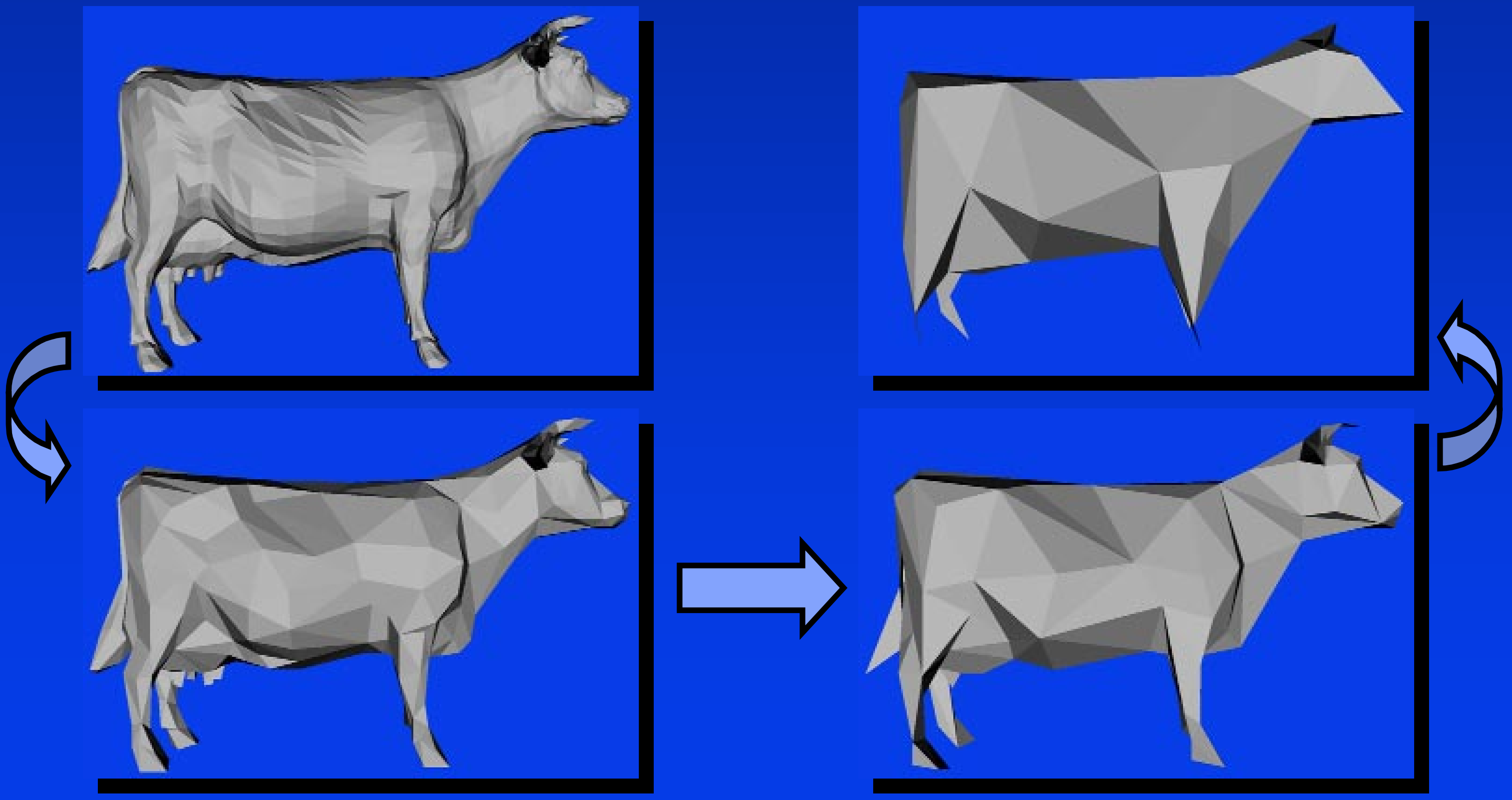
Surface Simplification Using Quadric Error Metrics

Carnegie Mellon

Michael Garland
Paul S. Heckbert

August 1997

Surface Simplification



Overview

Fast algorithm & good quality approximation

- Practical trade-off of efficiency and quality

Convenient characterization of error/shape

- Compact & efficient

Fairly general surface handling

- Simplify non-manifold surfaces
- Simplify topology as well as geometry

Surface Simplification: Our World View

Simplify overly complex models rapidly

- Minimize error between original and approximate

Produce multiresolution models for rendering

- Appearance is paramount
  - *Holes and gaps usually disappear in the distance*
  - *Topology can be simplified*
- Want fine-grained operation for altering surface

Assumptions About Input

Triangulated model with valid connectivity

- Corners which coincide in space share vertices

Surface need not have manifold topology

- Edges can border any number of faces
- Vertices shared by arbitrary collection of faces

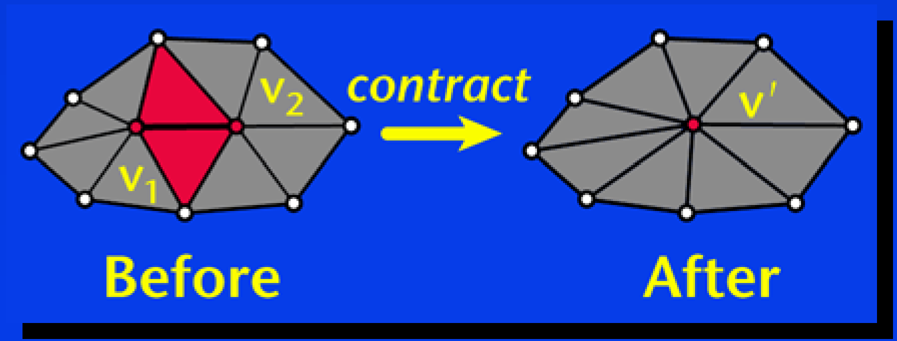
Application domain does not rely on topology

- Medical Imaging: can't leave out hole in the heart

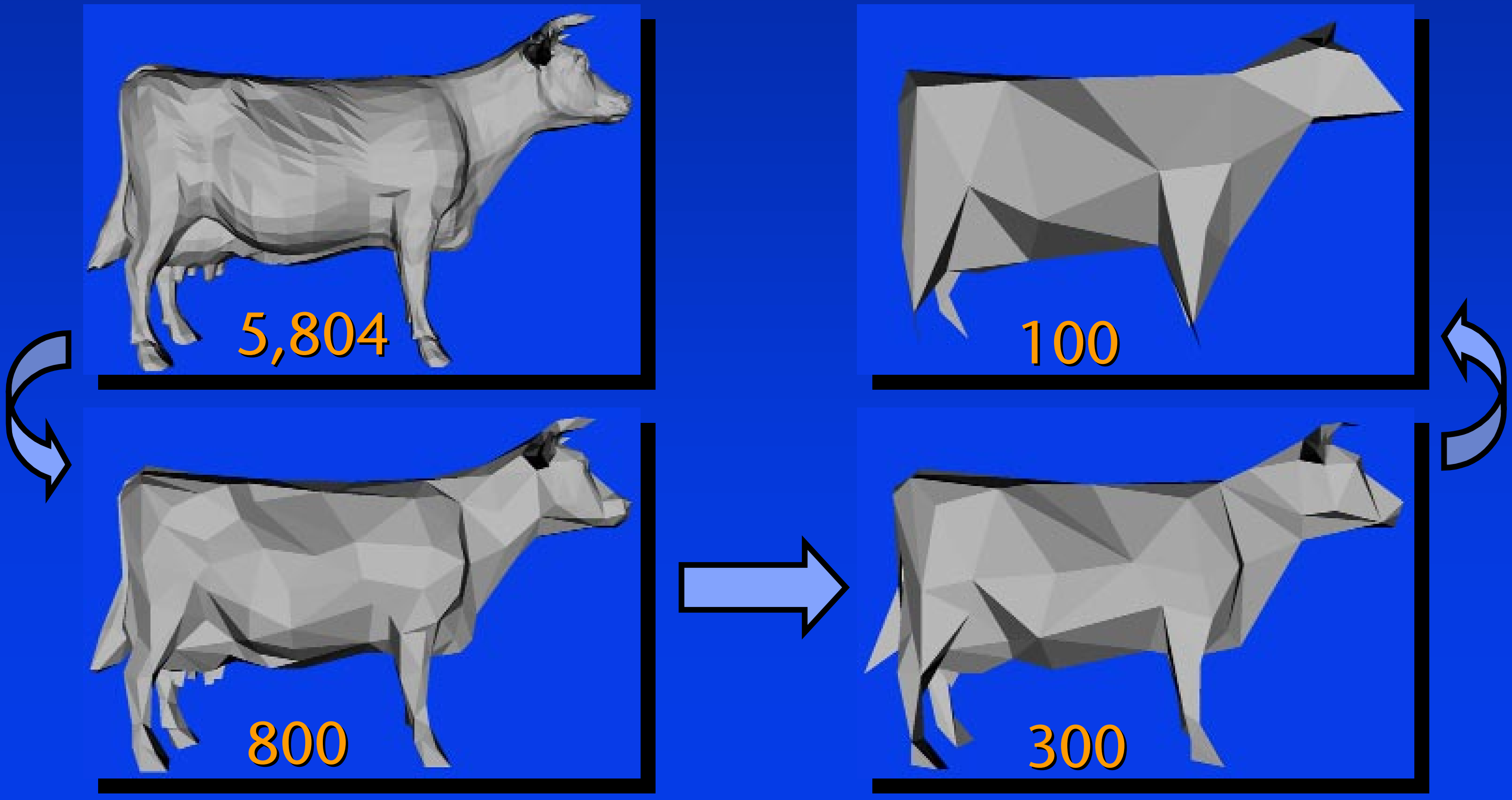
Our Basic Operation: Vertex Pair Contraction

Contract vertex pair $(v_1, v_2) \rightarrow v'$

- Move v_1 and v_2 to position v'
- Replace all occurrences of v_2 with v_1
- Remove v_2 and degenerate triangles
- Typically, we contract edges, as others have done



Simplifying a Cow in Under One Second



How We Measure Error

Measure error at current vertices

For a given point v, measure sum of squared distances to associated set of planes

- Each vertex v has an associated set of planes
  - *Initialize with planes of incident faces in original*
  - *Merge sets when contracting pairs*
  - *Initial error of each vertex is 0*

How We Measure Error

For a given point v, measure sum of squared distances to associated set of planes

- Locally, distance to plane equals distance to face
- More efficient & compact than distance to face
- Vertex's set of planes records its history
  - *Ronfard & Rossignac stored these sets explicitly*
  - *We track them implicitly using quadrics*

Measuring Error With Quadrics

Sum of squared distance to a set of planes

- Vertex v has associated set of planes
- Planes defined by $ax+by+cz+d = 0$, $a^2+b^2+c^2=1$

$$\mathbf{Error}(\mathbf{v}) = \mathbf{v}^T \left(\sum \mathbf{K}_p\right)\mathbf{v}$$

- Each plane p defines a quadric matrix K_p
- Set of planes represented by sum of quadrics

What's a Quadric Error Metric?

A quadric matrix Q is a 4x4 symmetric matrix

Assigns real number to every point v by $v^T Q v$

$$\mathbf{v}^T \mathbf{Q} \mathbf{v} = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{12} & q_{22} & q_{23} & q_{24} \\ q_{13} & q_{23} & q_{33} & q_{34} \\ q_{14} & q_{24} & q_{34} & q_{44} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

What's a Quadric Error Metric?

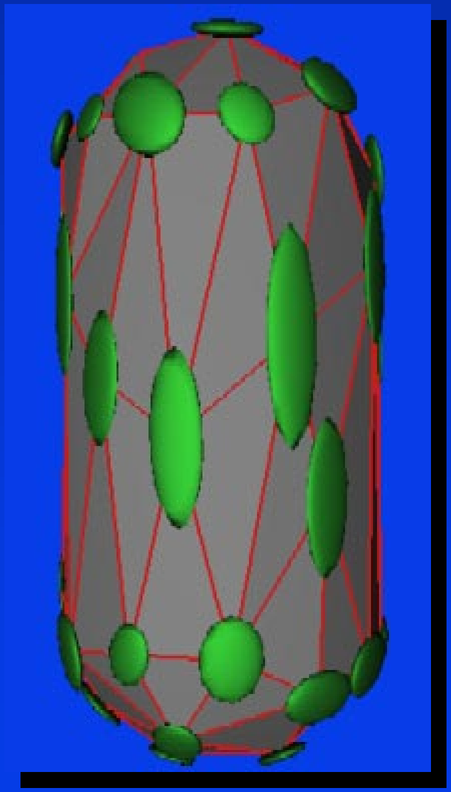
We get a 2nd degree polynomial in x, y, & z

Level surface $v^TQv = k$ is a quadric surface

- Ellipsoid, paraboloid, hyperboloid, plane, etc.

$$\begin{aligned} v^T Qv = {} & q_{11}x^2 + 2q_{12}xy + 2q_{13}xz + 2q_{14}x \\ & + q_{22}y^2 + 2q_{23}yz + 2q_{24} \\ & + q_{33}z^2 + 2q_{34}z + q_{44} \end{aligned}$$

But What Are These Quadrics Really Doing?



Almost always ellipsoids

- When **Q** is positive definite

Characterize error at vertex

- Vertex at center of each ellipsoid
- Move it anywhere on ellipsoid with constant error

Capture local shape of surface

- Stretch in least curved direction

Algorithm Outline

Initialization

- Compute quadric **Q** for each vertex
- Select set of valid vertex pairs (edges + non-edges)
- Compute minimal cost candidate for each pair

Iteration

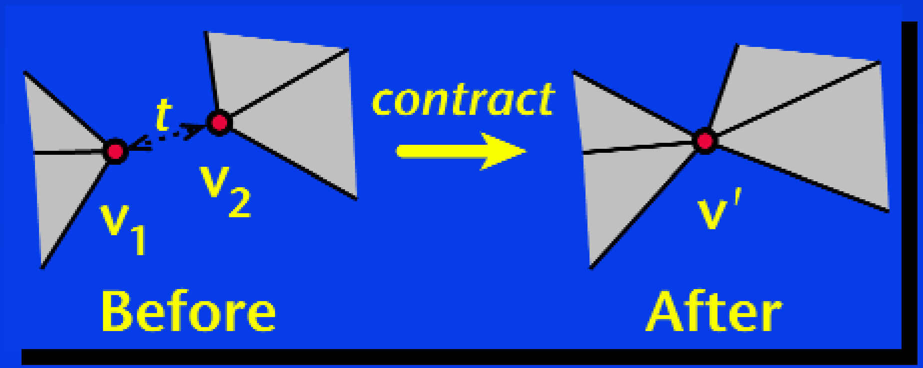
- Select lowest cost pair (v_1,v_2)
- Contract (v_1,v_2) — **Q** for new vertex is $\mathbf{Q}_1+\mathbf{Q}_2$
- Update all pairs involving v_1 & v_2

Non-Edge Vertex Pairs

In addition to edges, select some non-edges

- Select pairs separated by less than distance *t*

Allows us to simplify topology of object



Video Clip

Additional Algorithm Details

Open boundaries may be eaten away

- For each edge with a single incident face
  - *Find plane through edge perpendicular to face*
  - *Assign large weight & add into endpoints*

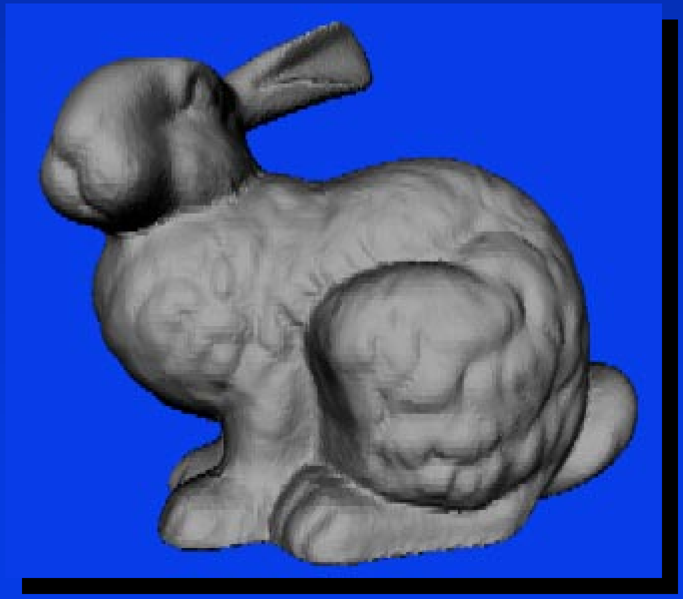
Contraction may fold mesh over on itself

- Check normals before & after; reject if any flip

Error metric dependent on initial mesh

- Weight quadrics by triangle area

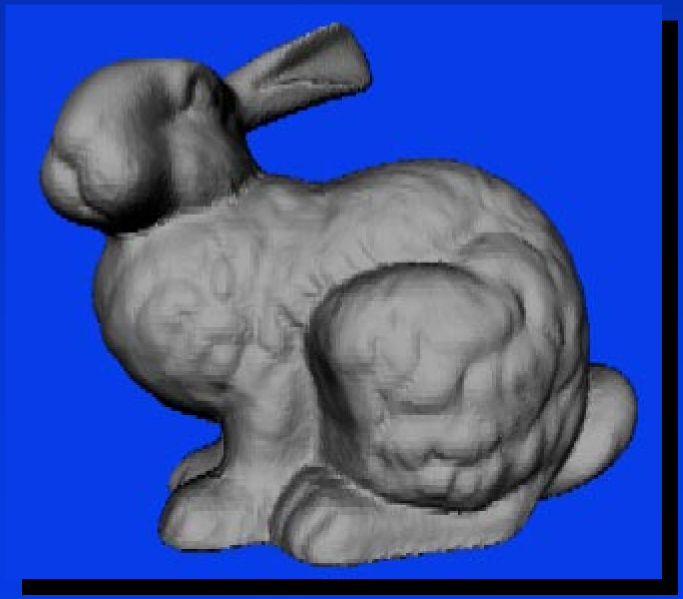
Sample Model: Stanford Bunny



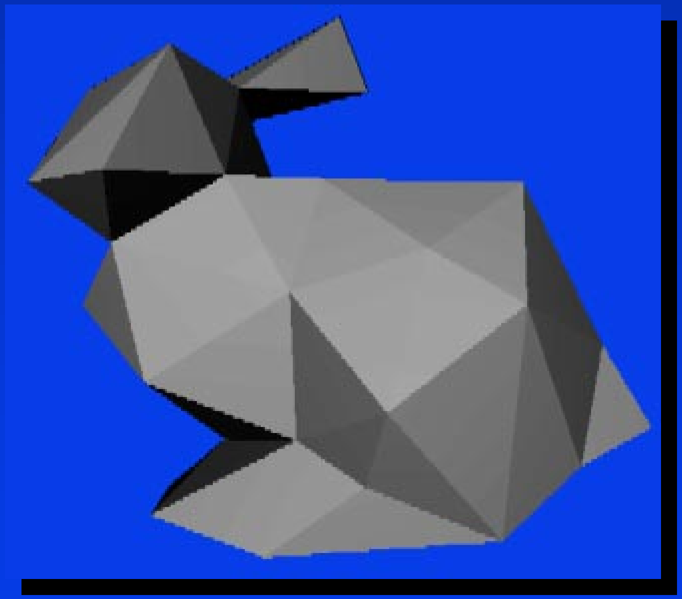
69,451 faces

1,000 faces (30 sec)

Sample Model: Stanford Bunny

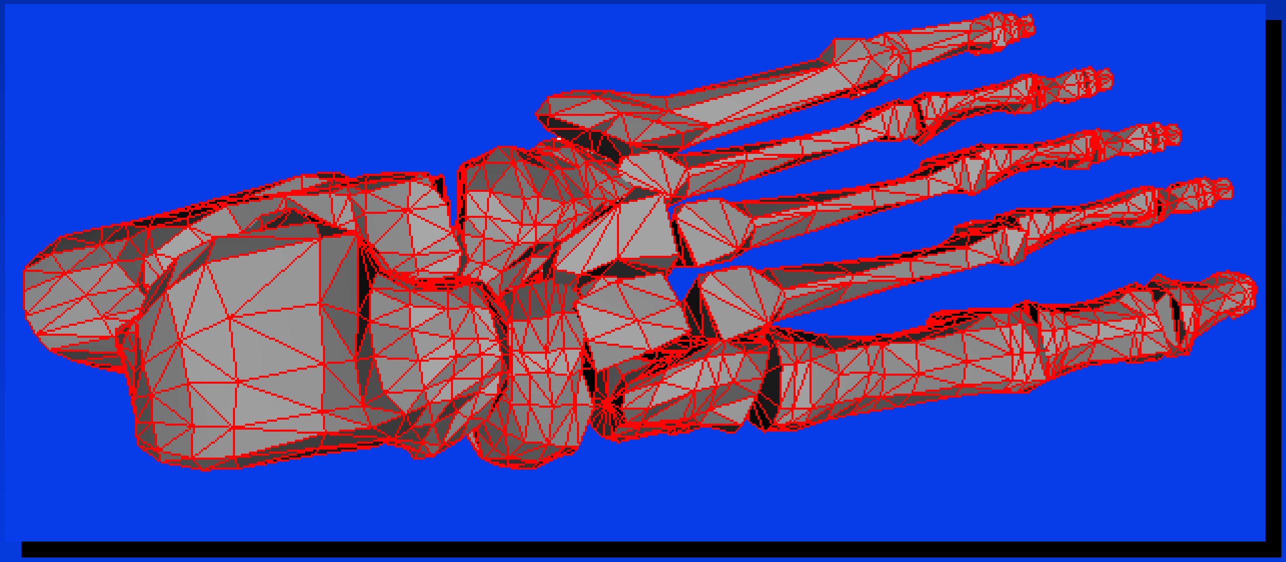


69,451 faces

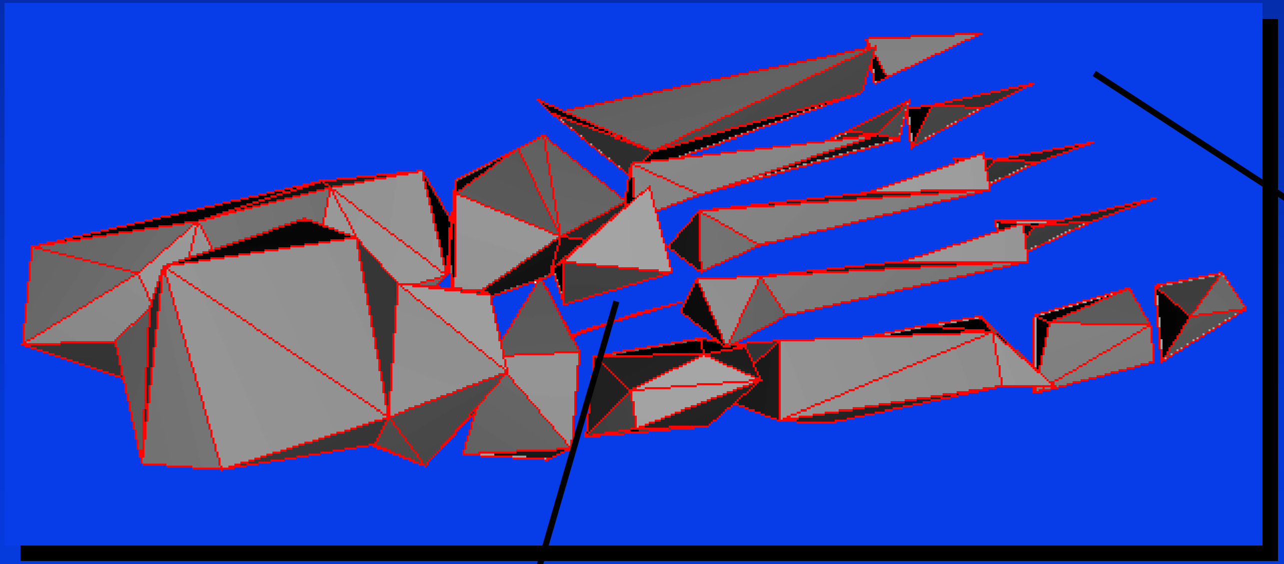


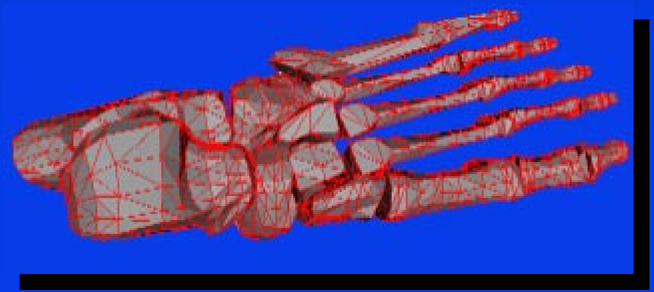
100 faces (30 sec)

Foot Model With Many Separate Components

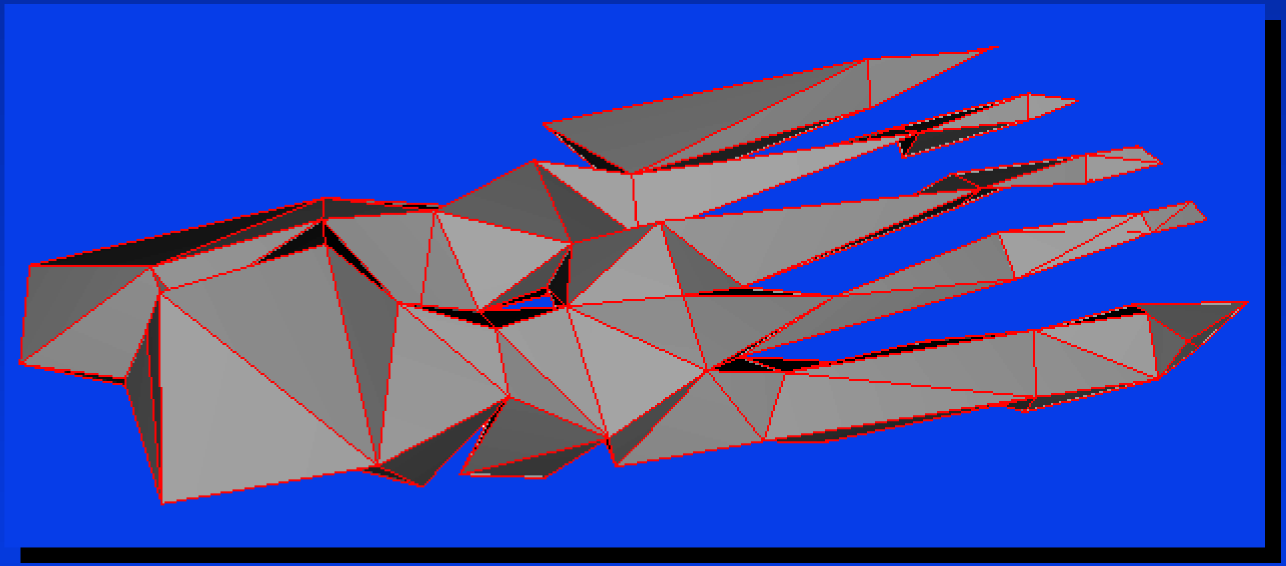


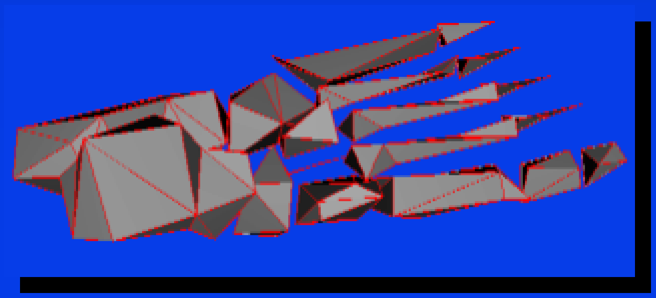
Foot Model Approximation (Edge Contraction Only)



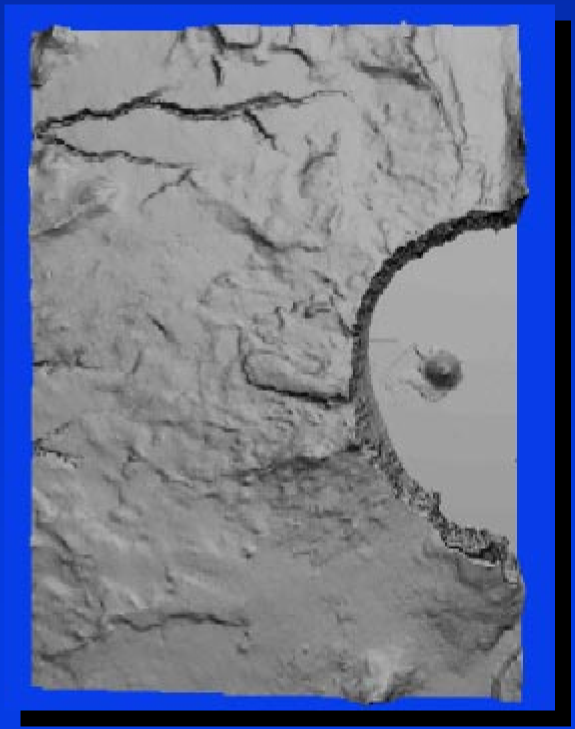


Foot Model Approximation (Non-edge Selection)





Sample Model: Crater Lake Terrain Data



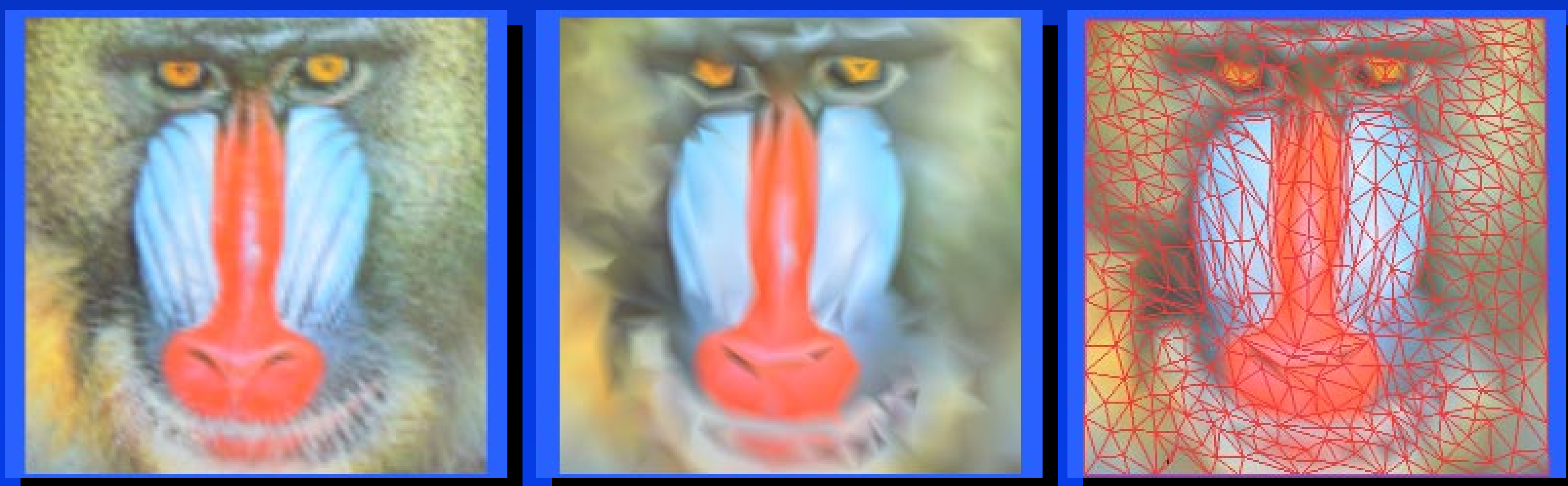
Original 199,114 faces

1,000 faces (54 sec)

Handling Surfaces With Colored Vertices

Before: $v = (x,y,z,1)$ ***and Q is a 4x4 matrix***

Now: $v = (x,y,z,r,g,b,1)$ ***and Q is a 7x7 matrix***



19,404 faces

1,000 face approximation

Related Prior Work

Vertex clustering [Rossignac-Borrel 93; Low-Tan 96]

- Partition space; merge vertices within cells
- Fast and general, but comparatively low quality

Vertex Decimation [Schroeder *et al* 92; Soucy-Laurendeau 96]

- Remove single vertex; retriangulate neighborhood
- Comparable or lesser quality & typically slower
- Typically limited to manifold regions
- Less aggressive simplification

Related Prior Work

Edge Collapse [Hoppe 96; Guéziec 95; Ronfard-Rossignac 96]

- Iteratively contract edges; differ in edge selection
- Some higher quality (e.g., Hoppe 96), but significantly slower
- Some comparable quality (e.g., Ronfard-Rossignac), but not as general or efficient

Summary

Efficient algorithm with quality results

- Simple to implement; fast simplification
- High quality approximations
- Good compromise between highest quality and fastest simplification

Simplifies both geometry and topology

Can also handle surface properties (e.g., color)

Future Work

Constructing multiresolution models

- Progressive meshes/simplicial complexes

Quality and reliability joining components

Planar regions are simplified rather randomly

- Needs some bias towards good planar meshes

Further Details & Free Stuff

Free sample implementation,

further information,

available now …

http://www.cs.cmu.edu/~garland/quadrics/