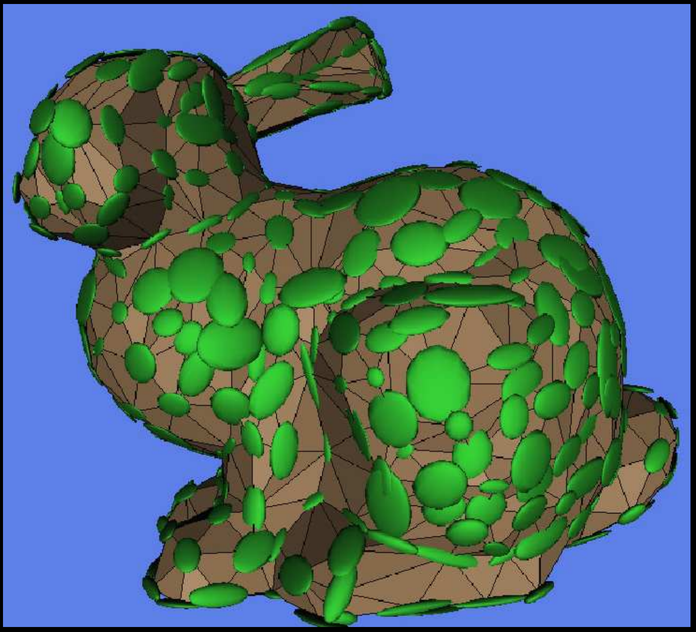
Mesh Simplification

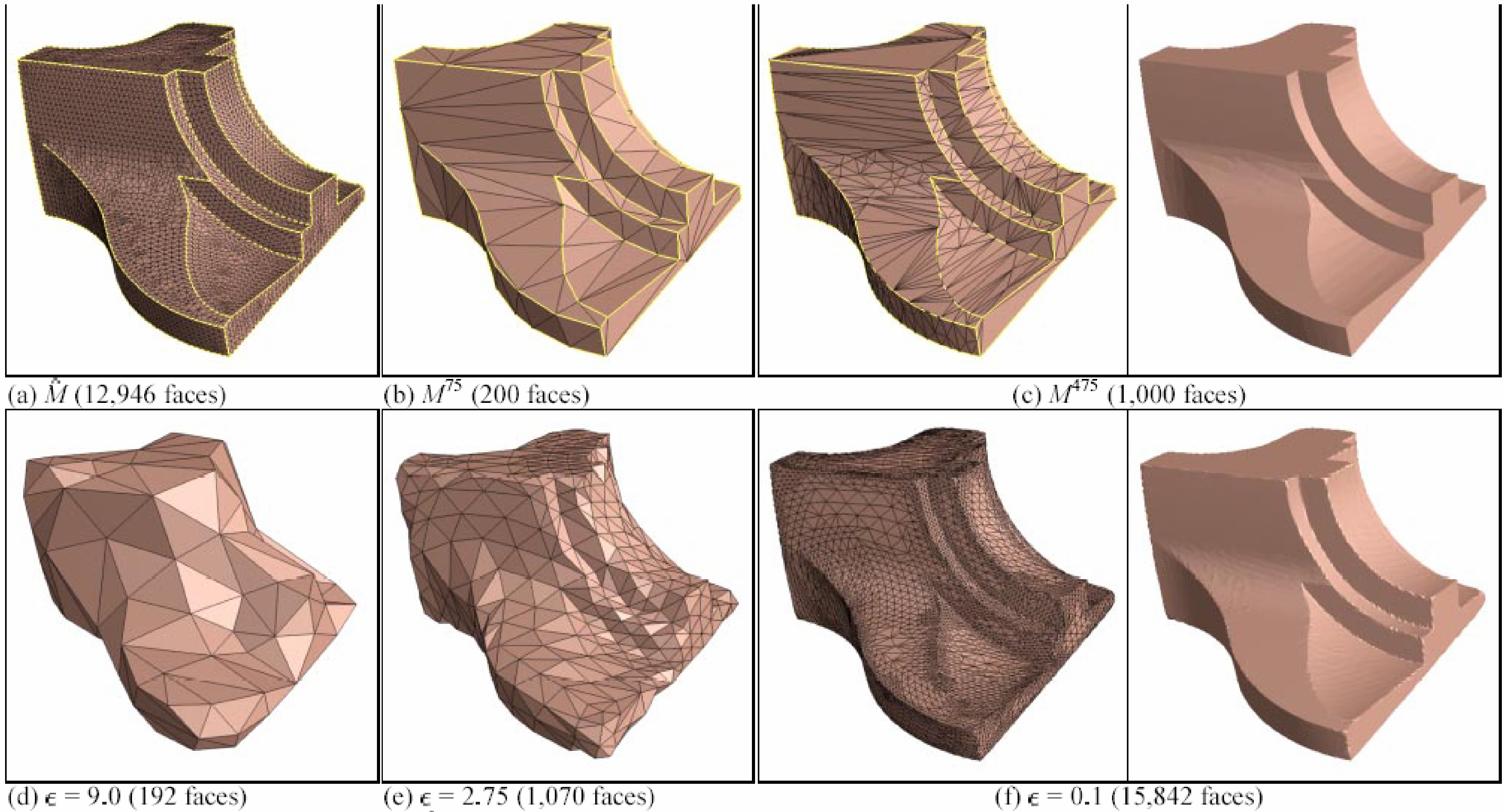


Doug James

February 3, 2004

15-864 Advanced Computer Graphics

Motivation: Multiresolution Analysis (MRA)



(a) $\hat{M}$ (12,946 faces) (b) M^{75} (200 faces) (c) M^{475} (1,000 faces)

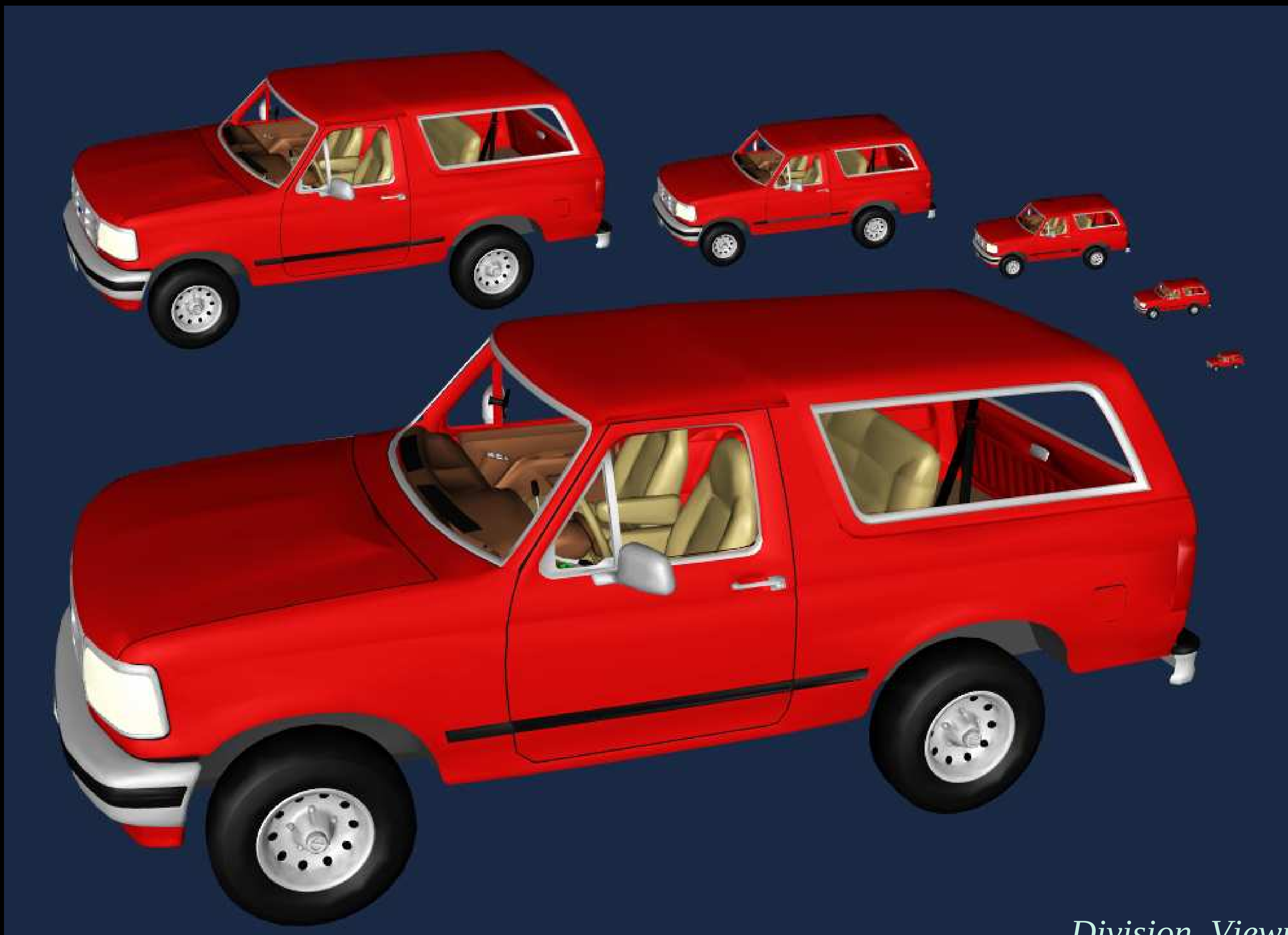
(d) $\epsilon = 9.0$ (192 faces) (e) $\epsilon = 2.75$ (1,070 faces) (f) $\epsilon = 0.1$ (15,842 faces)

[Hoppe, 1995]

Acknowledgement

- Tom Funkhouser
  - many slides borrowed from Princeton C0S 526, Fall 2002

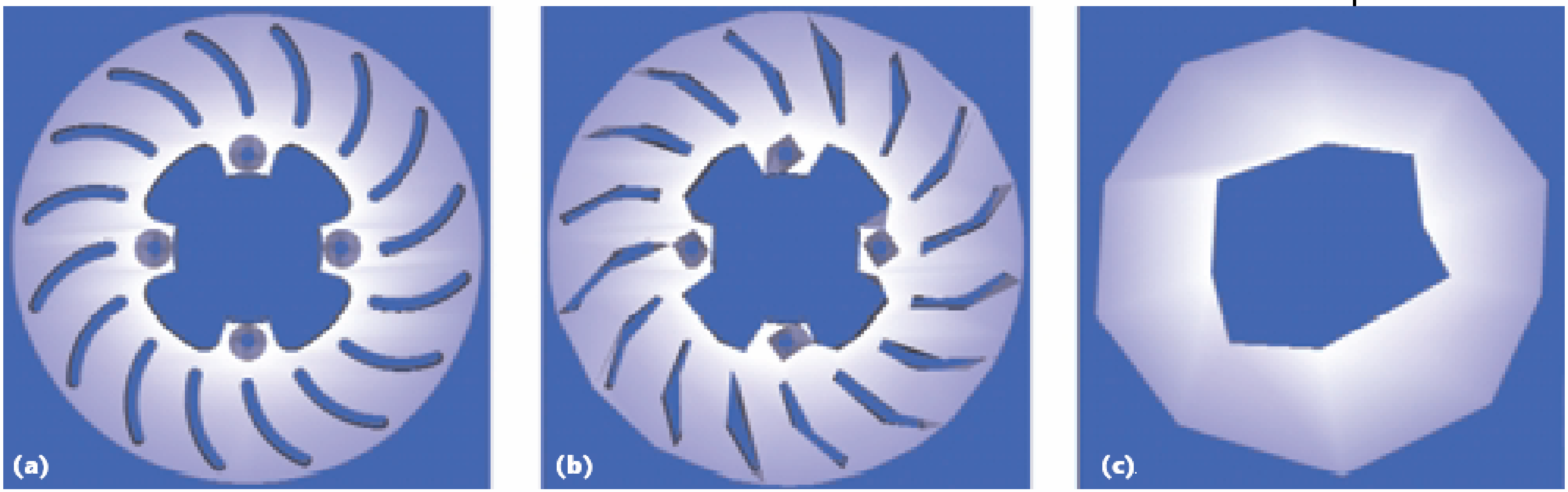
Mesh Simplification



Triangles:
41,855
27,970
20,922
12,939
8,385
4,766

Division, Viewpoint, Cohen

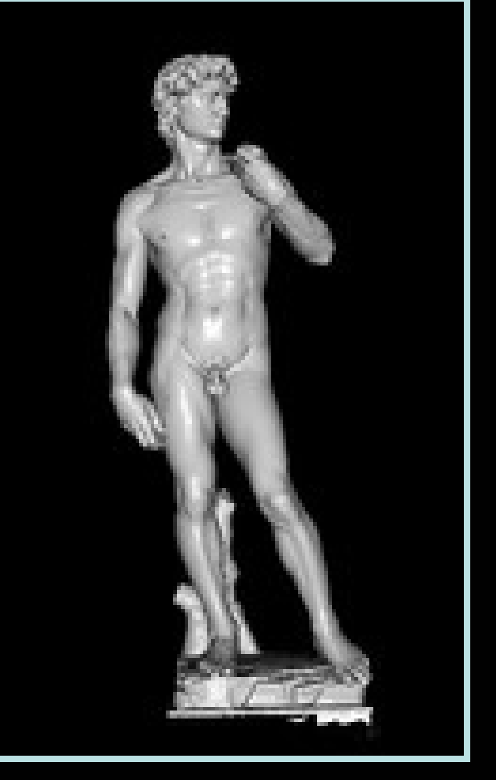
Mesh Simplification



4 Preserving genus limits drastic simplification. The original model of a brake rotor with (a) 4,736 triangles and 21 holes is simplified with a topology-preserving algorithm using (b) 1,006 triangles and 21 holes and a topology-modifying algorithm with (c) 46 triangles and one hole. Model courtesy of the Alpha_1 Project, University of Utah.

Mesh Simplification Goals

- Reduce number of polygons
  - Faster rendering
  - Less storage
  - Simpler manipulation
- Desirable properties
  - Generality, efficiency, scalability
  - Produces “good” approximation



Stanford Graphics Lab

Simplification Algorithms

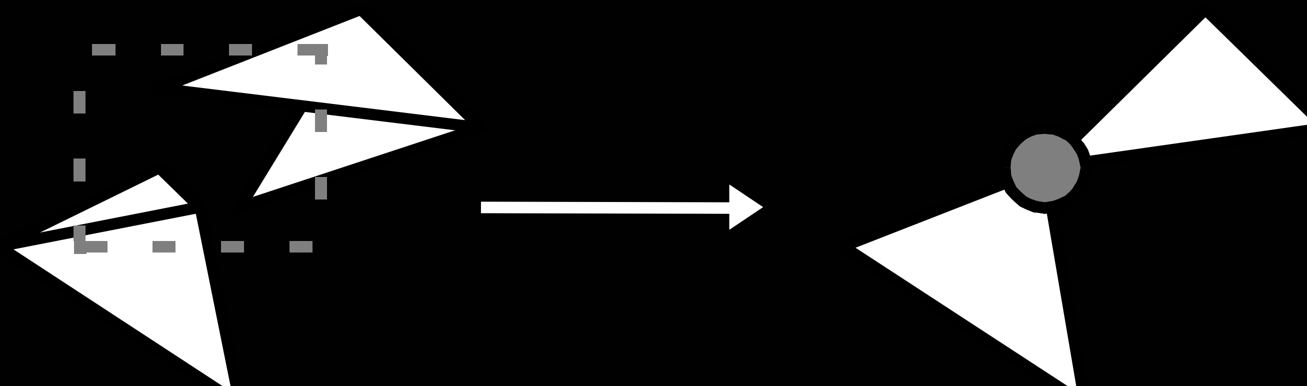
- Measure cost of possible decimation operations according to error measure
- Place operations in queue according to error
- Perform operations in queue successively
  - After each operation, re-evaluate error metrics

Mesh Simplification Operations

- General idea:
  - Each operations simplifies model by small amount
  - Apply many operations in succession
- Types of operations
  - Vertex cluster
  - Vertex remove
  - Edge collapse
  - Vertex pair

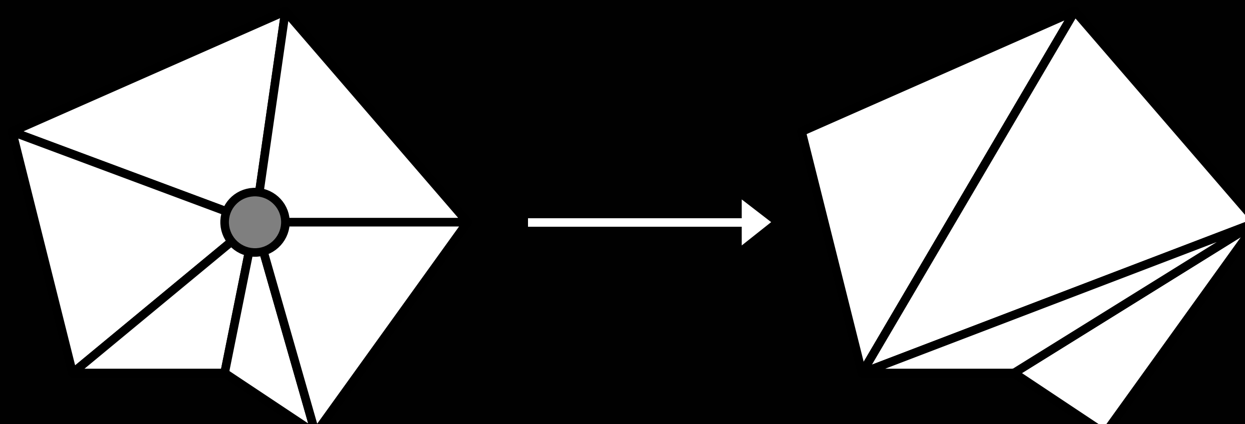
Vertex Cluster

- Method
  - Merge vertices based on proximity
  - Triangles with repeated vertices become edge or point
- Properties
  - General and robust
  - Not usually attractive



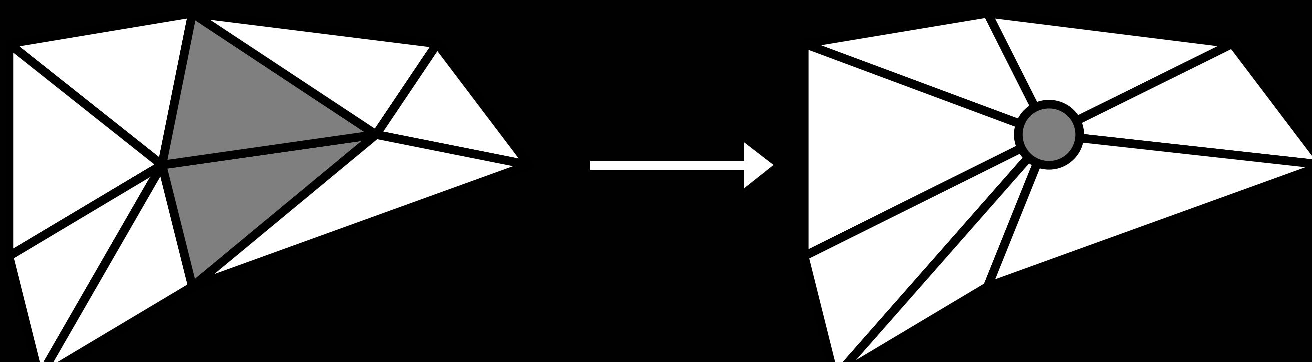
Vertex Remove

- Method
  - Remove vertex and adjacent faces
  - Fill hole with new triangles (reduction of 2)
- Properties
  - Requires manifold surface around vertex
  - Preserves local topological structure
  - Typically more attractive



Edge Collapse

- Method
  - Merge two edge vertices to one
  - Delete degenerate triangles
- Properties
  - Requires manifold surface around vertex
  - Preserves local topological structure
  - Typically more attractive
  - Allows smooth transition (*Geomorph* [Hoppe95])



Operation Considerations

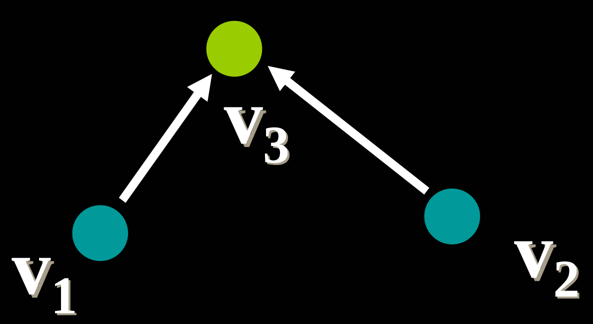
- Topology considerations
  - Attention to topology promotes better appearance
  - Allowing non-manifolds increases robustness and ability to simplify
- Operation considerations
  - Collapse-type operations allow smooth transitions
  - Vertex remove affects smaller portion of mesh than edge collapse

Geometric Error Metrics

- Motivation
  - Promote accurate 3D shape preservation
  - Preserve screen-space silhouettes and pixel coverages
- Types
  - Vertex-Vertex Distance
  - Vertex-Plane Distance
  - Point-Surface Distance
  - Surface-Surface Distance

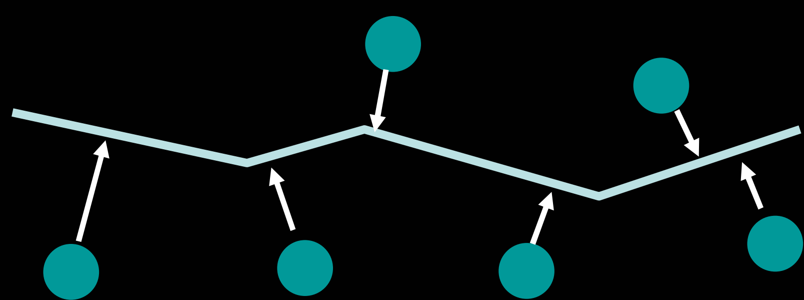
Vertex-Vertex Distance

- E = max(|| v3-v1||, || v3-v2 ||)
- Appropriate during topology changes
  - Rossignac and Borrel 93
  - Luebke and Erikson 97
- Loose for topology-preserving collapses



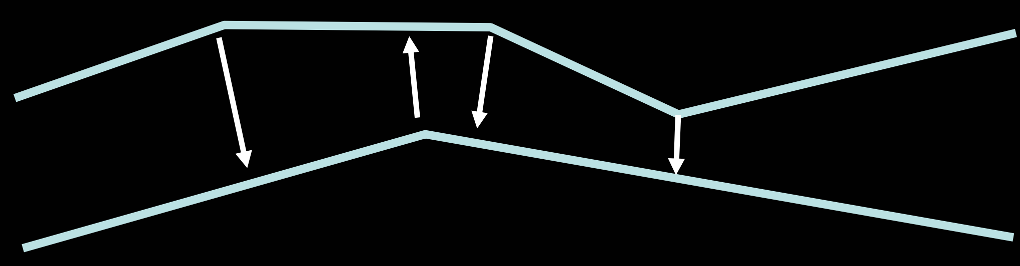
Point-Surface Distance

- Map point set to closest points on simplified surface
- Compute sum of square distances



Surface-Surface Distance

- Bound maximum distance between input and simplified surfaces
  - Tolerance Volumes - Guéziec 96
  - Simplification Envelopes - Cohen/Varshney 96
  - Hausdorf Distance - Klein 96
  - Mapping Distance - Bajaj/Schikore 96, Cohen et al. 97

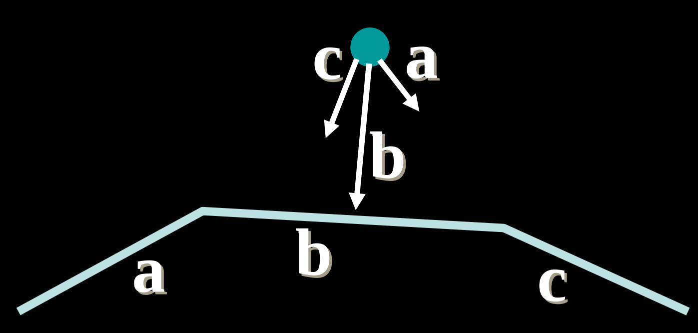


Vertex-Vertex != Surface-Surface

- Error is zero at vertices and exterior edges
- Error is non-zero everywhere else
  - not captured by vertex-vertex or vertex-plane metrics

Vertex-Plane Distance

- Store set of planes with each vertex
  - Error based on distance from vertex to planes
  - When vertices are merged, merge sets
- Ronfard and Rossignac 96
  - Store plane sets, compute max distance
- Error Quadrics - Garland and Heckbert 96
  - Store quadratic form, compute sum of square distances

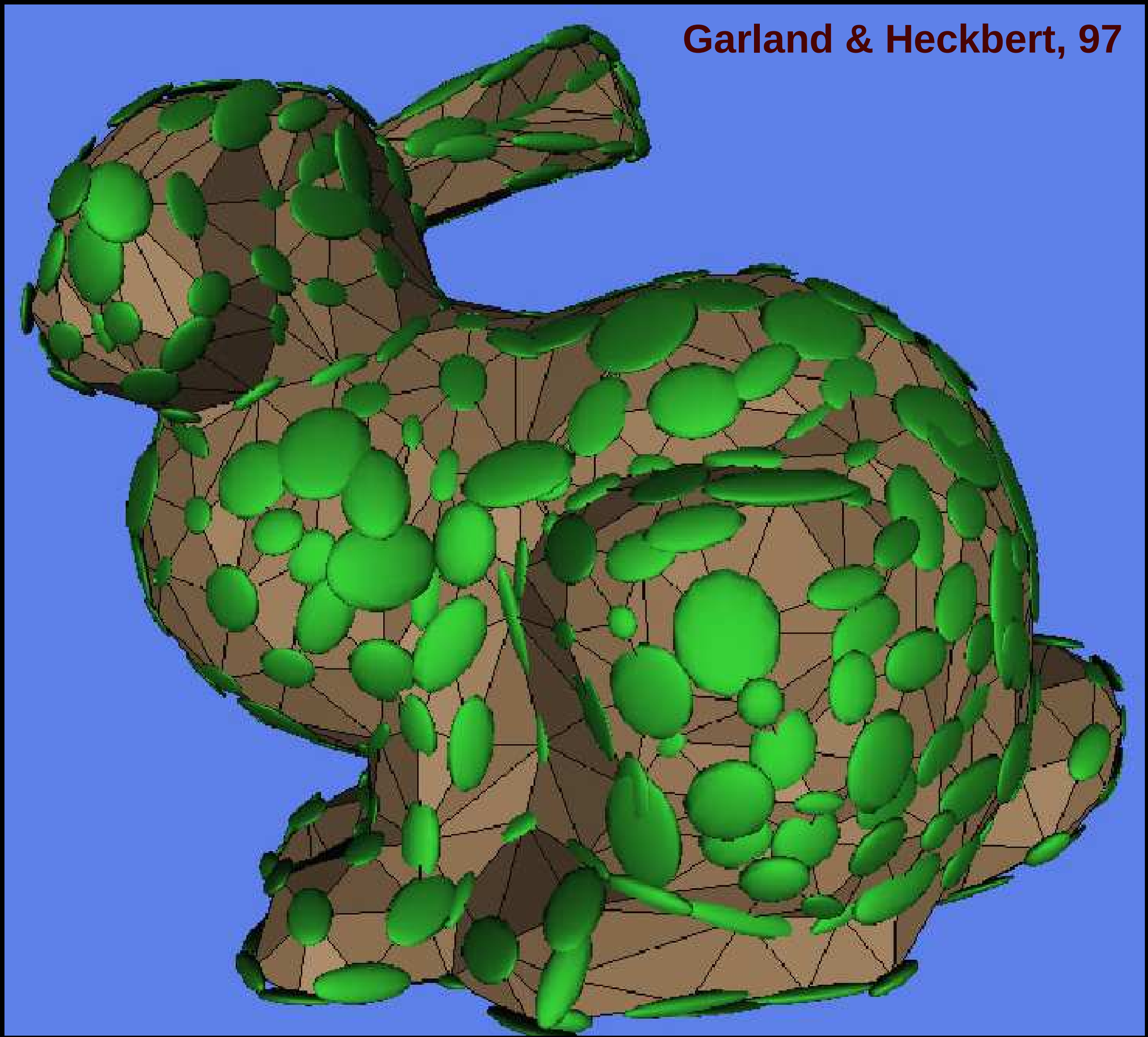


Geometric Error Observations

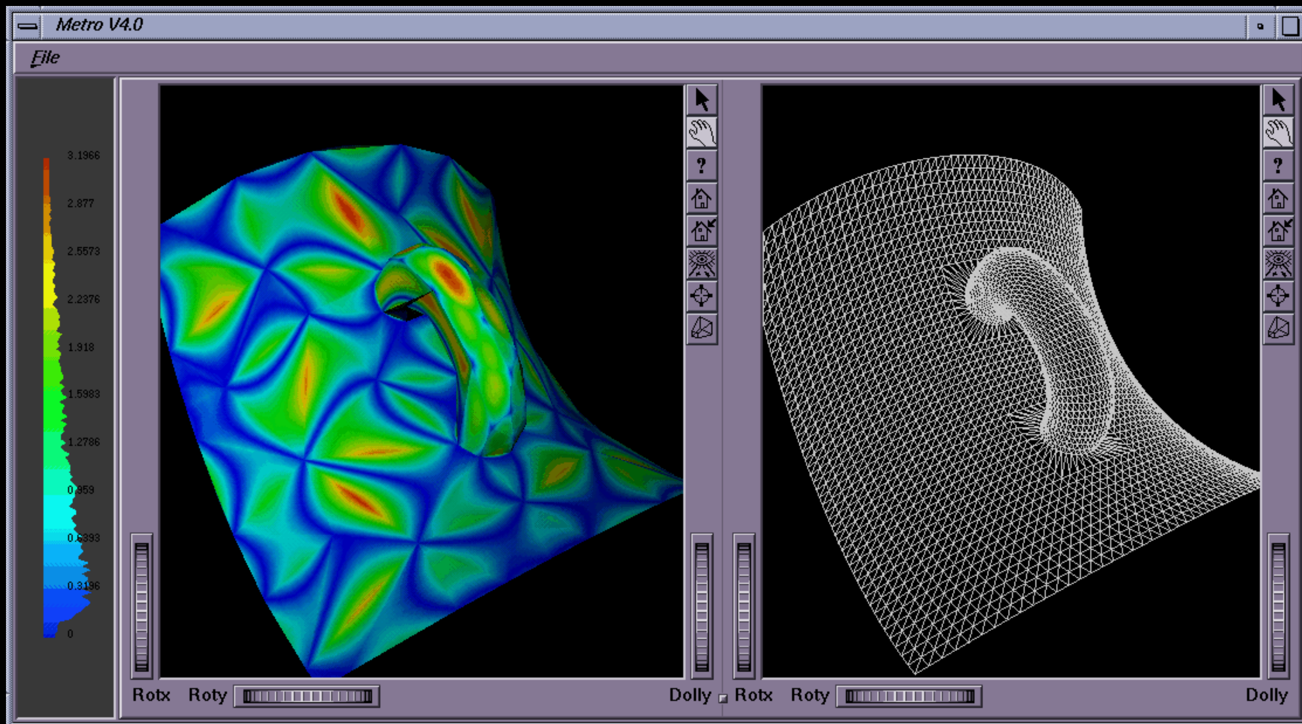
- Vertex-vertex and vertex-plane distance
  - Fast
  - Low error shown after-the-fact, but not guaranteed by metric
- Surface-surface distance
  - Required to guarantee errors

vertex-vertex != surface-surface

Whiteboard: Quadric Error Metrics

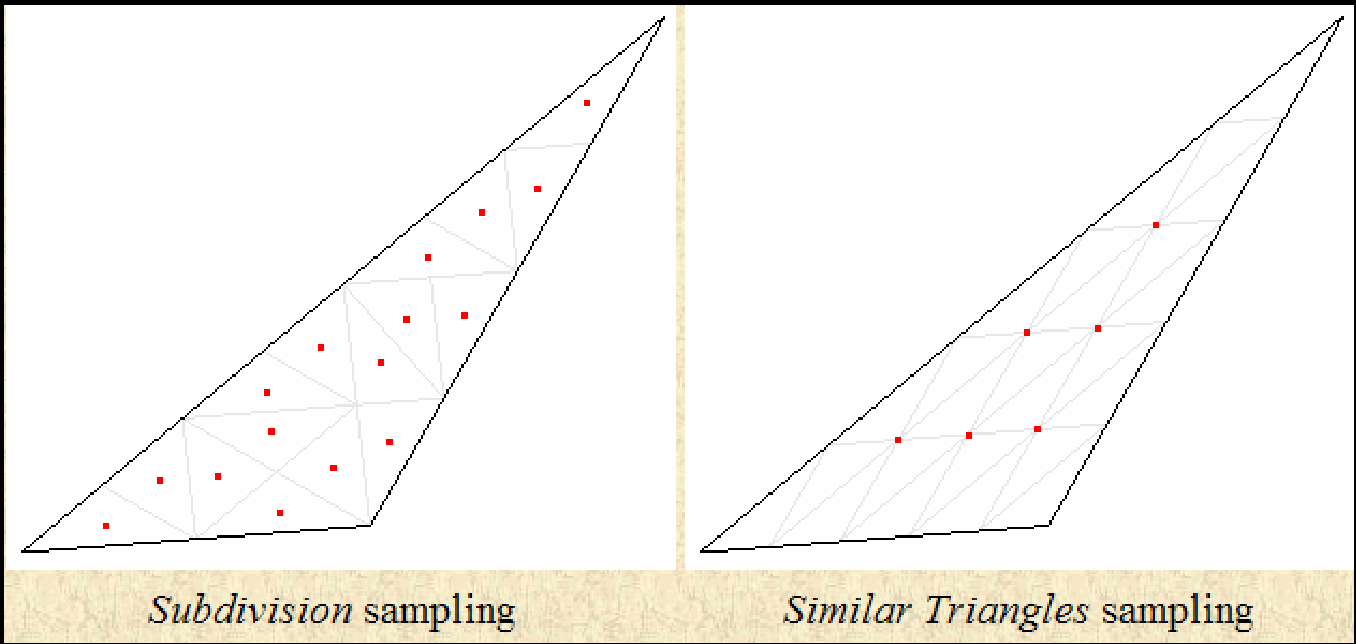


Computing simplification error: Metro



- Evaluates approximate Hausdorff distance
- **Metro: measuring error on simplified surfaces**
  P. Cignoni, C. Rocchini and R. Scopigno
  Computer Graphics Forum, Blackwell Publishers, vol. 17(2), June 1998, pp 167-174

Computing simplification error: Metro



- Evaluates approximate Hausdorff distance
- **Metro: measuring error on simplified surfaces**
  P. Cignoni, C. Rocchini and R. Scopigno
  Computer Graphics Forum, Blackwell Publishers, vol. 17(2), June 1998, pp 167-174

Metro software available online

- **http://vcg.iei.pi.cnr.it/metro.html**

- Usage: **Metro** file1 file2 [*opt*]

  where "file1" and "file2" are the input meshes in PLY or SMF format, and *opt* can be:

  -V disable vertex sampling
  -E disable edge sampling
  -F disable face sampling
  -Sx set the face sampling mode
  where x can be:
  M montecarlo sampling (default)
  S subdivision sampling
  T similar triangles sampling
  -N# set the required number of samples (overrides -A, default: 10 x #faces)
  -A# set the required number of samples *per area unit* (overrides -N)
  -Hxxx.yyy save histogram of error values in the file xxx.yyy

- By default vertex, edge and face sampling are enabled.

- ***Example*:** Metro mesh1.ply mesh2.smf -SS -N50000 -E
- compares the two input meshes using the subdivision method for face sampling with a sampling step of 12.5 samples/area_unit; edge sampling is disabled.
- The numerical results retuned are as follows:
-

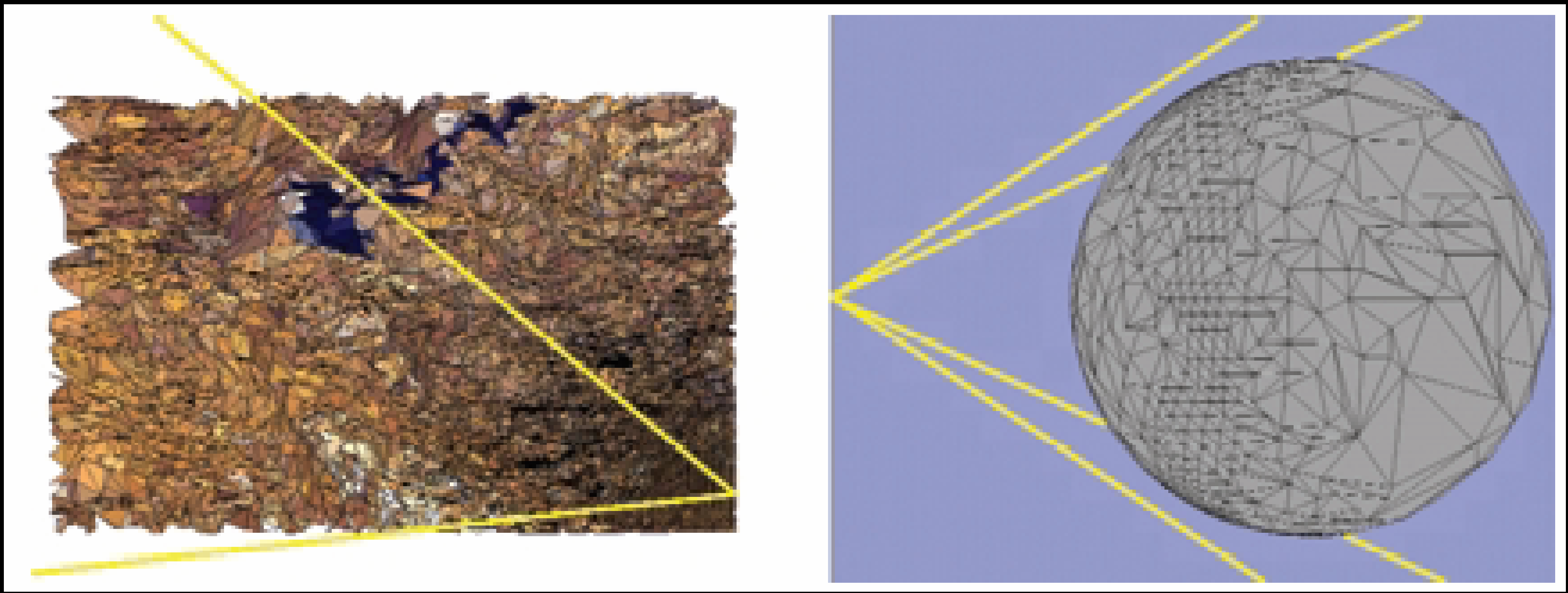
```
-------------------------------
Metro
release date: Nov 26 2002
-------------------------------

reading the mesh 'mesh1.ply'...done
reading the mesh 'mesh2.smf'...done
Mesh info:
M1: 'mesh1.ply'
vertices 5052
faces 10000
area 790329.3211
bbox (-256.5977 -399.9384 -1732.1525)-(210.8112 101.2027 -1472.8544)
bbox diagonal 732.699854
M2: 'mesh2.smf'
vertices 2031
faces 3999
area 789830.8269
bbox (-256.7550 -399.2310 -1732.2300)-(210.8110 101.1740 -1473.0200)
bbox diagonal 732.265616

Forward distance (M1 -> M2):
target # samples : 50000
target # samples/area : 0.063265
Vertex sampling
Subdivision face sampling
distance:
max : 3.160912 (0.004042 with respect to bounding box diagonal)
mean : 0.138594
RMS : 0.204879
# vertex samples 5052
# area samples 44947
# total samples 49999
samples per area unit: 0.063264


Backward distance (M2 -> M1):
target # samples : 50000
target # samples/area : 0.063305
Vertex sampling
Subdivision face sampling
distance:
max : 2.159845 (0.002762 with respect to bounding box diagonal)
mean : 0.129528
RMS : 0.188315
# vertex samples 2031
# area samples 47967
# total samples 49998
samples per area unit: 0.063302


Hausdorff distance: 3.160912 (0.004042 with respect to bounding box diagonal)
Computation time : 3174 ms
# samples/second : 31504.725898
```

Mesh Simplification Considerations

- Type of input mesh?
- Modifies topology?
- Continuous LOD?
- Speed vs. quality?
- In-core or out-of-core?

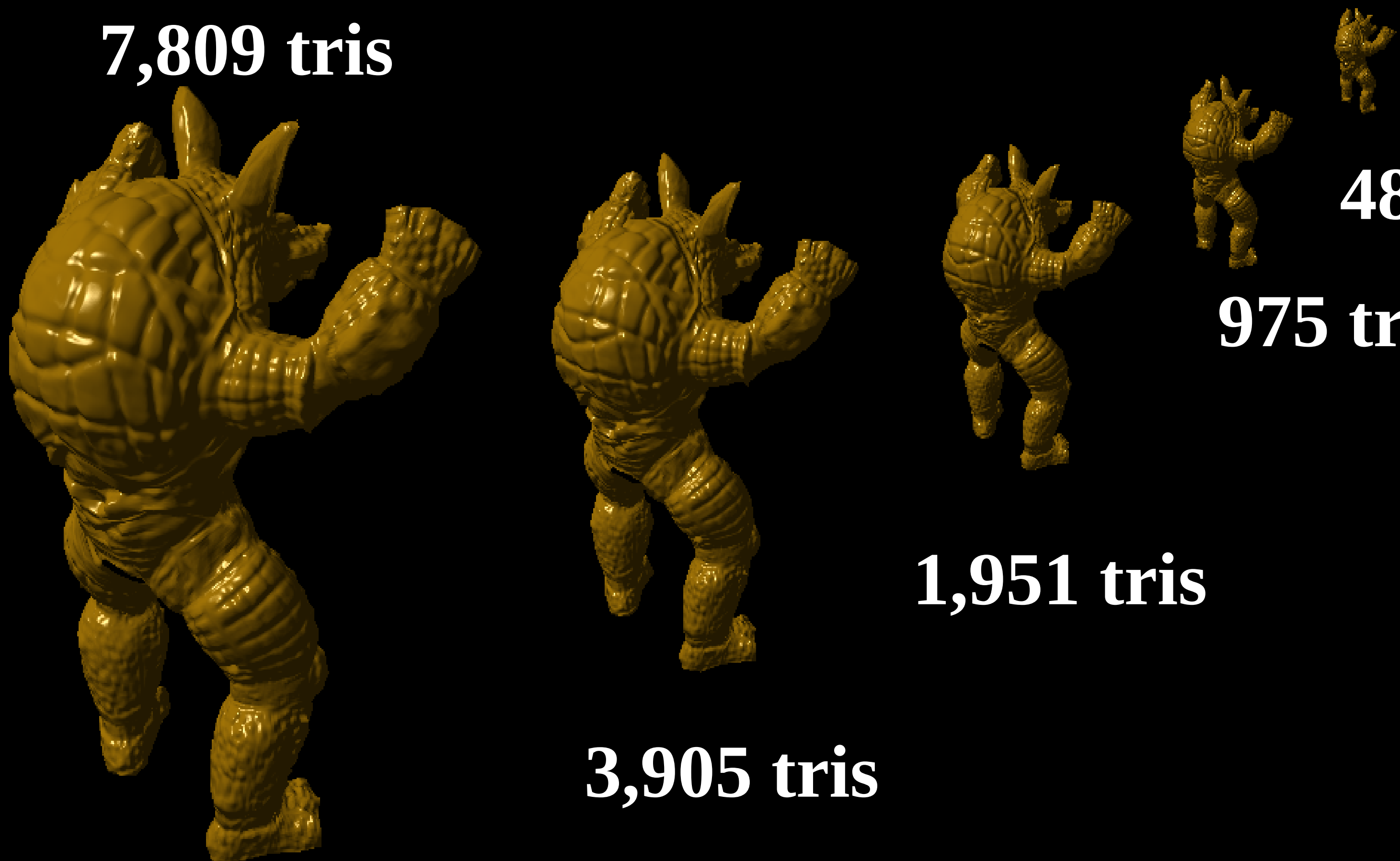
View-Dependent Simplification

- Simplify dynamically according to viewpoint
  - Visibility
  - Silhouettes
  - Lighting



Hugues Hoppe

Appearance Preserving Simplification



Caltech & Stanford Graphics Labs, UNC and Jonathan Cohen

Out-of-Core Algorithms

- Becoming increasingly important for huge meshes

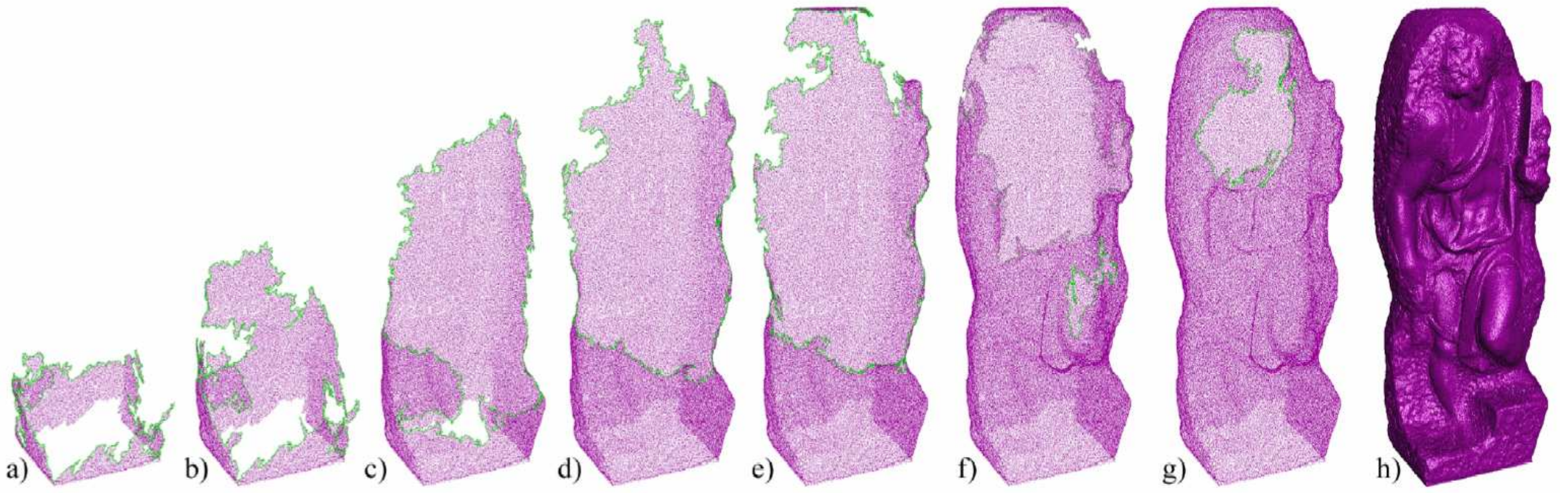


Figure 1: **(a) - (g)** Visualization of the decompression process for the St. Matthew statue. **(h)** Example Out-of-Core Rendering.

- Martin Isenburg and Stefan Gumhold, Out-of-Core Compression for Gigantic Polygon Meshes, SIGGRAPH 2003

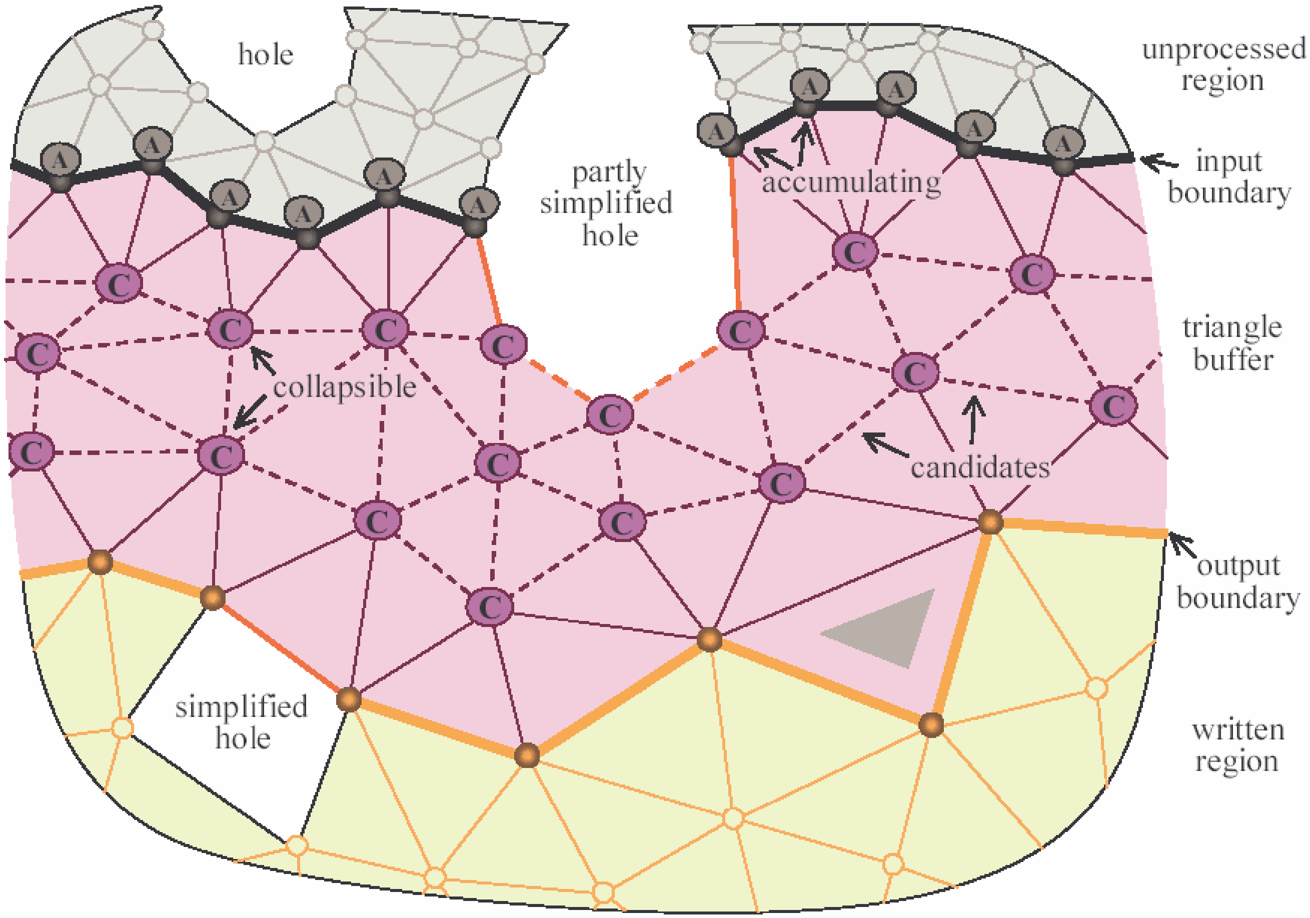


Figure 7: A 2D illustration of buffer-based computation using processing sequences. Such an algorithm, here the simplification algorithm of Wu and Kobbelt [25], operates on a triangle buffer between an input and an output boundary. Triangles generated at the input boundary are read from disk. They are not immediately processed, but used to (re-)fill the buffer in which the actual processing takes place. Their quadric error is added to the accumulating error quadrics of vertices on the input boundary. Edge collapse operations are restricted to those edges (shown dashed) that are not incident to vertices on either boundary. They merge collapsible quadrics. Triangles adjacent to the output boundary empty the buffer and are written to disk. The next candidate for output is the triangle with all three vertices on the output boundary (shown in gray).

Martin Isenburg, Peter Lindstrom, Stefan Gumhold, Jack Snoeyink *Large Mesh Simplification using Processing Sequences*, Visualization'03.

Also see out-of-core simplification work by [Lindstrom, 2000] and others…

Next class…

- More on progressive geometric representations & representing 3D animations
- Read: *"Progressive Meshes," Hoppe, SIGGRAPH 95*
- Coming soon… Point-based rendering

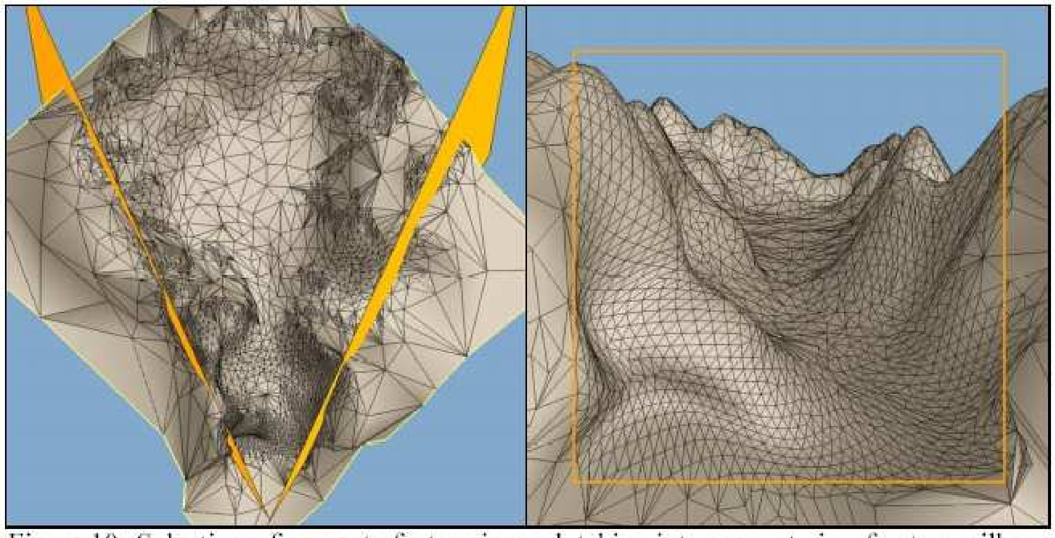


Figure 10: Selective refinement of a terrain mesh taking into account view frustum, silhouette regions, and projected screen size of faces (7,438 faces).