Mesh Simplification

Thomas Funkhouser
Princeton University
COS 526, Fall 2006

Mesh Simplification

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

Division, Viewpoint, Cohen

Mesh Simplification Motivation

Interactive visualization
- Store and draw simpler version for distant objects

Simulation proxies
- Store and process simpler version for approximate solutions first, and then refine details for "hits"

Mesh Simplification Goals

Reduce number of polygons
- Less storage
- Faster rendering
- Simpler manipulation

Desirable properties
- Generality, efficiency, scalability
- Produces "good" approximation
  - § Geometric
  - § Visual

Stanford Graphics Lab

Mesh Simplification Overview

Some algorithms
- Vertex clustering
- Mesh retiling
- Mesh optimization
- Mesh decimation

Considerations
- Speed of algorithm
- Quality of approximation
- Generality (types of meshes)
- Topology modifications
- Control of approximation quality
- Continuous LOD
- Smooth transitions

Vertex Clustering

Partition vertices into clusters and replace all vertices in each cluster by one representative

10,108 polys 1,383 polys 474 polys 46 polys

Rossignac

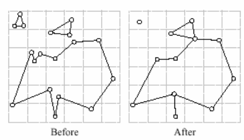
Vertex Clustering

Example algorithm [Rossignac93]:

1. Build grid containing vertices
2. Merge vertices in same grid cell
   a. Select new position for representative vertex
   b. Collapse degenerate edges and faces

Comments:

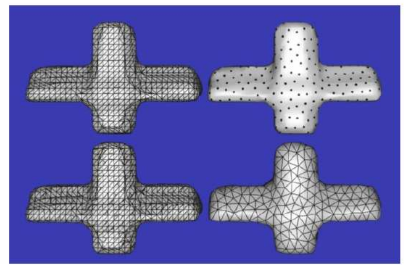
- Fast
- Collapses topology
- Low quality
- Hard to control



Rossignac

Mesh Re-Tiling

Resample mesh with "uniformly spaced" vertices



Turk

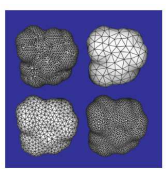
Mesh Re-Tiling

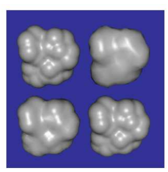
Example algorithm [Turk92]:

- Generate random points on surface
- Use diffusion/repulsion to spread them uniformly
- Tessellate vertices (many details here)

Comments:

- Slow
- Blurs sharp features



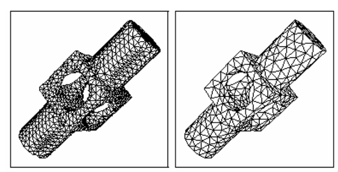


Turk

Mesh Optimization

Apply optimization procedure to minimize an objective function E(K,V)

$$E(K,V) = E_{dist}(K,V) + E_{rep}(K) + E_{spring}(K,V)$$



(e) Optimum for fixed K_0 (f) Optimum with $\kappa = 10^{-2}$

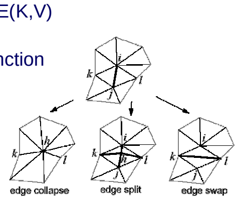
Hoppe

Mesh Optimization

Example algorithm [Hoppe92]:

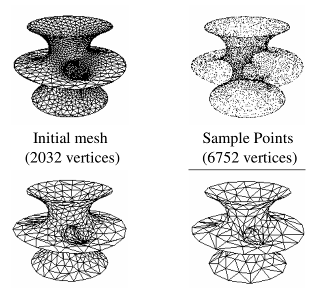
Iterate with a decreasing spring term …

1. Randomly modify topology with edge collapse, edge swap, or edge split
2. Move vertices to minimize E(K,V)
3. Keep topological change if reduce overall objective function



Hoppe

Mesh Optimization



Initial mesh (2032 vertices)

Sample Points (6752 vertices)

$c_{rep}=10^{-5}$ (487 vertices)

$c_{rep}=10^{-4}$ (239 vertices)

Hoppe

Mesh Decimation

Apply iterative, greedy algorithm to gradually reduce complexity of mesh
- Measure error of possible decimation operations
- Place operations in queue according to error
- Perform operations in queue successively
- After each operation, re-evaluate error metrics

Mesh Decimaion Operations

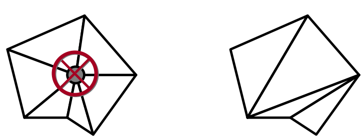
General idea:
- Each operations simplifies model by small amount
- Apply many operations in succession

Types of operations
- Vertex remove
- Edge collapse
- Vertex cluster

Vertex Remove

Method
- Remove vertex and adjacent faces
- Fill hole with new triangles (reduction of 2)

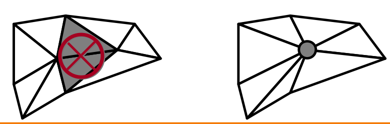
Properties
- Requires manifold surface around vertex
- Preserves local topological structure



Edge Collapse

Method
- Merge two edge vertices to one
- Delete degenerate triangles

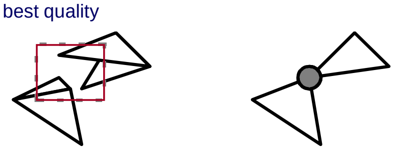
Properties
- Requires manifold surface around vertex
- Preserves local topological structure
- Allows smooth transition



Vertex Cluster

Method
- Merge vertices based on proximity
- Triangles with repeated vertices become edge or point

Properties
- General and robust
- Allows topological changes
- Not best quality



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

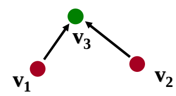
Mesh Decimation Error Metrics

Motivation
- Promote accurate 3D shape preservation
- Preserve screen-space silhouettes and pixel coverages

Types
- Vertex-Vertex Distance
- Surface-Surface Distance
- Point-Surface Distance
- Vertex-Plane Distance

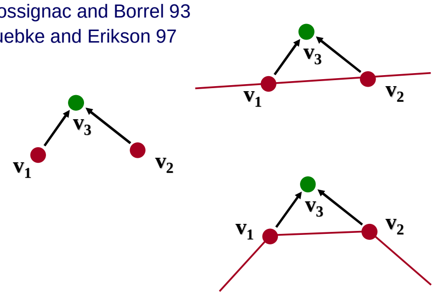
Vertex-Vertex Distance

E = max(|| v3-v1||, || v3-v2 ||)
- Rossignac and Borrel 93
- Luebke and Erikson 97



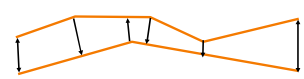
Vertex-Vertex Distance

E = max(|| v3-v1||, || v3-v2 ||)
- Rossignac and Borrel 93
- Luebke and Erikson 97



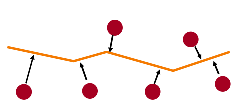
Surface-Surface Distance

Error is maximum distance between original and simplified surface
- Tolerance Volumes - Guéziec 96
- Simplification Envelopes - Cohen/Varshney 96
- Hausdorf Distance - Klein 96
- Mapping Distance - Bajaj/Schikore 96, Cohen et al. 97



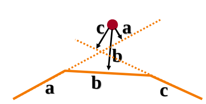
Point-Surface Distance

Error is sum of squared distances from original vertices to closest point on simplified surface
- Hoppe et al. 92



Vertex-Plane Distance

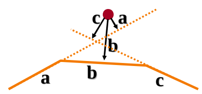
Error is based on distances from original vertices to planes of faces in simplified surface
- Max distance to plane
  - § Maintain set of planes for each vertex [Ronfard96]
- Sum of squared distances
  - § Approximated by quadric at each vertex [Garland97]



Quadric Error Metric

Error is sum of squared distances from original vertices to planes of faces in simplified surface

- How compute
- When vertices are merged, merge sets



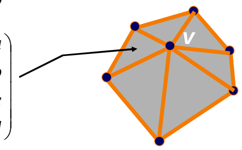
Quadric Error Metric

Sum of squared distances from vertex to planes:

$$\Delta_{\mathbf{v}} = \sum_{\mathbf{p}} Dist(\mathbf{v},\mathbf{p})^2$$

$$\mathbf{v} = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}, \quad \mathbf{p} = \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}$$

$$Dist(\mathbf{v},\mathbf{p}) = ax + by + cz + d = \mathbf{p}^{\mathrm{T}}\mathbf{v}$$



Quadric Error Metric

Common mathematical trick:

- quadratic form = symmetric matrix Q multiplied twice by a vector

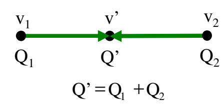
$$\Delta = \sum_{\mathbf{p}} (\mathbf{p}^{\mathrm{T}}\mathbf{v})^2$$
$$= \sum_{\mathbf{p}} \mathbf{v}^{\mathrm{T}}\mathbf{p}\mathbf{p}^{\mathrm{T}}\mathbf{v}$$
$$= \mathbf{v}^{\mathrm{T}}\left(\sum_{\mathbf{p}} \mathbf{p}\mathbf{p}^{\mathrm{T}}\right)\mathbf{v}$$
$$= \mathbf{v}^{\mathrm{T}}\mathbf{Q}\mathbf{v}$$

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

Using Quadric Error Metric

Approximate error of edge collapses

- Each vertex v_i has associated quadric Q_i
- Error of collapsing v_1 and v_2 to v' is $v'^{T}Q_1v' + v'^{T}Q_2v'$
- Quadric for new vertex v' is $Q' = Q_1 + Q_2$



Using Quadric Error Metric

Find optimal location v' after collapse:

$$\mathbf{Q}' = \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{12} & q_{22} & q_{23} & q_{24} \\ q_{13} & q_{23} & q_{33} & q_{34} \\ q_{14} & q_{24} & q_{34} & q_{44} \end{bmatrix}$$

$$\min_{\mathbf{v}'} \mathbf{v}'^{\mathrm{T}}\mathbf{Q}'\mathbf{v}': \quad \frac{\partial}{\partial x} = \frac{\partial}{\partial y} = \frac{\partial}{\partial z} = 0$$

Using Quadric Error Metric

Find optimal location v' after collapse:

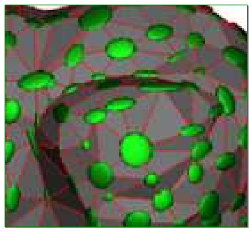
$$\begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{12} & q_{22} & q_{23} & q_{24} \\ q_{13} & q_{23} & q_{33} & q_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{v}' = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\mathbf{v}' = \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{12} & q_{22} & q_{23} & q_{24} \\ q_{13} & q_{23} & q_{33} & q_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Quadric Error Visualization

Ellipsoids: iso-error surfaces

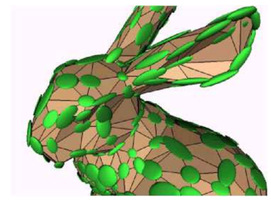
- Smaller ellipsoids represent greater error for a given motion
- Lower error for motion parallel to surface
- Lower error in flat regions than at corners
- Elongated in "cylindrical" regions near ridges



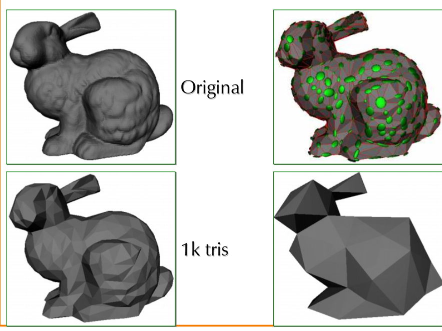
Quadric Error Visualization

Ellipsoids: iso-error surfaces

- Smaller ellipsoids represent greater error for a given motion
- Lower error for motion parallel to surface
- Lower error in flat regions than at corners
- Elongated in "cylindrical" regions near ridges



Quadric Error Metric Results



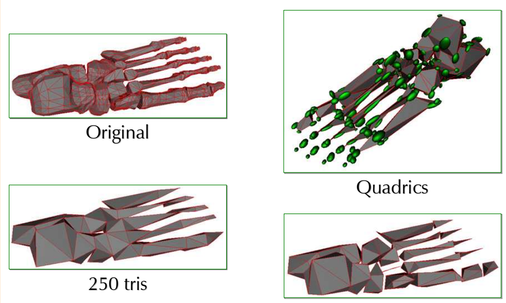
Original

Quadrics

1k tris

100 tris

Quadric Error Metric Results



Original

Quadrics

250 tris

250 tris, edge collapses only

Quadric Error Metric Details

Boundary preservation: add planes perpendicular to *boundary edges*

Prevent foldovers: check for normal flipping

Create *virtual edges* between vertices closer than some threshold *t*

Look in Garland and Heckbert, SIGGRAPH 1997

Mesh Decimation Summary

Properties

- Fast (with quadric error metric)
- Good quality approximation
- Only connected meshes
- Allows topology modifications (if allow vertex merging)
- Allows control over amount of simplification
- Continuous LOD
- Smooth transitions