Polygonal Simplification

Thomas A. Funkhouser

Today's Lecture

- **Polygonal Simplification**
- **Project Proposals**
  – What it should contain
- **Programming Assignment #2**
  – RING software overview

Polygonal Simplification

Simplification Goals

- **Reduce number of polygons**
  - Faster rendering
  - Less storage
  - Simpler manipulation
- **Find ‘‘good’’ approximation**
  - Visual approximation
  - Geometric approximation
  - Data approximation
- **Other desirable qualities**
  - General applicability
  - Efficiency (time)
  - Scalability (space)
  - Preserve attributes
    - Texture coordinates
    - Normals
    - Other vertex attributes

Polygonal Data Sources

- **Digitizers**
  - Laser range finders
  - 3D coordinate mapping
- **Scientific Visualization**
  - Iso–surface generation
  - Geometry extraction
- **Terrain**
  - Satellite
  - Radar mapping
  - Sonar
- **Modeling**
  - Graphics representations
  - Tesselation

Simplification Approaches

- **Geometry refinement**
  – Adaptive subdivision
- **Geometry resampling**
  – Mesh re–tiling
  – Wavelet reconstruction
- **Geometry decimation**
  – Vertex decimation
  – Edge contraction
  – Vertex clustering
  – Vertex merging

Algorithm Characteristics

- **Speed vs. quality?**
- **Type of mesh?**
  – Height field or parametric
  – Manifold surface
  – Polygon soup
- **Modifies topology?**
  – Progressive
- **Continuous LOD?**
  – Progressive
  – Wavelet
- **View−dependent refinement?**

Adaptive Refinement

- **Input:**
  - Height field or parametric
- **Preprocessing:**
  - Build power–of–two pyramid/quad tree
- **Run–time:**
  - Decide for each node whether to decend tree further based on error
- **Comments:**
  - Compact storage (x & y are implicit)
  - Simple and fast
  - Poor quality approximations
  - Must prevent cracks during display

DeHaemer & Zyda '91

Vertex Decimation

- **Input:**
  - Manifold surface mesh
  - Distance−based error parameters

- **Algorithm:**
  For each vertex with error $< \varepsilon$
    1. Delete vertex (making hole)
    2. Re−triangulate hole (loop splitting)

- **Comments:**
  - Simple and fast
  - Constrained to use original vertices
  - Moderate quality approximations
  - Must choose error parameters

Schroeder et al. '92

Mesh Re–Tiling

- **Input:**
  - Manifold surface mesh
  - Number of output vertices (M)

- **Algorithm:**
  1. Randomly spread M new vertices on surface
  2. Re–distribute new vertices along surface by simulating repulsion forces
  3. Insert new vertices into original mesh
  4. One by one, remove original vertices

- **Commentss:**
  - Lose sharp discontinuities
  - Cannot specify error tolerance

Turk '92

Mesh Optimization

- **Input:**
  - Manifold surface mesh
  - Number of output vertices

- **Algorithm:**

  For 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 reduces overall objective function, E(K,V)

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

- **Comments:**
  - High–quality approximations
  - Slow (hours)

Hoppe et al. '93

Topology Decimation

- **Input:**
  - Surface mesh
  - Number of output vertices or maximum error
- **Algorithm:**
  Iteratively replace a set of vertices by
  a single representative one
- **Issues:**
  - What order to select vertex merges
  - How to select representative vertex
- **Examples:**
  - Edge collapsing
  - Vertex clustering
  - Pairwise vertex merging

Edge Collapsing

- **Input:**
  - Manifold surface mesh
- **Vertex selection:**
  - Must be two vertices on same edge
  - Order edge collapses by estimated ΔE
- **Representative vertex:**
  - Midpoint
  - Weighted average of endpoints
  - Optimize to minimize error
- **Comments:**
  - Continuous LODs
  - Approximate error bounds
  - Must worry about mesh folding

Ronfard et al. '96
Xia & Varshney '96
Hoppe et al. '96

Vertex Clustering

- **Input:**
  - Arbitrary polygon soup
- **Vertex selection:**
  - All vertices in same grid/octree cell
- **Representative vertex:**
  - Weighted average of selected vertices
- **Comments:**
  - Fast
  - Poor quality

Rossignac & Borrel. '93
Low & Tan '97
Luebke et al. '97

Vertex Merging

- **Input:**
  - Arbitrary polygon soup
- **Vertex selection:**
  - Pair of vertices that minimizes error
- **Representative vertex:**
  - Minimize error metric
- **Comments:**
  - Continuous LODs
  - Approximate error bounds
  - Must worry about mesh folding
  - Arbitrary topology

Garland & Heckbert. '97
Popovic & Hoppe '97

Simplification Envelopes

- **Input:**
  - Manifold surface mesh
- **Idea:**
  - Bound error by requiring simplified surface to lie within surfaces offset by ε
  - Can be combined with most other meshes

Summary

- ♦ **Geometry refinement**
  - – Adaptive subdivision
- ♦ **Geometry resampling**
  - – Mesh re−tiling
  - – Wavelet reconstruction
- ♦ **Geometry decimation**
  - – Vertex decimation
  - – Edge contraction
  - – Vertex clustering
  - – Vertex merging

Project Proposals

- **Goal**
  - What am I going to do? Who would benefit?
- **Problem Statement**
  - Why is it hard?
- **Previous Work**
  - What have others tried?
- **Approach**
  - What approach am I going to try?
  - Why do I think it will work well?
- **Methodology**
  - What steps (task list) are required?
  - Which of these steps is particularly hard?
  - What to do if the hard steps don’t work out?
- **Metrics**
  - How will I measure success?