CS559 Lecture 26 – Shading and Surfaces

Lecture notes not for display

October, 2007

Lighting Recap

- Compute lighting at a point
- Given
  - Position (used for eye vector, light vector)
  - Normal
  - Surface Properties (Cs Cd Ca s)
  - Light Properties(position, intensity, color, shape)
  - Eye position

$$color = A * C_A + \sum_{i \in lights} \left(I_i * \left(C_D \cdot (\hat{n} \bullet \hat{l}) + C_S (\hat{N} \cdot \hat{h})^s \right) \right)$$

What are we lighting?

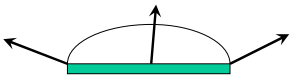
- Surfaces (discuss in detail later)
- Triangles (what we really end up drawing)
- Smooth objects (that we approximate with triangles)

When do we do lighting

- Per-Triangle
  - Flat shading
  - If things really were triangles, would be right
  - Faceted appearance – CG = quaint
- Per Vertex
  - Allow each vertex to have a different normal
    - Fake a smooth object
  - Interpolate colors inside of the triangle
  - Gouraud (smooth) shading
  - Small number of shading computations

When to shade (2)

- Per pixel
  - Interpolate NORMAL vectors (and other stuff)
  - PHONG shading (as opposed to Phong lighting)
- Why is this better than interpolating colors?
  - Catches real geometric changes
  - Shaper features
- Looks MUCH better



Modeling 3D Shapes

- Modeling = process of describing an object
  - Representation
- Can model shape, physical properties, behvaior, …
- Many uses of (geometric) models
  - Graphics – make a picture
  - CAD – represent for manufacture

Types of Shape Models in 3D

- Points
- Curves
- Surfaces and Solids
- Volumes

Surface vs. Volume

- Cube
  - Volume = space inside 0 <= x,y,z < 1
  - Surface = 6 squares (0,0,0)(0,0,1)(0,1,1)…
- Surface can be a boundary
  - But might not be
- Graphics (often) only need surfaces

When might we care about Volumes?

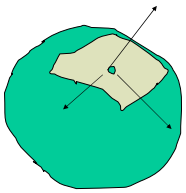
- Engineering / Manufacturing / Design
  - Can't be non-physical
- Some kinds of data has "insides"
  - Medical data (scanned)
- Some operations make sense
  - Constructive solid geometry
  - Cut / Join / Subtract / Union
    - Makes less sense on surfaces

How to do volumes?

- Hard: need to insure that you always have a volume!
- Operations on primitives
  - Make solid pieces (spheres, cylinders, polyhedra, …)
  - Combine with sensible operators (union, intersection, difference)
  - Construction Solid Geometry
- Boundary Representations
  - Store the surface
  - Represent what's inside
  - Be careful that there always is an inside – no holes!
- Implicit Representations
  - F(x) < 0 – for some fancy F
  - Distance fields, union of blobs, …
  - Tend to be special purpose
- Sampled Volumes (like medical data)

Surface Basics

- Locally flat
- At any point
  - Normal
  - Tangent Plane
  - Tangent vectors in plane



Surfaces

- Generally what we use in graphics
  - Hard enough!
- Similar issues to curves, but worse
- Named vs. Free-Form
- Build out of little pieces
- Linear pieces (polygons) – analogy to lines

Basic Strategy

- Break complicated surfaces into pieces
- Need to choose good pieces
- Need to make sure that the pieces connect
- Connections are more complicated