Level of Detail in Computer Graphics

Lakulish Antani

April 23, 2006

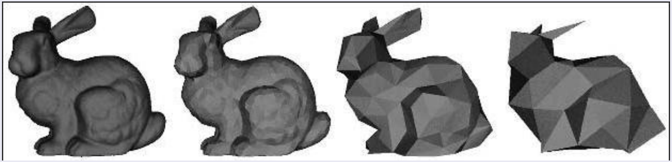
Outline

The Basic Idea

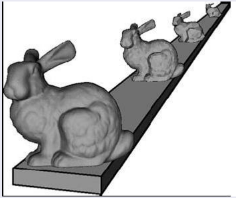
- We want to render increasingly detailed objects during interactive visualization.
- Object detail grows faster than hardware speed.
- We need a way to scale the detail of rendered objects according to visual importance.
- Techniques to do this are called Level of Detail techniques.

An Example

Multiple LODs of a Bunny



When Rendered



LOD Hierarchies

- LOD systems build a hierarchical data structure.
- Deeper levels in the hierarchy represent the object at finer detail.

Classification of LOD Hierarchies

- Discrete
- Continuous
- View-Dependent

Types of Models

- Models used in interactive rendering could be of any form:
  - triangle meshes
  - point sets
  - images
- We concentrate on LOD techniques for triangle meshes.

Mesh Basics

- Triangles are the simplest 2-D primitive to render.
- Informally, a triangle mesh is a set of triangles which share some common edges and common vertices.
- A triangle is a 2-D simplex. Triangle meshes are simplicial meshes.
- The topology and geometry of a triangle mesh can be separated into a tuple $M = (K, V)$.
  - K is a simplicial complex.
  - V is a set of vertex positions.

Overview of Mesh Simplification

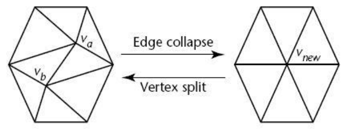
- Two components of mesh LOD systems:
  - offline mesh simplification
  - run-time LOD management
- Offline simplification involves iteratively choosing from a set of simplification operators.
- Uses error metrics to guide the selection.

Simplification Operators

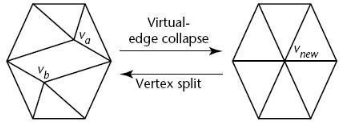
- A simplification operator, when applied to a mesh, produces a simplified version of the mesh.
- Changes made by an operator may be local or global.
- An operator may or may not be genus-preserving.

Examples of Simplification Operators

Edge Collapse

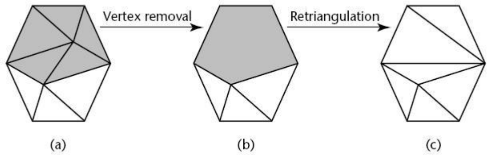


Vertex Pair Collapse



Examples of Simplification Operators

Vertex Removal



Cell Collapse

This operator collapses all vertices within some volume to a single vertex.

Error Metrics

- Measure the "deviation" between a mesh and a simplified version of it.
- Used for two purposes:
  - guiding the simplification process
  - selecting LODs to use at run-time
- Two kinds of error:
  - geometric
  - attribute (eg. texture, colour, normals)

Examples of Error Metrics

- If using cell collapse, cell size is used as an error metric.
- Maximum distance from supporting planes can be used as a measure of geometric error.
- Image-based metrics evaluate the difference between rendered images of two meshes.
- Perceptual metrics evaluate how perceptible a change to the mesh is.

View-Independent LOD Management

- At run-time, we need a criteria for selecting the LOD at which to render an object.
- A view-independent criterion uniformly refines/simplifies the entire object.
- Examples of simple view-independent criteria:
  - distance to viewpoint
  - screen-space size
  - semantic importance

Fixed Frame Rate Scheduling

- A method to give a smooth user experience by keeping a fixed frame rate.
- Uses a deadline, or equivalently, a minimum frame rate.
- Reactive scheduling modifies detail based on whether or not the previous frame finished within the deadline.
- Predictive scheduling tries to render at the maximum detail possible without exceeding the deadline.

View-Dependent LOD Management

- The LOD hierarchy used is view-dependent (often a tree or DAG).
- Allows selective refinement of only some portions of a mesh.
- A view-dependent criteria decides which portions of the hierarchy to refine/simplify.
- This way, closer portions of a large mesh can be shown in more detail than farther portions.

Progressive Meshes

- Progressive Meshes are a popular framework for continuous LOD.
- The representation naturally supports progressive transmission.
- PMs also support smooth transitions (geomorphs) between LODs.

The Progressive Mesh Representation

- PM simplification uses the edge collapse operator.
- PMs represented as a base mesh M_0 and a sequence $\{vs_i\}$ of vertex splits.
- Such a sequential data structure naturally supports progressive transmission.

Simplification of Progressive Meshes

- Simplification algorithm must select from multiple edge collapse operations in each iteration.
- An energy function is used as the error metric.
- Simplification follows a greedy strategy: the edge collapse leading to the mesh with lowest energy is picked.

Progressive Mesh Error Metric

Definition

$$E(M) = E_{dist}(M) + E_{spring}(M) + E_{scalar}(M) + E_{disc}(M)$$

The Individual Terms

- E_{dist} measures the distance of the mesh from a point set sampled from the original mesh.
- E_{spring} is a spring energy term, assuming edges to be springs.
- E_{scalar} measures scalar attribute error, similar to E_{dist}.
- E_{disc} measures deviation of attribute discontinuity curves.

Geomorphs

- Two consecutive LODs differ only by one vertex split.
- In this case, linear interpolation between initial and final vertex positions can give smooth transitions.
- The topology in intermediate meshes is the same as that of the more detailed LOD.
- This method extends to geomorphs between any two LODs.

Vertex Hierarchy

- A forest of binary trees whose nodes represent vertices.
- Each interior node has two children, these are the vertices resulting from a vertex split.
- A cut across the forest is maintained, representing the set of vertices used to render the object currently.
- View-dependent refinement criteria move portions of the cut up or down the hierarchy.

Selective Refinement

- Viewing parameters are used to adjust the cut across the vertex hierarchy.
- Portions of the mesh may be coarsened if:
  - they lie outside the view frustum
  - they are oriented away from the viewer
  - their screen-space error is within a given tolerance
- Otherwise, the portion of the mesh is refined.

References

Cohen, J., and Manocha, D.
Model Simplification.
Elsevier, 2005.

DeFloriani, L., and Magillo, P.
Multiresolution Mesh Representation: Models and Data Structures.
Springer Verlag, 2002.

Garland, M., and Heckbert, P.
Surface simplification using quadric error metrics.
In *SIGGRAPH '97 Proceedings* (1997), pp. 209–216.

Hoppe, H.
Progressive meshes.
In *SIGGRAPH '96 Proceedings* (1996), pp. 99–108.

Hoppe, H.
View-dependent refinement of progressive meshes.
In *SIGGRAPH '97 Proceedings* (1997), pp. 189–198.

Luebke, D., Reddy, M., Cohen, J., Varshney, A., Watson, B., and Huebner, R.
Level of Detail for 3D Graphics, 1st ed.
Morgan Kaufmann Publishers, 2003.

Thank You