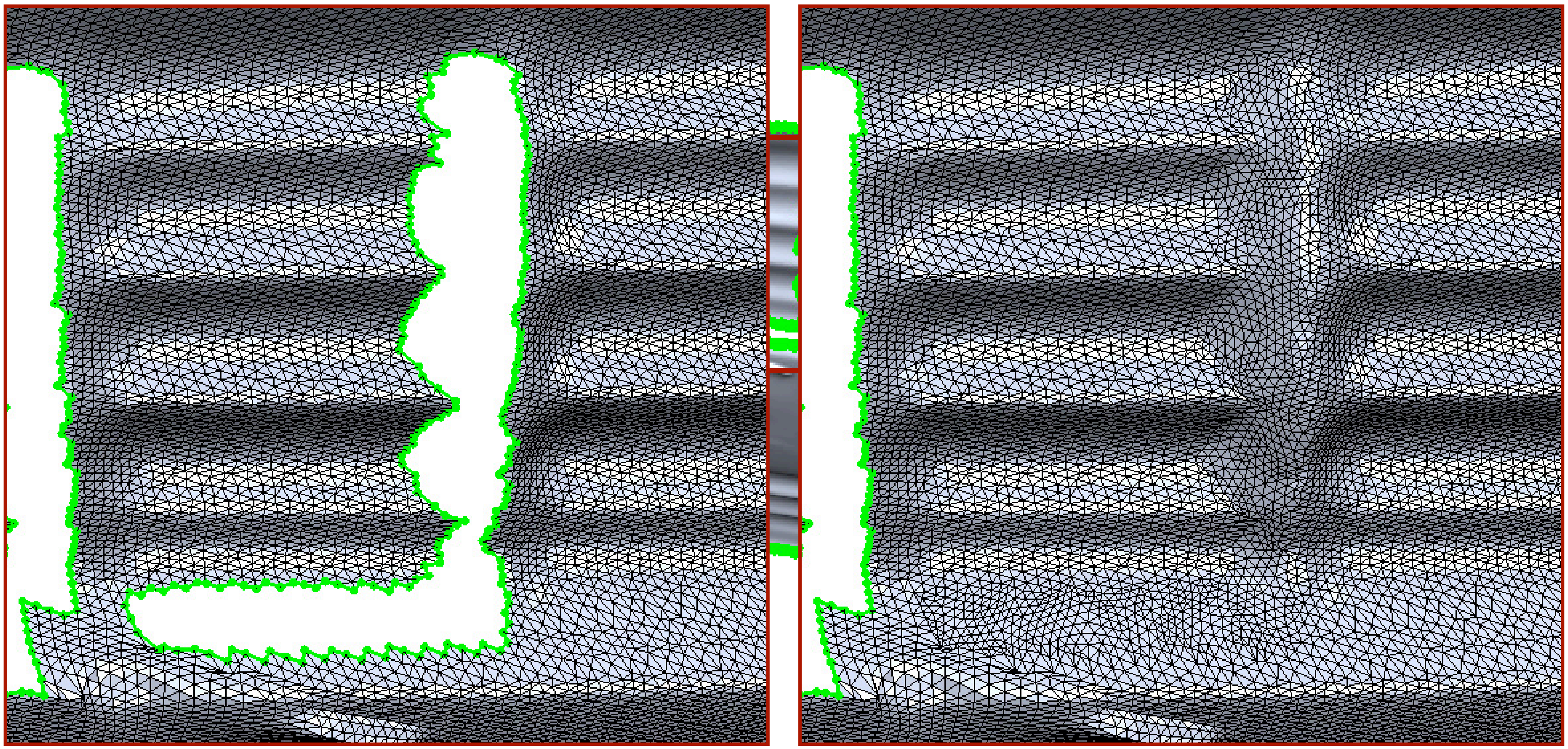
- Scanning artifacts
  - Noise



- Holes
  - Occlusion during scanning



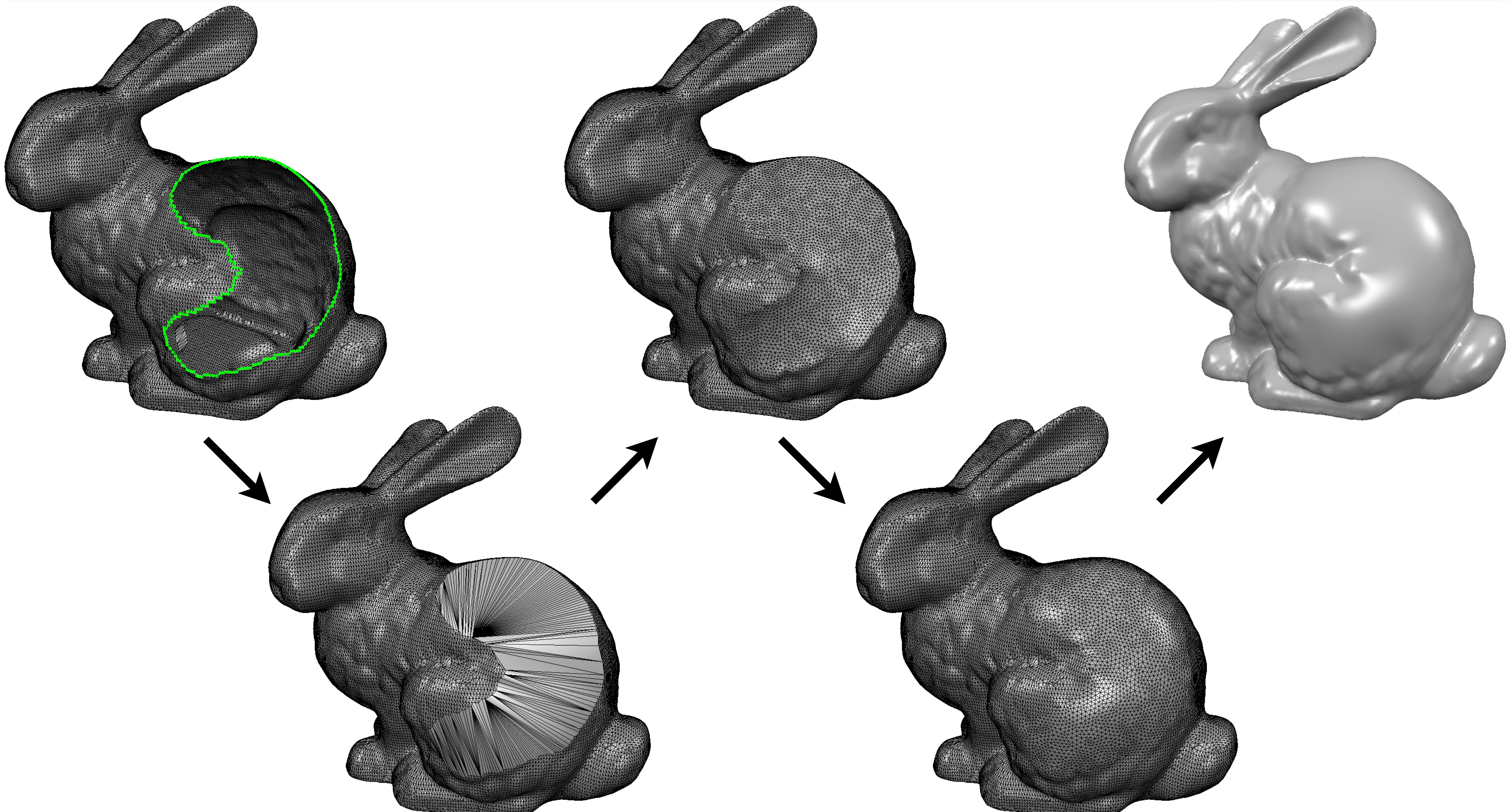
Hole Filling



Hole Filling

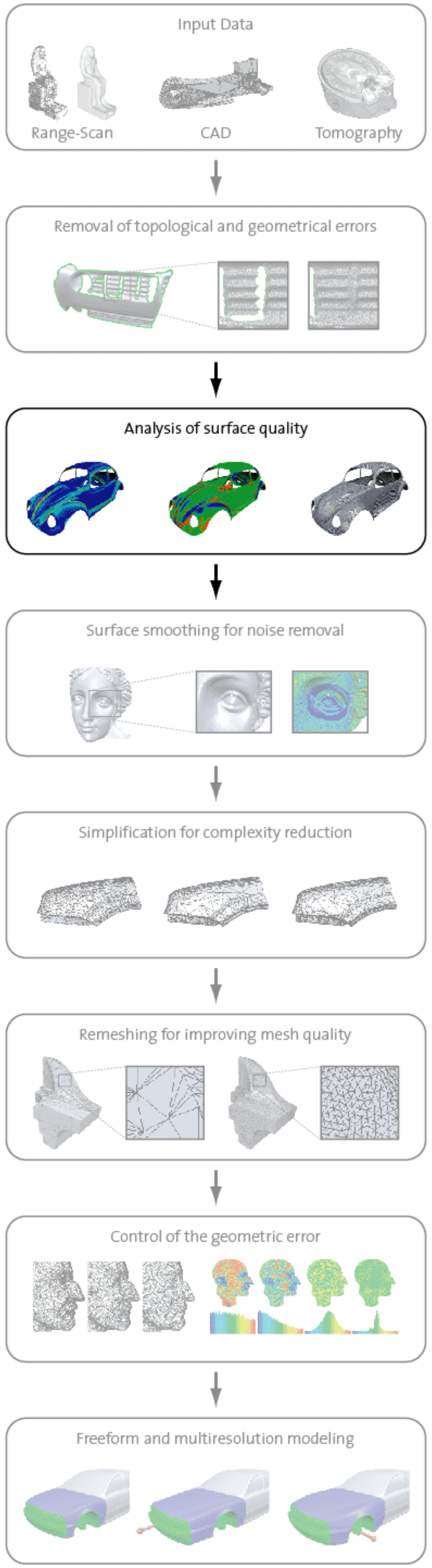
1. Triangulate hole
   – Many possibilities
   – Minimize the maximal dihedral angle
   – Avoids overlaps and fold-overs
2. Refine fill-in
   – Later: isotropic remeshing
3. Smooth fill-in
   – Later: mesh smoothing

Hole Filling

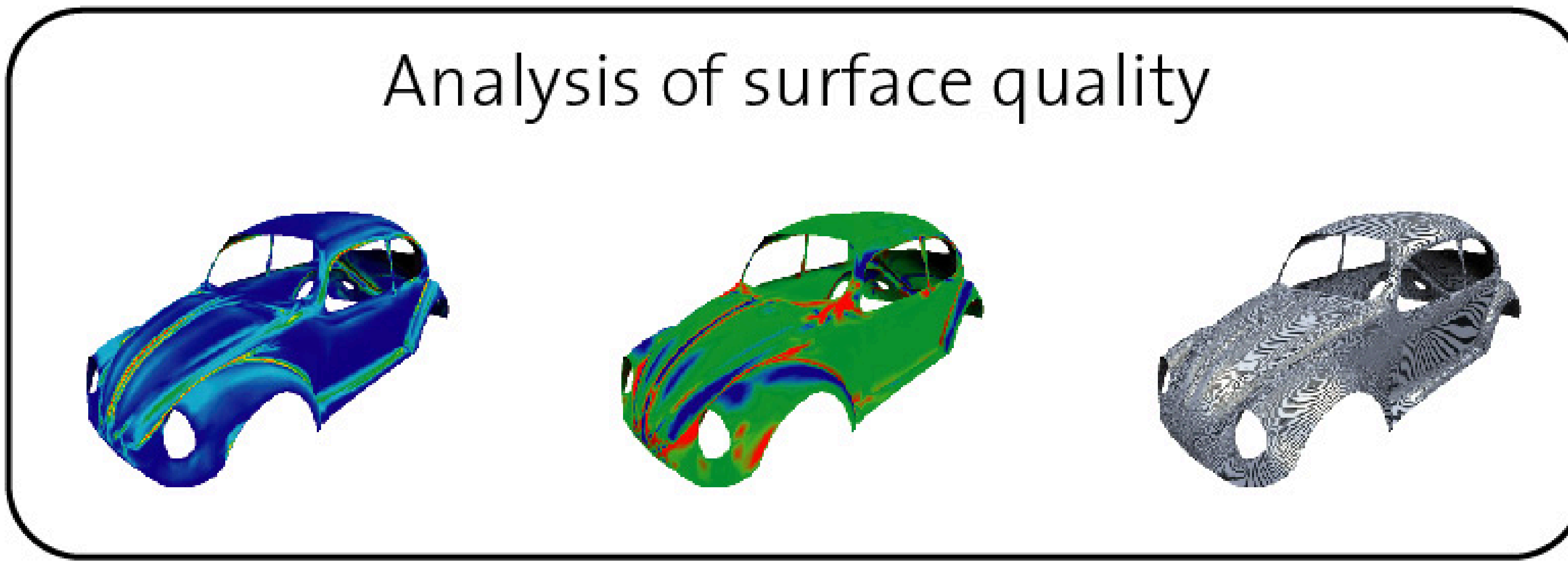


Literature

- Botsch & Kobbelt, *"A Robust Procedure to Eliminate Degenerate Faces from Triangle Meshes"*, VMV 2001
- Peter Liepa, *"Filling holes in meshes"*, Symp. on Geometry Processing 2003
- Bischoff et al, *"Automatic restoration of polygon models"*, ACM Trans. on Graphics 24(4), 2005
- Bischoff & Kobbelt, *"Structure preserving CAD model repair"*, Eurographics 2005



Analysis of Surface Quality



Outline

- Curves and surfaces
- Curvature
  - normal
  - principal
  - mean
  - Gaussian
- Discretization

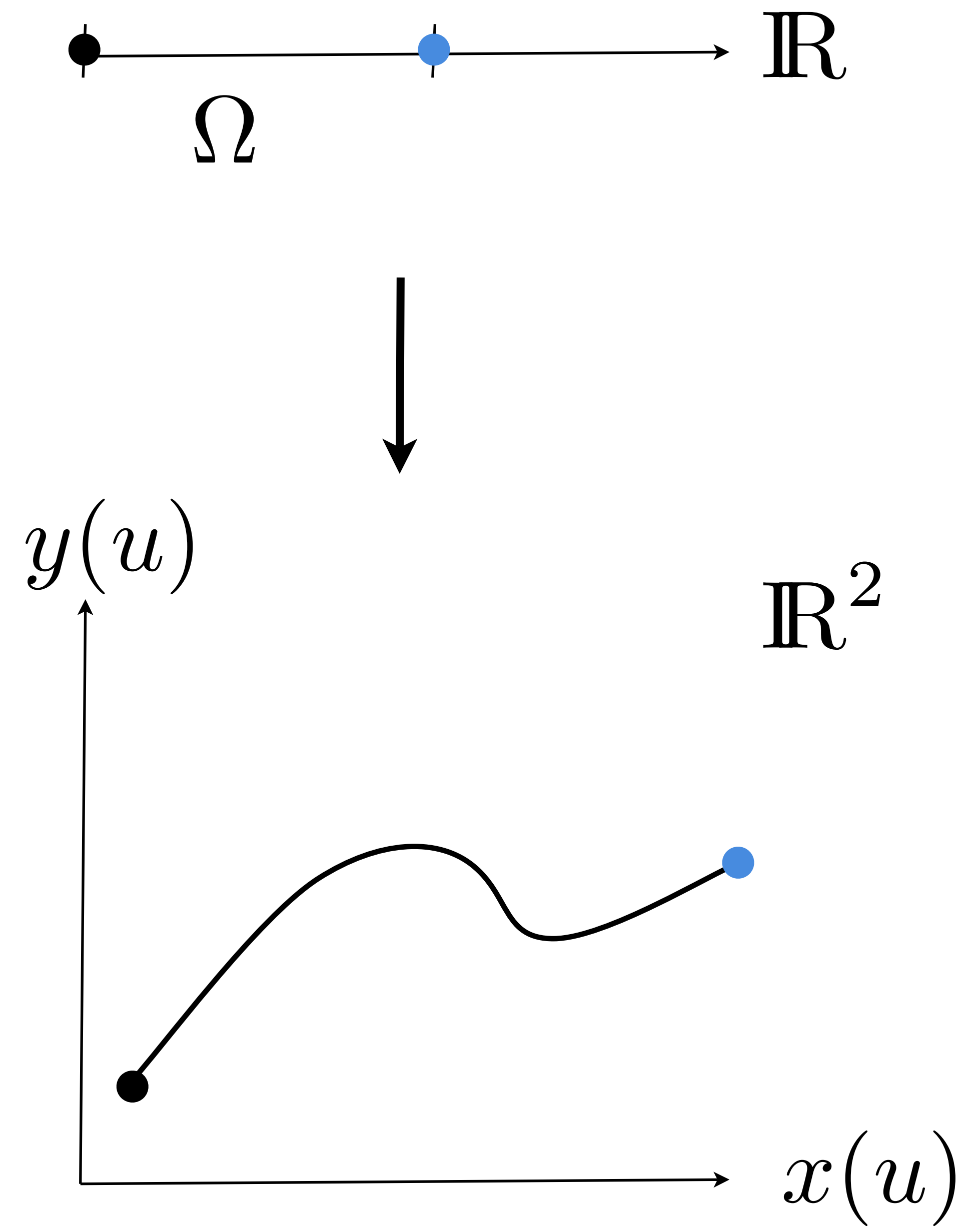
Curves

- Continuous curves

$$\mathbf{f} : \Omega \subset \mathbb{R} \rightarrow \mathbb{R}^d$$

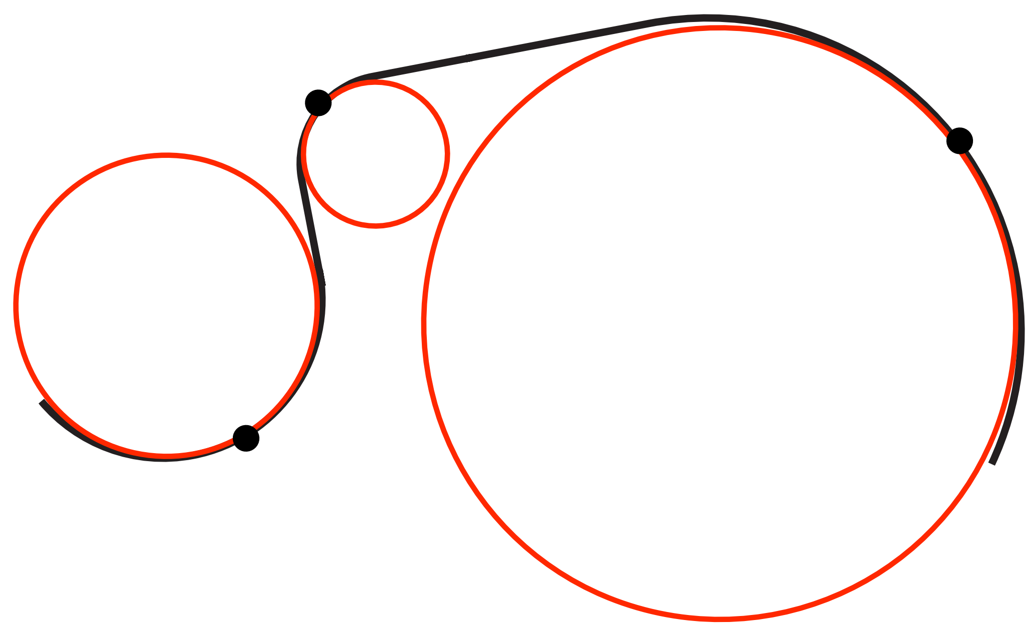
- 2D example

$$\mathbf{f}(u) = \begin{bmatrix} x(u) \\ y(u) \end{bmatrix}$$



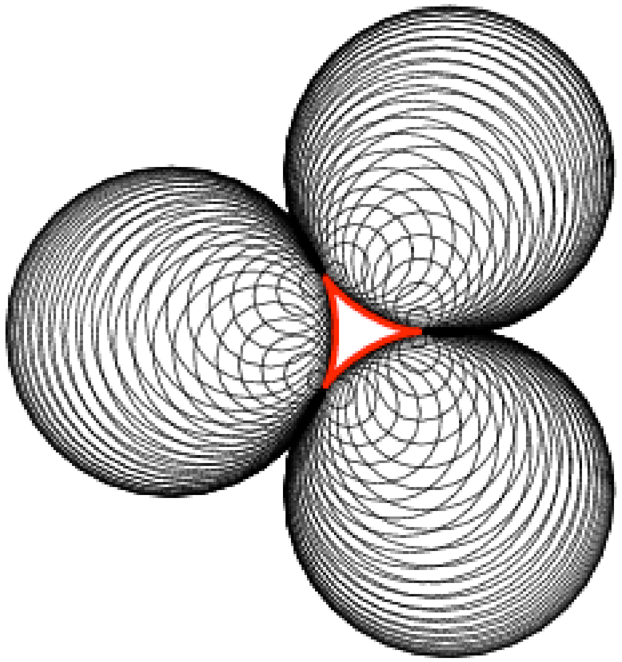
Curvature

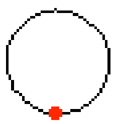
- Continuous curves
  - arc length parameterization: $|\mathbf{f}'(u)| = 1$
  - curvature: $\kappa(u) = |\mathbf{f}''(u)|$
  - inverse of radius of *osculating circle*



Curvature

- Example





from mathworld.wolfram.com

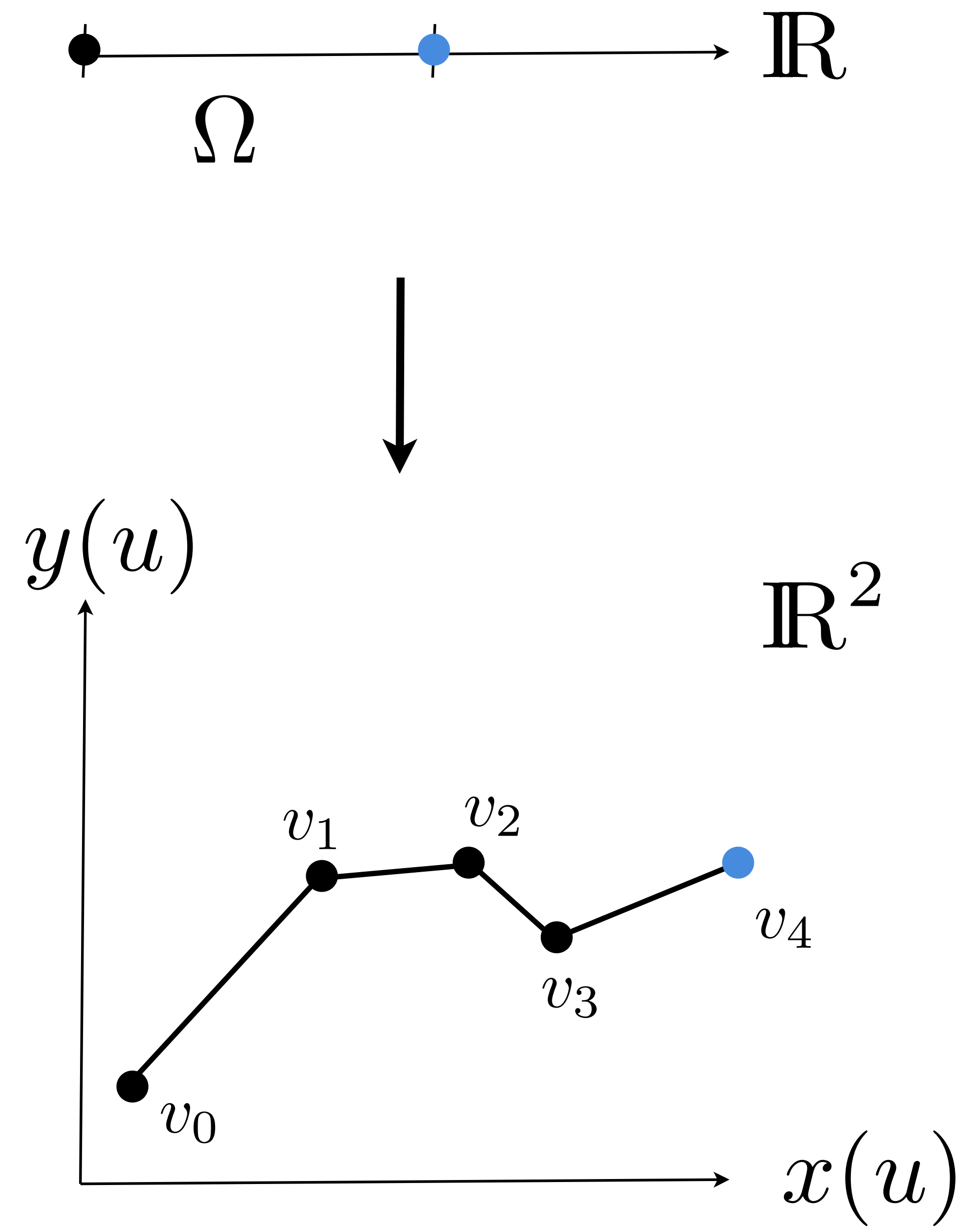
Curves

- Discrete curves

$$\mathbf{f} : \Omega \subset \mathbb{R} \rightarrow \mathbb{R}^d$$

- 2D example

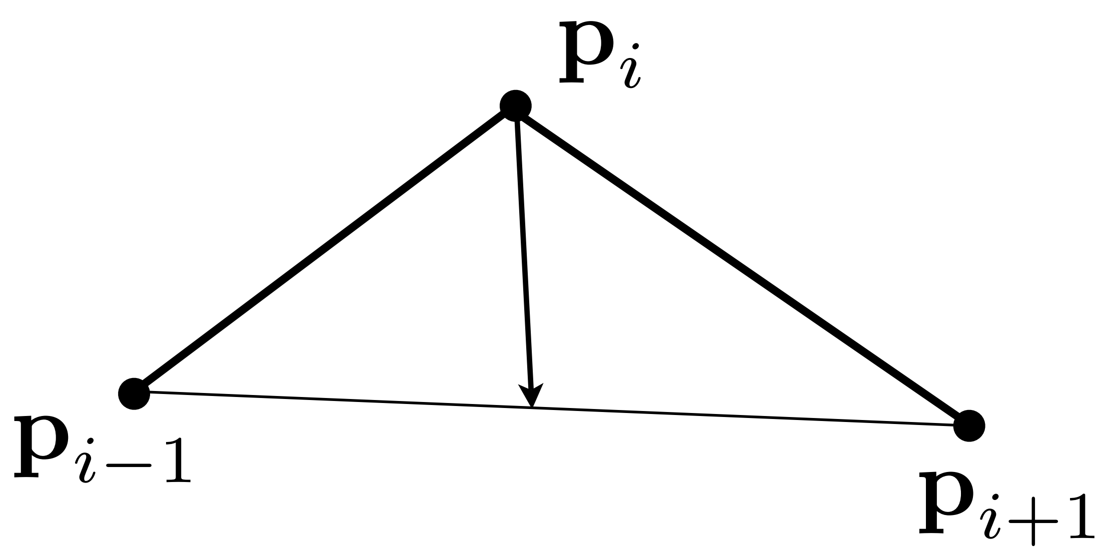
$$\mathbf{f}(u) = \left[\begin{array}{c} x(u) \\ y(u) \end{array} \right]$$



Curvature

- Discrete curves
  - approximate derivatives with divided differences

$$\mathbf{f}''(\mathbf{p}_i) \approx \frac{\mathbf{p}_{i-1} - 2\mathbf{p}_i + \mathbf{p}_{i+1}}{2}$$



  - discrete curvature

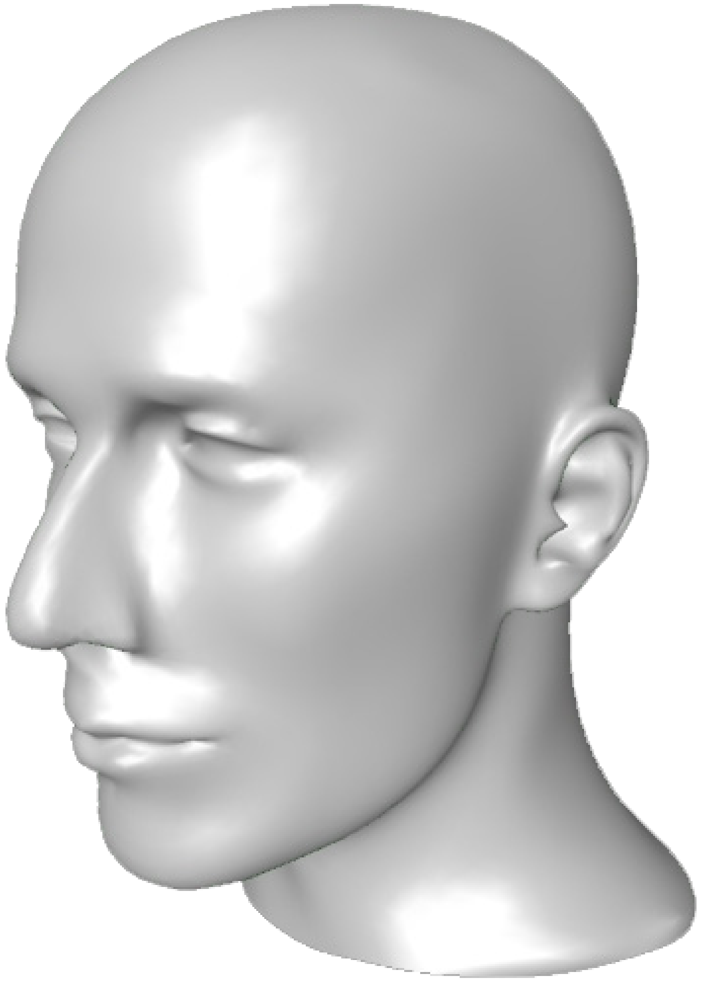
$$\kappa(\mathbf{p}_i) \approx |\mathbf{f}''(\mathbf{p}_i)|$$

Surfaces

- Continuous surfaces

$$\mathbf{f} : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}^d$$

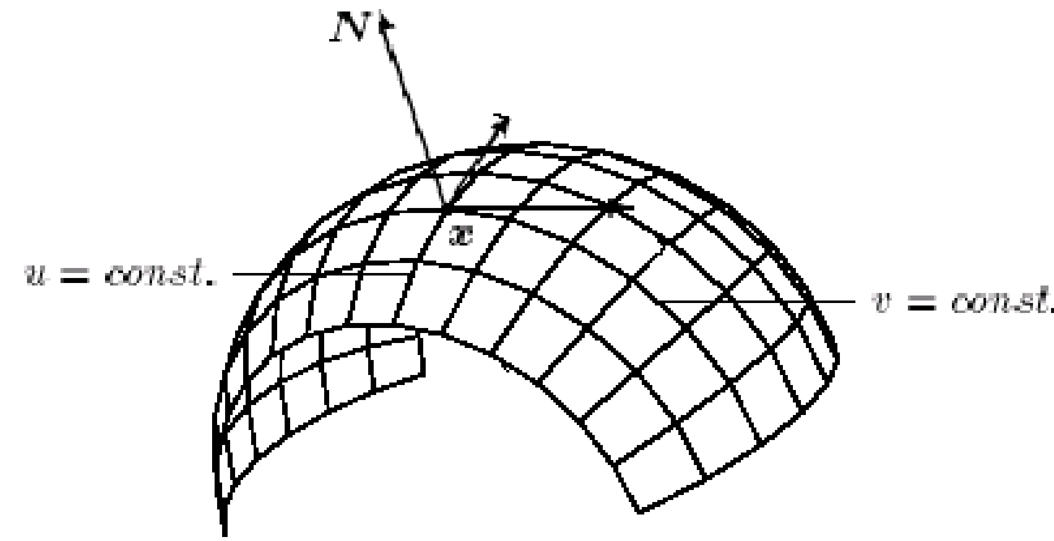
$$\mathbf{f}(u,v) = \begin{bmatrix} x(u,v) \\ y(u,v) \\ z(u,v) \end{bmatrix}$$



Surfaces

- Curves on surfaces

$$\mathbf{c}(t) = \mathbf{f}(u(t), v(t))$$

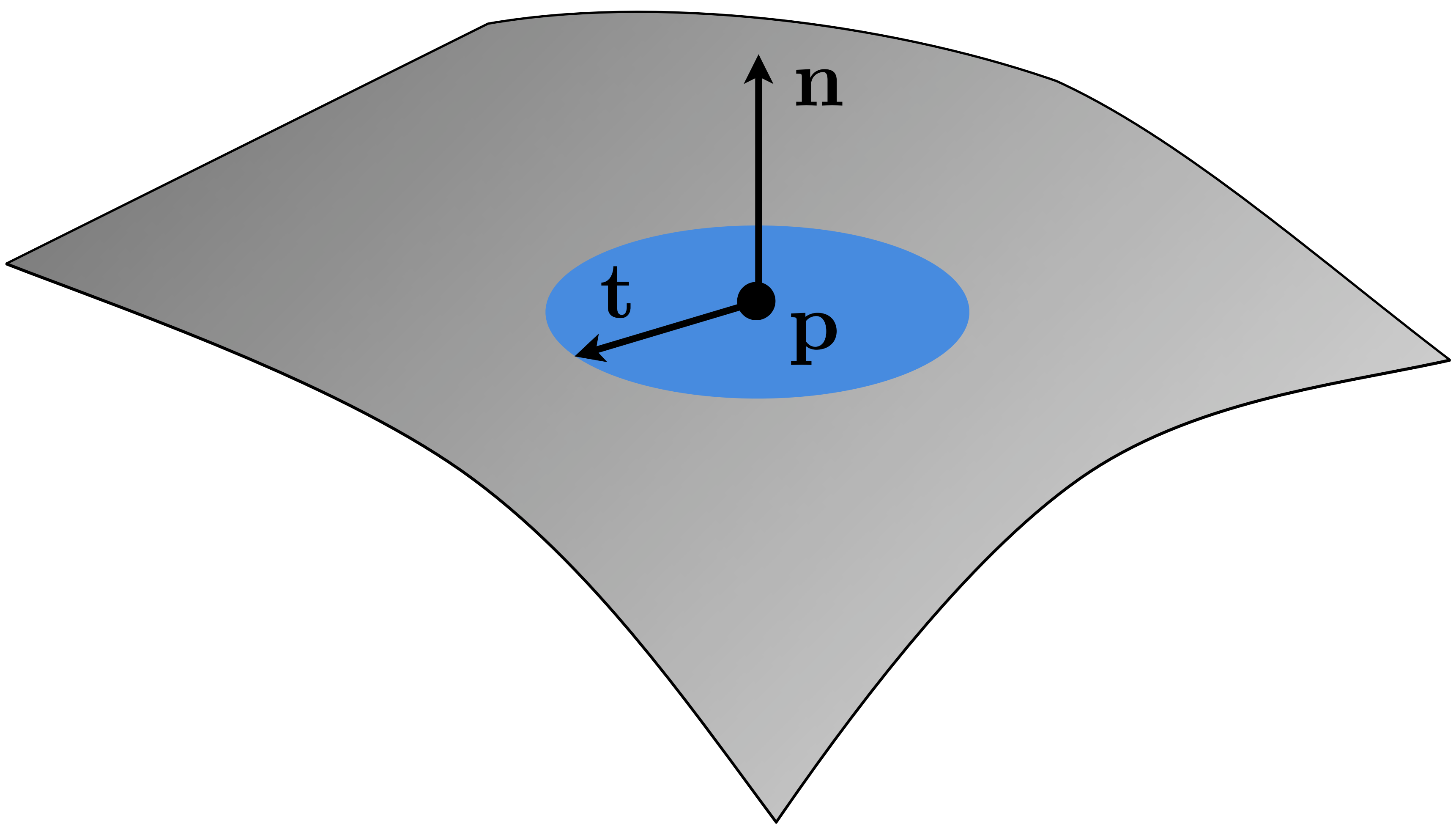


- Example:Parametric lines
  – fix one parameter

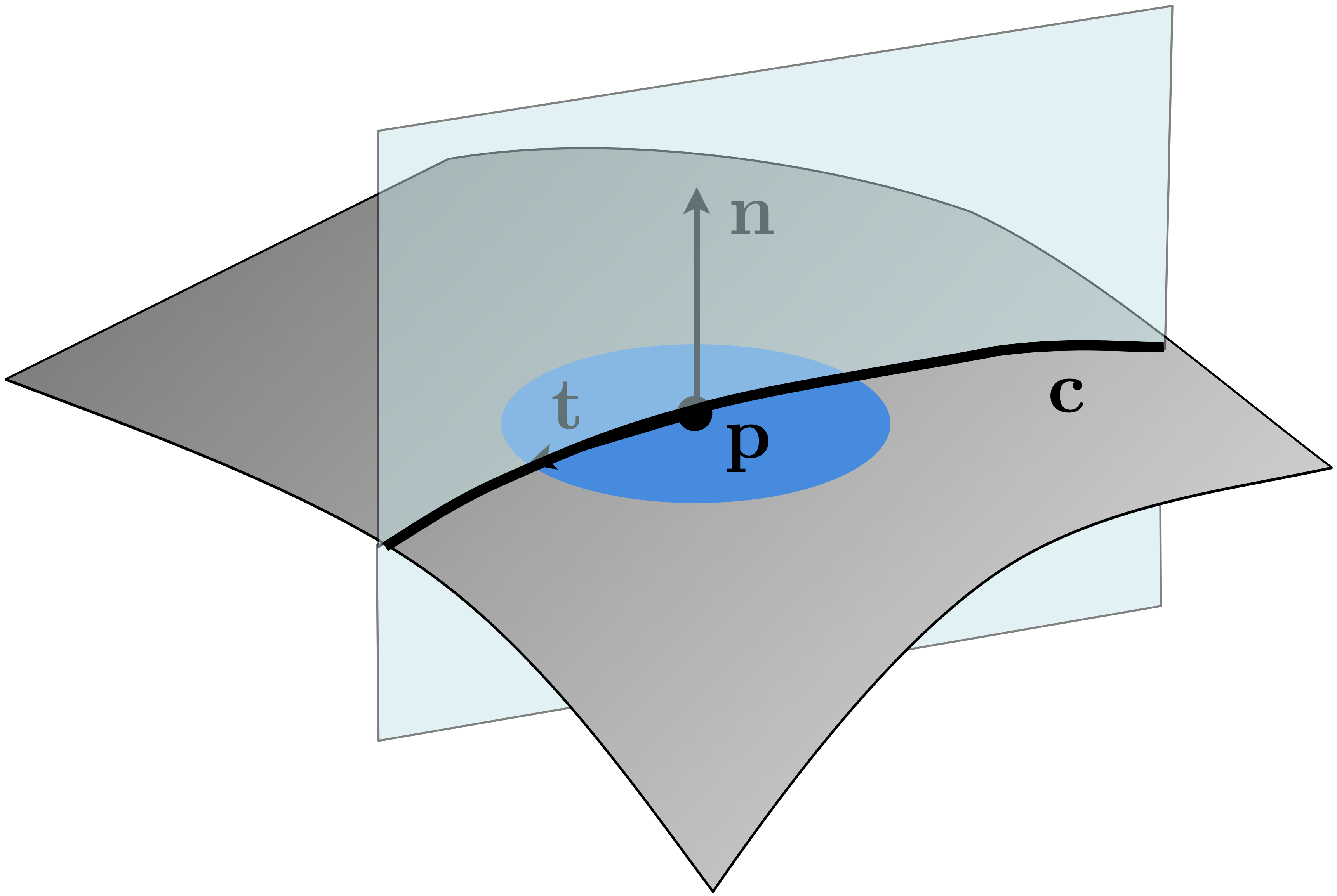
$$\mathbf{u}(t) = \mathbf{f}(t, v) \quad v = const$$

$$\mathbf{v}(t) = \mathbf{f}(u, t) \quad u = const$$

Normal Curvature

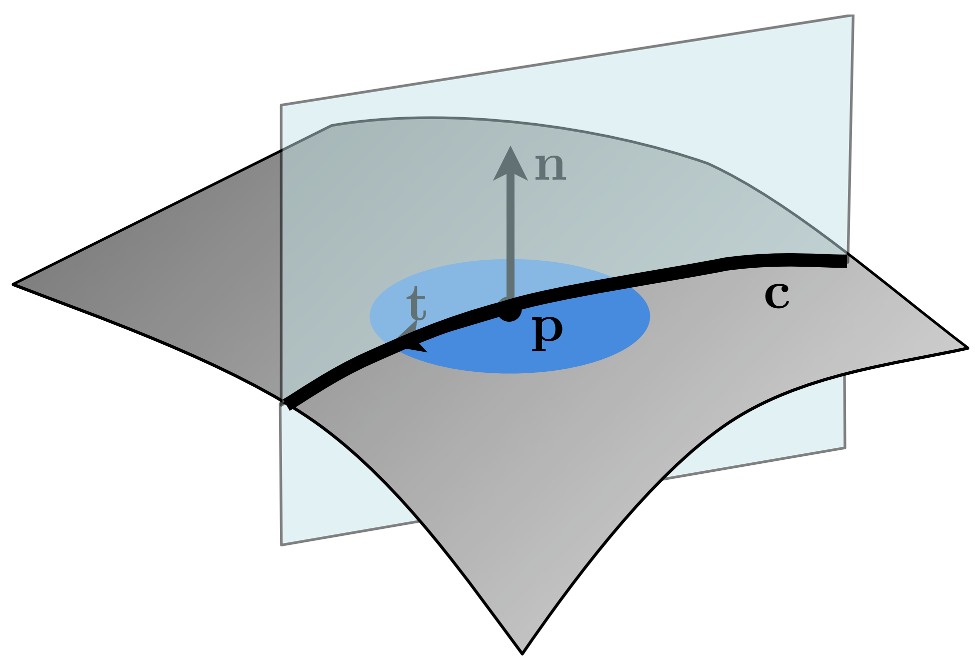


Normal Curvature



Normal Curvature

- Given a normal curve $\mathbf{c} \subset \mathbf{f}(u, v)$ and a point $\mathbf{p} \in \mathbf{c}$
- the normal curvature at p with respect to c is defined as $\kappa_n(\mathbf{p}, \mathbf{c}) = \kappa_{\mathbf{c}}(\mathbf{p})$



Curvature

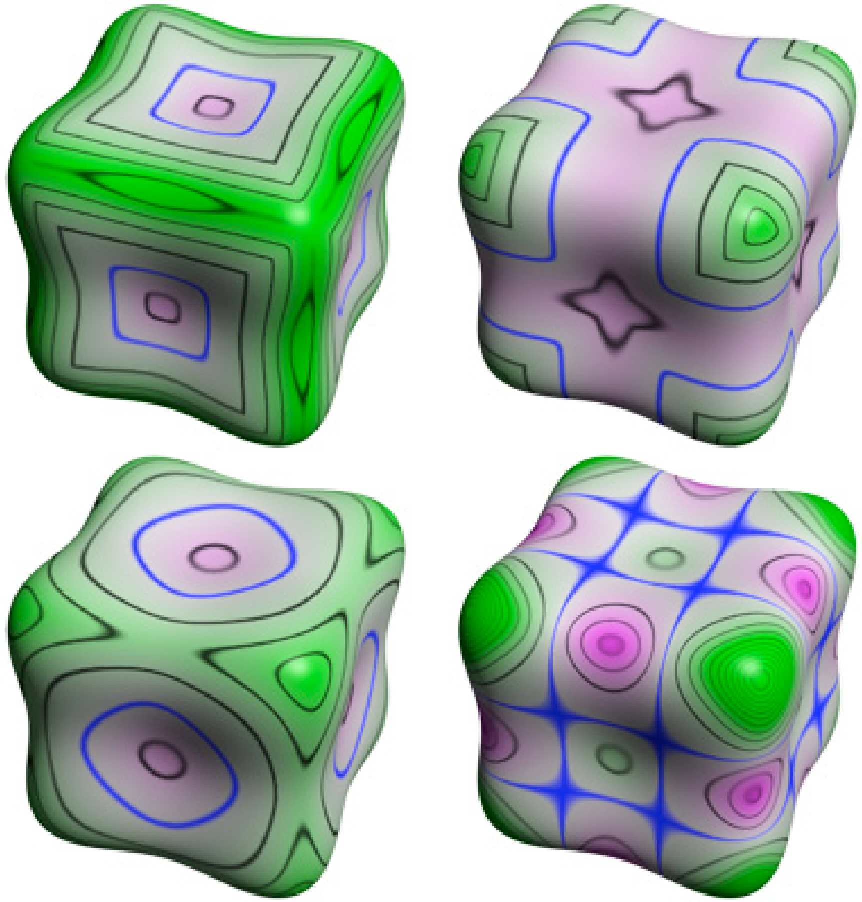
- Principal Curvatures
  - maximum curvature $\kappa_1(\mathbf{p}) = \max_{\mathbf{c}} \kappa_{\mathbf{c}}(\mathbf{p})$
  - minimum curvature $\kappa_2(\mathbf{p}) = \min_{\mathbf{c}} \kappa_{\mathbf{c}}(\mathbf{p})$
- Mean Curvature $H = \frac{1}{2}(\kappa_1 + \kappa_2)$
- Gaussian Curvature $K = \kappa_1 \cdot \kappa_2$

Curvature

- Example

$$\kappa_1(\mathbf{p}) = \max_{\mathbf{c}} \kappa_{\mathbf{c}}(\mathbf{p})$$

$$\kappa_2(\mathbf{p}) = \min_{\mathbf{c}} \kappa_{\mathbf{c}}(\mathbf{p})$$

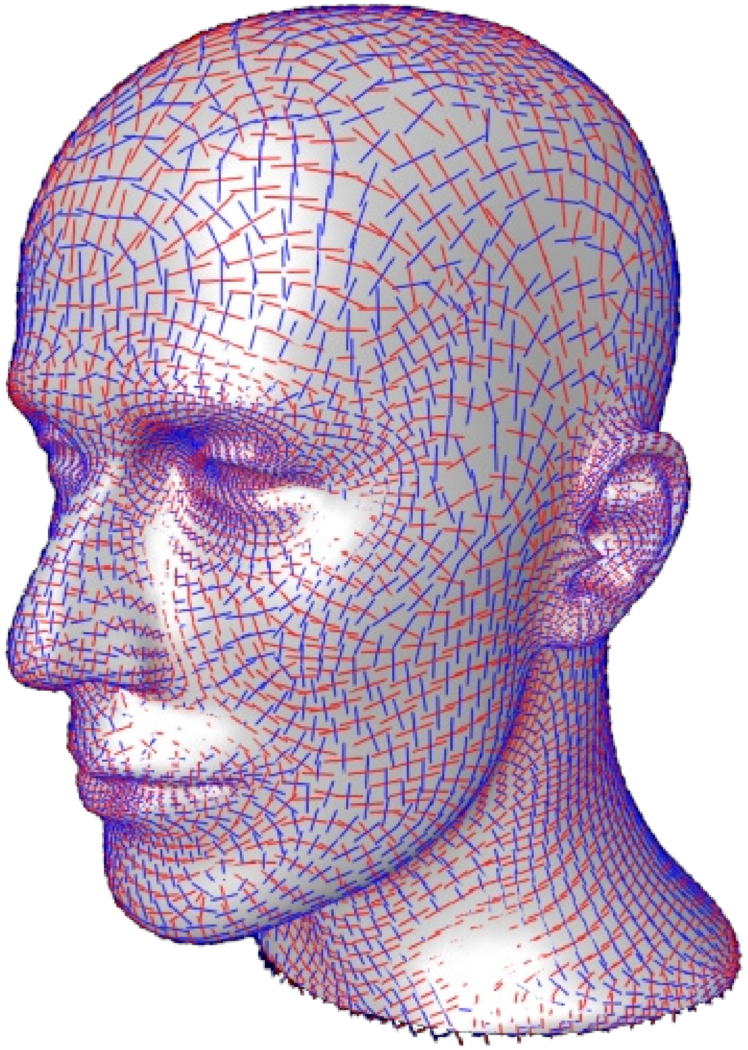


$$H = \frac{1}{2}(\kappa_1 + \kappa_2)$$

$$K = \kappa_1 \cdot \kappa_2$$

Curvature

- Principal Directions
  - tangents to curve of minimum resp. maximum curvature



http://www-sop.inria.fr/geometrica/team/Pierre.Alliez/demos/curvature/

Curvature on Meshes

- Laplace-Beltrami operator

surface — mean curvature

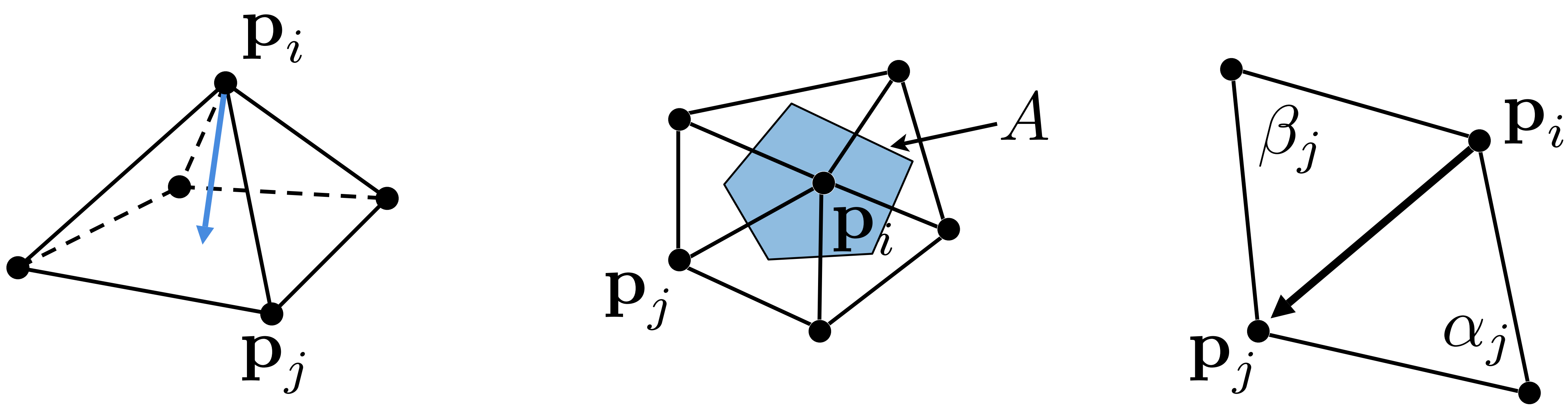
$$\Delta_B \mathbf{S} = -\mathbf{n} H$$

surface normal

Curvature on Meshes

- Discrete Laplace-Beltrami operator

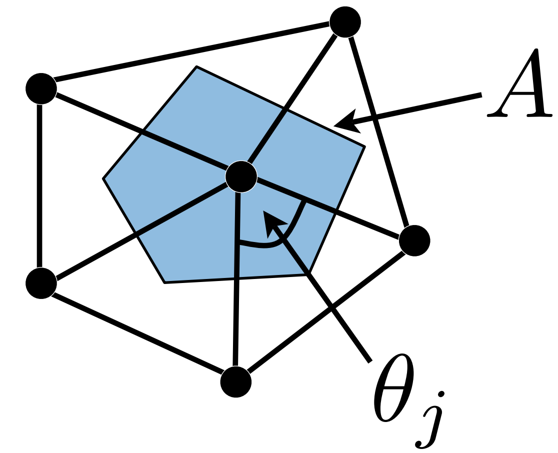
$$\Delta_B \mathbf{p}_i = \frac{1}{2A} \sum_j (\cot \alpha_{ij} + \cot \beta_{ij})(\mathbf{p}_j - \mathbf{p}_i)$$



Meyer, Desbrun, Schroeder, Barr: *Discrete Differential-Geometry Operators For Triangulated 2-Manifolds*, VisMath 2002

Curvature on Meshes

- Mean curvature $H = |\Delta_B \mathbf{p}_i|$
- Gaussian curvature



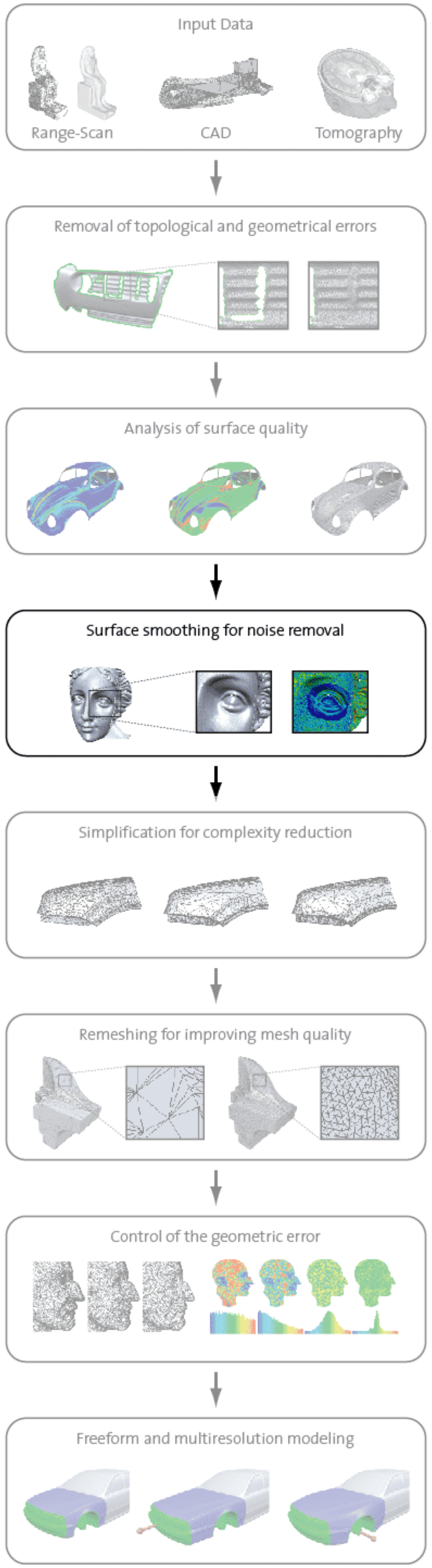
$$G = (2\pi - \sum_j \theta_j)/A$$

- Principal curvatures

$$\kappa_1 = H + \sqrt{H^2 - G} \qquad \kappa_2 = H - \sqrt{H^2 - G}$$

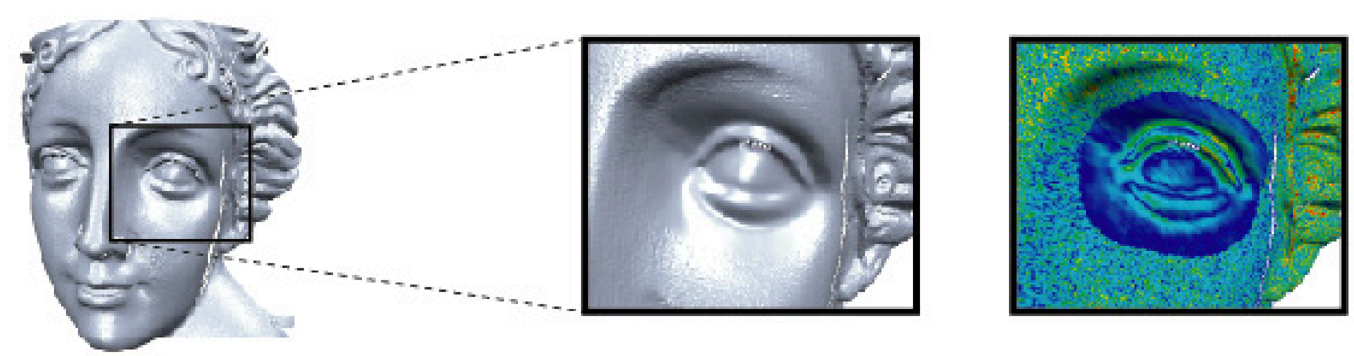
Links & Literature

- Pierre Alliez: *Estimating Curvature Tensors on Triangle Meshes* (source code)
  - http://www-sop.inria.fr/geometrica/team/Pierre.Alliez/demos/curvature/
- Meyer, Desbrun, Schröder, Barr: *Discrete Differential-Geometry Operators for Triangulated 2-Manifolds*, VisMath 2002.



Surface Smoothing

Surface smoothing for noise removal

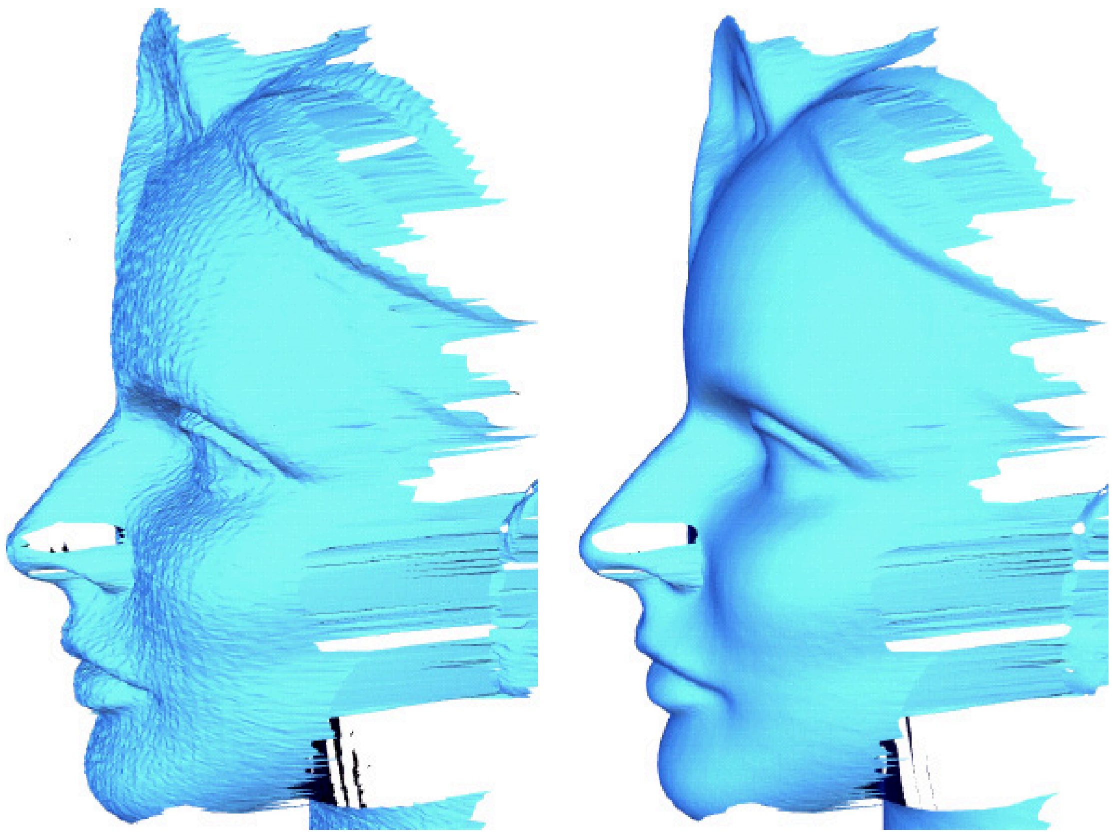


Outline

- Motivation
- Smoothing as Diffusion
  - iterative Laplacian smoothing
- Smoothing as Energy Minimization
  - membrane & thin plate functionals

Motivation

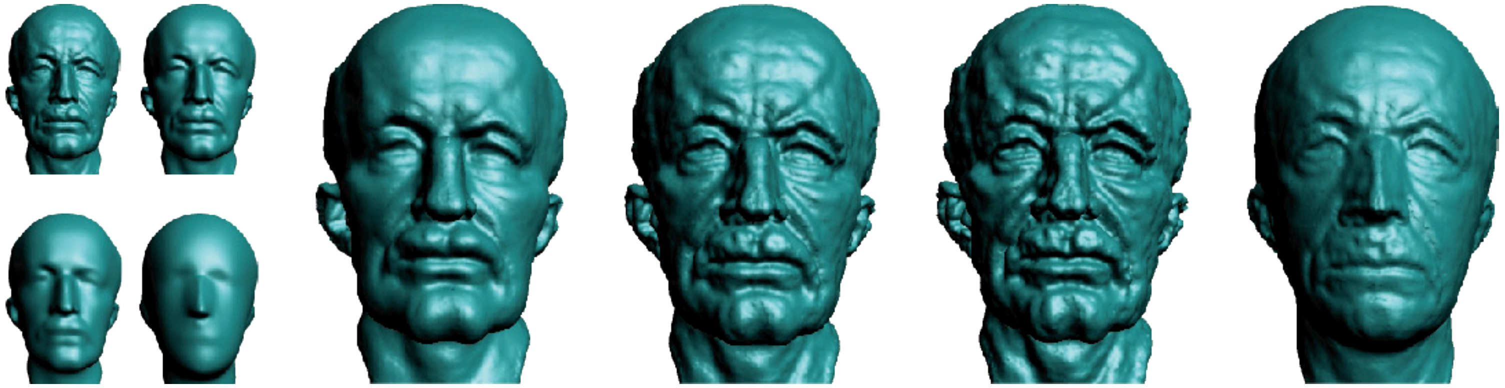
- Filter out high frequency components for noise removal



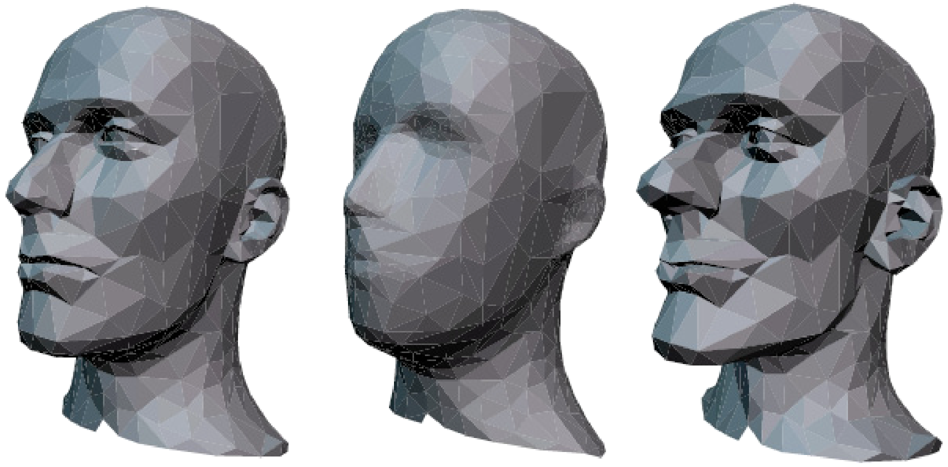
Desbrun, Meyer, Schroeder, Barr: *Implicit Fairing of Irregular Meshes using Diffusion and Curvature Flow*, SIGGRAPH 99

Motivation

- Advanced Filtering



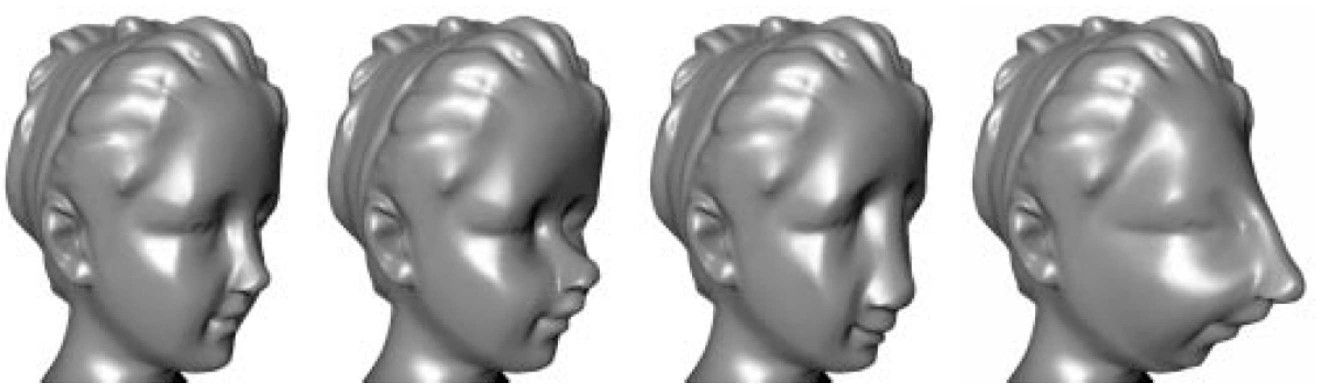
Pauly, Kobbelt, Gross: *Point-Based Multi-Scale Surface Representation*, ACM TOG 2006



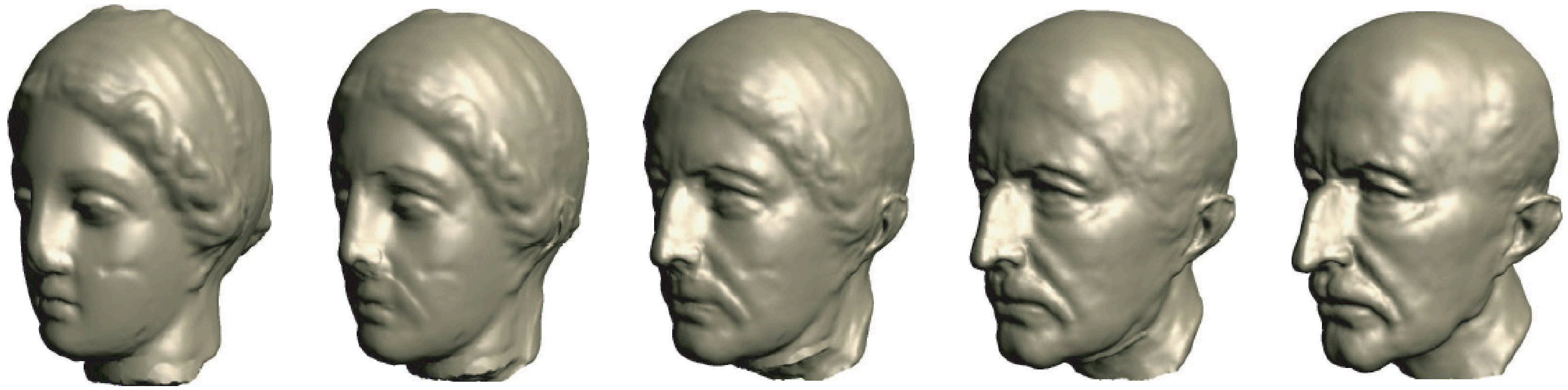
Guskow, Sweldens, Schroeder: *Multiresolution Signal Processing for Meshes*, SIGGRAPH 99

Motivation

- Multi-resolution Editing & Morphing



Kobbelt, Campagna, Vorsatz, Seidel: *Interactive Multi-Resolution Modeling on Arbitrary Meshes*, SIGGRAPH 98



Pauly, Kobbelt, Gross: *Point-Based Multi-Scale Surface Representation*, ACM TOG, 2006

Diffusion

- Diffusion equation

diffusion constant

$$\frac{\partial f}{\partial t} = \lambda \Delta f$$

Laplace operator

Diffusion on Meshes

- Discretization of diffusion equation

$$\frac{\partial \mathbf{p}_i}{\partial t} = \lambda \Delta \mathbf{p}_i$$

- leads to simple update rule

$$\mathbf{p}_i \leftarrow \mathbf{p}_i + \lambda \Delta \mathbf{p}_i$$

Laplacian Operator

- Laplace Operator

$$\Delta f = f_{uu} + f_{vv}$$

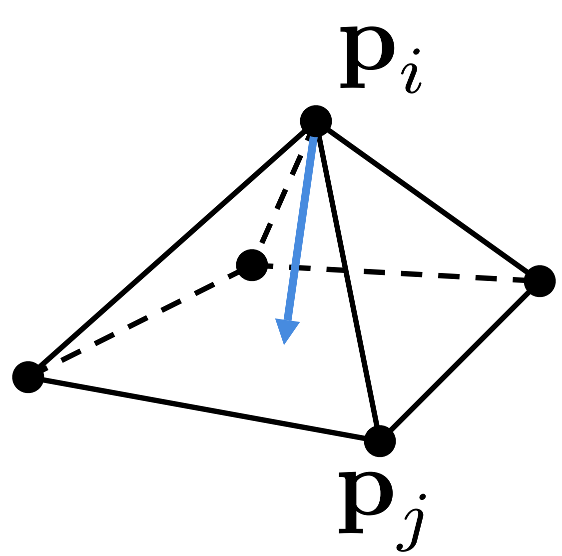
- Discrete Laplacian on meshes

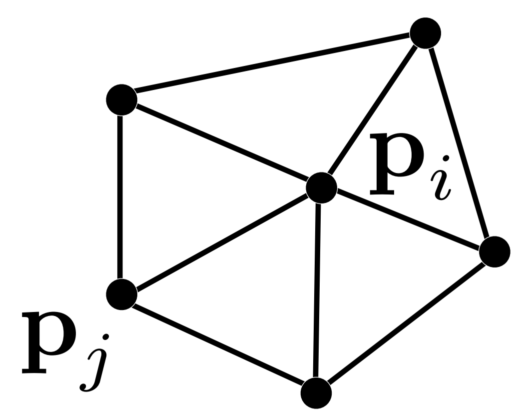
$$\Delta \mathbf{p}_i = \mu_i \sum_j \omega_{ij} (\mathbf{p}_j - \mathbf{p}_i)$$

Laplacian Operator

- Uniform (umbrella operator)

$$\omega_{ij} = 1 \quad \rightarrow \quad \Delta \mathbf{p}_i = \frac{1}{N_i} \sum_{j \in N_i} (\mathbf{p}_j - \mathbf{p}_i)$$

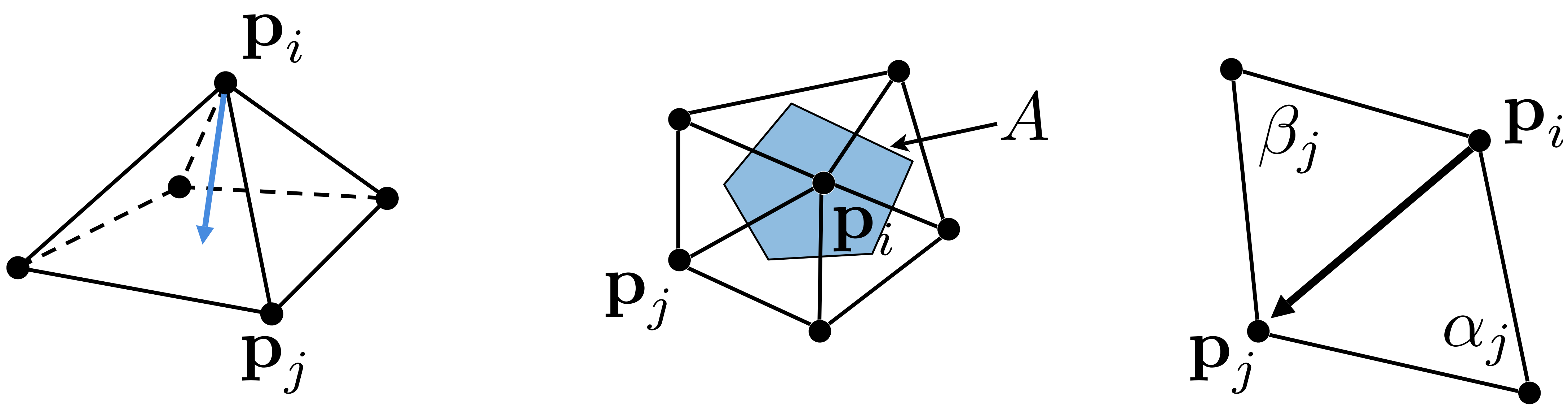




Laplacian Operator

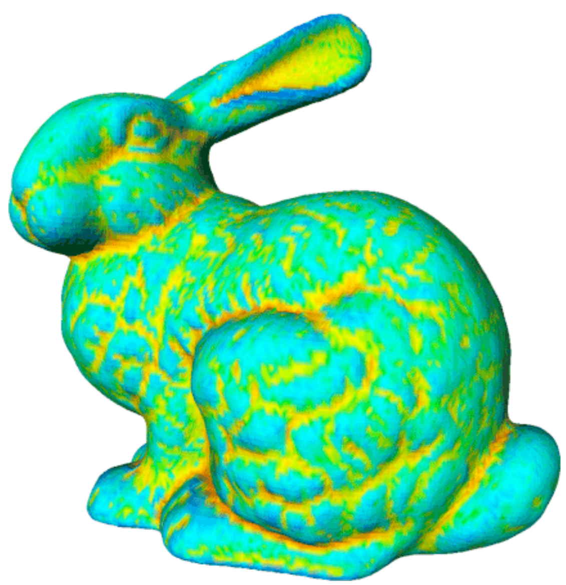
- Laplace-Beltrami operator

$$\Delta_B \mathbf{p}_i = \frac{1}{2A} \sum_j (\cot \alpha_{ij} + \cot \beta_{ij})(\mathbf{p}_j - \mathbf{p}_i)$$

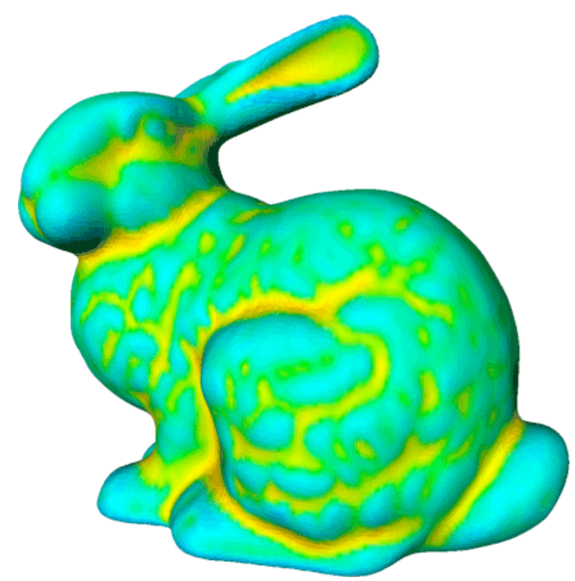


Diffusion on Meshes

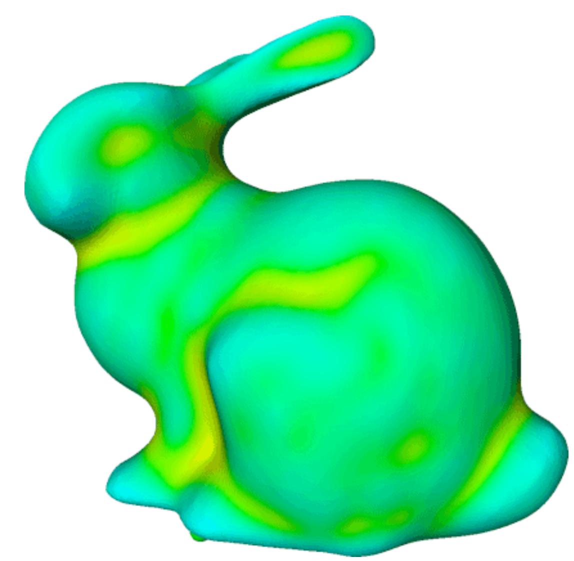
- Iterate $\mathbf{p}_i \leftarrow \mathbf{p}_i + \lambda \Delta \mathbf{p}_i$



0 Iterations



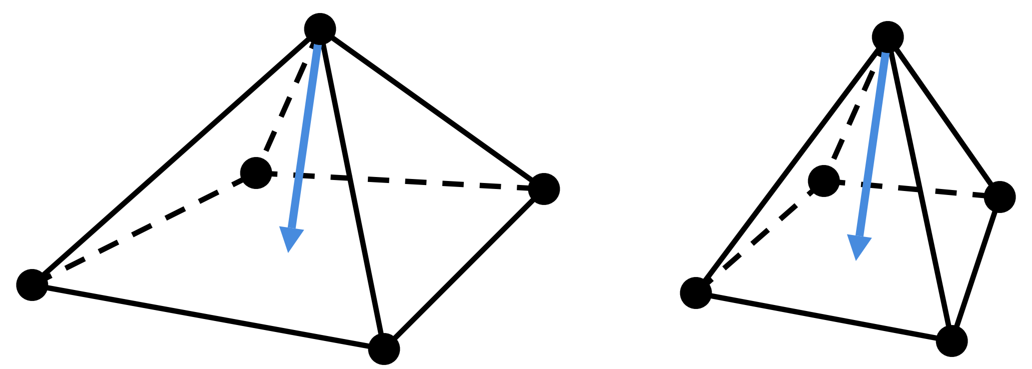
5 Iterations

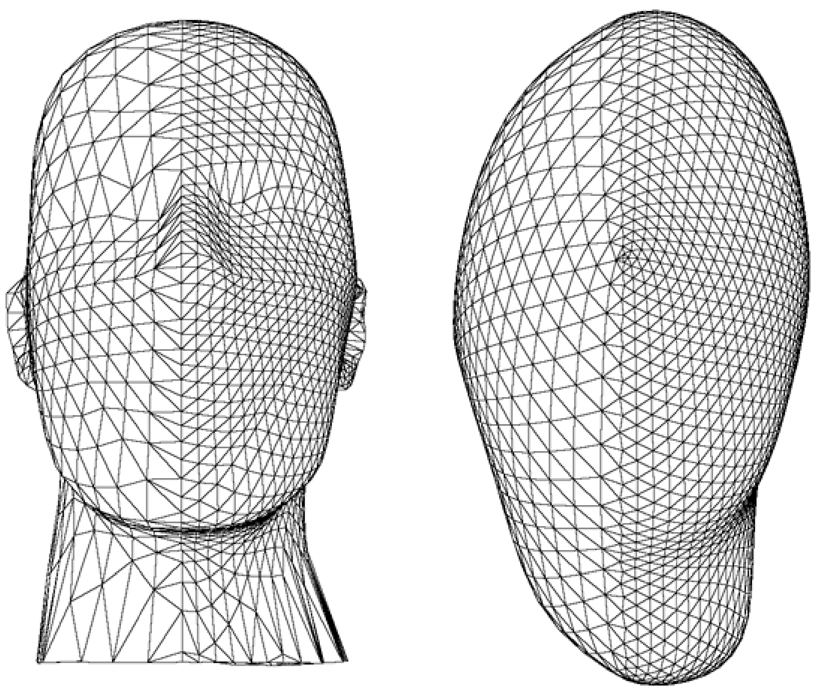


20 Iterations

Linear Umbrella Operator

- Smoothes geometry and discretization
- Frequency Confusion
  - linear umbrella operator can evaluate to the same vector even for different geometry 'frequencies'
- Vertex drift can lead to distortions

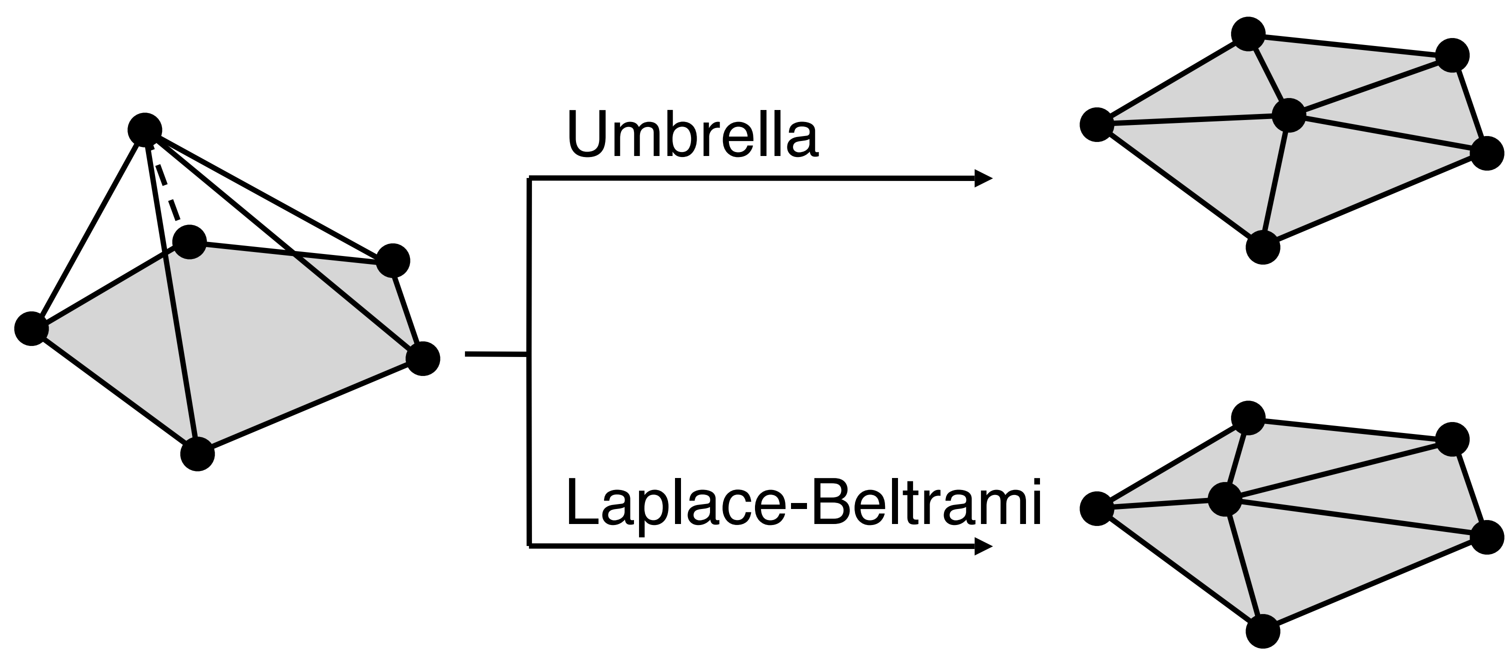




Desbrun et al., Siggraph 1999

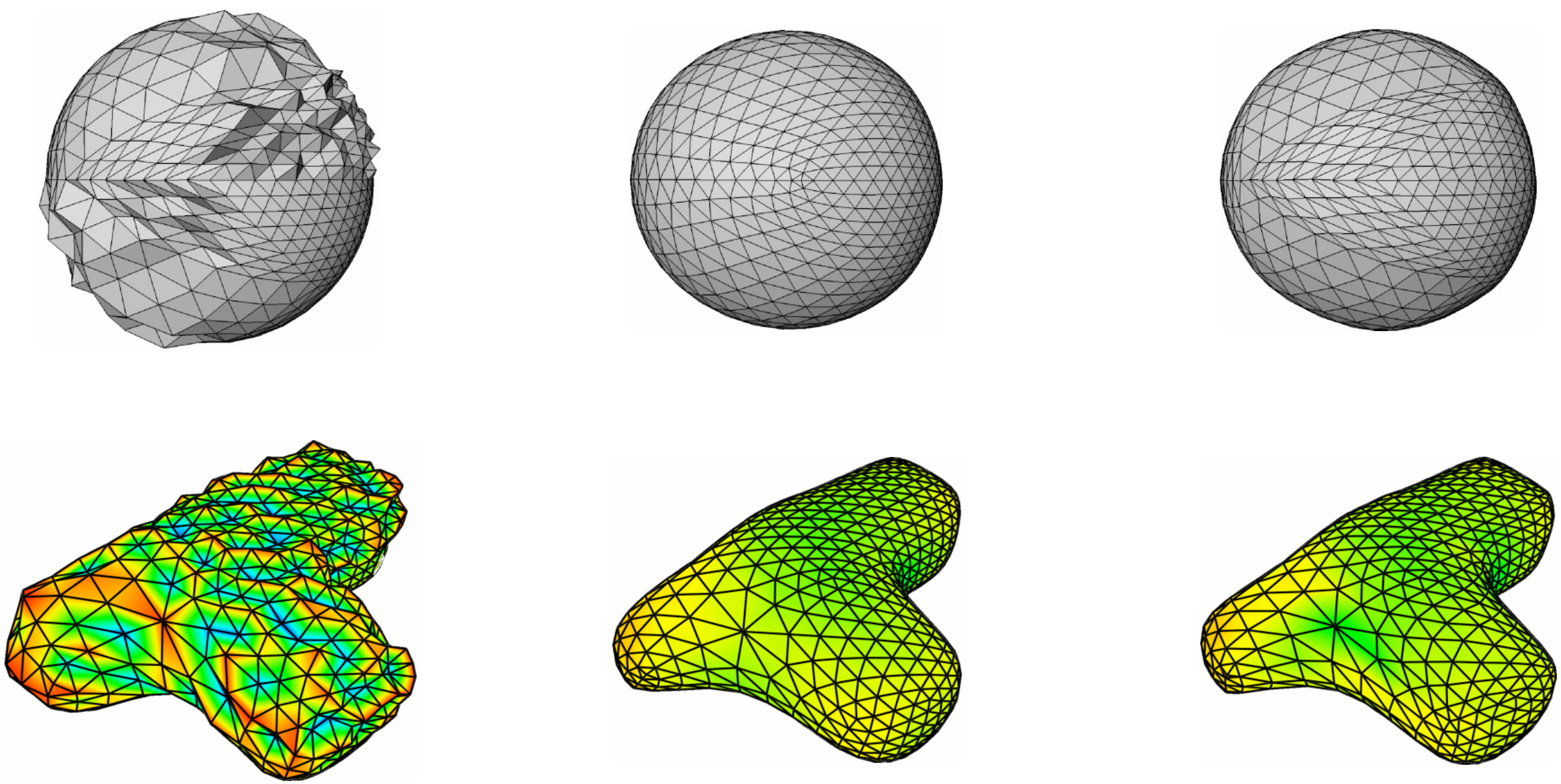
Non-linear Laplace-Beltrami

- Vertices can only move along their normal
  - no vertex drifting in parameter space



Comparison

Original | Umbrella | Laplace-Beltrami



Energy Minimization

- Surface fairing as *energy minimization*
  - Minimize the thin-plate energy

$$E(S) = \int_S \kappa_1^2 + \kappa_2^2 dS$$

  - with appropriate boundary constraints

$$\partial S = c$$

$$\mathbf{n}(\partial S) = d$$

Energy Minimization

- Variational Calculus
  - parameterization $f : \Omega \to \mathbb{R}^3$
  - membrane energy

$$\int_\Omega f_u^2 + f_v^2 du dv \to \min$$

  - variational formulation

$$\Delta f = f_{uu} + f_{vv} = 0$$

Energy Minimization

- Variational Calculus
  - parameterization $f : \Omega \to \mathbb{R}^3$
  - thin-plate energy

$$\int_{\Omega} f_{uu}^2 + 2f_{uv}^2 + f_{vv}^2 dudv$$

  - variational formulation

$$\Delta^2 f = f_{uuuu} + 2f_{uuvv} + f_{vvvv} = 0$$

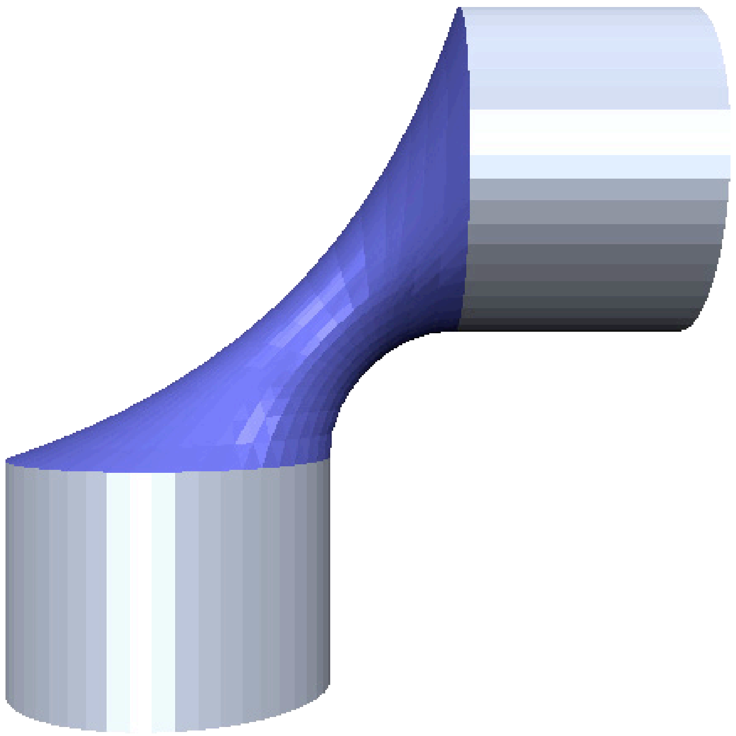
Linear System Characteristics

- Sparse linear system (κ ≈ 7 / n)

$$[\ldots, \omega_{ij}, \ldots, -\sum_j \omega_{ij}, \ldots, \omega_{ij}, \ldots]$$

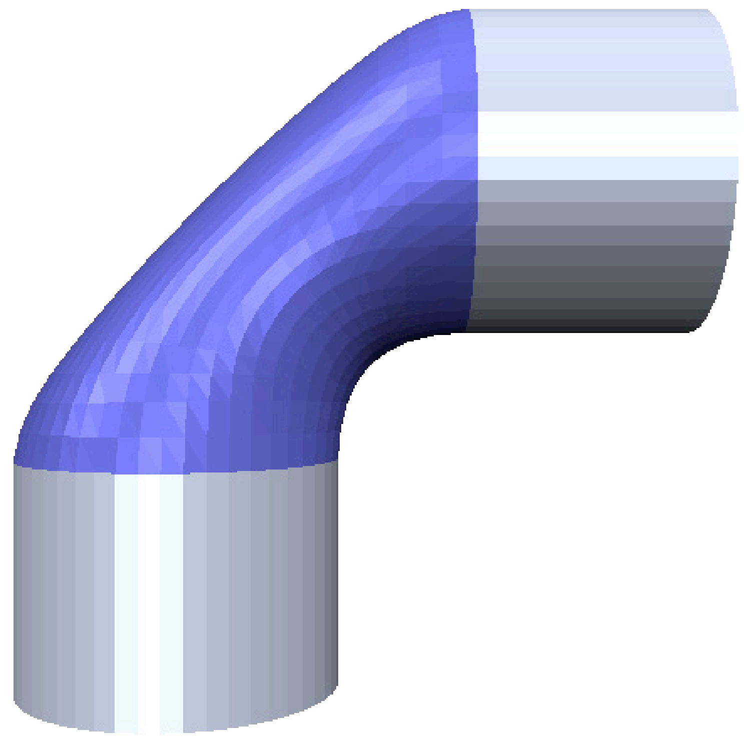
- Positive weights
  - weakly diagonal dominant
  - Linear: w_{ij} computed once
  - Non-stationary: w_{ij} updated in every step
- Laplace-update: iterative solver

Comparison



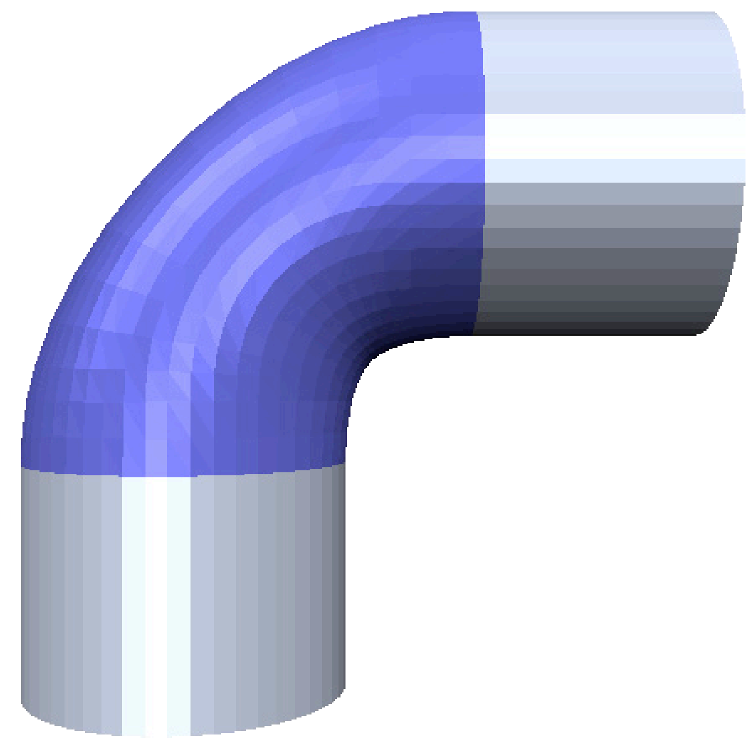
- Membrane

$$\Delta \mathbf{v}_i = 0$$



- Thin Plate

$$\Delta^2 \mathbf{v}_i = 0$$



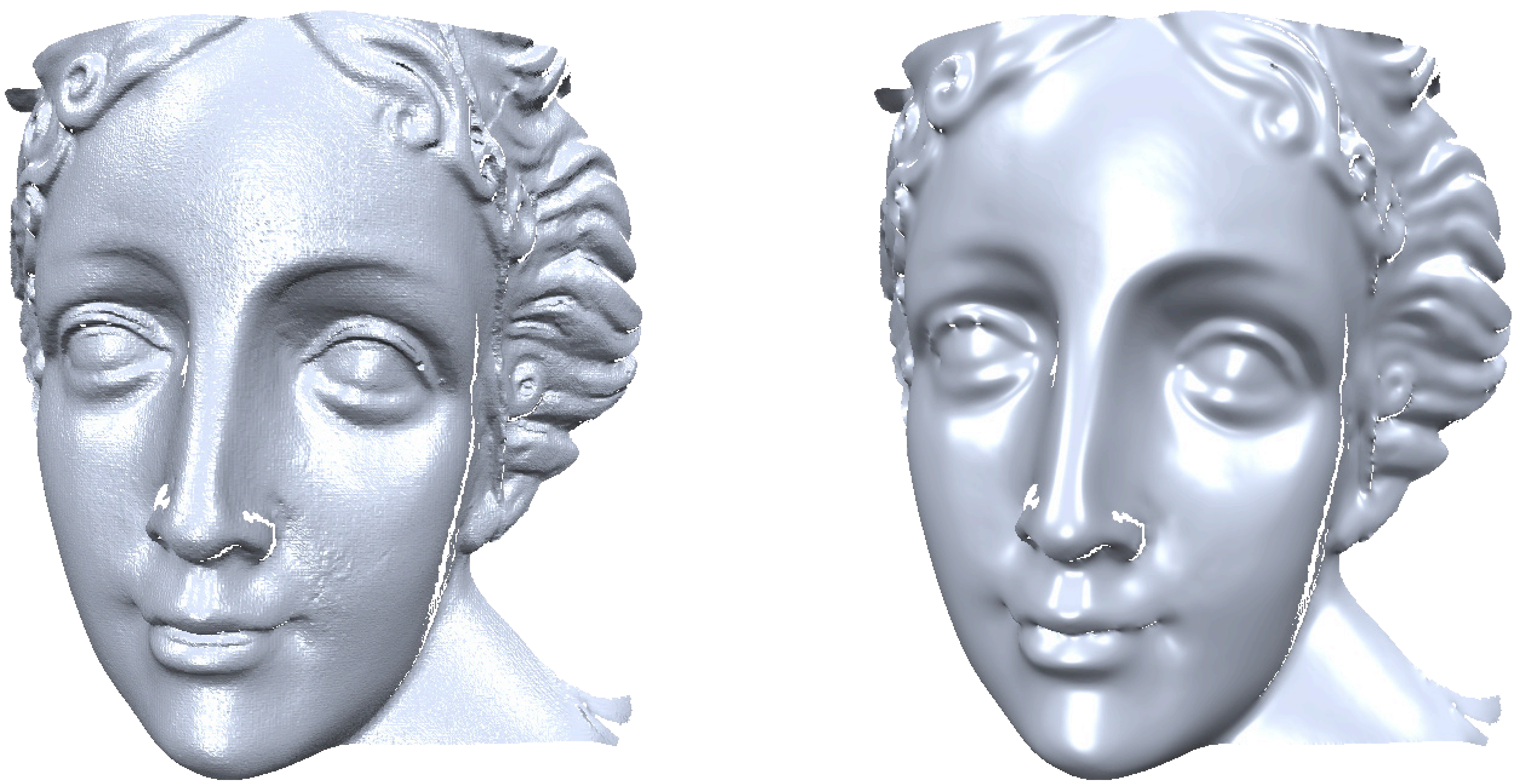
$$\Delta^3 \mathbf{v}_i = 0$$

Laplacian Smoothing

- Geometric interpretation
  - Laplacian smoothing approximates (hinged) membrane surfaces
- Physical justification
  - membranes (soap films) are smooth

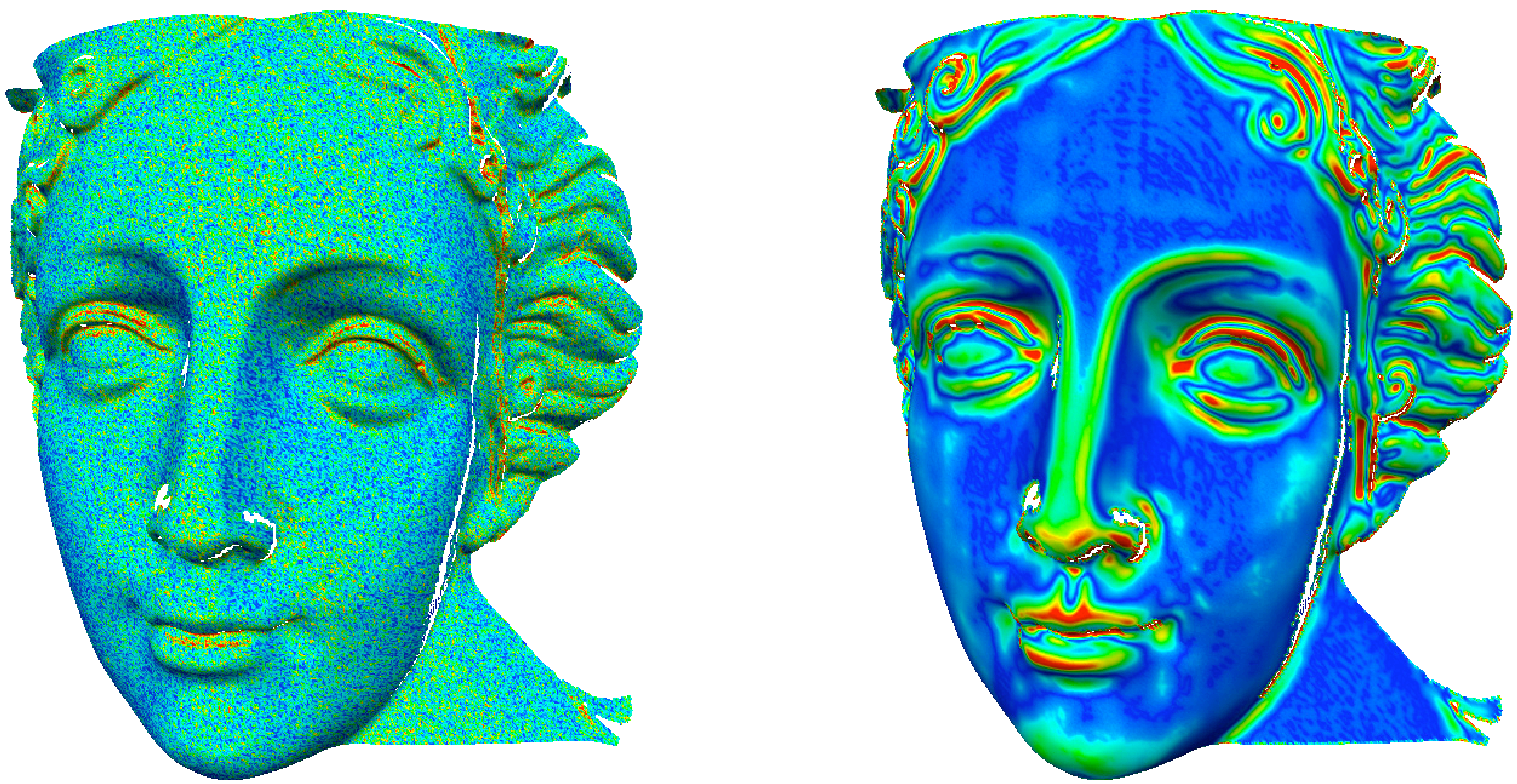
Applications

- Noise removal



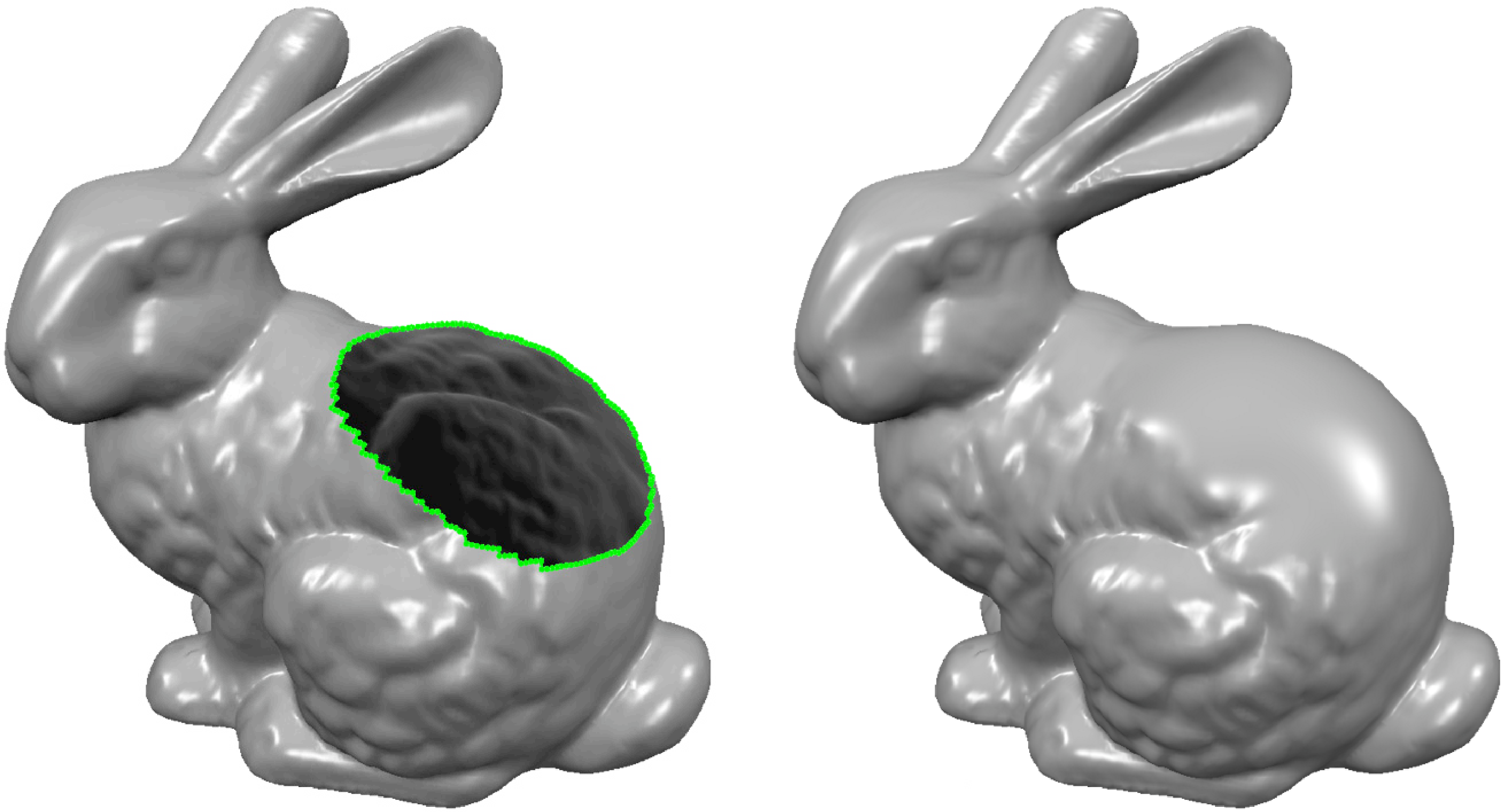
Applications

- Noise removal



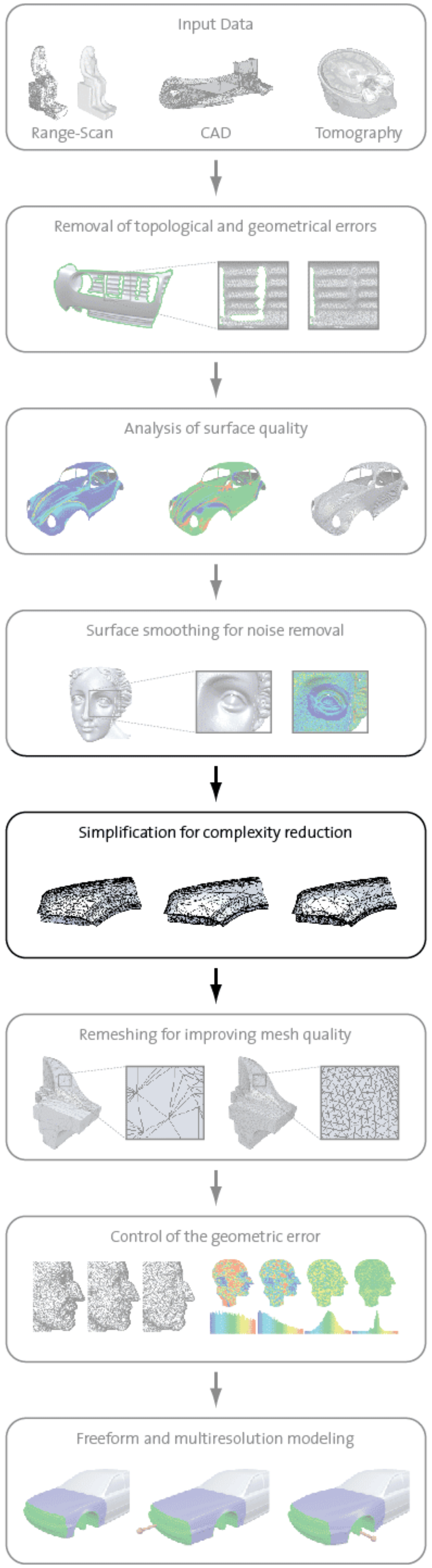
Applications

- Hole-filling

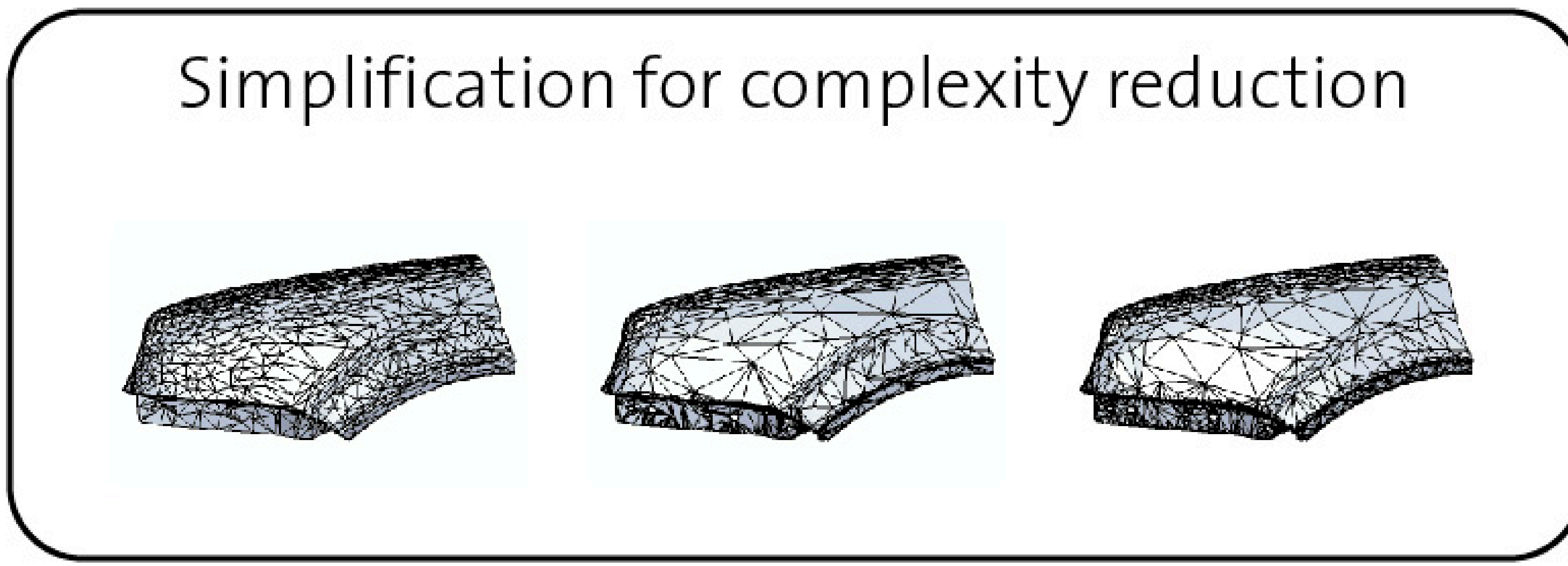


Links & Literature

- http://openmesh.org/
- Desbrun, Meyer, Schroeder, Barr: *Implicit Fairing of Irregular Meshes using Diffusion and Curvature Flow*, SIGGRAPH 99
- Taubin: *A signal processing approach to fair surface design*, SIGGRAPH 1996
- Botsch, Kobbelt: An Intuitive Framework for Real-Time Freeform Modeling, SIGGRAPH 2004



Mesh Simplification



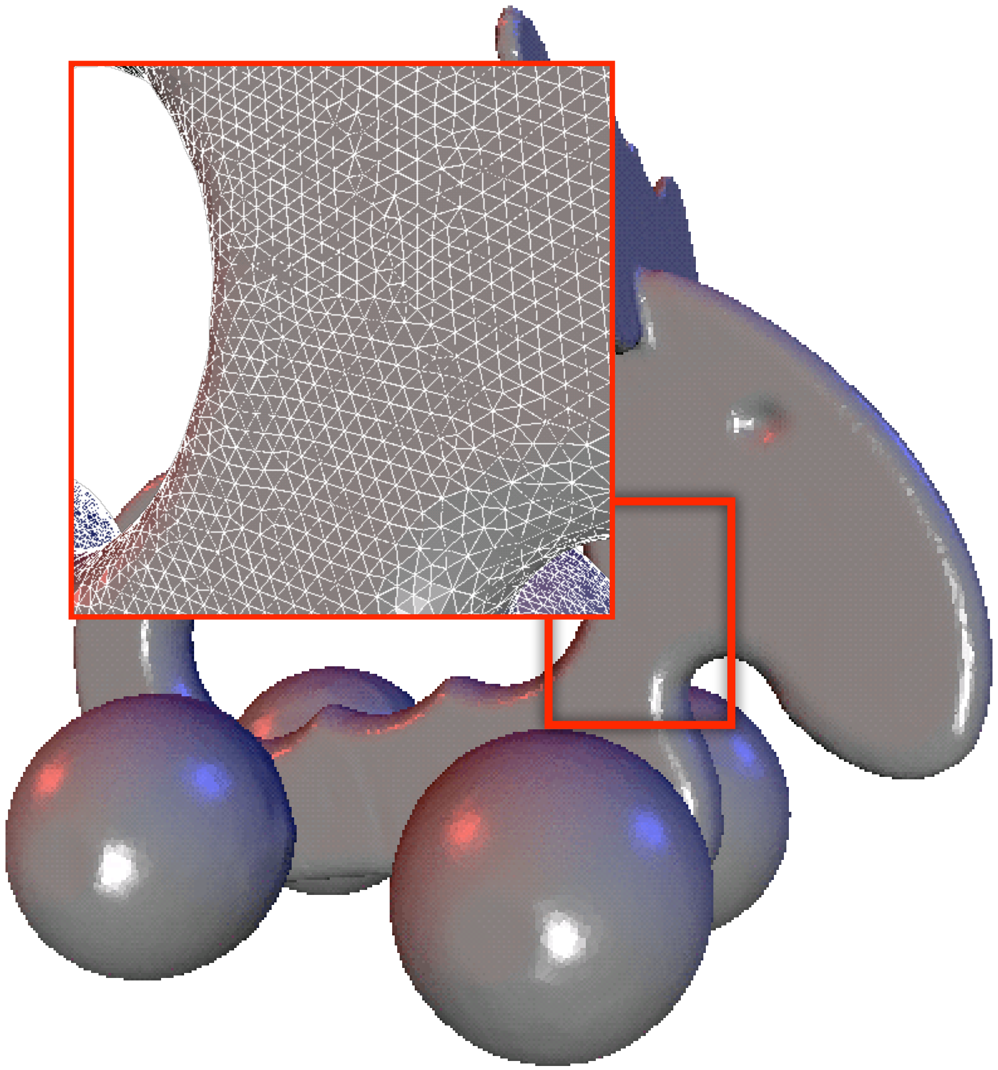
Outline

- Applications
- Requirements
- Mesh Decimation Methods
  - Error Control
  - Fairness Criteria
- Summary

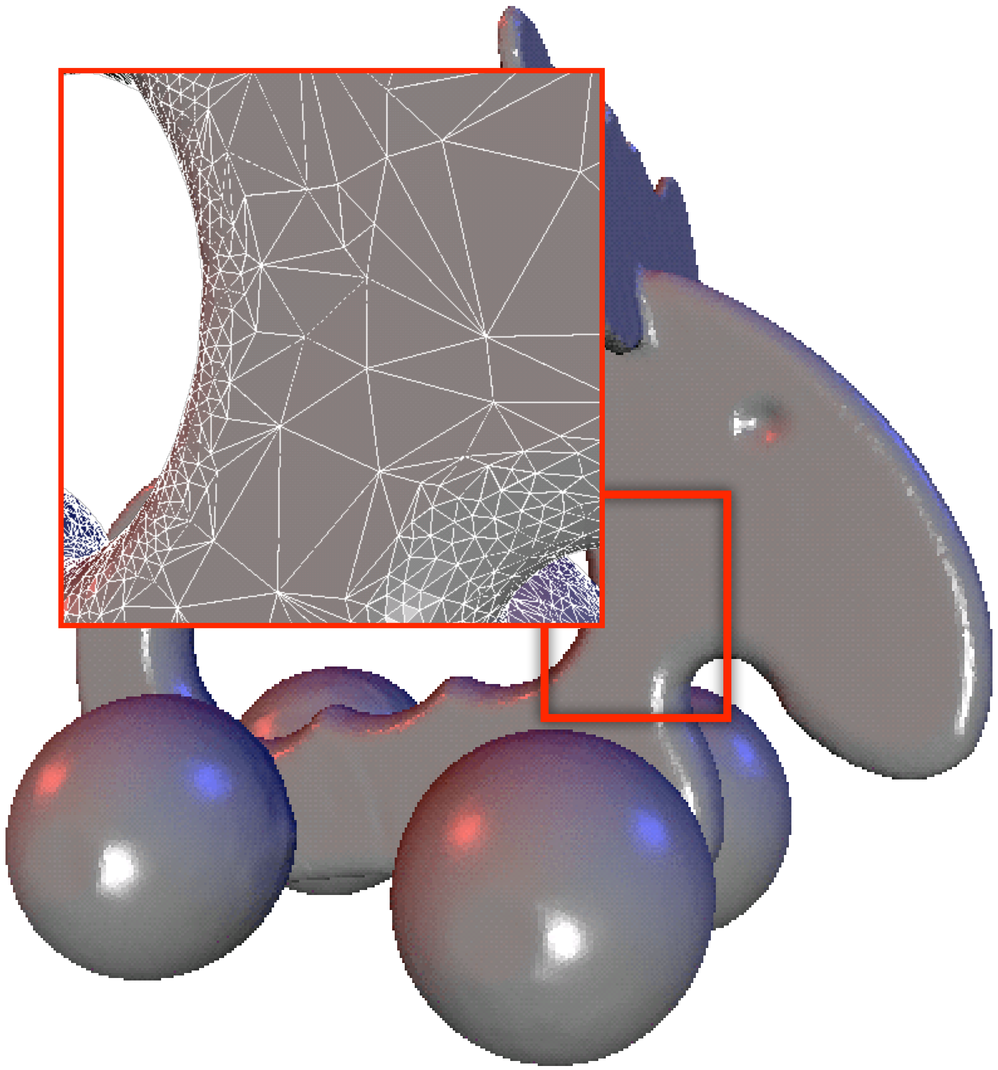
(Some slides taken from Kobbelt et al, Eurographics 2000 Course Notes)

Applications

- Oversampled 3D scan data



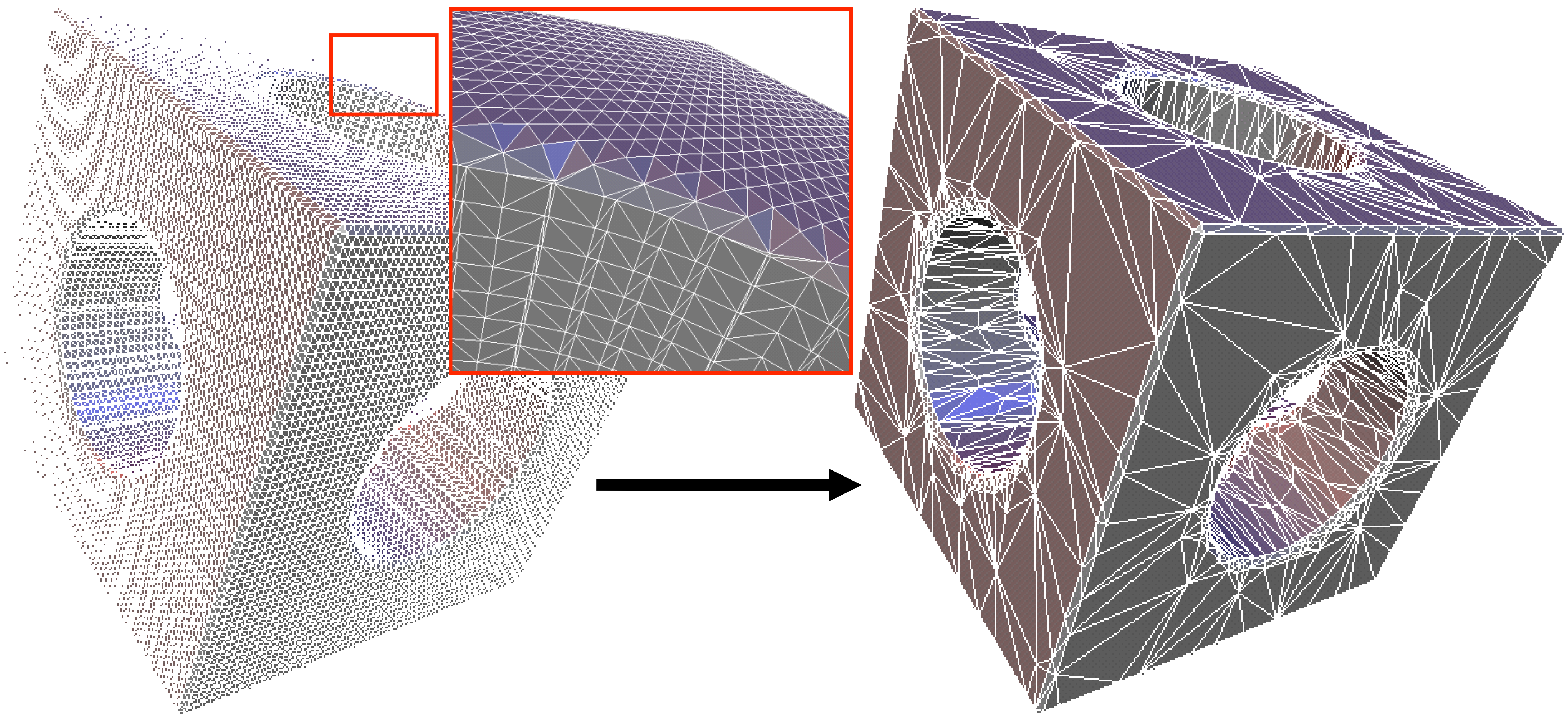
~150k triangles



~80k triangles

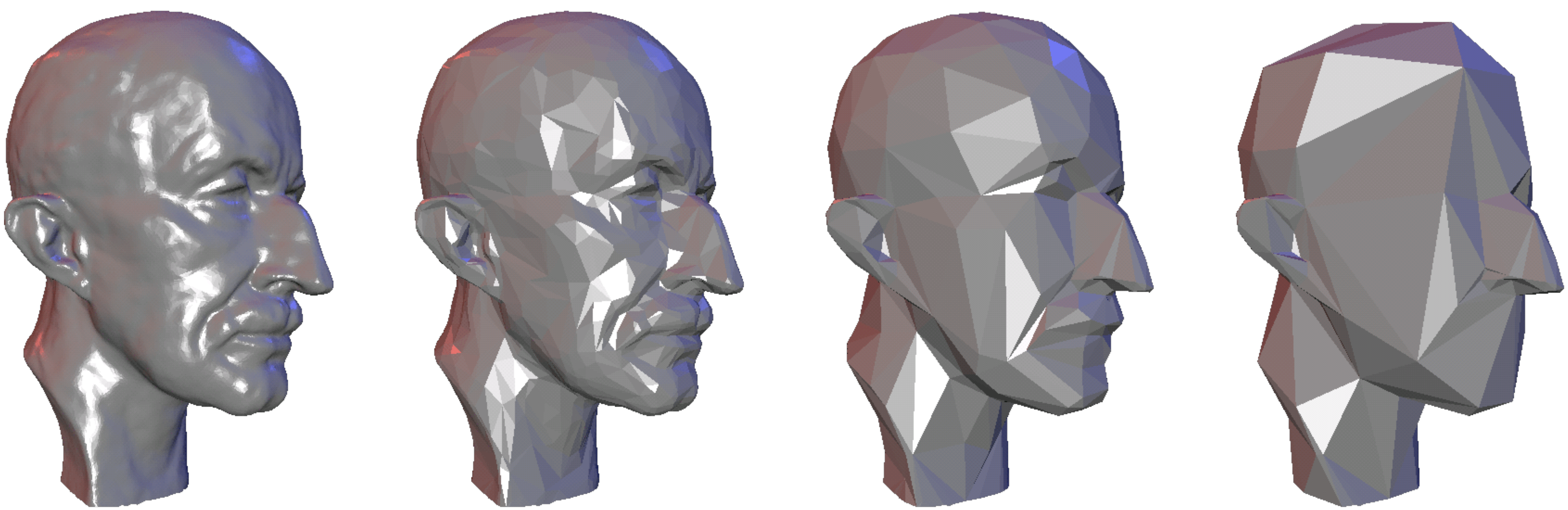
Applications

- Overtessellation: E.g. iso-surface extraction



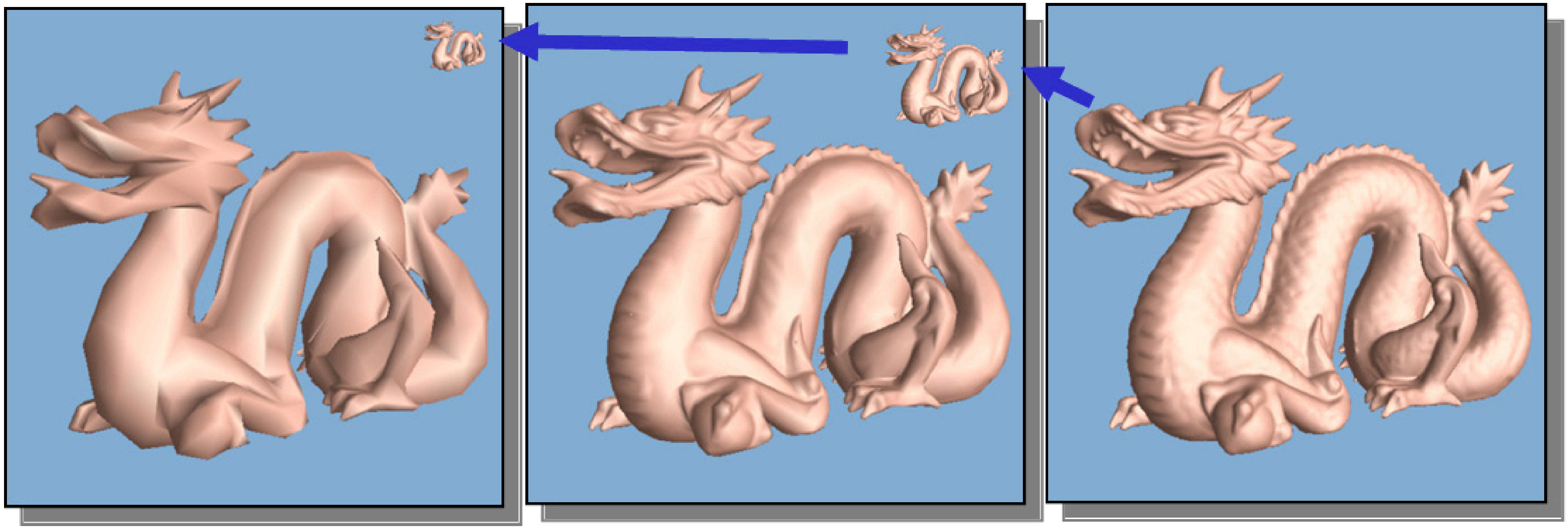
Applications

- Multi-resolution hierarchies for efficient geometry processing



Applications

- Level-of-detail rendering

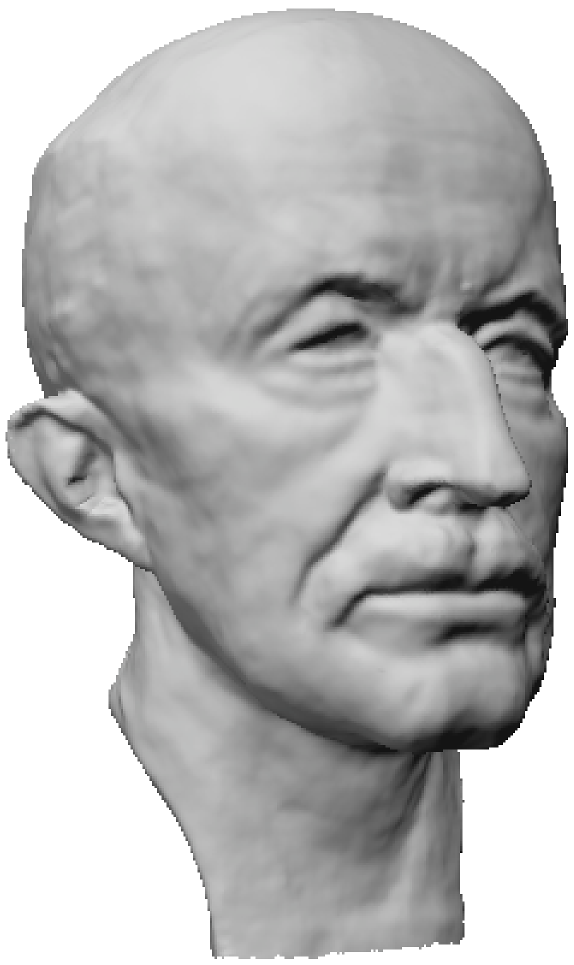


2k triangles 10k triangles 50k triangles

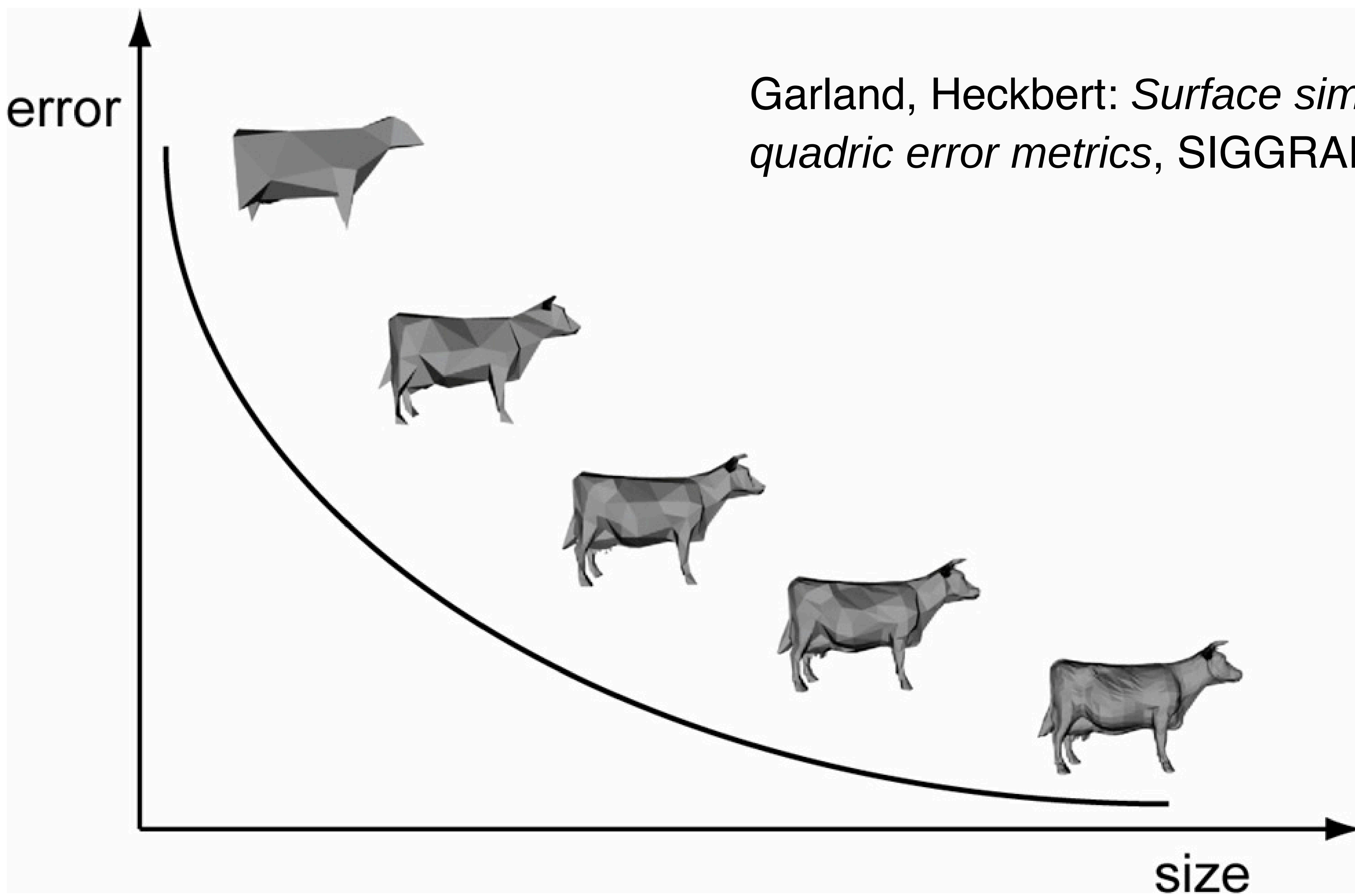
Hoppe: View-dependent refinement of progressive meshes, SIGGRAPH 1997

Applications

- Adaptation to hardware capabilities



Size-Quality Tradeoff



Garland, Heckbert: *Surface simplification using quadric error metrics*, SIGGRAPH 1997

Problem Statement

- Given: 3D model $M = (\{P_i\}, \{T_j\})$
  - Point samples $\{P_i\}$
  - Mesh connectivity $\{T_j\}$

- Find: 3D model $M' = (\{P_i'\}, \{T_j'\})$

$$\#\{P_i'\} << \#\{P_i'\}$$

Requirements

- Global error control

$$\|M - M'\| < \epsilon$$

- Target complextiy

$$\#\{P'_i\} = n$$

- Fairness criteria ...

Overview

	Global error	Target complexity
Vertex clustering	✓	✗
Remeshing	✗	✓
Incremental decimation	✓	✓

Overview

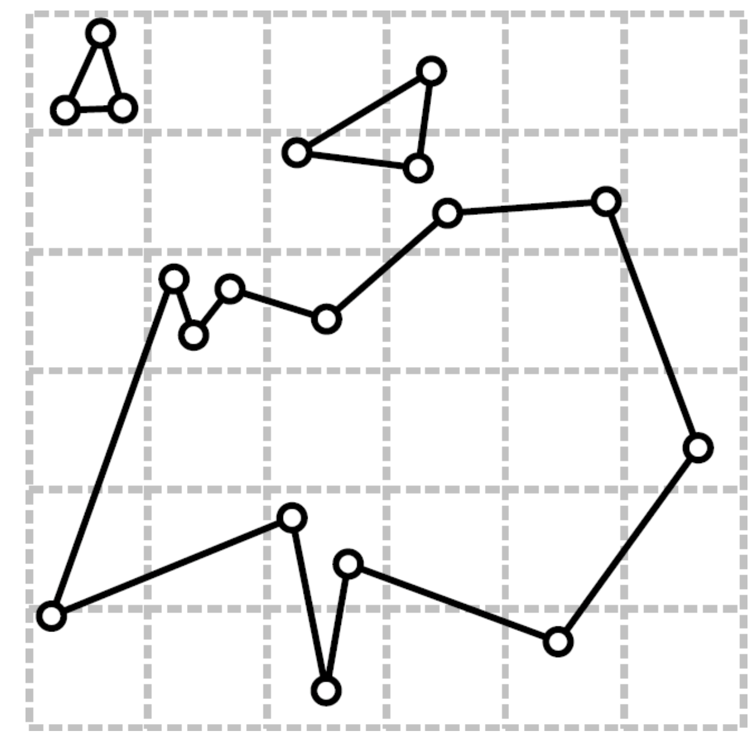
	Global error	Target complexity
Vertex clustering	✓	✗
Remeshing	✗	✓
Incremental decimation	✓	✓

Vertex Clustering

- Cluster Generation
- Computing a representative
- Mesh generation
- Topology changes

Vertex Clustering

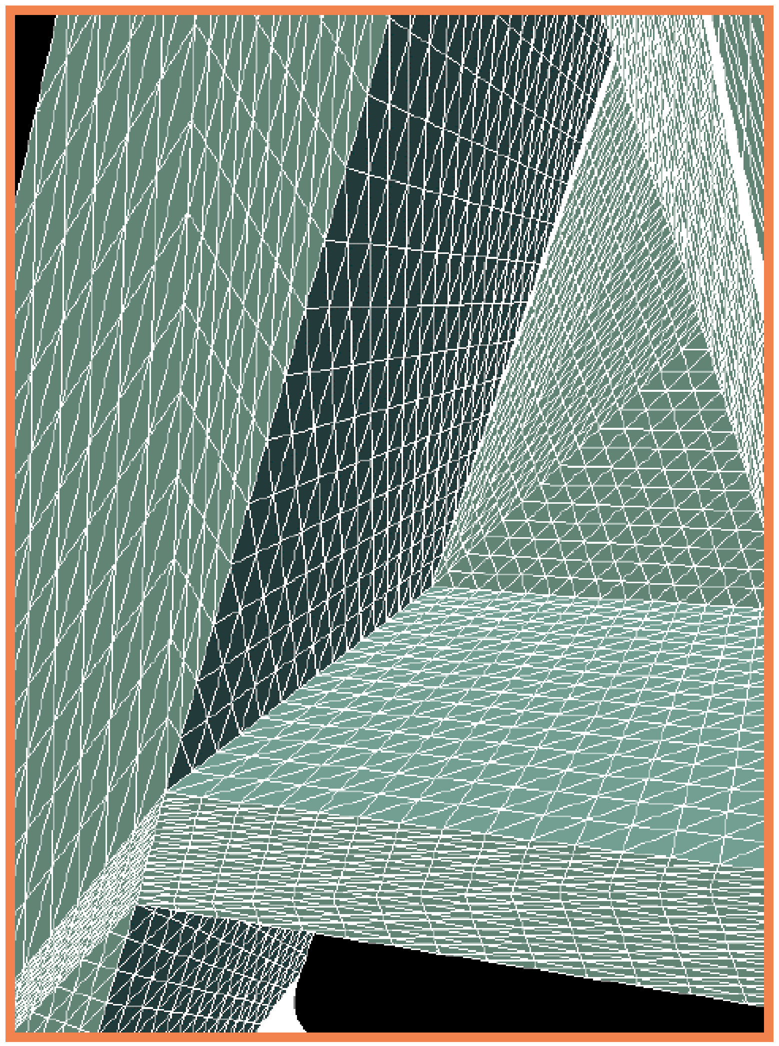
- Cluster Generation
  - Uniform 3D grid
  - Map vertices to cluster cells
- Computing a representative
- Mesh generation
- Topology changes

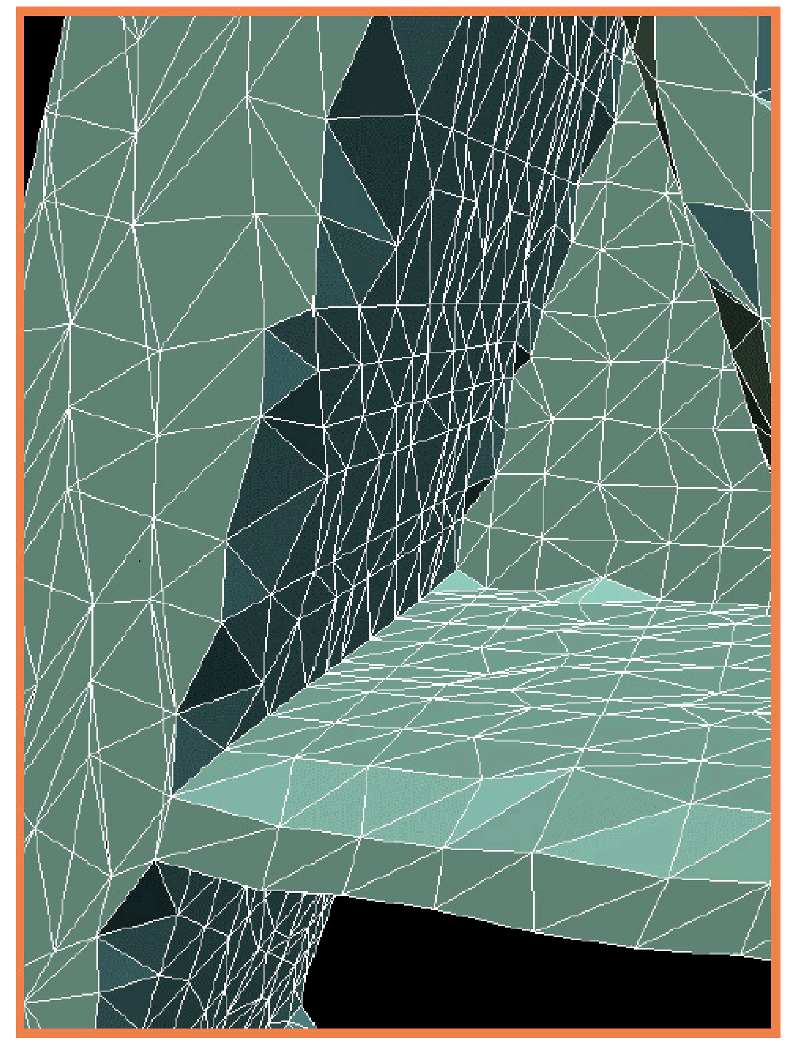


Vertex Clustering

- Cluster Generation
- Computing a representative
  - Average/median vertex position
  - Error quadrics
- Mesh generation
- Topology changes

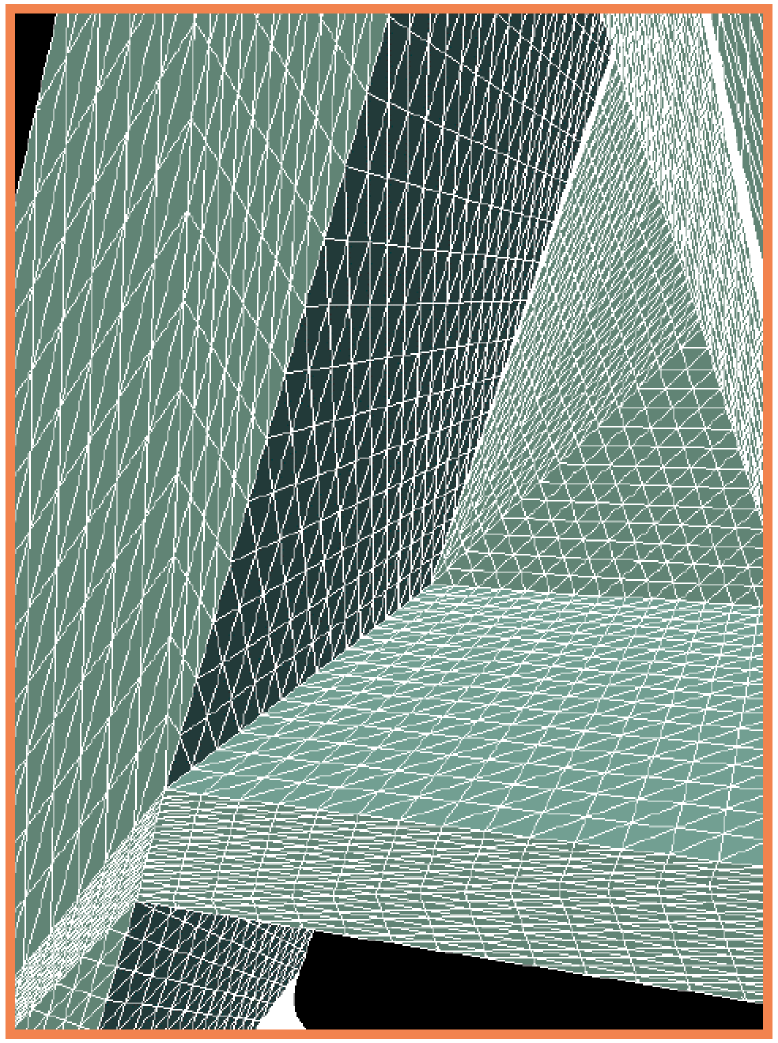
Computing a Representative

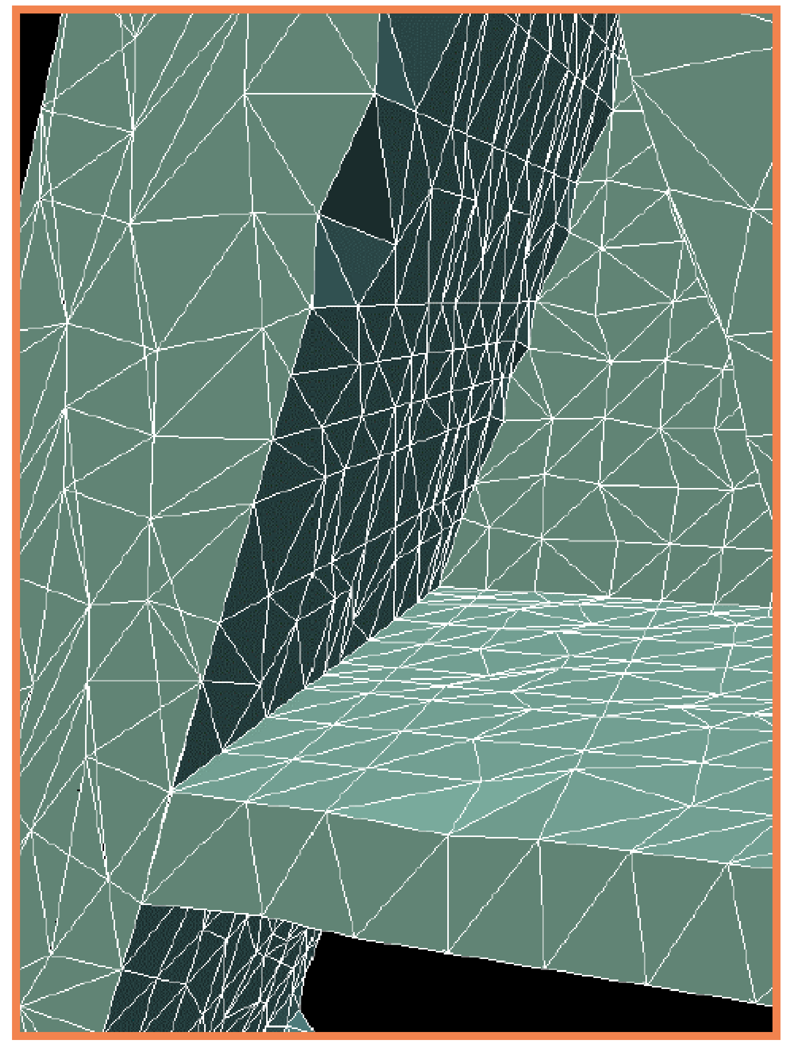




Average vertex position → Low-pass filter

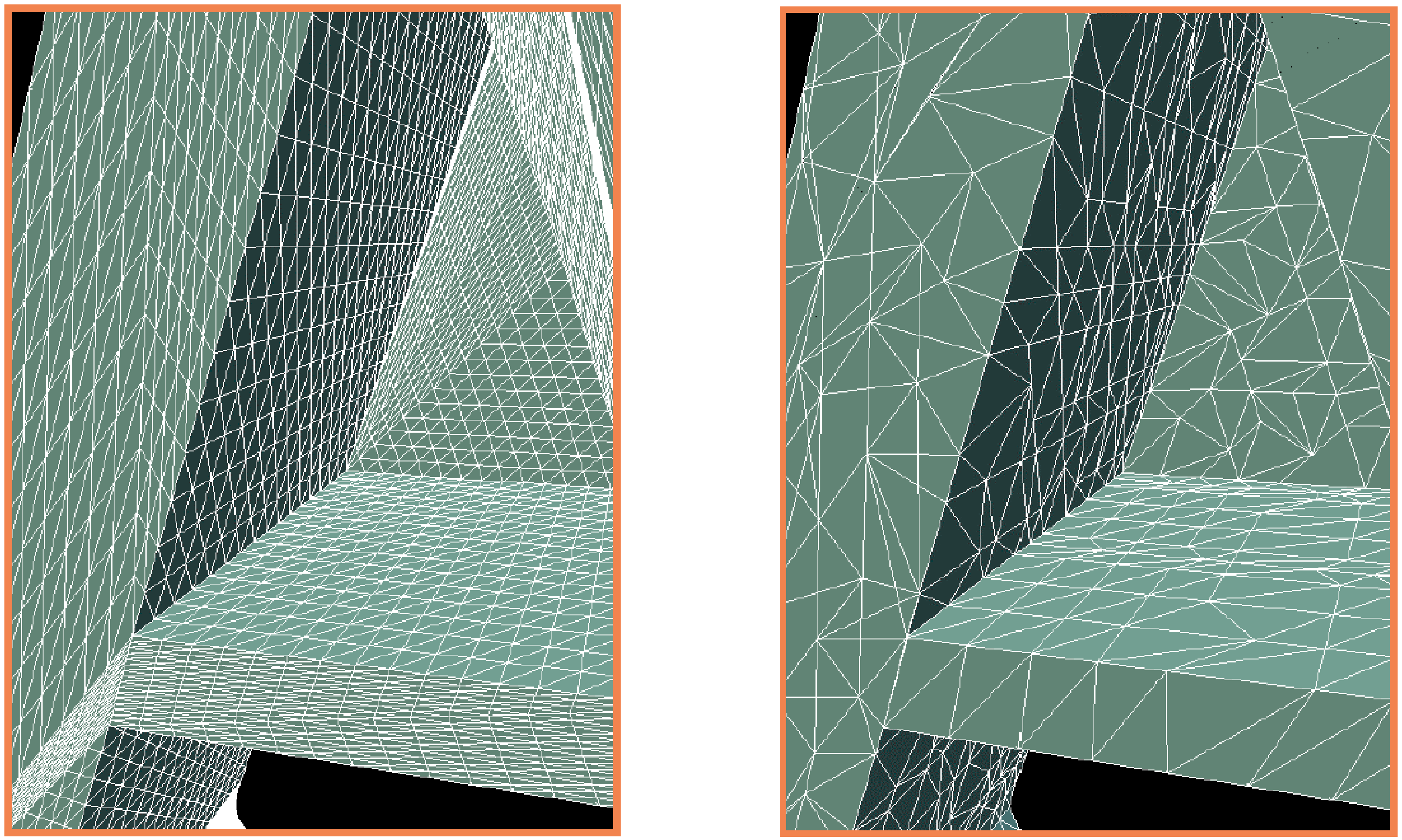
Computing a Representative





Median vertex position → Sub-sampling

Computing a Representative



Error quadrics

Error Quadrics

- Squared distance to plane

$$p = (x, y, z, 1)^T, \quad q = (a, b, c, d)^T$$

$$dist(q, p)^2 = (q^T p)^2 = p^T (q q^T) p =: p^T Q_q p$$

$$Q_q = \begin{bmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & b^2 & cd \\ ad & bd & cd & d^2 \end{bmatrix}$$

Error Quadrics

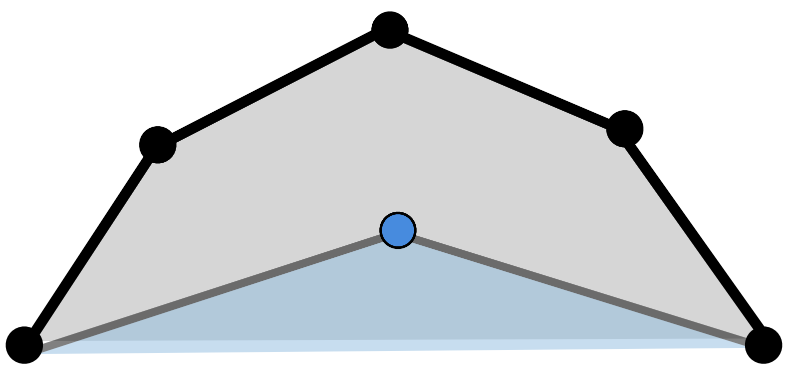
- Sum distances to vertex' planes

$$\sum_i dist(q_i, p)^2 = \sum_i p^T Q_{q_i} p = p^T \left(\sum_i Q_{q_i} \right) p =: p^T Q_p p$$

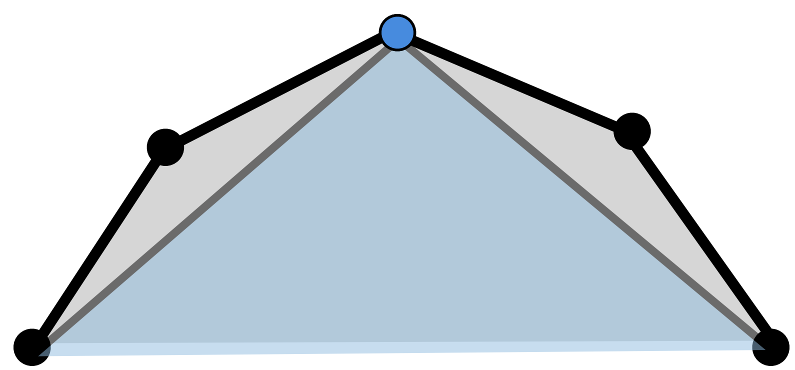
- Point that minimizes the error

$$\begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ q_{31} & q_{32} & q_{33} & q_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} p^* = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

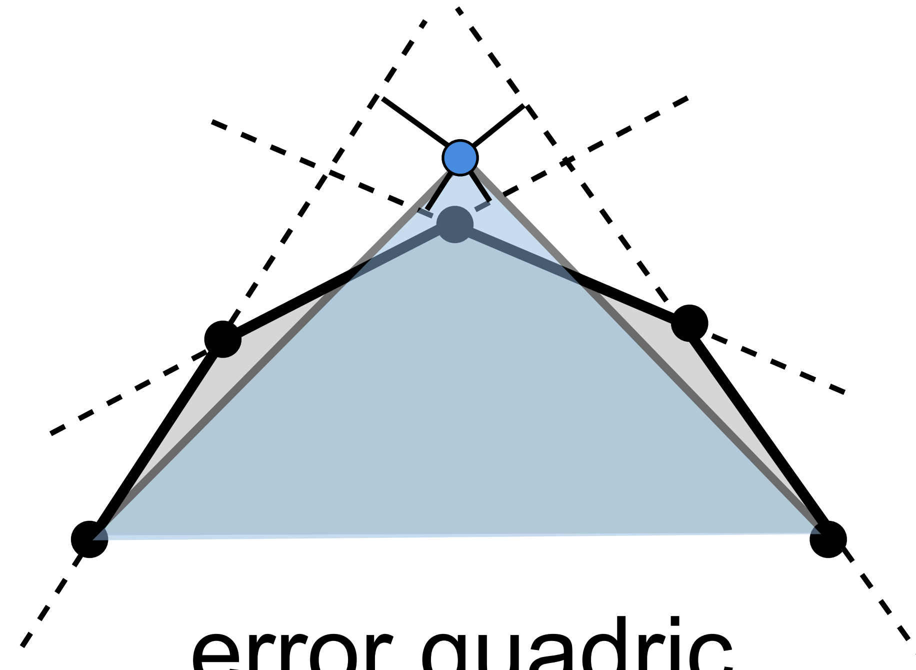
Comparison



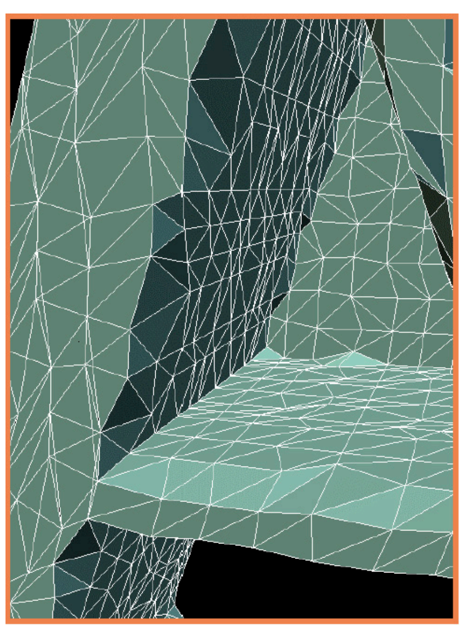
average

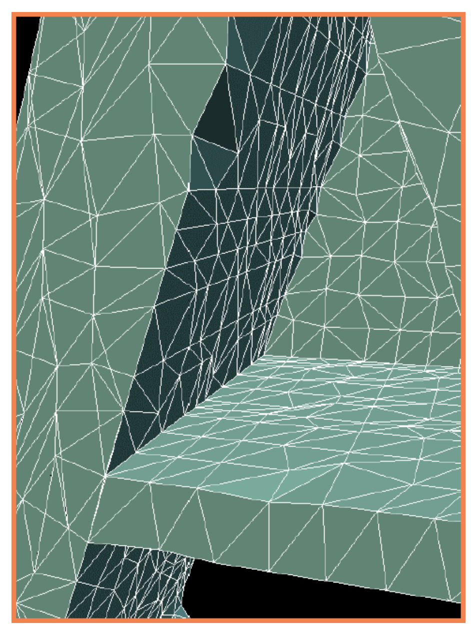


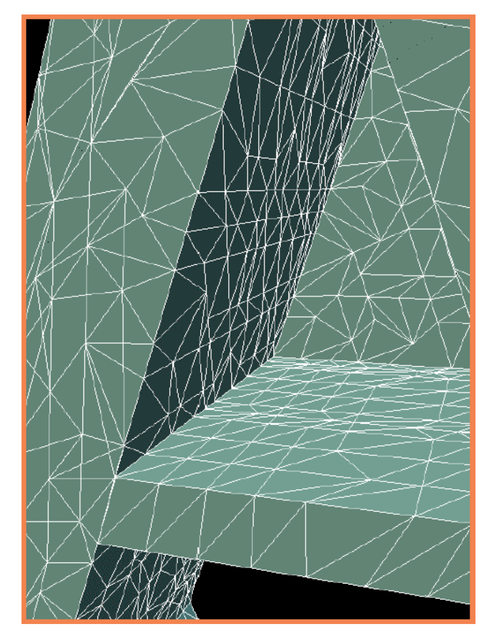
median



error quadric





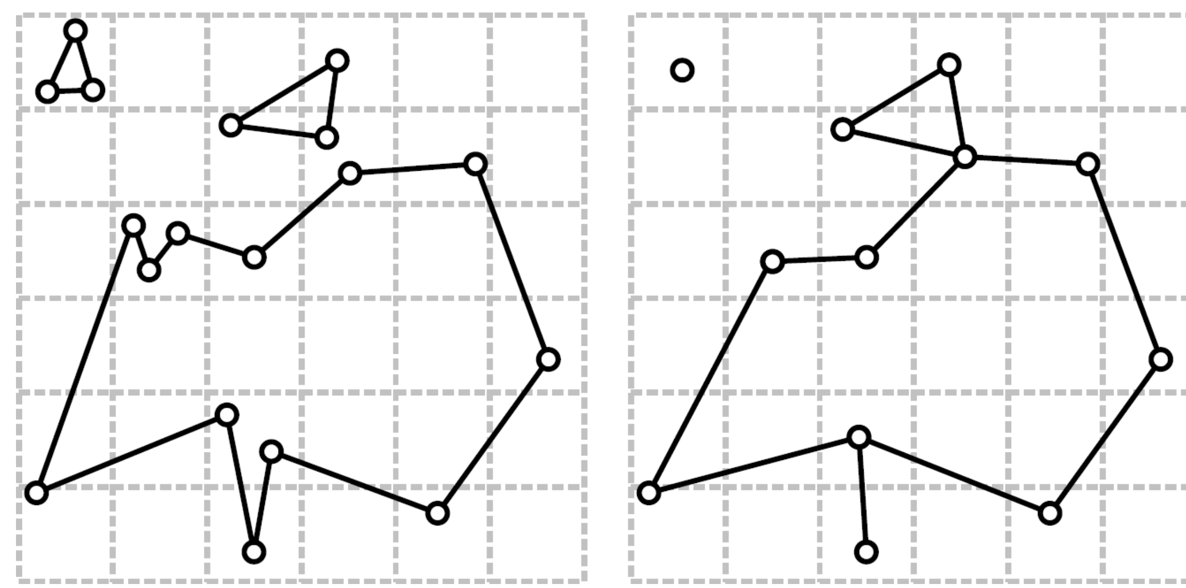


Vertex Clustering

- Cluster Generation
- Computing a representative
- Mesh generation
  - Clusters $p \Leftrightarrow \{p_0,\ldots,p_n\}$, $q \Leftrightarrow \{q_0,\ldots,q_m\}$
  - Connect (p,q) if there was an edge (p_i,q_j)
- Topology changes

Vertex Clustering

- Cluster Generation
- Computing a representative
- Mesh generation
- Topology changes
  - If different sheets pass through one cell
  - Not manifold



Overview

	Global error	Target complexity
Vertex Clustering	✓	✗
Remeshing	✗	✓
Incremental decimation	✓	✓

Overview

	Global error	Target complexity
Vertex Clustering	✓	✗
Remeshing	✗	✓
Incremental decimation	✓	✓

Incremental Decimation

- General Setup
- Decimation operators
- Error metrics
- Fairness criteria
- Topology changes

General Setup

```
Repeat:
  pick mesh region
  apply decimation operator
Until no further reduction possible
```

Greedy Optimization

```
For each region
  evaluate quality after decimation
  enqeue(quality, region)

Repeat:
  pick best mesh region
  apply decimation operator
  update queue
Until no further reduction possible
```

Global Error Control

```
For each region
  evaluate quality after decimation
  enqeue(quality, region)

Repeat:
  pick best mesh region
  if error < ε
     apply decimation operator
     update queue
Until no further reduction possible
```

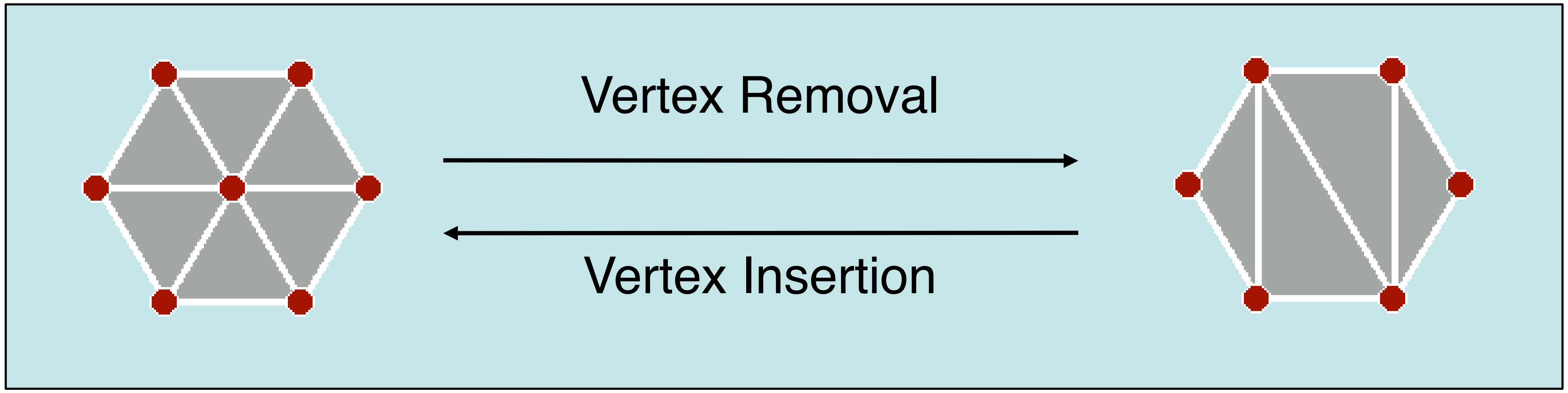
Incremental Decimation

- General Setup
- Decimation operators
- Error metrics
- Fairness criteria
- Topology changes

Decimation Operators

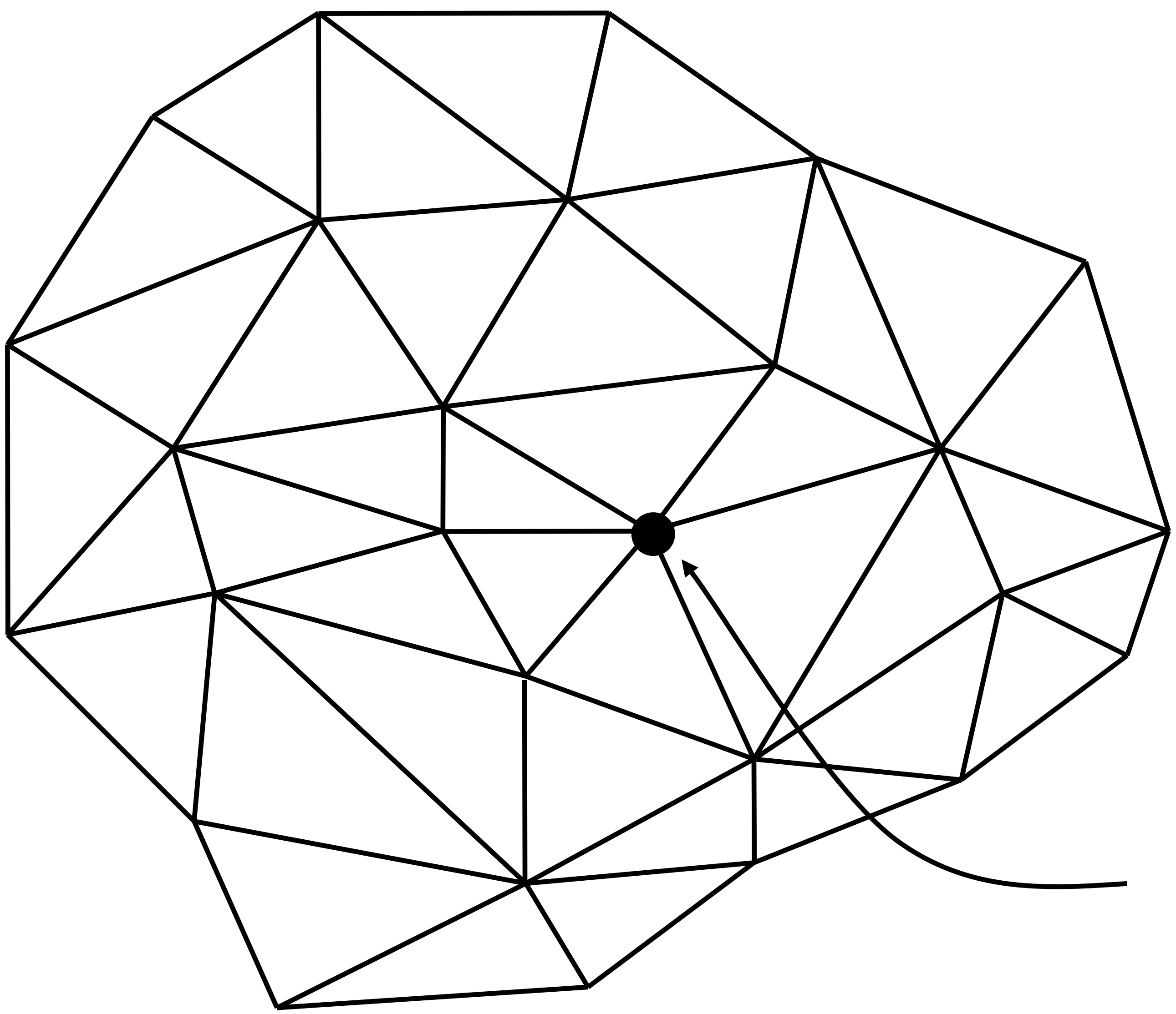
- What is a "region" ?
- What are the DOF for re-triangulation?
- Classification
  - Topology-changing vs. topology-preserving
  - Subsampling vs. filtering
  - Inverse operation → progressive meshes

Decimation Operators



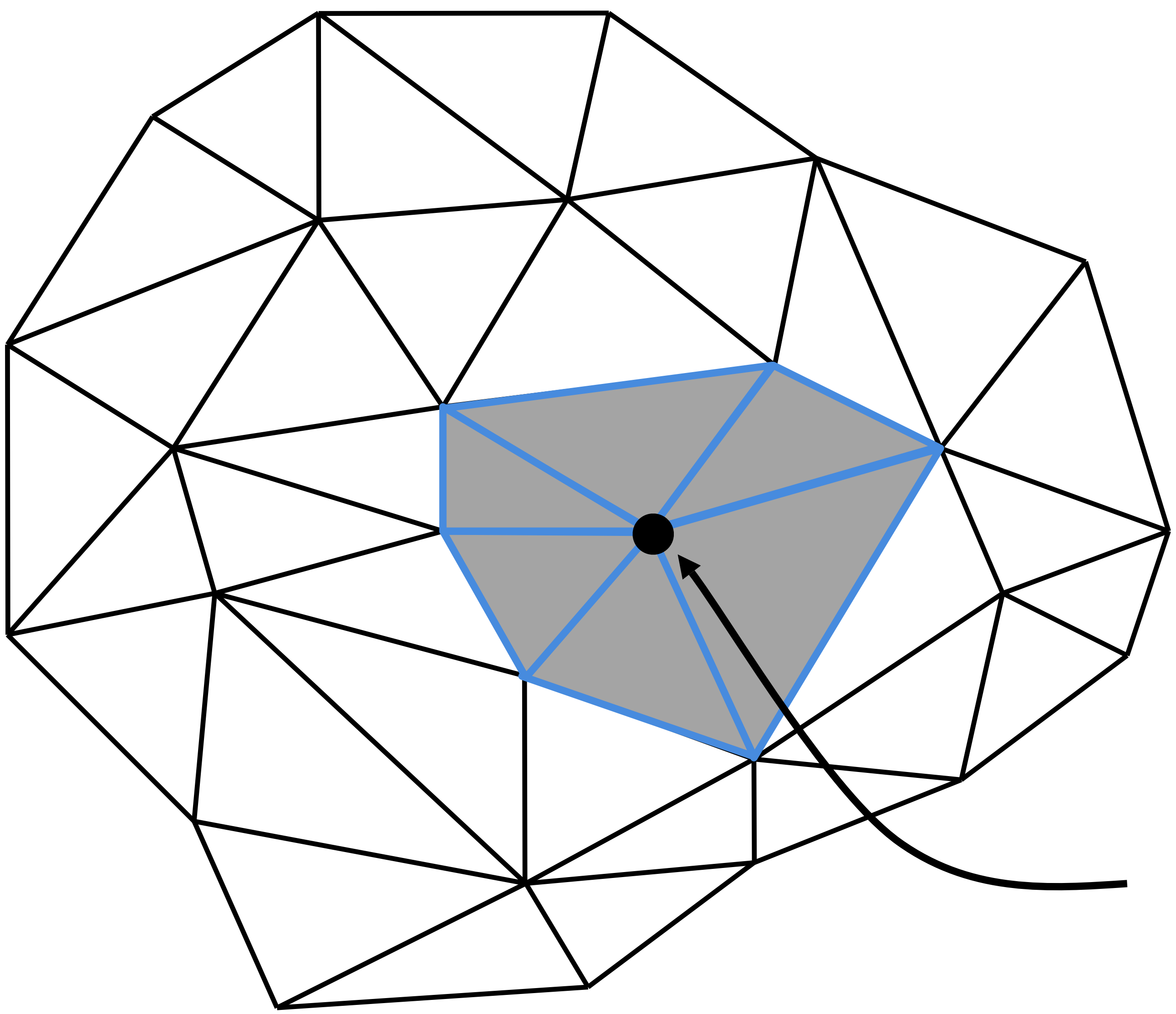
- Remove vertex
- Re-triangulate hole
    - Combinatorial DOFs
    - Sub-sampling !

Vertex Removal



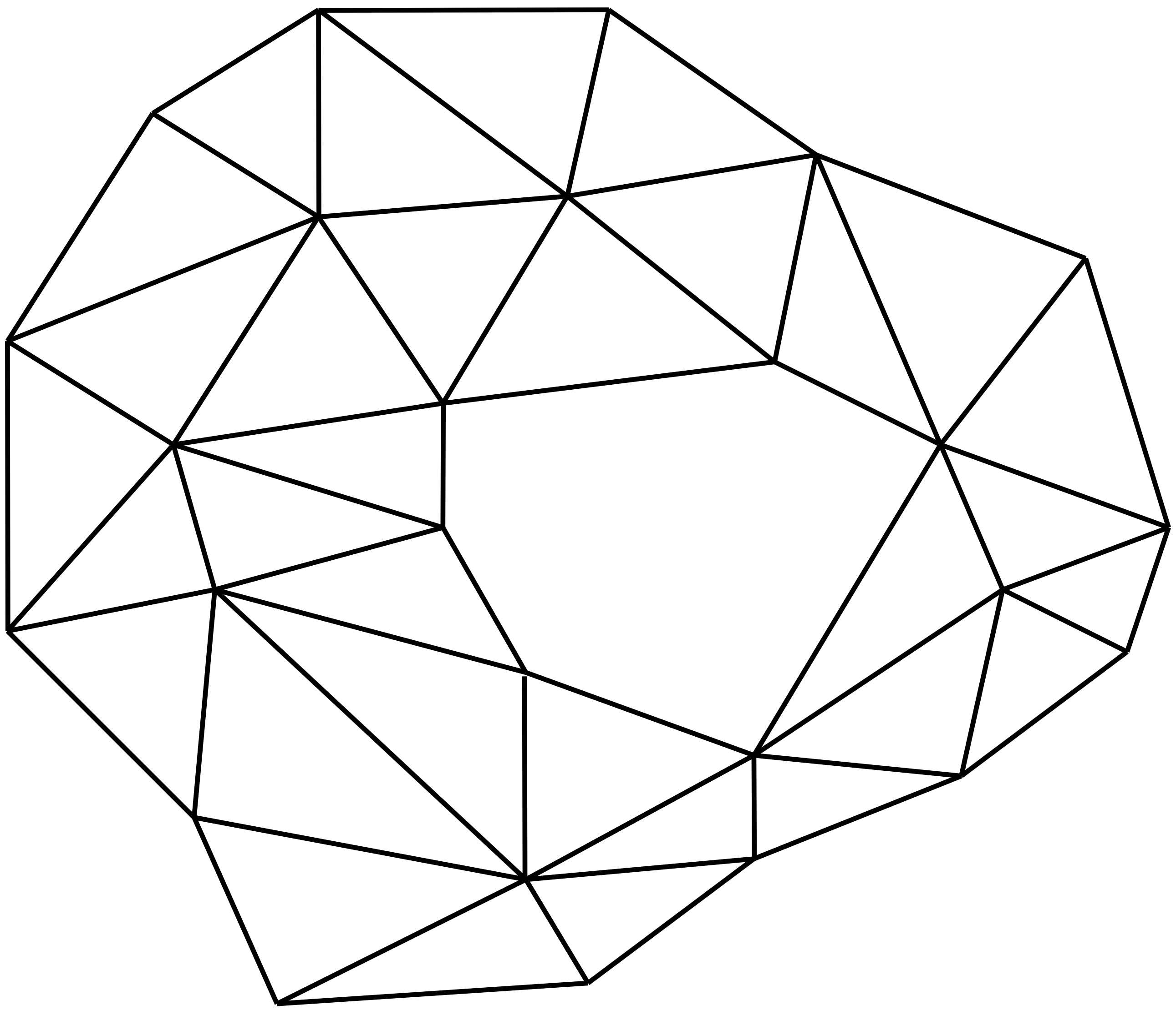
Select an element
to be eliminated

Vertex Removal



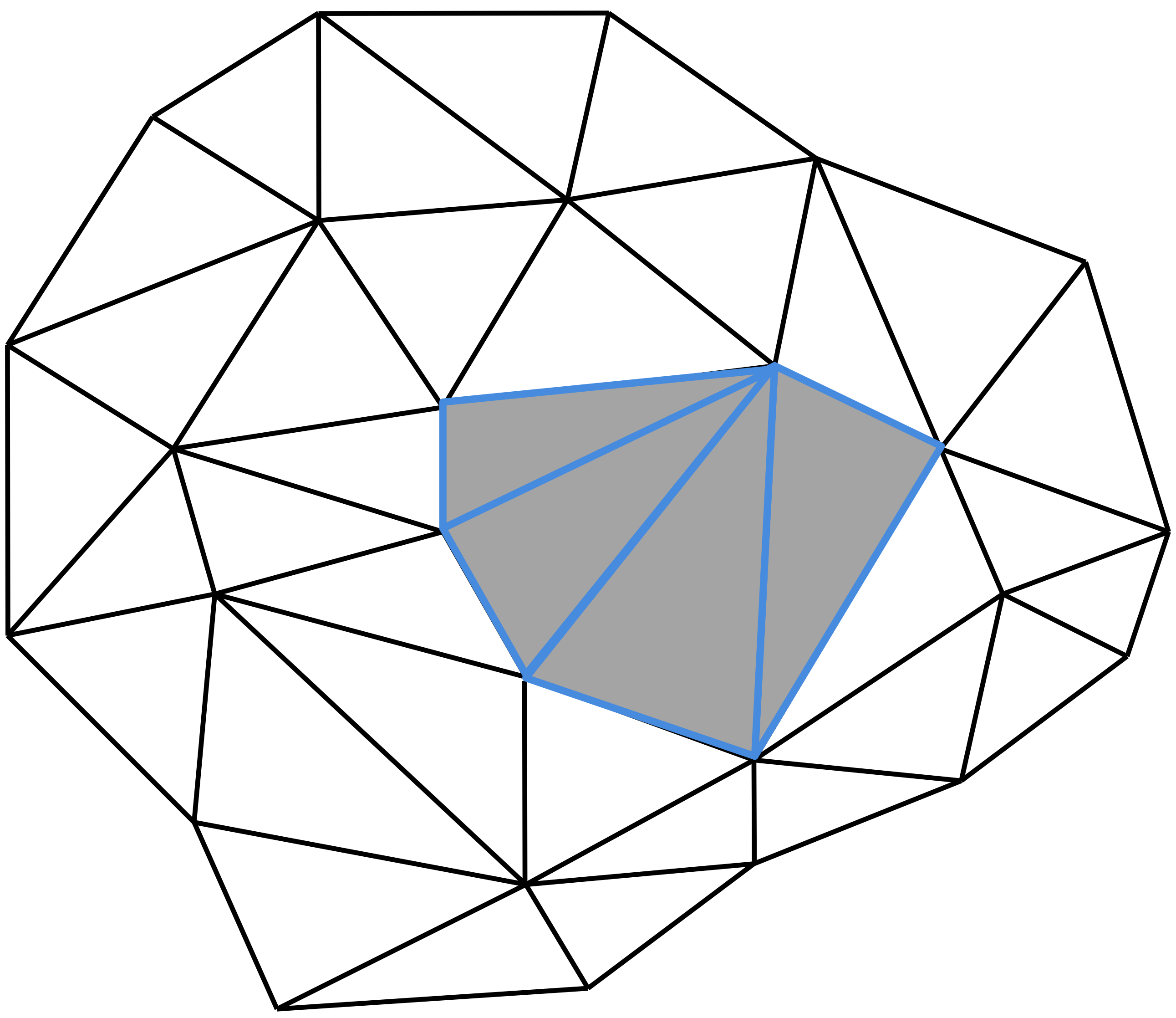
Select all triangles sharing this vertex

Vertex Removal



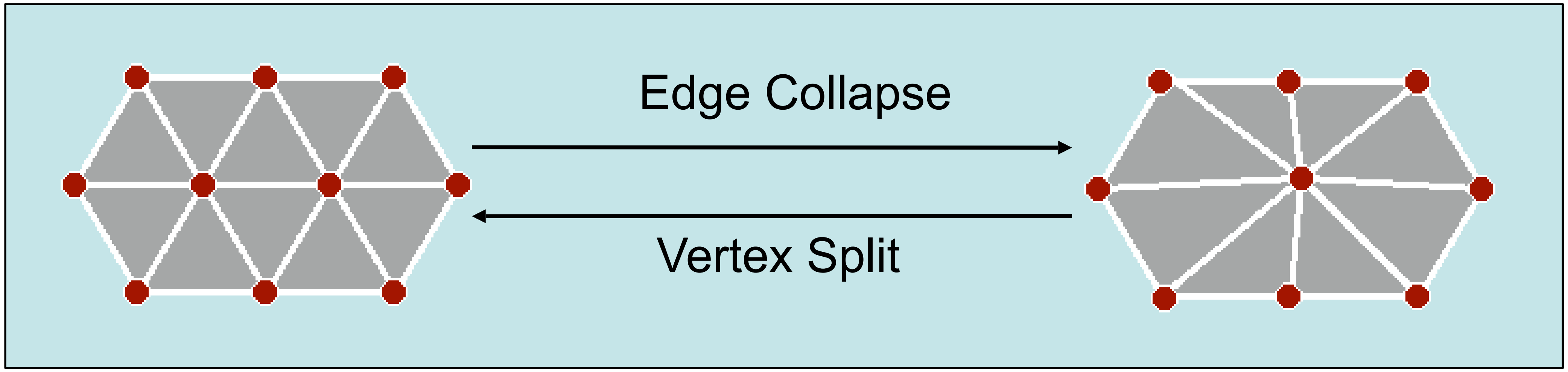
Remove the selected triangles, creating the hole

Vertex Removal



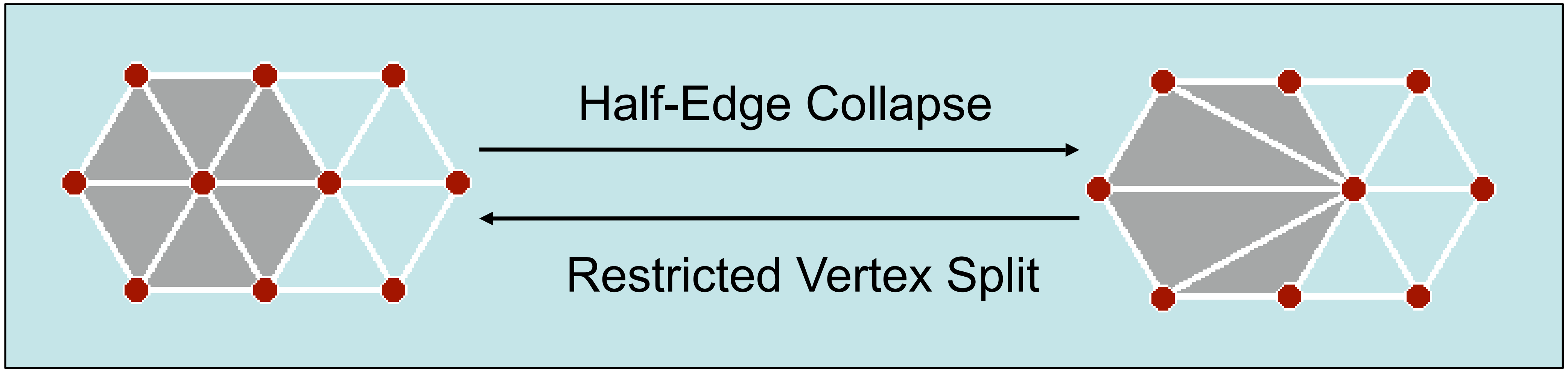
Fill the hole
with triangles

Decimation Operators



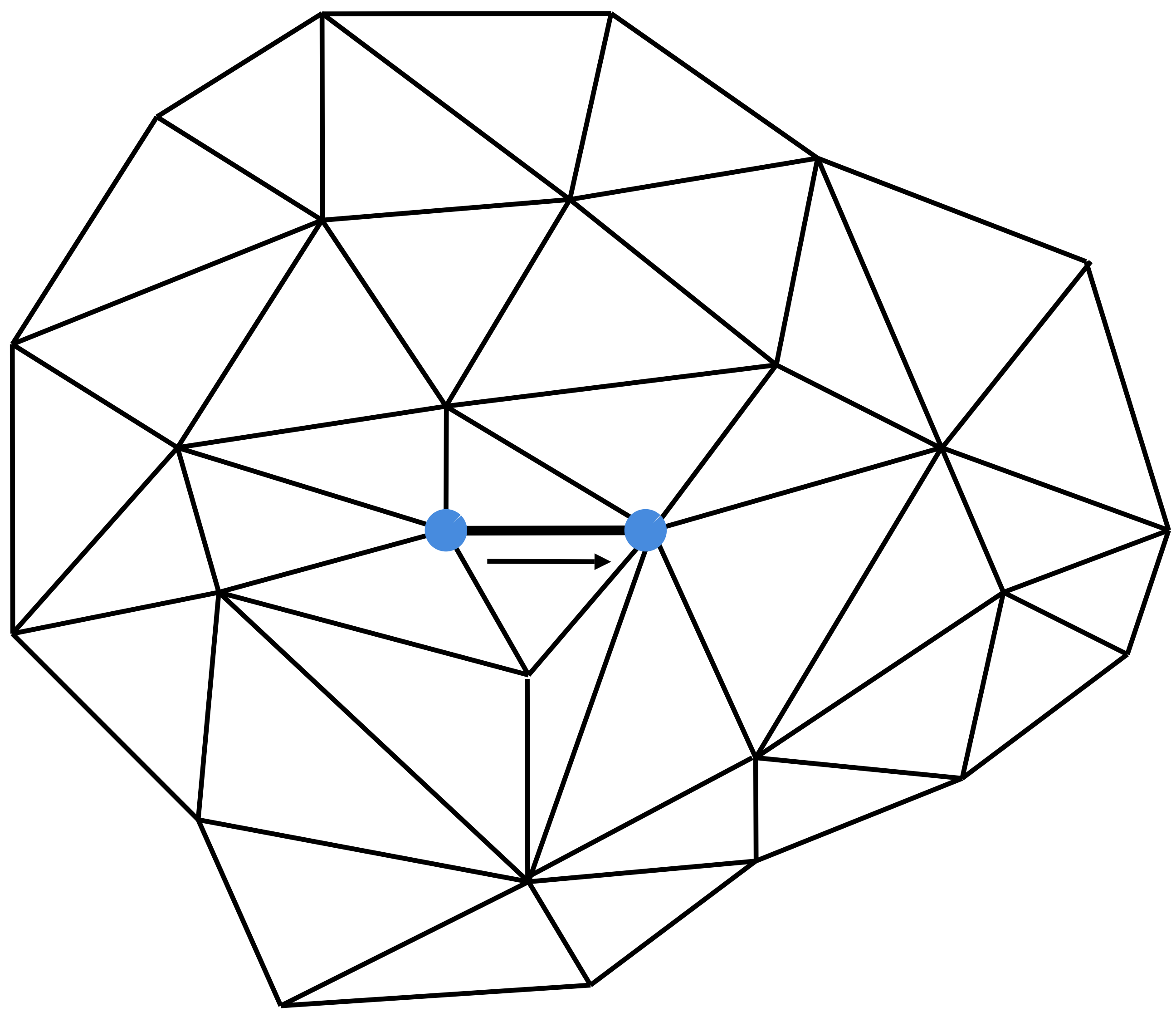
- Merge two adjacent triangles
- Define new vertex position
  - Continous DOF
  - Filtering !

Decimation Operators

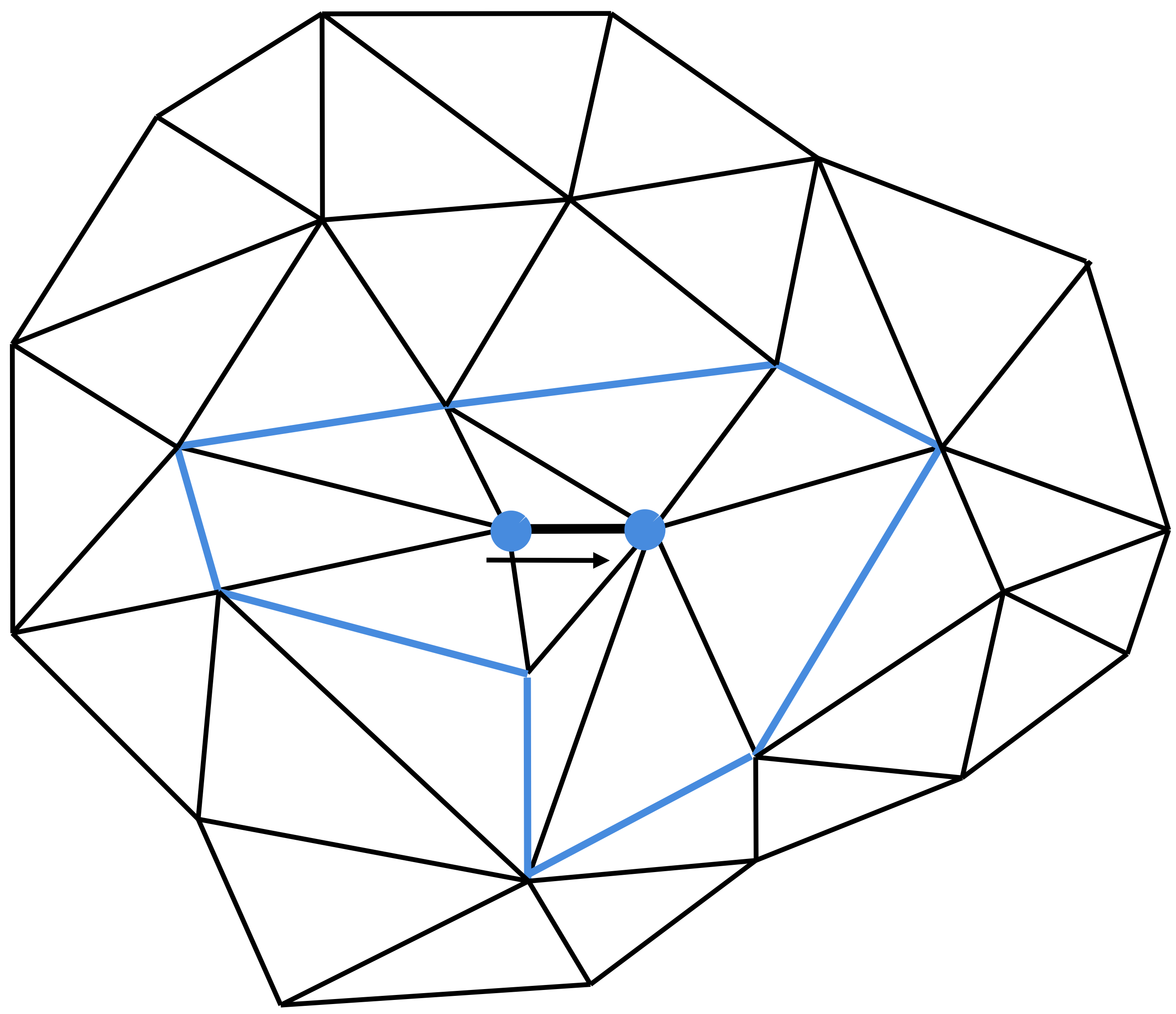


- Collapse edge into one end point
  - Special vertex removal
  - Special edge collapse
- No DOFs
  - One operator per half-edge
  - Sub-sampling !

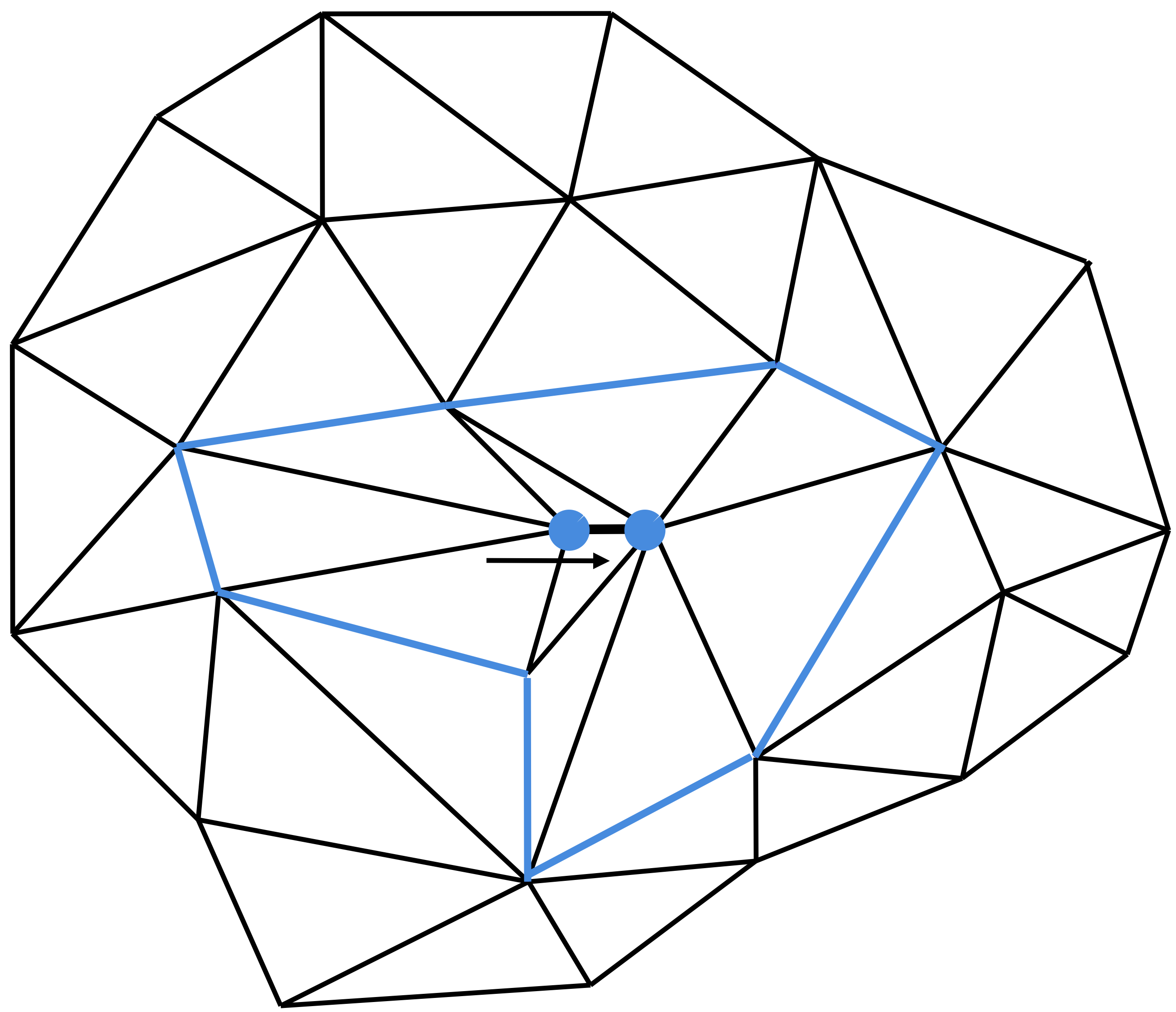
Edge Collapse



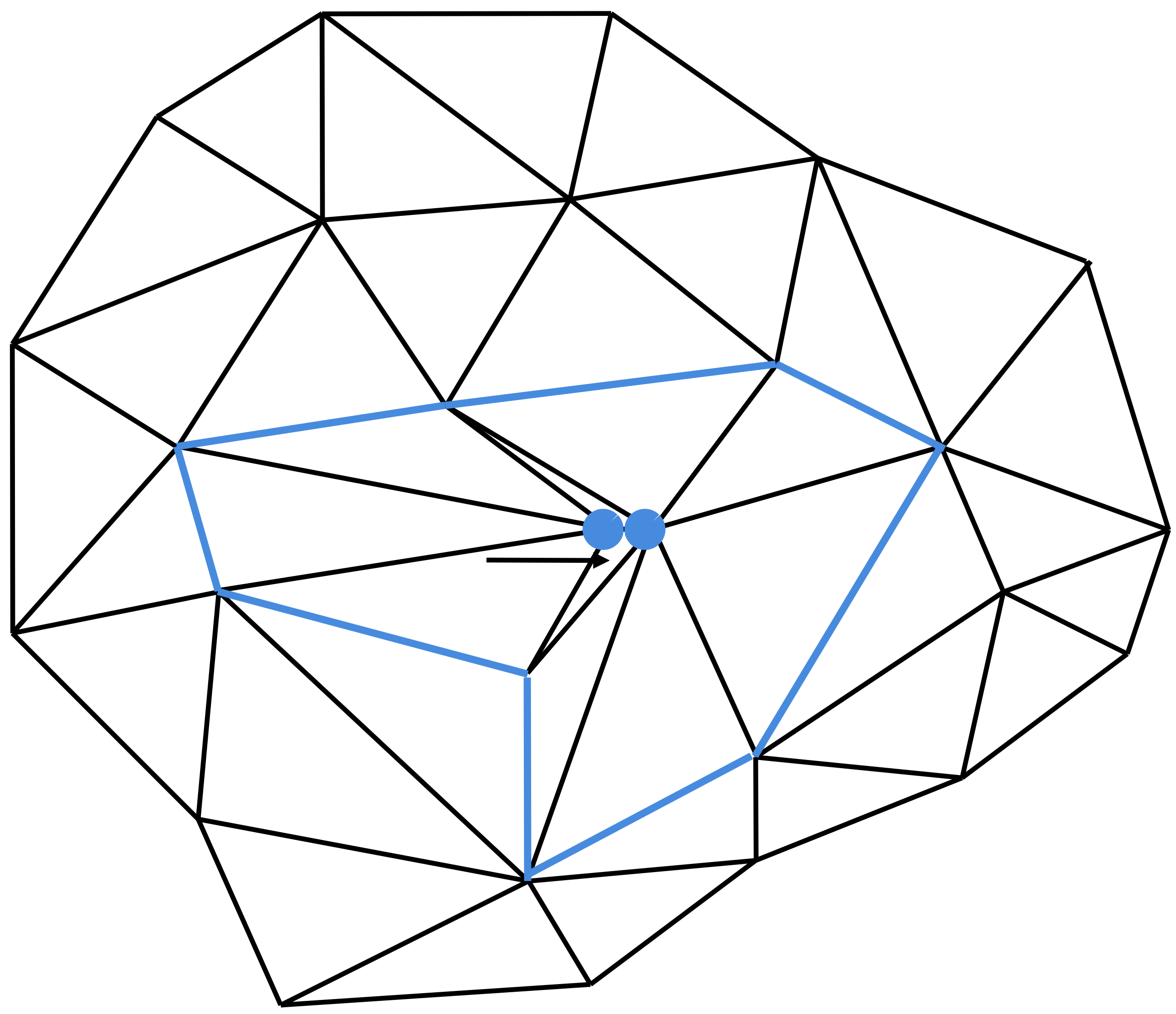
Edge Collapse



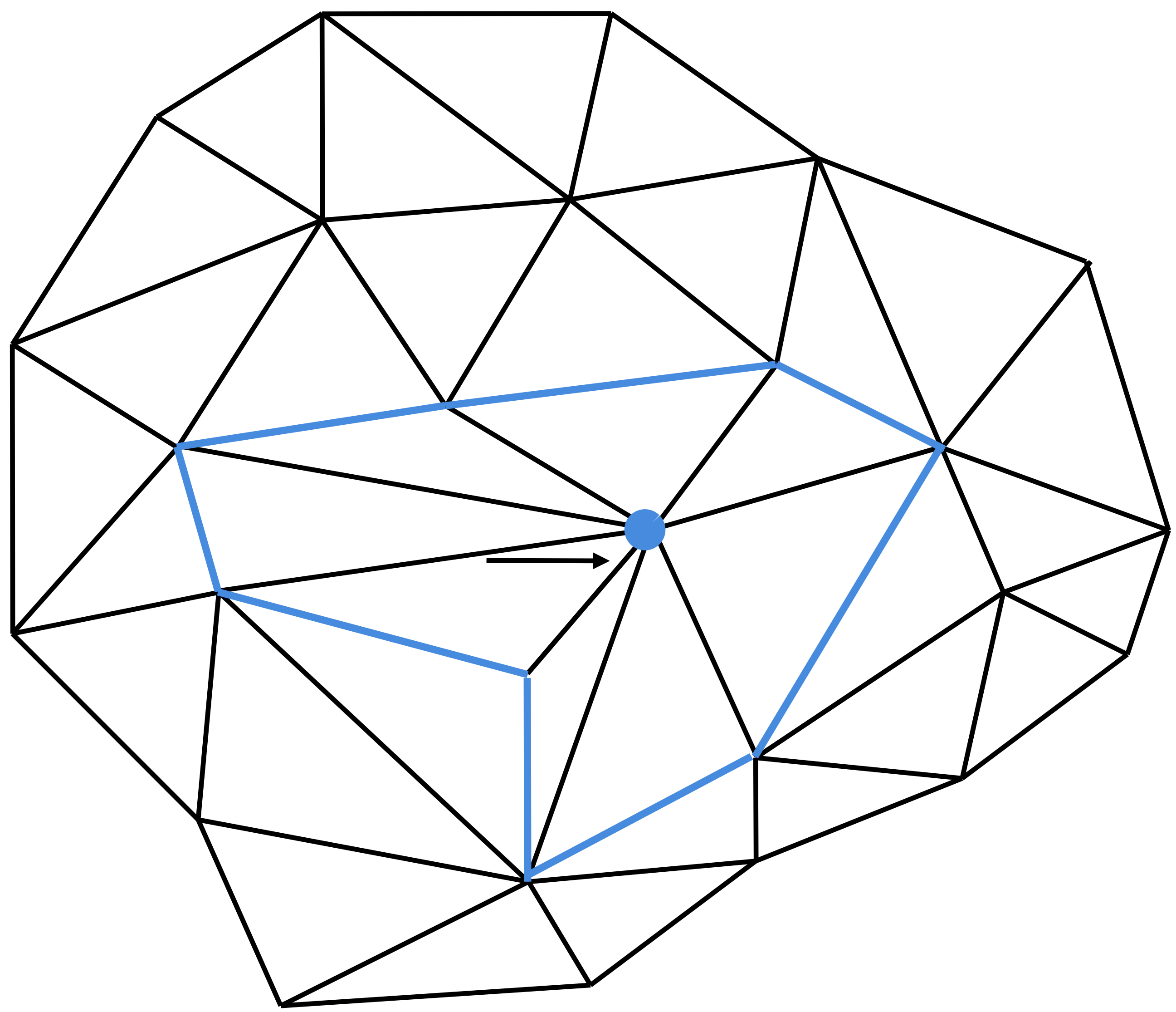
Edge Collapse



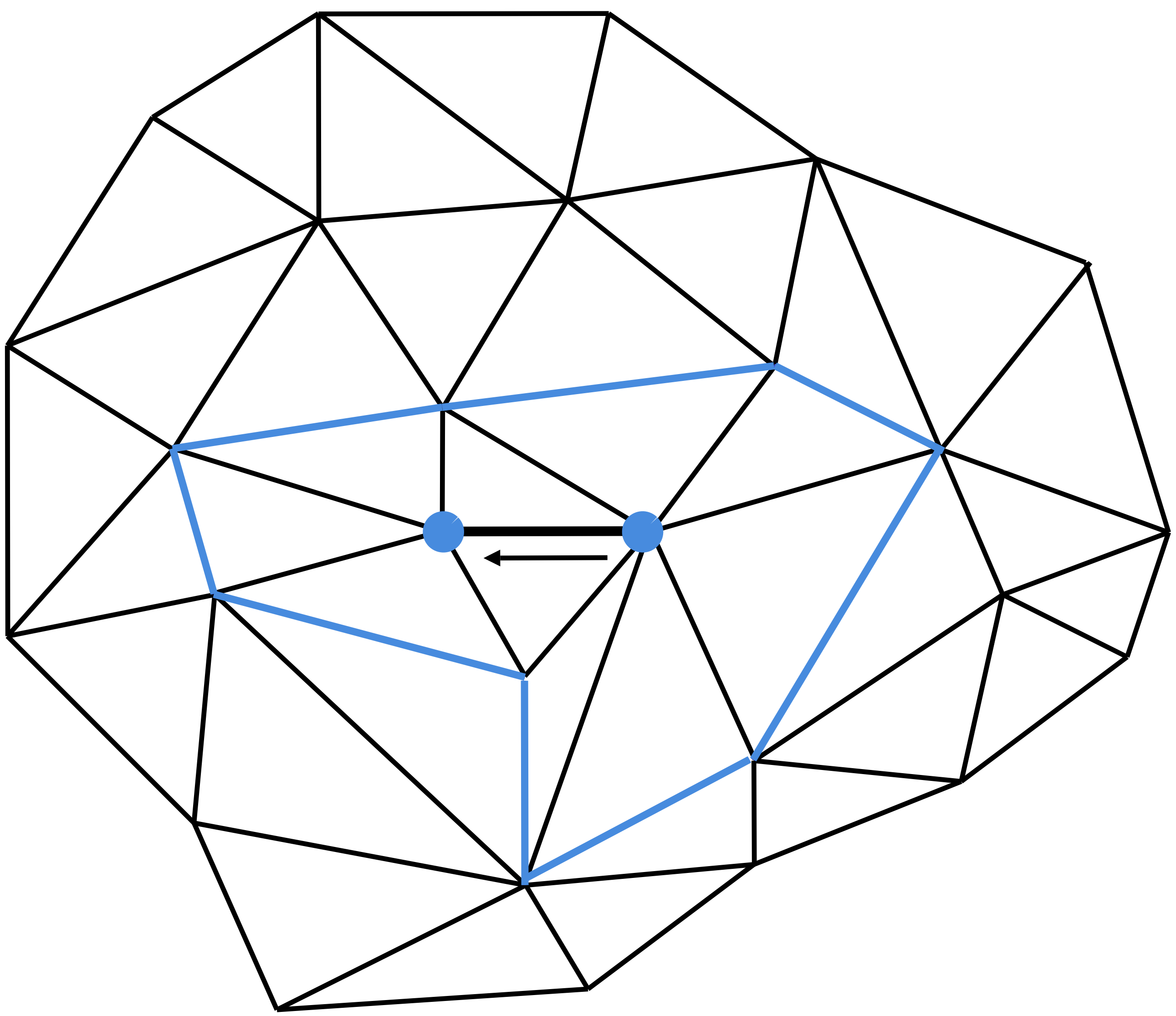
Edge Collapse



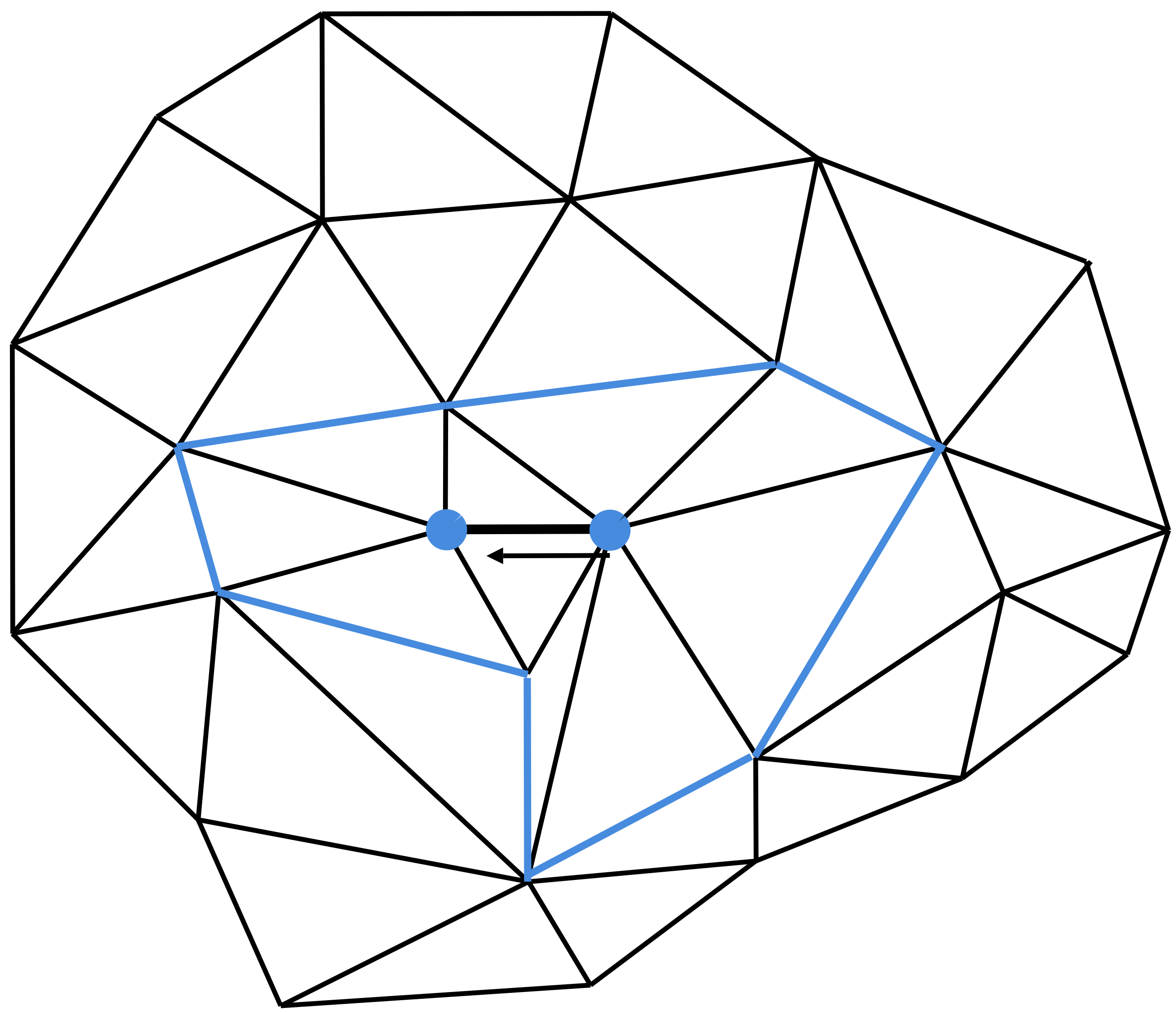
Edge Collapse



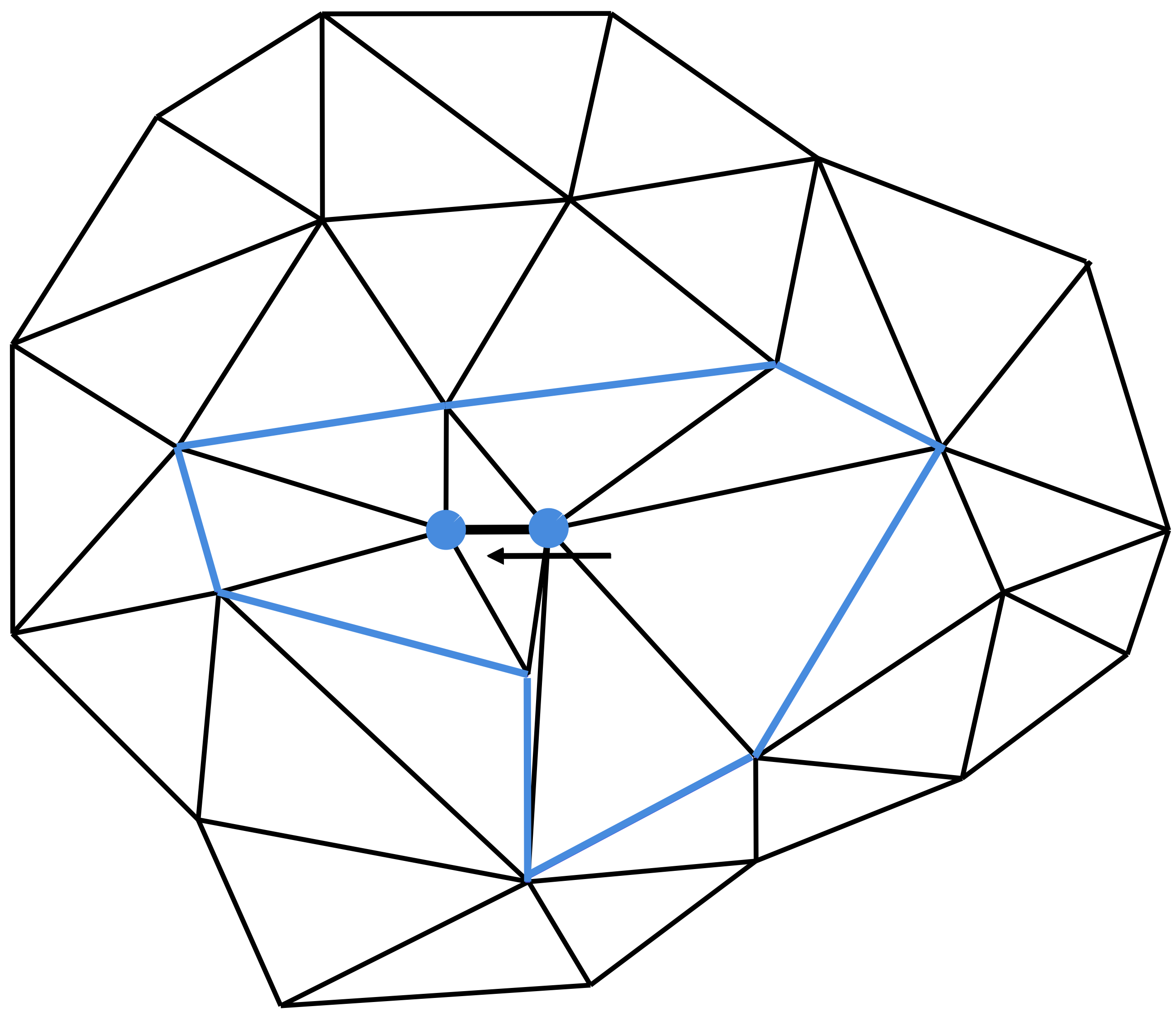
Edge Collapse



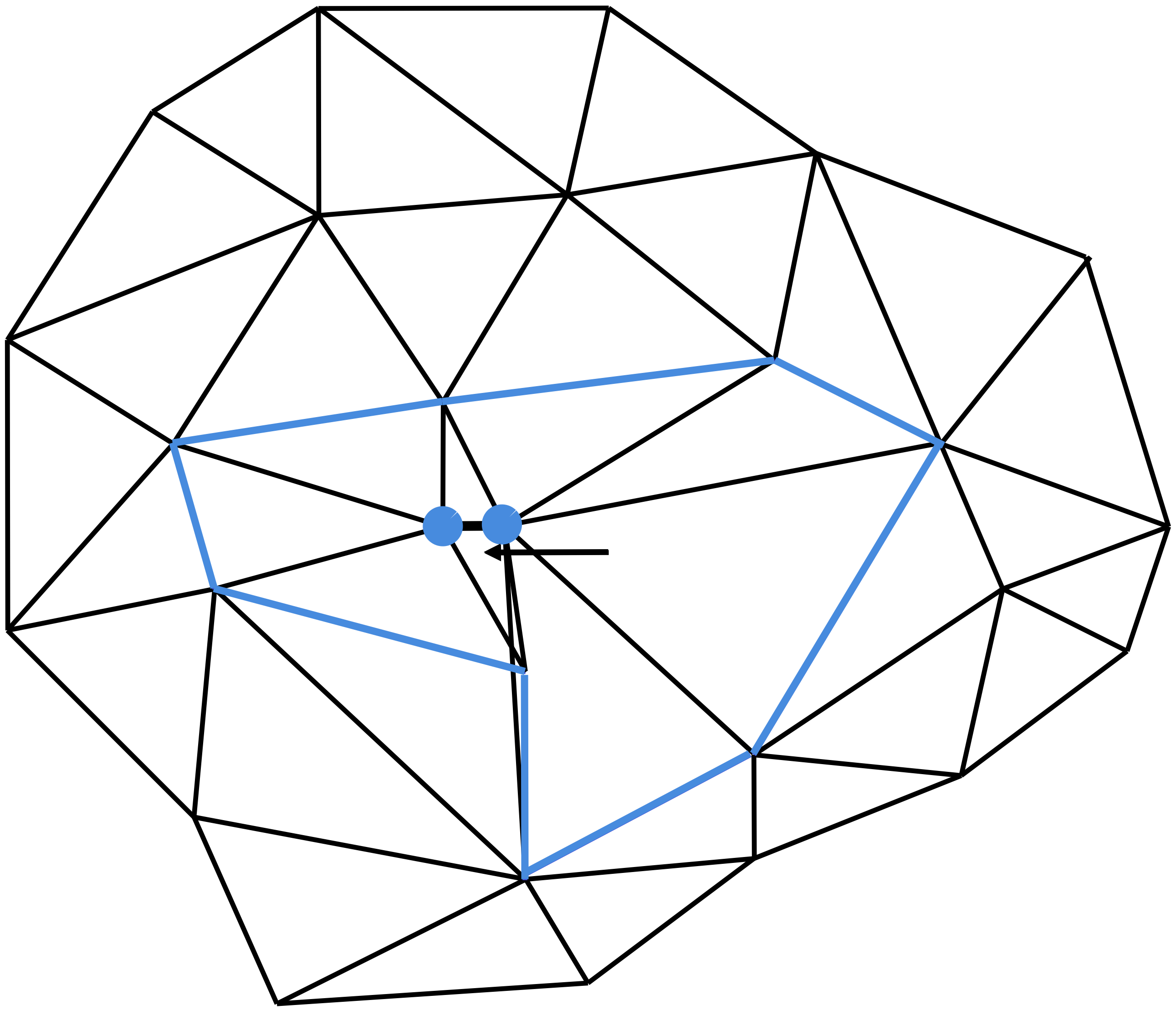
Edge Collapse



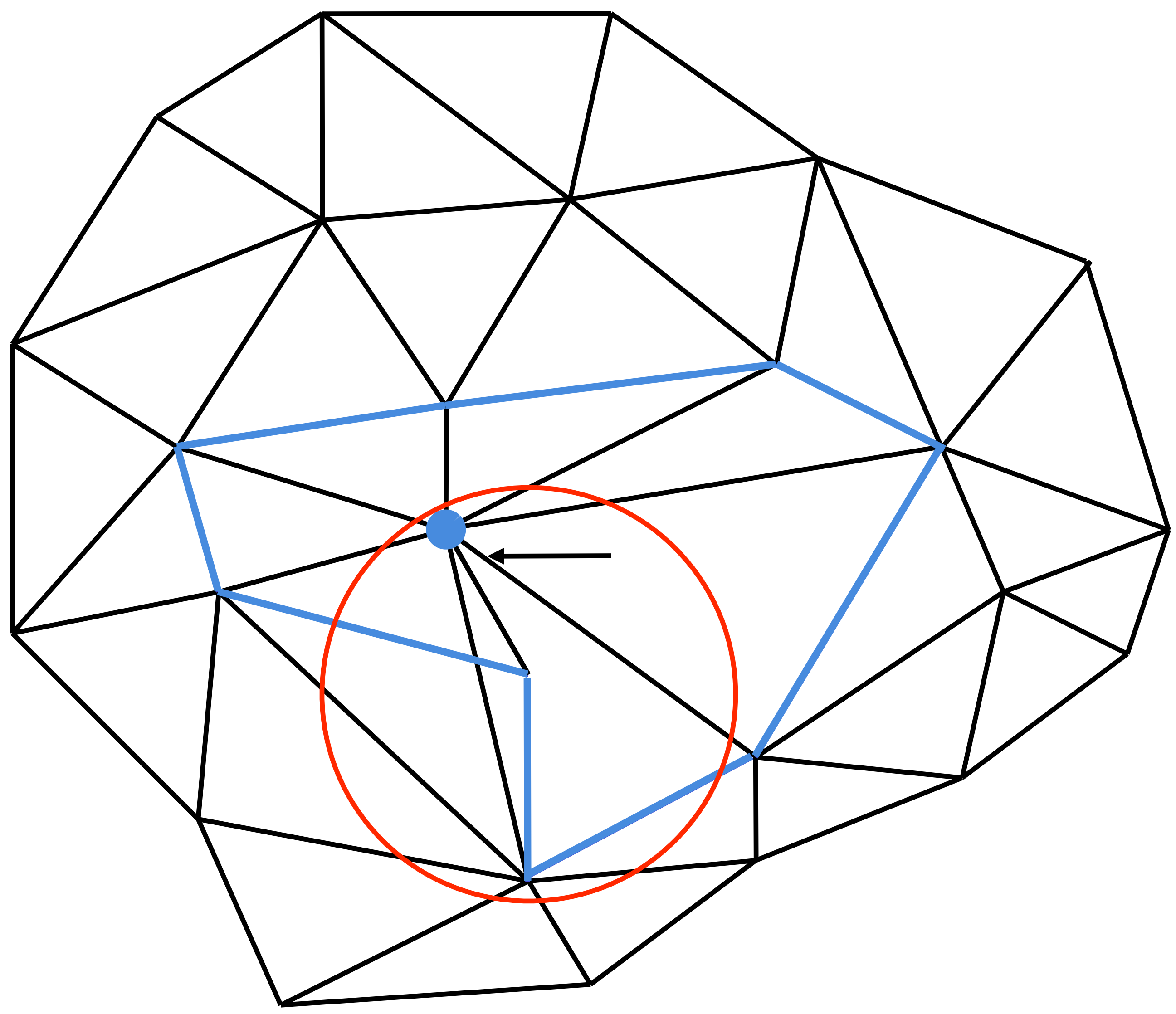
Edge Collapse



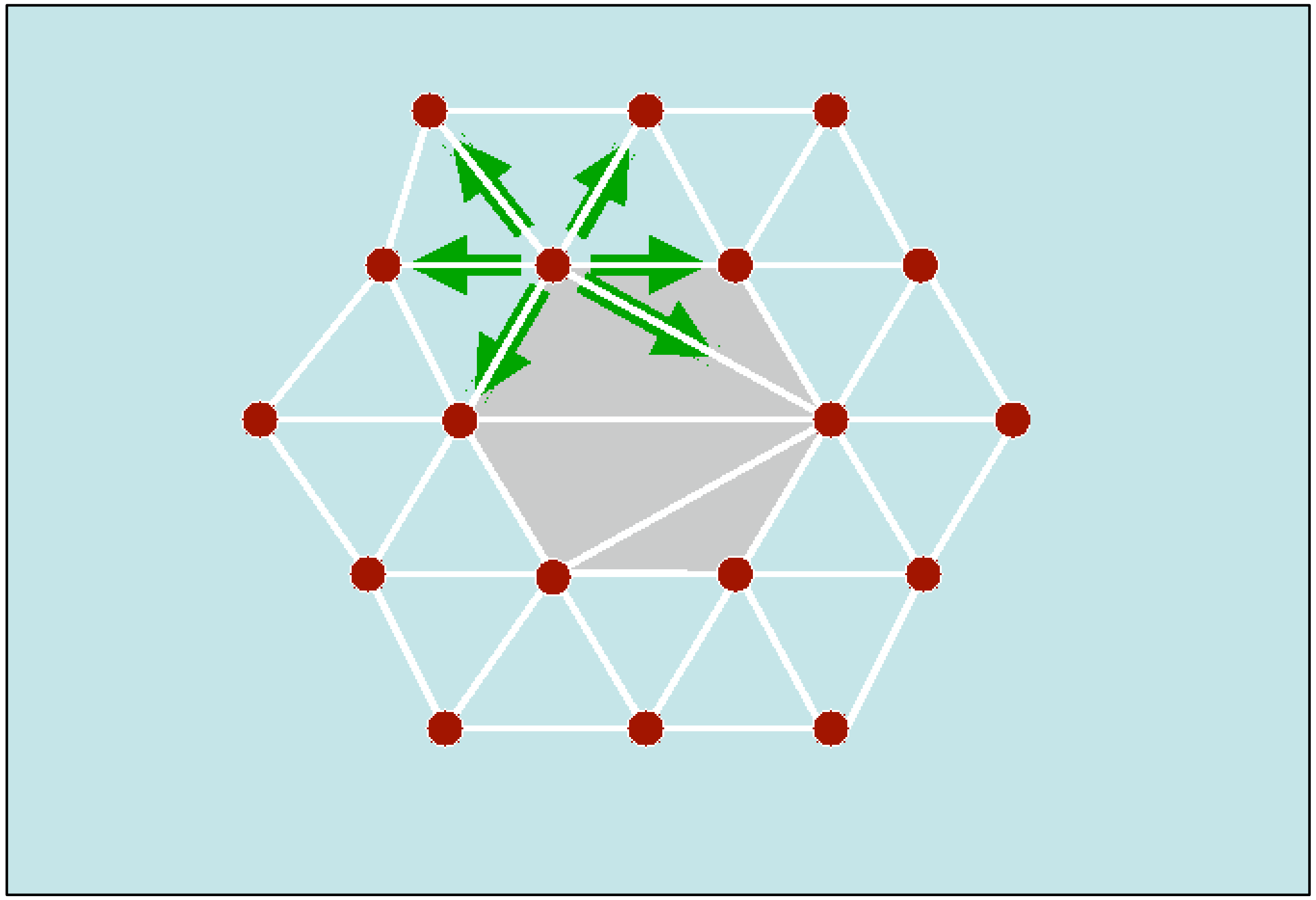
Edge Collapse



Edge Collapse



Priority Queue Updating

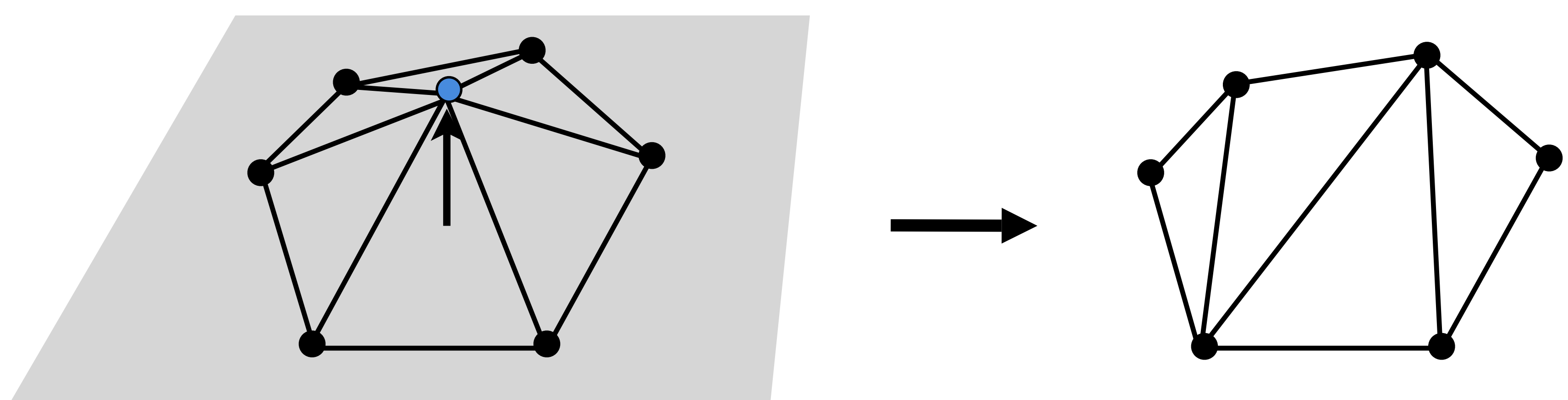


Incremental Decimation

- General Setup
- Decimation operators
- Error metrics
- Fairness criteria
- Topology changes

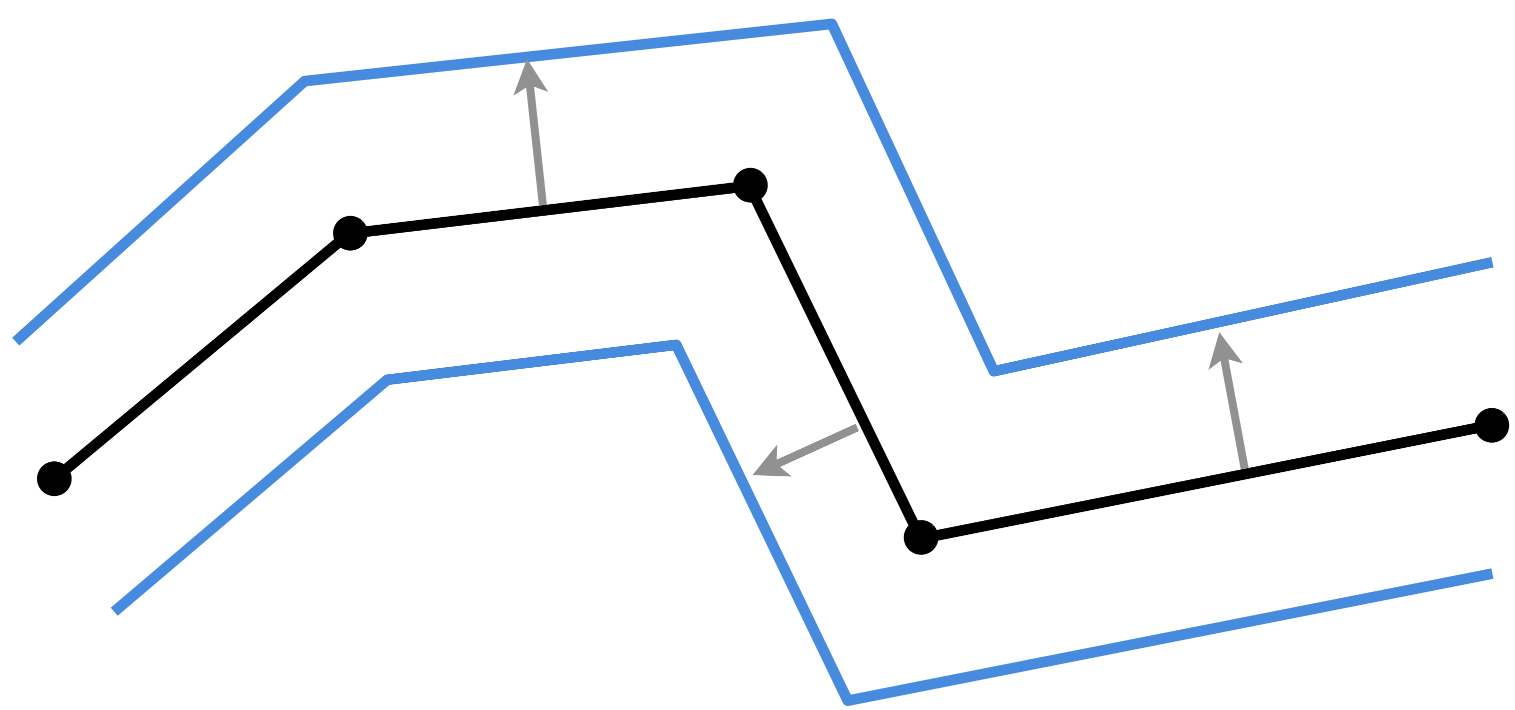
Local Error Metrics

- Local distance to mesh [Schroeder et al. 92]
  - Compute average plane
  - No comparison to *original* geometry



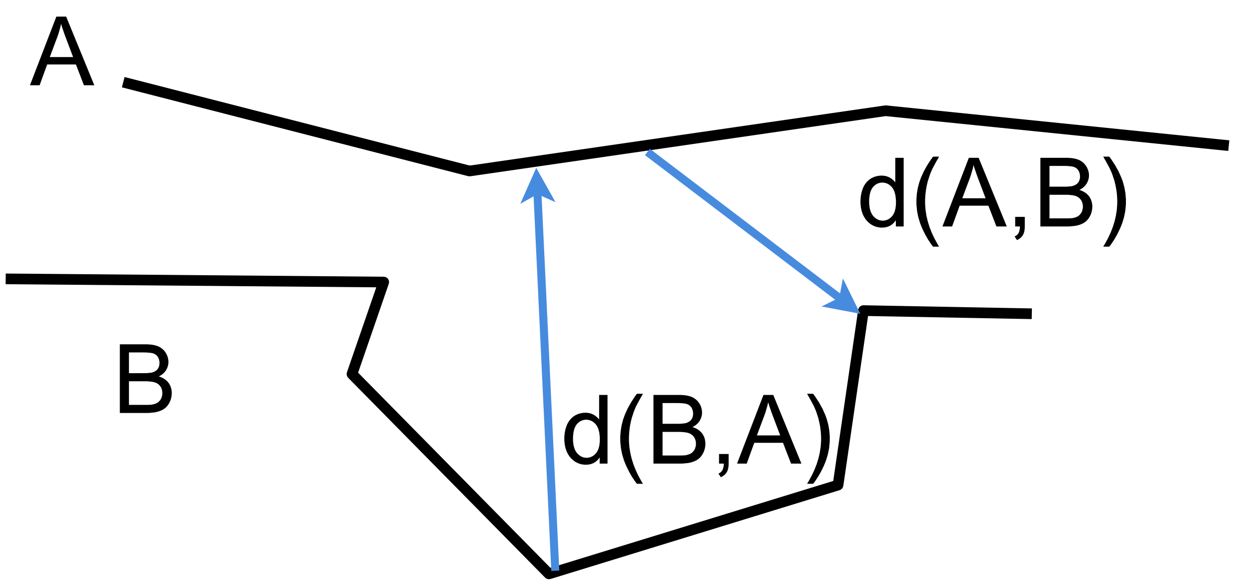
Global Error Metrics

- Simplification envelopes [Cohen et al. 96]
  - Compute (non-intersecting) offset surfaces
  - Simplification guarantees to stay within bounds



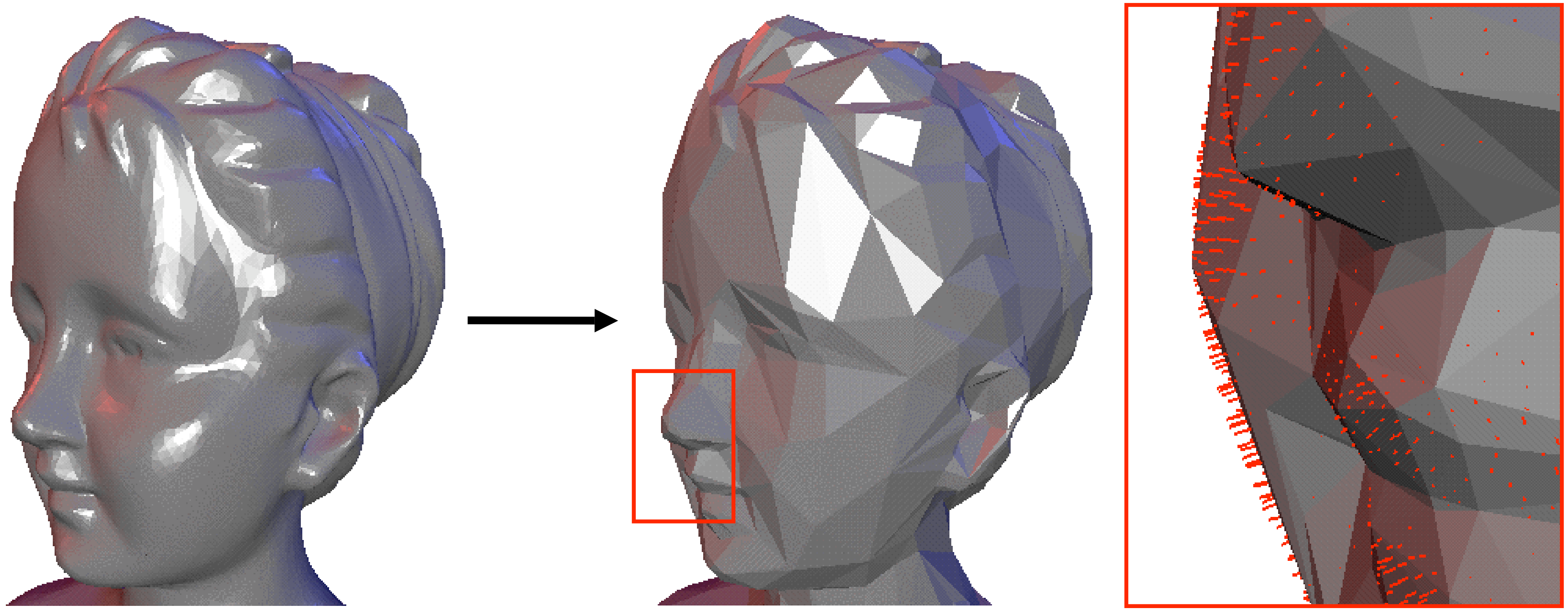
Global Error Metrics

- (Two-sided) Hausdorff distance: Maximum distance between two shapes
  - In general d(A,B) ≠ d(B,A)
  - Computationally involved



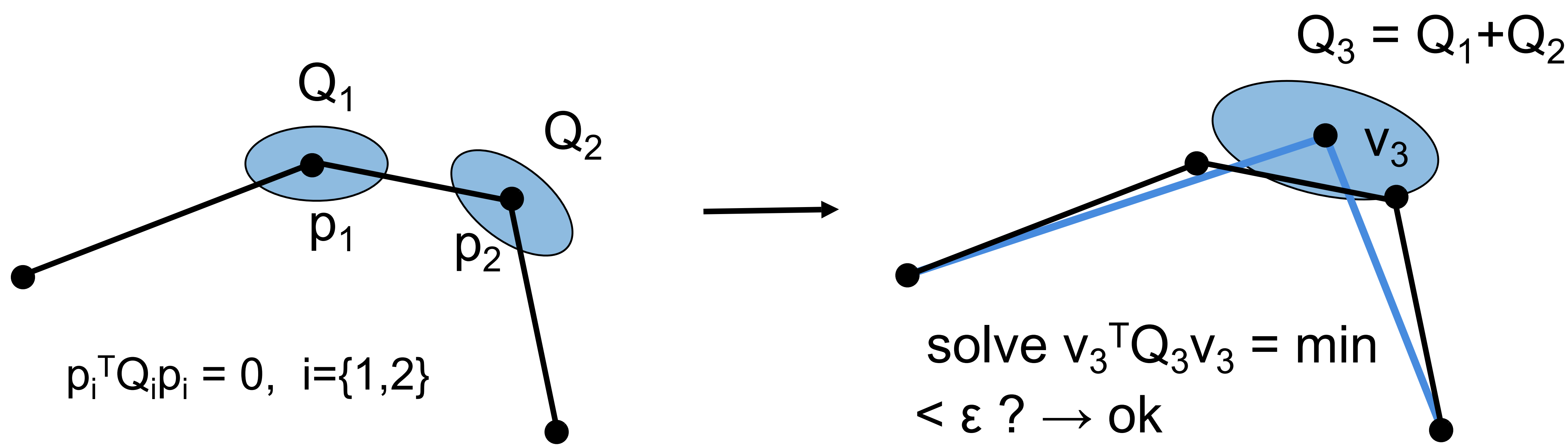
Global Error Metrics

- Scan data: One-sided Hausdorff distance sufficient
  - From original vertices to current surface



Global Error Metrics

- Error quadrics [Garland, Heckbert 97]
  – Squared distance to planes at vertex
  – No upper/lower bound on true error



Complexity

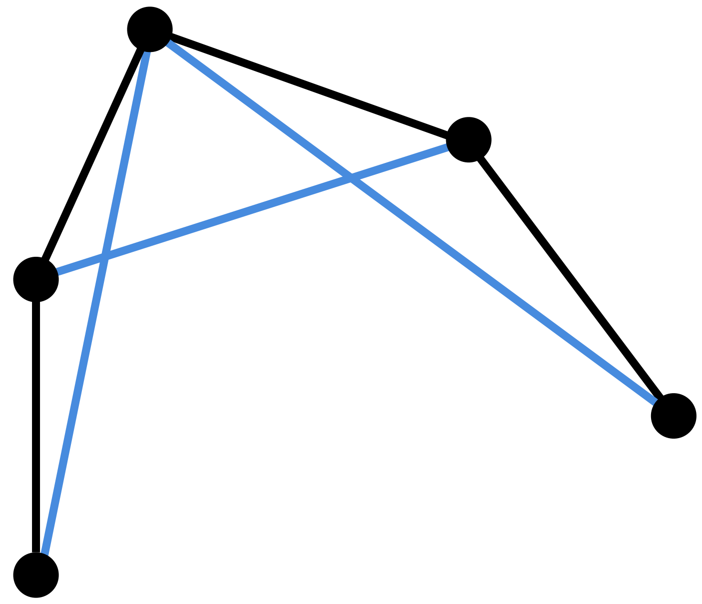
- N = number of vertices
- Priority queue for half-edges
  - 6 N * log (6 N)
- Error control
  - Local O(1) ⇒ global O(N)
  - Local O(N) ⇒ global O(N2)

Incremental Decimation

- General Setup
- Decimation operators
- Error metrics
- Fairness criteria
- Topology changes

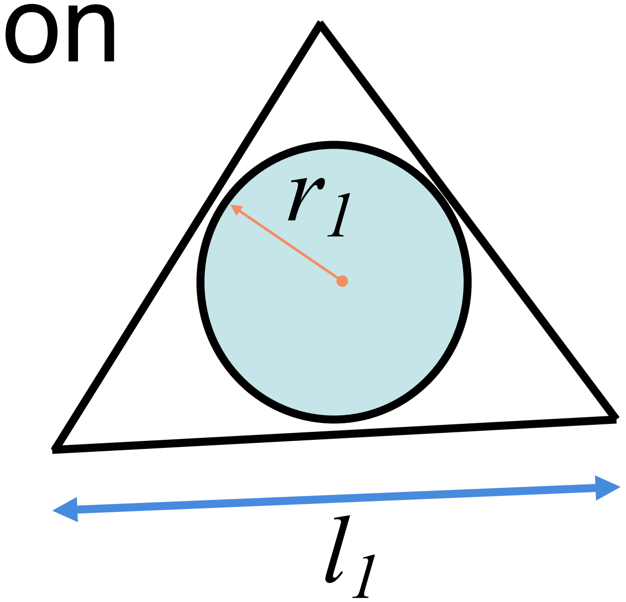
Fairness Criteria

- Rate quality of decimation operation
  - Approximation error
  - Triangle shape
  - Dihedral angles
  - Valence balence
  - Color differences
  - ...

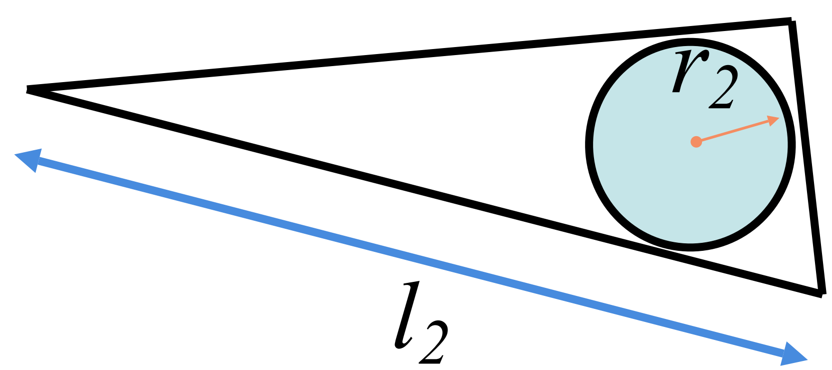


Fairness Criteria

- Rate quality of decimation operation
  - Approximation error
  - Triangle shape
  - Dihedral angles
  - Valence balence
  - Color differences
  - ...

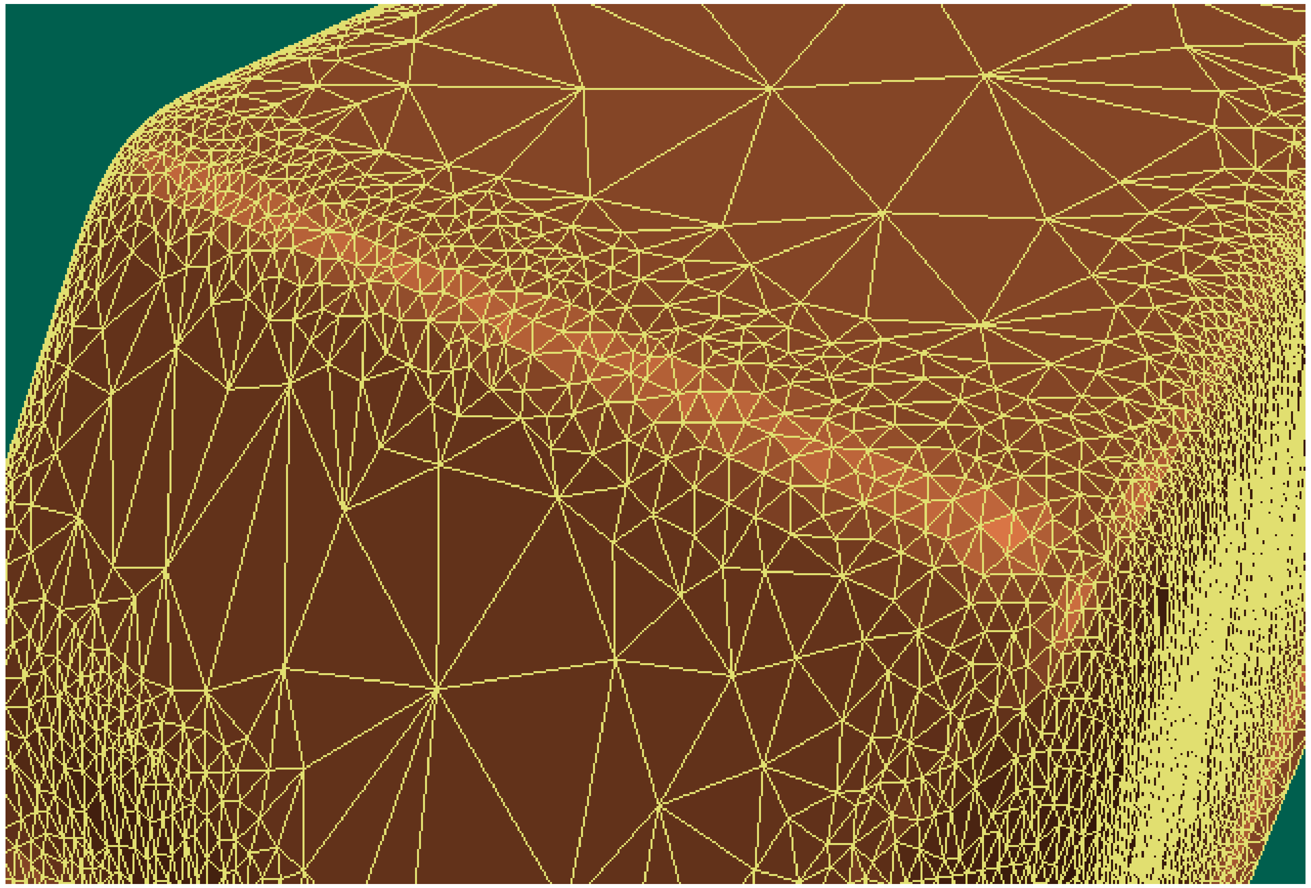


$$\frac{r_1}{l_1} > \frac{r_2}{l_2}$$



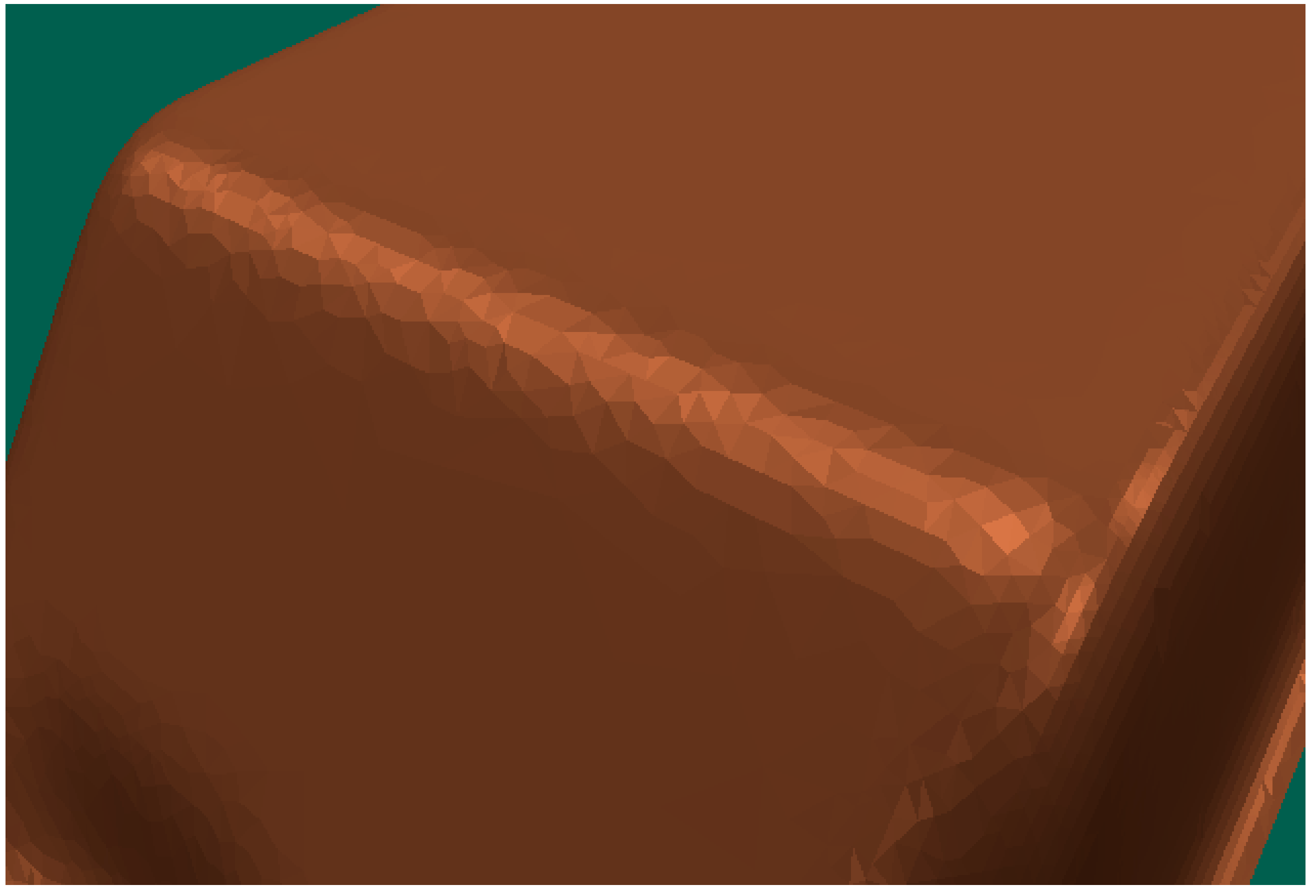
Fairness Criteria

- Rate quality after decimation
  - Approximation error
  - Triangle shape
  - Dihedral angles
  - Valence balence
  - Color differences
  - ...



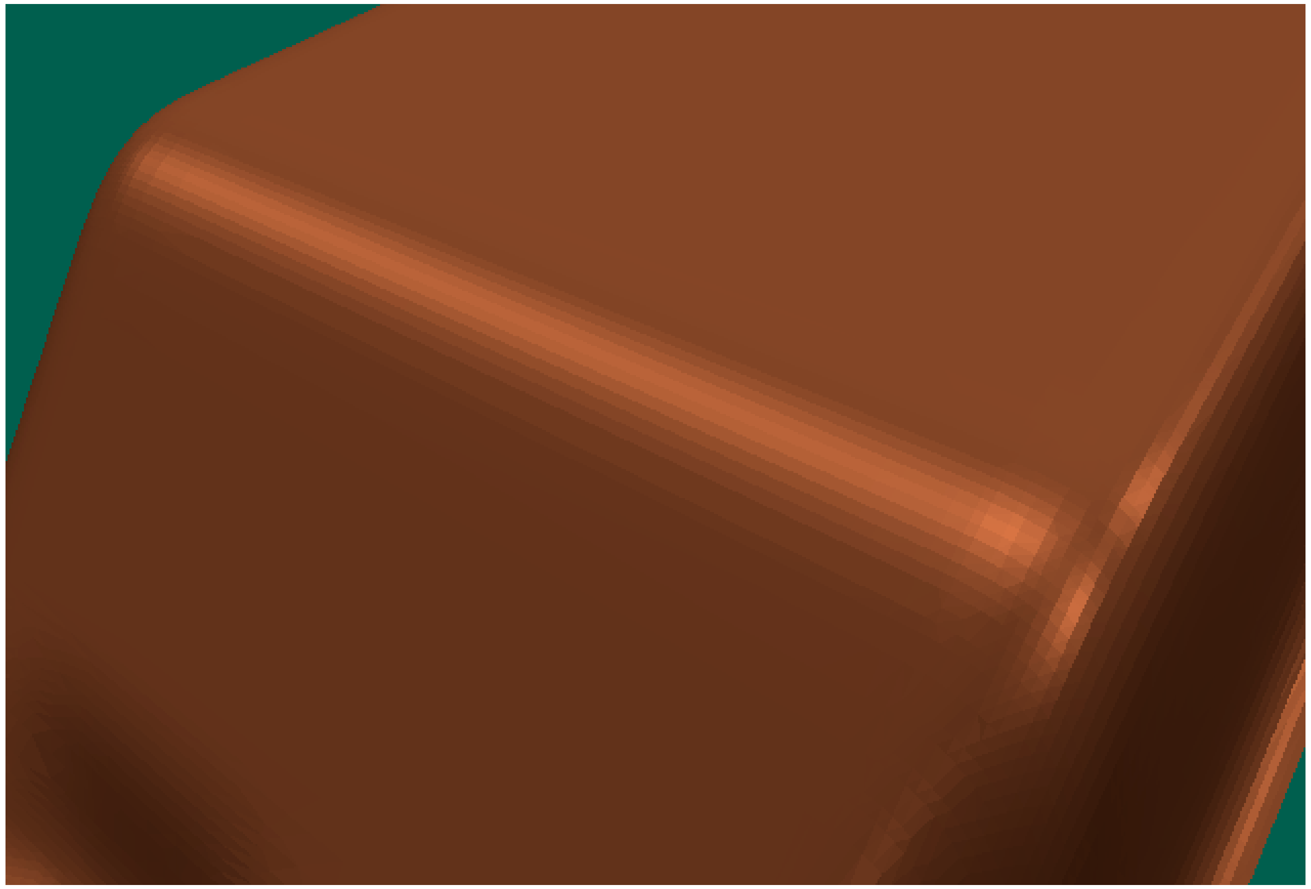
Fairness Criteria

- Rate quality after decimation
  - Approximation error
  - Triangle shape
  - Dihedral angles
  - Valence balence
  - Color differences
  - ...



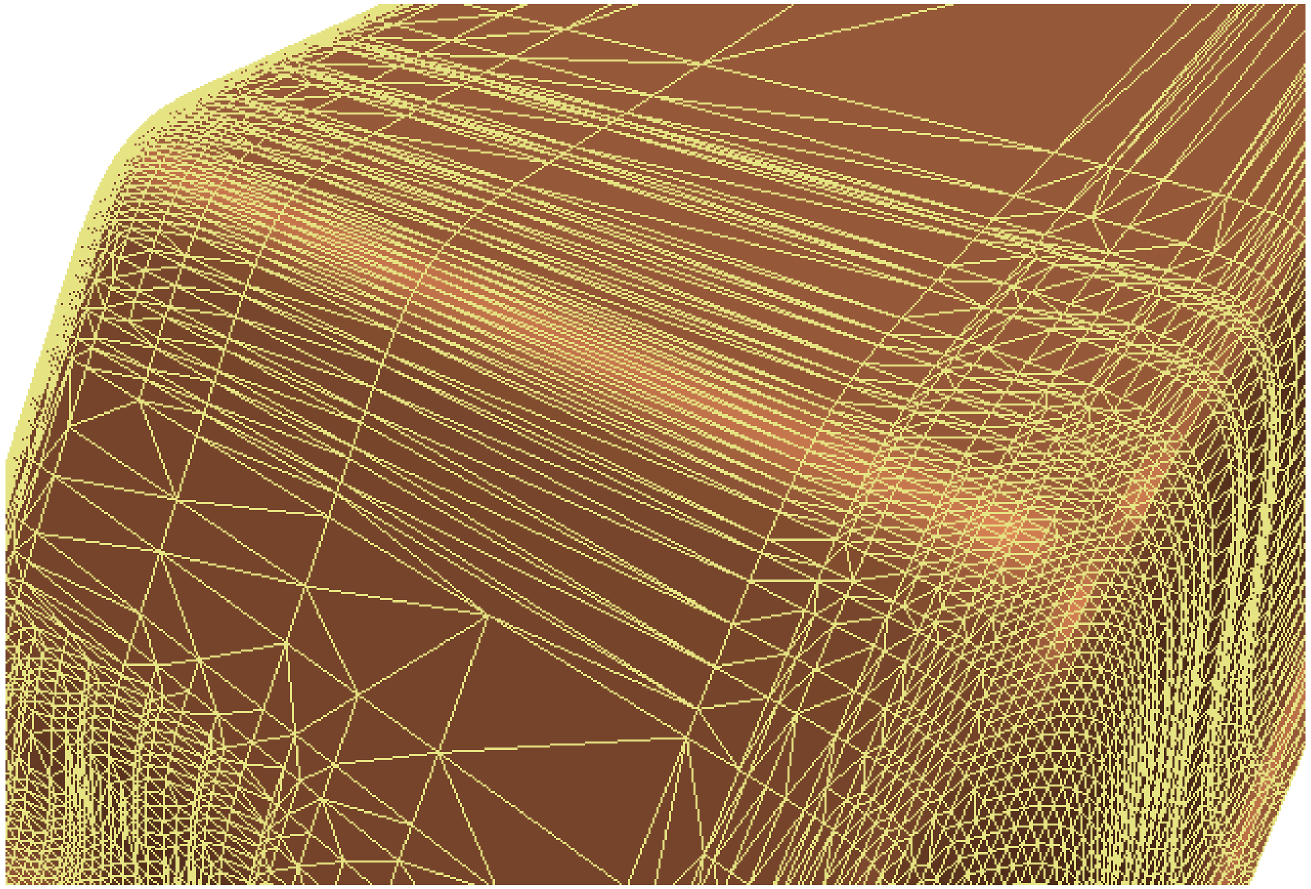
Fairness Criteria

- Rate quality after decimation
  - Approximation error
  - Triangle shape
  - Dihedral angles
  - Valence balence
  - Color differences
  - ...



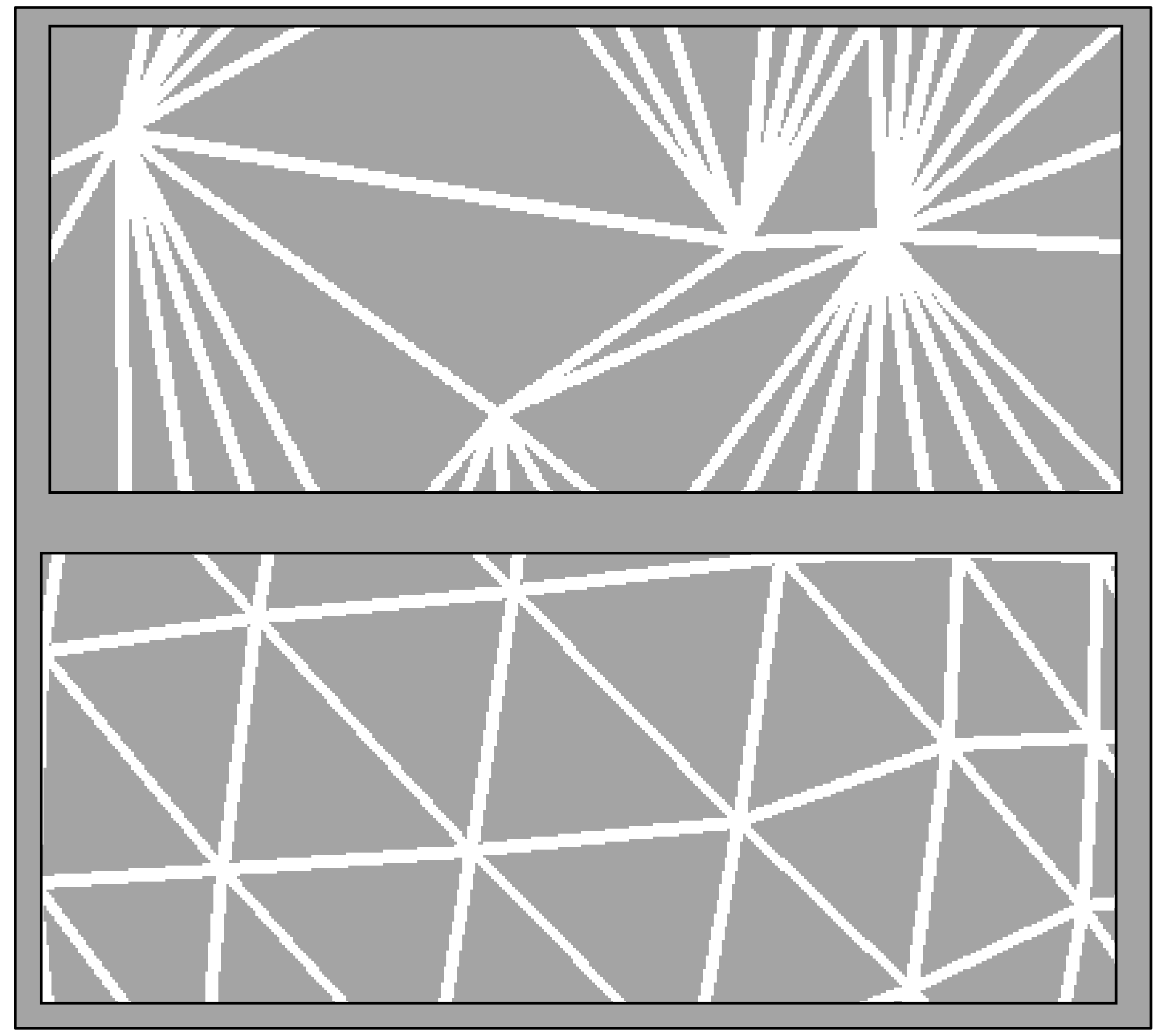
Fairness Criteria

- Rate quality after decimation
  - Approximation error
  - Triangle shape
  - Dihedral angles
  - Valence balence
  - Color differences
  - ...



Fairness Criteria

- Rate quality after decimation
  - Approximation error
  - Triangle shape
  - Dihedral angles
  - Valence balance
  - Color differences
  - ...



Fairness Criteria

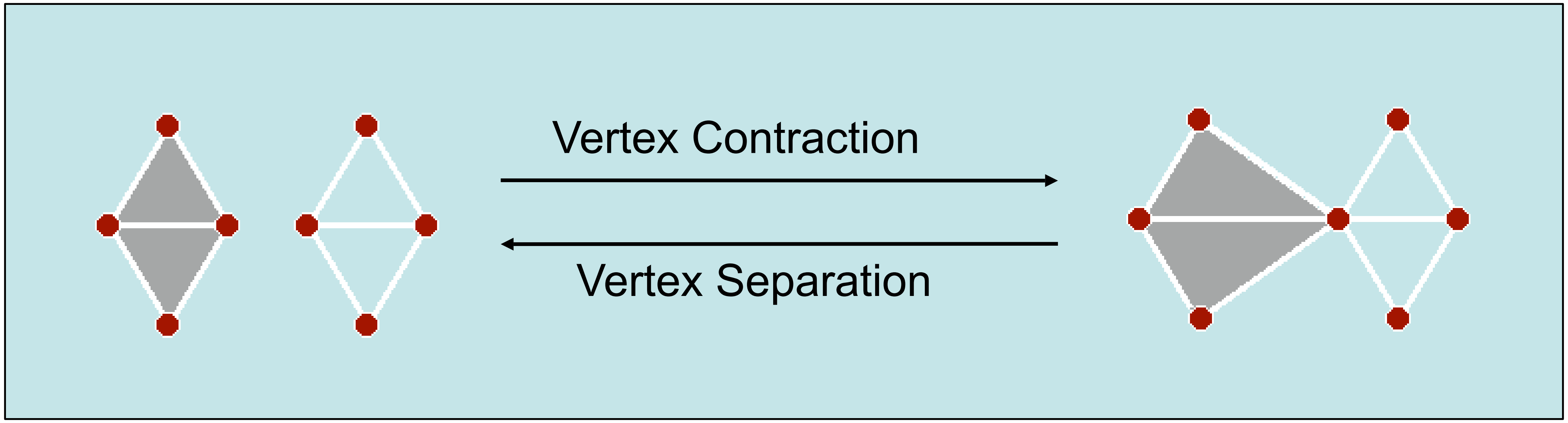
- Rate quality after decimation
  - Approximation error
  - Triangle shape
  - Dihedral angles
  - Valance balance
  - Color differences
  - ...

Incremental Decimation

- General Setup
- Decimation operators
- Error metrics
- Fairness criteria
- Topology changes

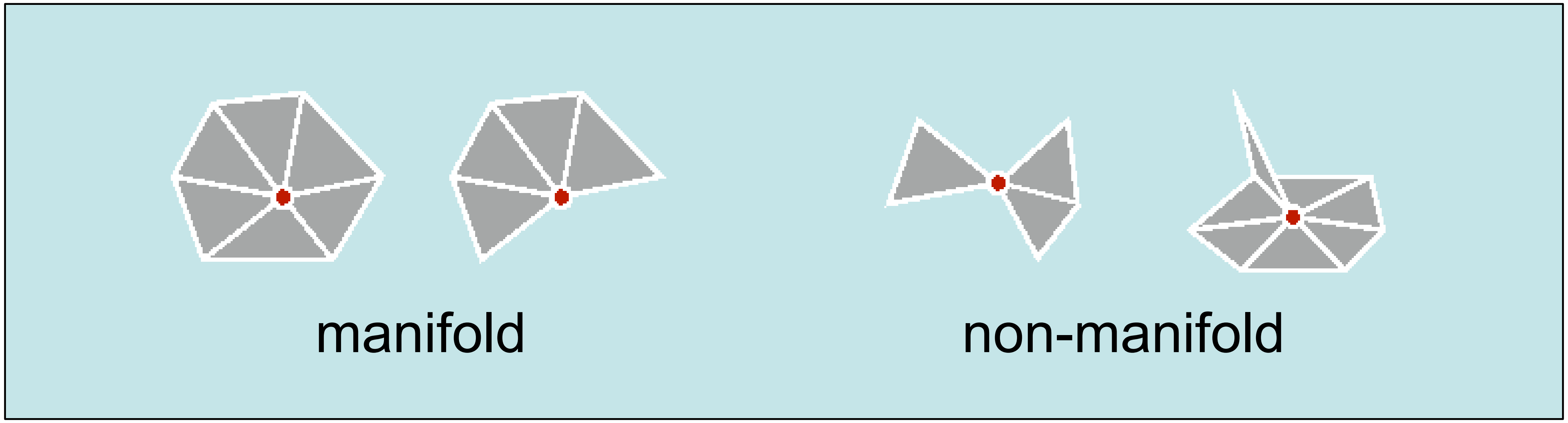
Topology Changes ?

- Merge vertices across non-edges
  - Changes mesh topology
  - Need *spatial neighborhood* information
  - Generates *non-manifold* meshes



Topology Changes ?

- Merge vertices across non-edges
  - Changes mesh topology
  - Need *spatial neighborhood* information
  - Generates *non-manifold* meshes

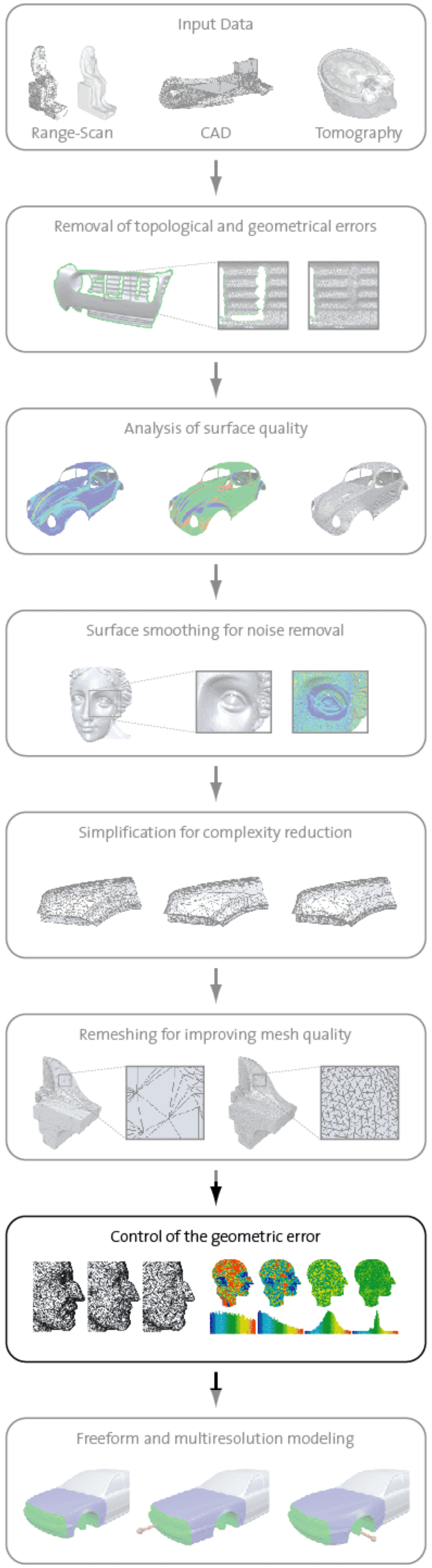


Summary

- Vertex clustering
  - fast, but difficult to control simplified mesh
- Iterative decimation with quadric error metrics
  - good trade-off between mesh quality and speed
  - explicit control over mesh topology
  - restricting normal deviation improves mesh quality
- Global error control

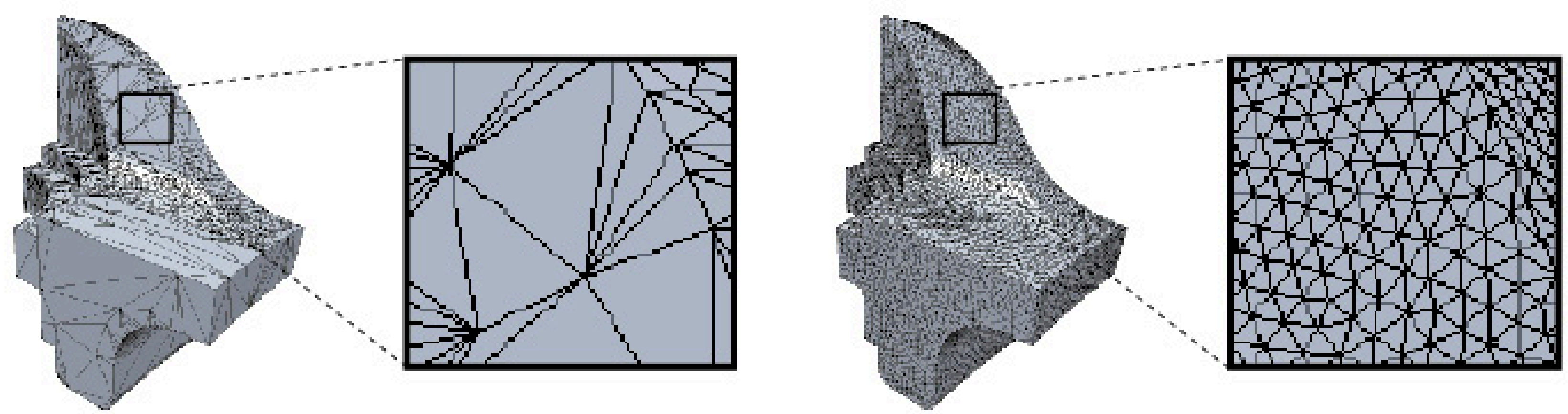
Links & Literature

- Kobbelt et al: *Geometric Modeling based on Polygonal Meshes,* Eurographics 2000 Course Notes
- Schroeder, Zarge, Lorensen: *Decimation of triangle meshes*, SIGGRAPH 1992
- Cohen, Varshney, Manocha, Turk, Weber, Agarwal, Brooks, Wright: *Simplification envelopes*, SIGGRAPH 1996
- Garland, Heckbert: *Surface simplification using quadric error metrics*, SIGGRAPH 1997
- David Luebke: *A Developer's Survey of Polygonal Simplification Algorithms*, IEEE Computer Graphics &Applications, 2001



Isotropic Remeshing

Remeshing for improving mesh quality

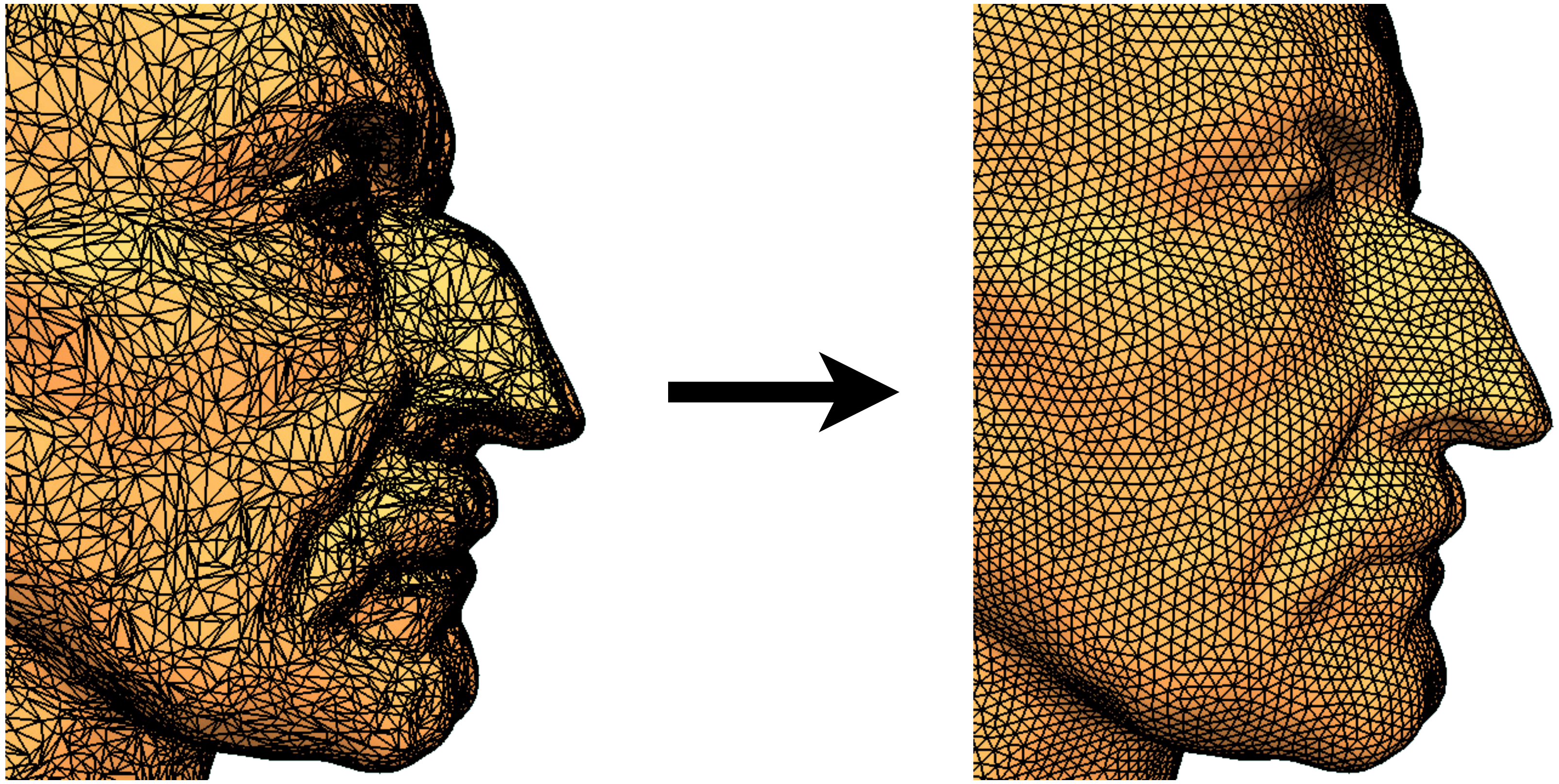


ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

inf | Informatik Computer Science

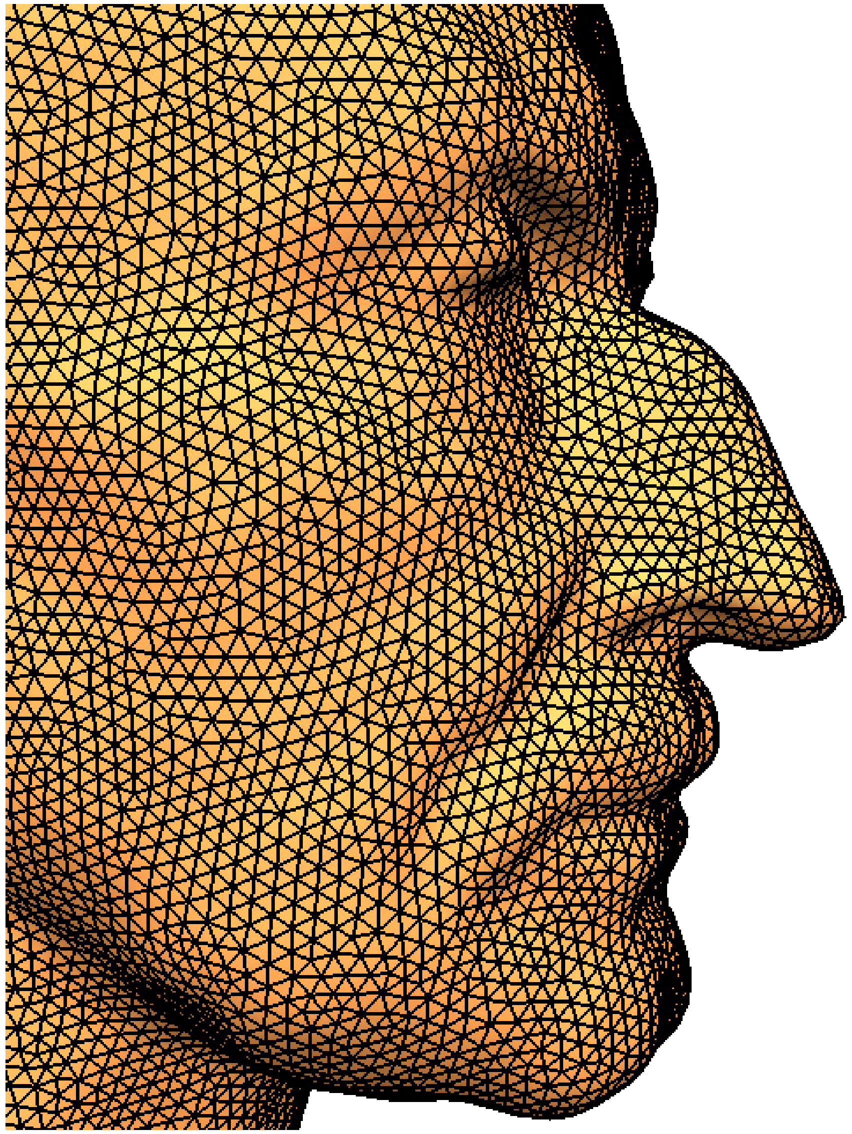
Isotropic Remeshing

- High quality tessellation (numerical simulation)
  - Keep surface geometry
  - Optimize triangulation



Isotropic Remeshing

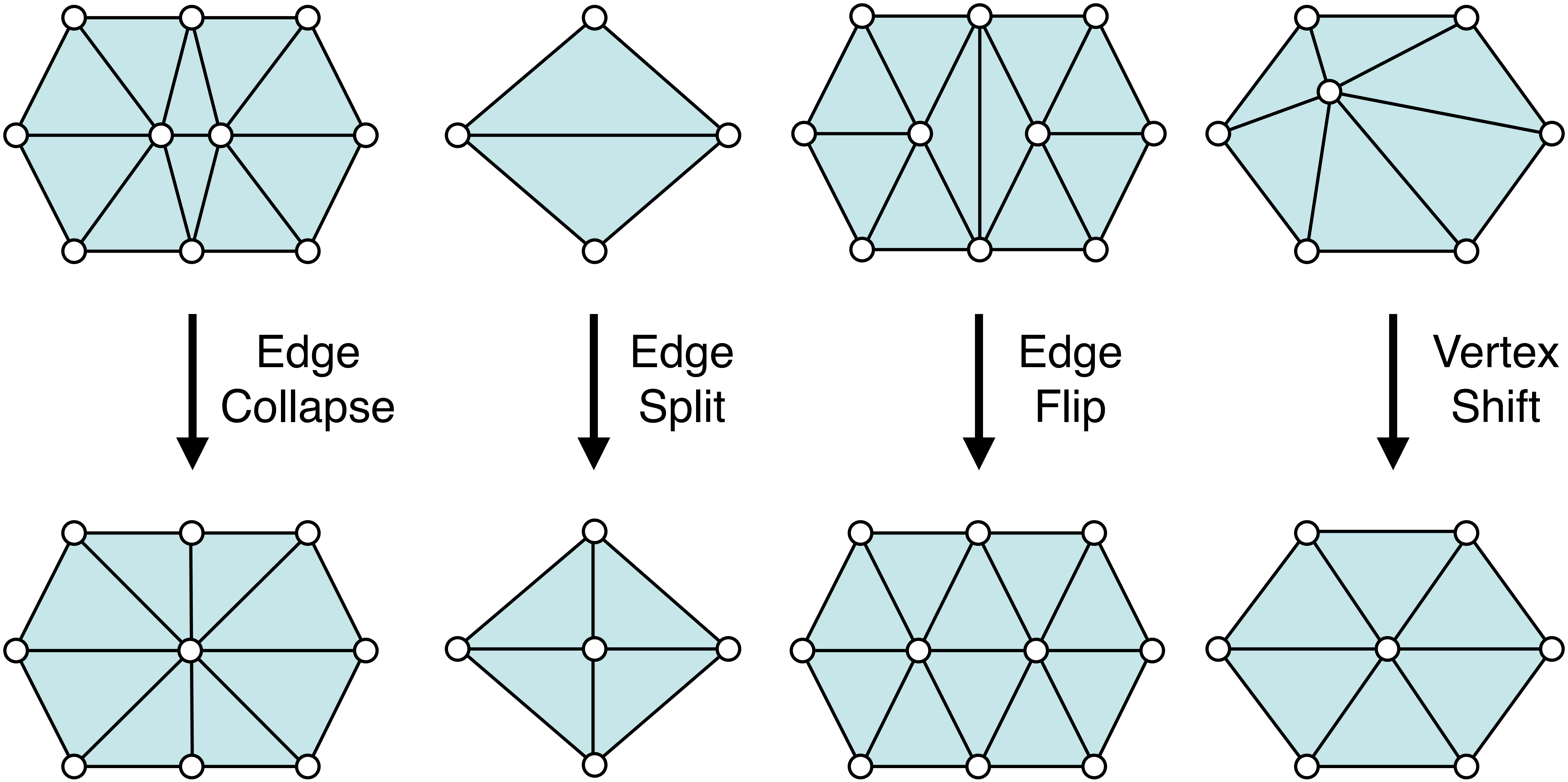
- High quality tessellation (numerical simulation)
  - Keep surface geometry
  - Optimize triangulation
    - Equilateral triangles
    - Equal edge lengths
    - Uniform vertex density
    - Vertex-valence 6



Isotropic Remeshing

- Use global parametrization?
  - Numerically very sensitive
  - Topological restrictions
- Use local parametrization?
  - Expensive computations
- Use local operators & back-projections!
  - Resampling of 100k triangles in < 5s

Local Remeshing Operators



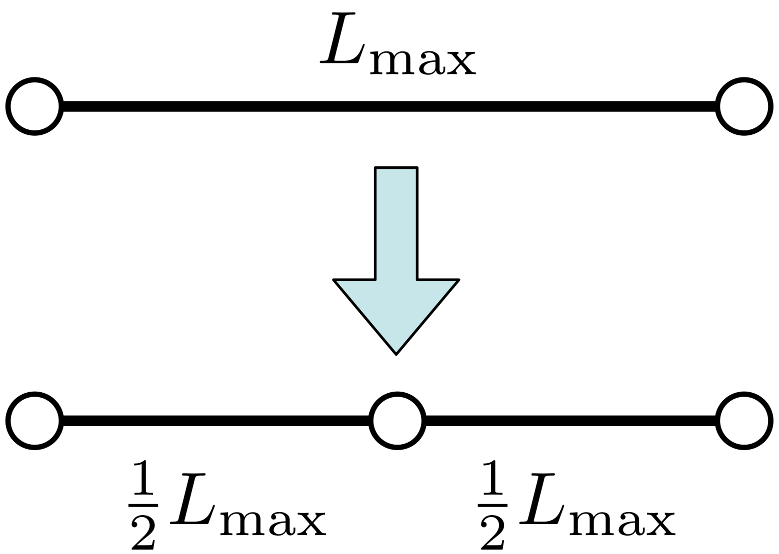
Isotropic Remeshing

Specify target edge length L

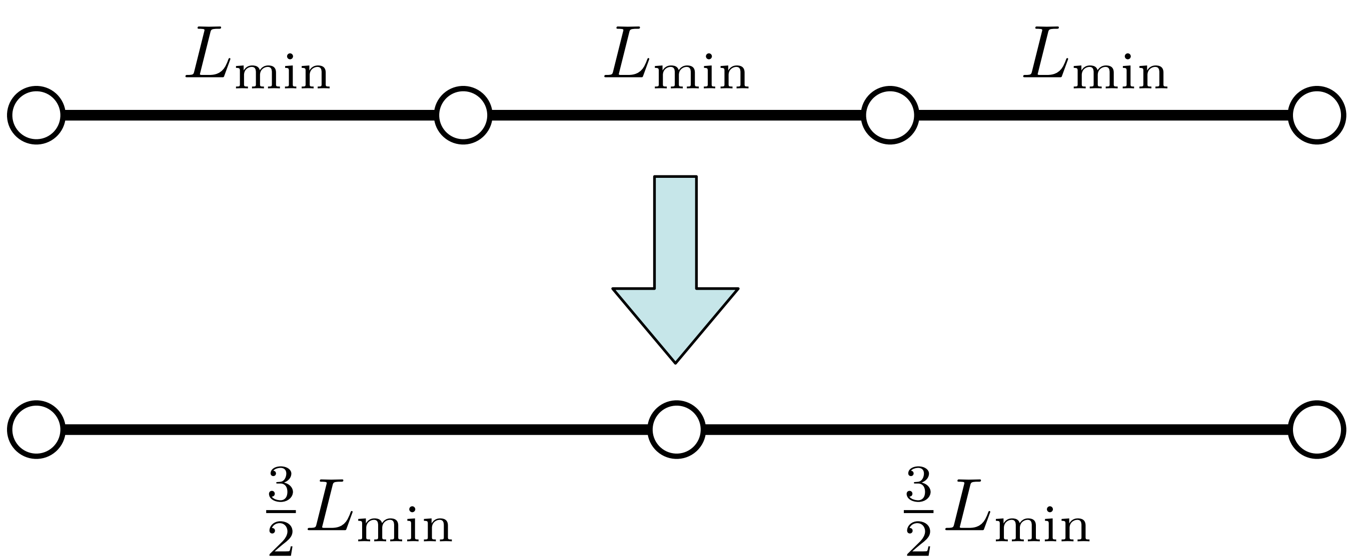
Iterate:

1. Split edges longer than L_{max}
2. Collapse edges shorter than L_{min}
3. Flip edges to get closer to valence 6
4. Vertex shift by tangential relaxation
5. Project vertices onto reference mesh

Edge Collapse / Split



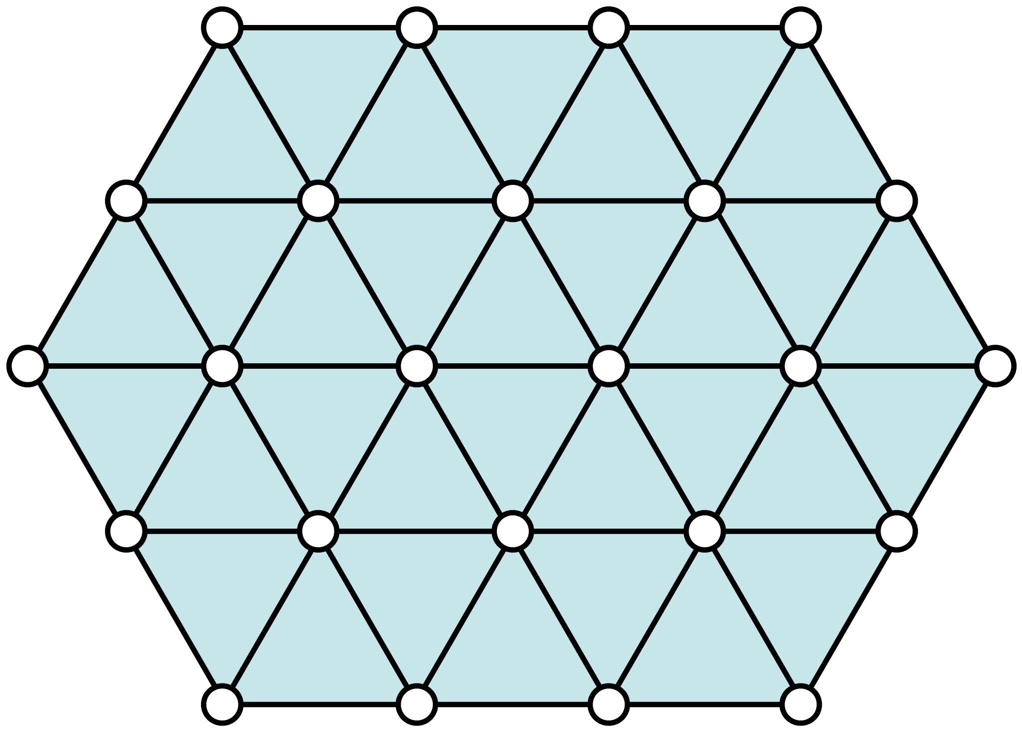
$$|L_{\max} - L| = \left|\tfrac{1}{2}L_{\max} - L\right|$$
$$\Rightarrow L_{\max} = \tfrac{4}{3}L$$



$$|L_{\min} - L| = \left|\tfrac{3}{2}L_{\max} - L\right|$$
$$\Rightarrow L_{\min} = \tfrac{4}{5}L$$

Edge Flip

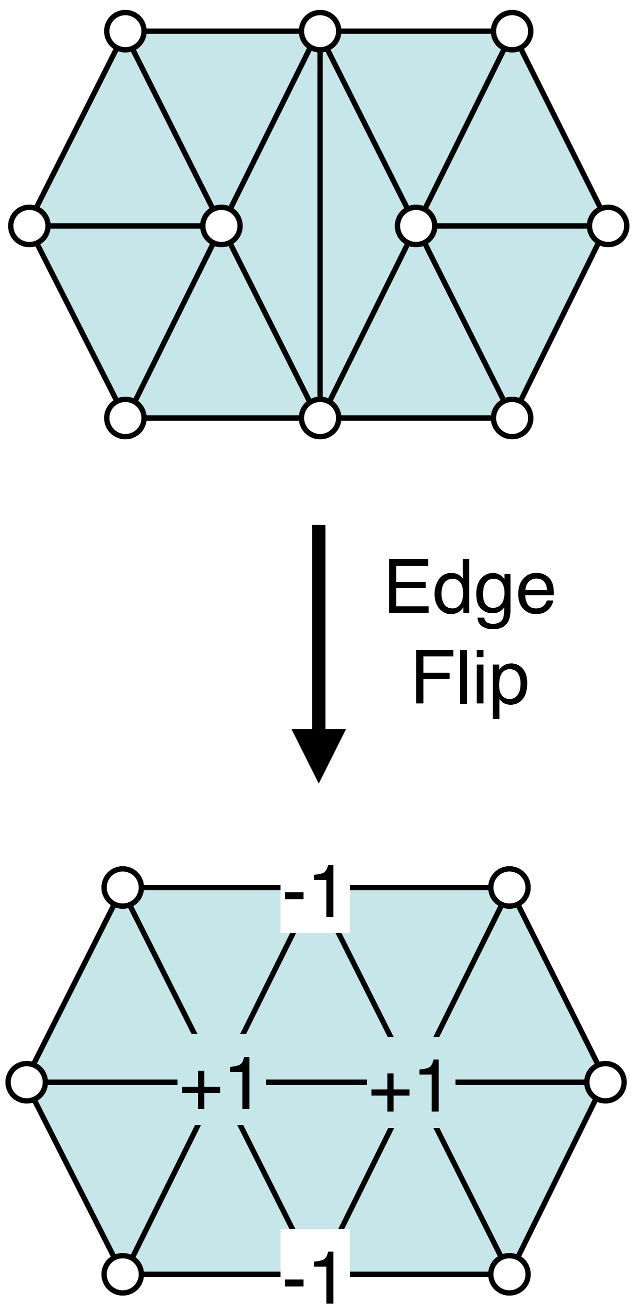
- Improve valences
  - Avg. valence is 6 (Euler)
  - Reduce variation
- Optimal valence is
  - 6 for interior vertices
  - 4 for boundary vertices



Edge Flip

- Improve valences
  - Avg. valence is 6 (Euler)
  - Reduce variation
- Optimal valence is
  - 6 for interior vertices
  - 4 for boundary vertices
- Minimize valence excess

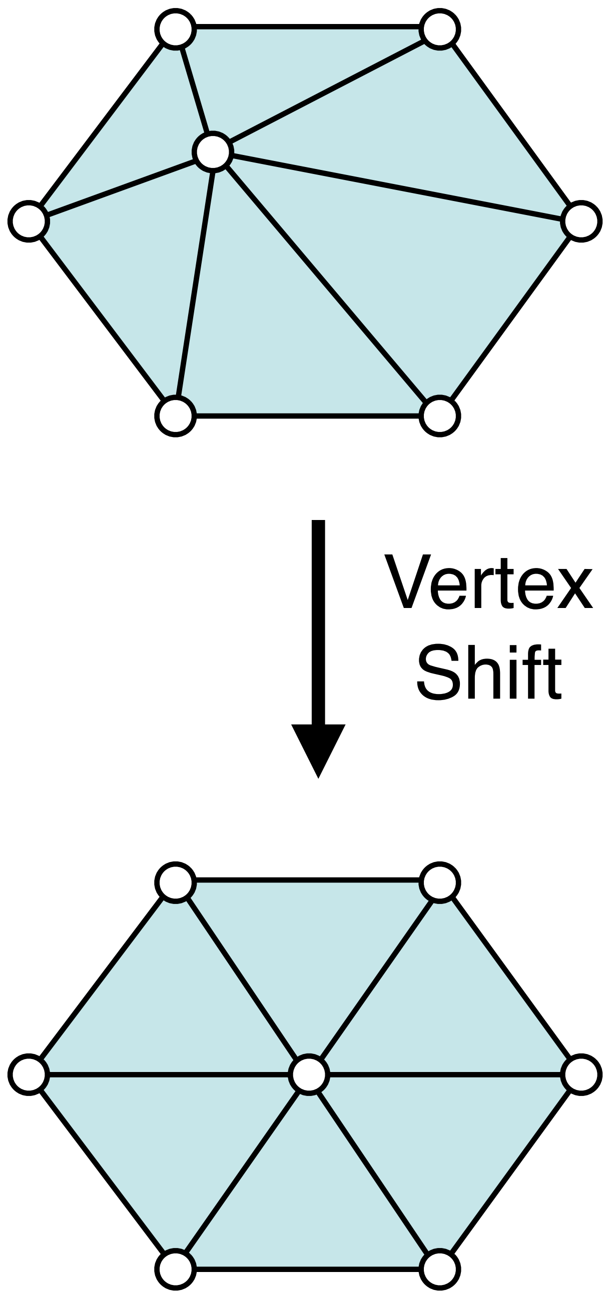
$$\sum_{i=1}^{4} (\text{valence}(v_i) - \text{opt\_valence}(v_i))^2$$



Vertex Shift

- Local "spring" relaxation
  - Uniform Laplacian smoothing
  - Bary-center of one-ring neighbors

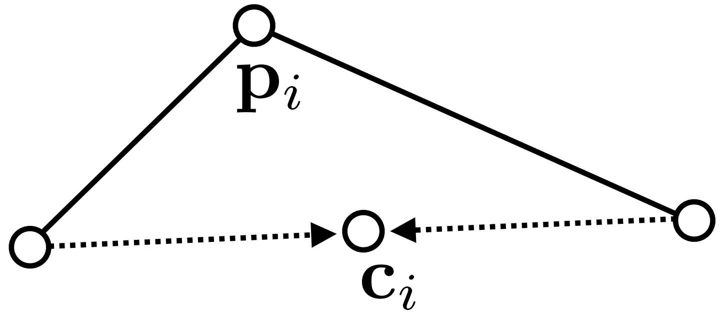
$$\mathbf{c}_i = \frac{1}{\text{valence}\,(v_i)} \sum_{j \in N(v_i)} \mathbf{p}_j$$



Vertex Shift

- Local "spring" relaxation
  - Uniform Laplacian smoothing
  - Bary-center of one-ring neighbors

$$\mathbf{c}_i = \frac{1}{\text{valence}\,(v_i)} \sum_{j \in N(v_i)} \mathbf{p}_j$$



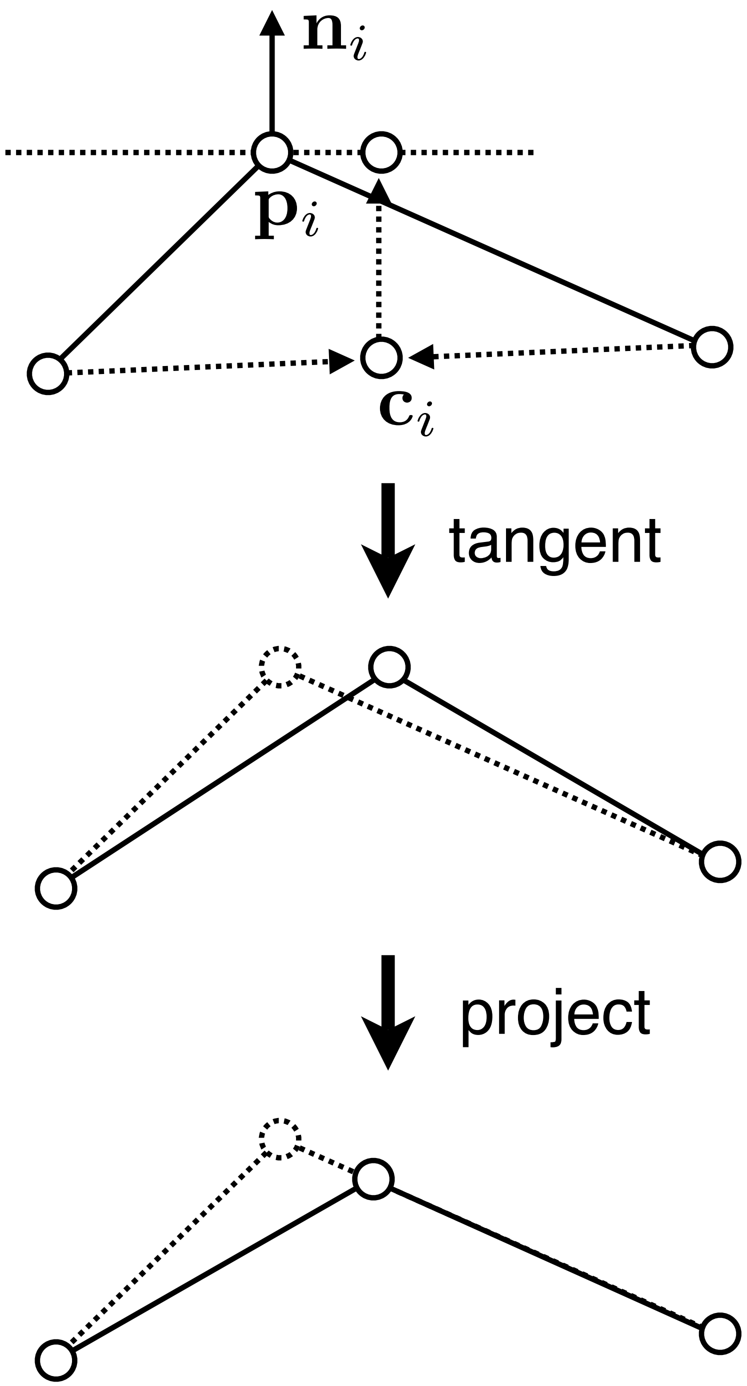
Vertex Shift

- Local “spring” relaxation
  - Uniform Laplacian smoothing
  - Bary-center of one-ring neighbors

$$\mathbf{c}_i = \frac{1}{\text{valence}\,(v_i)} \sum_{j \in N(v_i)} \mathbf{p}_j$$

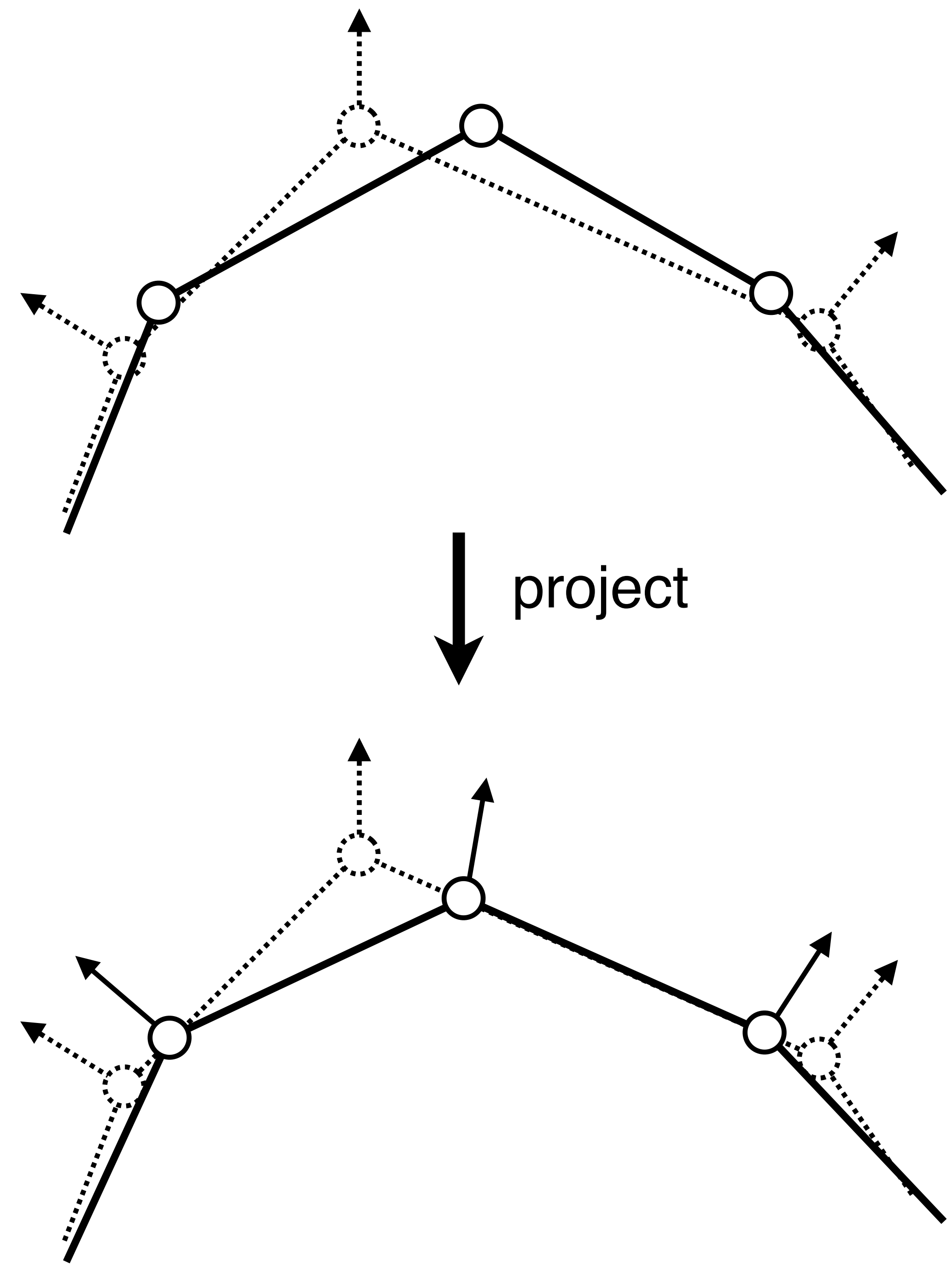
- Keep vertex (approx.) of surface
  - Restrict movement to tangent plane

$$\mathbf{p}_i \leftarrow \mathbf{p}_i + \lambda \left(I - \mathbf{n}_i \mathbf{n}_i^T \right) (\mathbf{c}_i - \mathbf{p}_i)$$



Vertex Projection

- Project vertices onto original reference mesh
  - Static reference mesh
  - Precompute BSP
- Assign position & interpolated normal



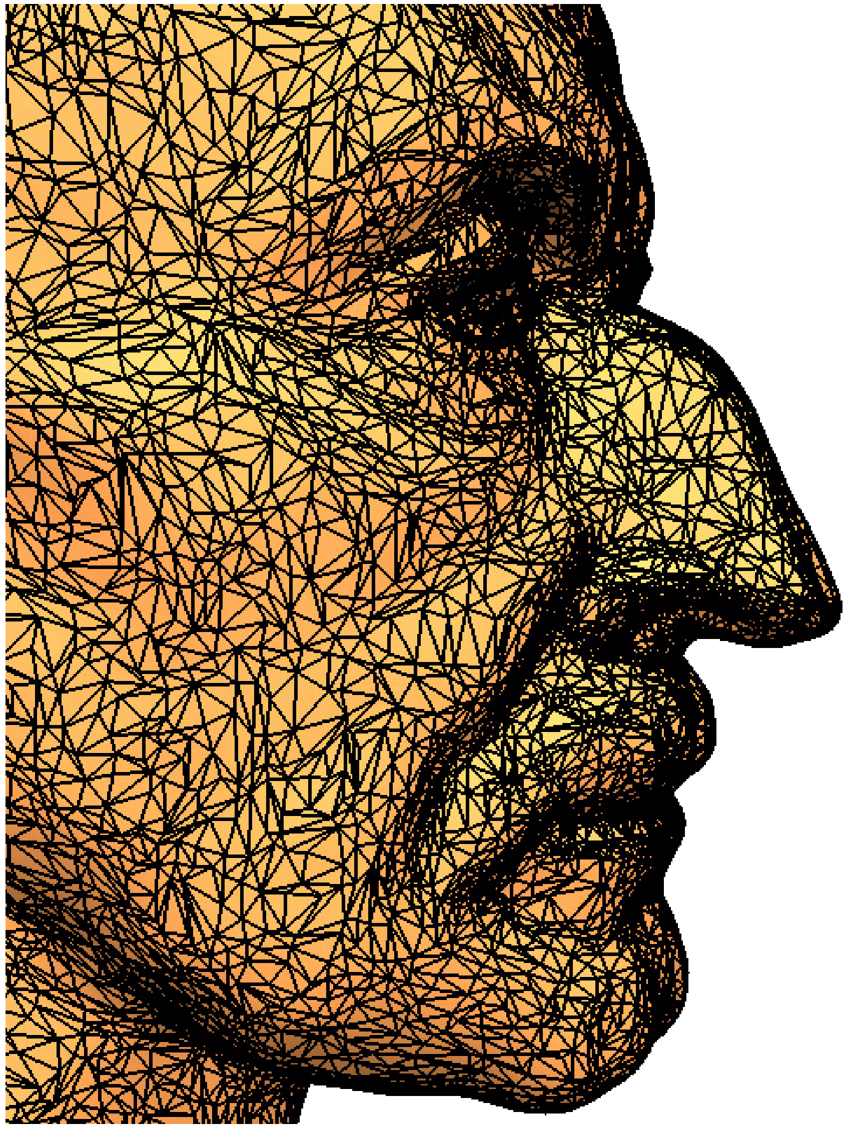
Isotropic Remeshing

Specify target edge length L

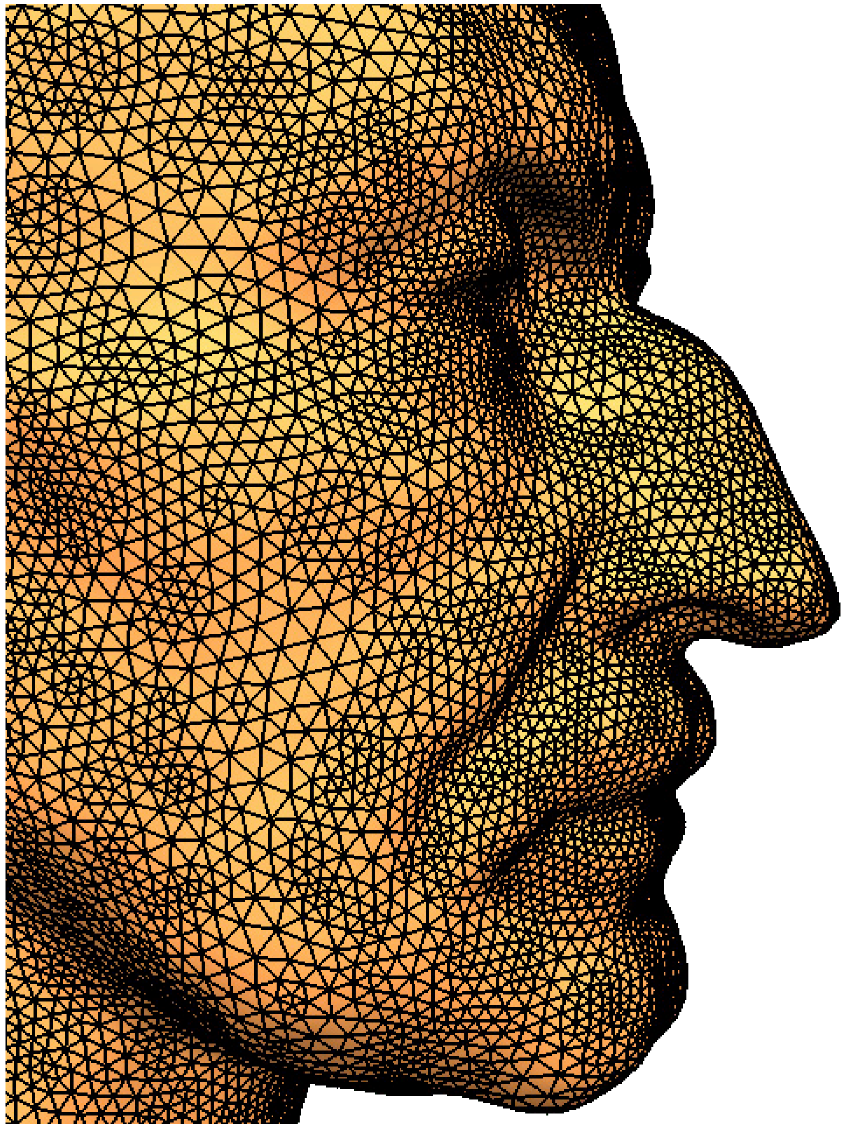
Iterate:

1. Split edges longer than L_{max}
2. Collapse edges shorter than L_{min}
3. Flip edges to get closer to valence 6
4. Vertex shift by tangential relaxation
5. Project vertices onto reference mesh

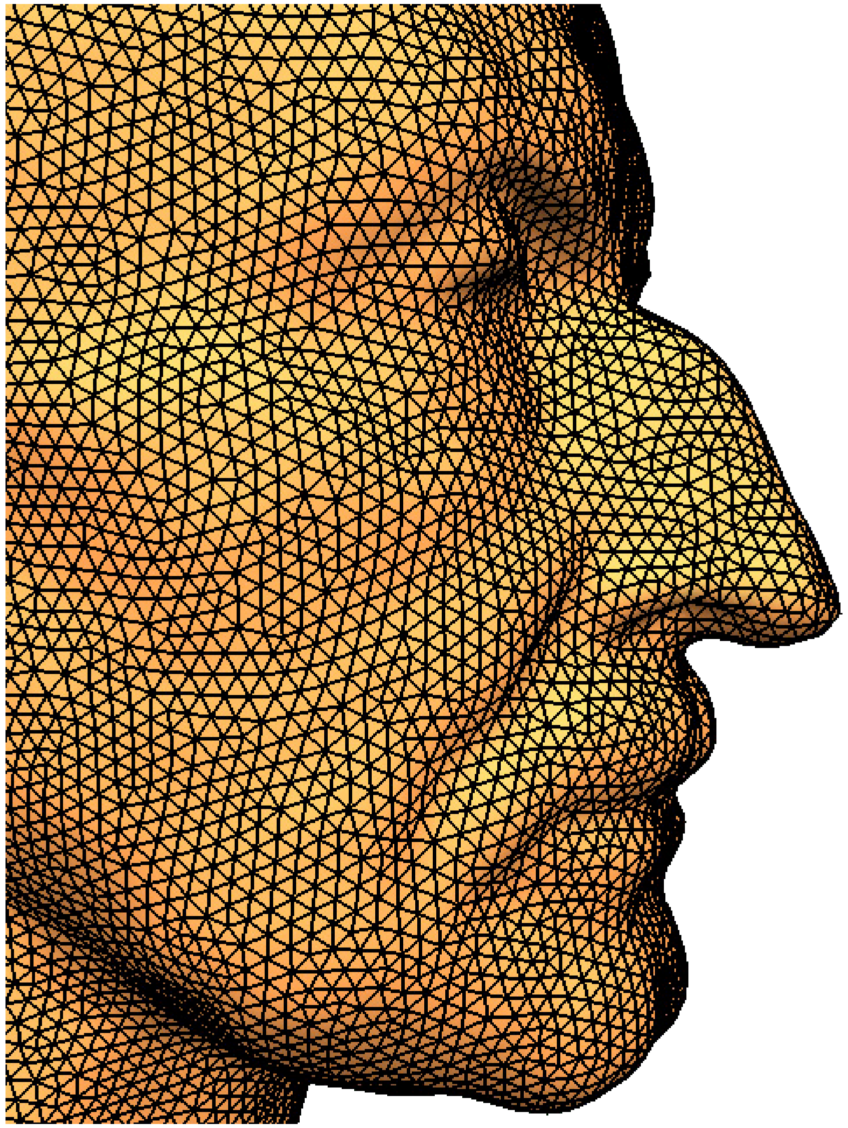
Remeshing Results



Original

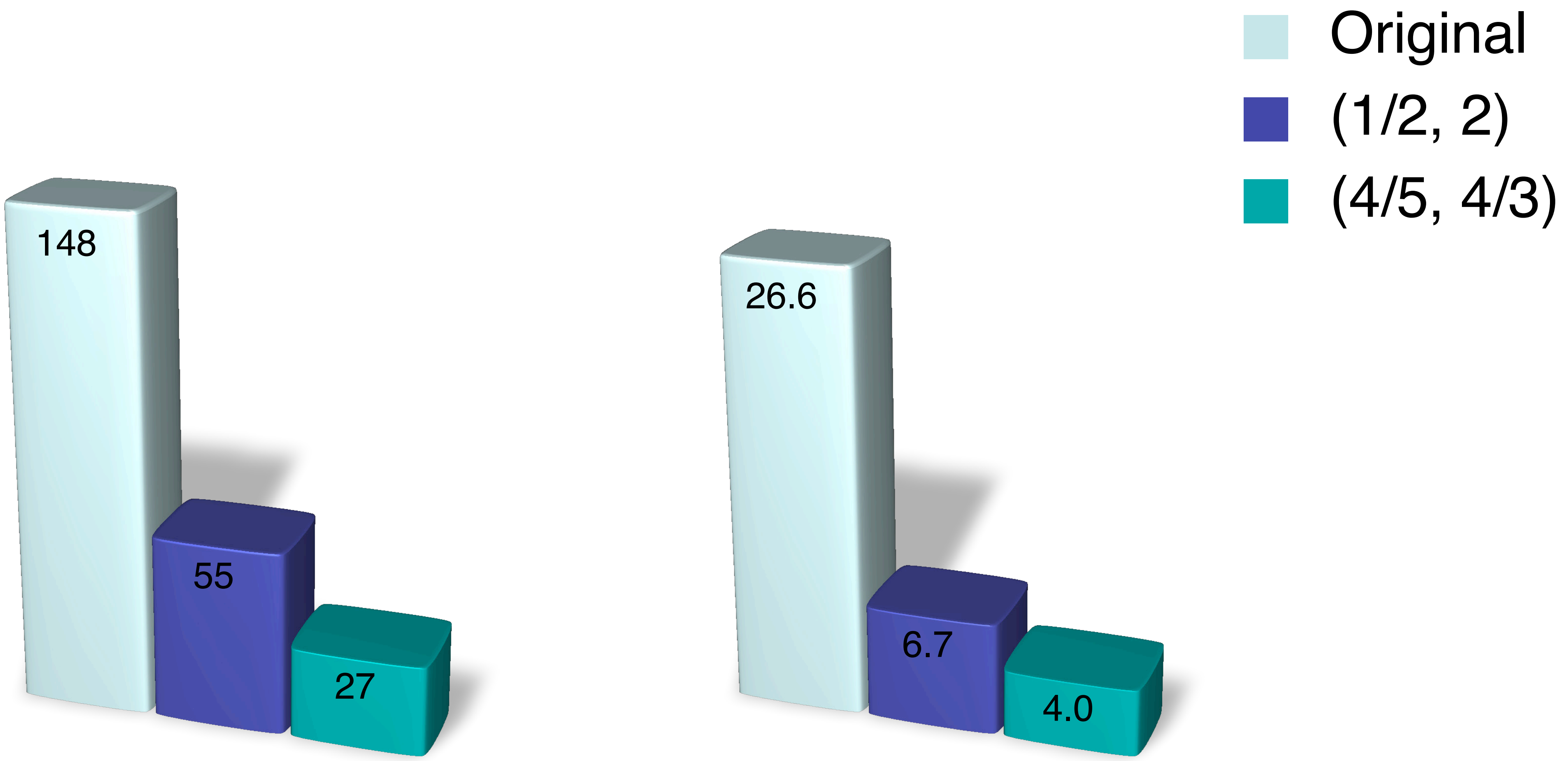


$\left(\frac{1}{2}, 2\right)$



$\left(\frac{4}{5}, \frac{4}{3}\right)$

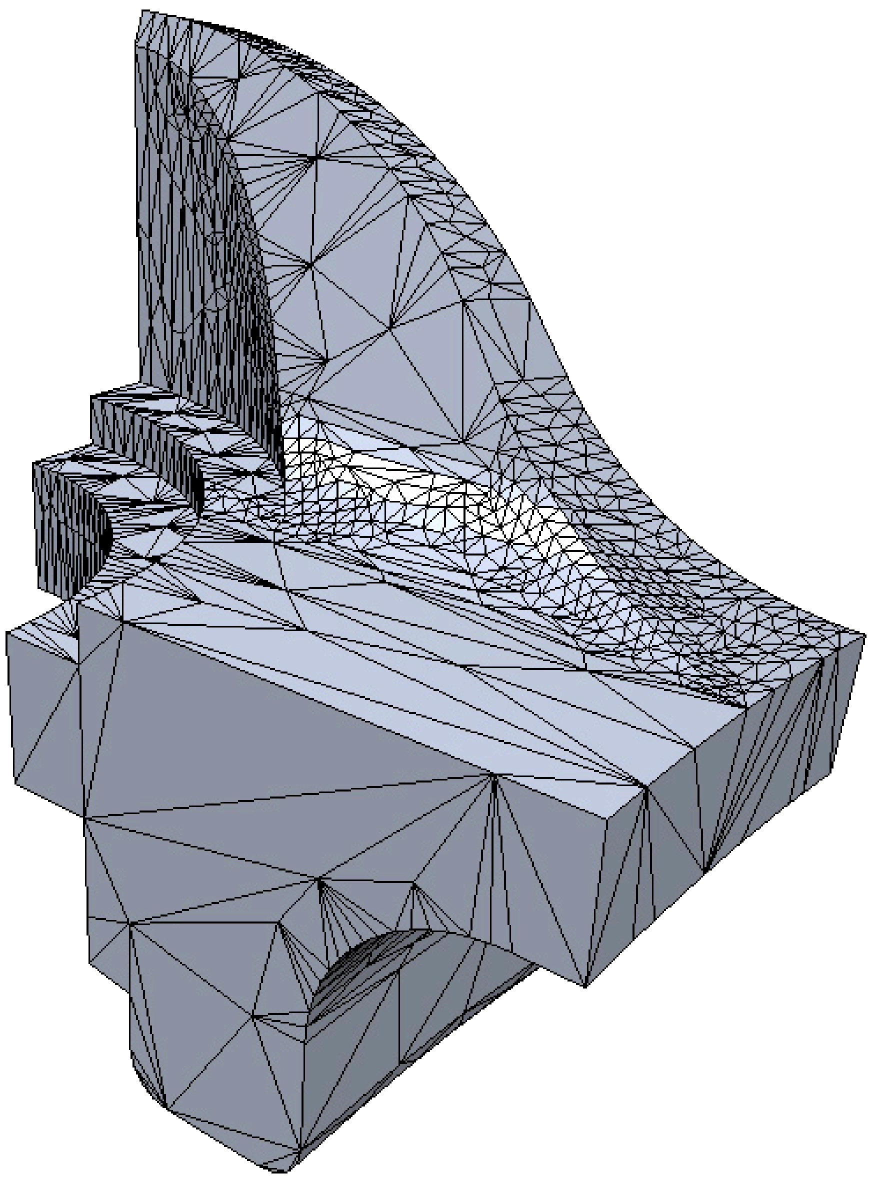
Remeshing Results

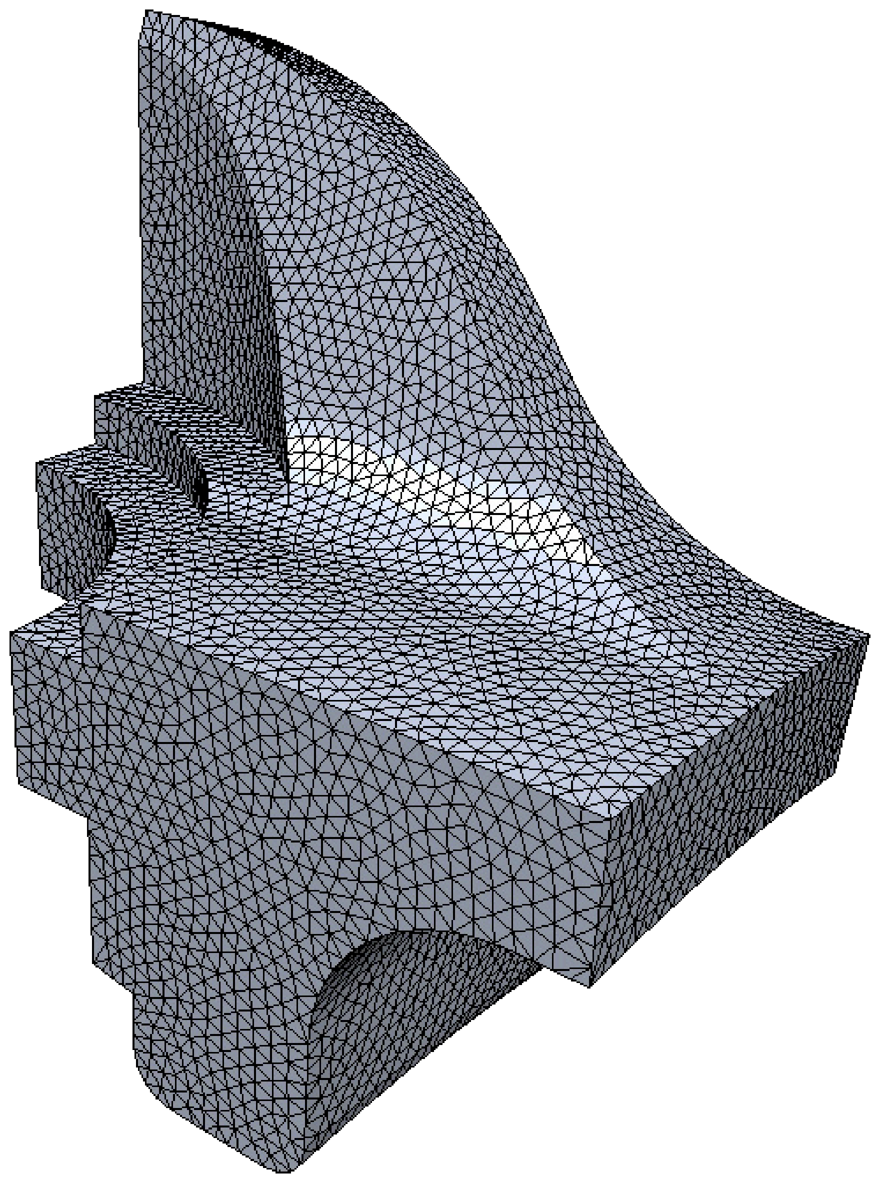


Edge length deviation (%)

Deviation from 60º

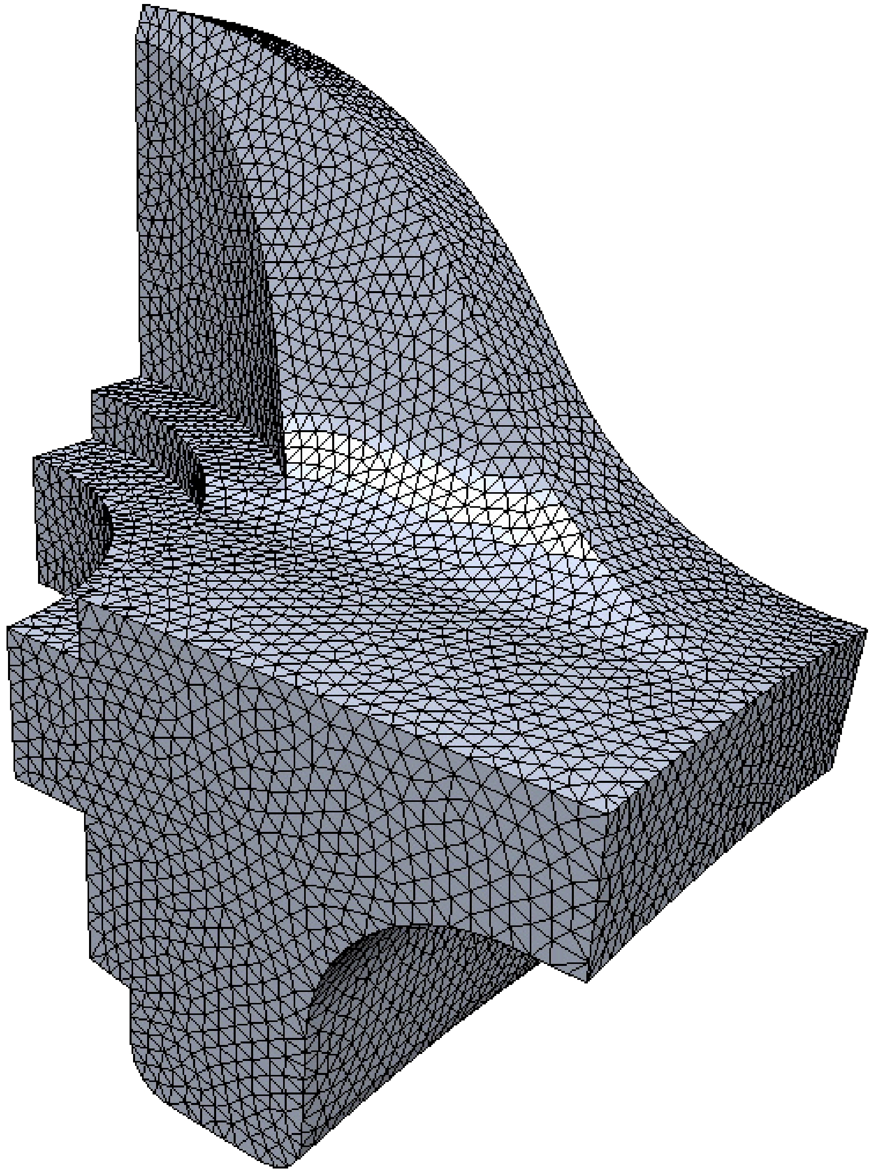
Feature Preservation?





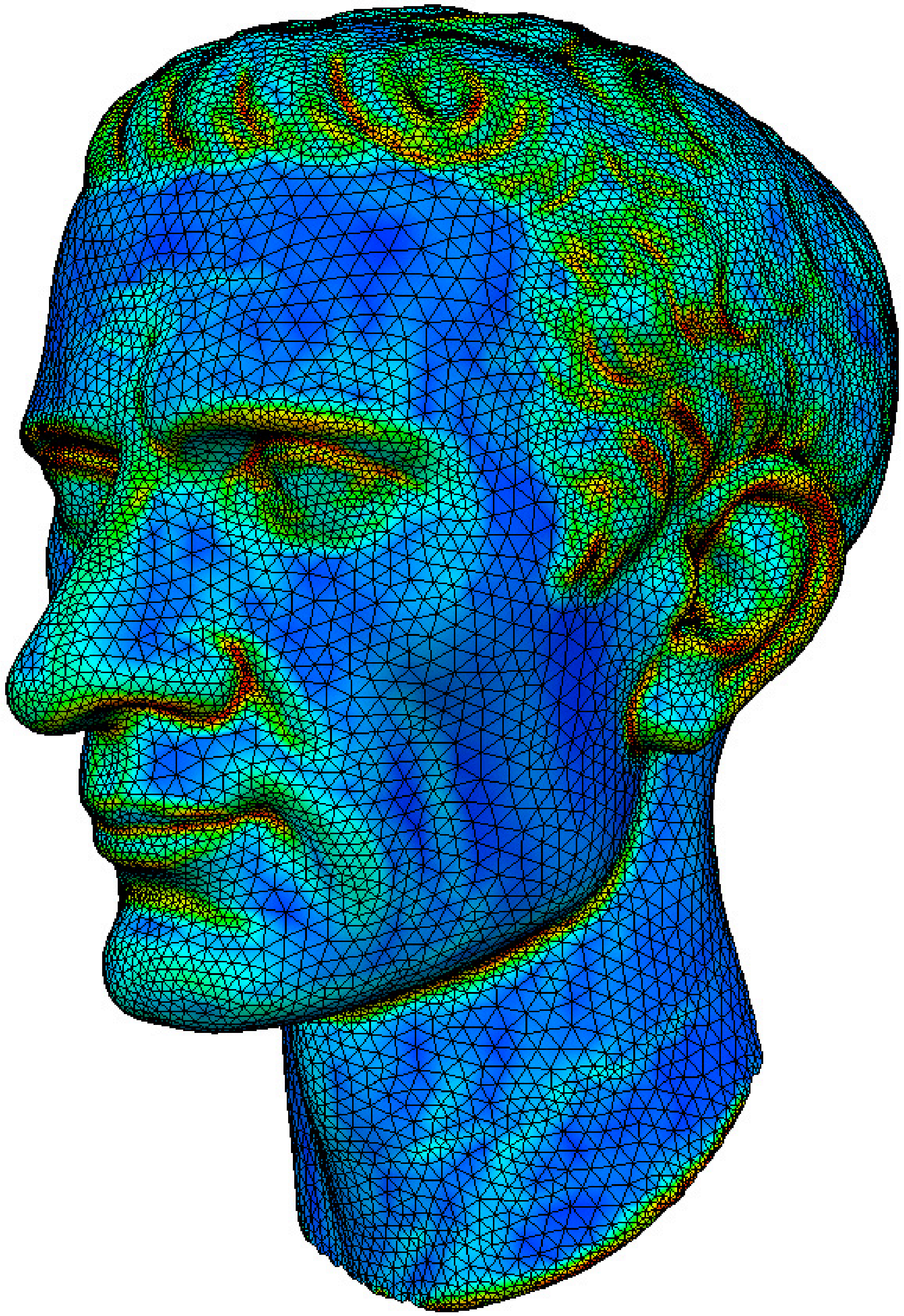
Feature Preservation

- Define features
  - Sharp edges
  - Material boundaries
- Adjust local operators
  - Don't flip
  - Collapse only along features
  - Univariate smoothing
  - Project to feature curves



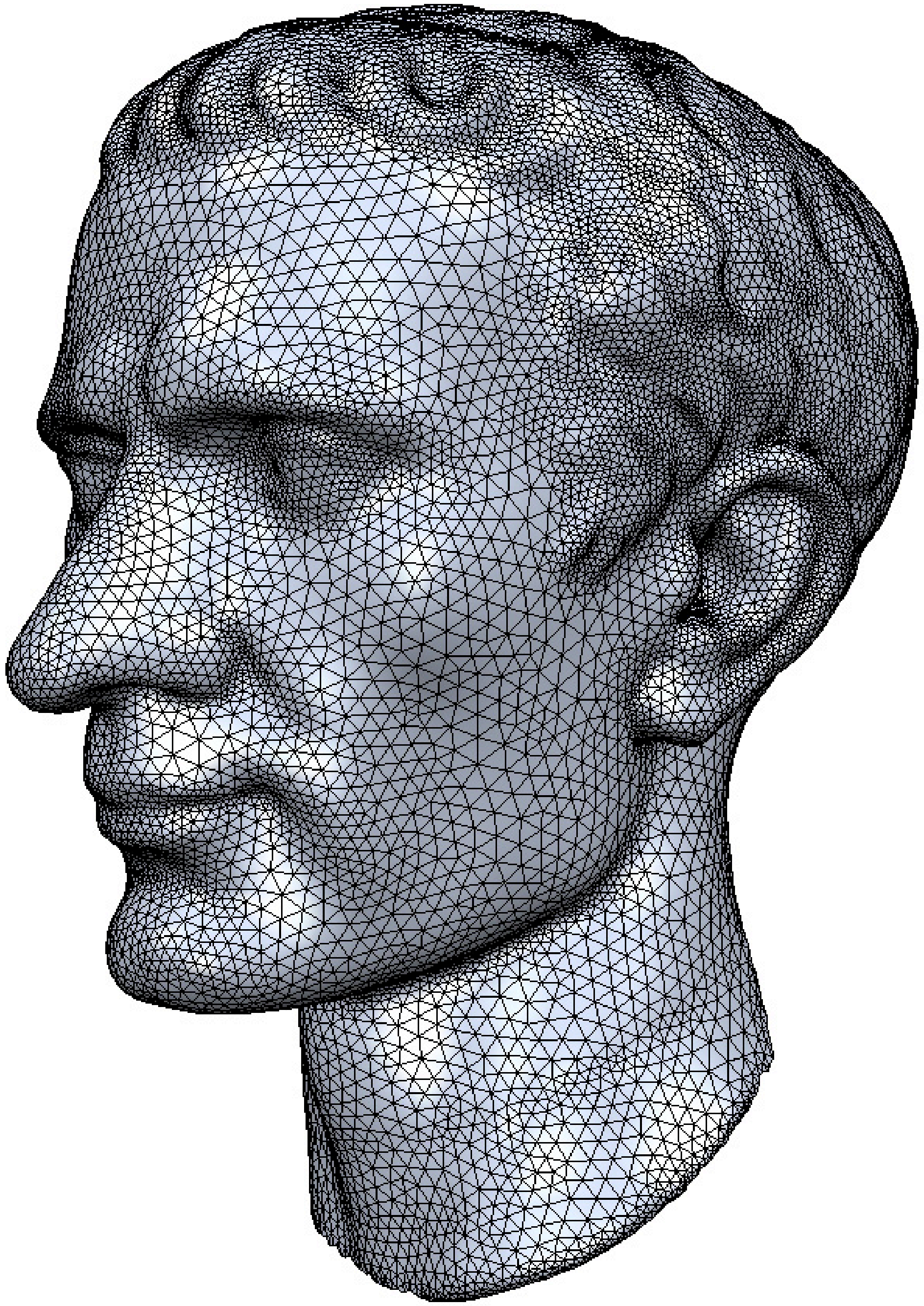
Adaptive Remeshing

- Precompute max. curvature on reference mesh
- Target edge length locally determined by curvature
- Adjust split / collapse criteria



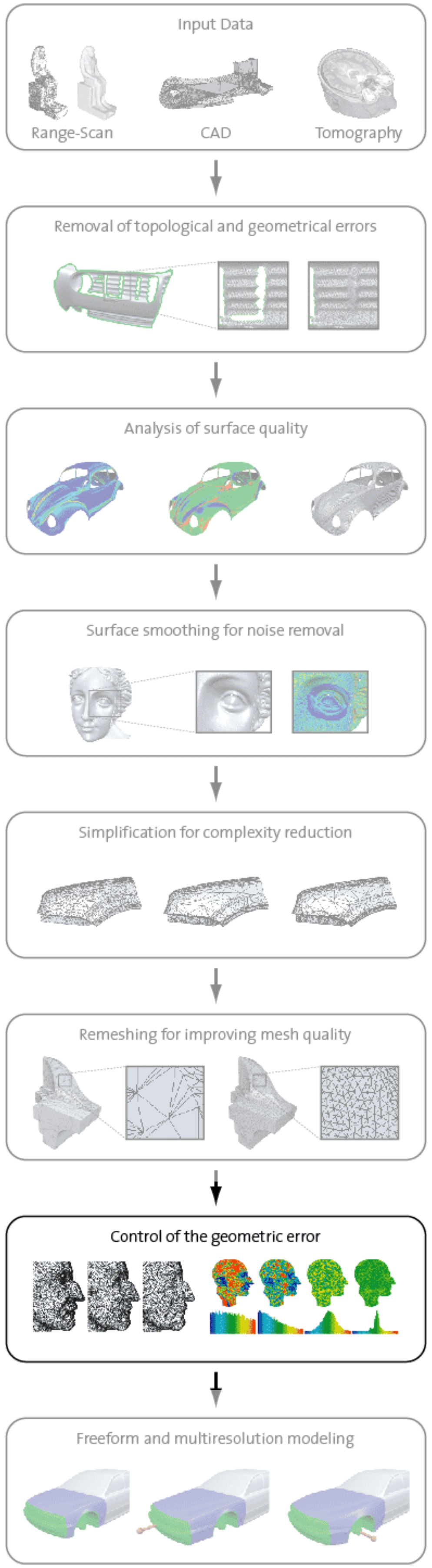
Isotropic Remeshing

- High quality triangulations
  - Equilateral triangles
  - Valence 6
- Extensions
  - Feature preservation
  - Curvature adaptation
- Local operators & projection
  - Easy to implement
  - Computationally efficient



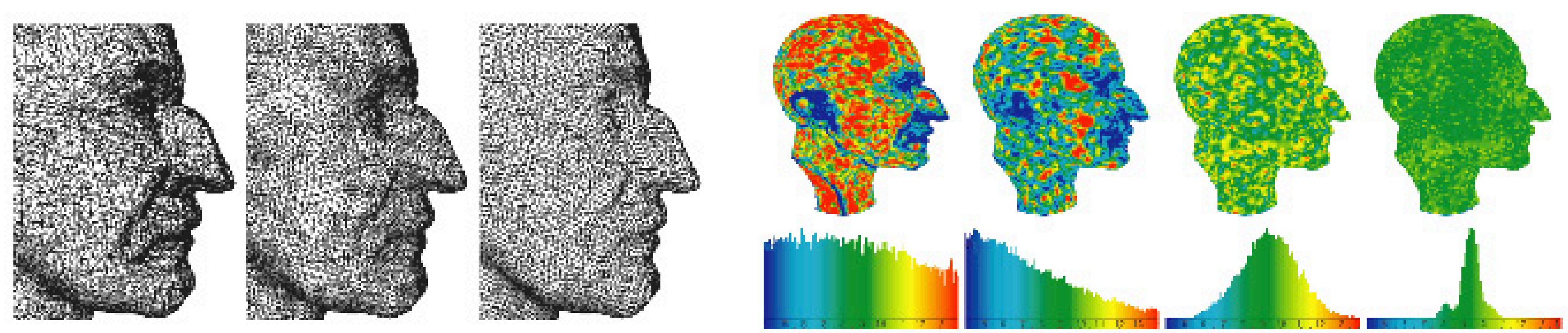
Literature

- Vorsatz et al, *"Dynamic remeshing and applications"*, Solid Modeling 2003
- Botsch & Kobbelt, *"A remeshing approach to multiresolution modeling"*, Symp. on Geometry Processing 2004
- Alliez et al, *"Recent advances in remeshing of surfaces"*, AIM@Shape state of the art report, 2006

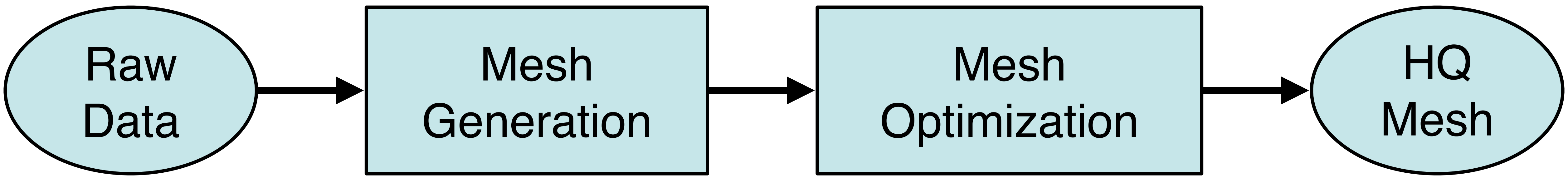


Global Error Control

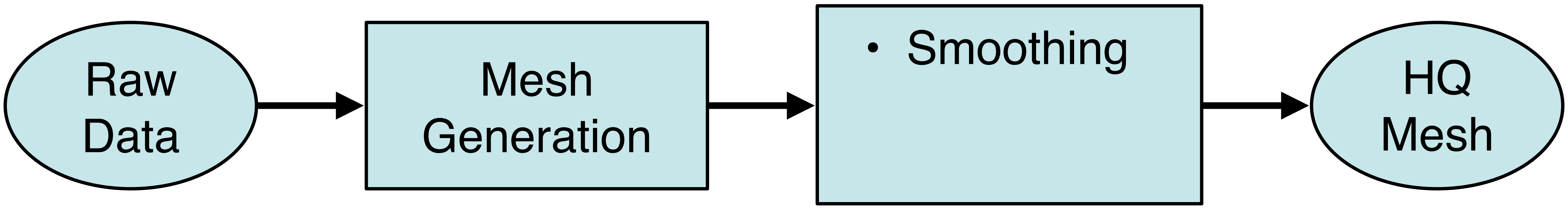
Control of the geometric error

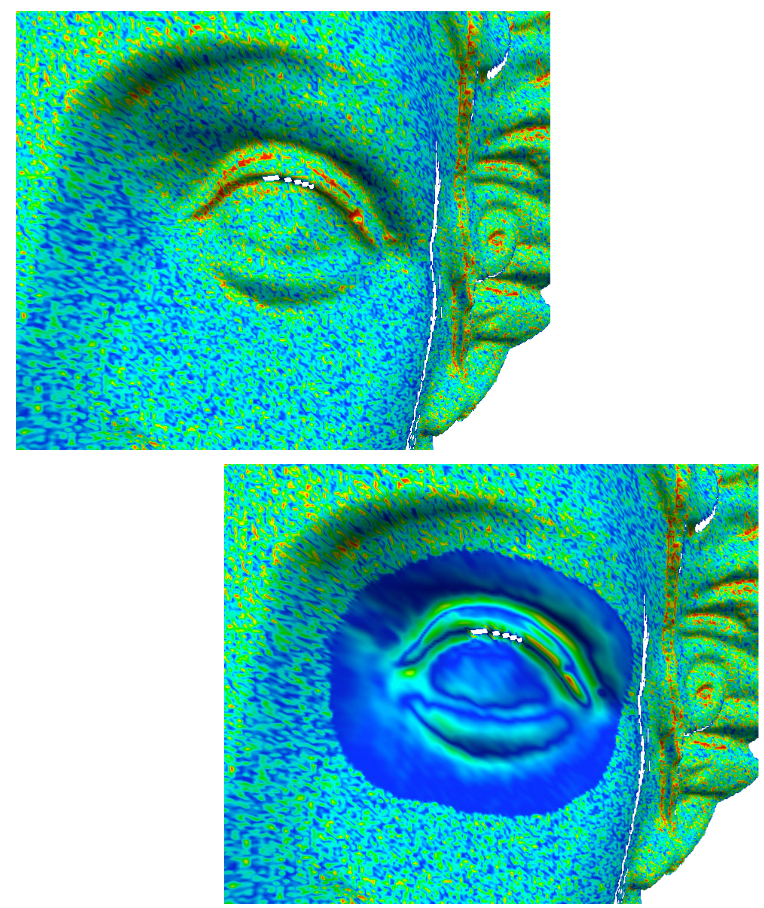


Global Error Control

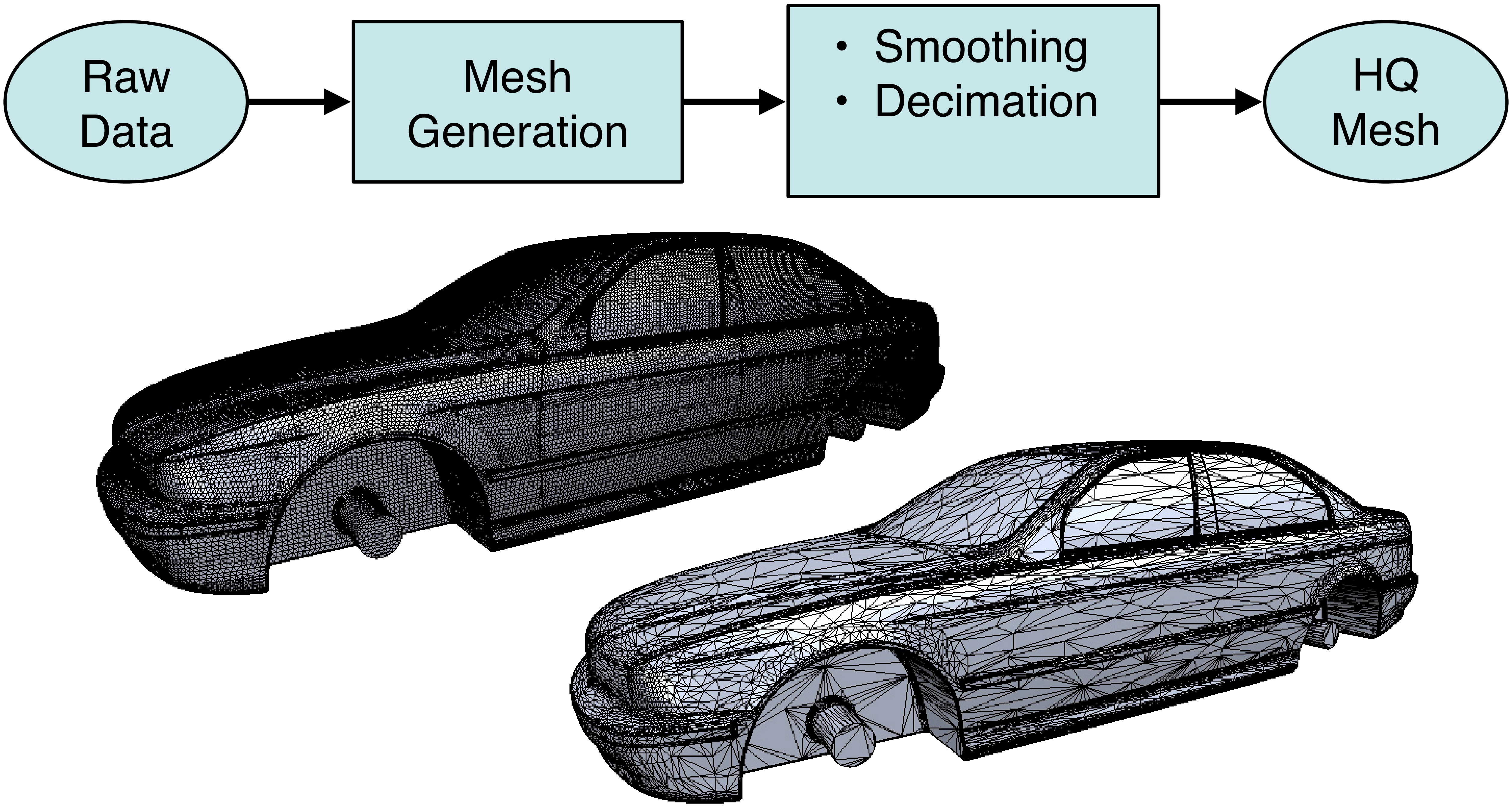


Global Error Control

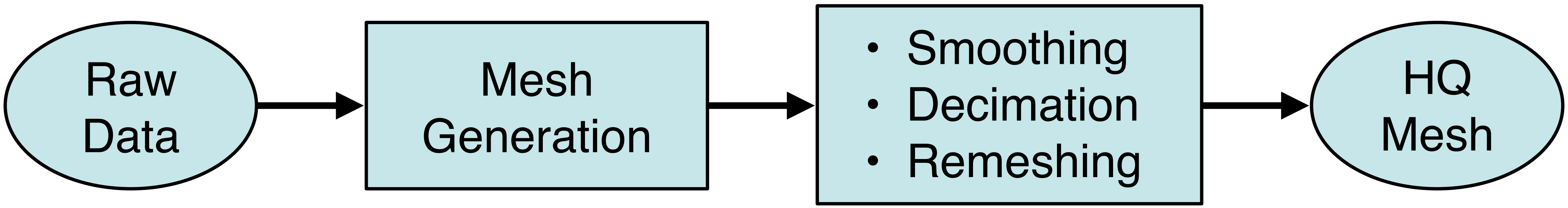


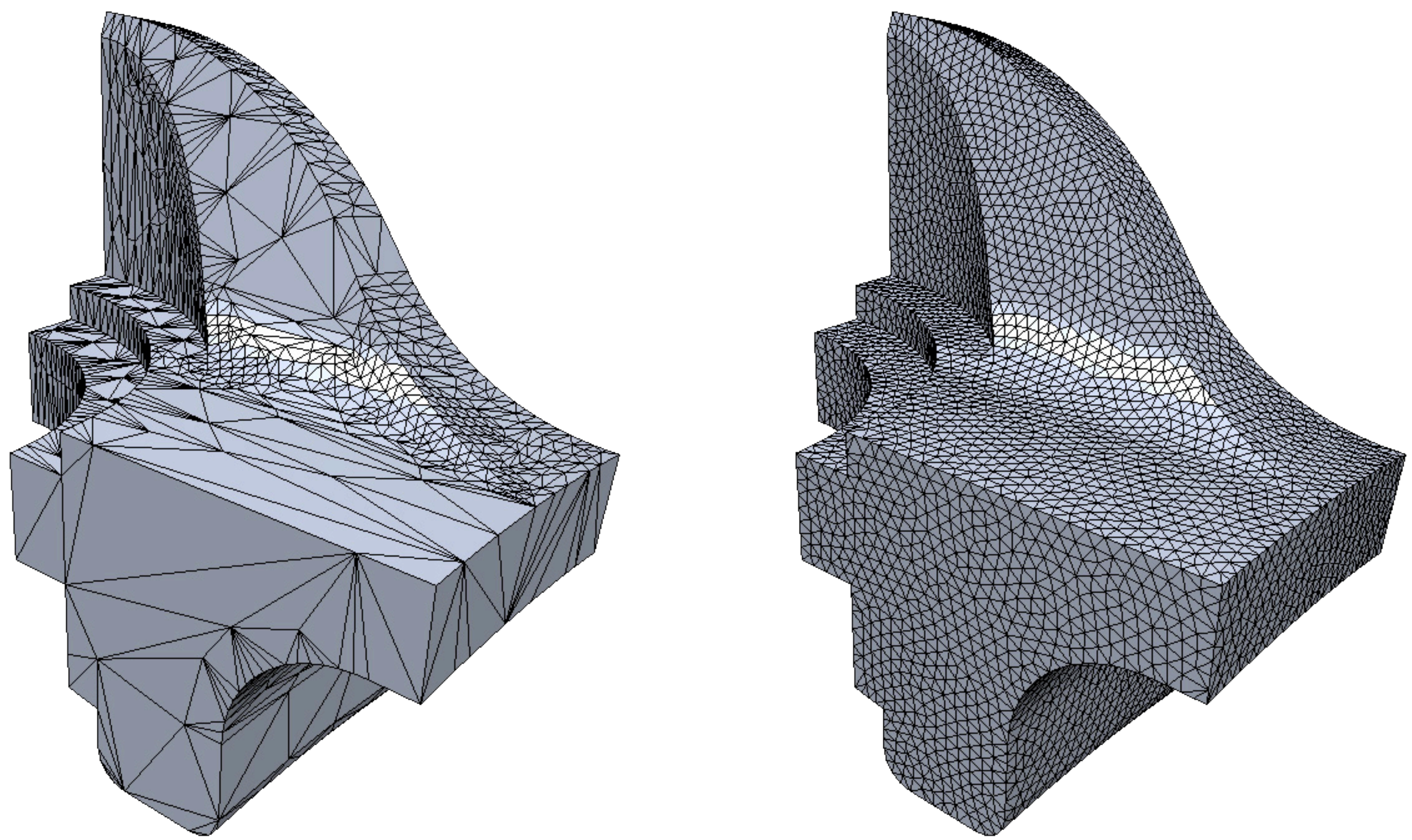


Global Error Control

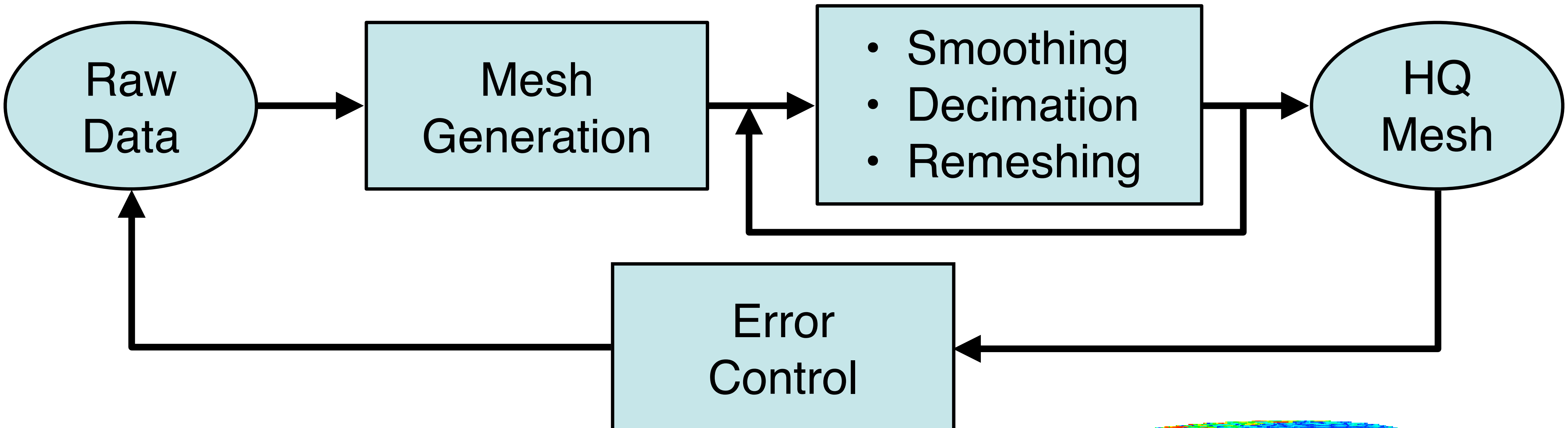


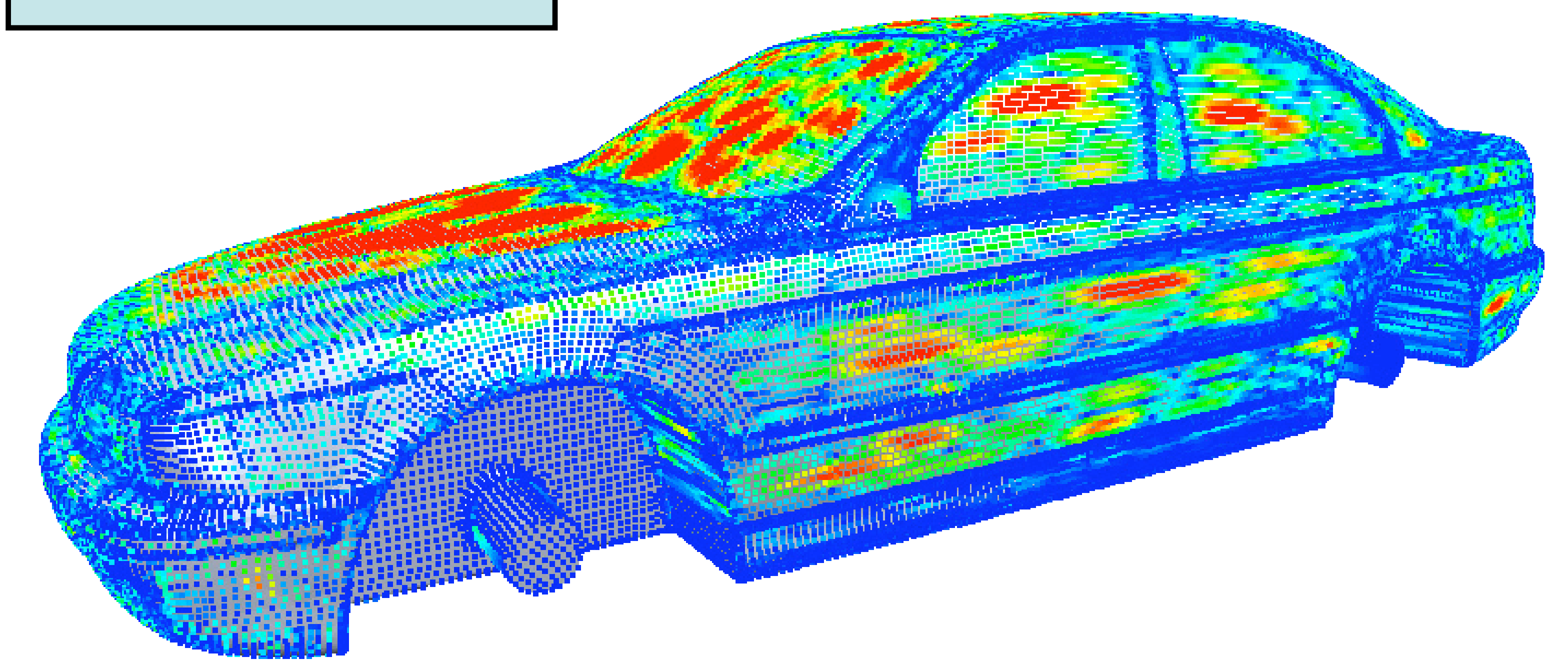
Global Error Control



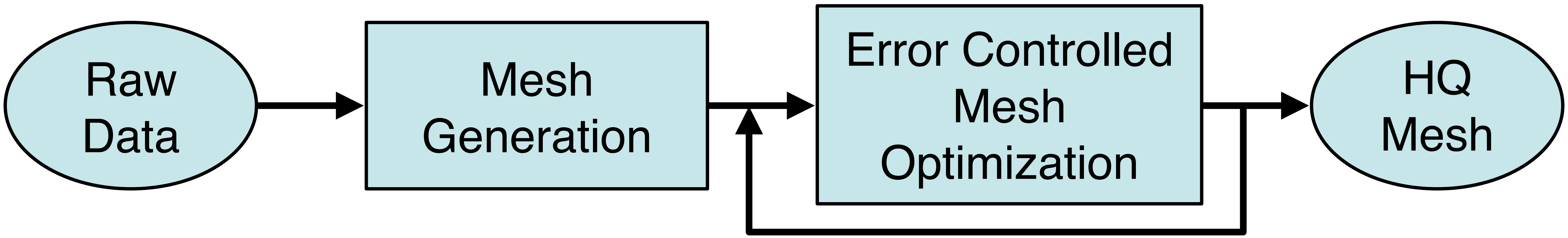


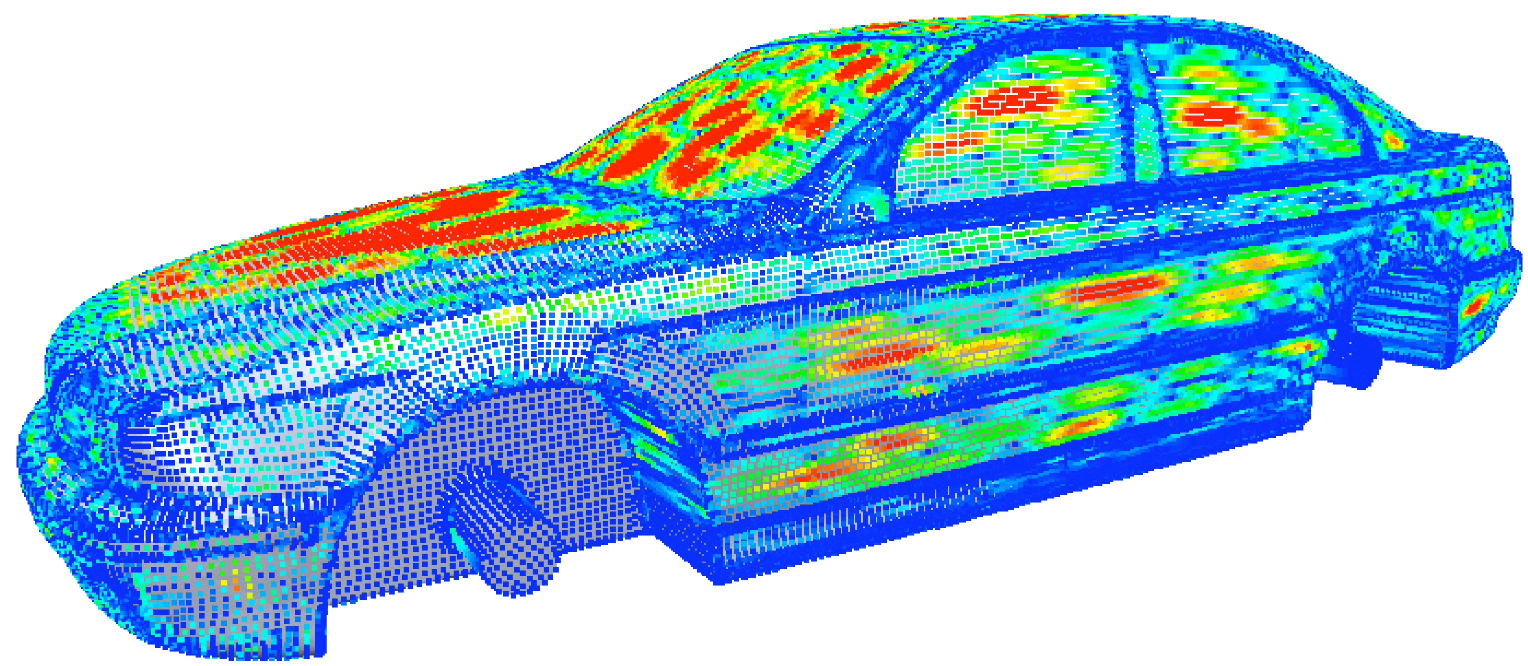
Global Error Control





Global Error Control



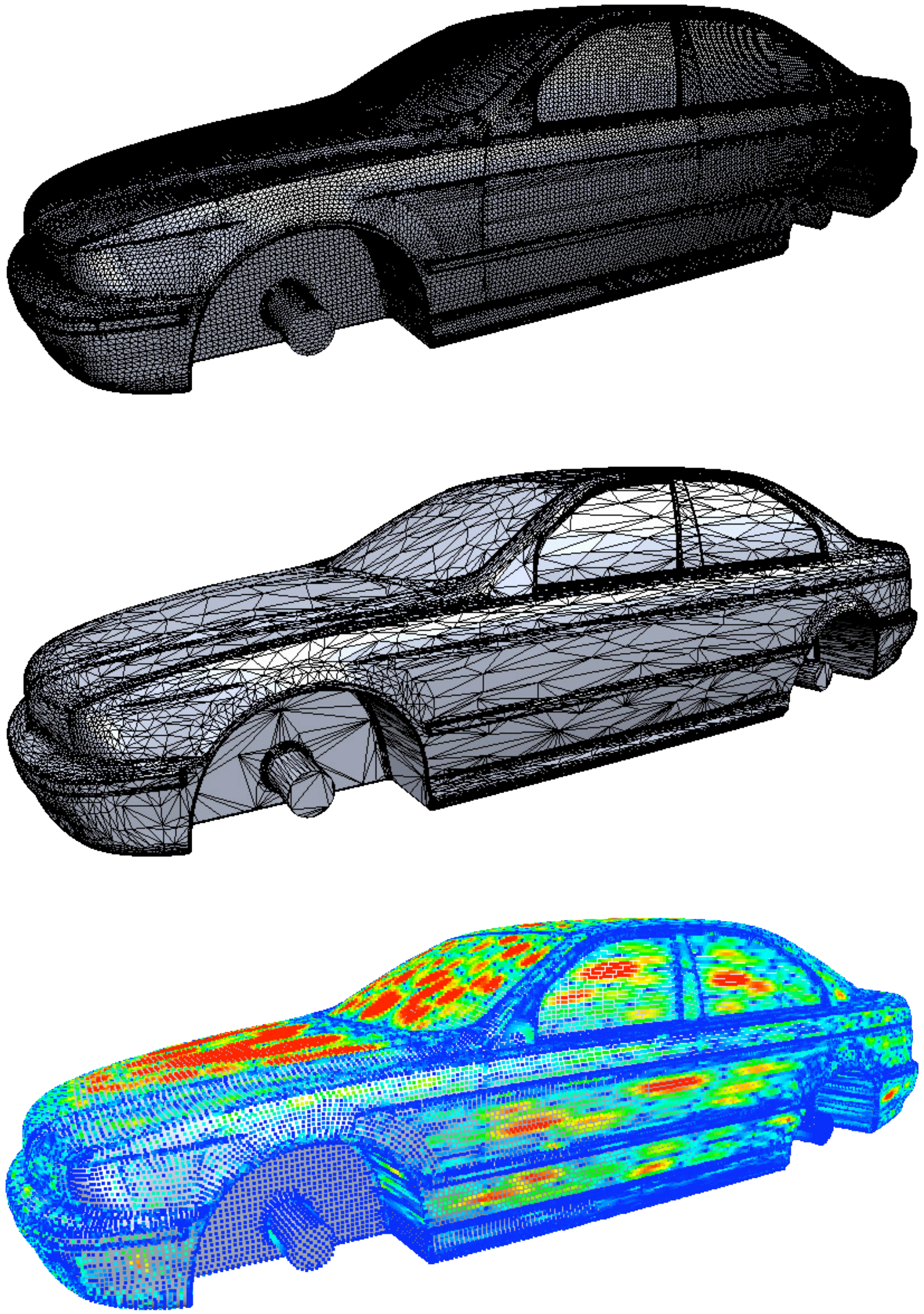


Global Error Control

- Explicit
  - One- or two-sided Hausdorff distance
  - A-posteriori error check
- Implicit
  - Tolerance volumes
  - A-priori error control

Hausdorff Error

- One- or two sided distance?
  - Usually one-sided is sufficient
- Post-processing
  - Both meshes are static
  - Precompute BSP
- A-priori error control
  - Possible for decimation
  - Too complex in general



Literature

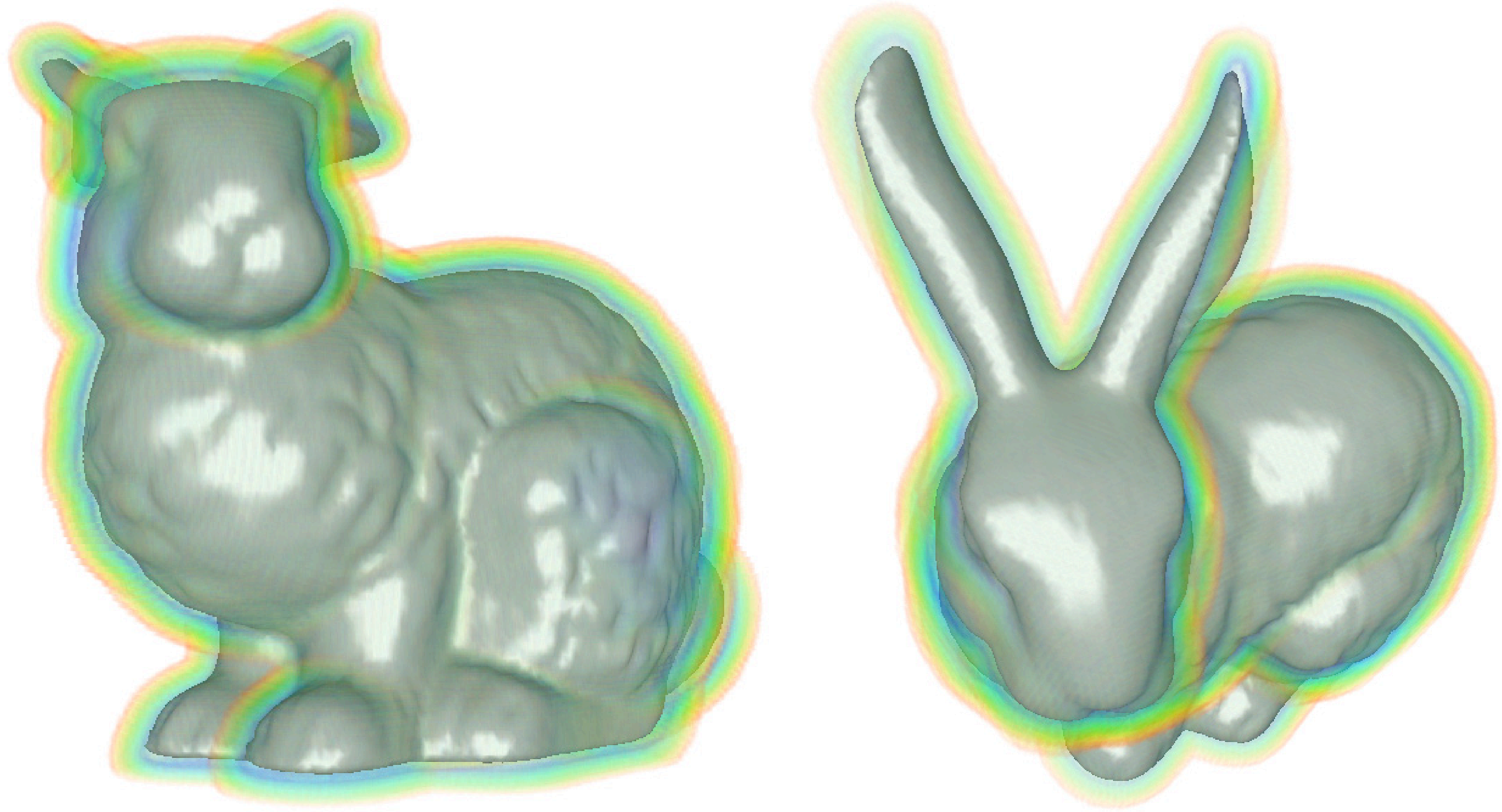
- Cignoni, *“Metro: measuring error on simplified surfaces”*, Computer Graphics Forum 17(2), 1998
- Kobbelt et al, *“A general framework for mesh decimation”*, Graphics Interface, 1998

Global Error Control

- Control global approximation error
  - Exact (or conservative)
- Each method may provide error control
  - Local errors may accumulate
- Need general global error control
  - Independent of mesh algorithm!

Tolerance Volumes

- Tolerance volume around reference mesh
  - Triangles have to stay within it

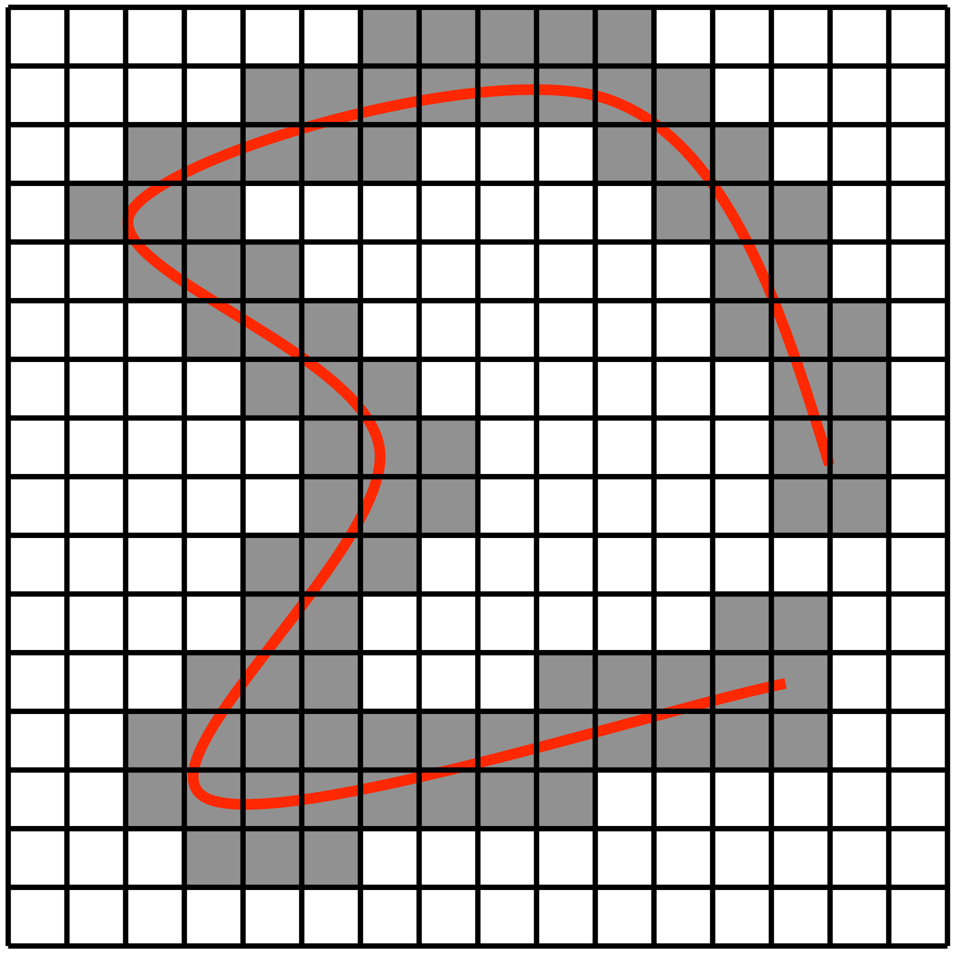


Tolerance Volumes

- Tolerance volume around reference mesh
  - Triangles have to stay within it
- General distance query
  - Implicit representation best suited
  - Approximate signed distance field
- Check each modified triangle
  - Find SDF maximum over triangle
- How to approximate SDF ?

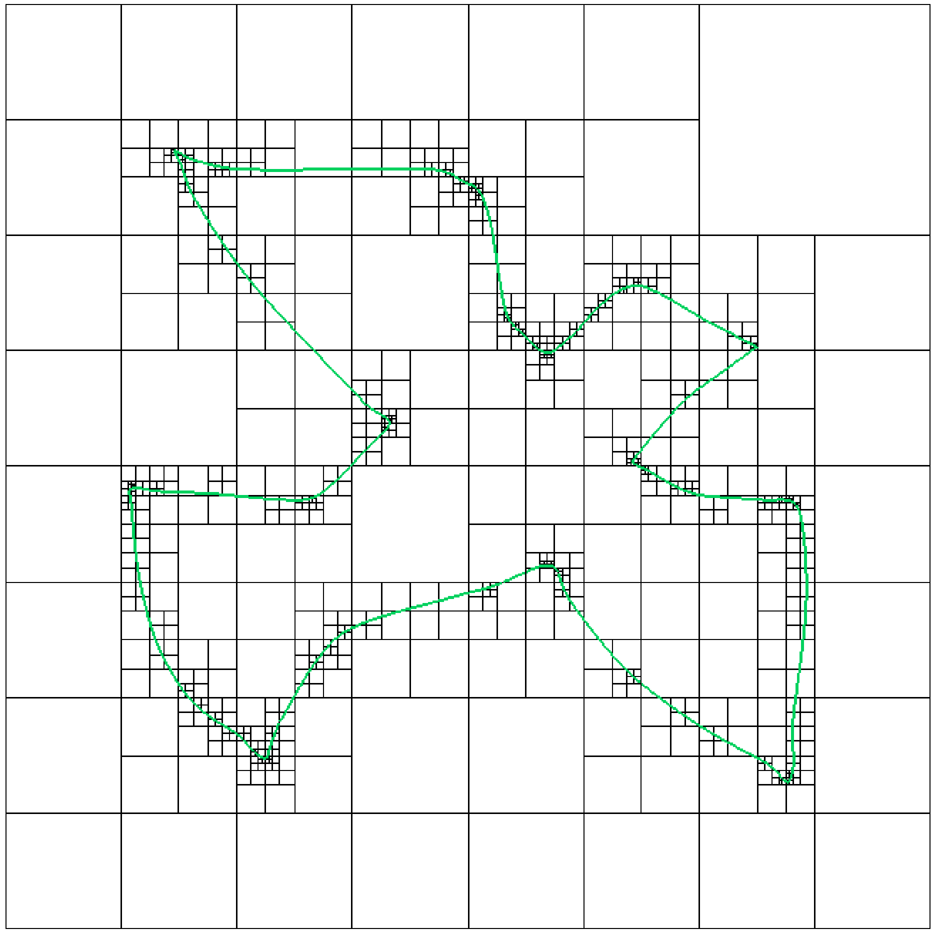
SDF Approximations

- Piecewise constant, C^{-1}, regular grid
  - Permission Grids [Zelinka & Garland]
  - Simple triangle test
  - Needs high grid resolution



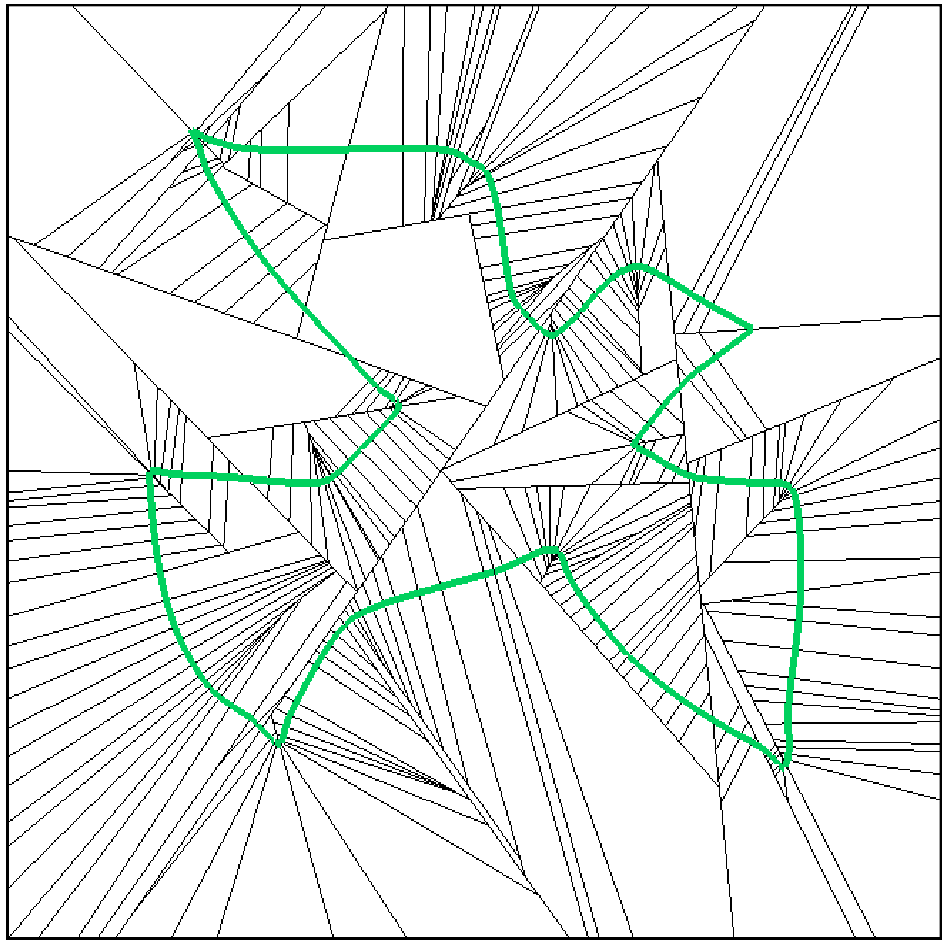
SDF Approximation

- Piecewise tri-linear, C^0, adaptive octree
  - Adaptively sampled SDFs [Frisken et al.]
  - Low memory consumption
  - Complicated triangle test (piecewise cubic function)



SDF Approximation

- Piecewise linear, C^{-1}, BSP tree
  - Linear approximation [Wu & Kobbelt]
  - Low memory consumption
  - Complicated triangle test (split triangles to BSP leaves)

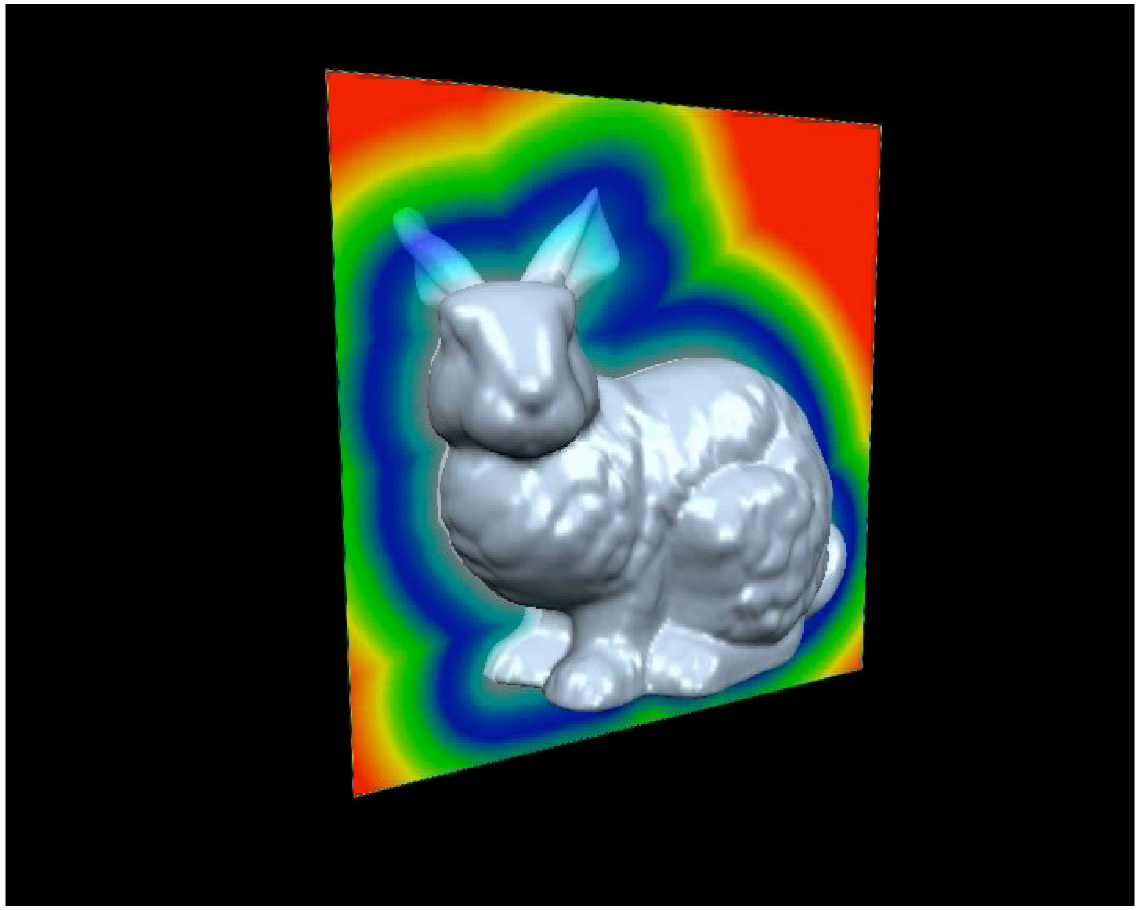


SDF Approximation

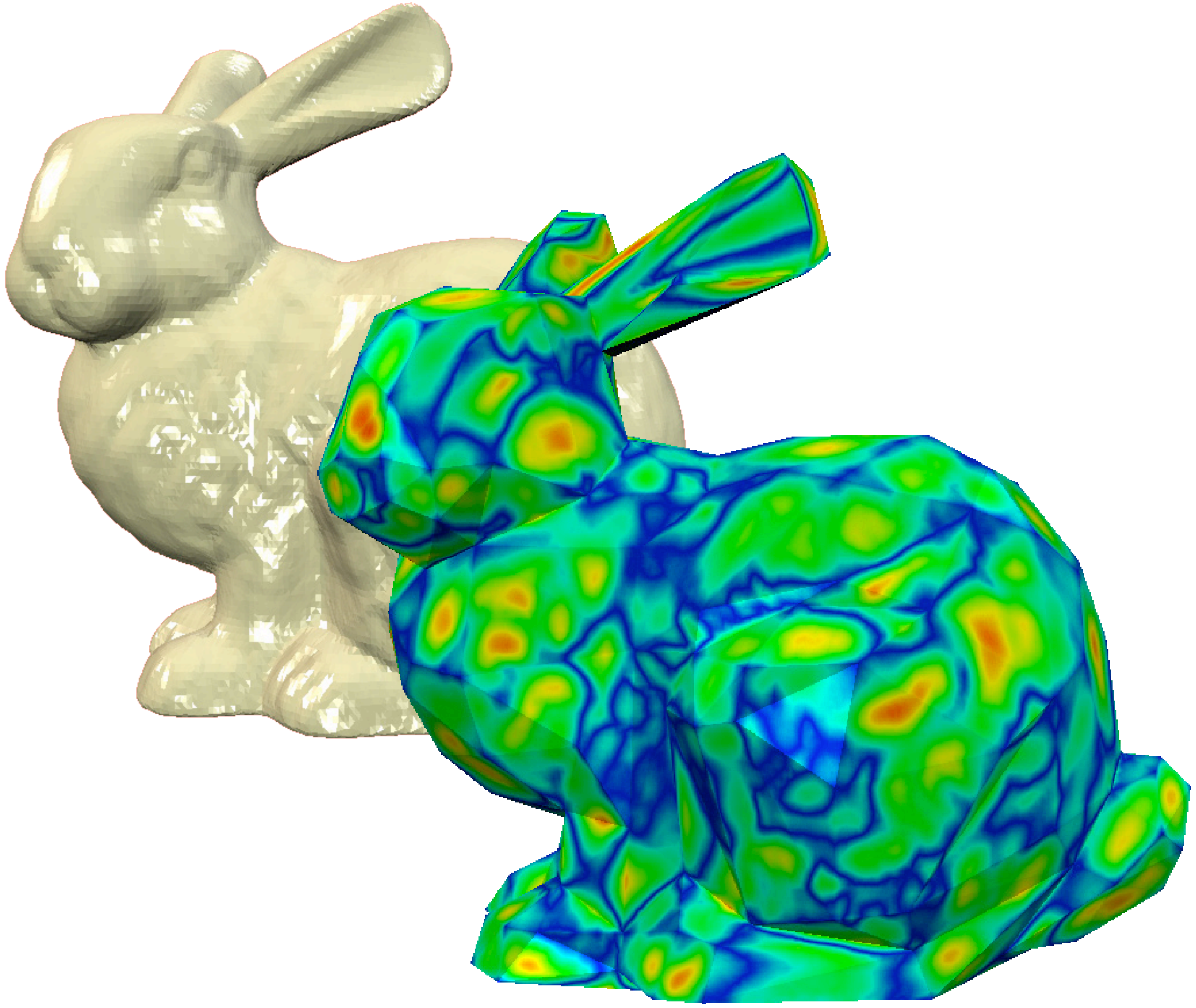
- Piecewise tri-linear, C^0, regular grid
  - 3D texture [Botsch et al 2004]
  - Medium memory consumption (regular grid, linear approximation)
  - Map to graphics card (GPU)

GPU-Based Tolerance Volumes

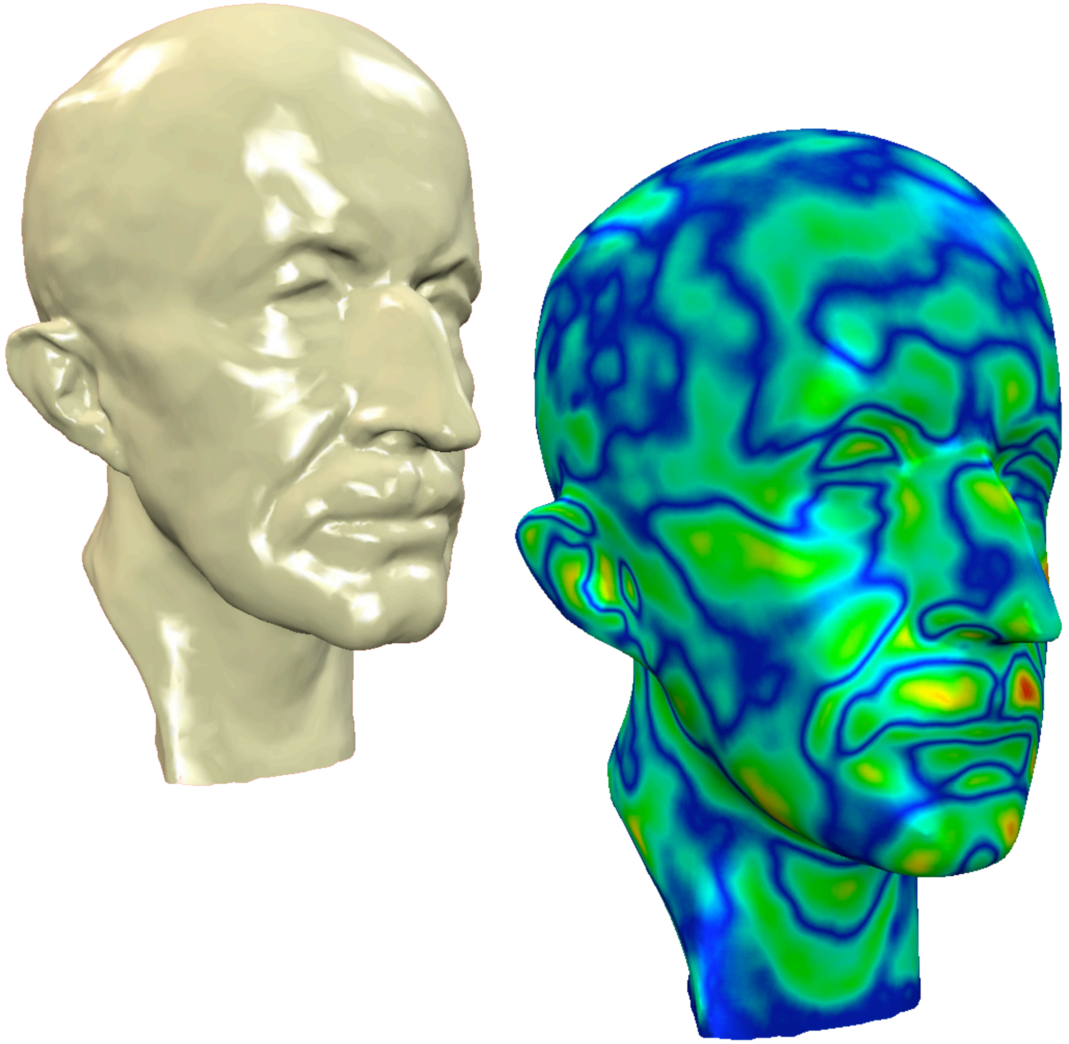
- Represent SDF as 3D texture
- Triangle test: Just render it!
  - Automatic voxelization
  - Automatic tri-linear interpolation
- GPUs are efficient
  - Real-time error control



Error Control & Visualization



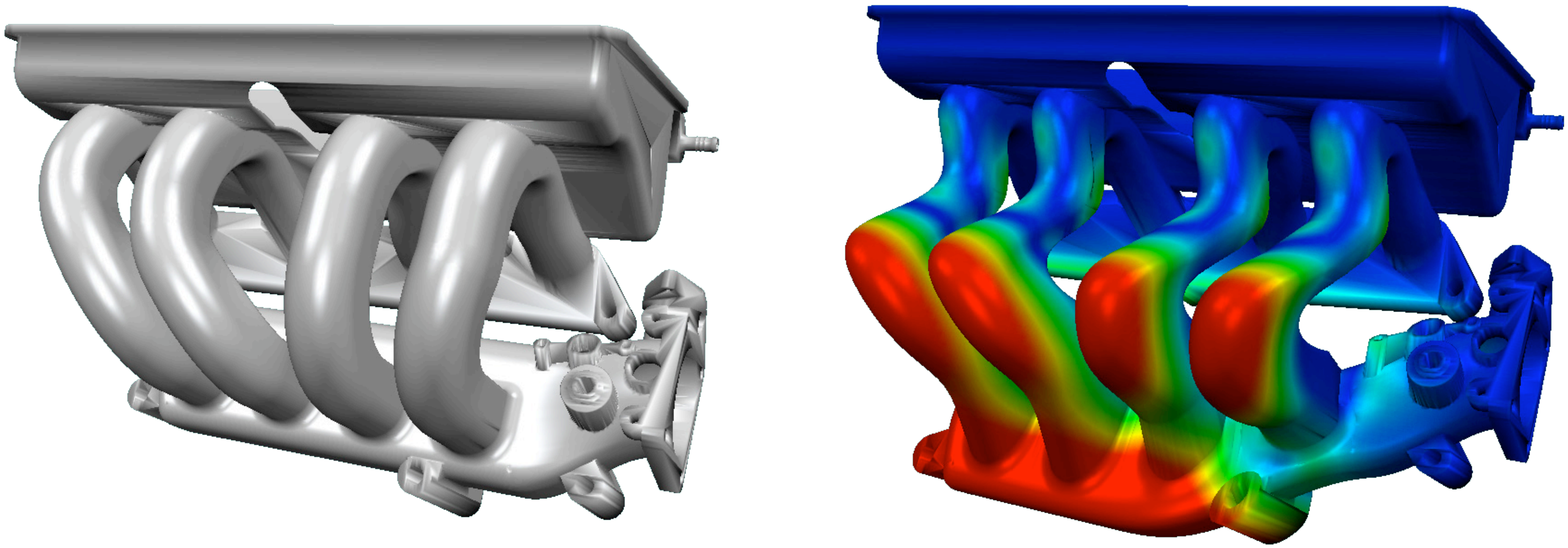
Decimation



Smoothing

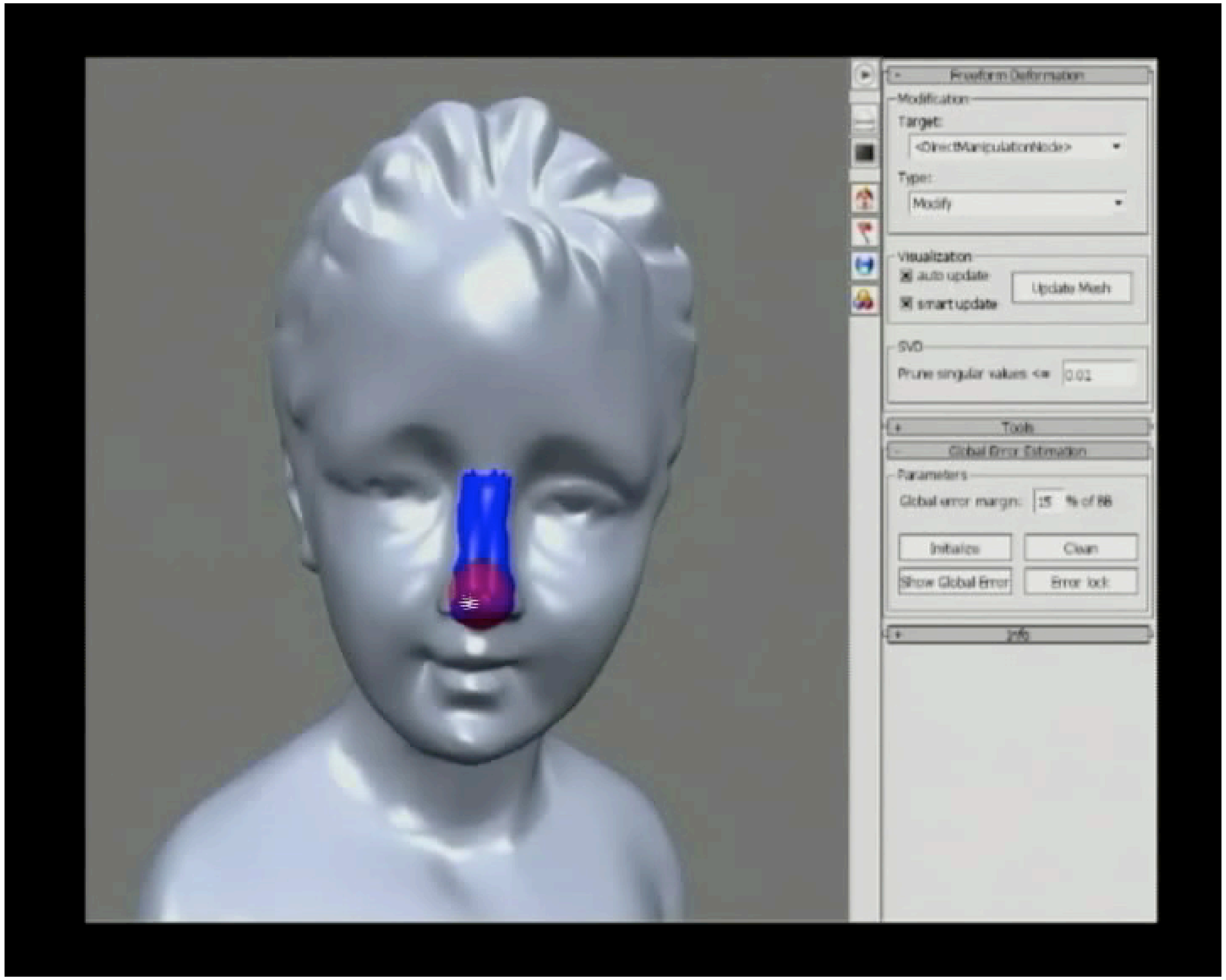
Surface Deformation

- Exact deformation



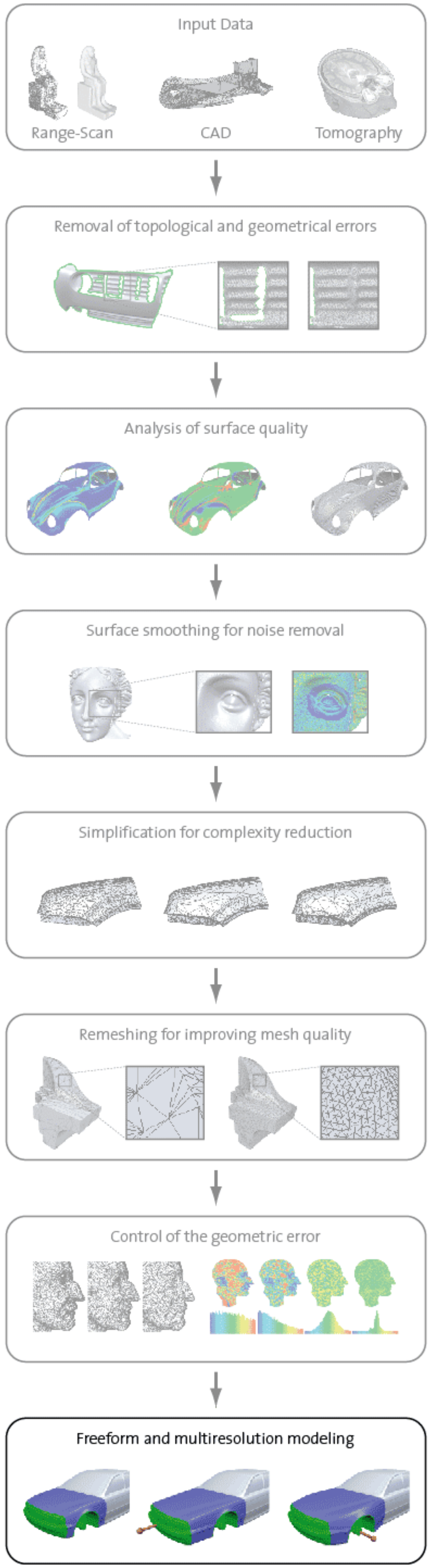
Surface Deformation

- Exact deformation
- Real-time feedback



Literature

- Zelinka & Garland, *"Permission Grids: Practical, Error-Bounded Simplification"*, ACM Trans. on Graphics 21 (2), 2002
- Botsch et al, *"GPU-based tolerance volumes for mesh processing"*, Pacific Graphics, 2004
- Wu & Kobbelt, *"Piecewise Linear Approximation of Signed Distance Fields"*, VMV 2003



Mesh Modeling

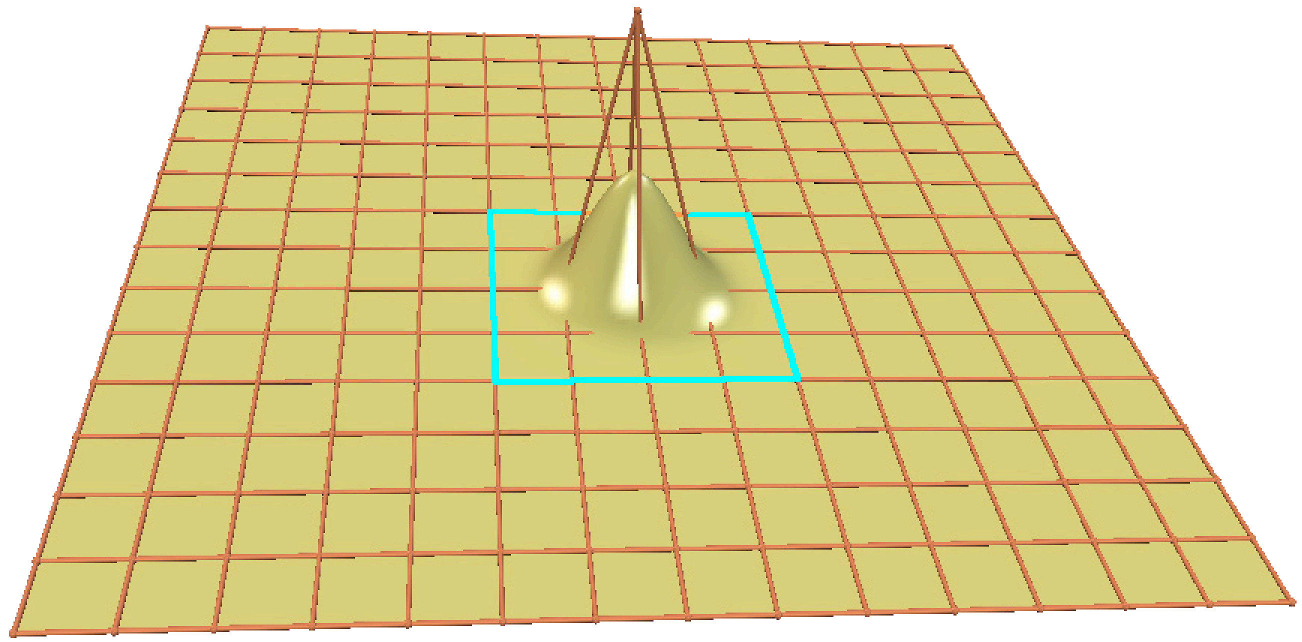
Freeform and multiresolution modeling

Shape Editing

- Surface-Based Deformation
  - Distance-Based Propagation
  - Boundary Constraint Modeling
- Space Deformation
  - Freeform Deformation
  - RBF Deformation
- Multiresolution Deformation

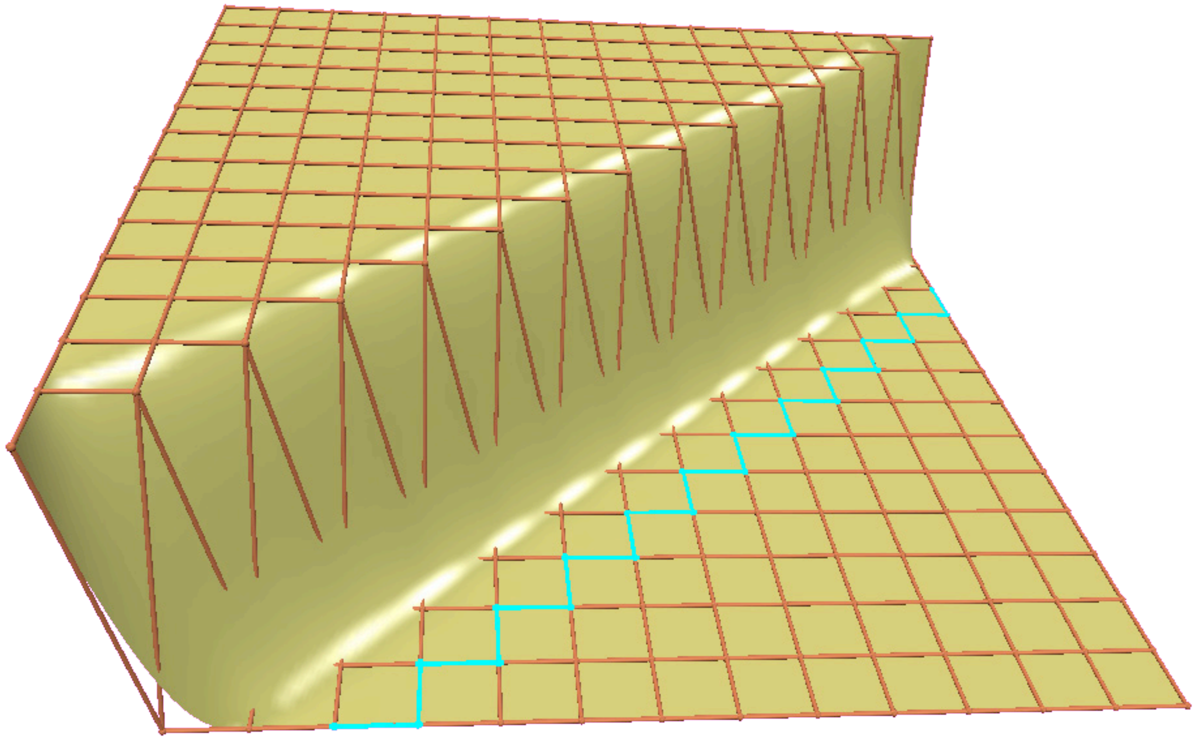
Spline Surfaces

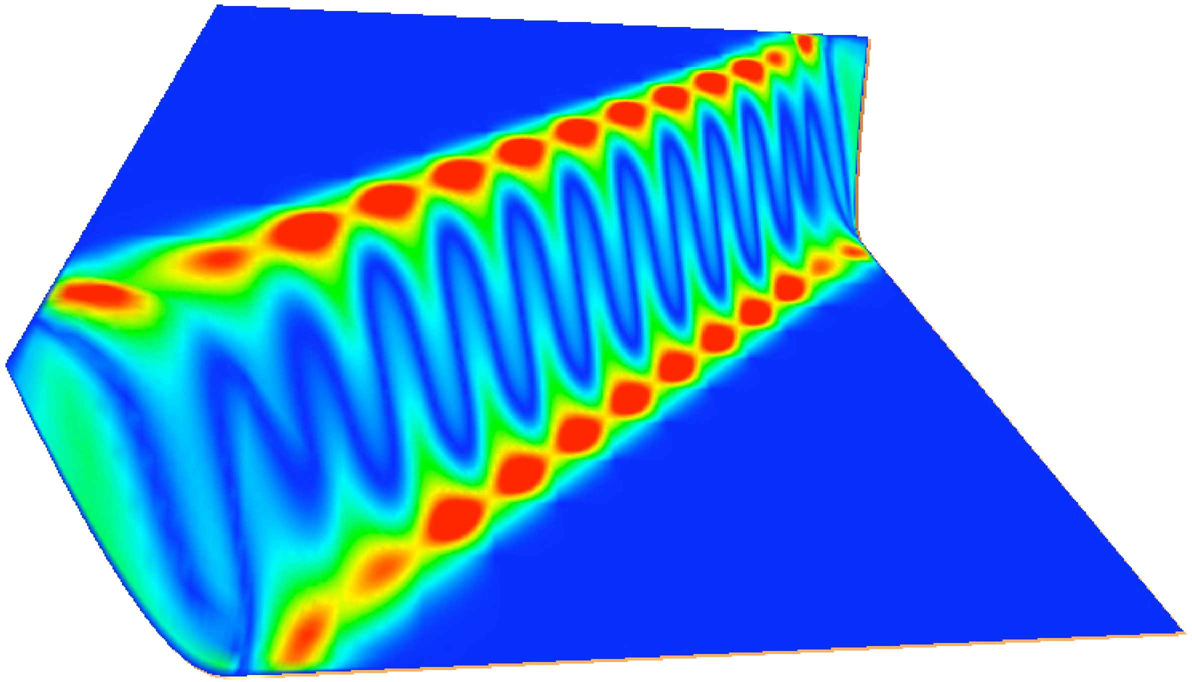
- Basis functions are smooth bumps



Spline Surfaces

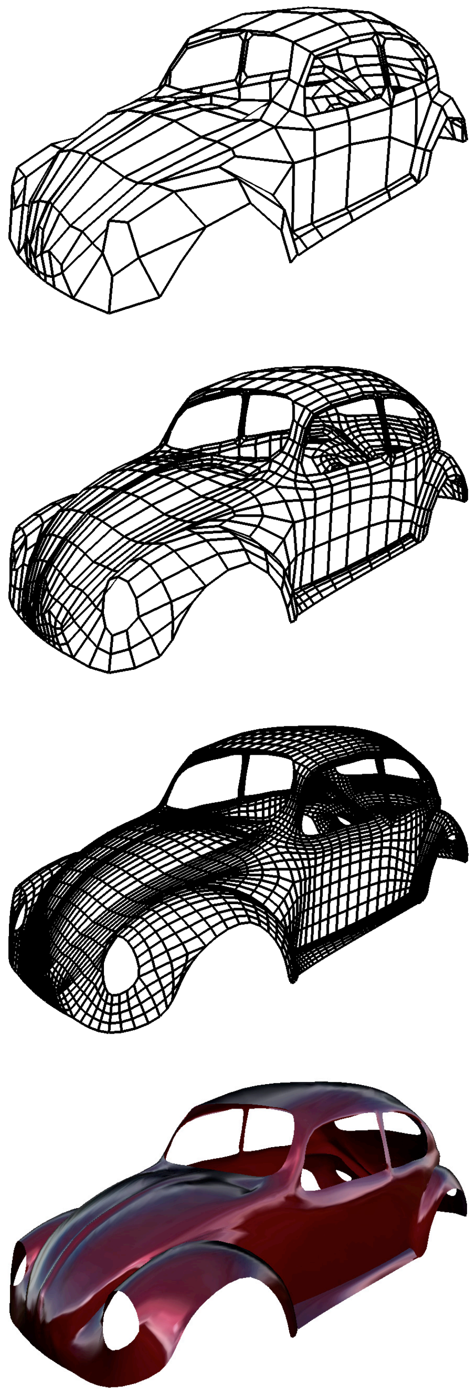
- Basis functions are smooth bumps
  - Fixed support
  - Regular grid





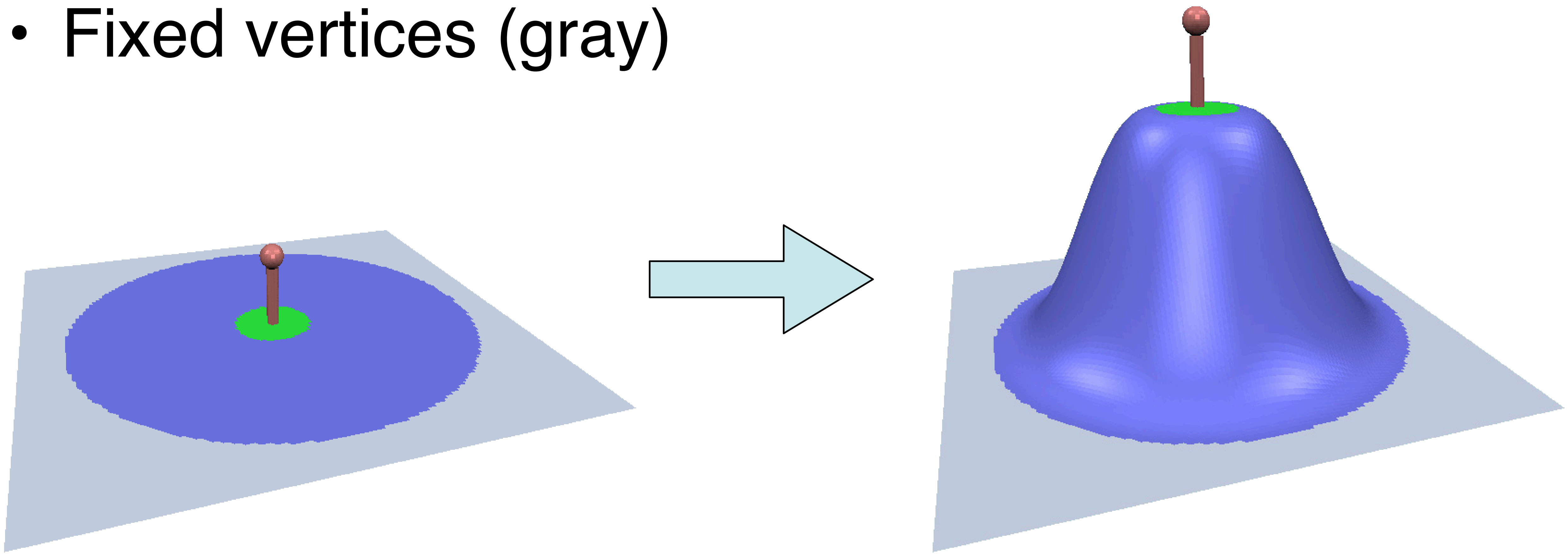
Spline & Subdivision Surfaces

- Basis functions are smooth bumps
  - Fixed support
  - Regular grid
- Bound to control points
  - Initial patch layout is crucial
  - Requires experts!
- Decouple deformation from surface representation!



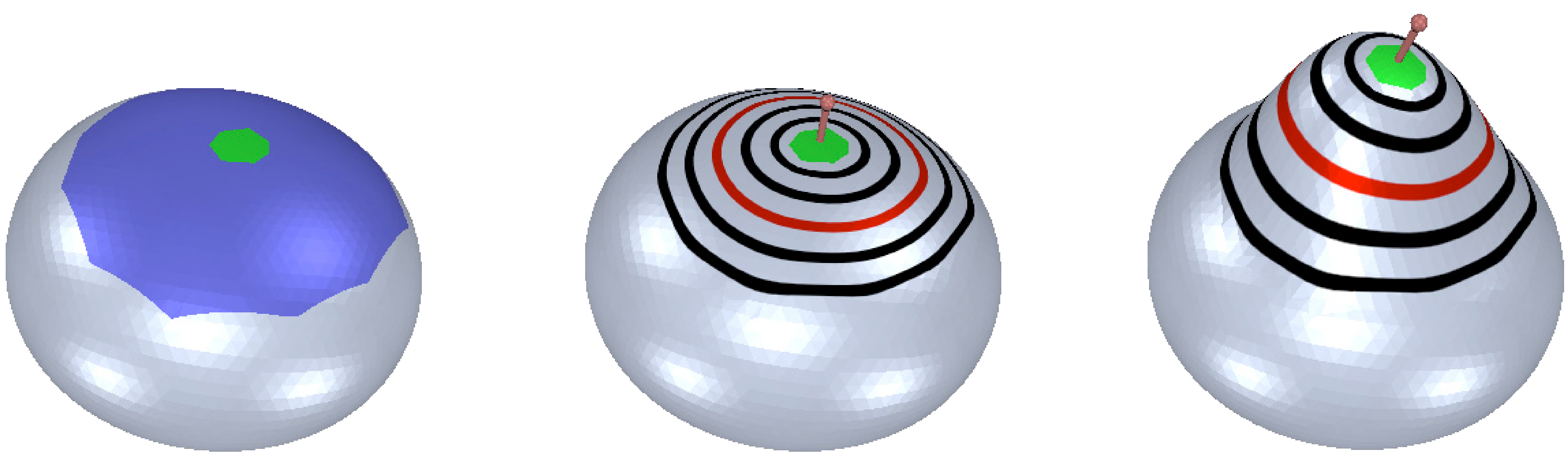
Modeling Metaphor

- Support region (blue)
- Handle regions (green)
- Fixed vertices (gray)



Distance-Based Propagation

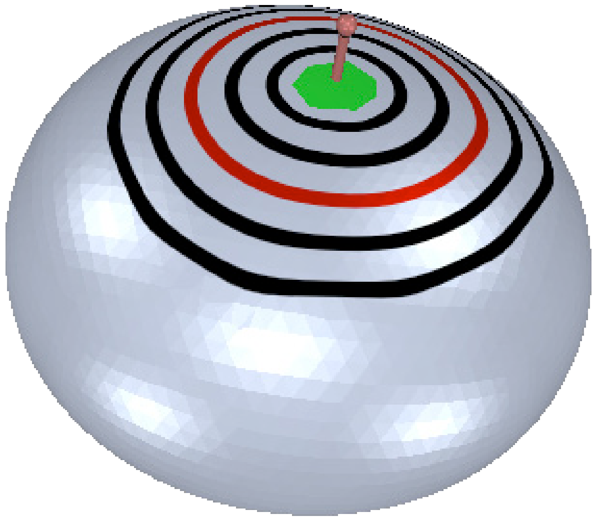
- Construct smooth scalar field [0,1]
  - s(x)=1: Full deformation (handle)
  - s(x)=0: No deformation (fixed part)
  - s(x)∈(0,1): Damp handle transformation (in between)



Distance-Based Propagation

- How to construct scalar field?
  - Euclidean/geodesic distance

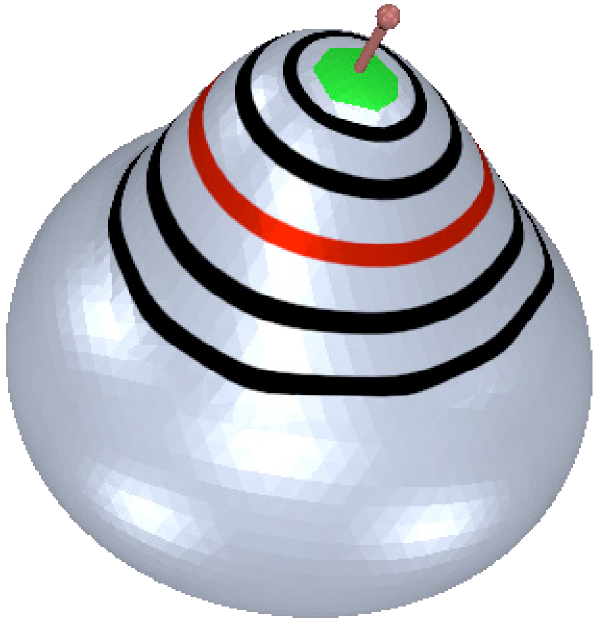
$$s(\mathbf{p}) = \frac{\mathrm{dist}_0(\mathbf{p})}{\mathrm{dist}_0(\mathbf{p}) + \mathrm{dist}_1(\mathbf{p})}$$



  - Harmonic field
    - Solve $\Delta(s) = 0$
    - with $s(\mathbf{p}) = \begin{cases} 1 & \mathbf{p} \in \text{handle} \\ 0 & \mathbf{p} \in \text{fixed} \end{cases}$

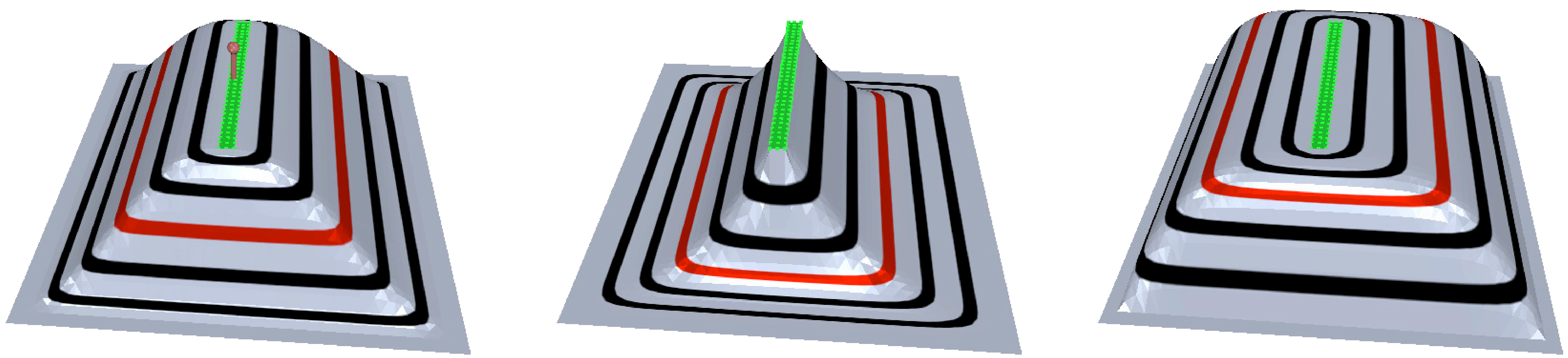
Distance-Based Propagation

- Full affine handle deformation
  - Rotation: $R(\mathbf{c},\mathbf{a},\alpha)$
  - Scaling: $S(s)$
  - Translation: $T(\mathbf{t})$
- Damp with scalar λ
  - Rotation: $R(\mathbf{c},\mathbf{a}, \lambda\cdot\alpha)$
  - Scaling: $S(\lambda\cdot s)$
  - Translation: $T(\lambda\cdot\mathbf{t})$

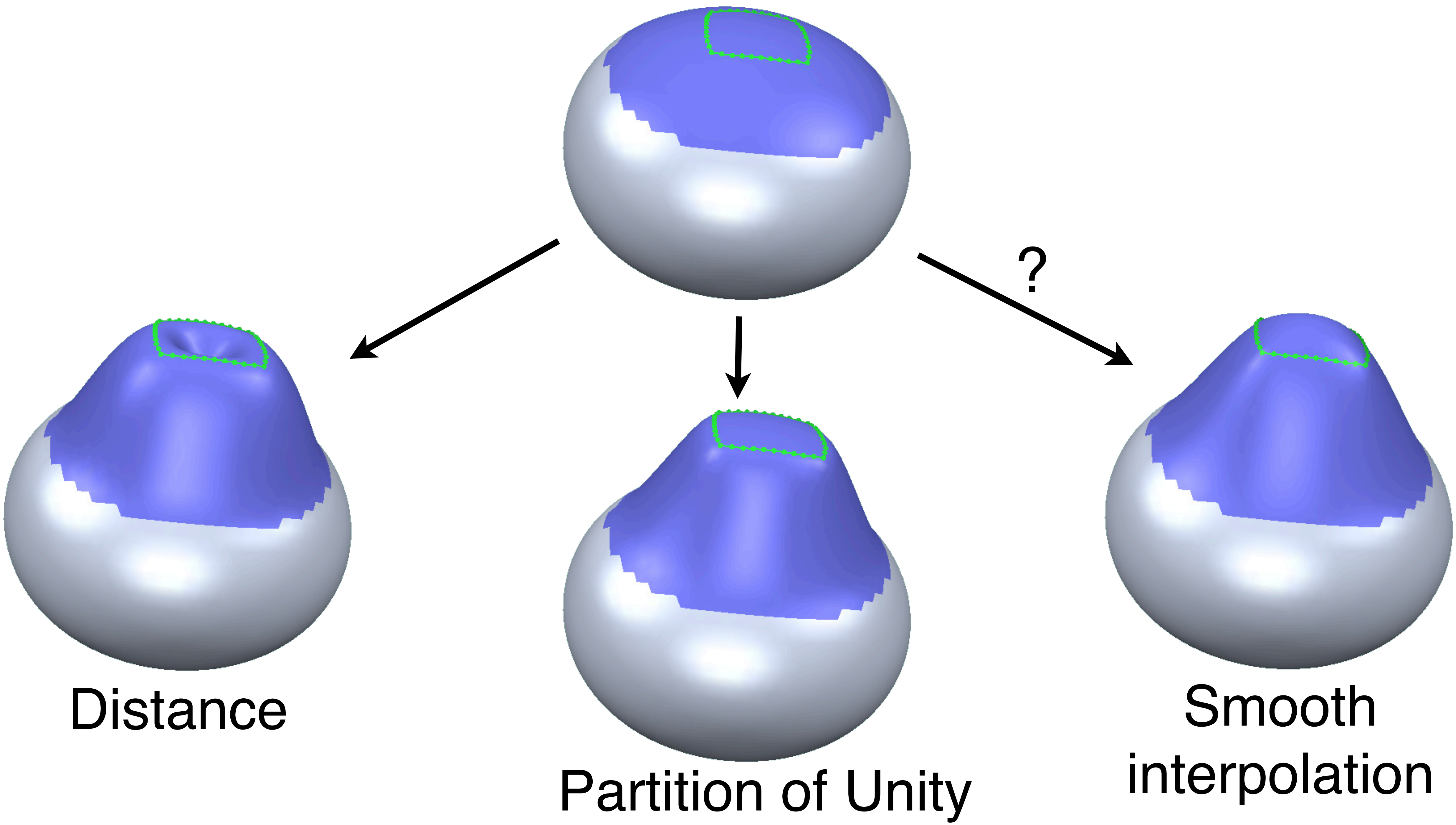


Distance-Based Propagation

- Transfer function t(x)
  - Damp deformation by t(s(x))



Distance-Based Propagation



Boundary Constraint Modeling

1. Control: Prescribe constraints:

$$\mathbf{p}_i \mapsto \mathbf{p}_i'$$

2. Fitting: Smoothly interpolate constraints by a displacement function:

$$\mathbf{d} : S \to \mathbb{R}^3 \quad \text{with} \quad \mathbf{d}(\mathbf{p}_i) = \mathbf{p}_i'$$

3. Evaluation: Displace all points:

$$\mathbf{p}_i \mapsto \mathbf{d}(\mathbf{p}_i) \quad \forall \mathbf{p}_i \in S$$

How to interpolate?

- Constrained energy minimization (thin plate)

$$\int_S \|\mathbf{d}_{uu}\|^2 + 2\,\|\mathbf{d}_{uv}\|^2 + \|\mathbf{d}_{vv}\|^2 \, dS$$

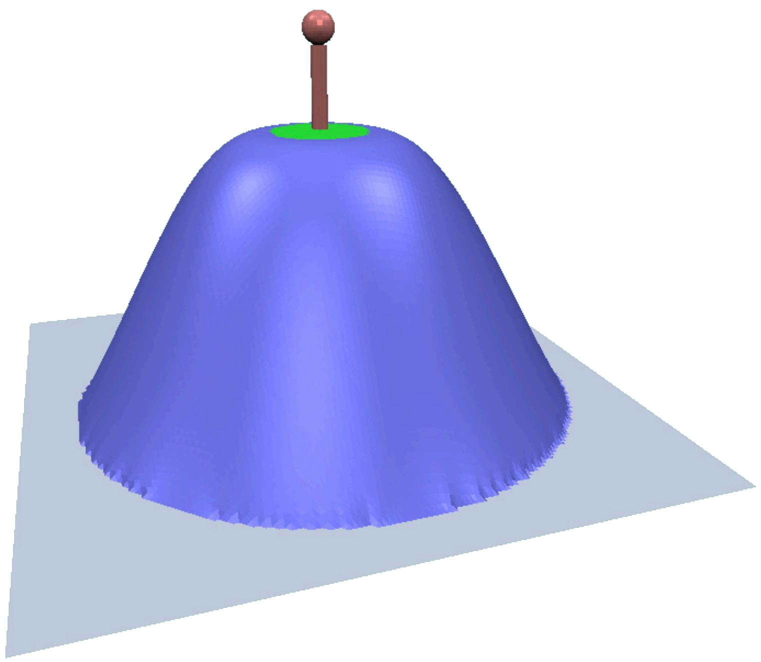
- Euler-Lagrange PDE (sparse linear system)

$$\Delta_S^2 \mathbf{d} \equiv 0$$

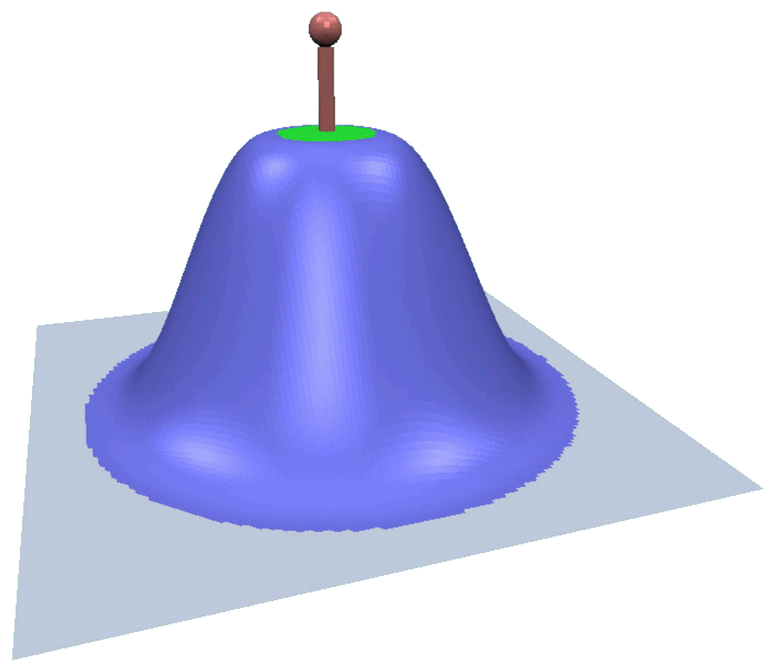
- “Best” deformation which satisfies constraints

Boundary Smoothness

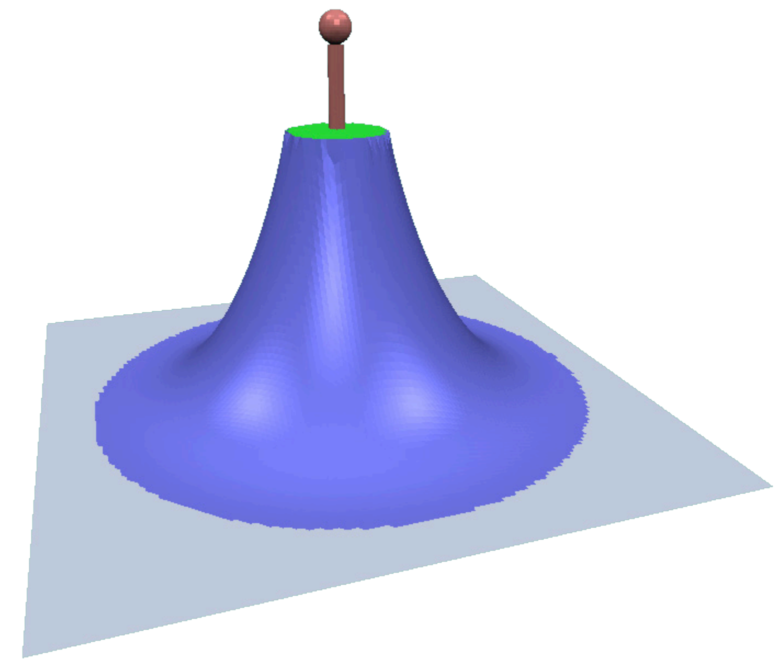
- How smooth does the deformed region blend with fixed surface parts?



C^0/C^2



C^2/C^2



C^2/C^0

Boundary Smoothness

- Δ^k surfaces can do up to C^{k-1} continuity
  - Real-valued smoothness in [0, k-1]

- Adjust recursive Laplace definition

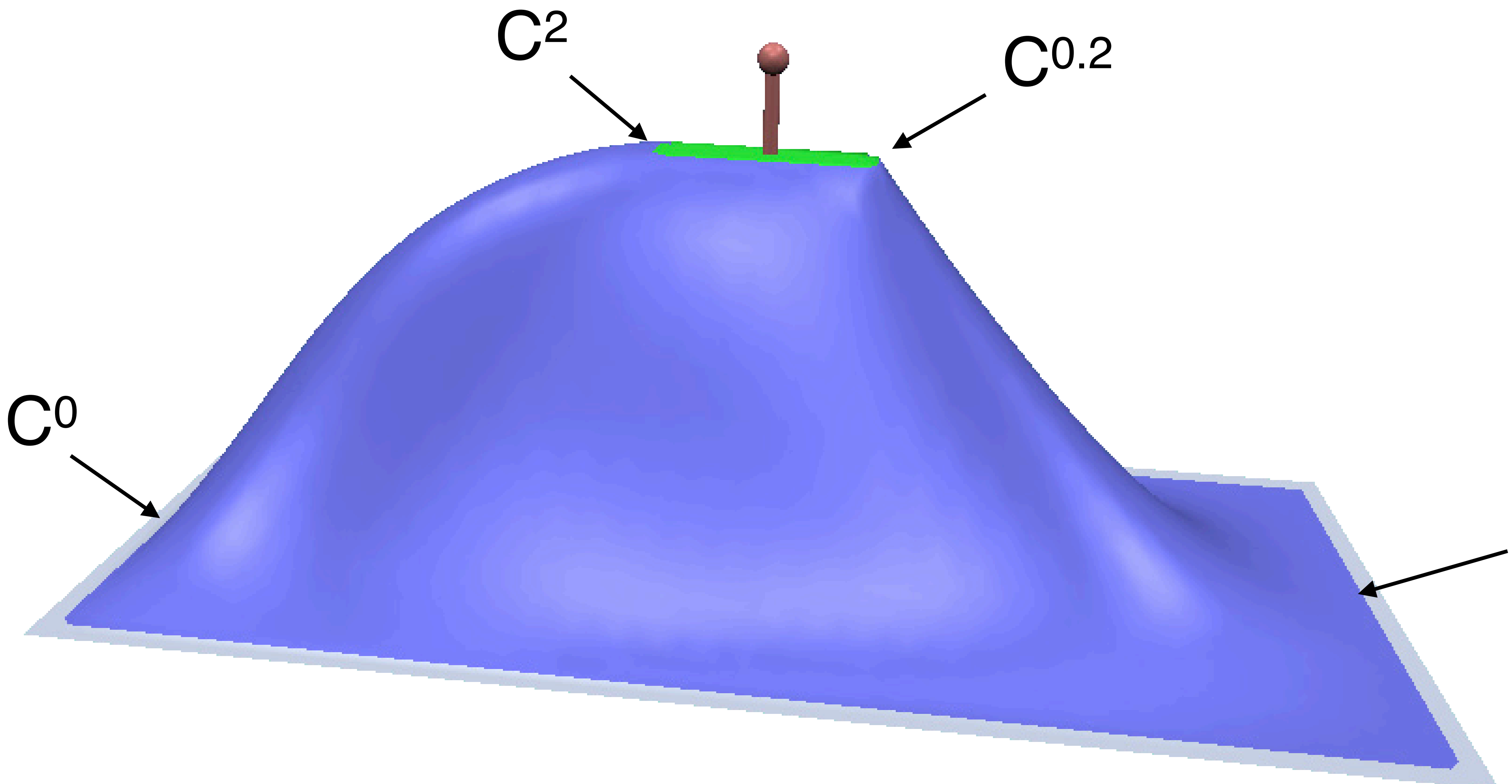
$$\Delta_S^3 \mathbf{d}(\mathbf{p}) = \Delta_S\left(\Delta_S\left(\Delta_S \mathbf{d}(\mathbf{p})\right)\right)$$

↓

$$\bar{\Delta}_S^3\, \mathbf{d}(\mathbf{p}) = \Delta_S\left(\lambda_2(\mathbf{p})\, \Delta_S\left(\lambda_1(\mathbf{p})\, \Delta_S \mathbf{d}(\mathbf{p})\right)\right)$$

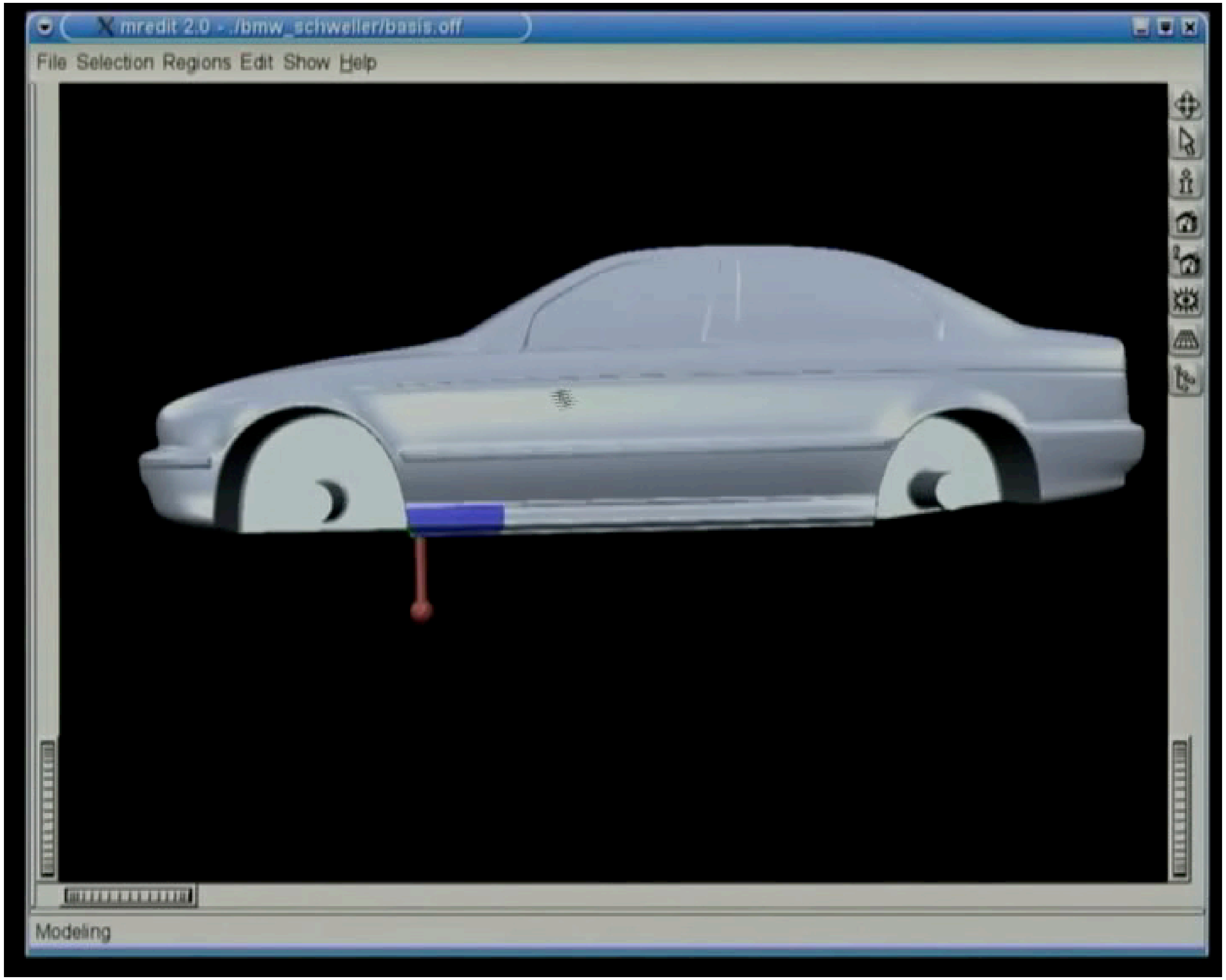
$\lambda_2(\mathbf{p})$	$\lambda_1(\mathbf{p})$	
0	$[0,1]$	$C^{0+\lambda_1(\mathbf{p})}$
$[0,1]$	1	$C^{1+\lambda_2(\mathbf{p})}$

Boundary Smoothness



Per-vertex "continuous" boundary smoothness

Sillboard Deformation



Literature

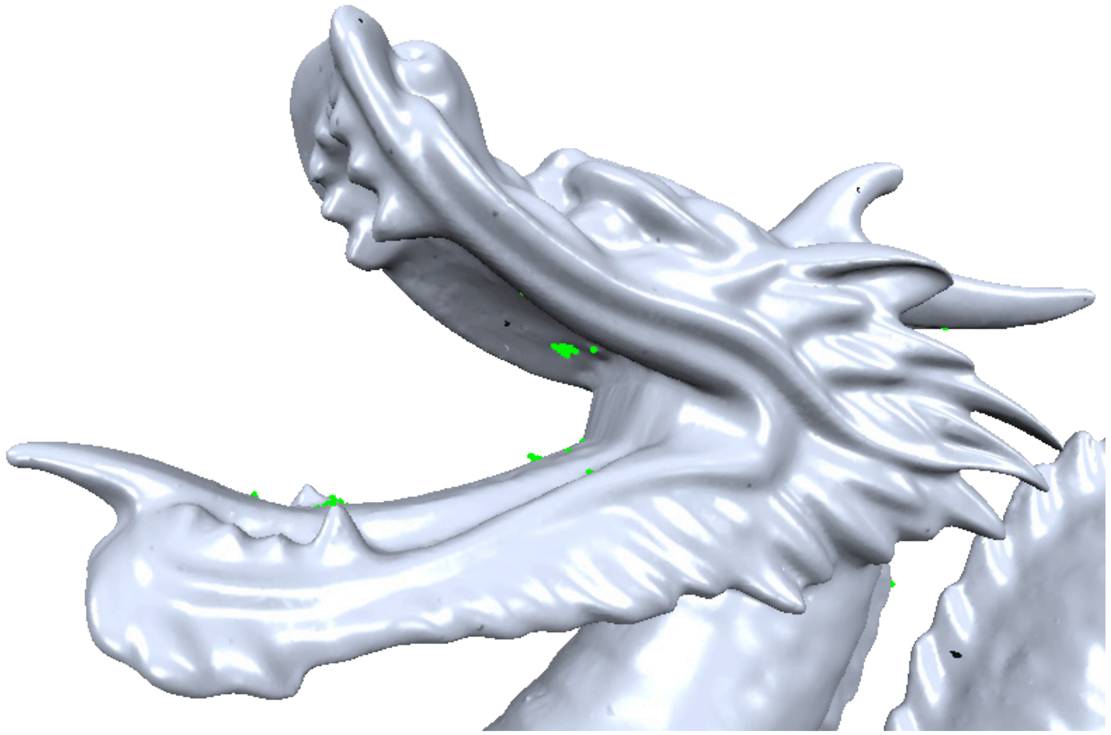
- Pauly et al, *"Shape modeling with point-sampled geometry"*, SIGGRAPH 2003
- Bendels & Klein, *"Mesh forging: editing of 3D-meshes using implicitly defined occluders"*, Symp. on Geometry Processing 2003
- Botsch & Kobbelt, *"An intuitive framework for real-time freeform modeling"*, SIGGRAPH 2004

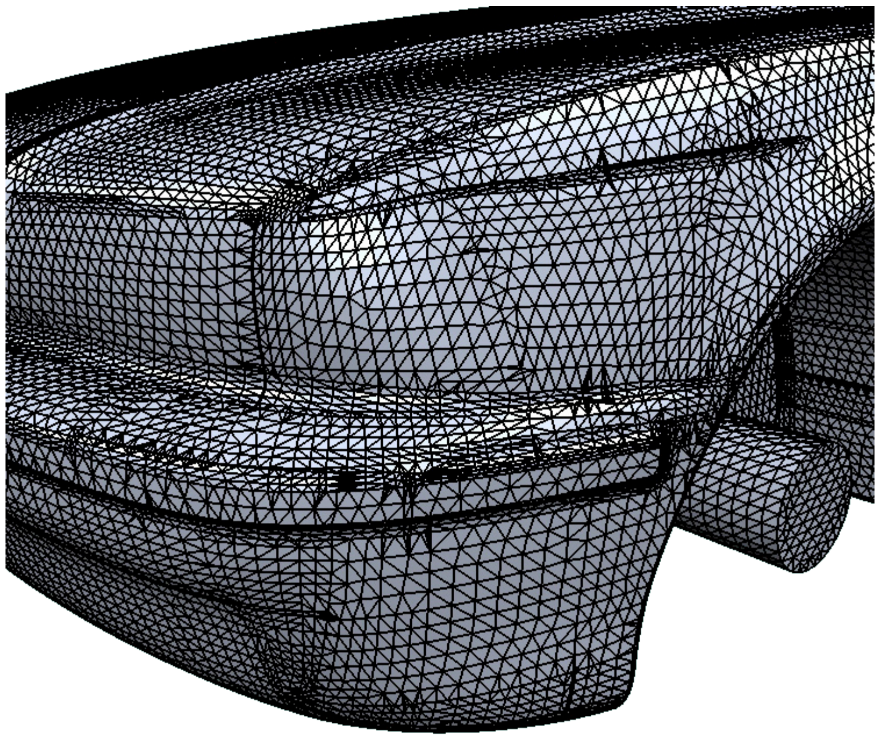
Shape Editing

- Surface-Based Deformation
  - Distance-Based Propagation
  - Boundary Constraint Modeling
- Space Deformation
  - Freeform Deformation
  - RBF Deformation
- Multiresolution Deformation

Surface-Based Deformation

- Problems with
  - Highly complex models
  - Topological inconsistencies
  - Geometric degeneracies





Surface-Based Deformation

1. Control: Prescribe constraints:

$$\mathbf{p}_i \mapsto \mathbf{p}_i'$$

2. Fitting: Smoothly interpolate constraints by a displacement function:

$$\mathbf{d} : S \rightarrow \mathbb{R}^3 \quad \text{with} \quad \mathbf{d}(\mathbf{p}_i) = \mathbf{p}_i'$$

3. Evaluation: Displace all points:

$$\mathbf{p}_i \mapsto \mathbf{d}(\mathbf{p}_i) \quad \forall \mathbf{p}_i \in S$$

Space Deformation

1. Control: Prescribe constraints:

$$\mathbf{p}_i \mapsto \mathbf{p}_i'$$

2. Fitting: Smoothly interpolate constraints by a displacement function in space:

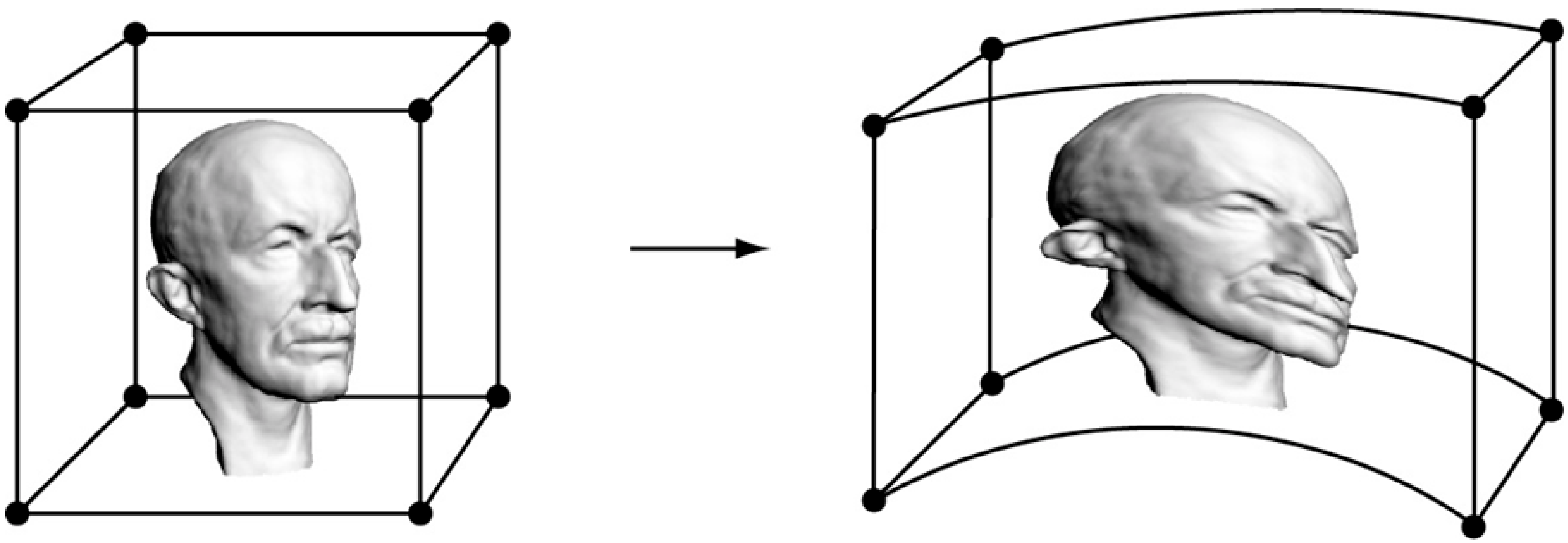
$$\mathbf{d} : \mathbb{R}^3 \to \mathbb{R}^3 \quad \text{with} \quad \mathbf{d}(\mathbf{p}_i) = \mathbf{p}_i'$$

3. Evaluation: Displace all points:

$$\mathbf{p}_i \mapsto \mathbf{d}(\mathbf{p}_i) \quad \forall \mathbf{p}_i \in S$$

Freeform Deformation

- Deform object's bounding box
  - Implicitly deforms embedded objects



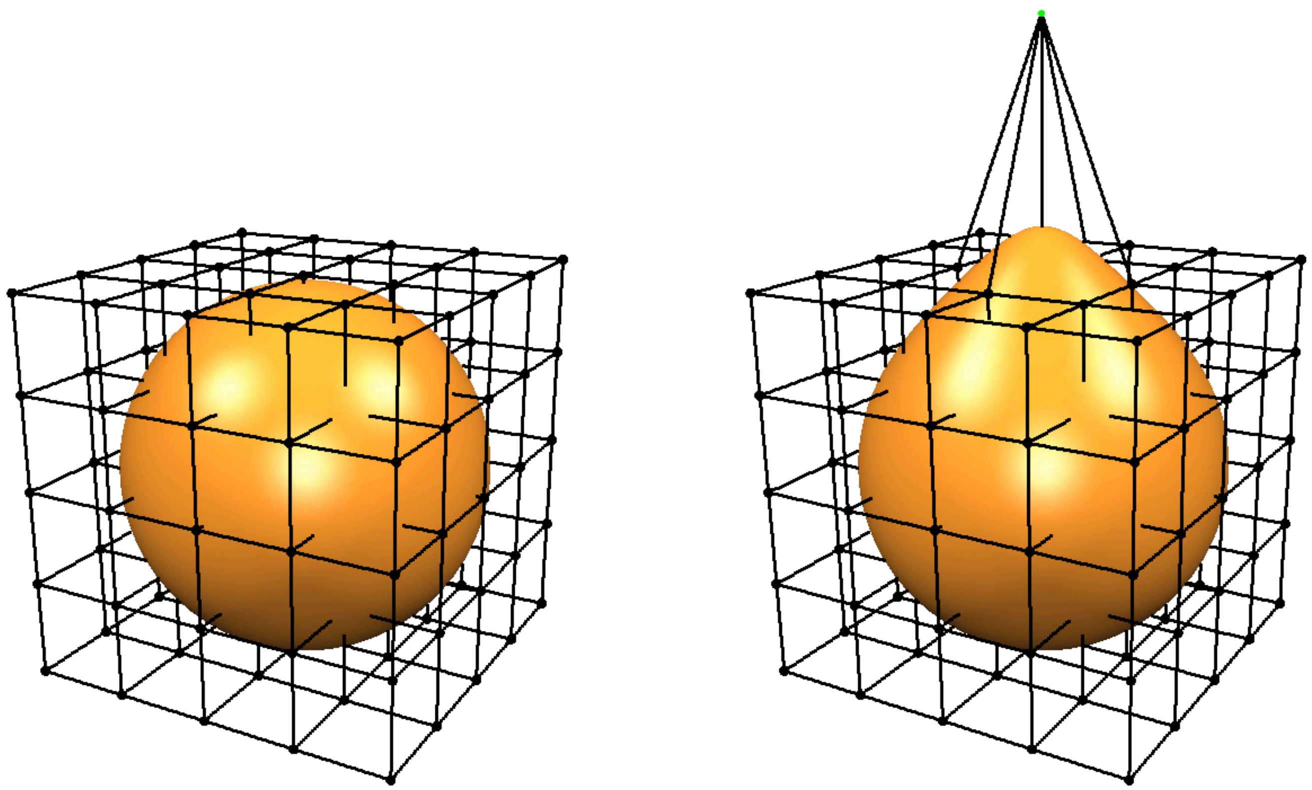
Freeform Deformation

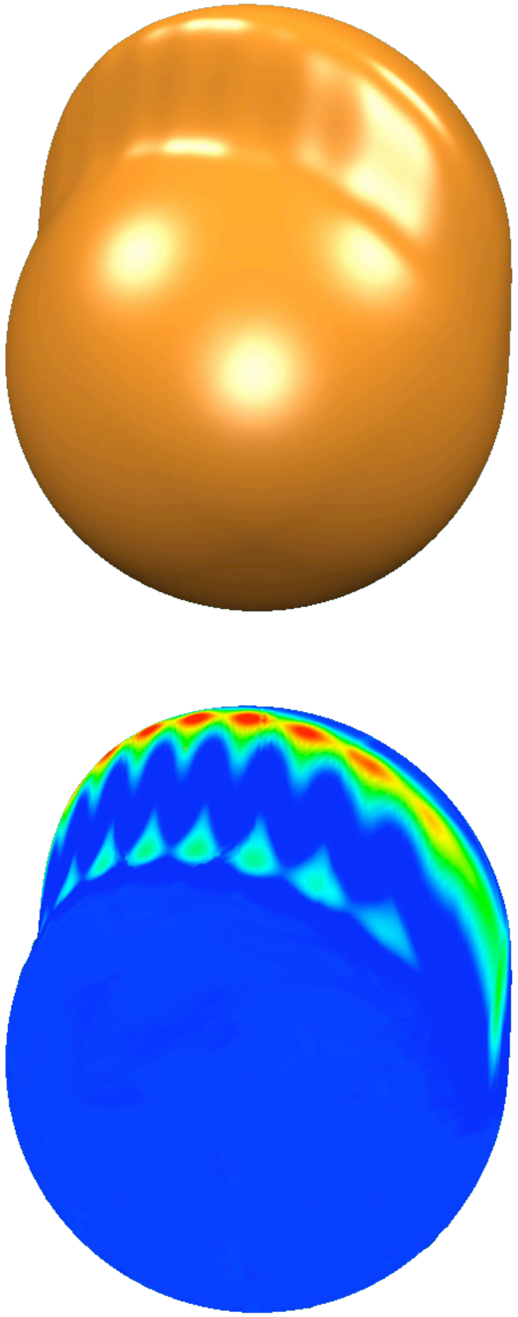
- Deform object's bounding box
  - Implicitly deforms embedded objects
- Tri-variate tensor-product spline

$$\mathbf{d}(u, v, w) = \sum_{i=0}^{l} \sum_{j=0}^{m} \sum_{k=0}^{n} \mathbf{c}_{ijk} N_i^l(u) \, N_j^m(v) \, N_k^n(w)$$

Freeform Deformation

- Deform object's bounding box
  - Implicitly deforms embedded objects
- Tri-variate tensor-product spline





Freeform Deformation

- Deform object's bounding box
  - Implicitly deforms embedded objects
- Tri-variate tensor-product spline
  - Aliasing artifacts
- Interpolate deformation constraints?
  - Only limited constraints...

Space Deformation

1. Control: Prescribe constraints:

$$\mathbf{p}_i \mapsto \mathbf{p}_i'$$

2. Fitting: Smoothly interpolate constraints by a displacement function in space:

$$\mathbf{d} : \mathbb{R}^3 \to \mathbb{R}^3 \quad \text{with} \quad \mathbf{d}(\mathbf{p}_i) = \mathbf{p}_i'$$

3. Evaluation: Displace all points:

$$\mathbf{p}_i \mapsto \mathbf{d}(\mathbf{p}_i) \quad \forall \mathbf{p}_i \in S$$

Radial Basis Functions

- Represent deformation by RBFs

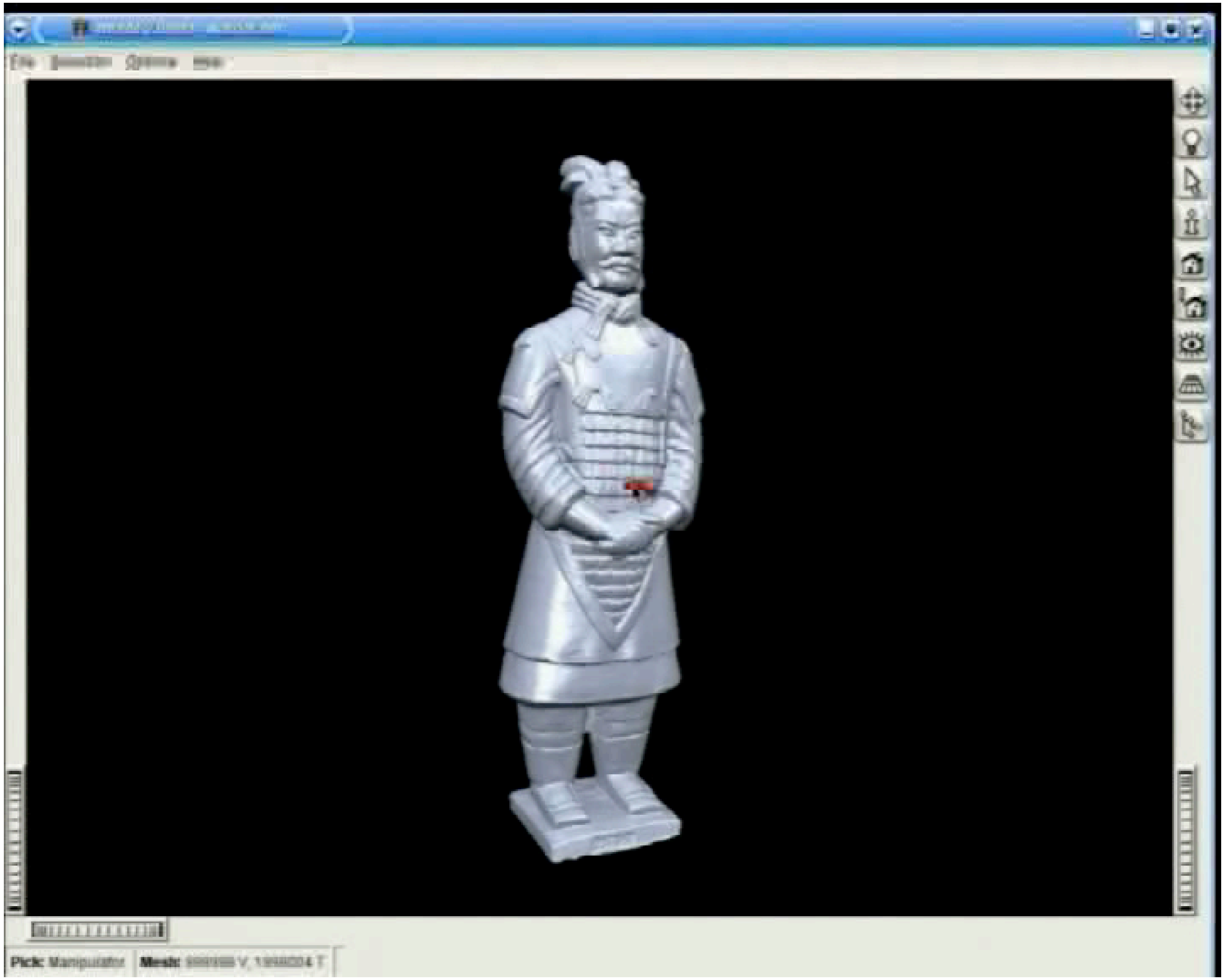
$$\mathbf{d}(\mathbf{x}) = \sum_j \mathbf{w}_j \cdot \varphi\left(\|\mathbf{c}_j - \mathbf{x}\|\right) + \mathbf{p}(\mathbf{x})$$

- Well suited for scattered data interpolation
  - Smooth interpolation
  - Irregularly placed constraints

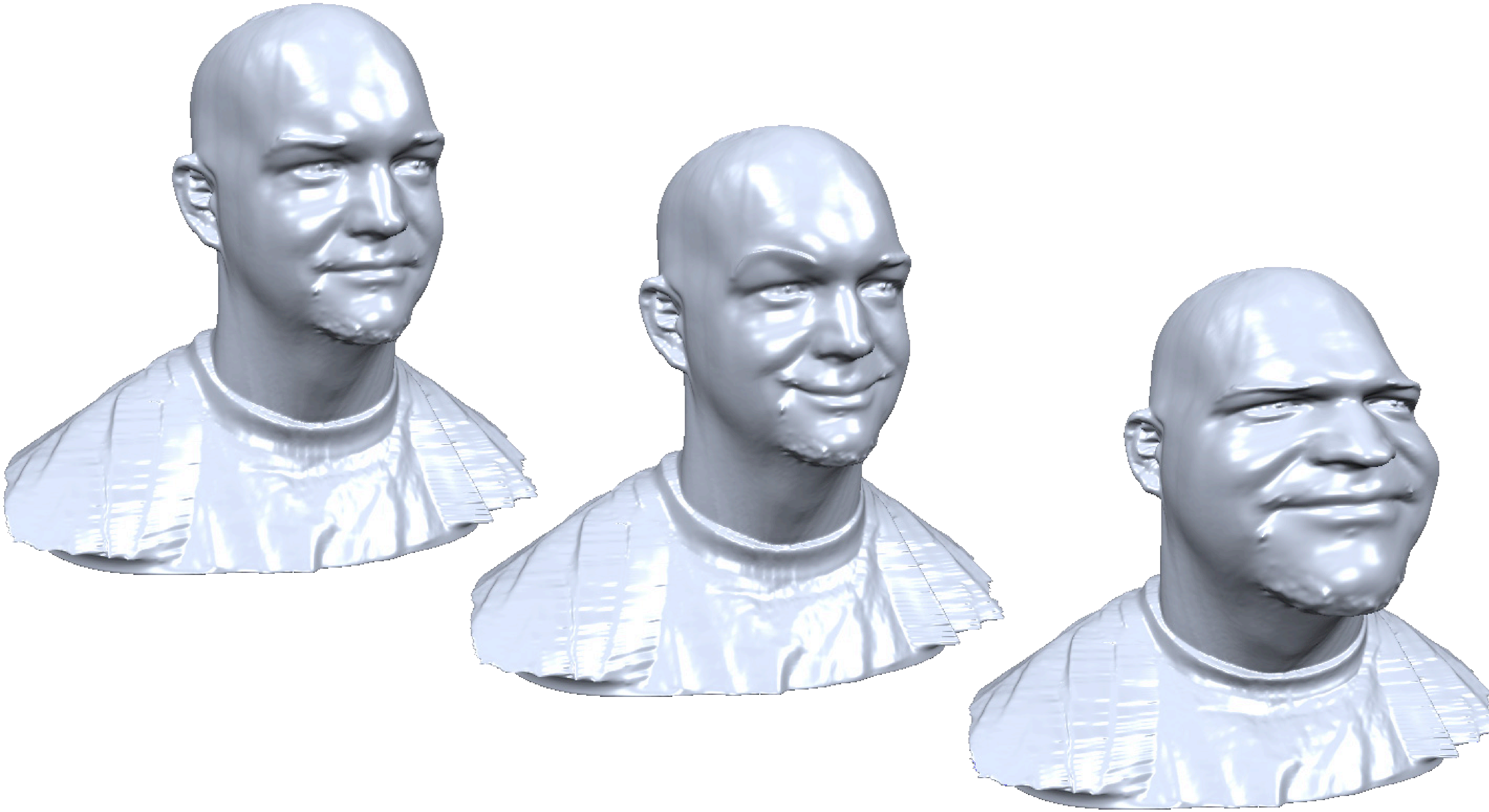
Which basis function?

- Triharmonic RBF $\varphi(r) = r^3$
  - C2 boundary constraints
  - High smoothness *(energy minimization)*
  - Leads to linear system [Botsch & Kobbelt 2005]
  - Can be evaluated on the GPU

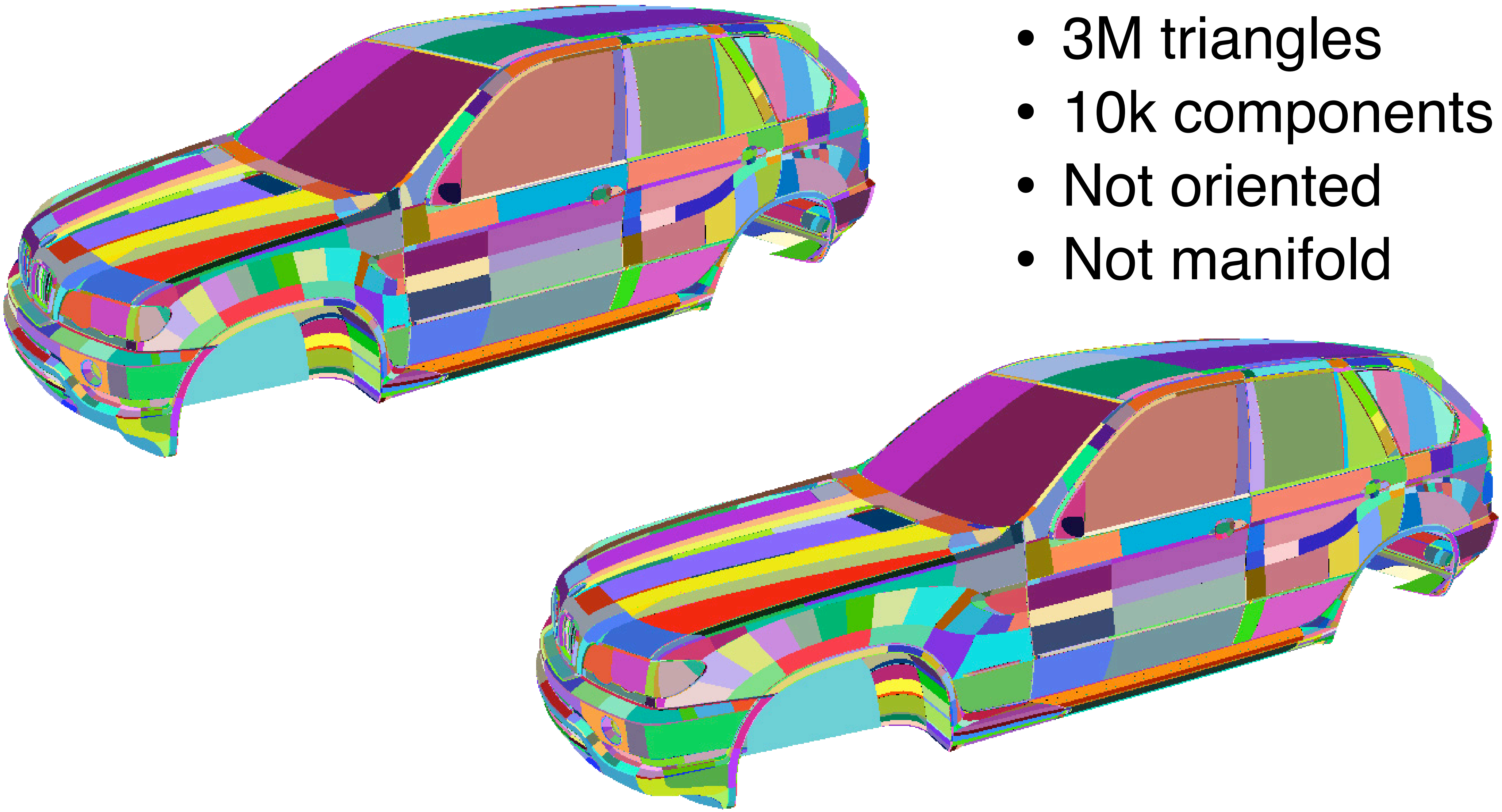
Statue: 1M vertices



Local & Global Deformations



"Bad Meshes"



- 3M triangles
- 10k components
- Not oriented
- Not manifold

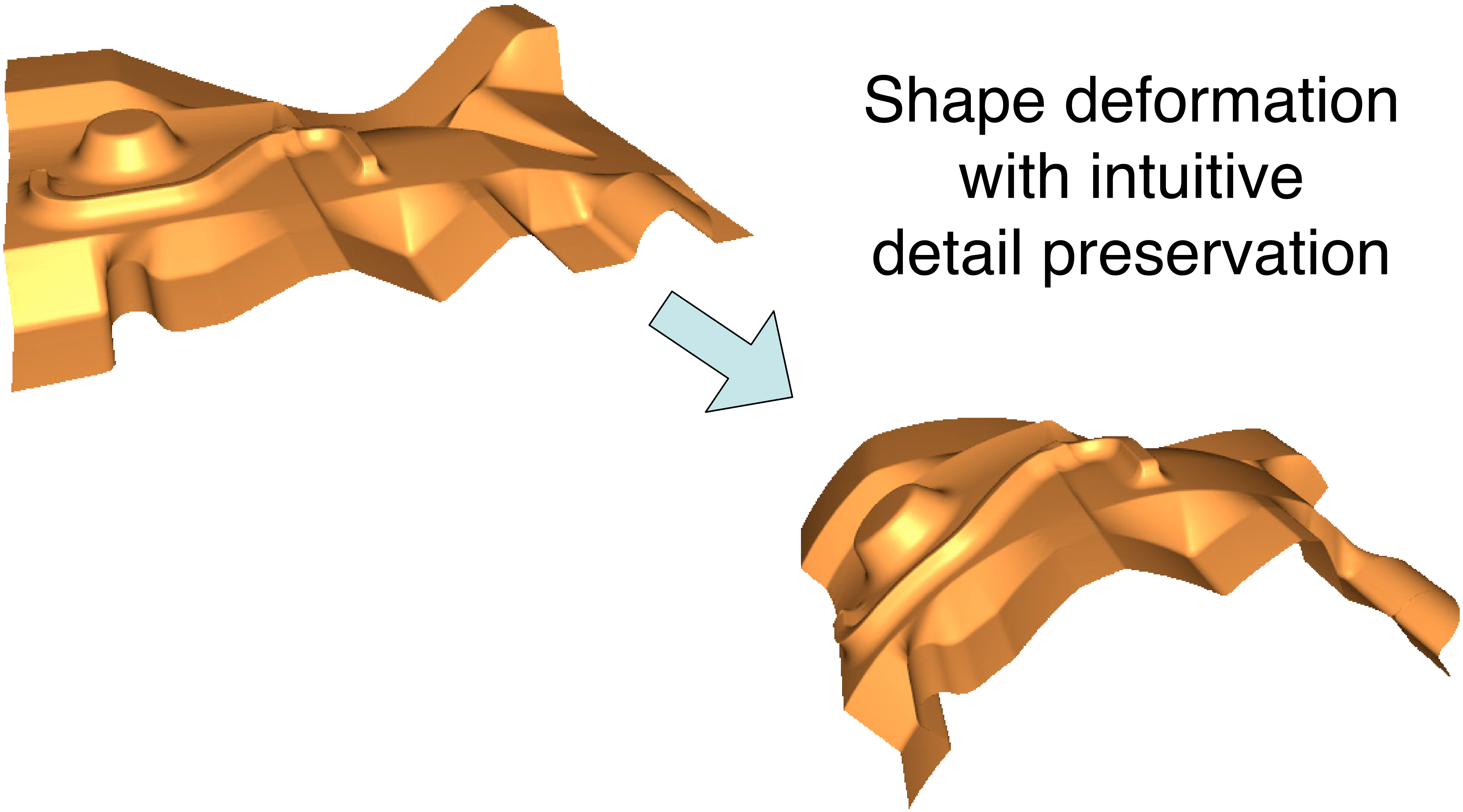
Literature

- Sederberg & Parry, *“Free-Form Deformation of Solid Geometric Models”*, SIGGRAPH 1986
- Botsch & Kobbelt, *“Real-time shape editing using radial basis functions”*, Eurographics 2005

Shape Editing

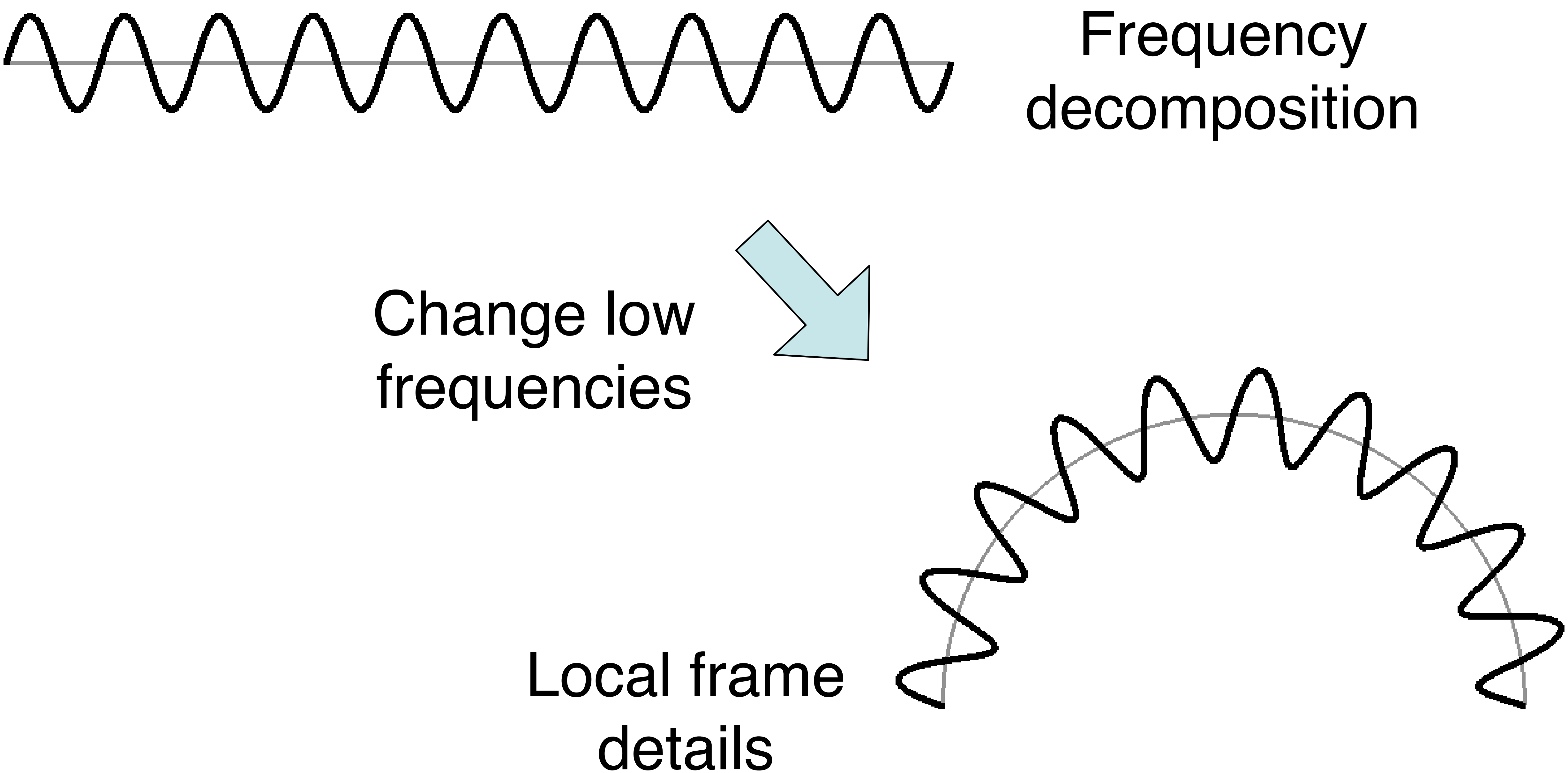
- Surface-Based Deformation
  - Distance-Based Propagation
  - Boundary Constraint Modeling
- Space Deformation
  - Freeform Deformation
  - RBF Deformation
- Multiresolution Deformation

Multiresolution Editing

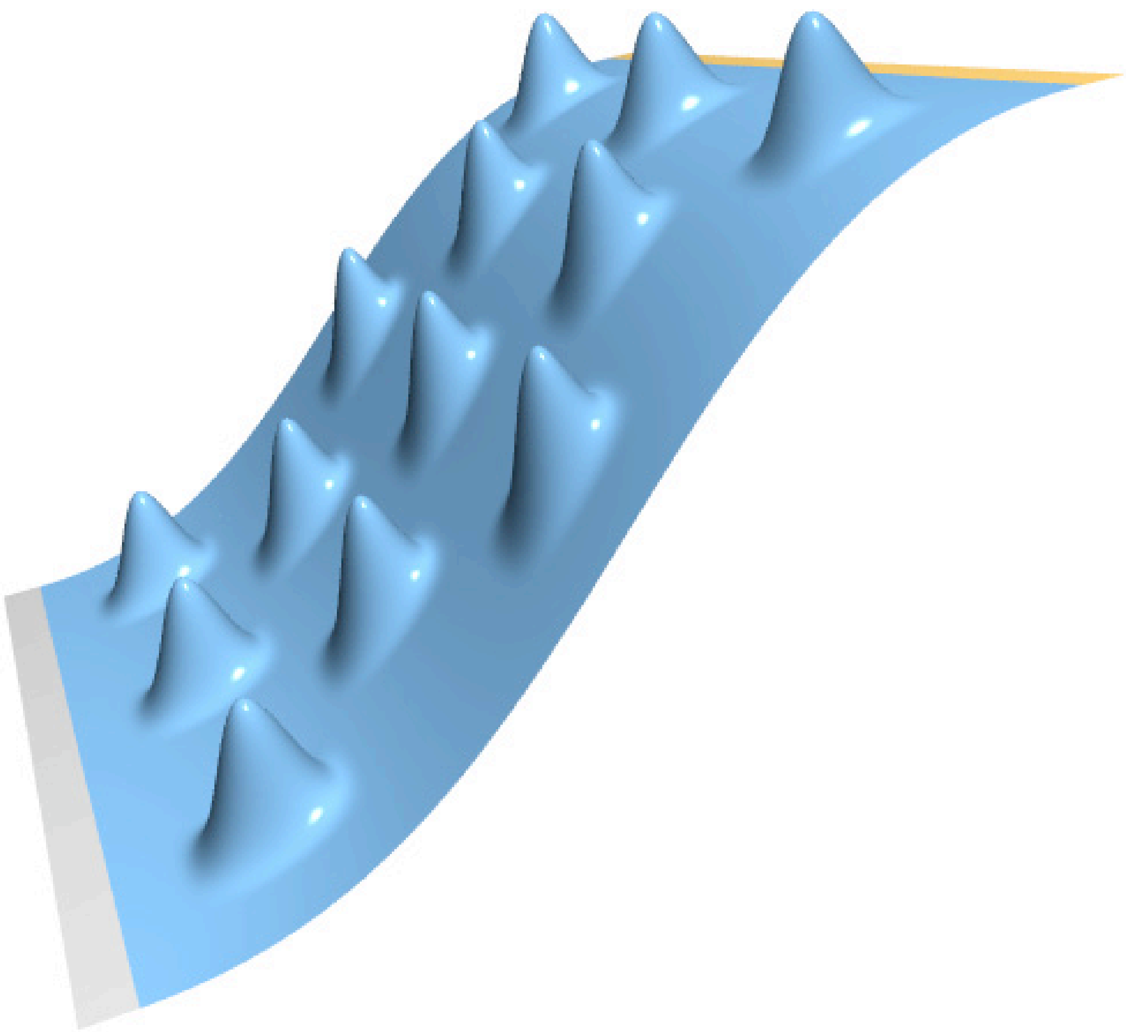


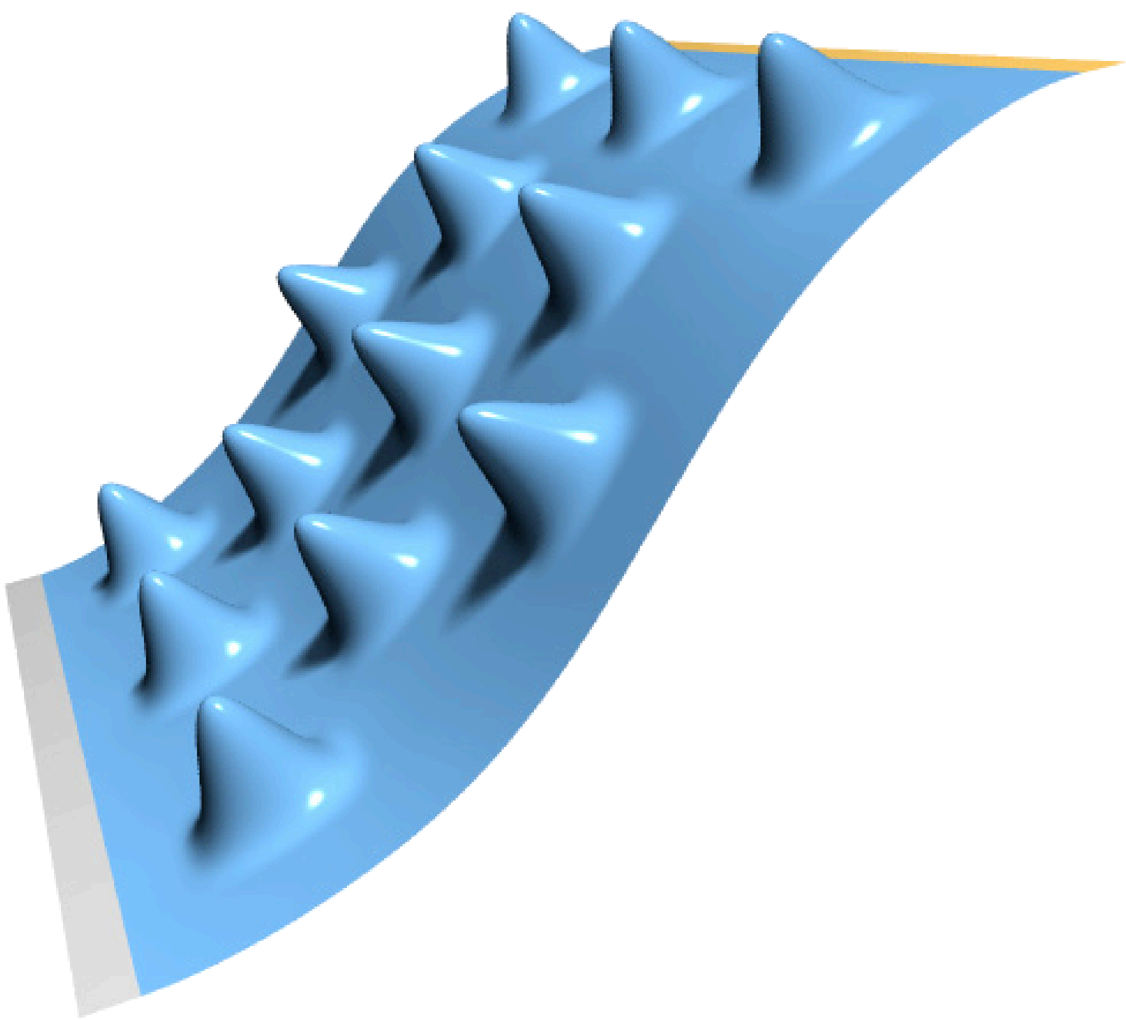
Shape deformation
with intuitive
detail preservation

Multiresolution Editing

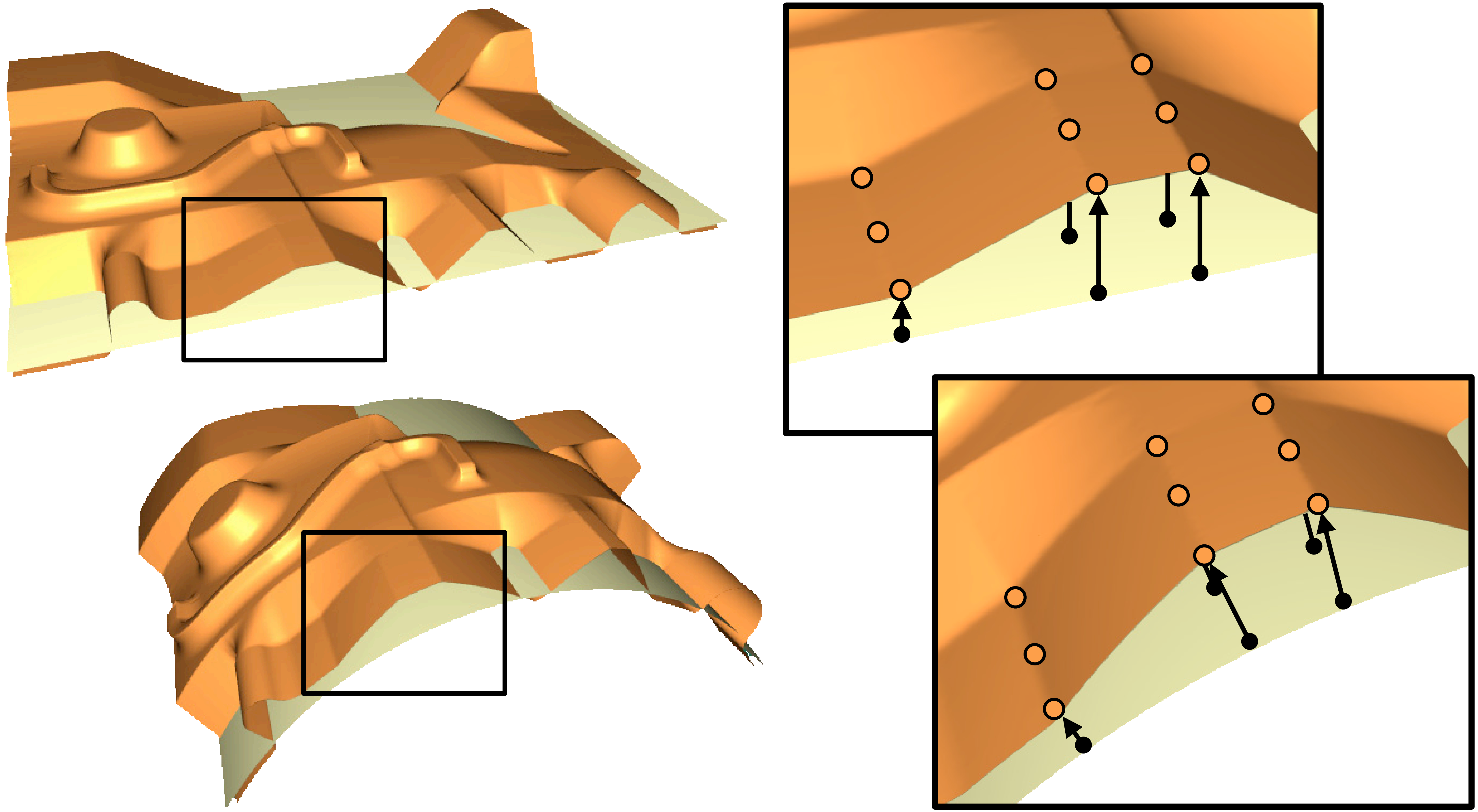


Multiresolution Modeling

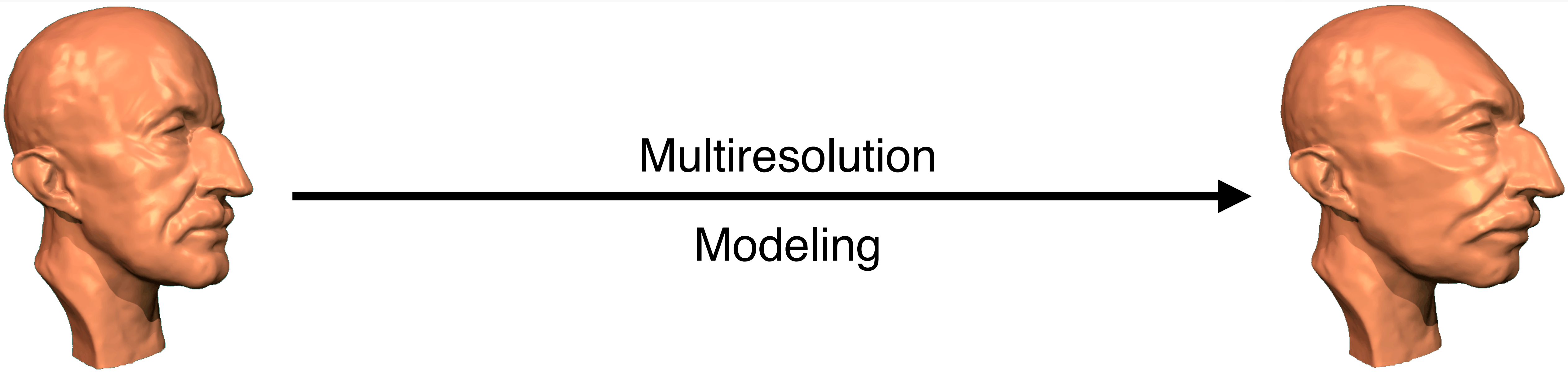


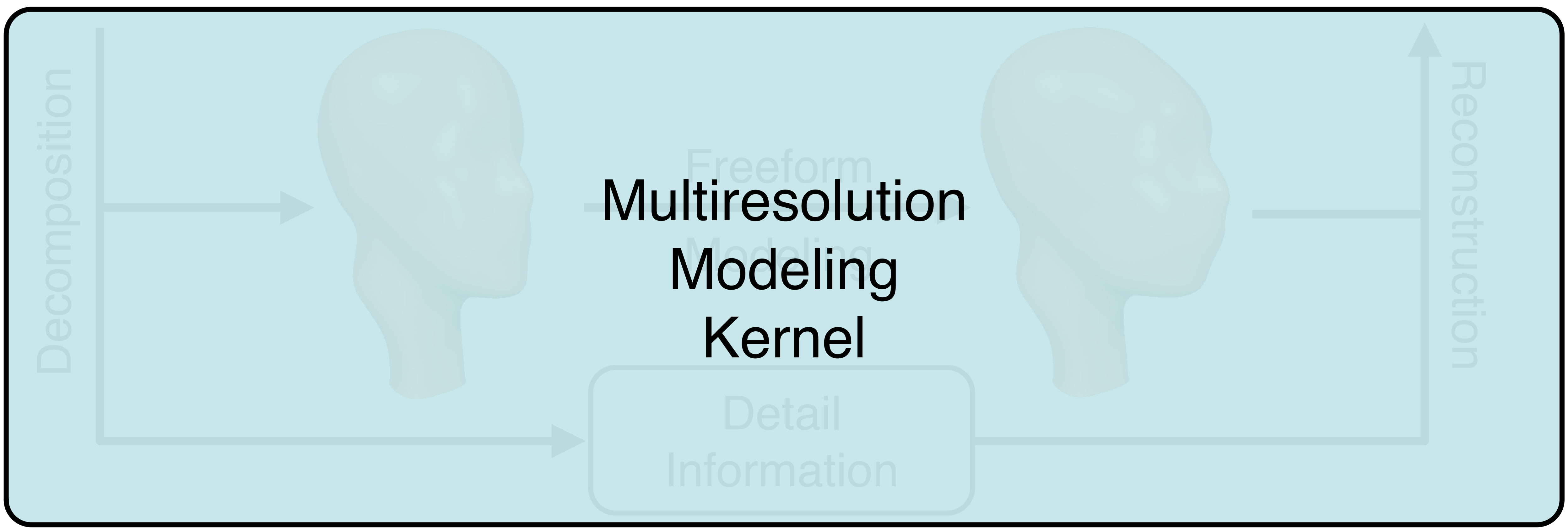


Multiresolution Editing

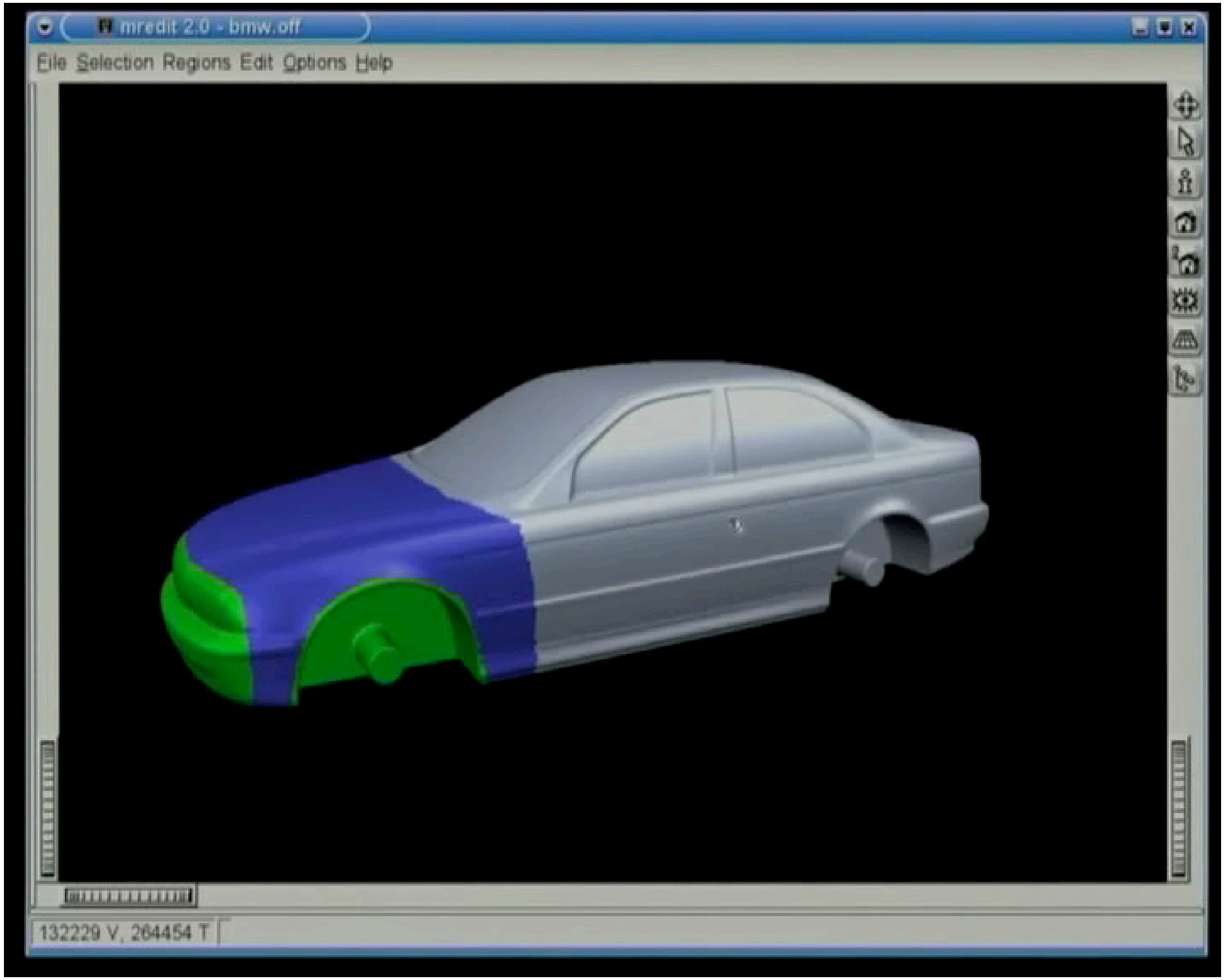


Multiresolution Editing





Front Deformation



Literature

- Kobbelt et al, *“Multiresolution hierarchies on unstructured triangle meshes”*, Comput. Geom. Theory Appl. 14(1-3), 1999
- Botsch & Kobbelt, *“A remeshing approach to multiresolution modeling”*, Symp. on Geometry Processing 2004

Shape Editing

- Surface-Based Deformation
  - Distance-Based Propagation
  - Boundary Constraint Modeling
- Space Deformation
  - Freeform Deformation
  - RBF Deformation
- Multiresolution Deformation