CS 563 Advanced Topics in Computer Graphics
Primitives and Acceleration

by Travis Grant
grant_travis@emc.com
PBRT Primitives

- geometric vs. volumetric primitives
  - geometric
    - shapes + materials
    - i.e. Spheres, Cylinders, Disks bound with texture properties
  - volumetric (participating media)
    - covered in Ch. 12 (not tonight)
    - particles distributed throughout a region of 3D space
    - i.e. atmospheric haze, beams of light through clouds...
PBRT Primitives (cont.)

Object Instancing
- geometry of a shape is referenced in order to reduce the memory requirements for representing many instances.
- a unique transformation is stored for each instance
- over 4000 individual plants (19.5 Million triangles total)
- only 61 unique plant models (1.1 M triangles stored in memory)
- consumes 300MB of memory during rendering
Aggregates

- Logical concatenation of multiple primitives
- A container that can hold many Primitives
- Basis for implementing acceleration structures

Scene::aggregate = single primitive that references to many other primitives (superset of scene) that is implemented with a class that stores the scene’s primitives in an acceleration data structure.
Primitives and Acceleration

- Acceleration
  - spatial subdivision
    - regionalized decomposition of 3D space
    - GridAccel & KDTreeAccel (Ch. 4)
  - object subdivision
    - progression of granularity through objects in scene
    - i.e. room analogy (four walls, ceiling, and a chair)
    - culled immediately if ray does not intersect room <-> or if a hit, subsequent testing of items in room...
GridAccel
- refinement and grid granularity
- refine immediately
- teapot in a stadium
- mailboxing
- bbox and overcounting voxels
- traversal
Primitives and Acceleration

Letters correspond to planes (A, B, C, D)
Point Location by recursive search
Primitives and Acceleration

- **KD-Tree Accelerator**
  - more adept at handling uneven distribution of primitives
  - traversal (depth first, front-to-back)
  - below, above & near, far (splitting plane)
References and acknowledgements

- [Apodaca] “Advanced RenderMan”
- [PHARR] “Physically Based Rendering”
- Pat Hanrahan’s Ray Tracing Presentation