CSI 4341, Computer Graphics

Lecture 6: 3D Geometry

Date: 2012-09-17
Author(s): Edward Collier, Sam Wilson
Lecturer: Fredrik Niemelä

Lecture 6 reviewed assignment 3 and the materials required to do the program.

1 Representing Meshes

Mesh is connected set of polygons.
Each vertex in a mesh has a location.

2 Simple Representation

Define each polygon by the geometric locations of its vertices.

3 Inward and Outward Facing Polygons

4 Geometry vs. Topology

Generally it is a good idea to look for data structures that seperate from the topology.
Geometry: locations of the vertices.
Topology: organization of the vertices and edges.
Example: a polygon is an ordered list of vertices with an edge connecting successive pairs of vertices and the last of the first.
Topology holds even if geometry changes.

5 Vertex Lists

Put the geometry in an array.
use the pointers from the vertices into the array.

6 Shared Edges

Vertex lists will draw filled polygons correctly but if we draw the polygon by its edges, shared edges are drawn twice.
Store mesh by edge list.

7 Edge List

Needs Geometry, Edges, Vertices

8 Modeling a Cube

Define global arrays for vertices and colors

9 Vertex Arrays

OpenGL provided a facility called vertex arrays that allows us to store array data in the implementation
Six types of arrays were supported initially:
- Verticies
- Colors
- Color Indices
- Normals
- Texture Coordinates
- Edges Flags

Now Vertex arrays can be used for any attribute

10 Mapping Indices to Faces

Form an array of face indices.
Each successive four indices describe a face of the cube.