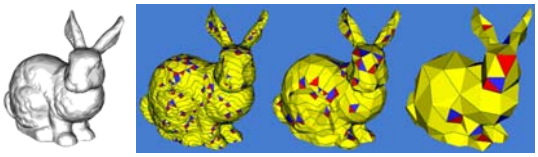
Surface Simplification and Optimization Multiresolution Representations

ENGN2911I
3D Photography and Geometry Processing
Brown Spring 2008
Gabriel Taubin

Surface Simplification

- Algorithms to reduce the number of vertices and faces while preserving geometric approximation

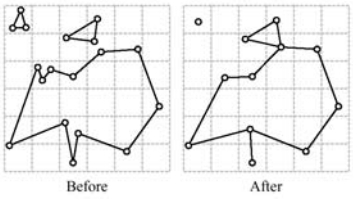


Surface Simplification Methods

- Vertex Clustering
  - Rossignac-Borrel 93
- Vertex Removal
  - Schroeder-Zarge-Lorensen 92
- Edge Collapse
  - Garand Heckbert 97

Vertex Clustering

- Vertex Clustering
  - Quantize coordinates wrt Bbox
  - Identify vertices with same coordinates
  - Remove empty triangles



Vertex Clustering Algorithm

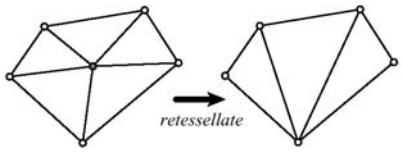
- Quantize coordinates wrt Bbox
- Assign a new vertex index to each occupied cell
- Determine coordinates of new vertices
- Construct newVertex index look-up table
- Replace vertex indices in coordIndex
- Remove empty triangles from coordIndex

Vertex Clustering

- Advantages
  - Simple to implement
  - Works on large scenes with multiple objects
  - No manifold restriction
- Disadvantages
  - Produces non-manifold meshes
  - Quality of simplified model is not very good

Vertex Removal

- Vertex Removal
  - Choose "removable" vertex
  - Remove vertex
  - Triangulate hole



Vertex Removal

- Advantages
  - Better surface quality
  - Maintains manifold structure
- Disadvantages
  - Complexity
  - Multiple hole triangulations

Vertex Removal Algorithm 1

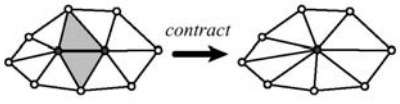
- Put all "removable" vertices in priority queue according to "removal error"
- While queue is not empty
  - Delete minimum vertex from queue
  - Remove vertex and triangulate hole
  - Remove all incident vertices from the queue, recompute "removal error", re-insert in queue
- Need dynamic data structures

Vertex Removal Algorithm 2

- Put all "removable" vertices in priority queue according to "removal error"
- Choose an "independent set" of vertices
  - Initialze set to empty
  - While queue is not empty
    - Delete minimum vertex from queue
    - If vertex is independent of vertices in set
      - Add vertex to set
- Remove all vertices in independent set at once, in any order
- Can use IndexedFaceSet as output DS

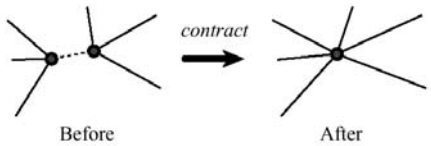
Edge Collapse

- Identify endpoints
- Determine vertex position
- Remove incident triangles
- Which edges to collapse ?
- In which order ?



Collapsing Unconnected Vertices

- Identify pair of vertices
- Determine vertex position
- Which pairs of vertices to collapse ?
- In which order ?



Edge Collapse Algorithm 1

- Put all "collapsible" edges in priority queue according to "removal error"
- While queue is not empty
  - Delete minimum edge from queue
  - Collapse edge
    - Identify vertices
    - Delete incident edges
  - Remove all incident edges from the queue,
  - Determine if "collapsible", recompute "removal error", re-insert in queue
- Need dynamic data structures

Edge Collapse Algorithm 2

- Put all "collapsible" edges in priority queue according to "removal error"
- Choose an "independent set" of edges
  - Initialze set to empty
  - While queue is not empty
    - Delete minimum edge from queue
    - If edge is independent of all edges in set
      - Add edge to set
- Remove all edges in independent set at once, in any order
- Can use IndexedFaceSet as output DS

Garland-Heckbert

- Each vertex has a 4x4 Symmetric matrix Q which initially measures the square distance from a point to the normal plane
- When two vertices are merged the matrices are added
- New matrix measures the sum of the square distances to N planes

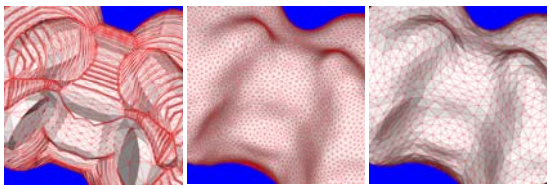
Remeshing

- Adaptive resampling
- Semi-regular connectivity
  - Subdivision surfaces
- Inverse of Simplification ?
- Retiling [Turk 92]

Dynamic connectivity

- Kobbelt et. al. Eurographics 2000
  - 2 Emin < Emax
  - Collapse all edges shorter than Emin
  - Split all edges longer than Emax
  - Flip edges to equalize vertex valence
  - Smooth
  - Iterate until all edges have length between bounds
  - Details ?

Dynamic connectivity



Boundaries and Ridges

- How to detect ?
- How to handle
  - in simplification algorithm ?
  - in optimization algorithm ?
  - In remeshing algorithm ?
  - In smoothing algorithm ?

Adaptive Triangle Subdivision

- Mark vertices of edges that need to be refined
- Create a new vertex at the midpoint of each edge with 2 marked vertices
- Quadrisect each triangle with 3 marked vertices
- Bisect each triangle with 2 marked vertices
- Keep other triangles