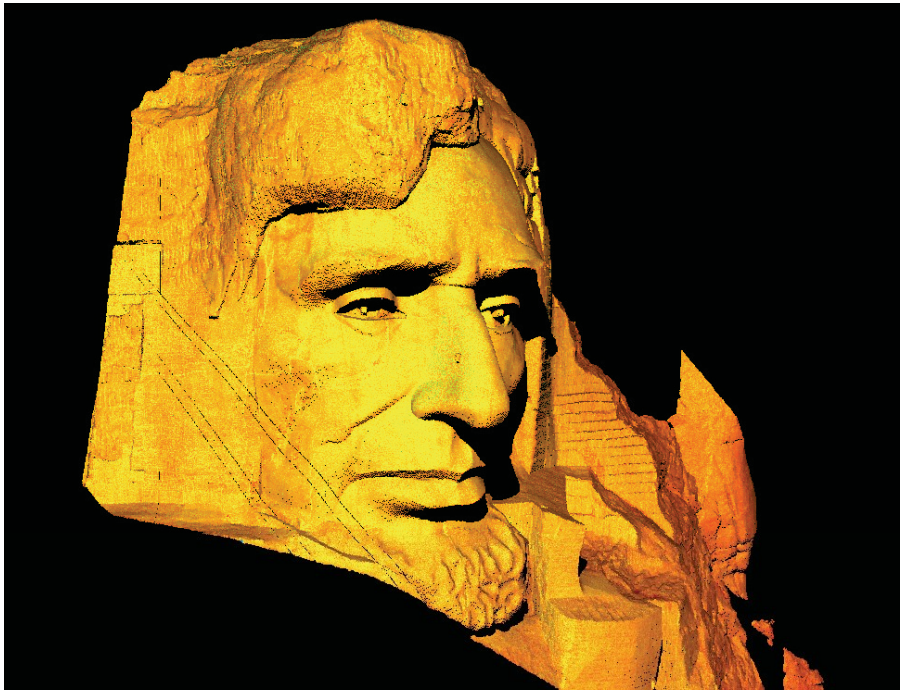
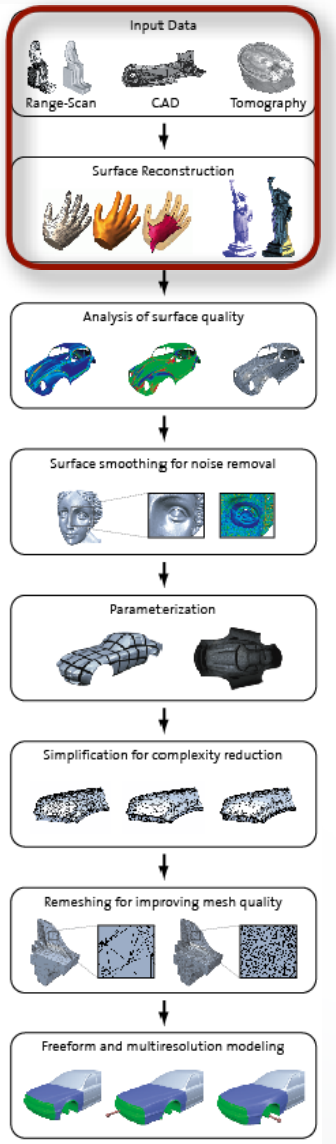
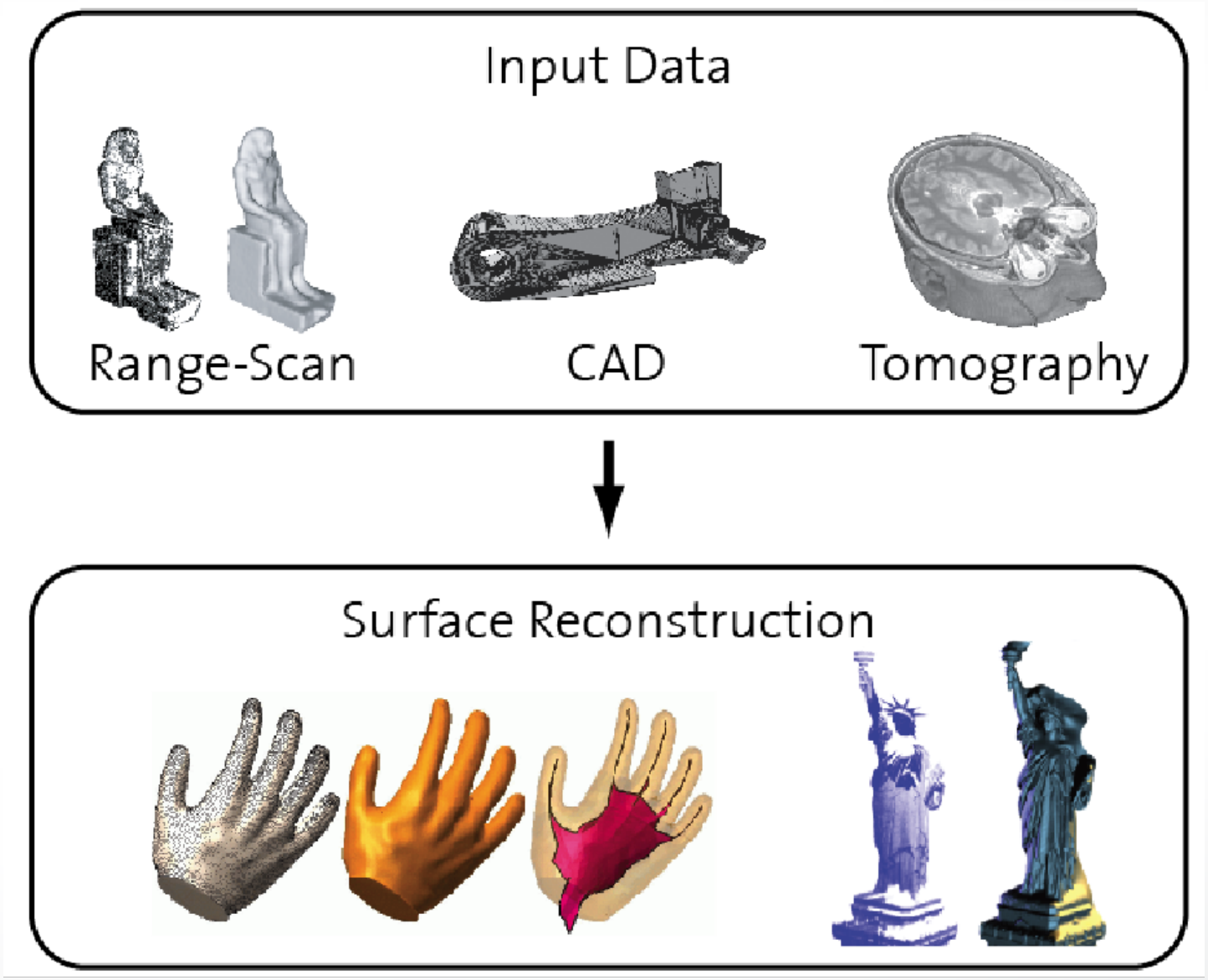
3D Capture and Reconstruction

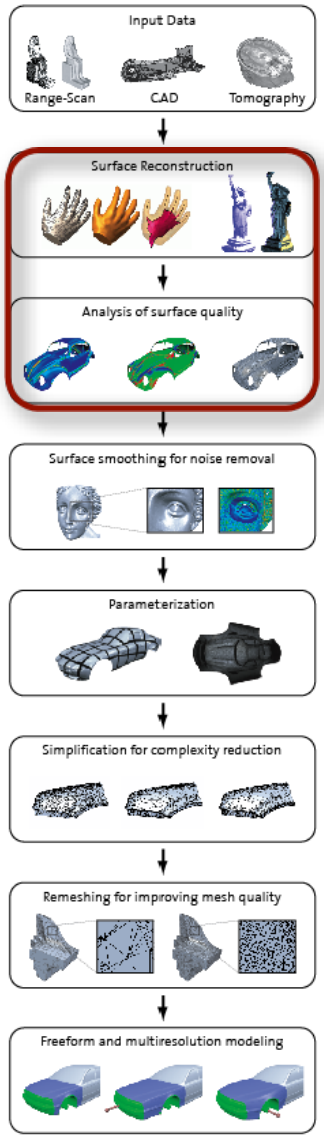


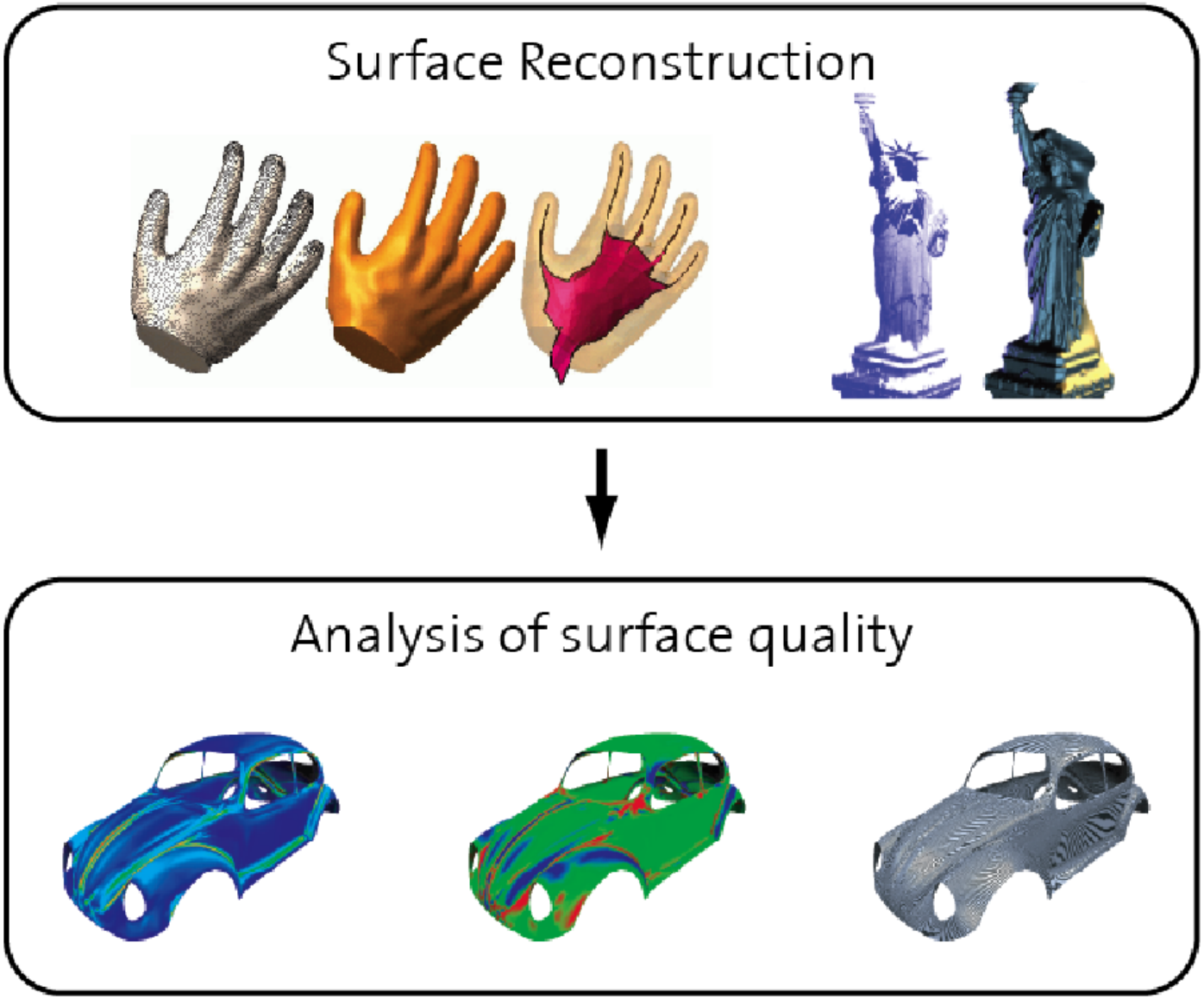
Geometry Processing Pipeline



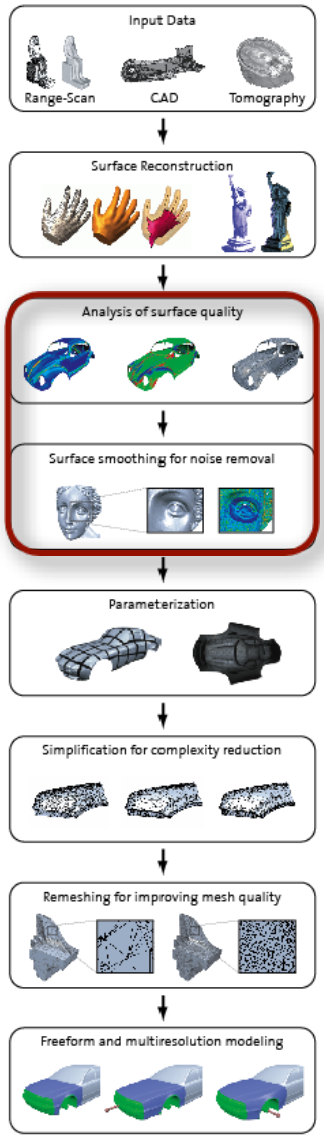


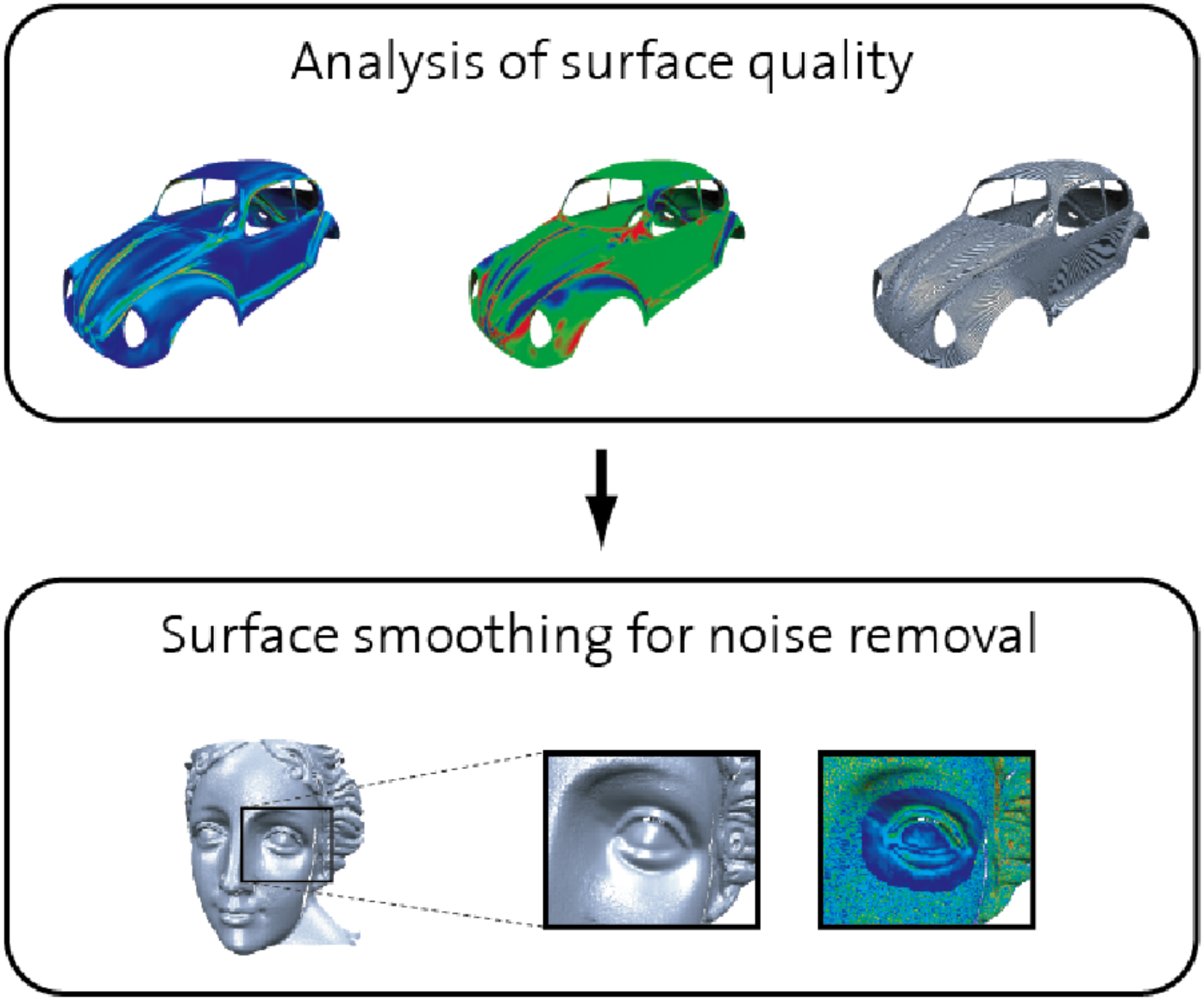
Geometry Processing Pipeline



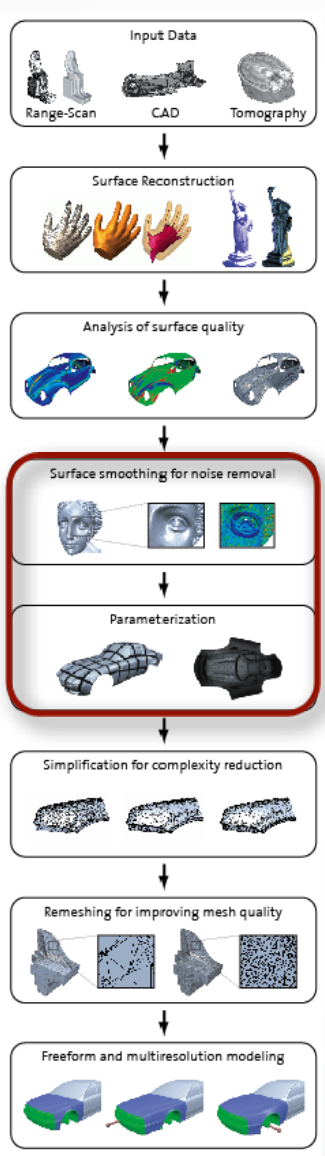


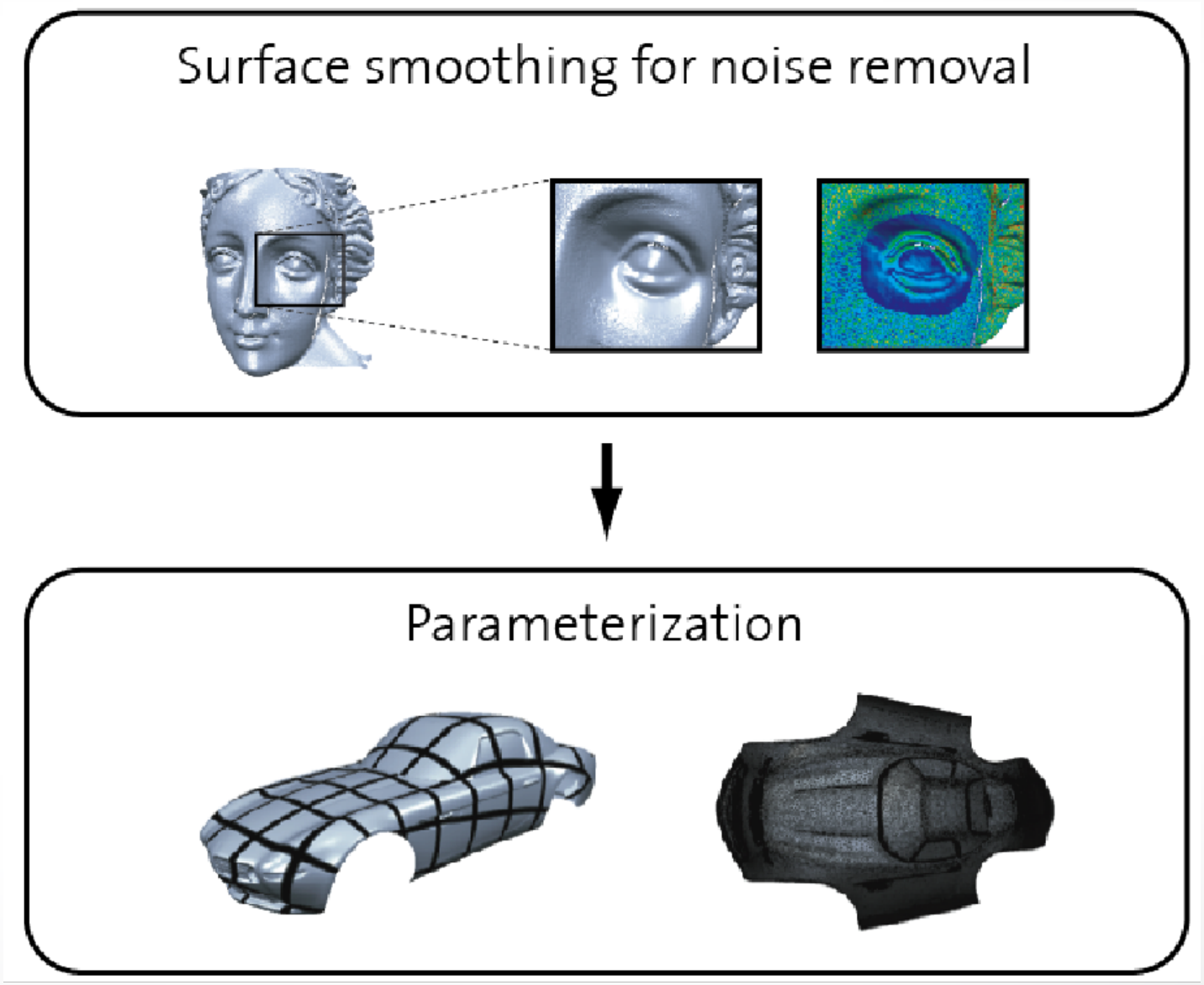
Geometry Processing Pipeline



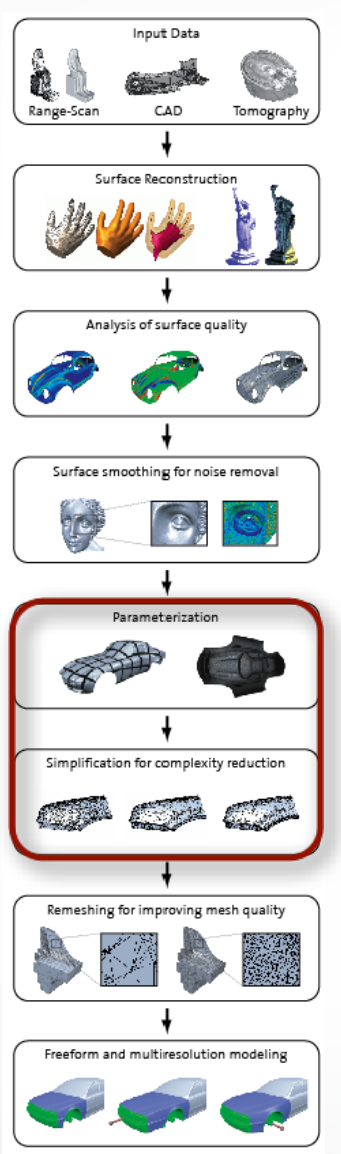


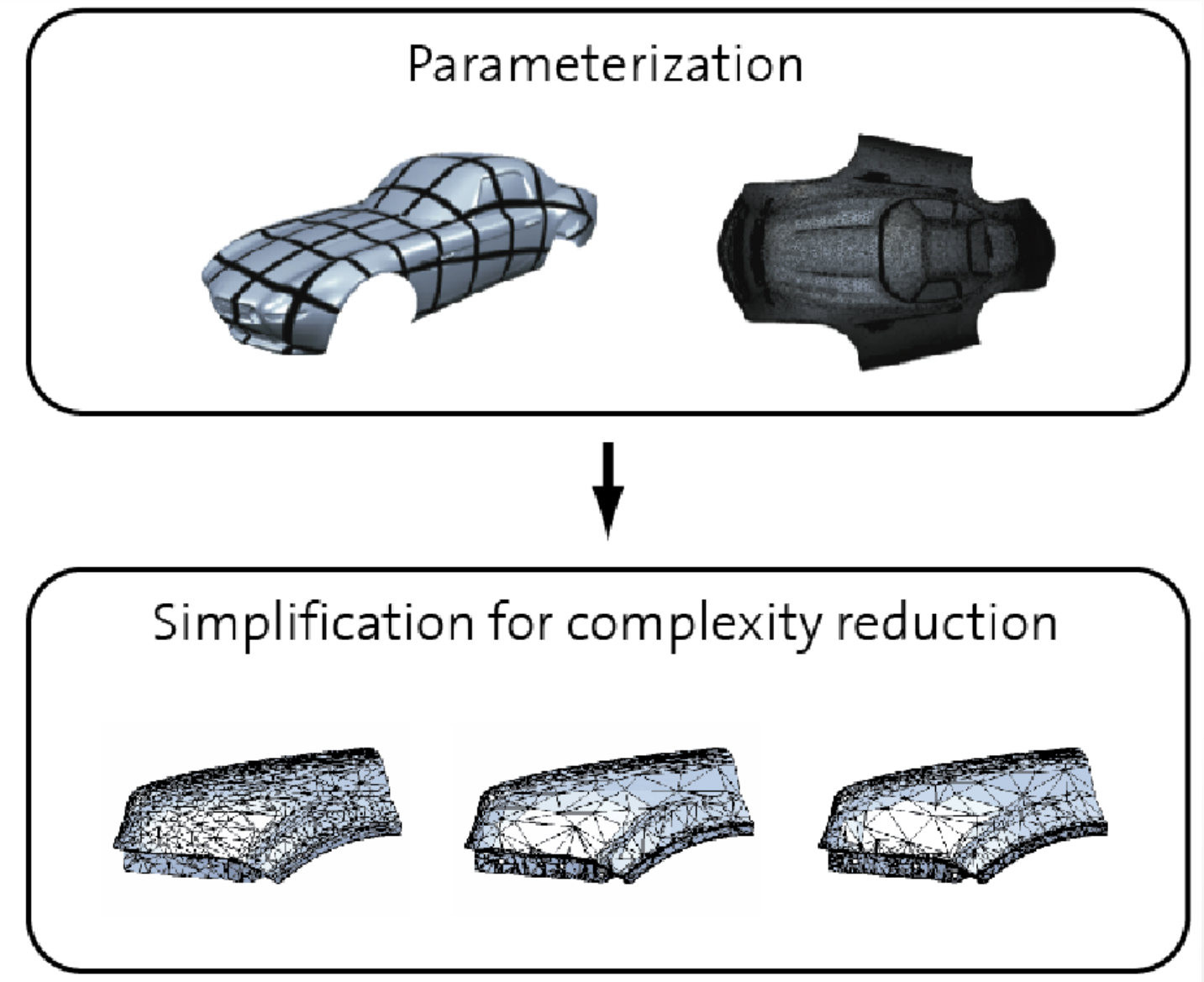
Geometry Processing Pipeline



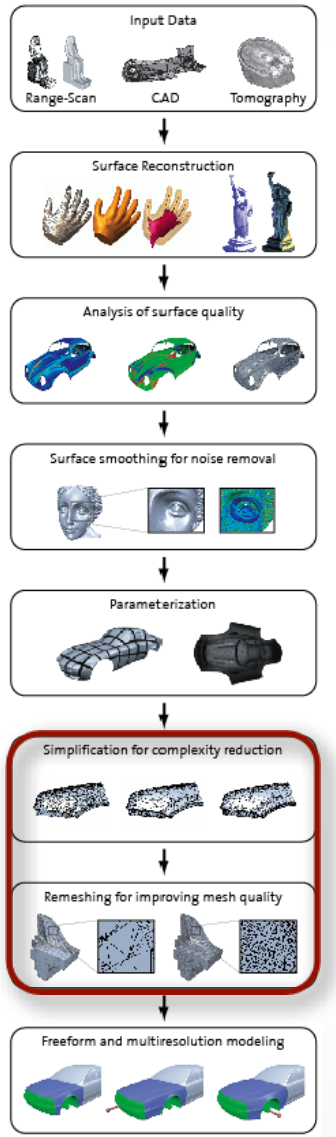


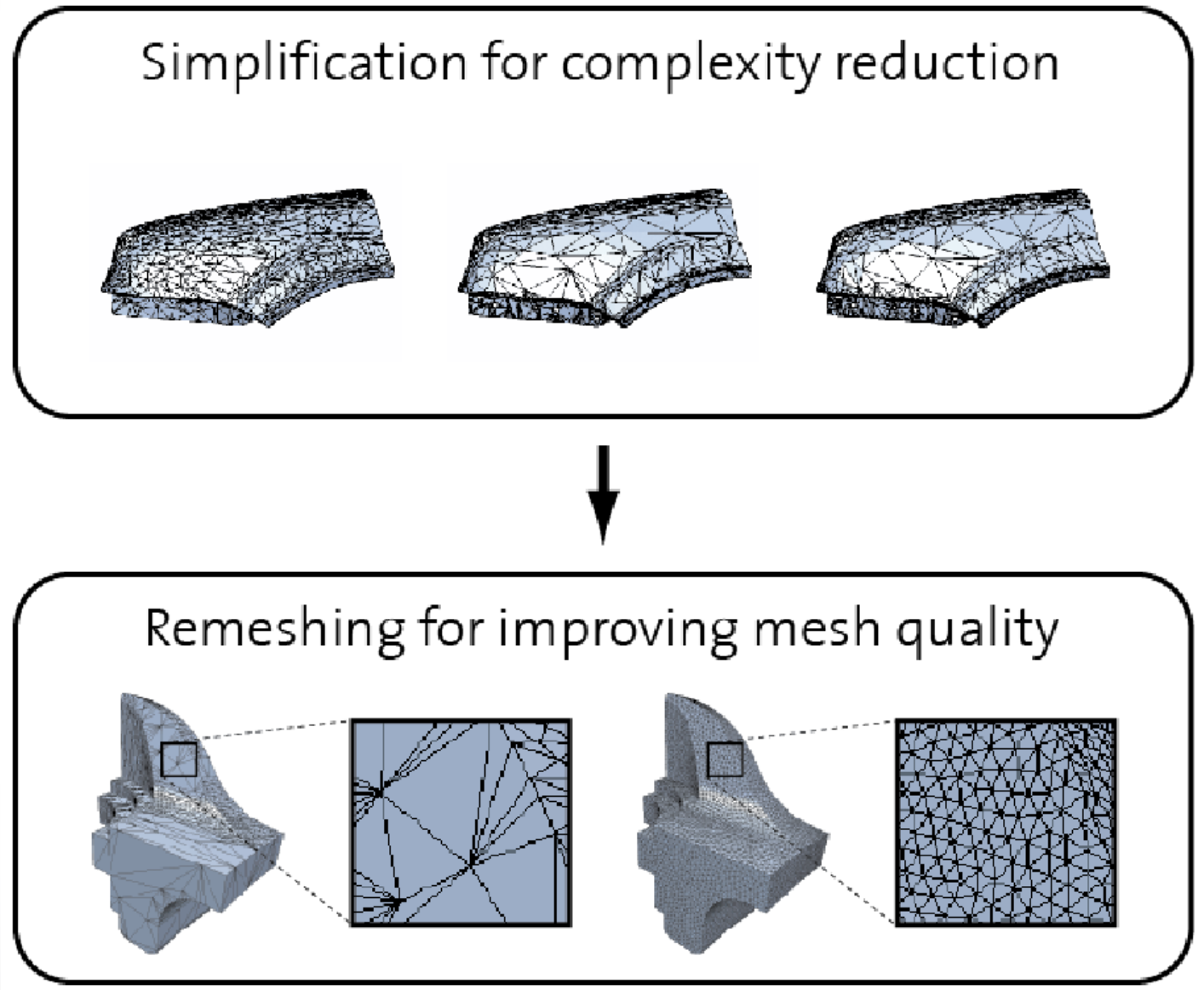
Geometry Processing Pipeline



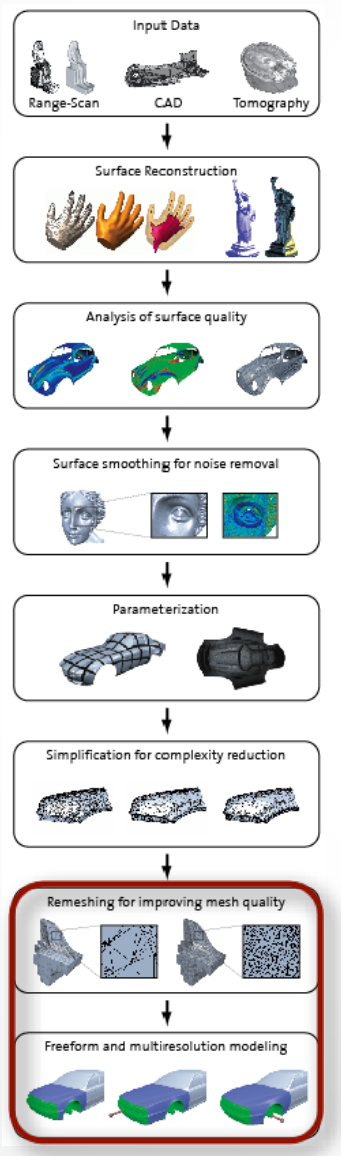


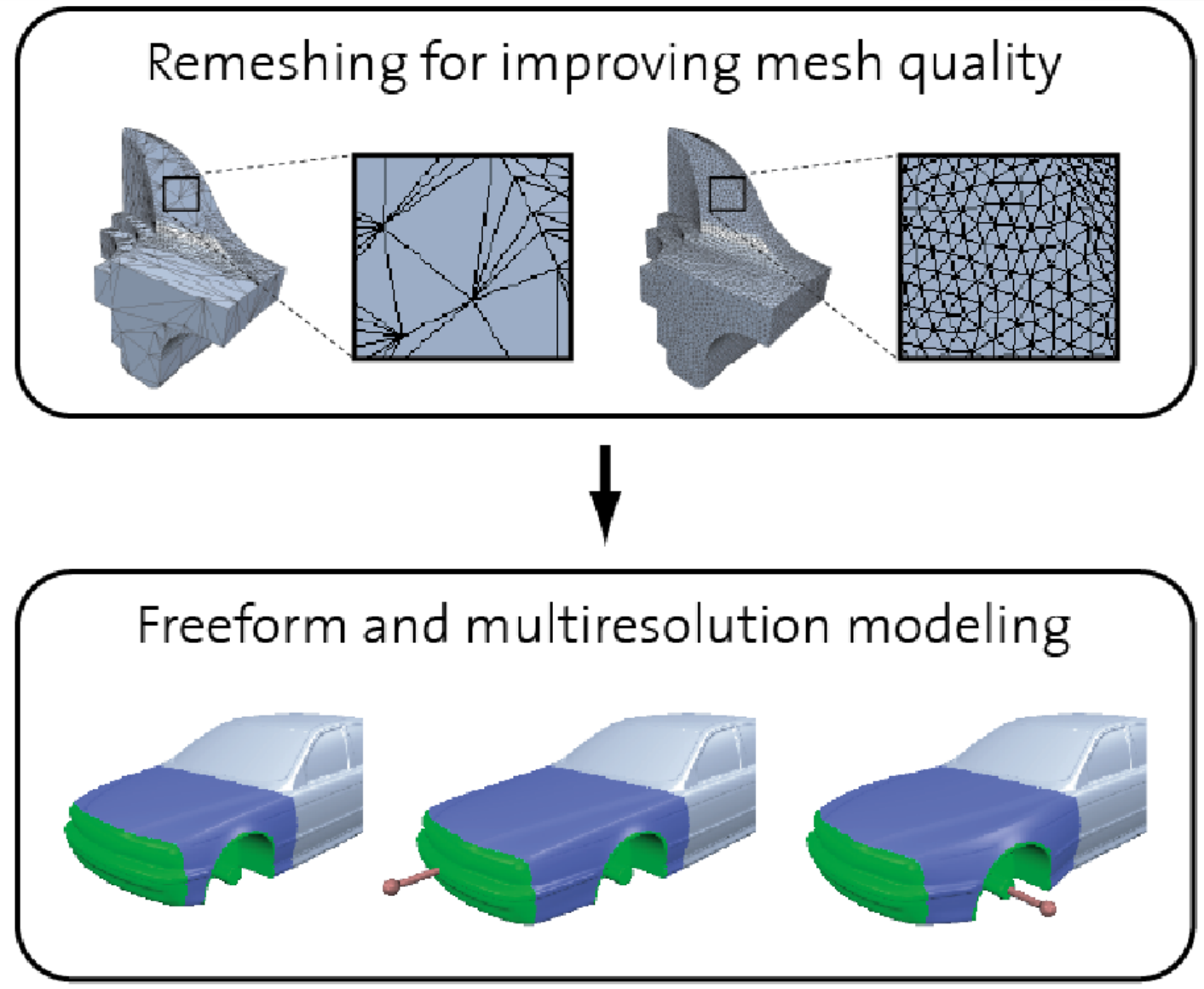
Geometry Processing Pipeline



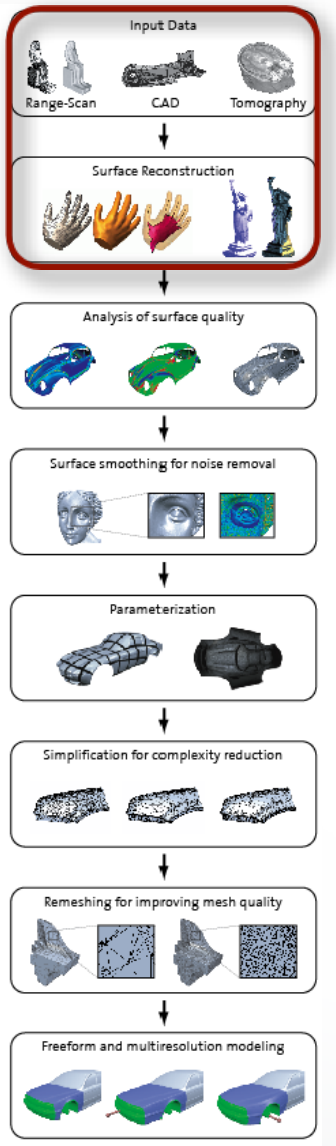


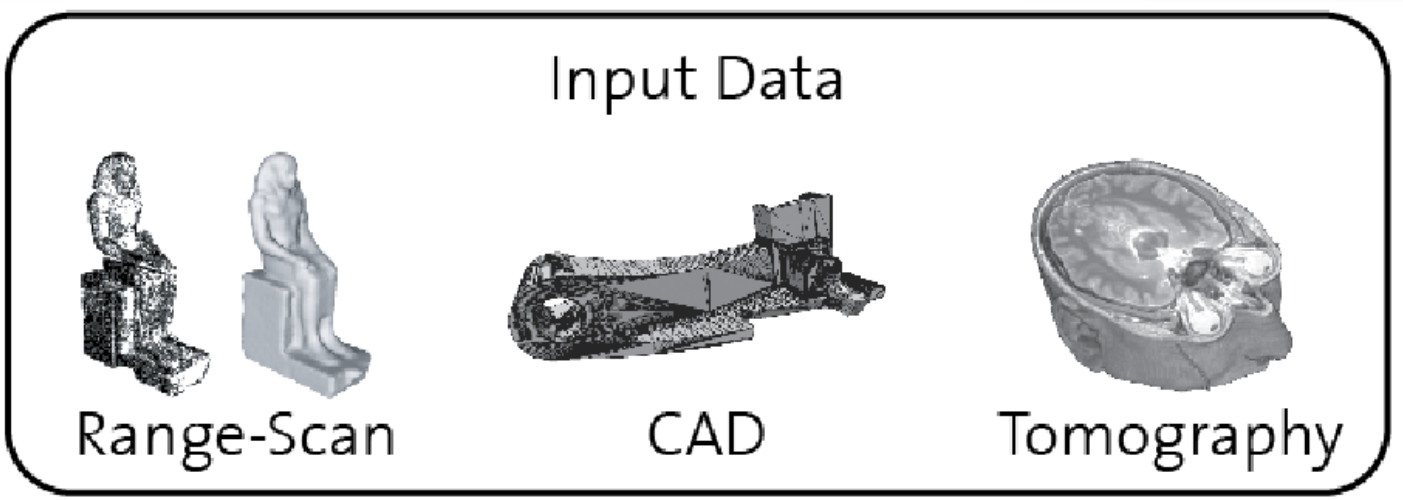
Geometry Processing Pipeline





Input Data

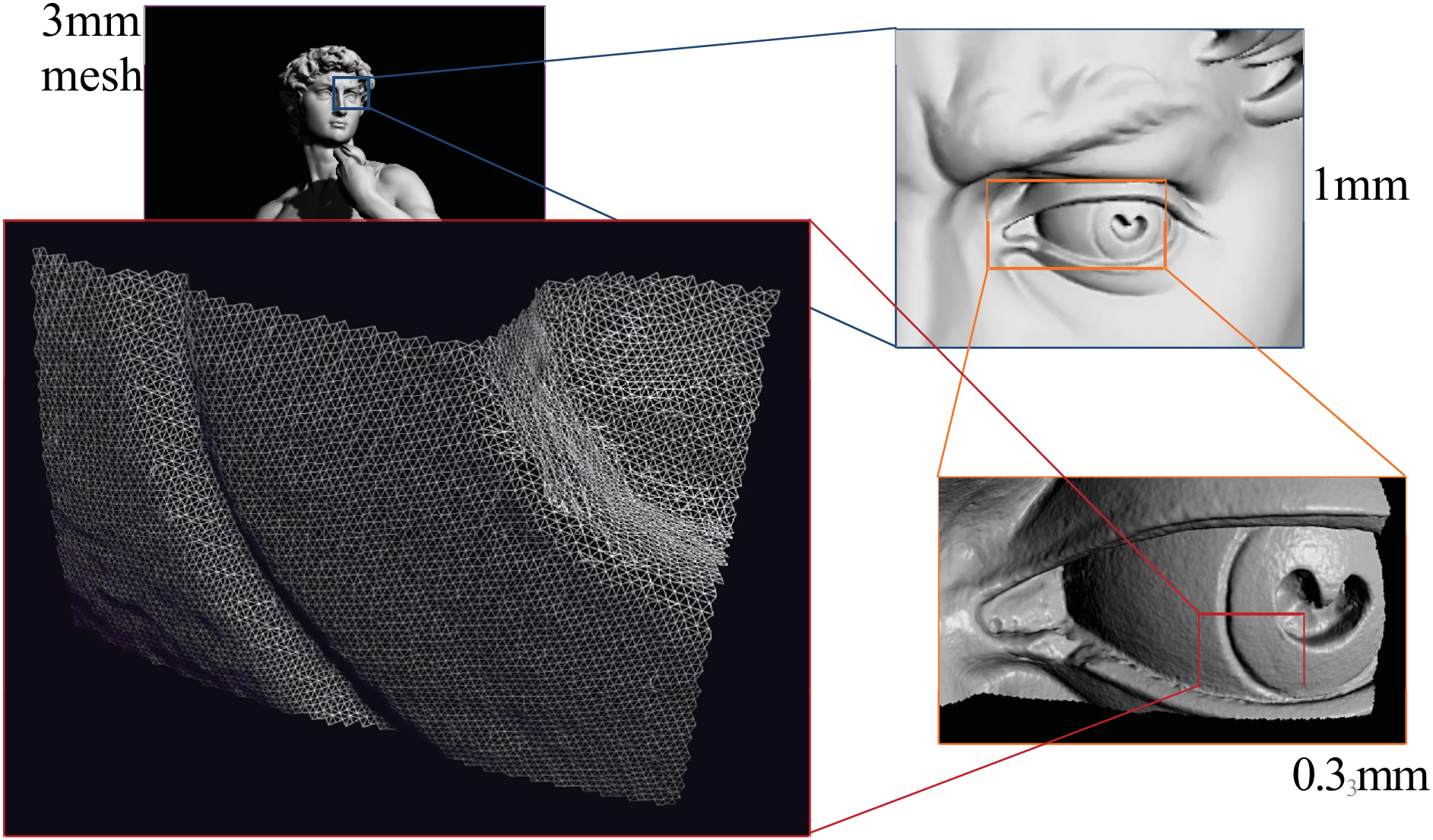

Input Data
Range-Scan
CAD
Tomography
Surface Reconstruction
Analysis of surface quality
Surface smoothing for noise removal
Parameterization
Simplification for complexity reduction
Remeshing for improving mesh quality
Freeform and multiresolution modeling


Input Data
Range-Scan
CAD
Tomography

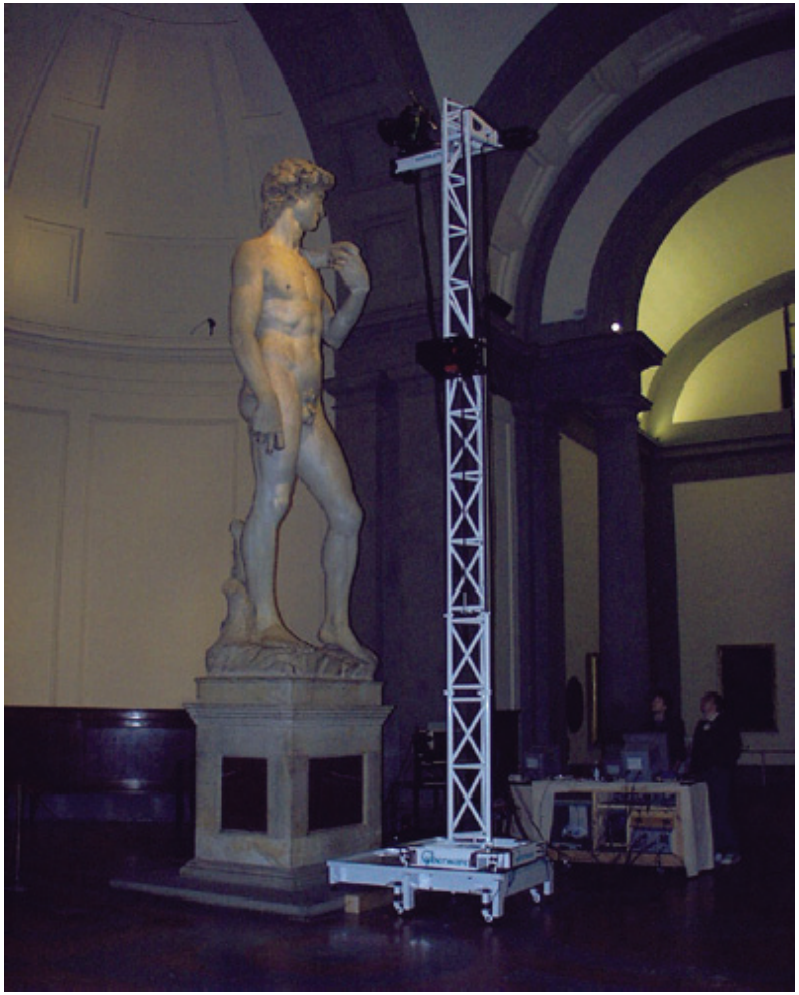
(Stanford's) Digital Michelangelo Project

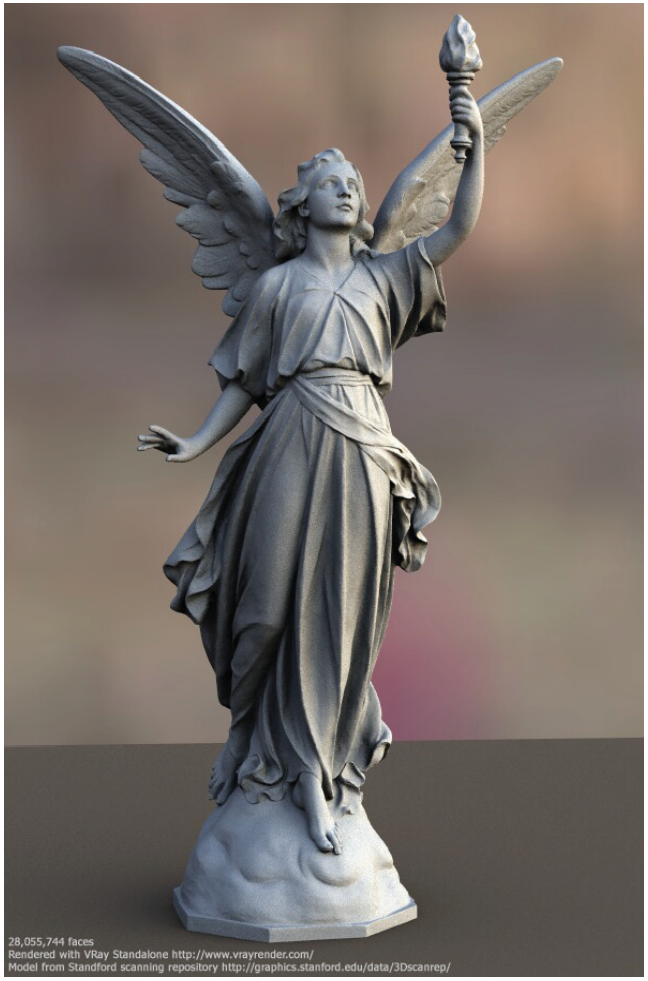
4G sample points → 8M triangles

(Stanford's) Digital Michelangelo Project

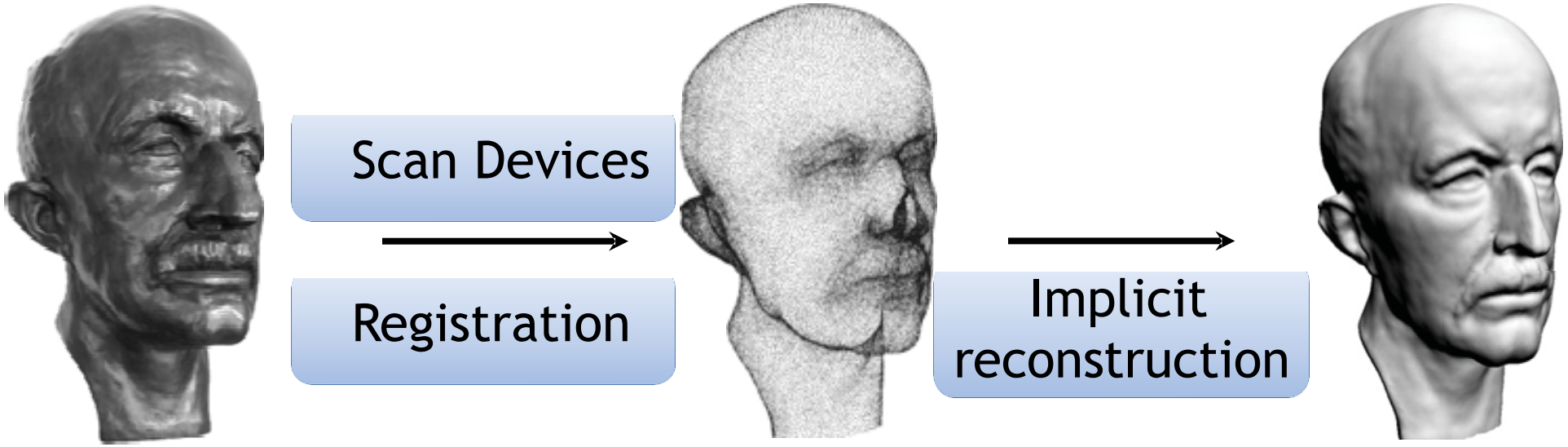


Local Sightseeing





Surface Reconstruction



physical model

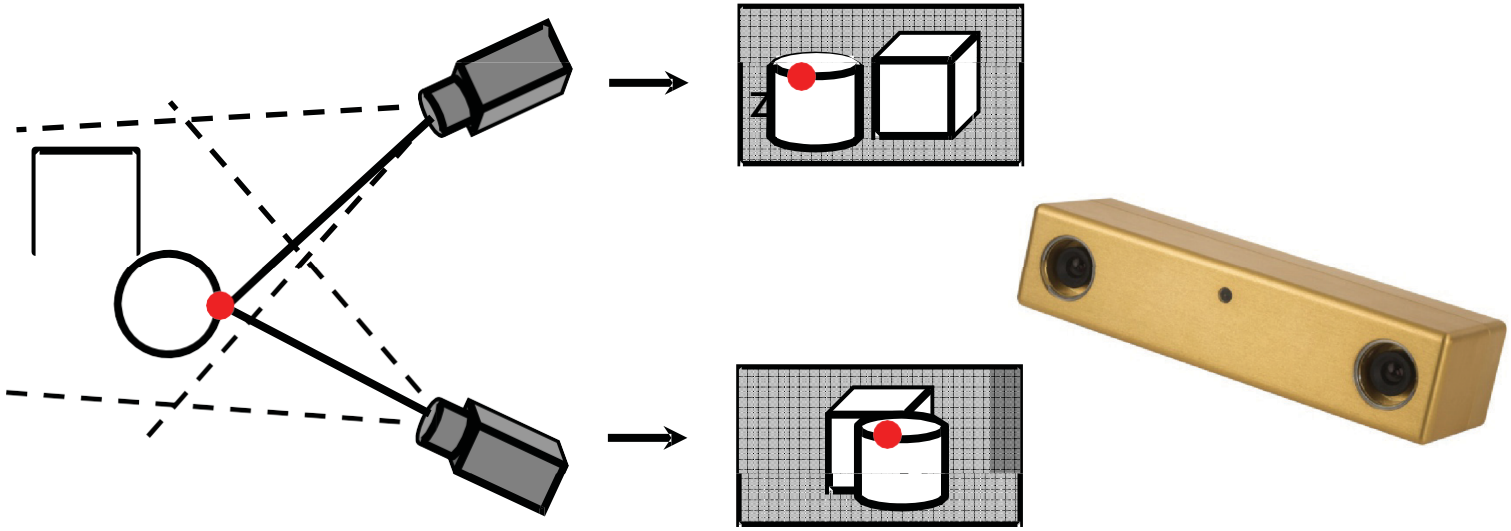
acquired point cloud

reconstructed model

Range Scanning Systems

Passive: Stereo matching

Find and match features in both images

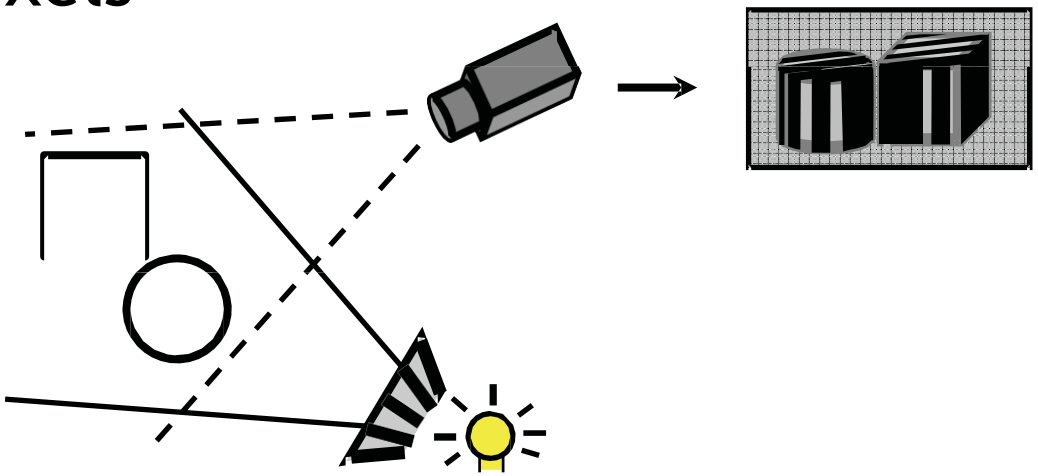


Problem: Needs features to match

Range Scanning Systems

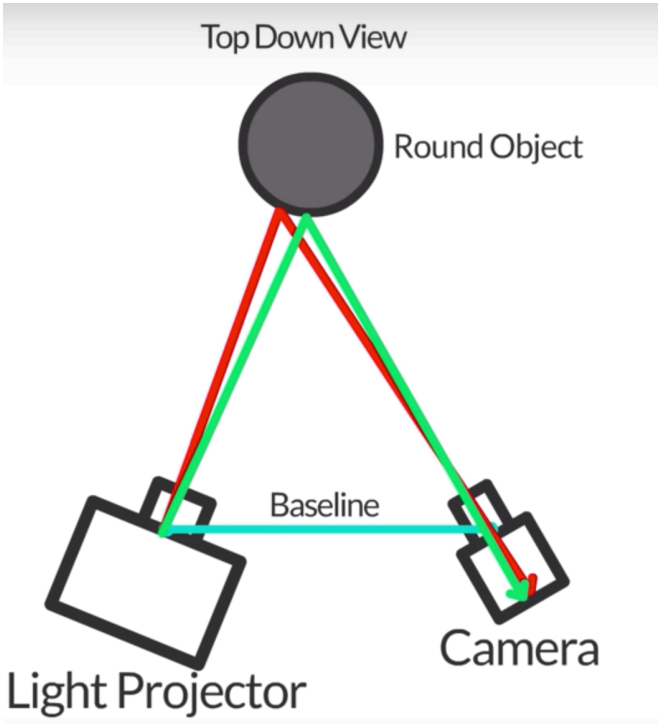
Active: Structured light

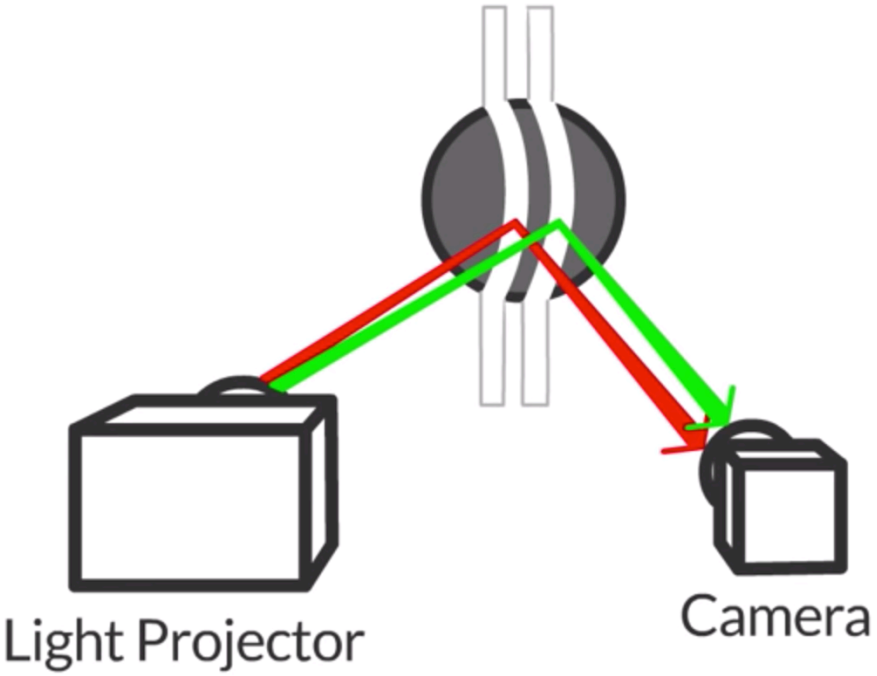
Project special b/w patterns to identify pixels



Problematic for materials / textures having strong color differences.

Range Scanning Systems





The object distorts the light lines. This light distortion or curvature, is captured by the camera and used to calculate object depth and structure

Range Scanning Systems

Active: Laser scanning, time-of-flight

Send laser pulse, time how long it takes to return

$$r = \frac{1}{2} c \Delta t$$

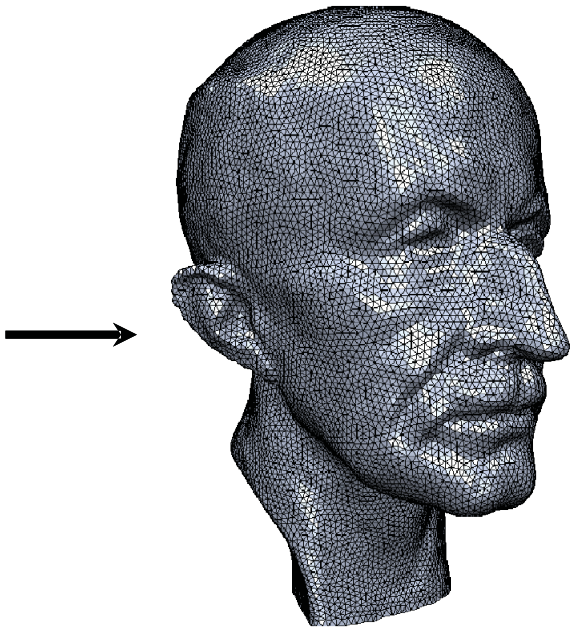
Accuracy depends on how precisely we can measure time

Operates on large distances – good for buildings, large scale objects

Range Scans

- Scanning generates multiple range images
- Each contain 3D points for different parts of the model in local coordinates of the scanner
- Need registration to one point cloud

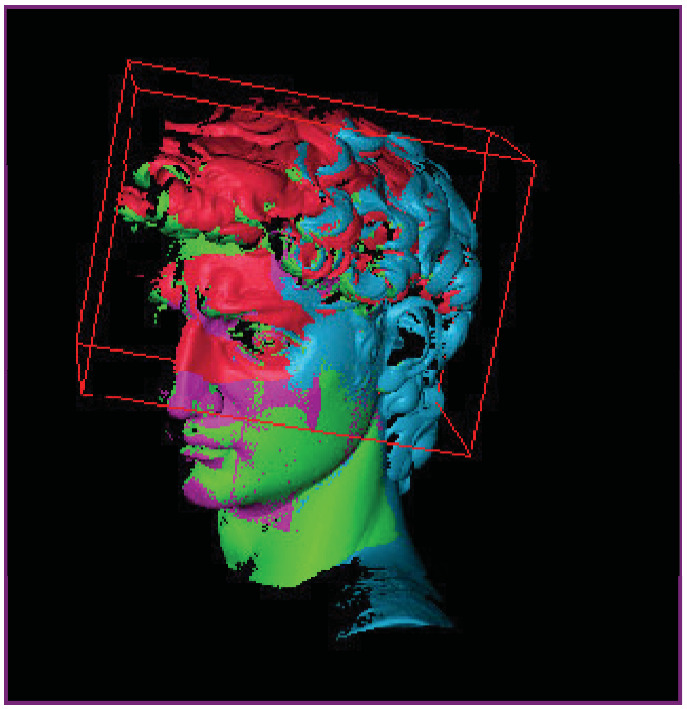
Goal



set of raw scans

reconstructed model

Range Processing Pipeline

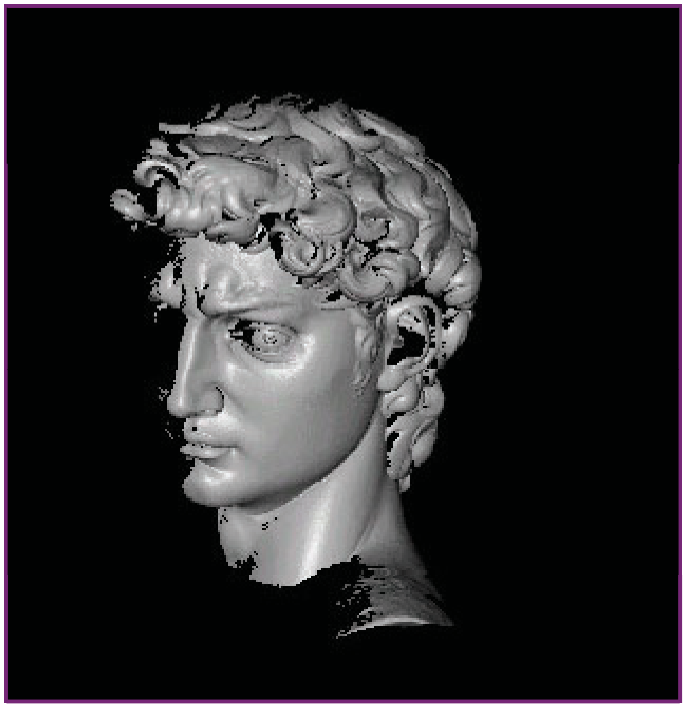


Steps

1. Initial registration
2. Pairwise registration
3. Global relaxation to spread out error
4. Generate surface

Slides from: http://www.cs.princeton.edu/~bjbrown/iccv05_course/

Range Processing Pipeline

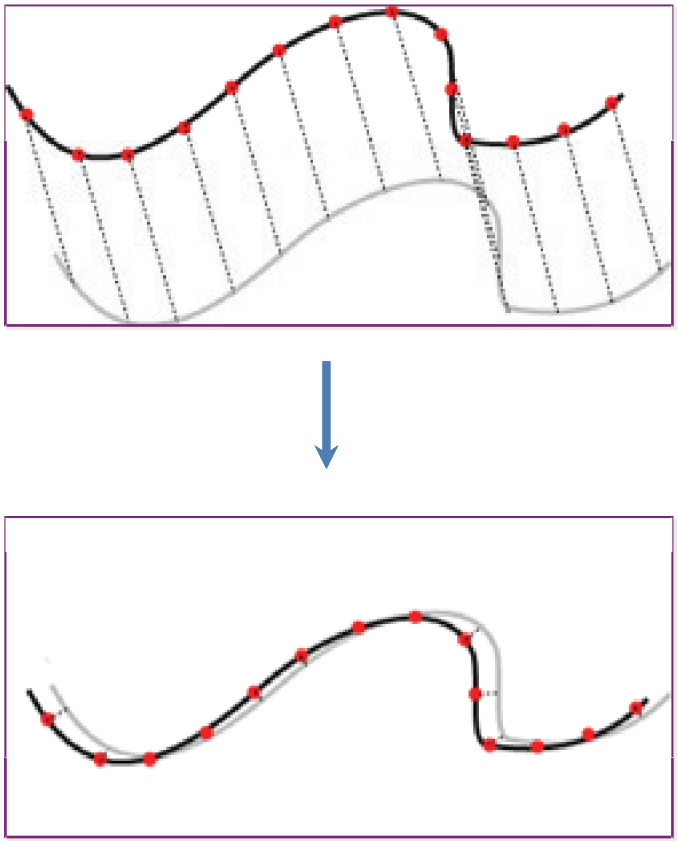


Steps

1. Initial registration
2. Pairwise registration
3. Global relaxation to spread out error
4. Generate surface

Slides from: http://www.cs.princeton.edu/~bjbrown/iccv05_course/

Range Processing Pipeline



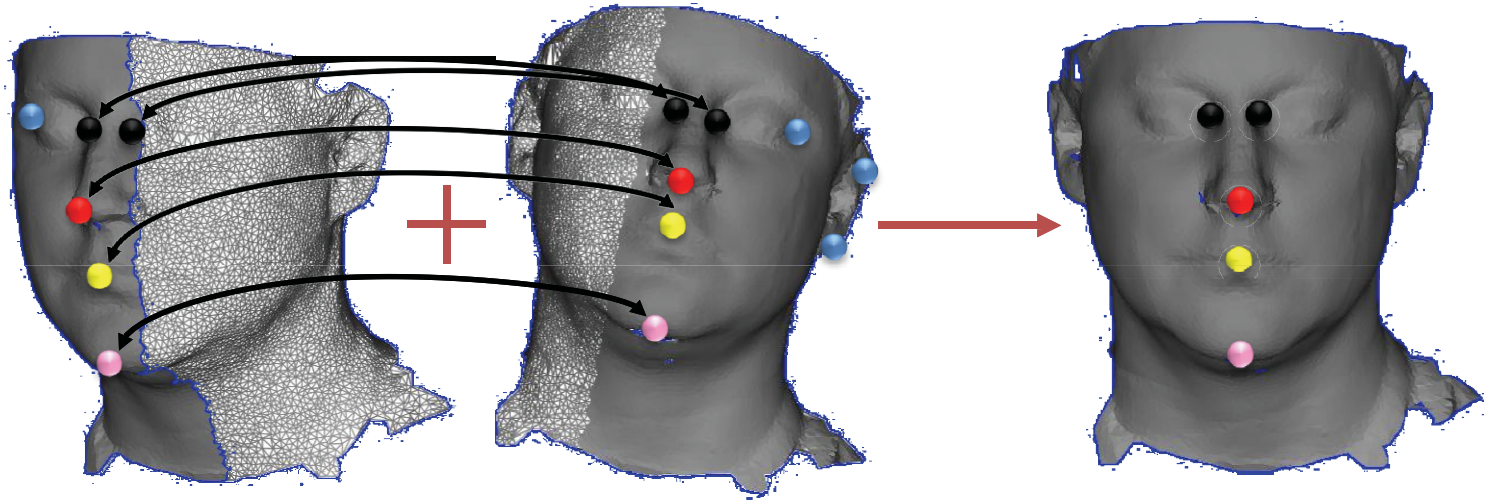
Steps

1. Initial registration
2. Pairwise registration
3. Global relaxation to spread out error
4. Generate surface

Slides from: http://www.cs.princeton.eu/~bjbrown/iccv05_course/

Initial Registration

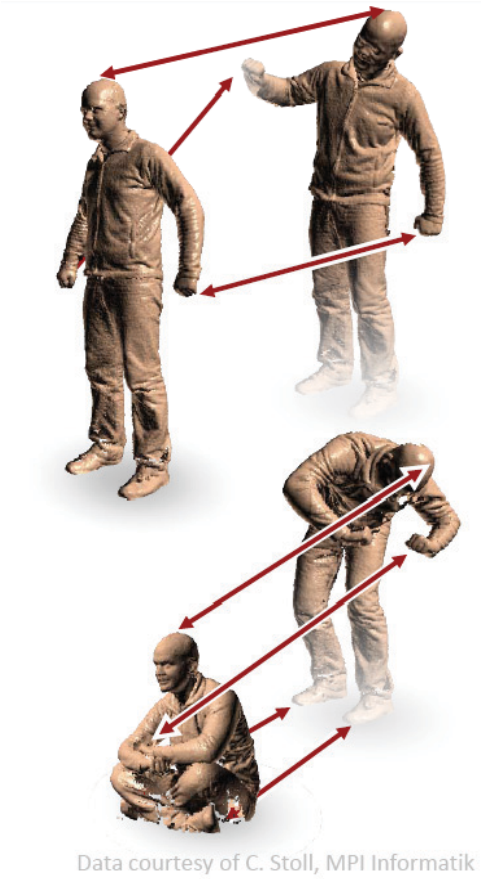
- Options:
  - Scans are calibrated – scan on a turntable
  - Manual feature selection, global coarse alignment
  - Automatic feature selection and matching



Partially Overlapped Scans

Aligned Scan

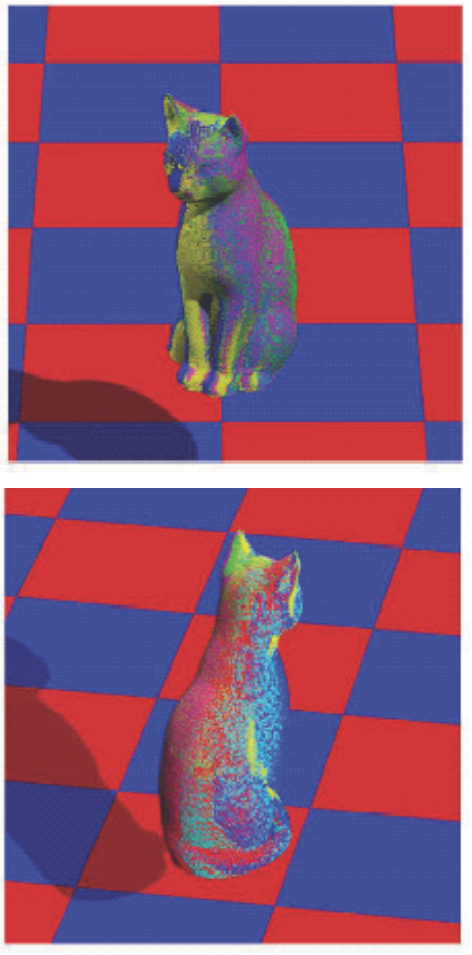
Non-Rigid Registration



Data courtesy of C. Stoll, MPI Informatik

Video

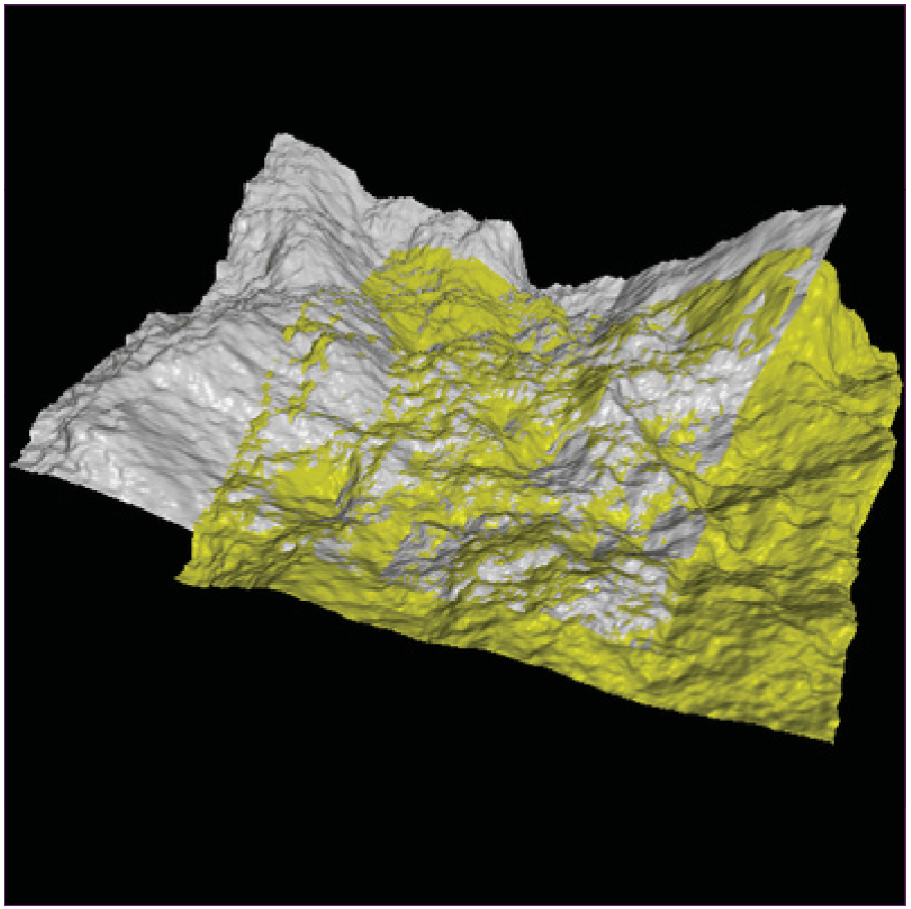
Pairwise Rigid Registration



Images from: "Geometry and convergence analysis of algorithms for registration of 3D shapes" by Pottman

Pairwise Rigid Registration Goal

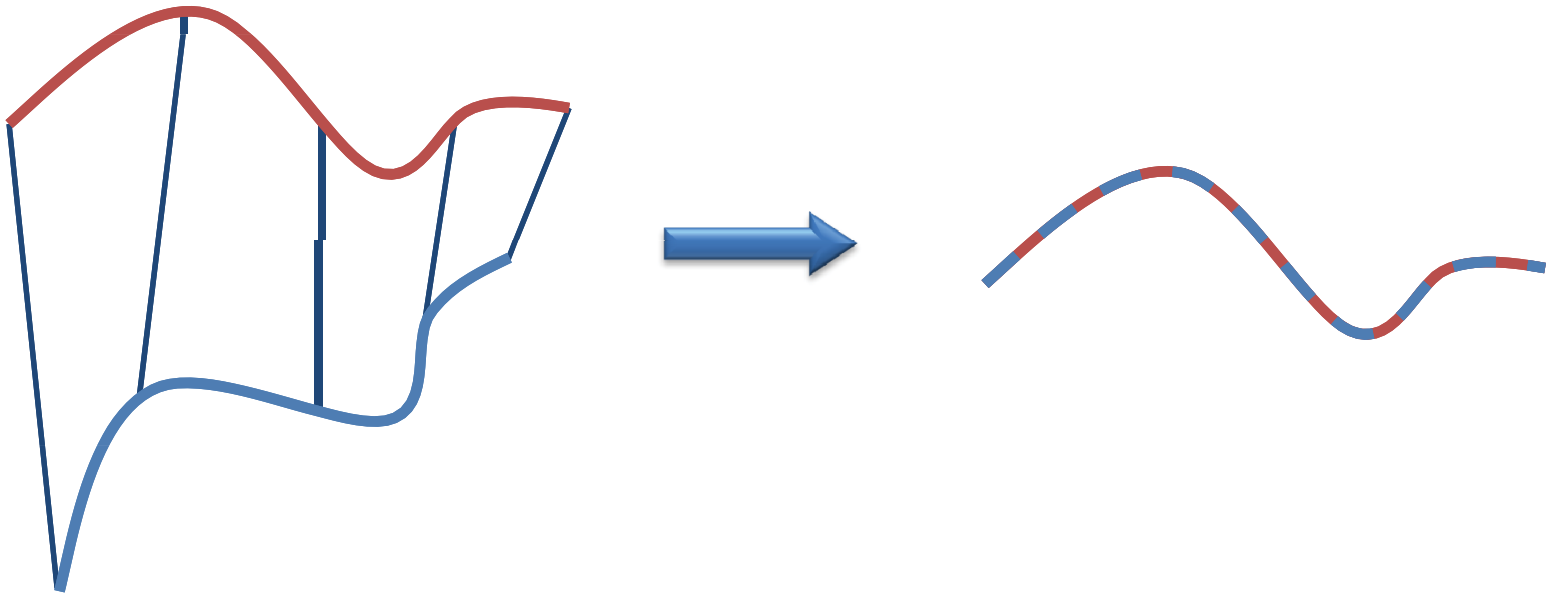
Align two partially-overlapping point clouds given initial guess



Slides from: http://www.cs.princeton.edu/~bjbrown/iccv05_course/

Aligning 3D Data

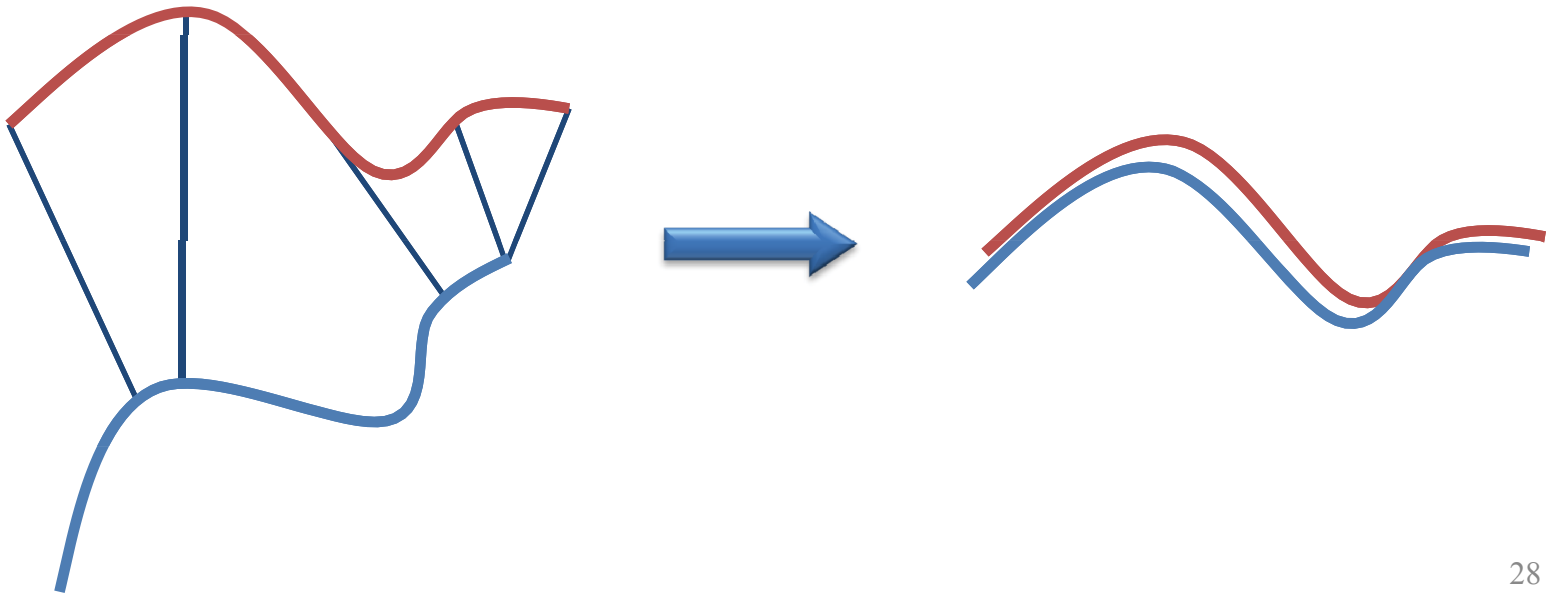
If correct correspondences are known, can find correct relative rotation/ translation



Slides from: http://www.cs.princeton.edu/~bjbrown/iccv05_course/

Aligning 3D Data

- How to find correspondences:
  User input? Feature detection? Signatures?
- Alternative: assume closest points correspond



Slides from: http://www.cs.princeton.edu/~bjbrown/iccv05_course/

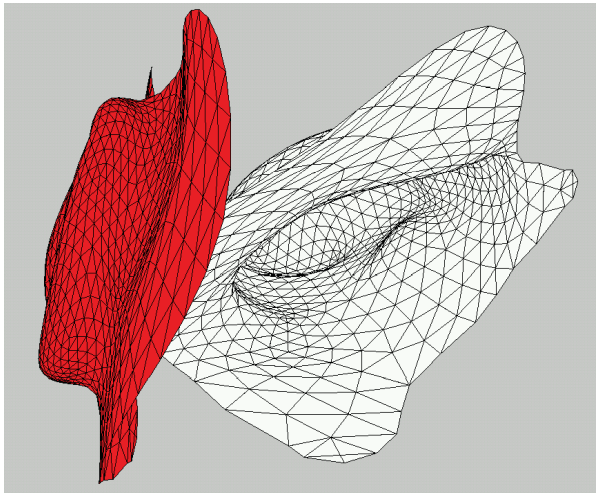
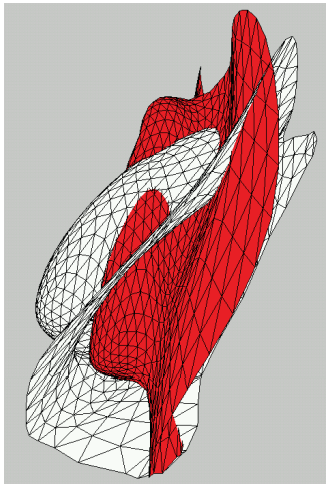
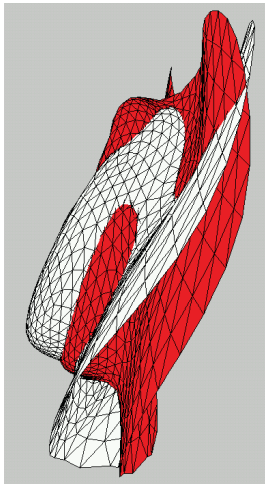
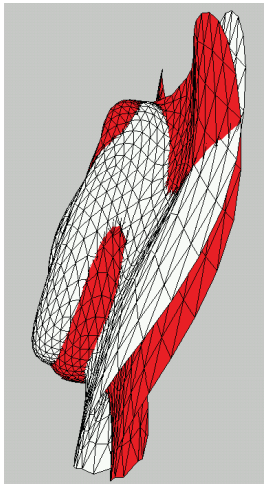
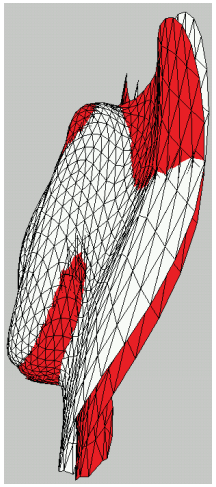
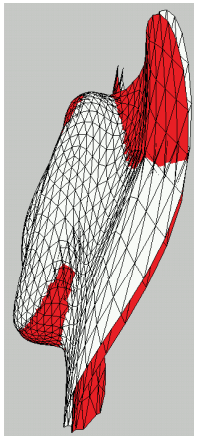
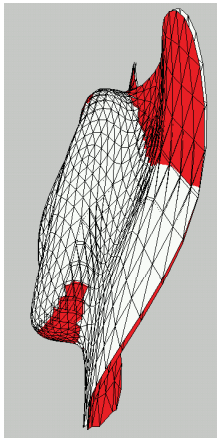
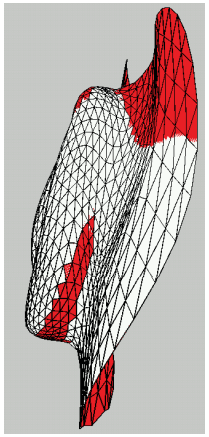
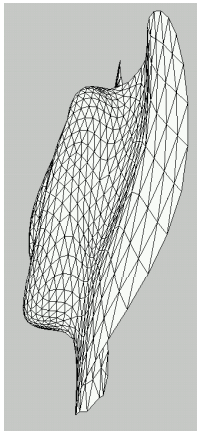
Iterative Closest Point (ICP)

- Given two scans *P* and *Q*.

- Iterate:
  1. Find some pairs of closest points $(\mathbf{p}_i, \mathbf{q}_i)$
  2. Find rotation $\mathbf{R}$ and translation $\mathbf{t}$ to minimize

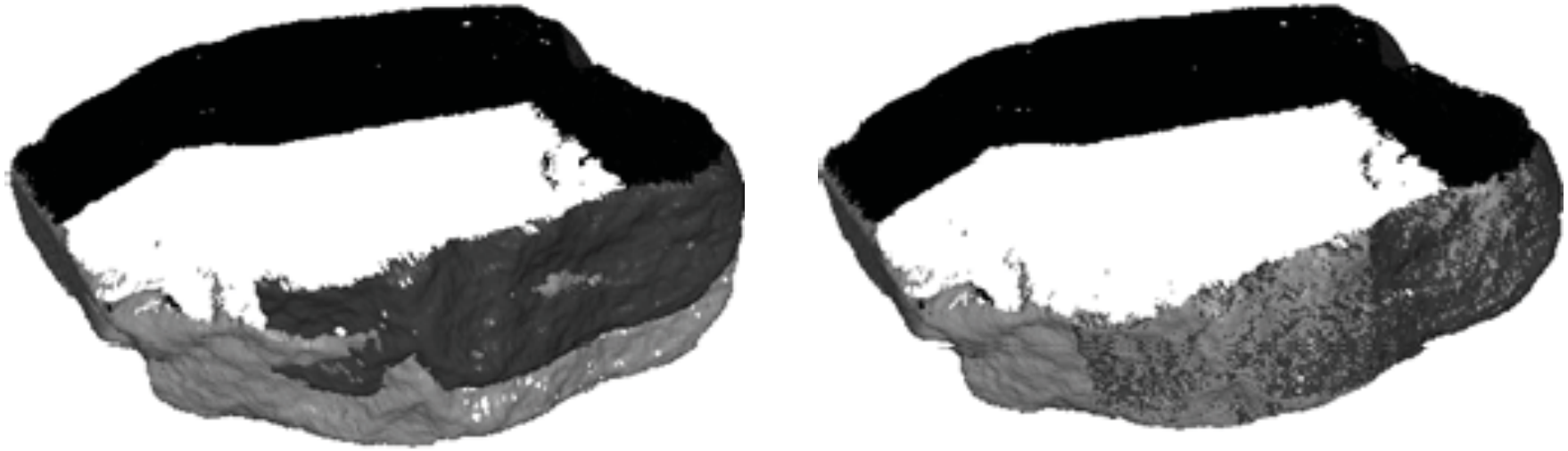
$$\min_{\mathbf{R},\mathbf{t}} \sum_i \|\mathbf{p}_i - \mathbf{R}\mathbf{q}_i - \mathbf{t}\|^2$$

Example

Global Registration Goal

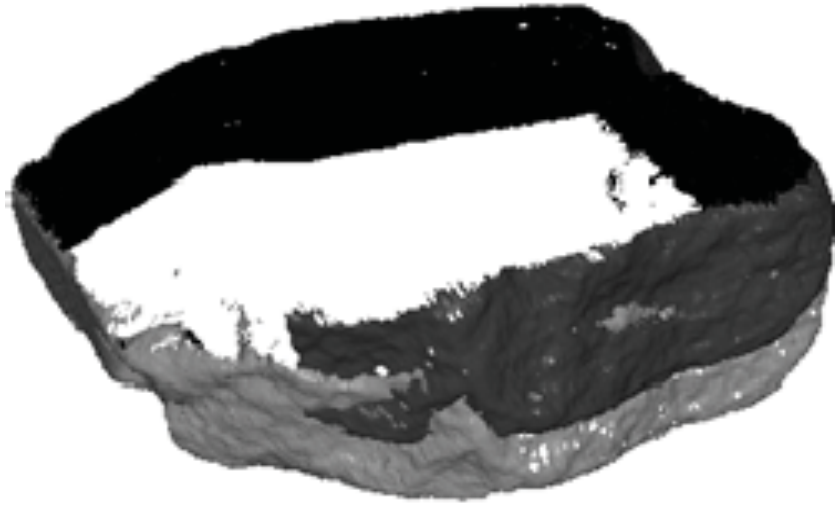
- Want method for distributing accumulated error among all scans



Slides from: http://www.cs.princeton.edu/~bjbrown/iccv05_course/

Global Registration Goal

- Given: *n* scans around an object
- Goal: align them all
- First attempt: ICP each scan to one other



Slides from: http://www.cs.princeton.edu/~bjbrown/iccv05_course/

Approach: The Greedy Solution

- Align each new scan to all previous scans
  [Masuda 96]
- Disadvantages:
  - Order dependent
  - Doesn't spread out error

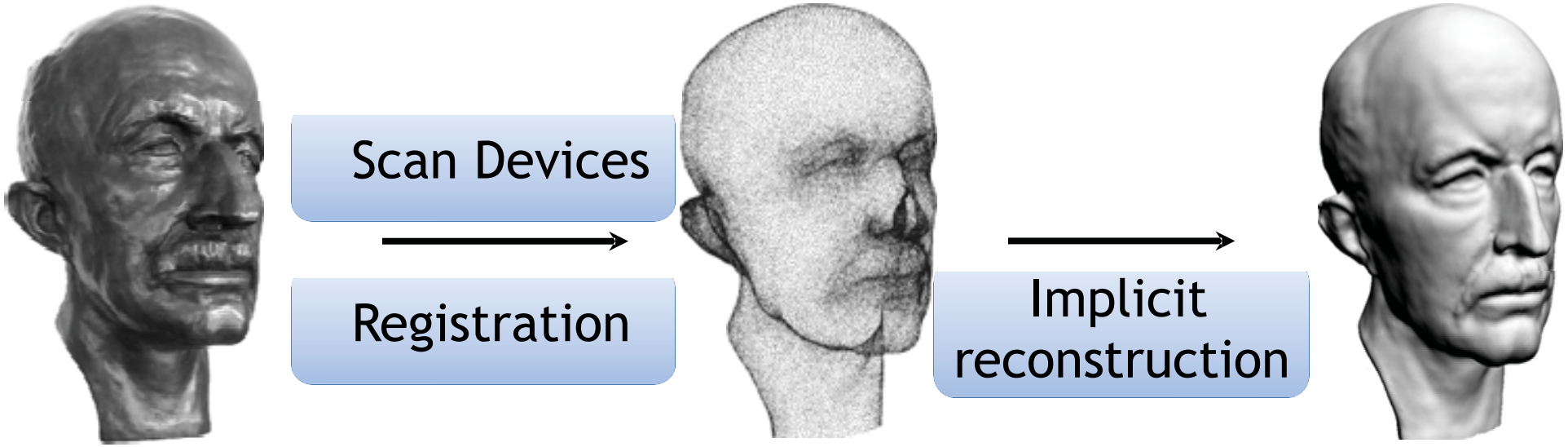
Slides from: http://www.cs.princeton.edu/~bjbrown/iccv05_course/

Approach 2: Slightly Less Brute-Force

- While not converged:
  - For each scan:
    - For each point:
      - For every other scan
        - » Find closest point
    - Minimize error w.r.t. transform of this scan
- Faster than previous method

Slides from: http://www.cs.princeton.edu/~bjbrown/iccv05_course/

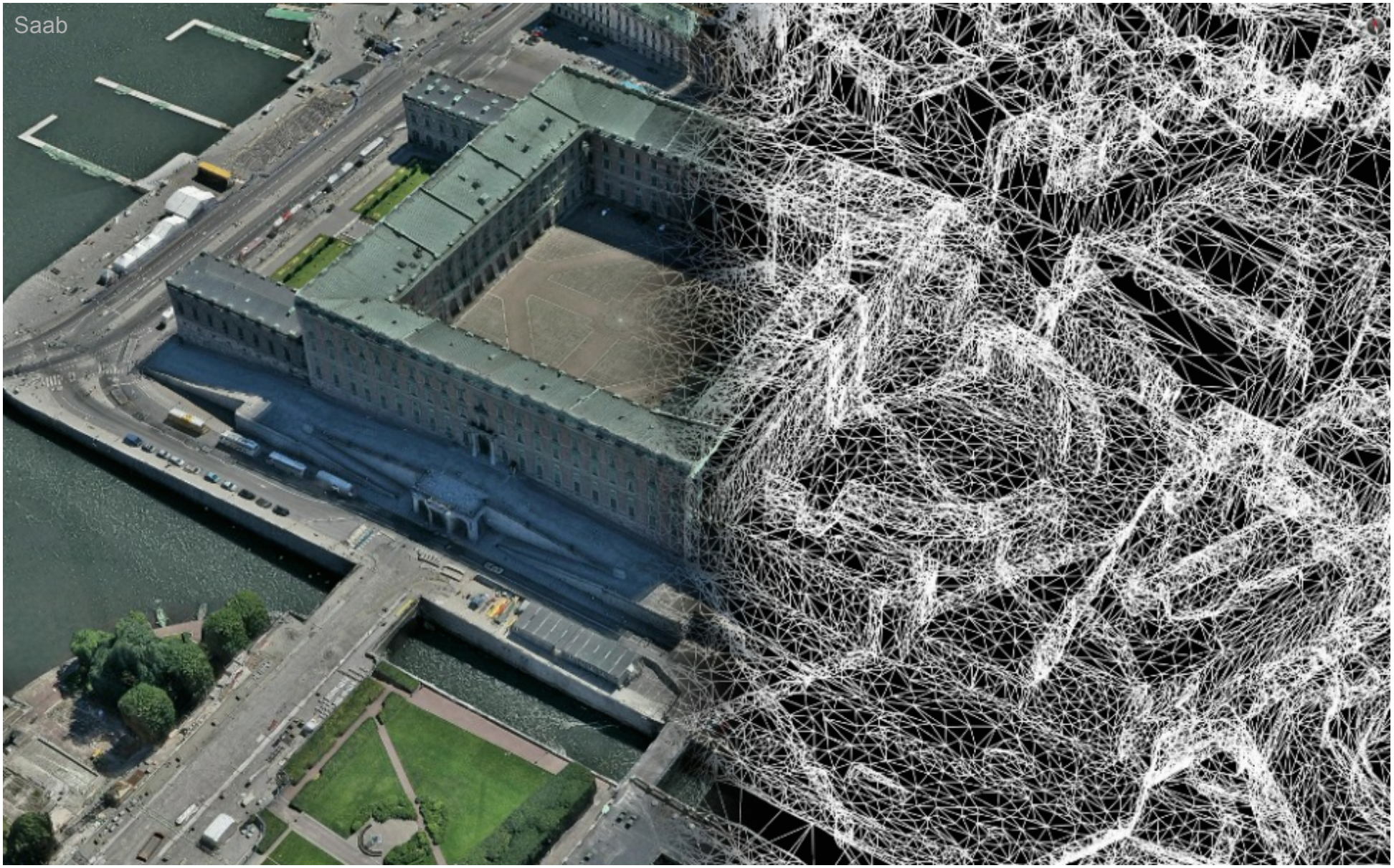
Surface Reconstruction



physical model

acquired point cloud

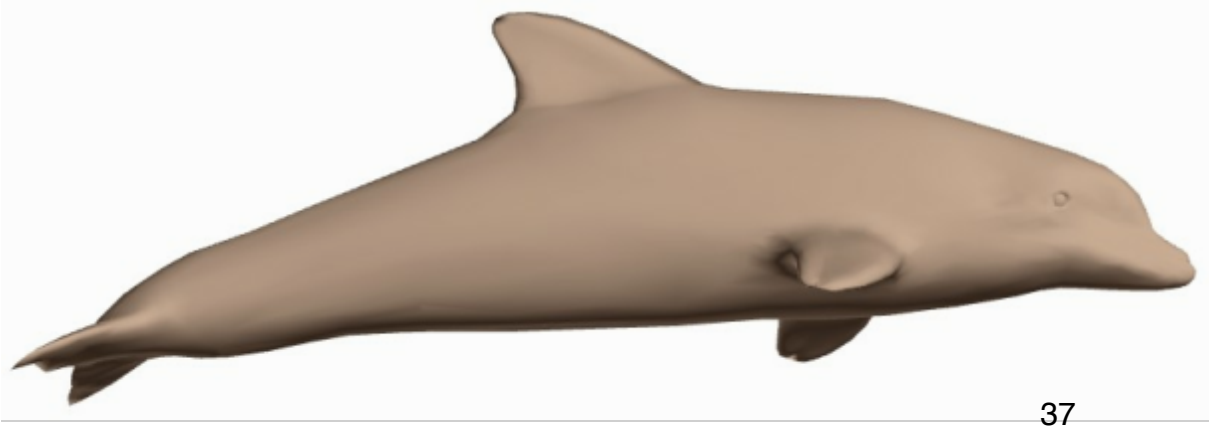
reconstructed model



Polygon Meshes

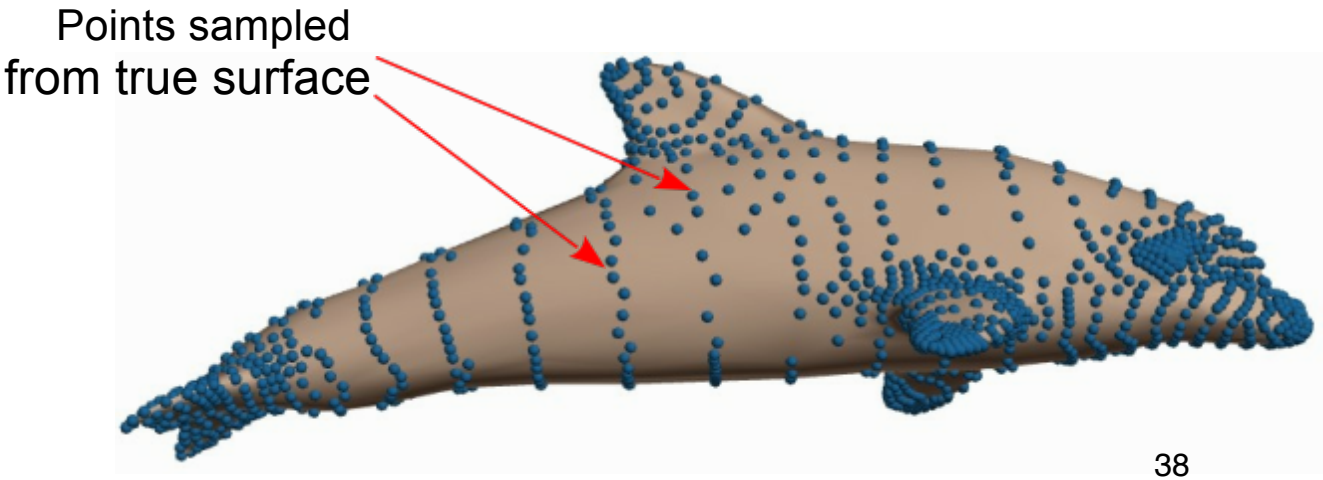
What is a polygon mesh?

- Like a point cloud, it is a discrete sampling of a surface
- ... but, it adds simple polygons (no holes or self-intersections) as linear (fat) approximations of local regions of the actual underlying surface



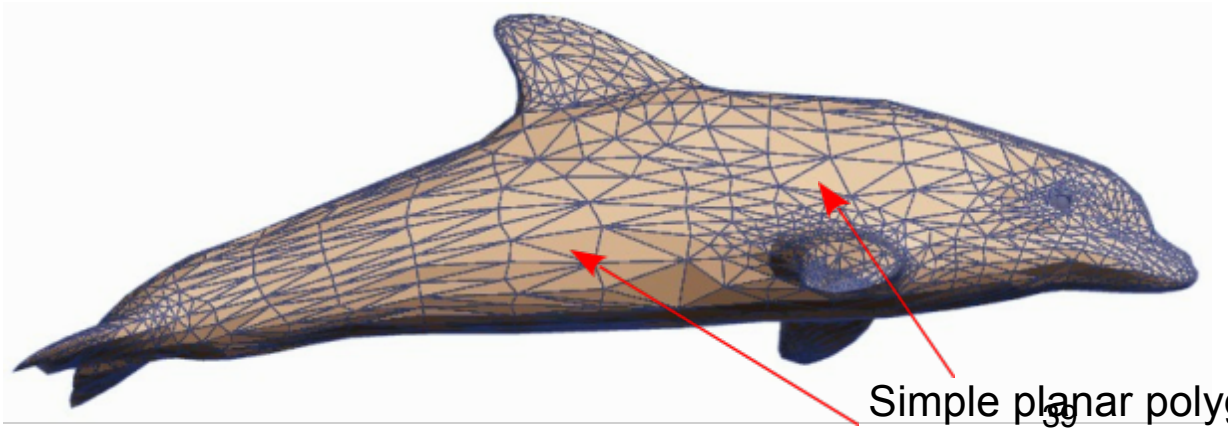
What is a polygon mesh?

- Like a point cloud, it is a discrete sampling of a surface
- ... but, it adds simple polygons (no holes or self-intersections) as linear (fat) approximations of local regions of the actual underlying surface



What is a polygon mesh?

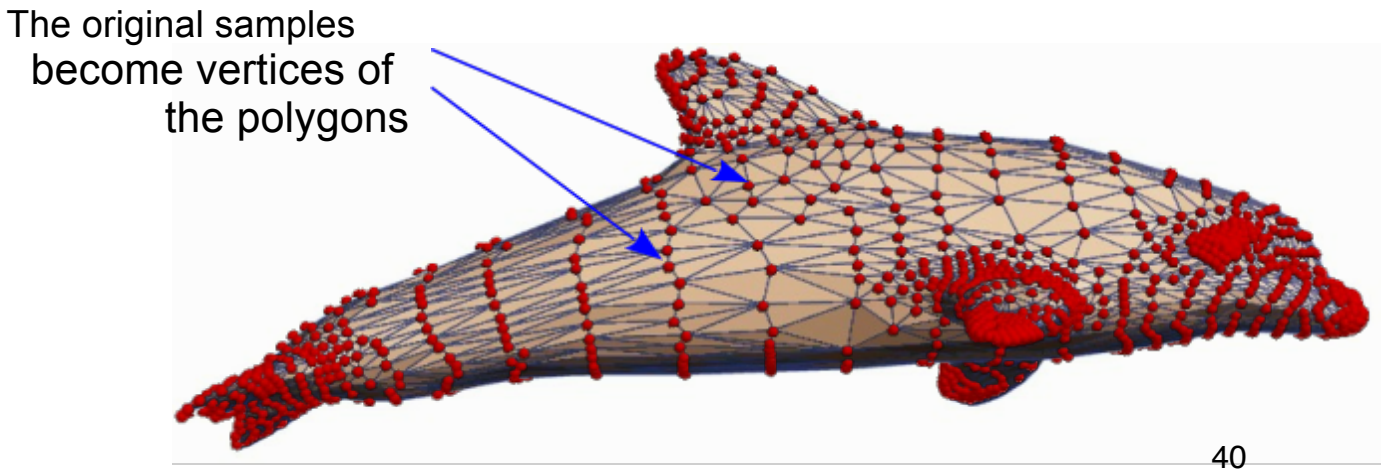
- Like a point cloud, it is a discrete sampling of a surface
- ... but, it adds simple polygons (no holes or self-intersections) as linear (fat) approximations of local regions of the actual underlying surface



Simple planar polygons: linear approximations to true surface

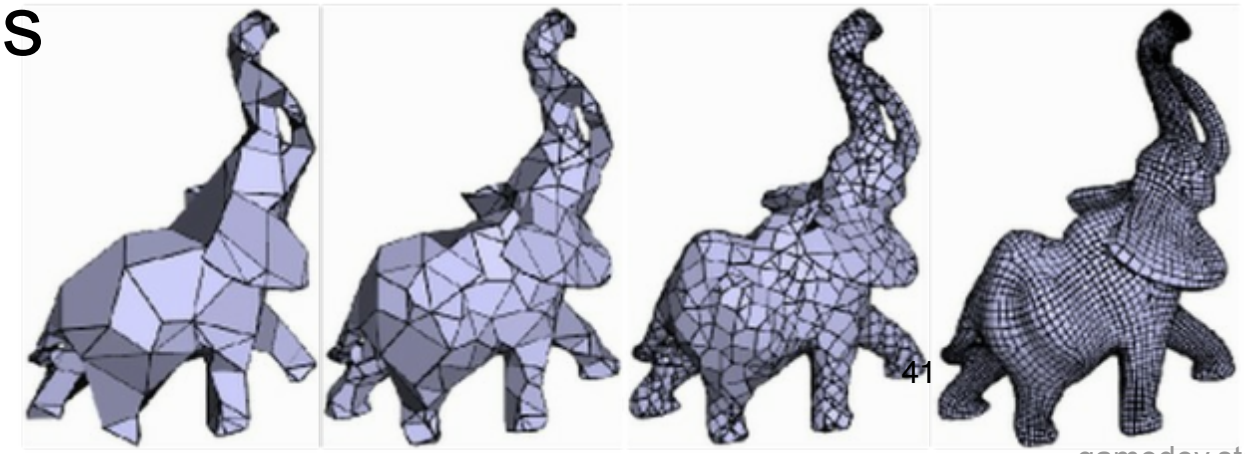
What is a polygon mesh?

- Like a point cloud, it is a discrete sampling of a surface
- ... but, it adds simple polygons (no holes or self-intersections) as linear (fat) approximations of local regions of the actual underlying surface



What is a polygon mesh?

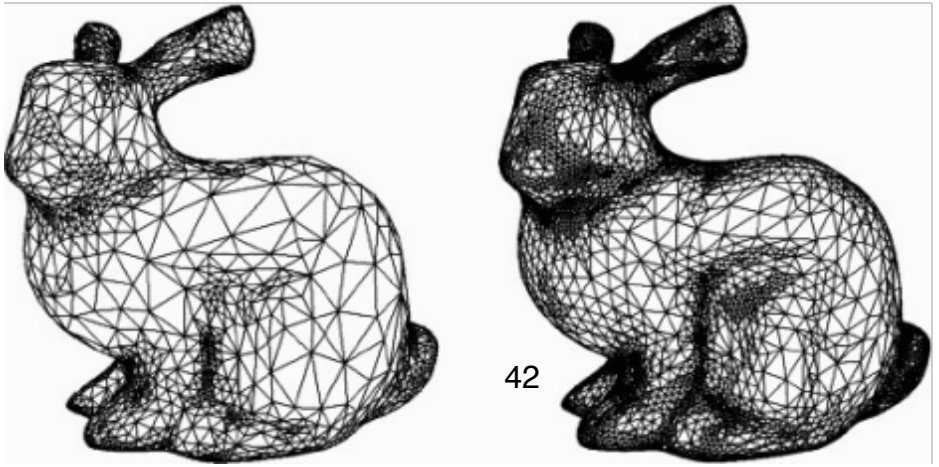
- Like a point cloud, it is based on a discrete sampling of a surface
- ... but, it adds simple polygons (no holes or self-intersections) as linear (fat) approximations of local regions of the actual underlying surface
- Like point clouds, meshes can have diferent resolutions



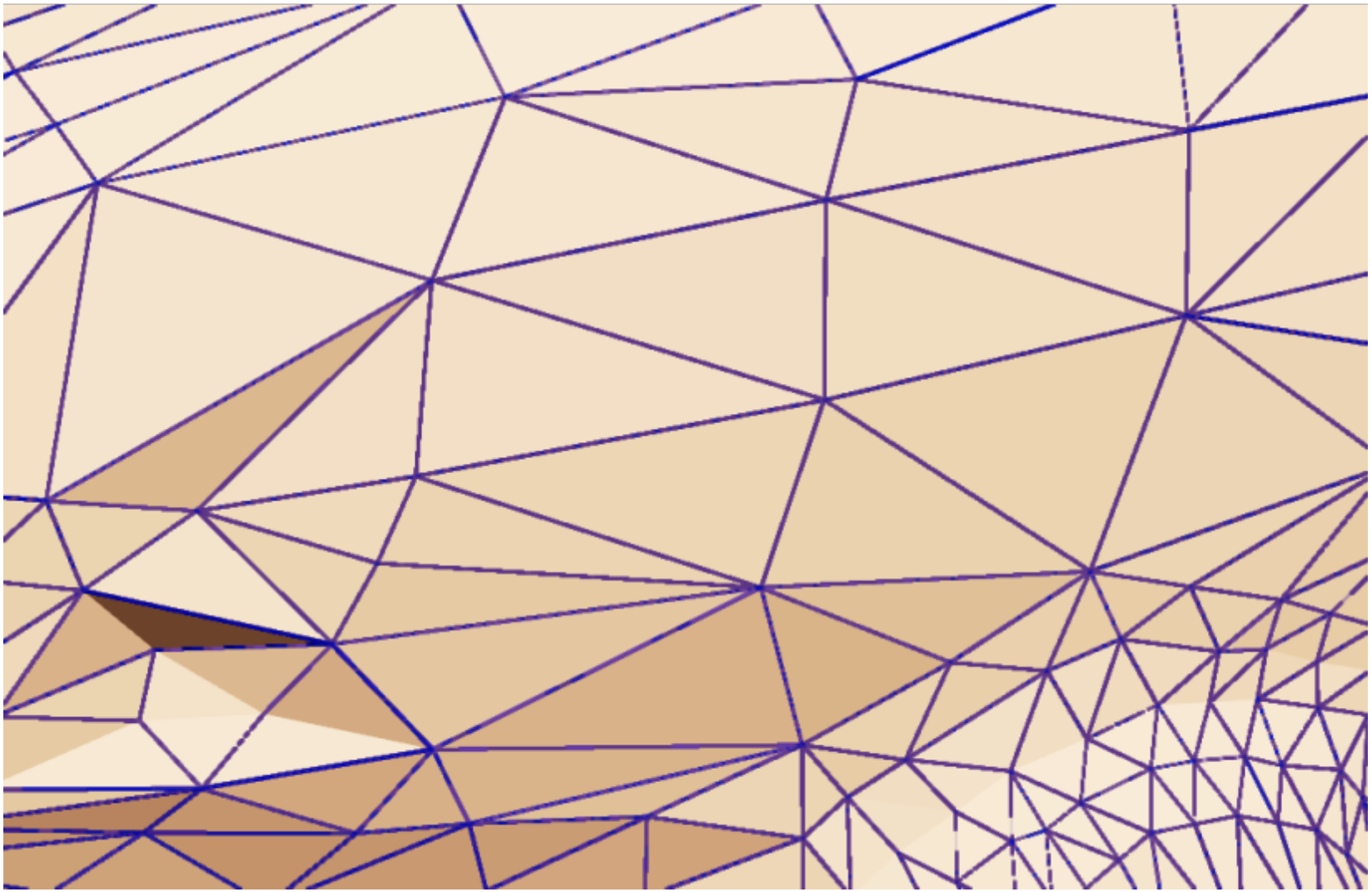
gamedev.stackexchange.com

What is a polygon mesh?

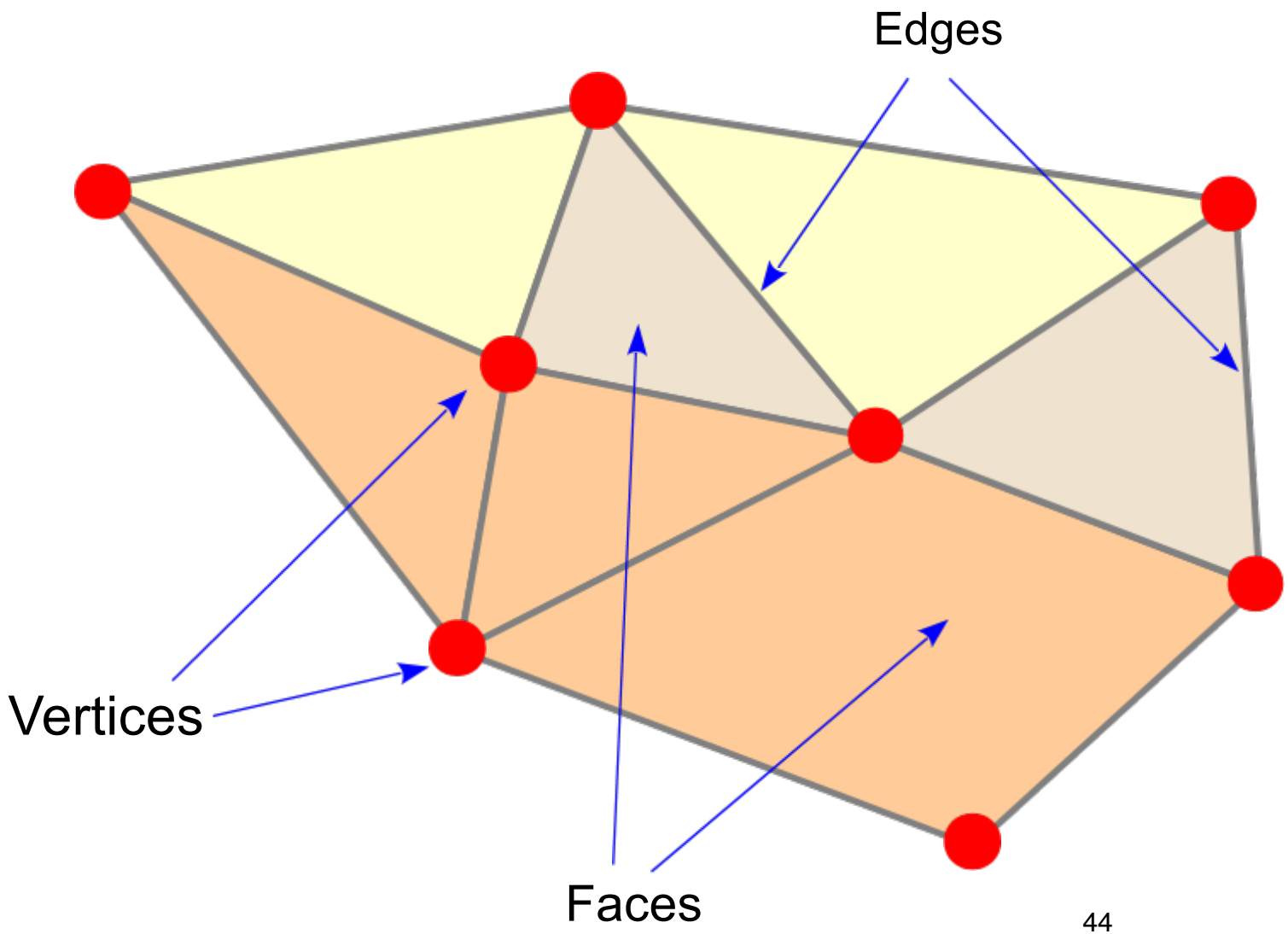
- Like a point cloud, it is based on a discrete sampling of a surface
- ... but, it adds simple polygons (no holes or self-intersections) as linear (fat) approximations of local regions of the actual underlying surface
- Like point clouds, meshes can have diferent resolutions
  - … at diferent places ("adaptive meshing")



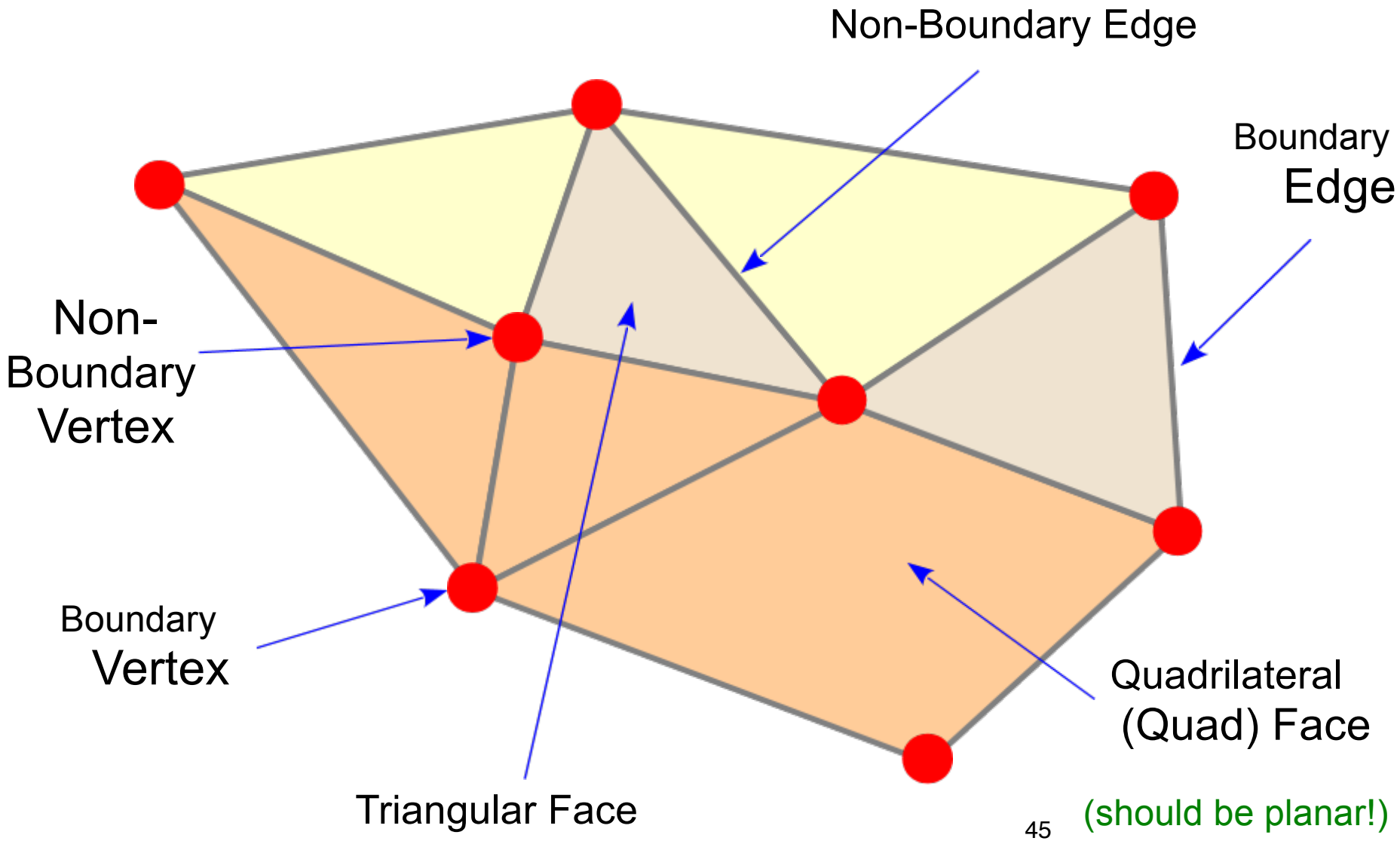
Elements of a mesh



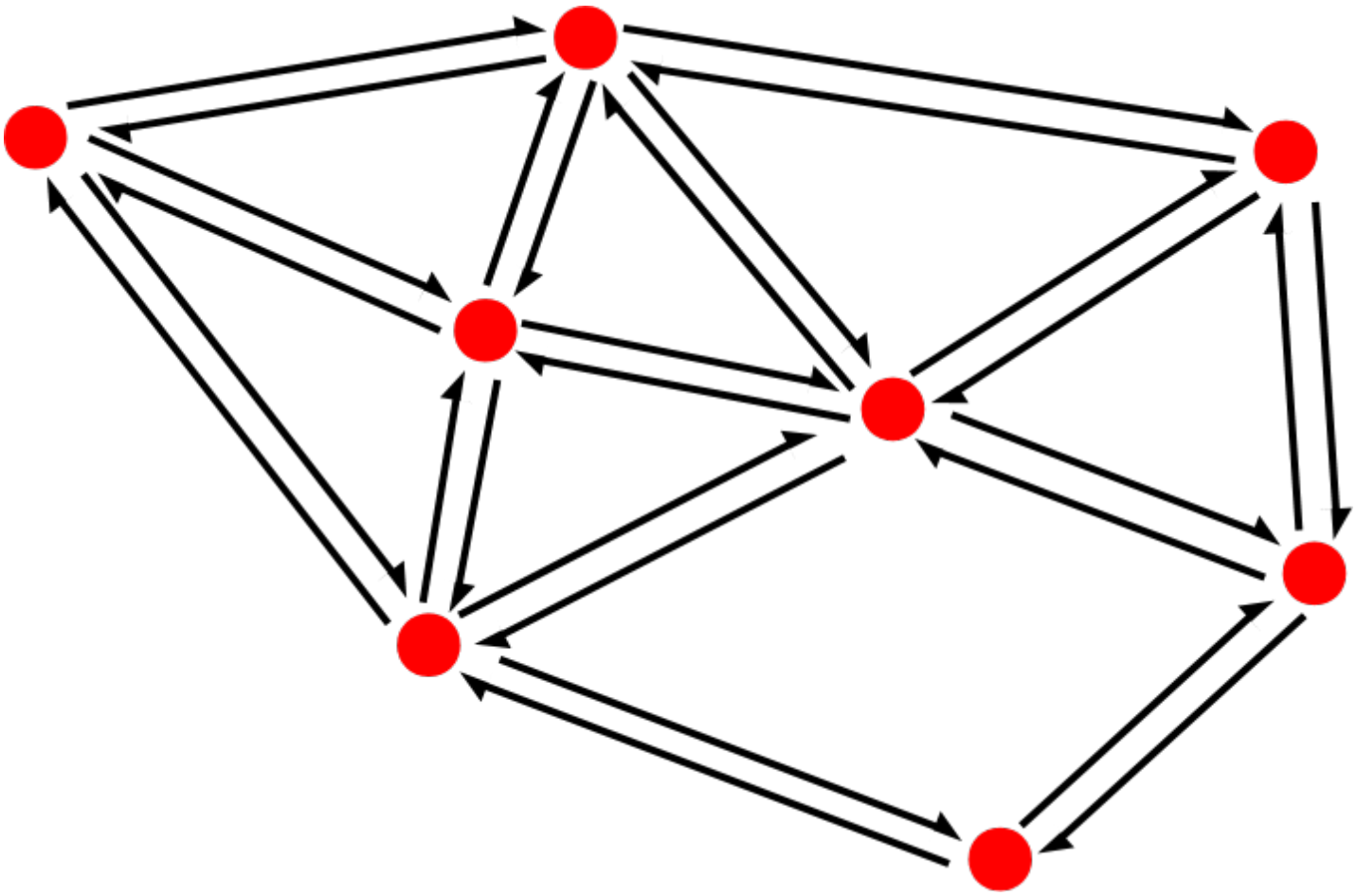
Elements of a mesh



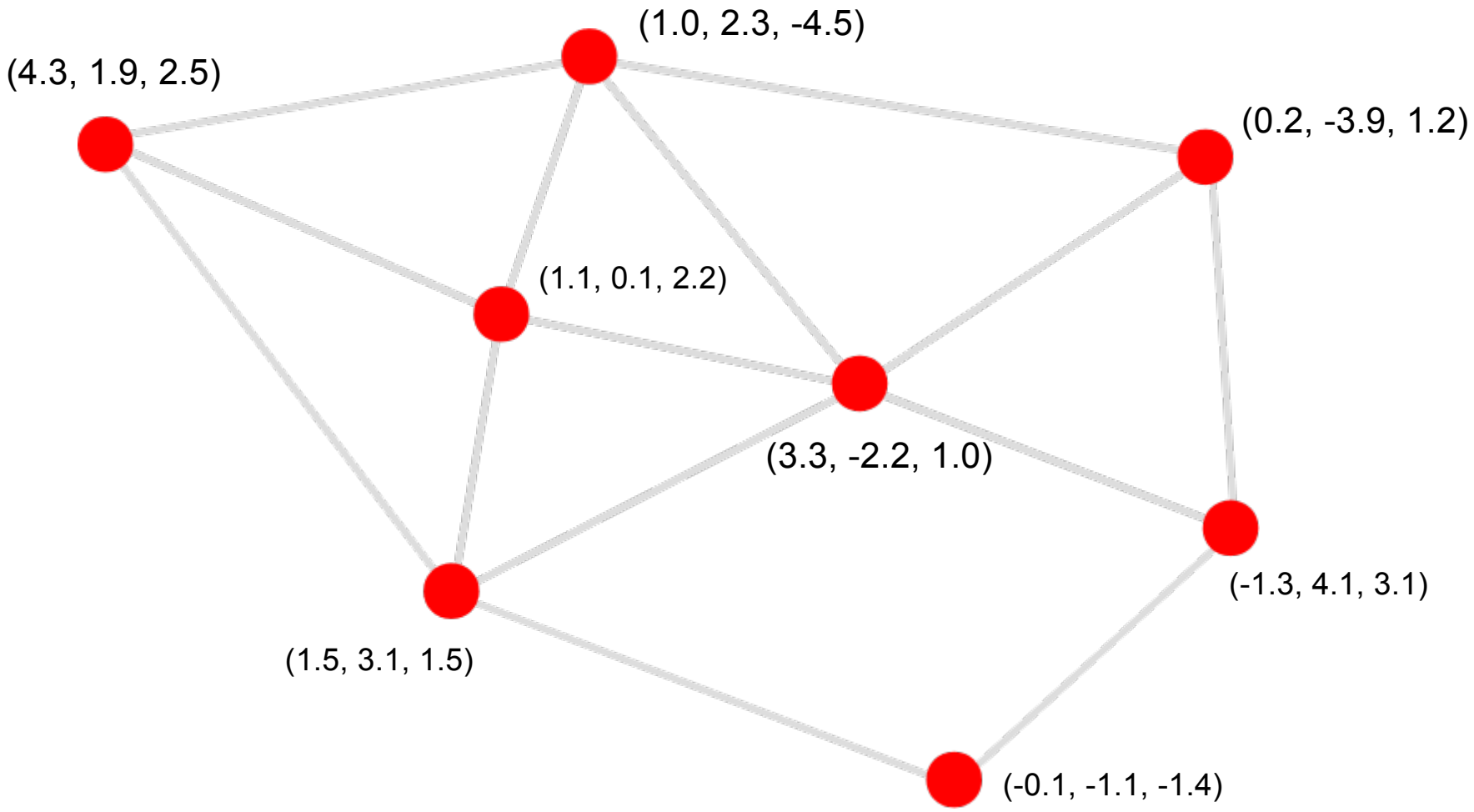
Elements of a mesh



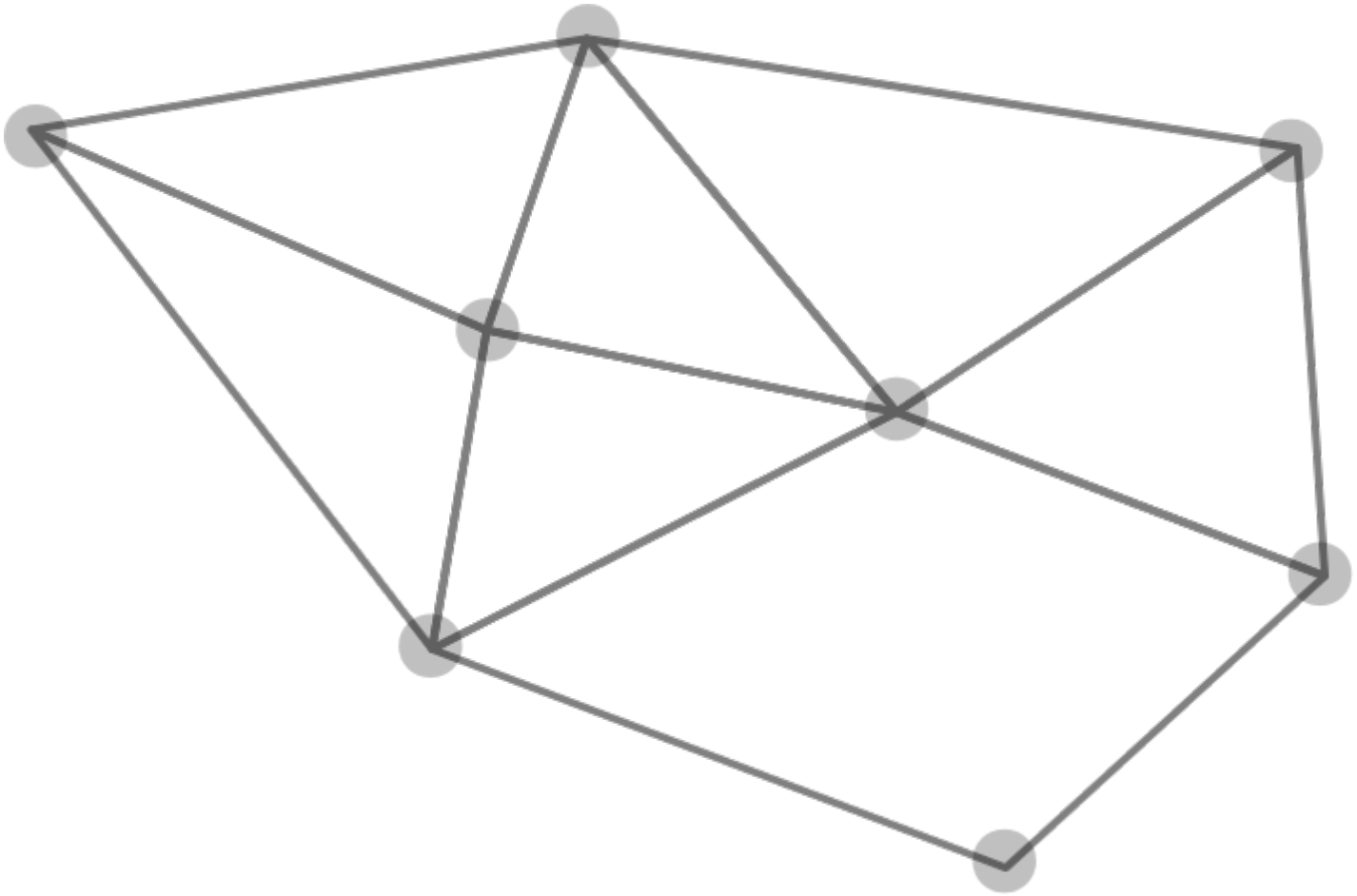
A mesh is an undirected graph



The vertex positions capture the **geometry** of the surface

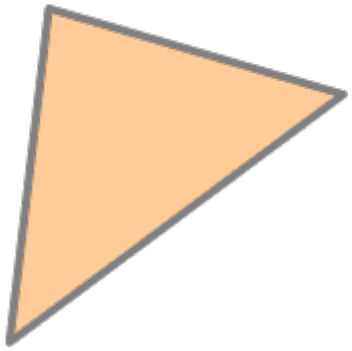


The mesh connectivity captures the **topology** of the surface

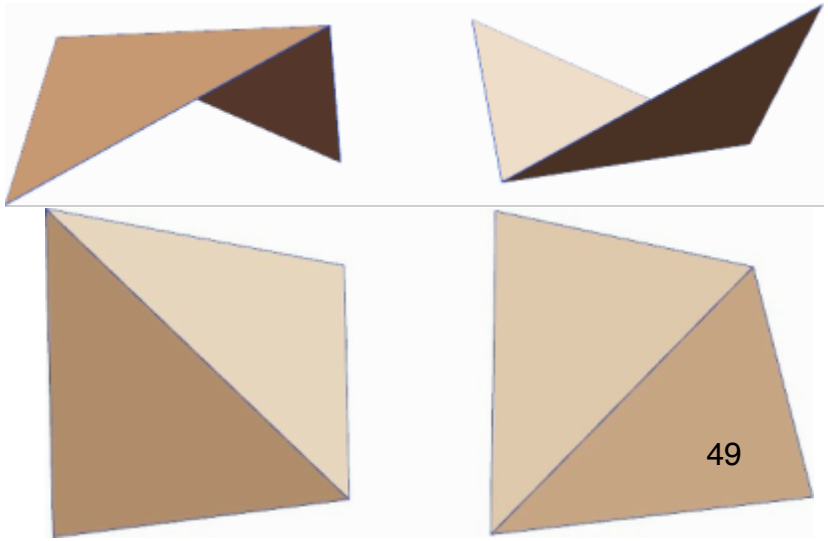


Mesh Geometry: Planes and Normals

- Each polygon is (assumed to be) planar
  - Triangular faces are always planar
  - Quads and higher degree faces need not be
    - Ambiguity revealed by triangulation
  - Many mesh formats allow non-planar faces, but most algorithms assume planar faces. Caveat emptor.



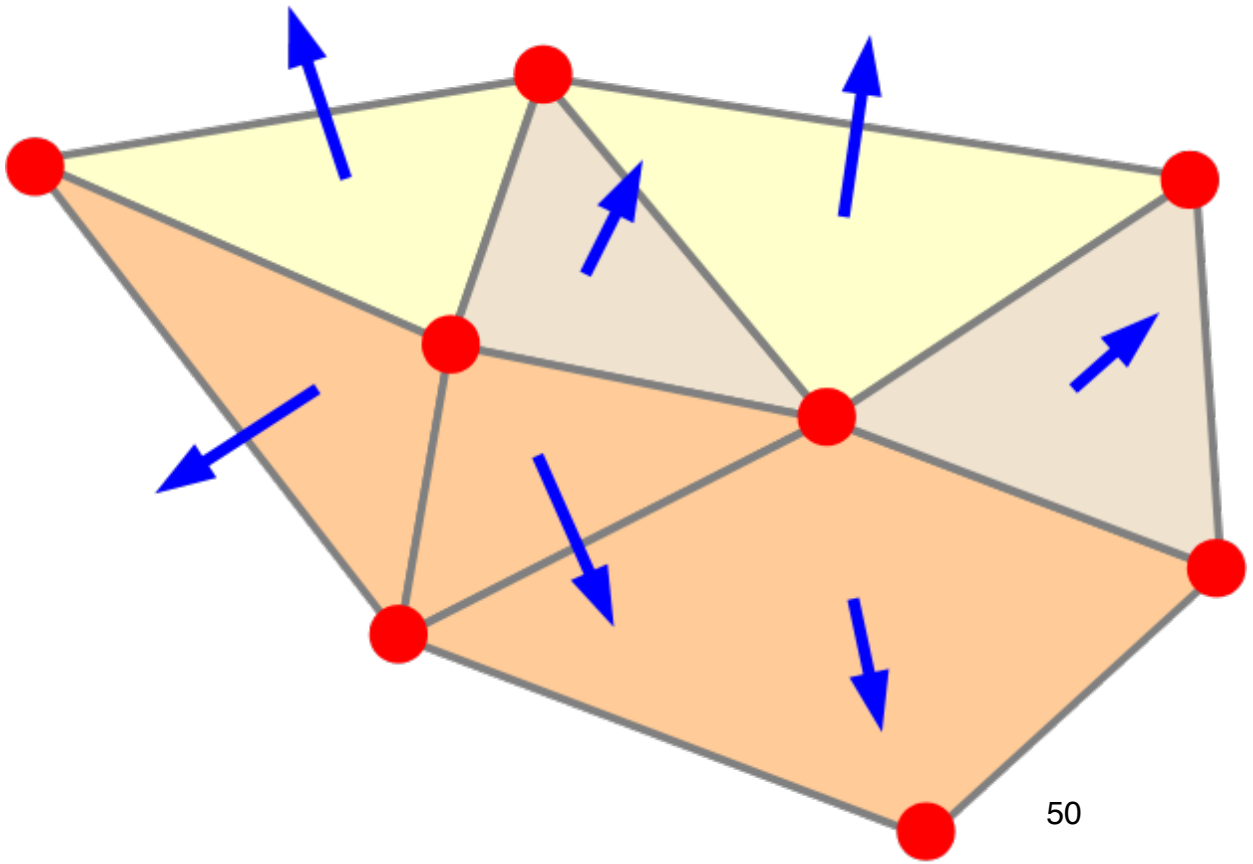
Always planar



Same 4 non-coplanar vertices, diferent geometry!

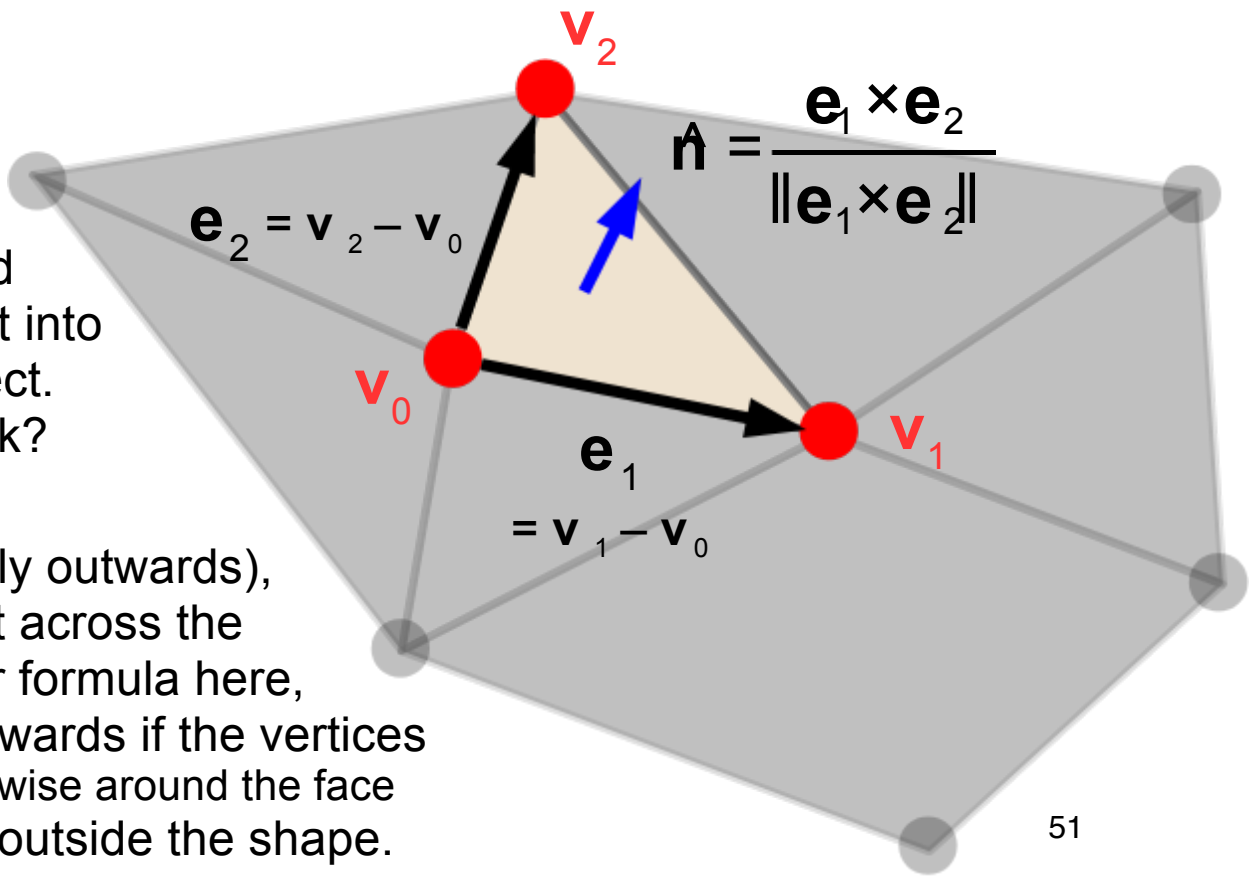
Mesh Geometry: Planes and Normals

- The plane of each polygon has an associated normal vector



Mesh Geometry: Planes and Normals

- The plane of each polygon has an associated normal vector

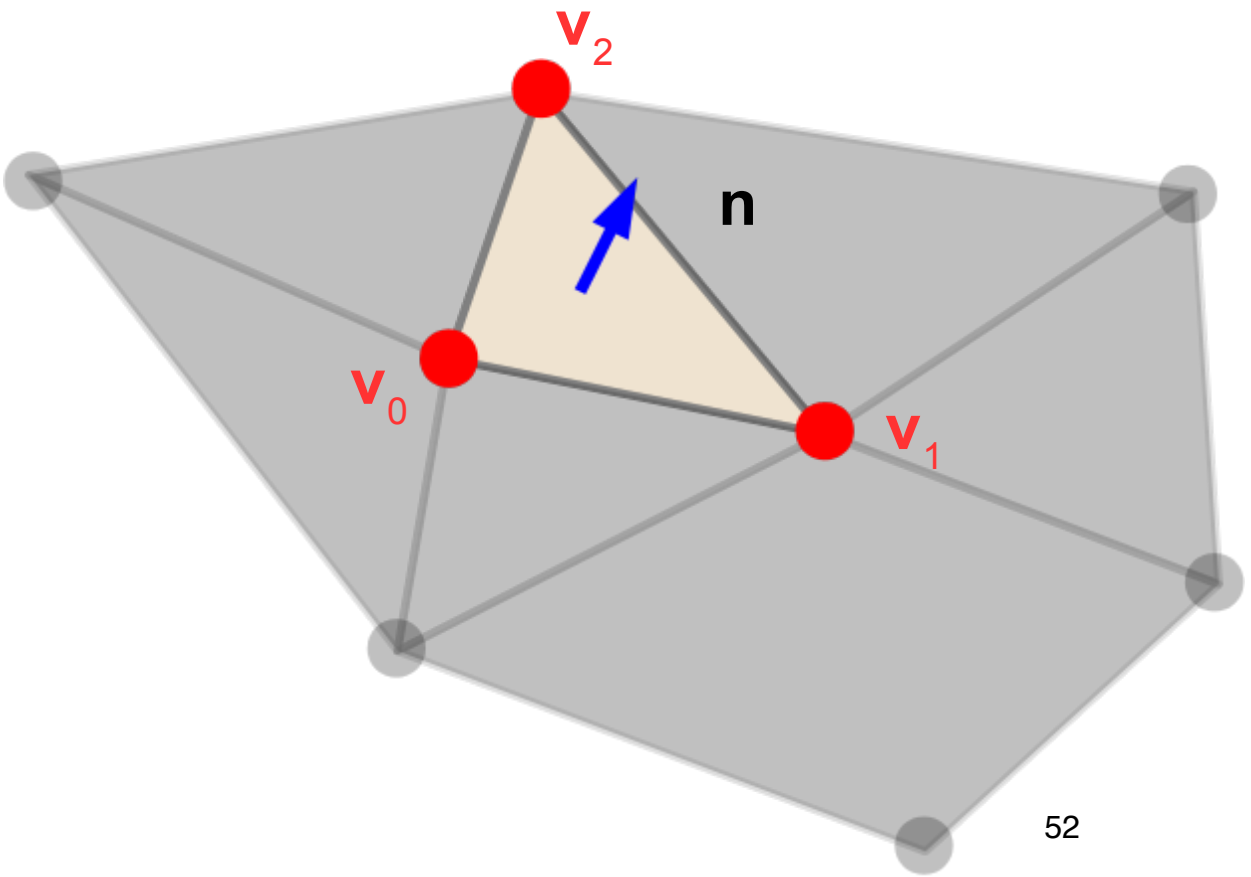


Q: The computed normal may point into or out of the object. Which one to pick?

A: Either (typically outwards), but be consistent across the shape! Using our formula here, the normal is outwards if the vertices wind counter-clockwise around the face when seen from outside the shape.

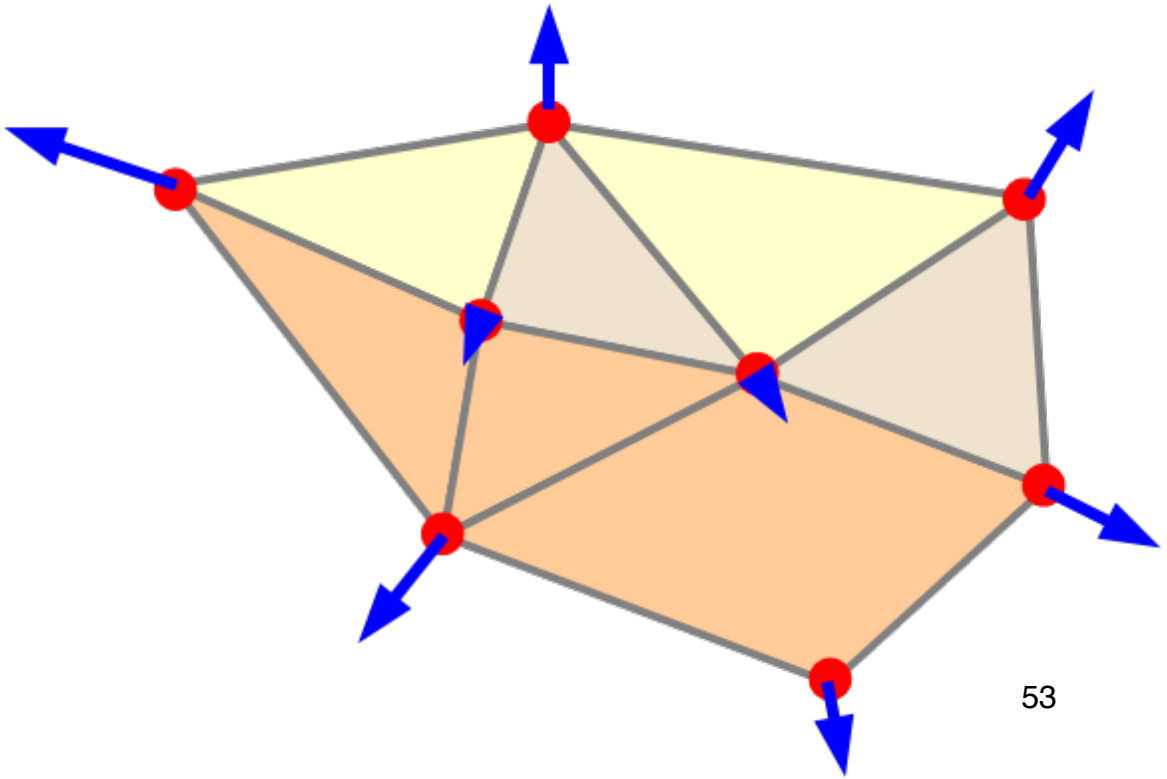
Mesh Geometry: Planes and Normals

- The plane of each polygon has an associated plane equation: $\mathbf{n}\,(\mathbf{p}_0 - \mathbf{v}\,) = 0$



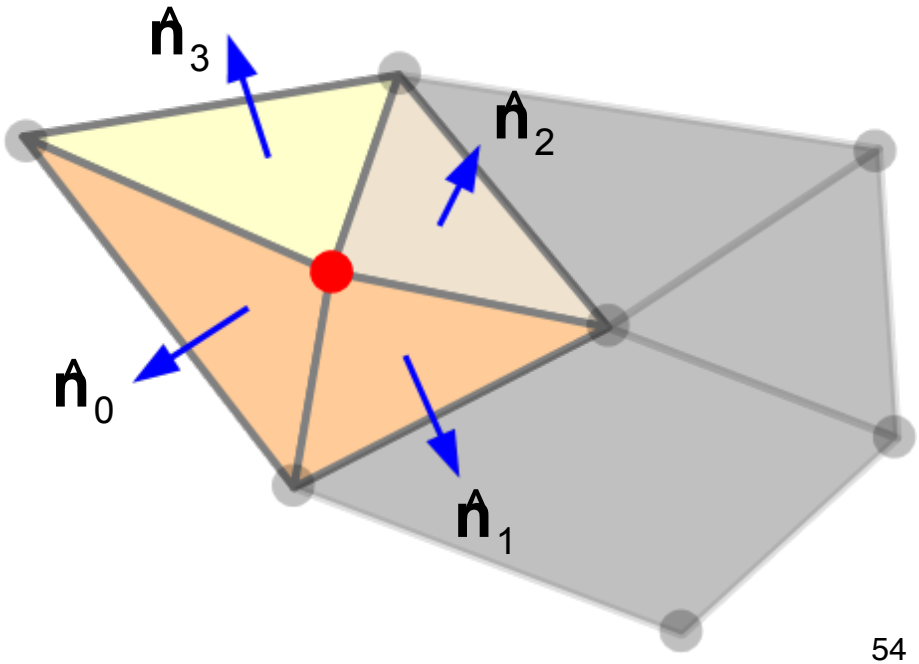
Mesh Geometry: Planes and Normals

- We can also associate vertices with normals
  - Sometimes they come with the mesh (e.g. if they were estimated when the mesh was constructed from a point cloud)
  - Sometimes we have to estimate them



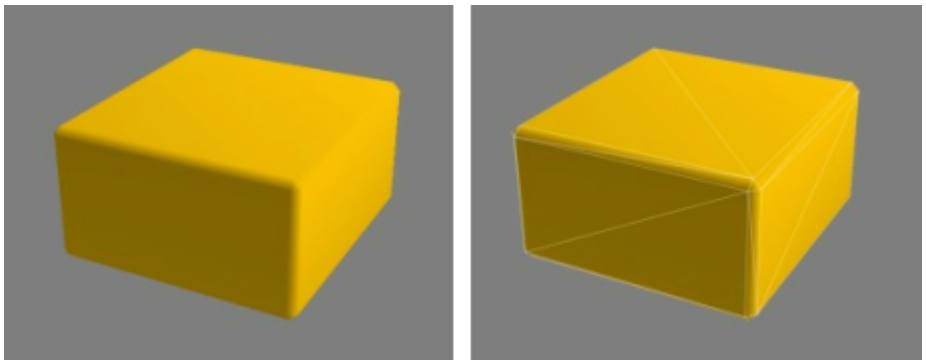
Estimating vertex normals

- **Simplest:** Add up the normals of adjacent faces and unitize



Estimating vertex normals

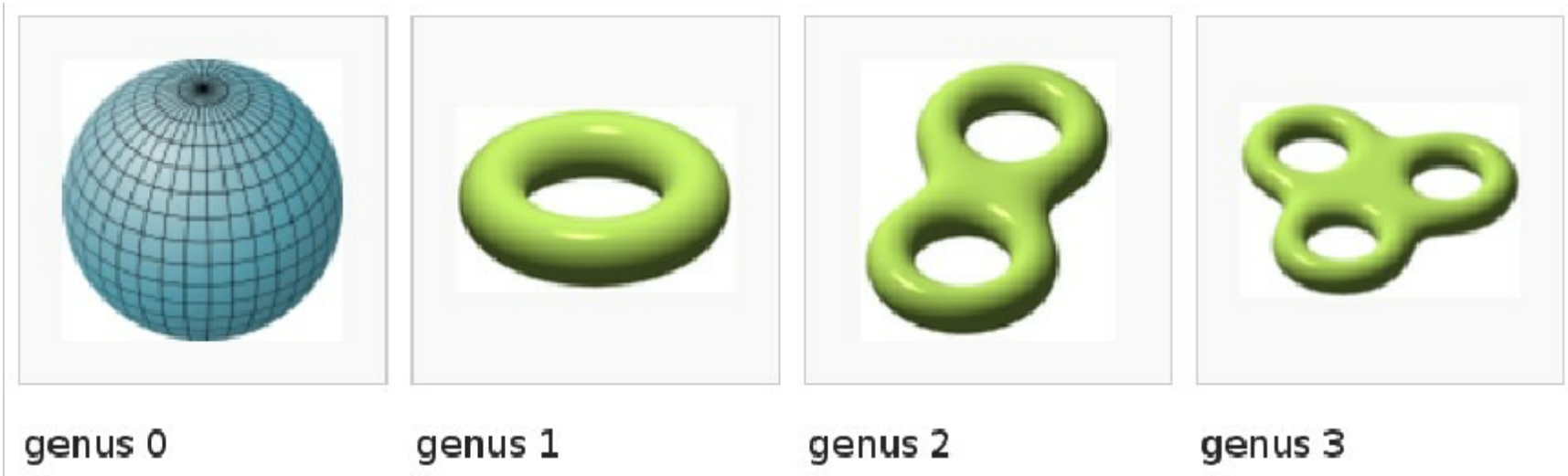
- **Simplest:** Add up the normals of adjacent faces and unitize
- **Simple and usually a bit better:** Add up the normals of adjacent faces, weighted by face areas



bytehazard.com/articles/vertnorm.html

Mesh Topology

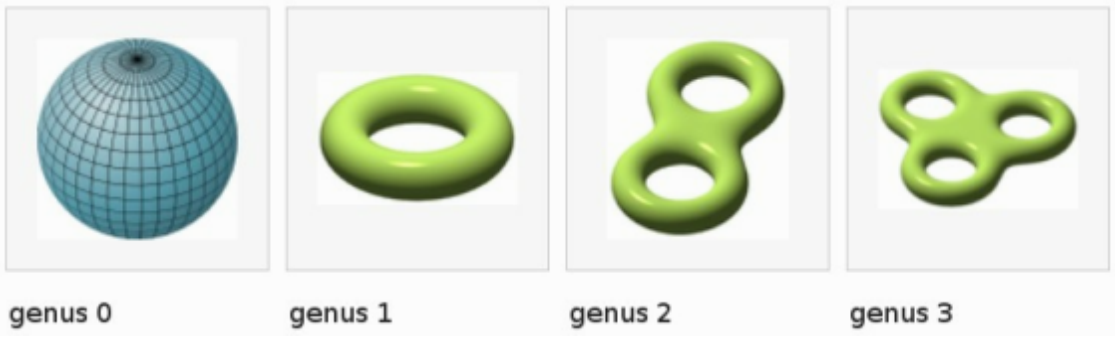
- **Topology** (loosely): The structure of a shape ignoring any measurements of distance, angle etc
  - i.e. the properties invariant to bending, twisting, folding, stretching... (but not tearing)
- E.g. Genus: The number of handles in a shape



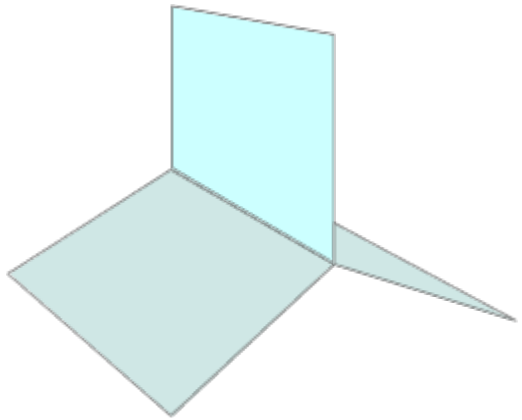
 Wikipedia

Mesh Topology

- **Manifold:** A topological space that is locally Euclidean (neighborhood has the topology of the unit ball)



Some manifold shapes



Not manifold

Manifold structure of a surface is approximated by its mesh connectivity

Thank You