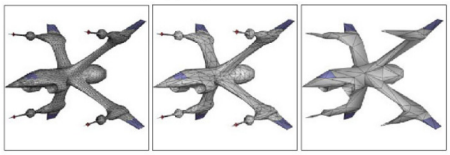
Introduction

- Context of Real-time Rendering
  - A large model to display
  - A fixed amount of time (30 fps)
  - A fixed resolution
- Strategies
  - Display only what is visible
  - Display objects with appropriate details
  - Use simplified version (also for side computations)
    - collision detection
    - shadows
- Optimize rendering

1

Level of Detail: Principle

- For each object in the scene
  - Store different versions
  - Choose appropriate version
- How to build the simplified versions?
  - How many? Which degree of simplification?
- How to choose appropriate version?



2

Overview of presentation

Mesh Simplification

Error Metrics

Selection of LOD

An example: QSlim

Extensions

3

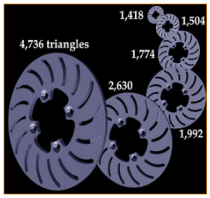
Simplification Framework

- Topology preservation
- Local operators
  - work on vertices, edges, faces
  - locally decrease polygon count
- Global operators
  - treat the object as a whole
  - more like "resampling"

4

Topology preservation

- Maintain Genus
- Can not merge parts
- Limit the simplification
- Not always required



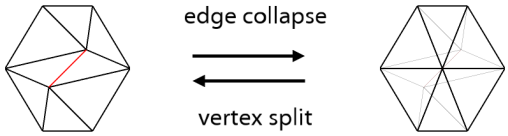
5

Local operators

- Edge Collapse
- Vertex pair collapse
- Vertex removal
- Cell collapse

6

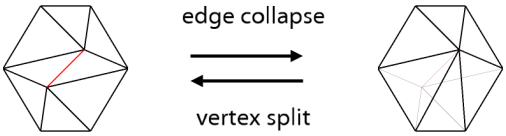
Edge Collapse



Where to place the new vertex?

7

Edge Collapse

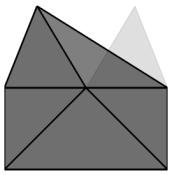


- On the edge
  - leaves freedom (but where?)
  - introduces a new point
- On one extremity
  - lower quality
  - no new point
  - ***half-edge collapse***

8

Vertex Pair Collapse

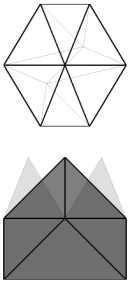
- Generalize edge collapse
  - Collapse any pair of vertices
  - Must choose new position
- Can not consider all pairs
  - avoids quadratic complexity
  - selects "close" vertices
  - achieves linear complexity in practice



9

Collapse Considerations

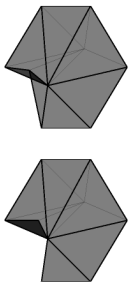
- Post collapse operations
  - Removal of degenerated faces
  - Update of adjacency relations



10

Collapse Considerations

- Post collapse operations
  - Removal of degenerated faces
  - Update of adjacency relations
- Pre collapse operation
  - Avoid mesh fold-over
  - Avoid topology inconstitency



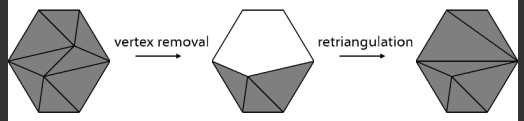
11

Local operators

- Edge Collapse
- Vertex pair collapse
- Vertex removal
- Cell collapse

12

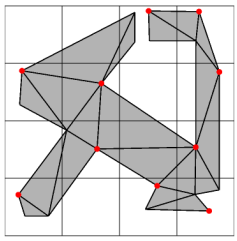
Vertex Removal



- "Filling the hole" can be tricky
  - Polygon is not "planar"
- Many triangulation
  - Encompass half edge collapse
- Generalized by "polygon merging"

13

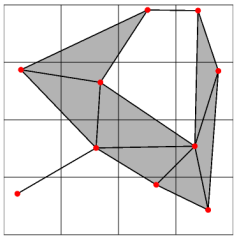
Cell collapse



- Place the model in a grid
- Choose a representant per cell
  - one of the point
  - center of the cell, barycenter,..

14

Cell collapse



- Place the model in a grid
- Choose a representant per cell
  - one of the point
  - center of the cell, barycenter,..
- Collapse cell's points on representant
- Clean degeneracies
  - remove empty triangles
  - create segments and points

15

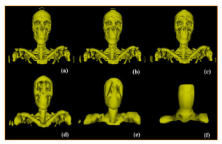
Simplification Framework

- Topology preservation
- Local operators
  - work on vertices, edges, faces
  - locally decrease polygon count
- Global operators
  - treat the object as a whole
  - more like "resampling"

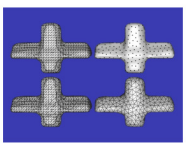
16

Global operators

- Volume Rendering [He95]
  - Rasterize the mesh in a grid
  - Low-pass filter the grid
  - Reconstruct the mesh (marching cube)



- Re-tiling polygonal surfaces [Turk92]
  - Distribute point on the surface
  - Re-triangulate the points



17

Morphological operators

- "Erase" small details
- Merge disconnected parts
- Cannot simplify past some point
- Generalized with alpha hulls [ElSana98]

18

Overview of presentation

Mesh Simplification
Error Metrics
Selection of LOD
An example: QSlim
Extensions

19

Error Metrics

- Why measure Error?
  - Guide the simplification process
  - Know the quality of the results
  - Know when to show a particular LOD
  - Balance quality over a scene
- Key elements
  - Geometric Error
  - Attribute Error
- Incremental vs. Total Error

20

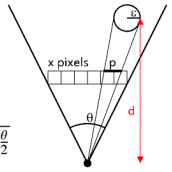
Error Metrics - geometric error

- How to measure the distance between two surfaces?
  - Hausdorff distance
  - Approximations

$$h(A,B) = \max_{a \in A} \min_{b \in B} \|a - b\|$$

$$H(A,B) = \max(h(A,B), h(B,A))$$

- Maximum vs. Average
- Screen Space Error

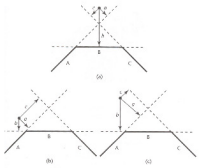


$$p = \frac{\varepsilon\, x}{2d \tan\frac{\theta}{2}}$$

21

Measuring distance: approaches

- Vertex-Vertex
  - natural for cell collapses
  - miss surface changes
- Vertex-Plane
  - distance to supporting faces
  - innacurate in theory
- Vertex-Surface
  - distance to closest point
  - Progressive Meshes [Hoppe93]
- Surface-Surface
  - strongest error bound
  - hard to compute
  - Simplification Enveloppes [Cohen96]



22

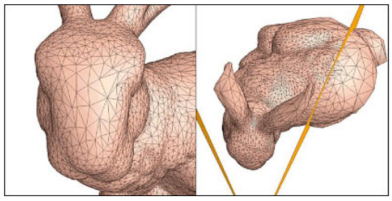
Overview of presentation

Mesh Simplification
Error Metrics
Selection of LOD
An example: QSlim
Extensions

23

LOD Selection

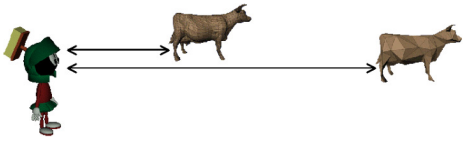
- LOD type
  - discrete, continuous, view-dependent



24

LOD Selection

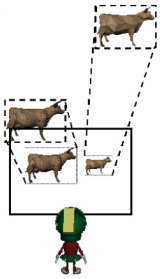
- LOD type
  - discrete, continuous, view-dependent
- Selection criteria
  - distance
  - screen size



25

LOD Selection

- LOD type
  - discrete, continuous, view-dependent
- Selection criteria
  - distance
  - screen size



26

LOD Selection

- LOD type
  - discrete, continuous, view-dependent
- Selection criteria
  - distance
  - screen size
  - priority, hysteresis, environmental conditions, perceptual factors
- Blending Between Transitions
  - alpha blending
  - geomorphs

27

Overview of presentation

Mesh Simplification

Error Metrics

Selection of LOD

An example: QSlim

Extensions

28

QSlim algorithm

- For each vertex compute initial quadric
- Build all pairs of "close" vertices
- For each pair:
  - find best position for collapsing
    - if possible, inverse matrix Q+Q'
    - otherwise, test extremities and middle point
  - compute associated error
- Choose collapsing with smallest error
  - compute new quadric for collapsed vertex
  - update list of pairs
- Iterate until:
  - given number of faces reached
  - given error reached

29

Overview of presentation

Mesh Simplification

Error Metrics

Selection of LOD

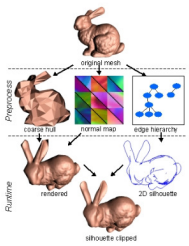
An example: QSlim

Extensions

30

Silhouette Clipping [Sander00]

- Silouhette is visually important
- Mesh simplification degrades silouhette



31

Alternatives representations

- Hard to represent a shape with few polygons
- An image can convey many details!
- Use Image Based Representations (IBR)!
- Imposters [Maciel95,Decoret99]
  - replace distant parts with a textured quad
  - looks nice when static
  - looks flat when moving
- Billboard Clouds [Decoret03]
  - approximate roughly shape with few planes
  - use semi-transparent textures for finer details
  - generalize classic tricks for games (trees)

32