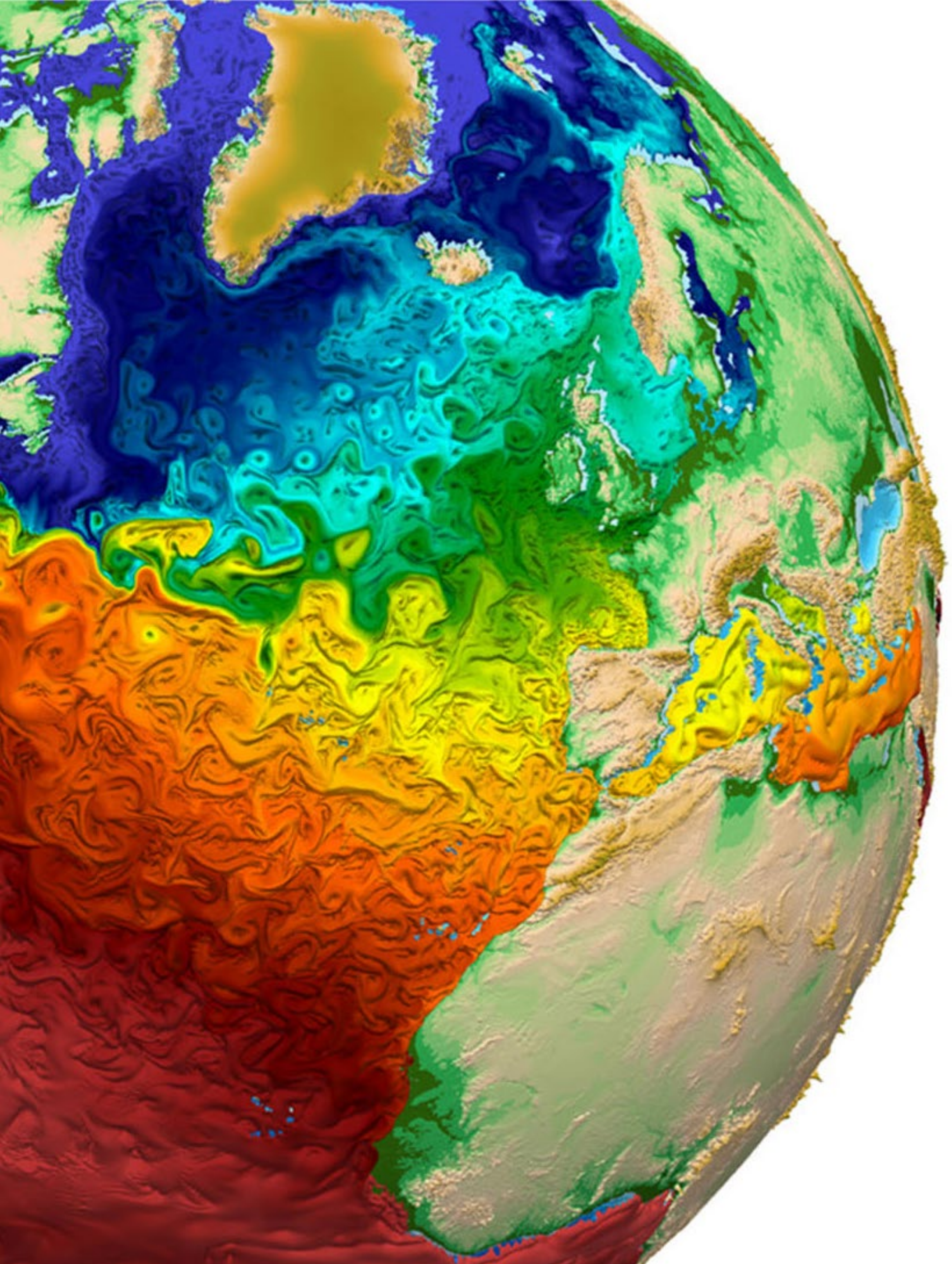


Mesh Simplification

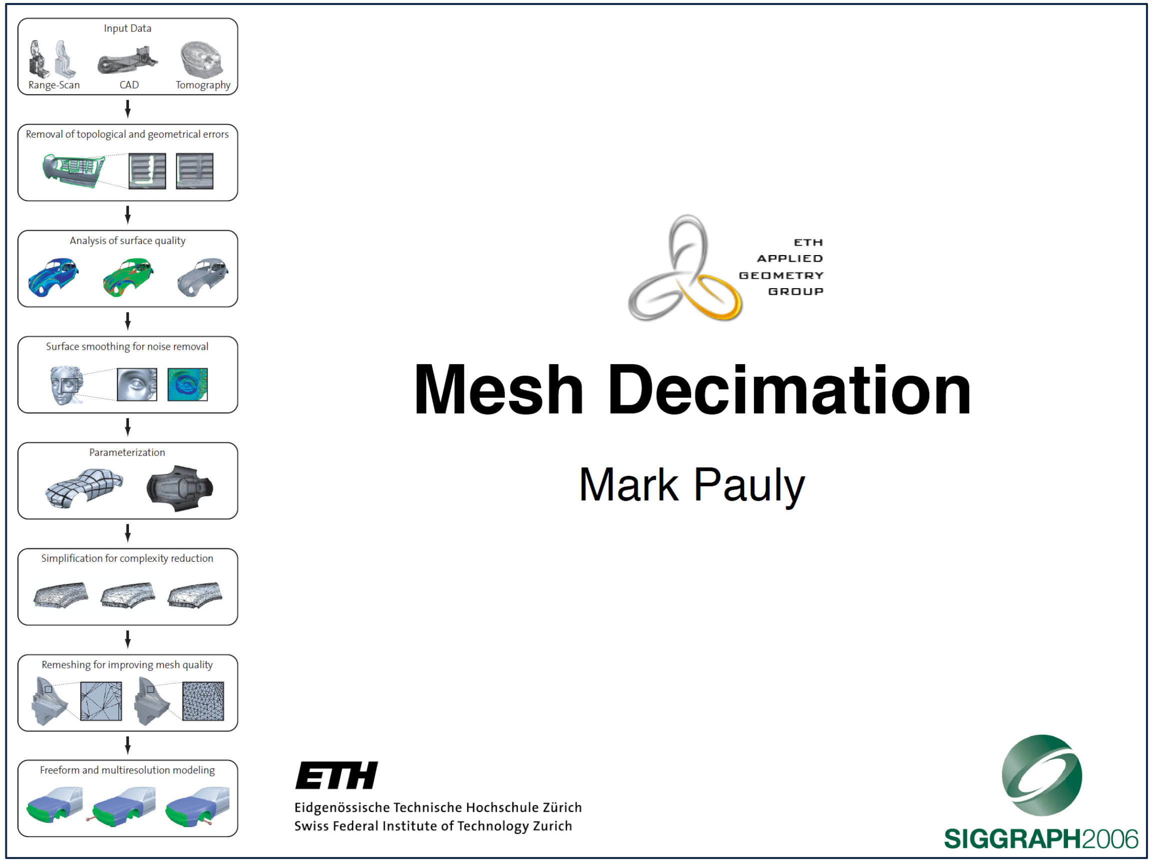
Scientific Visualization

Professor Eric Shaffer

ILLINOIS

Acknowledgements

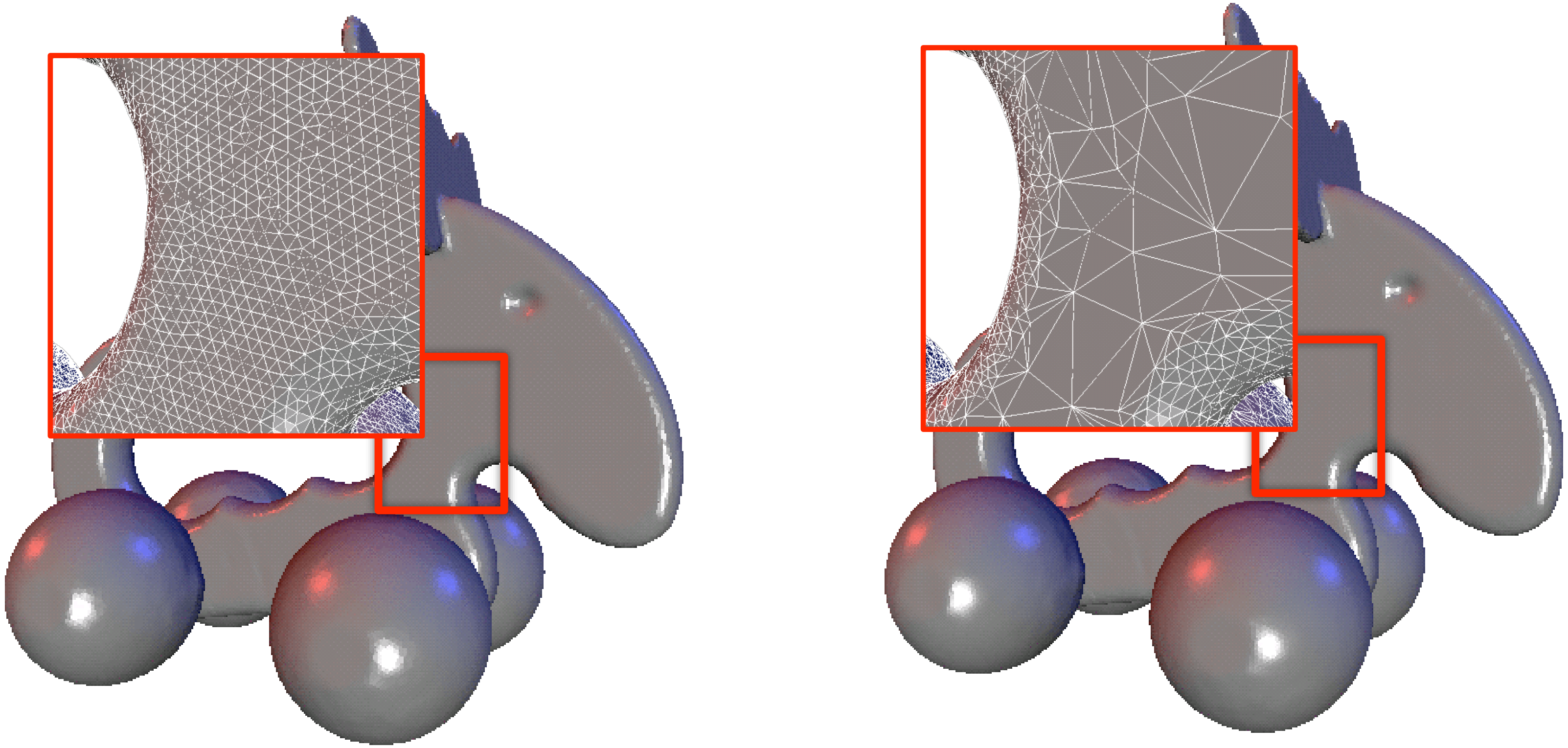
Slides based on presentation by Professor Mark Pauly of ETH Zurich



http://www.pmp-book.org/

Surface Meshes often Overtesselated

- Oversampled 3D scan data



~150k triangles

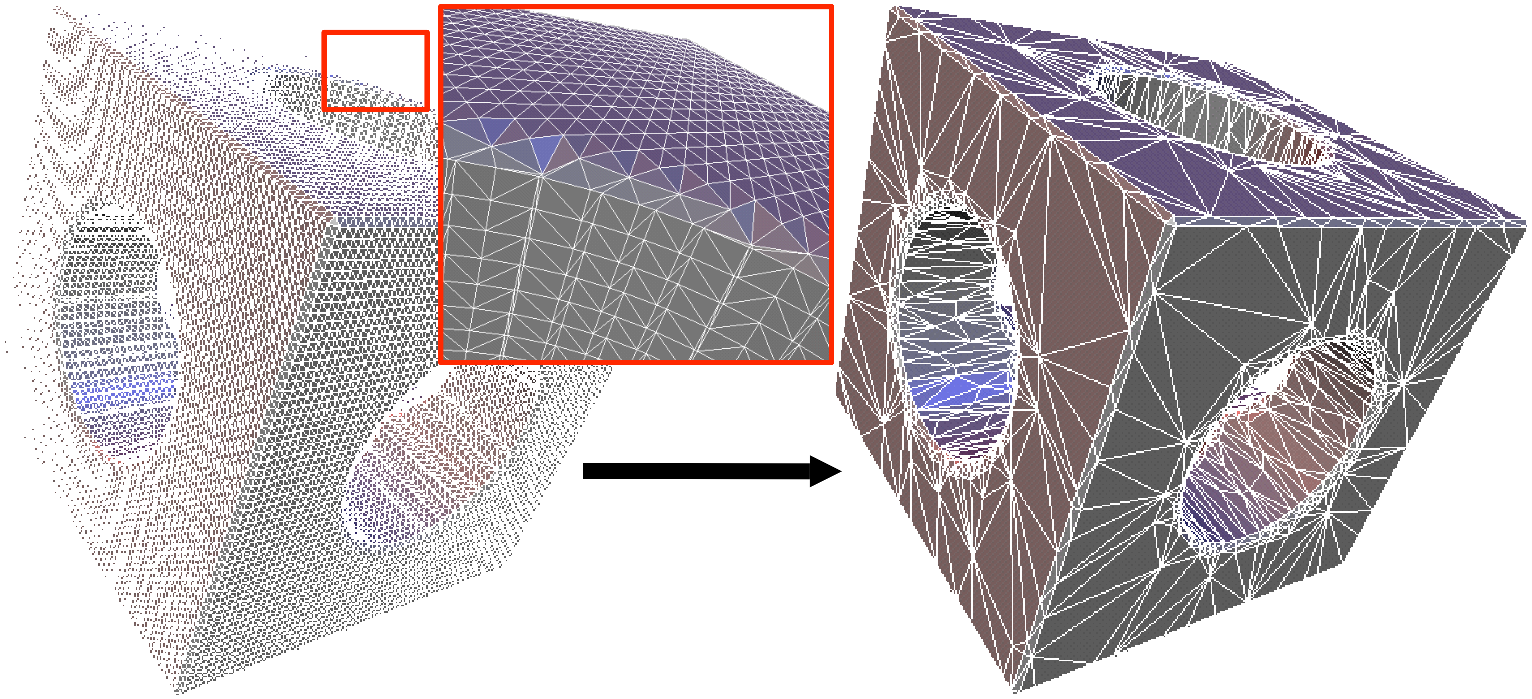
~80k triangles

Same shape can be well-approximated by many fewer triangles

ILLINOIS

Surface Meshes often Overtessellated

- Overtessellation: E.g. iso-surface extraction

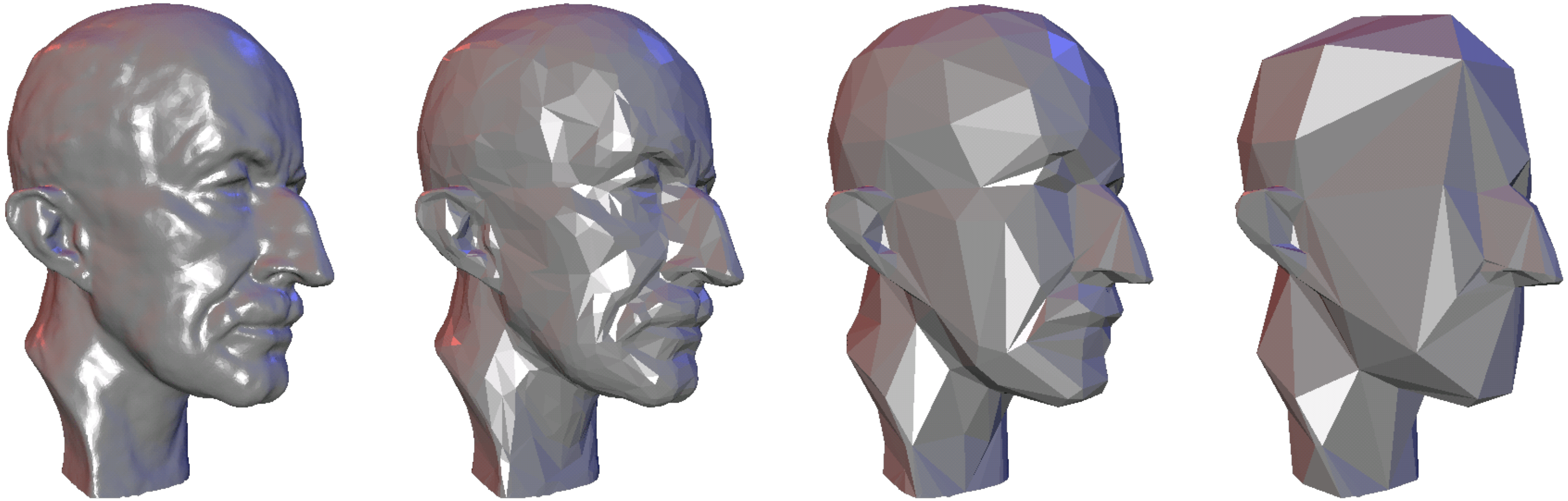


- Large polygon counts can make applications non-performant
- Problematic for interactive visualization

ILLINOIS

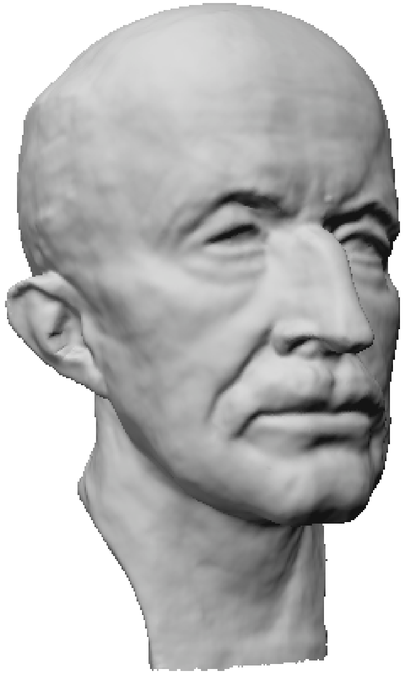
Multi-resolution Hierarchies

- Construct multiple versions of mesh
  - Varying polygon count
- Multi-resolution hierarchies enable
  - efficient geometry processing
  - level-of-detail (LOD) rendering



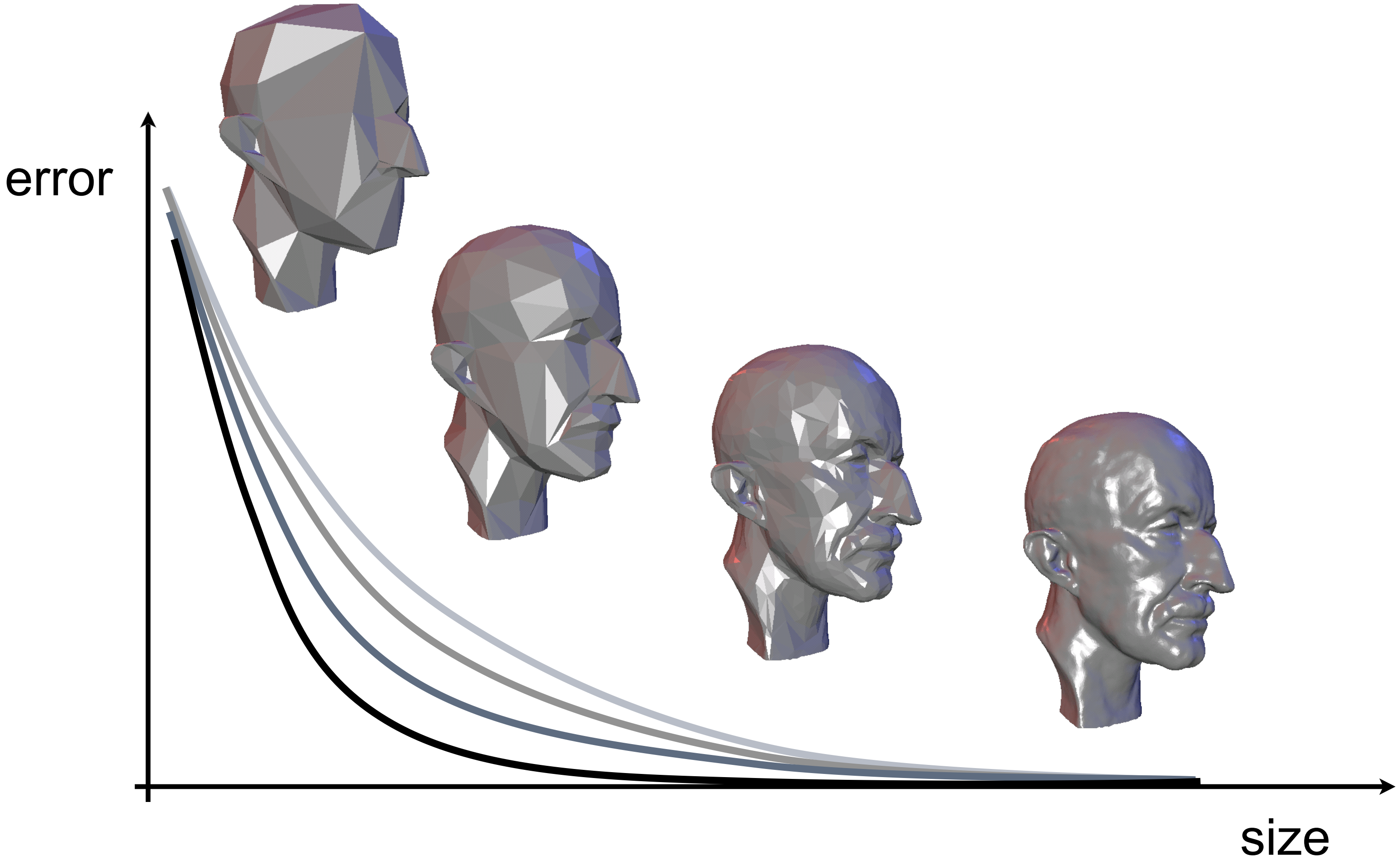
Applications

- Adaptation to hardware capabilities



ILLINOIS

Size-Quality Tradeoff

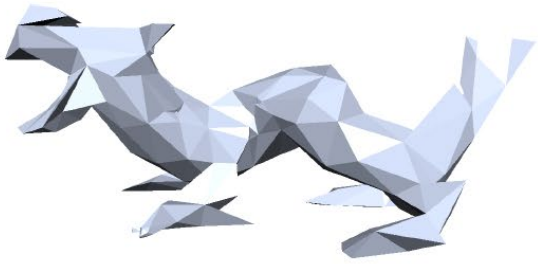


Problem Statement

- Given: $\mathcal{M} = (\mathcal{V}, \mathcal{F})$
- Find: $\mathcal{M}' = (\mathcal{V}', \mathcal{F}')$ such that
  1. $|\mathcal{V}'| = n < |\mathcal{V}|$ and $\|\mathcal{M} - \mathcal{M}'\|$ is minimal, or
  2. $\|\mathcal{M} - \mathcal{M}'\| < \epsilon$ and $|\mathcal{V}'|$ is minimal

$\mathcal{M}$

$\longrightarrow$



$\mathcal{M}'$

ILLINOIS

Problem Statement

- Given: $\mathcal{M} = (\mathcal{V}, \mathcal{F})$
- Find: $\mathcal{M}' = (\mathcal{V}', \mathcal{F}')$ such that

1. $|\mathcal{V}'| = n < |\mathcal{V}|$ and $\|\mathcal{M} - \mathcal{M}'\|$ is minimal, or
2. $\|\mathcal{M} - \mathcal{M}'\| < \epsilon$ and $|\mathcal{V}'|$ is minimal

hard!

→ look for sub-optimal solution

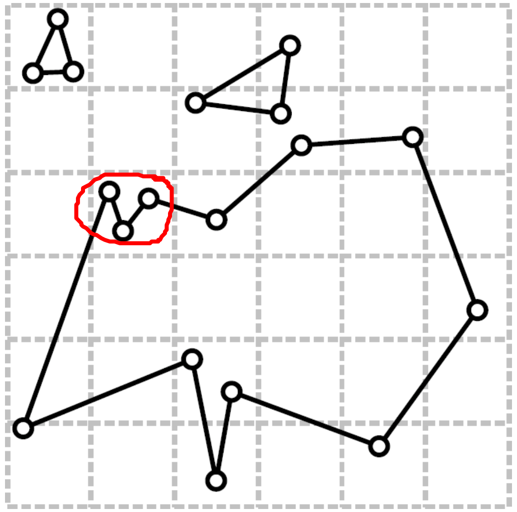
ILLINOIS

Mesh Simplification: Vertex Clustering

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

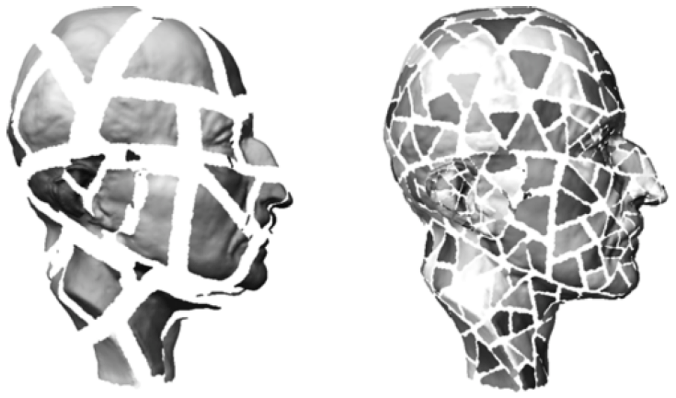
Vertex Clustering

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



Vertex Clustering

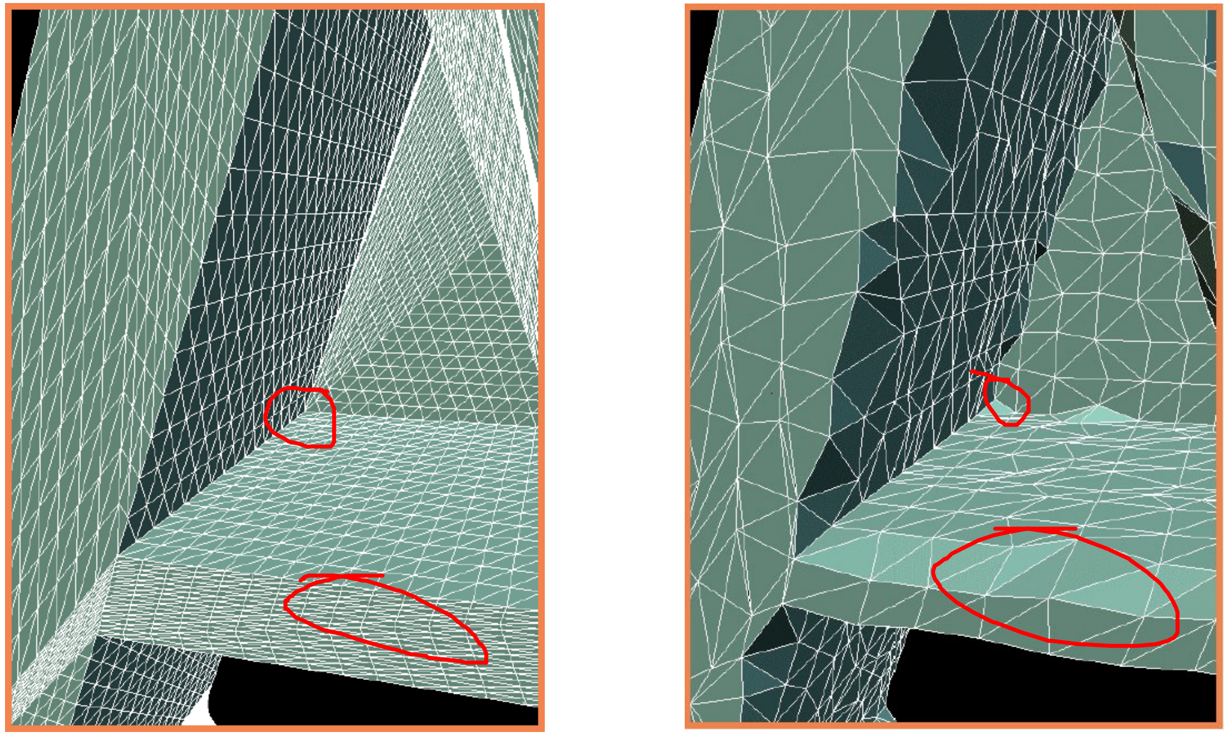
- Cluster Generation
  - Hierarchical approach
  - Top-down or bottom-up
- Computing a representative
- Mesh generation
- Topology changes



Vertex Clustering

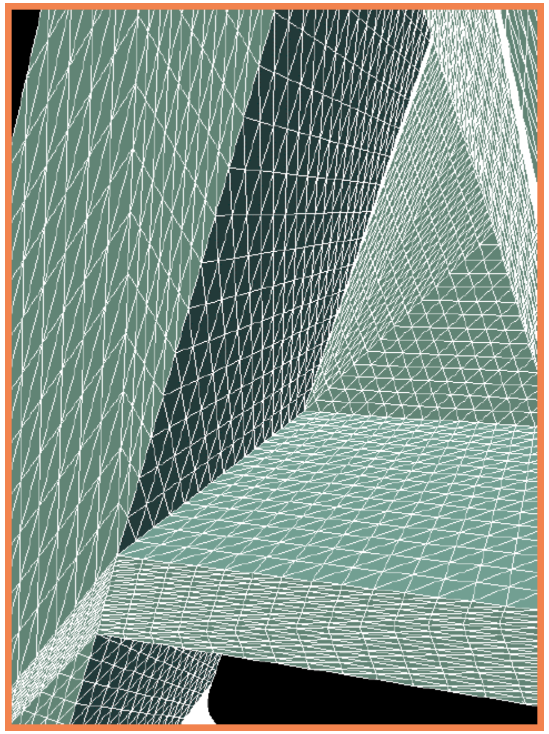
- Cluster Generation
- Computing a representative
  - Average/median vertex position
  - Error quadrics
- Mesh generation
- Topology changes

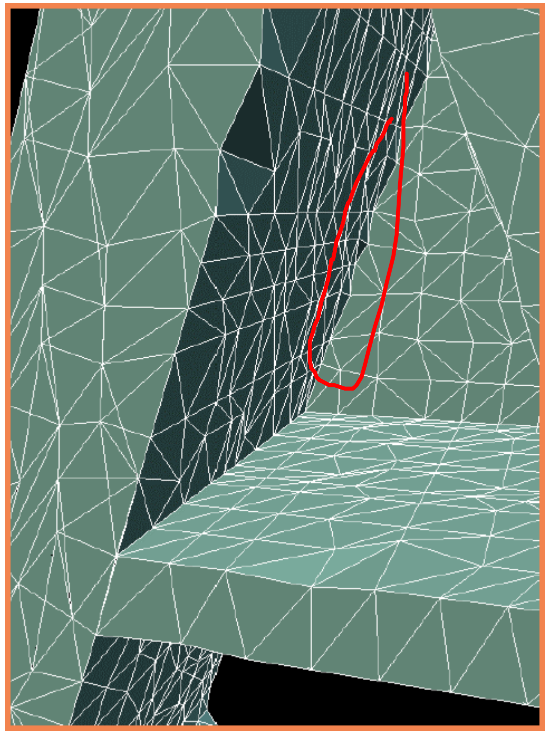
Computing a Representative



Average vertex position → Low-pass filter

Computing a Representative



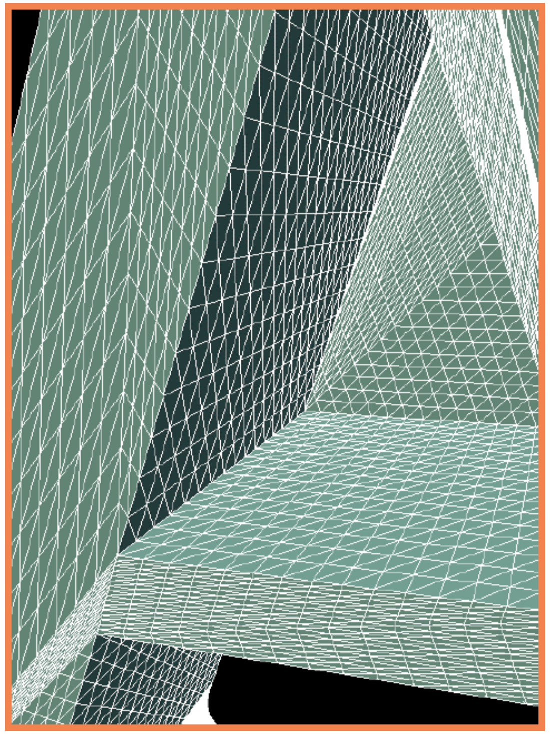


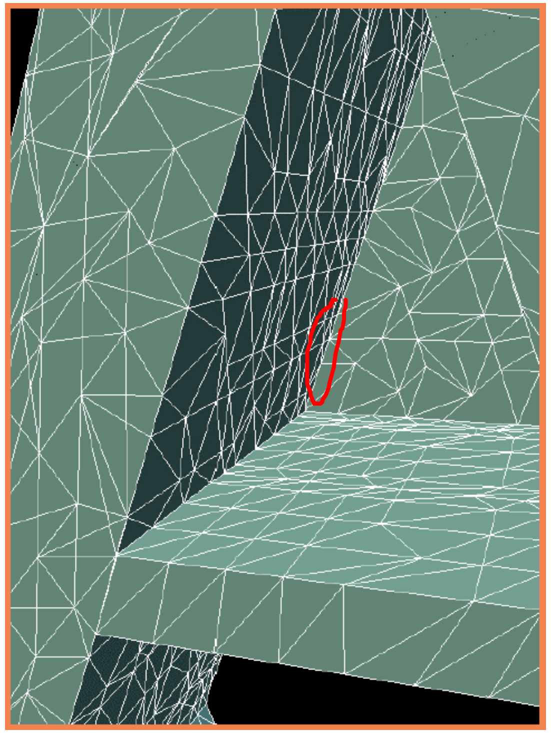
Median vertex position → Sub-sampling

Median vertex will be a vertex of the original mesh closest to the average position of the vertices in the cluster.

ILLINOIS

Computing a Representative





Error quadrics

Error Quadrics

- Squared distance to plane

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

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

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

Using implicit form of a plane equation
ax+by+cz+d=0

ILLINOIS

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

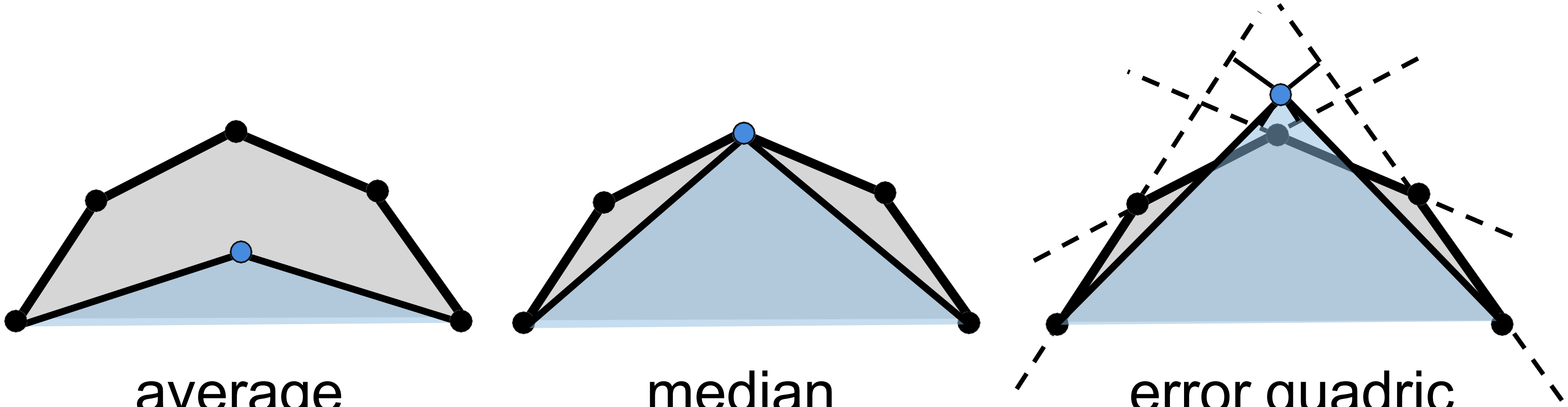
$$\begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ q_{31} & q_{32} & q_{33} & q_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} p^* = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

You can compute the sum of squared distances from p to N planes using a single 4x4 matrix

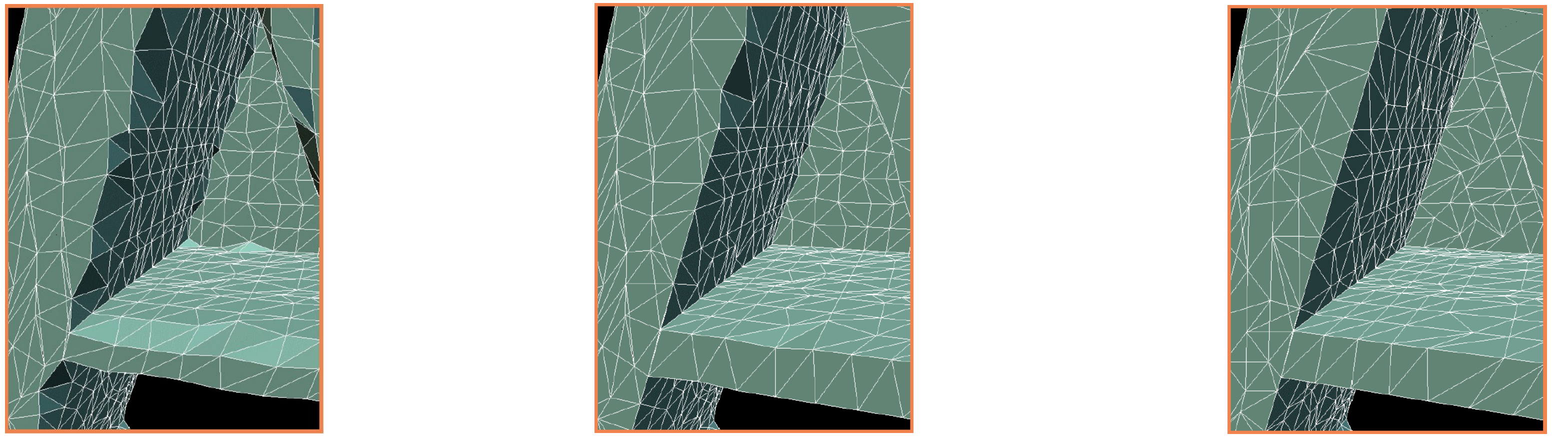
You simply sum up the N matrices Q_{q_i} component-wise and use it as shown here.

ILLINOIS

Comparison



average median error quadric

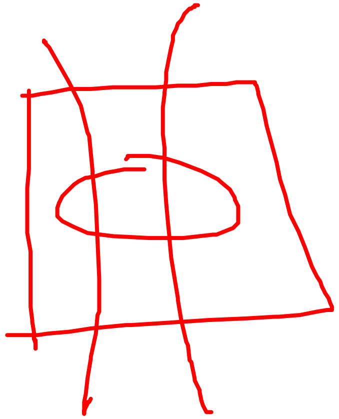


Vertex Clustering

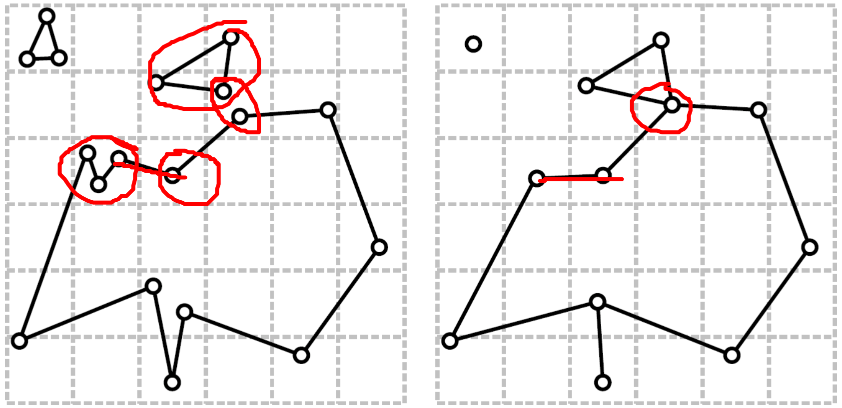
- Cluster Generation
- Computing a representative
- Mesh generation
  - Clusters $p \Leftrightarrow \{p_0,\ldots,p_n\}$, $q \Leftrightarrow \{q_0,\ldots,q_m\}$
  - Connect (p,q) if there was an edge (p_i,q_j)
- Topology changes

Vertex Clustering

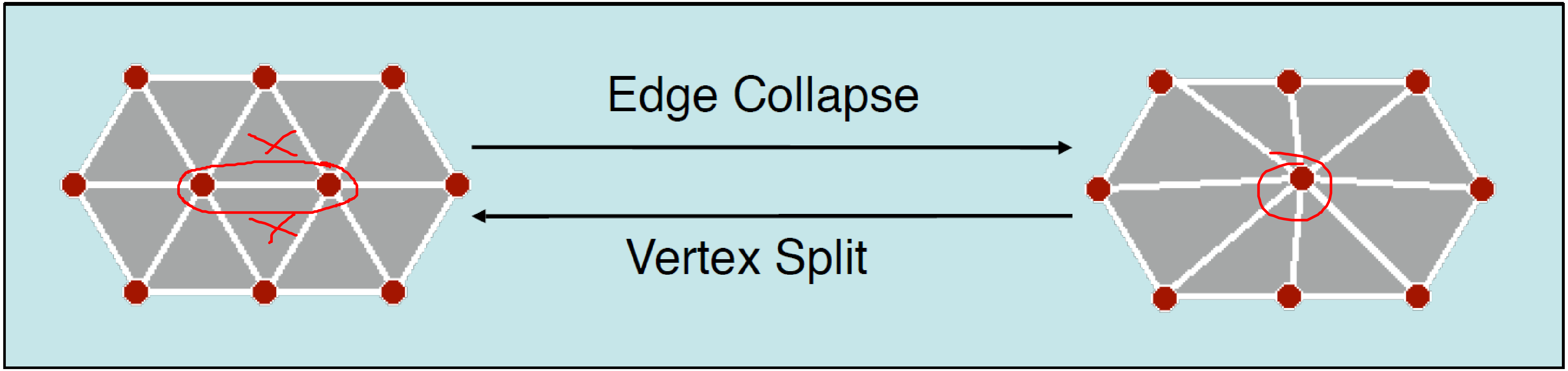
- Cluster Generation
- Computing a representative
- Mesh generation



- Topology changes
  - If different sheets pass through one cell
  - Not manifold



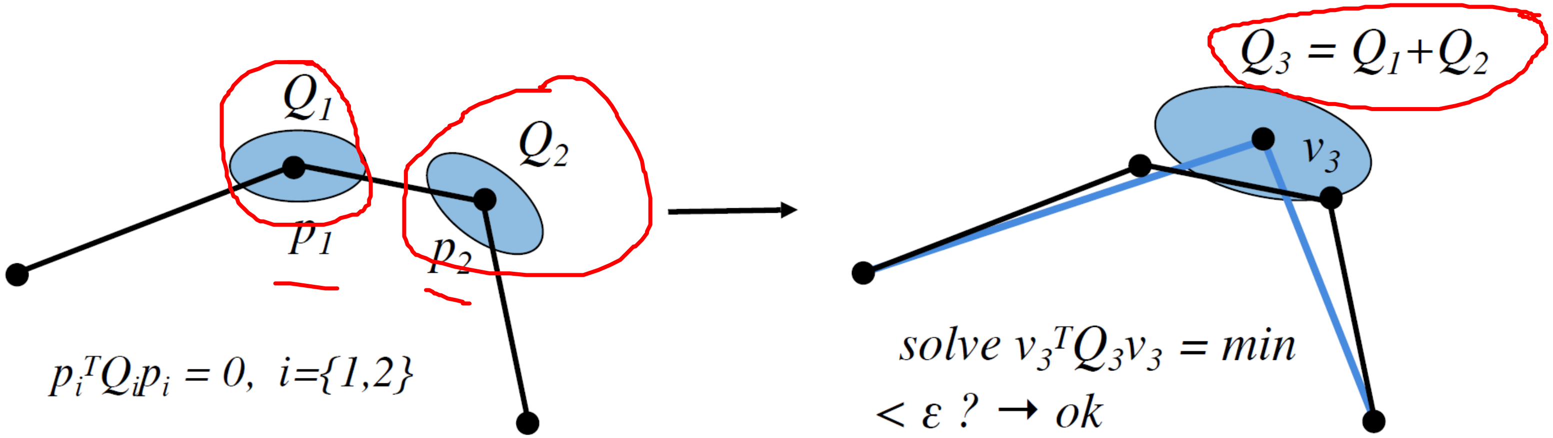
Incremental Simplification: Edge Collapse



- Merge two adjacent triangles
- Define new vertex position

Error Metrics

- Error quadrics [Garland, Heckbert 97]
  - Squared distance to planes at vertex
  - Can iteratively pick edge collapse that induces least error
  - Update error quadrics at each iteration



Comparison

- Vertex clustering
  - fast, but difficult to control simplified mesh
  - topology changes, non-manifold meshes
  - global error bound, but often not close to optimum
- Iterative simplification with quadric error metrics
  - good trade-off between mesh quality and speed
  - explicit control over mesh topology
  - restricting normal deviation improves mesh quality

Out-of-core Decimation

- Handle very large data sets that do not fit into main memory
- Key: Avoid random access to mesh data structure during simplification
- Examples
  - Garland, Shaffer: *A Multiphase Approach to Efficient Surface Simplification*, IEEE Visualization 2002
  - Wu, Kobbelt: *A Stream Algorithm for the Decimation of Massive Meshes*, Graphics Interface 2003

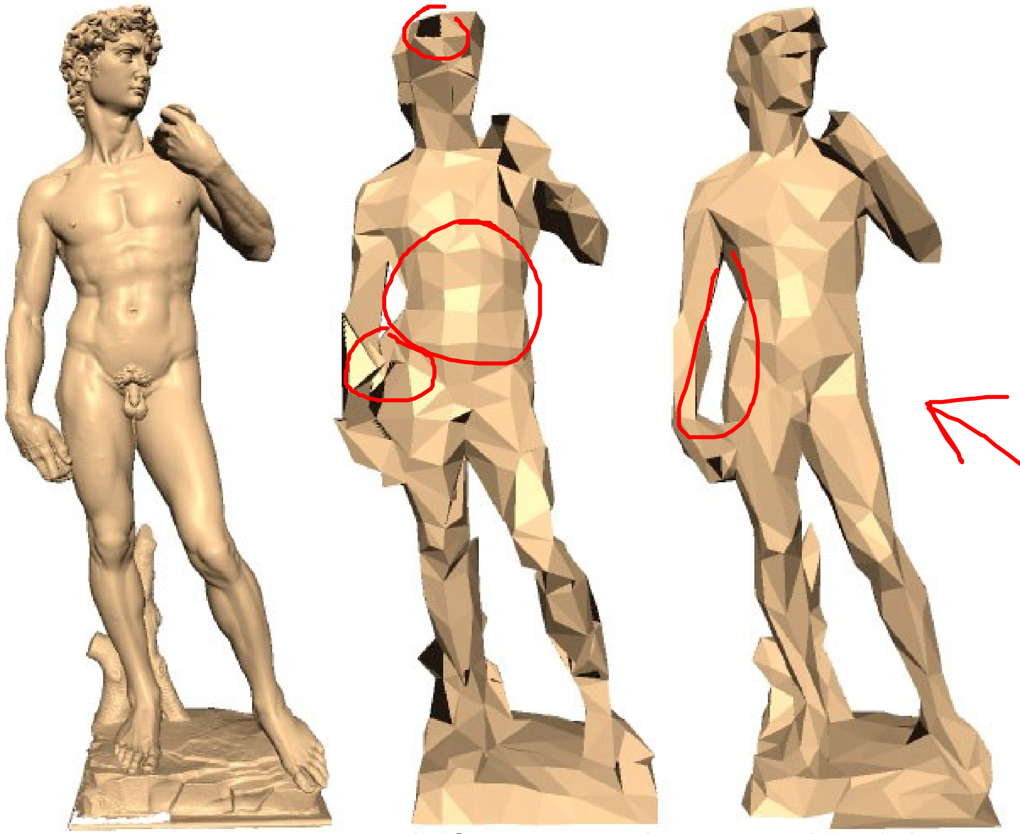
Multiphase Simplification

1. Phase: Out-of-core clustering
   - compute accumulated error quadrics and vertex representative for each cell of uniform voxel grid

2. Phase: In-core iterative simplification
   - use accumulated quadrics from clustering phase
   - iteratively contract edge of smallest cost

→ achieves a coupling between the two phases

Multiphase Simplification

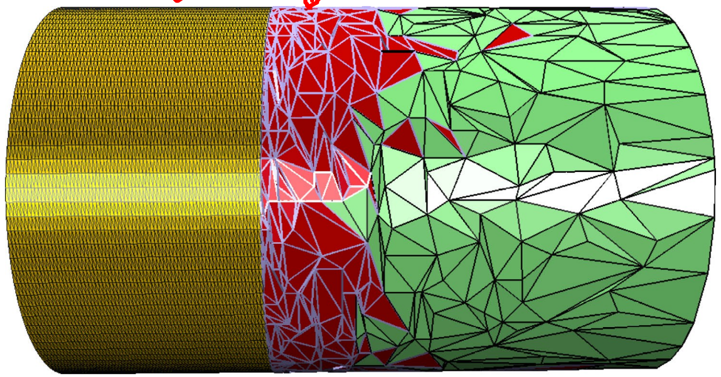


Original (8 million triangles) | Uniform clustering (1157 triangles) | Multiphase (1000 triangles)

Garland, Shaffer: *A Multiphase Approach to Efficient Surface Simplification*, IEEE Visualization 2002

Out-of-core Decimation

- Streaming approach based on edge collapse operations using QEM
- Pre-sorted input stream allows fixed-sized active working set independent of input and output model complexity



See also
Martin Isenburg, Peter Lindstrom:
Streaming Meshes.
IEEE Visualization 2005

Wu, Kobbelt: *A StreamAlgorithm for the Decimation of Massive Meshes*, Graphics Interface 2003