Approaches to Polygonal Model Simplification

Geometric Replacement

Replace complex objects with simpler objects

Reduces transformation and communication time

Rasterization time not changed

- replacements should cover similar number of pixels

Ford Bronco Model

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

courtesy of Division and Viewpoint

Automatic Simplification

Preprocess

- **create multiresolution representation**

Run-time

- **extract appropriate resolution model based on viewing parameters and rendering load**

Important Questions

1. What are the input restrictions?
2. How much is primitive count reduced?
3. How fast are primitives rendered?
4. How good can the results look?
5. How much space is used?
6. How much pre-processing time?

What I'll Talk About

Performing Simplification

- Simplification Operations
- Error Measures

Employing Simplification

- Static and Dynamic Representations
- Run-time Control

Simplification Operations

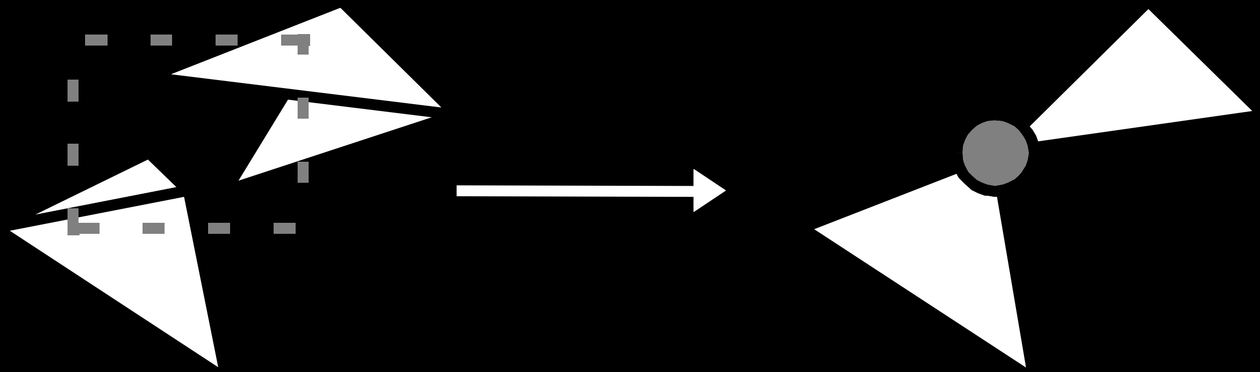
Types of operations

- Vertex cluster
- Vertex remove
- Edge collapse
- Vertex pair

Each operation reduces model complexity by small amount

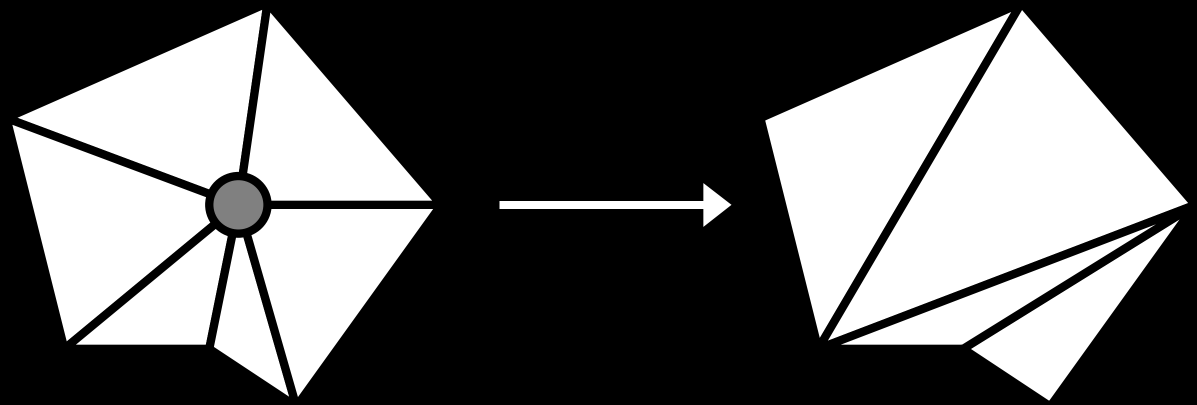
Apply many operations in succession to achieve large reductions

Vertex Cluster



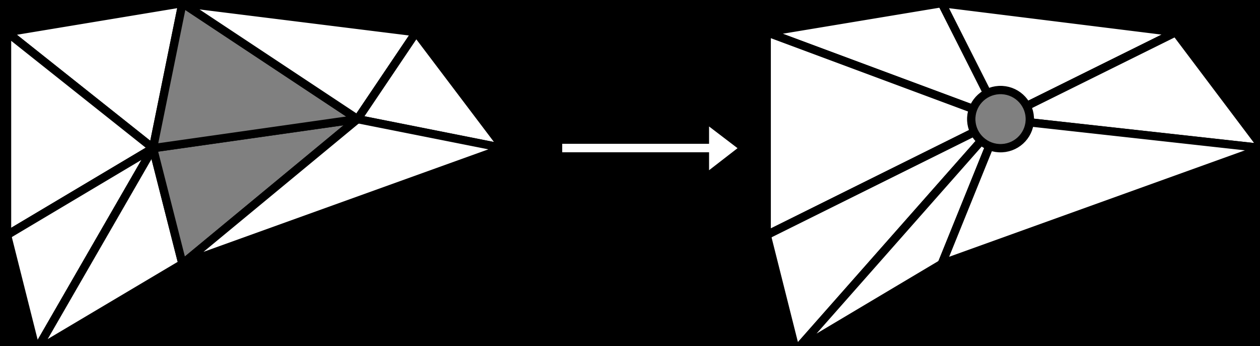
- Merge vertices based on geometric proximity
- Triangles with repeated vertices degenerate to edge or point
- General and robust
- Not usually attractive

Vertex Remove



- Remove vertex and adjacent faces
- Fill hole with new triangles (reduction of 2)
  - many possible triangulations (exponential)
- Requires *manifold* surface around vertex
- Preserves local topological structure
  - typically more attractive

Edge Collapse

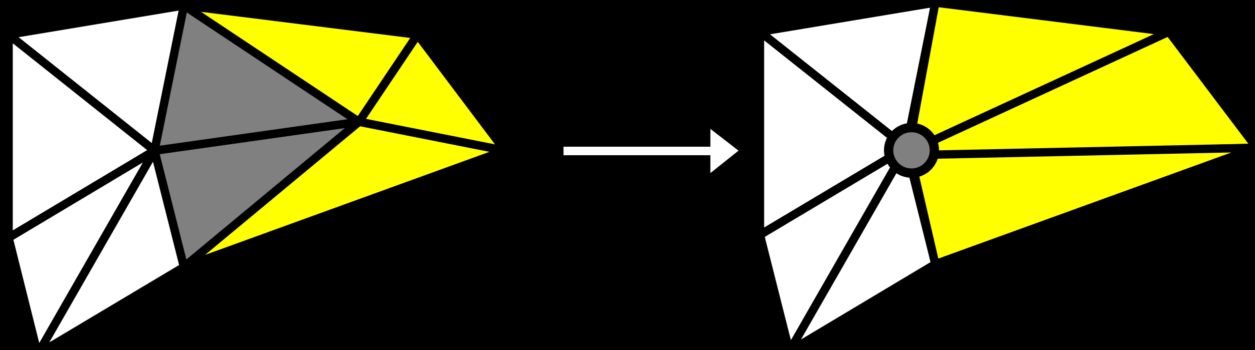


Merge two edge vertices to one
- choose position and attributes for vertex

Delete degenerate triangles
- those containing both vertices (entire edge)
- 2 triangles for manifold edge

Smooth transitions
- animate edge collapse(s) over time

Half-Edge Collapse



Collapse to endpoint of edge

- Vertex subset property of vertex remove
- Smooth transition property of edge collapse

Vertex Remove vs. Edge Collapse

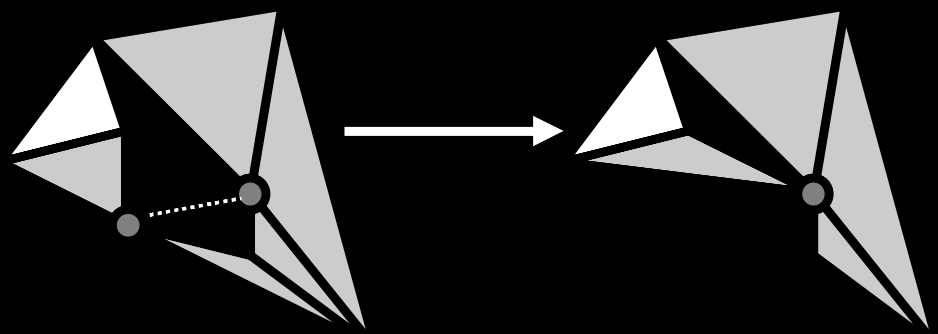
Vertex Remove

- Affects neighborhood around 1 vertex
- Does not introduce new vertices
- Many possible tesselations
- Smooth transitions difficult

Edge Collapse

- Affects neighborhood around 2 vertices
- Freely positions one new vertex
- Single possible tesselation
- Smooth transitions natural

Vertex Pair



Merge any two vertices

- based on geometry, topology, etc.

More flexibility than edge collapse

More local control than vertex cluster

Operation Considerations

- Attention to topology promotes better appearance
- Allowing non-manifolds increases robustness and ability to simplify
- Collapse-type operations allow smooth transitions
- Vertex remove affects smaller portion of mesh than edge collapse
- Subset of vertex remove equivalent to subset of edge collapse

Performing Simplification

Measure cost of possible operations according to error measure

- Crucial to simplification quality

Place operations in queue according to error

Perform operations in queue

- After each operation, re-evaluate error of operations in neighborhood

What I'll Talk About

Performing Simplification

- Simplification Operations
- Error Measures

Employing Simplification

- Static and Dynamic Representations
- Run-time Control

Why Measure Error?

Guide simplification process

- Making better choices produces better simplifications

Know quality of results

- Object-space error bounds describes quality

Know when to show a particular LOD

- Which LOD for a given screen-space error

Balance quality for large environments

- What error bound for a given polygon count

Geometric Error Measures

Promote accurate 3D shape preservation

Also preserves screen-space shape

- **Silhouettes**
- **Pixel coverage**

Classifying Geometric Error Metrics

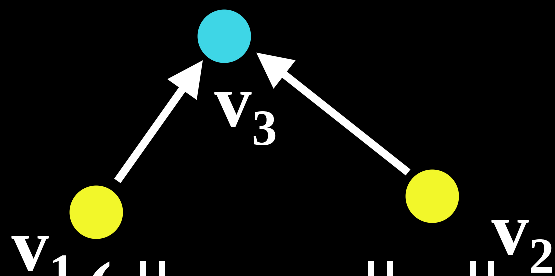
Vertex-Vertex Distance

Vertex-Plane Distance

Point-Surface Distance

Surface-Surface Distance

Vertex-Vertex Distance



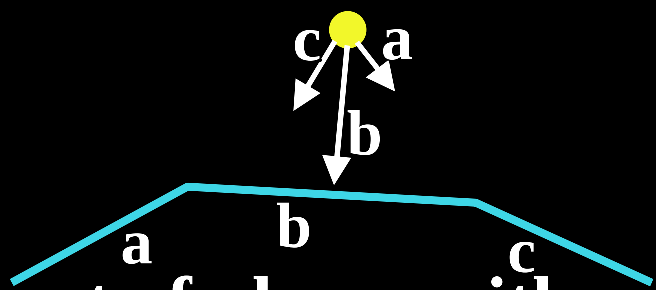
$$E = \max(\| v_3 - v_1 \|, \| v_3 - v_2 \|)$$

Appropriate during topology changes

- Rossignac and Borrel 93
- Luebke and Erikson 97

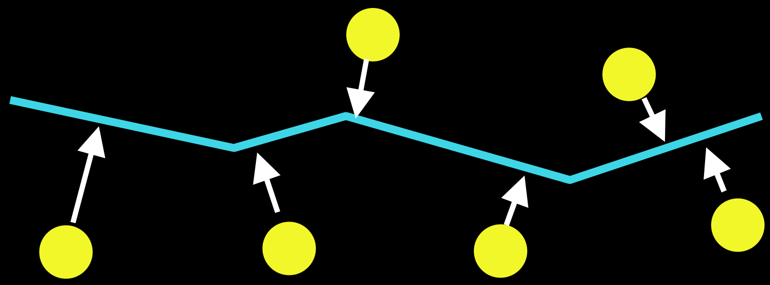
Loose for topology-preserving collapses

Vertex-Plane Distance



- Store set of planes with each vertex
  - Error based on distance from vertex to planes
  - When vertices are merged, merge sets
- Ronfard and Rossignac 96
  - Store plane sets, compute max distance
- *Error Quadrics* - Garland and Heckbert 96
  - Store quadratic form, compute sum of square distances

Point-Surface Distance

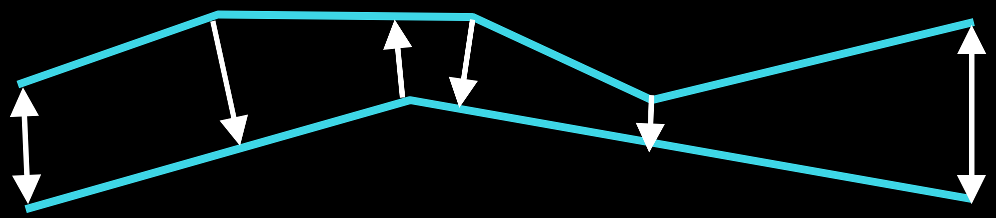


Used in Hoppe 93 and 96

Map point set to closest points on simplified surface

Compute sum of square distances

Surface-Surface Distance



Bound maximum distance between input and simplified surfaces

- Tolerance Volumes - Guéziec 96
- Simplification Envelopes - Cohen/Varshney 96
- Hausdorf Distance - Klein 96
- Mapping Distance - Bajaj/Schikore 96, Cohen et al. 97

Vertex-Vertex != Surface-Surface

Error is zero at vertices and exterior edges

Error is non-zero everywhere else

- **not captured by vertex-vertex or vertex-plane metrics**

Memoryless Simplification

Lindstrom/Turk 98

No measure of error from original mesh

- *Incremental* rather than total error

Preserve volume and area as simplification progresses

Error demonstrated to be low using after-the-fact error measurement

- Metro - Cignoni et al. 96

Geometric Error Observations

Vertex-vertex and vertex-plane distance

- Fast
- Low error shown after-the-fact, but not guaranteed by metric

Cannot guarantee quality without surface-surface distance bound

Hoppe's point-surface approximates one-sided surface-surface

Good error measures useful at run-time

- 3D average or maximum error distance

Attribute Error Metrics

Attributes include colors, normals, and texture coordinates

Promote accuracy of final *pixel colors*

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

Vertex-Vertex Distance

GAPS point clouds - Erikson/Manocha 98

- Measure sum of square distances from vertex to its constituent vertices (area-weighted)
- Used for colors, normals, and texture coordinates
- Stored as 5 floats for 3D attributes (e.g. rgb)

Normal cones

- Luebke/Erikson 97, Xia et al. 97

Vertex-Plane Distance

- Higher-dimensional error quadrics
  - Garland and Heckbert 98
  - Vertices live in higher-dimensional position + attribute space
  - Planes defined in this space
- Multiple attribute quadrics
  - Hoppe 99
  - Decouples affects of position and attributes
  - Reduces storage and computational complexity

Point-Surface Distance

- Extension of geometric point-surface distance
  - Hoppe 96
- Geometric correspondences found between original surface samples and simplified surface
- Sum of square attribute distances minimized
- Used primarily for vertex colors

Surface-Surface Distance

Bajaj / Schikore 96

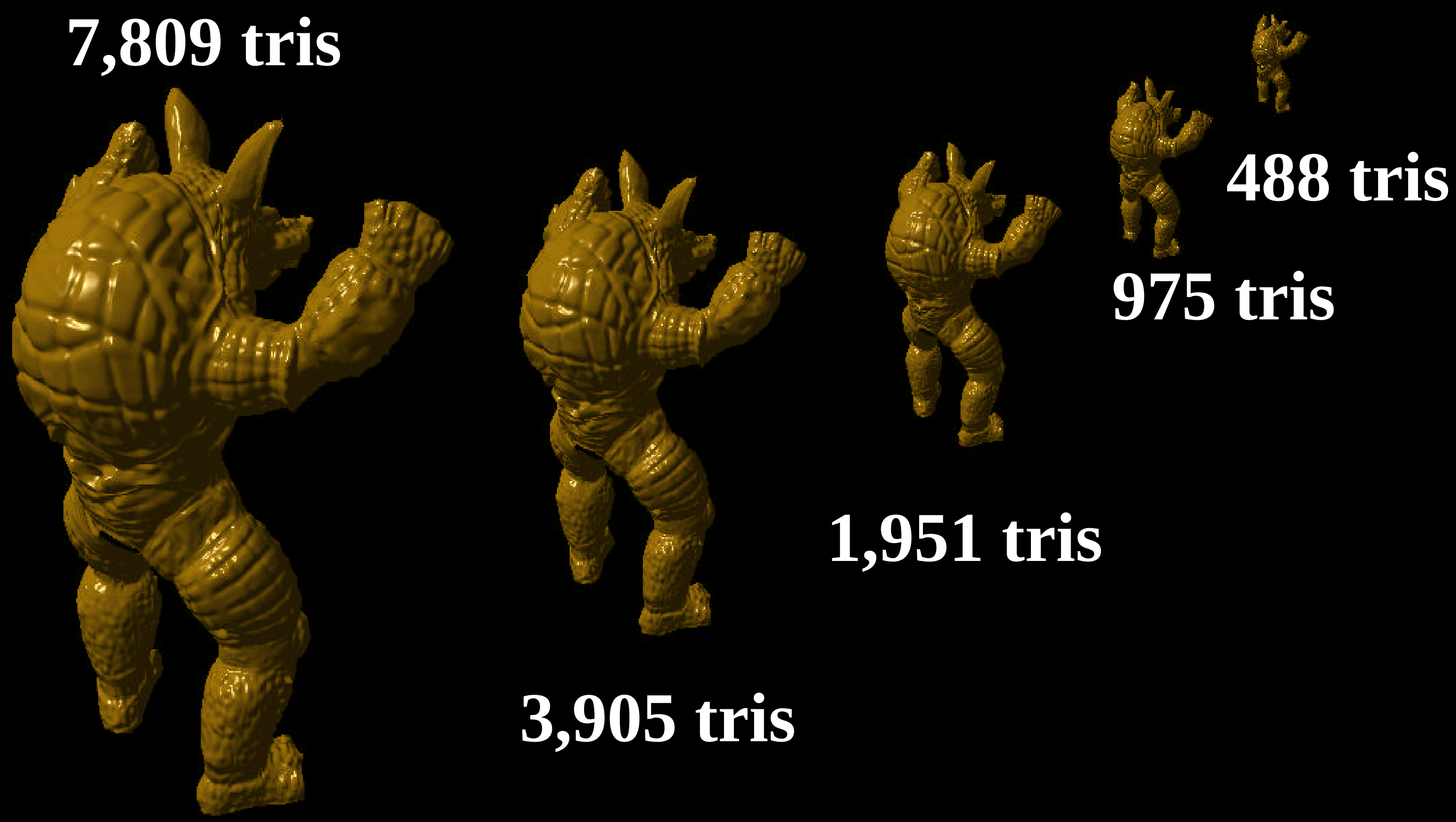
- Geometric projections provide local mappings
- Maximum distance of scalar attributes measured over surface

Appearance-Preserving Simplification

Cohen et al. 98

- **Colors and normals stored in texture and normal maps**
- ***Texture deviation* computed using parametric correspondence**
- **Preserves colors and normals, bounding texture motion in object and screen space**

Appearance-Preserving Simplification



model courtesy of Stanford and Caltech

Attribute Metric Observations

Open problem

- Employ bounds on color and normal deviation at run-time
- Guarantee appearance, but simplify more as objects recede

Texture and normal map approach

- Requires parameterization, texture management, and improved shading hardware

What I'll Talk About

Performing Simplification

- Simplification Operations
- Error Measures

Employing Simplification

- Static and Dynamic Representations
- Run-time Control

Multi-resolution Representations

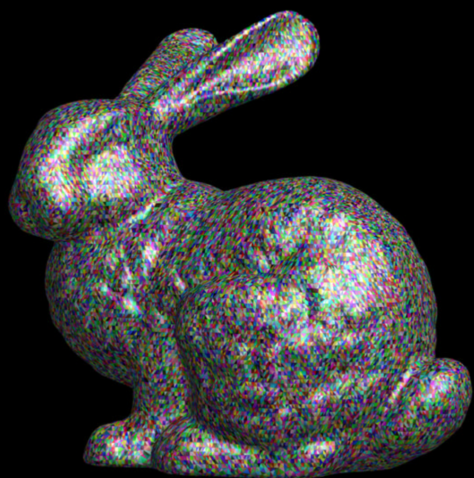
Static levels of detail

- Individual, complete simplified meshes

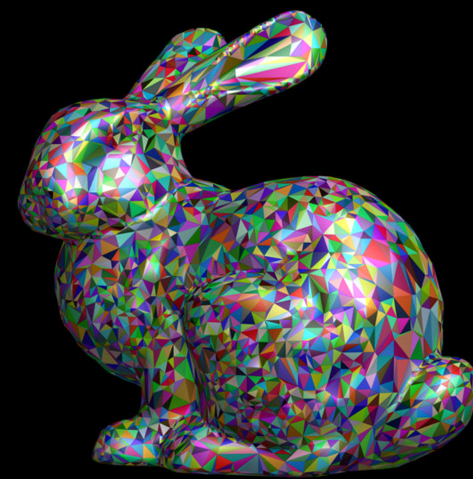
Dynamic representations

- Tree structure traversed & adapted at run-time
- DeFloriani et al. 97, Hoppe 97, Luebke/Erikson 97, Xia et al. 97

Static Levels of Detail

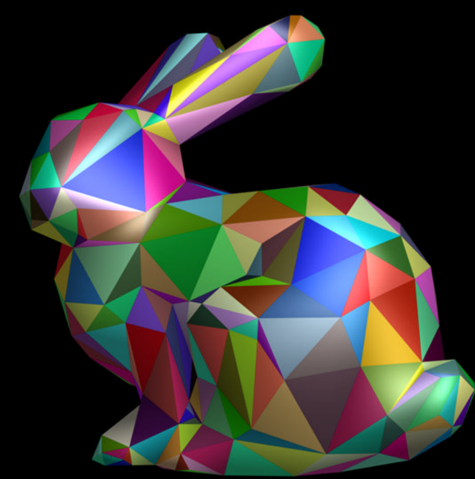


69 k tris



11 k tris

2 k tris



575 tris

Static Levels of Detail

Pre-process

- Generate set of independent levels of detail

Run-time

- Select level of detail based on distance from viewpoint

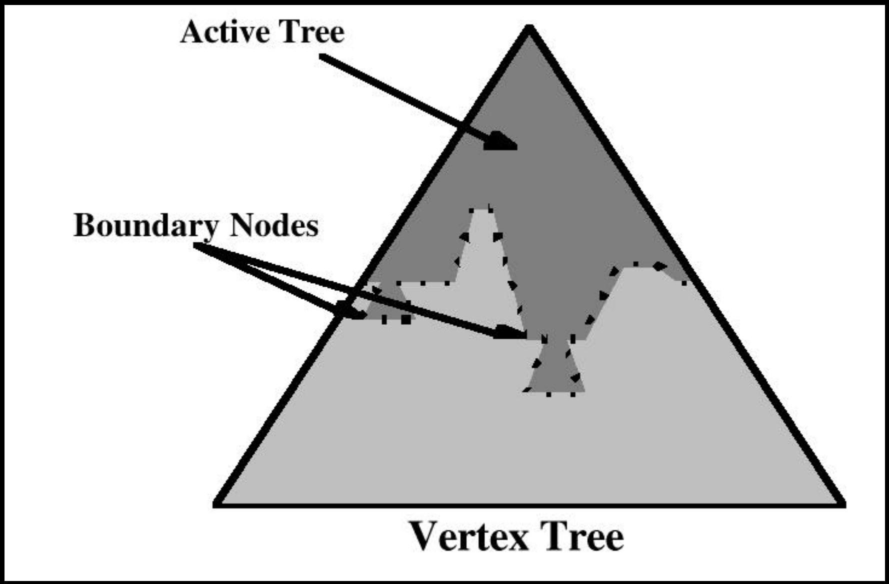
Advantages

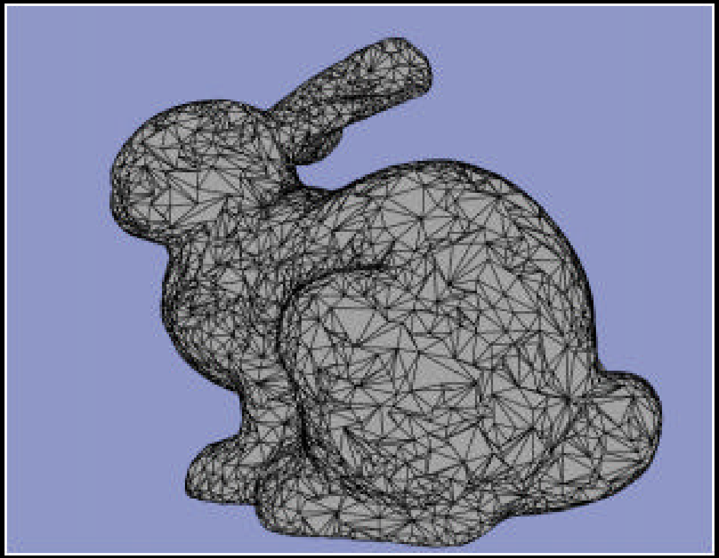
- Fairly efficient storage (2x original)
- No significant run-time overhead

Disadvantages

- Requires per-object simplification
- Not good for spatially large objects

Dynamic Levels of Detail





from Luebke and Erikson, **SIGGRAPH 97**

Dynamic Levels of Detail

Pre-process

- Generate tree of simplification operations

Run-time

- Refine/coarsen current model according to viewpoint

Advantages

- Allows finer control of tessellation

Disadvantages

- More run-time computation and complexity
- Difficult for retained-mode graphics

What I'll Talk About

Performing Simplification

- Simplification Operations
- Error Measures

Employing Simplification

- Static and Dynamic Representations
- Run-time Control

Run-time Goals

A) Guarantee frame rate, maximize quality

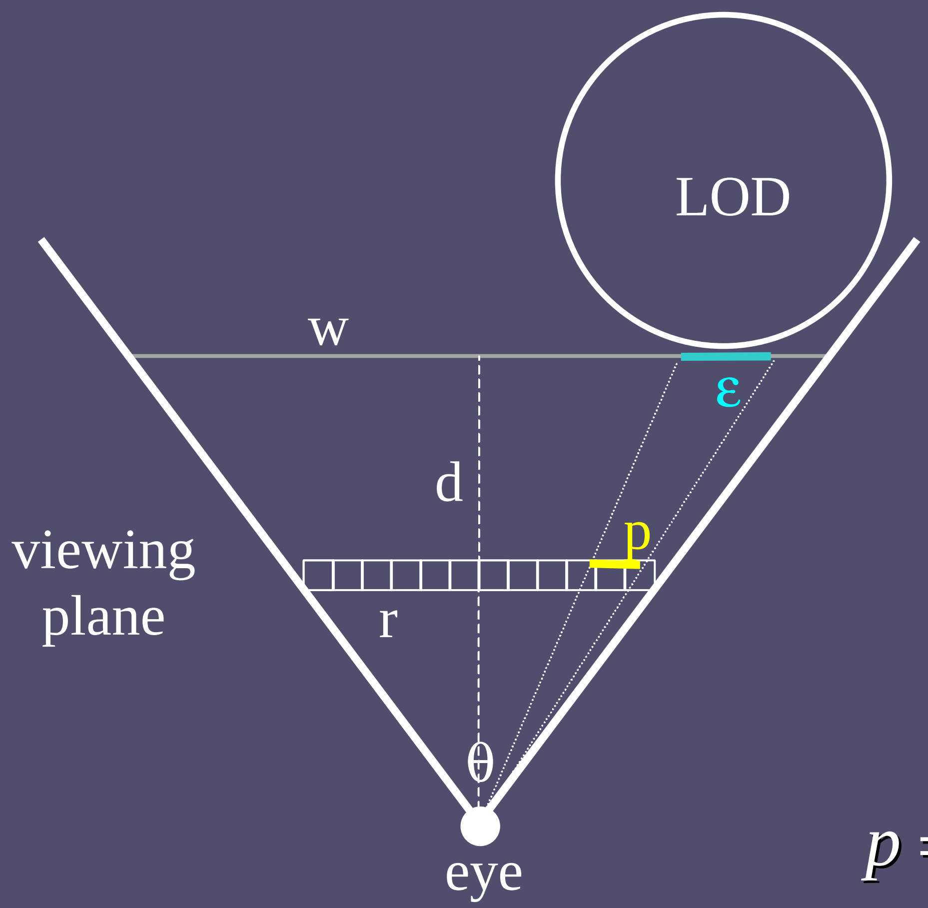
- For multiple objects, knapsack problem (Funkhouser/Sequin 93)
- Tricky for dynamic simplification (Luebke/Erikson 97, Hoppe 97)

or

B) Guarantee quality, maximize frame rate

- Easier (if good quality metric available)
- Each object or refinement may be considered independently

Screen-space Geometric Error



$$p = \frac{e\,r}{w} = \frac{e\,r}{2d\,\tan\left(q/2\right)}$$

Screen-space Attribute Error

- Texture coordinates work like geometric error
  - Cohen et al. 98
- Normal error controls dynamic refinement around highlights
  - Xia et al. 97, Klein 98
  - Doesn’t allow more simplification as objects recede
- Color control??

Questions Revisited

1. What are the input restrictions
   - Vertex cluster and vertex pair allow general triangle input
   - Vertex remove and edge collapse usually apply to manifold meshes
2. How much is primitive count reduced?
   - Topology modifying algorithms can often reduce more for complex environments
   - Dynamic simplification can reduce more than static for a given error bound

Questions Revisited (2)

3. How fast are primitives rendered
   - Static LODs are processed more efficiently
   - Normal-mapped primitives require constant additional shading time (currently BIG)
4. How good can the results look
   - Topology-preserving techniques usually produce prettier results than object merging
   - Attention to attributes as well as geometry important for preserving appearance

Questions Revisited (3)

5. How much space is used?
   - Dynamic simplification requires more space at run-time than static

6. How much pre-processing time?
   - Vertex-vertex and vertex-plane metrics generally faster than point-surface and surface-surface metrics

Video: Appearance-Preserving Simplification

References

- Bajaj, Chandrajit and Daniel Schikore. Error-bounded Reduction of Triangle Meshes with Multivariate Data. *SPIE*. vol. 2656. 1996. pp. 34-45.
- Cohen, Jonathan, Dinesh Manocha, and Marc Olano. Simplifying Polygonal Models using Successive Mappings. Proceedings of IEEE Visualization '97. pp. 395-402.
- Cohen, Jonathan, Marc Olano, and Dinesh Manocha. Appearance-Preserving Simplification. Proceedings of ACM SIGGRAPH 98. pp. 115-122.
- Cohen, Jonathan, Amitabh Varshney, Dinesh Manocha, Gregory Turk, Hans Weber, Pankaj Agarwal, Frederick Brooks, and William Wright. Simplification Envelopes. Proceedings of SIGGRAPH 96. pp. 119-128.
- DeFloriani, Leila, Paola Magillo, and Enrico Puppo. Building and Traversing a Surface at Variable Resolution. Proceedings of IEEE Visualization ‘97. pp. 103-110.
- Erikson, Carl and Dinesh Manocha. GAPS: General and Automatic Polygonal Simplification. Proceedings of 1999 ACM Symposium on Interactive 3D Graphics. pp. 79-88.

References

- Garland, Michael and Paul Heckbert. Simplifying Surfaces with Color and Texture using Quadric Error Metrics. Proceedings of IEEE Visualization '98. pp. 263-269, 542.
- Garland, Michael and Paul Heckbert. Surface Simplification using Quadric Error Bounds. Proceedings of SIGGRAPH 97. pp. 209-216.
- Guéziec, André. Surface Simplification with Variable Tolerance. Proceedings of Second Annual International Symposium on Medical Robotics and Computer Assisted Surgery (MRCAS '95). pp. 132-139.
- Hoppe, Hugues. Progressive Meshes. Proceedings of SIGGRAPH 96. pp. 99-108.
- Hoppe, Hugues. View-Dependent Refinement of Progressive Meshes. Proceedings of SIGGRAPH 97. pp. 189-198.
- Hoppe, Hugues, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Mesh Optimization. Proceedings of SIGGRAPH 93. pp. 19-26.
- Hoppe, Hugues H. New Quadric Metric for Simplifying Meshes with Appearance Attributes. Proceedings of IEEE Visualization '99. pp. 59-66.

References

- Klein, Reinhard, Gunther Liebich, and Wolfgang Straßer. Mesh Reduction with Error Control. Proceedings of IEEE Visualization '96.
- Lindstrom, Peter and Greg Turk. Fast and Memory Efficient Polygonal Simplification. Proceedings of IEEE Visualization '98. pp. 279-286, 544.
- Luebke, David and Carl Erikson. View-Dependent Simplification of Arbitrary Polygonal Environments. Proceedings of SIGGRAPH 97. pp. 199-208.
- Ronfard, Remi and Jarek Rossignac. Full-range Approximation of Triangulated Polyhedra. Computer Graphics Forum. vol. 15(3). 1996. pp. 67-76 and 462.
- Rossignac, Jarek and Paul Borrel. Multi-Resolution 3D Approximations for Rendering. Modeling in Computer Graphics. Springer-Verlag1993. pp. 455-465.
- Xia, Julie C., Jihad El-Sana, and Amitabh Varshney. Adaptive Real-Time Level-of-Detail-Based Rendering for Polygonal Models. IEEE Transactions on Visualization and Computer Graphics. vol. 3(2). 1997. pp. 171-183.