Lecture 19: Thursday, 6 November 1997

Acceleration of Ray Casting / Ray Tracing

6.837 Announcements:
No class Tuesday [Vet's day].
HKN evals 18 (not 11) Nov.

Today: Accelerating Ray Casting/Ray Tracing

Design considerations:

Single frame or multiple frames?
If multiple: are viewpoints known a priori?

Scene static or dynamic?
If dynamic: are object paths known a priori?

Ray Casting (zero bounces) or Tracing (k bounces)?
Casting highly coherent; tracing less so

Today:
static scene, static observer
static scene, moving observer
ray casting
ray tracing
generalizations

Observations:

Low-level:
    Majority of effort (> 90\%) for ray/object intersections
    Successive ray trees have similar topology, geometry

High-level:

   spatial coherence:
low-level:
  nearby rays have correlated radiance values (colors)
high-level:
  rays typically come in pencils (e.g., view frustum)

   temporal coherence:
low-level:
  if asked about ray, will soon be asked about ``nearby'' rays

high-level:
  sets of rays vary smoothly (e.g., view frustum)

Approaches:

1) Make each Ray/Object intersection faster

local: [Draw objects, leaves as nodes, then cluster up]
bounding hierarchy
(axial boxes; gen boxes; polyhedra; spheres)
group by: model semantics; proximity; best vol ratio

global: [draw objects, then surround by grid]
[Draw 2D regular grid; irreg grid; kd tree]
spatial data structure (voxel, octree, kdtree)

[Draw xy and theta]
five-d (x,y,z,theta,phi) subdivision

[show duality of 2d lines to 2d points]
line-space hierarchy, 4D per spatial cell

2) Trace different entity (beam, cone, radiance sample)

make each sample represent more information
beam tracing: polyhedral beam
cone tracing: circular cross section
radiance tracing: use gaussian curvature
problems? geometric approximation

3) Make fewer ray/object intersection queries

global visibility data structure:
determine fate of many rays at once
cell/portal subdivision, frustum advance

classify many rays at once
frustum subdivision for ray casting
works well for casting -- not tracing

spawn fewer samples; interpolate for reconstruction
fixed coarse subdivision
adaptive subdivision
how to know when to subdivide/terminate ?
look at variance of corners, but...

prove something about interior

4) Generalizations:

dynamic lights/material properties
store full color expression at each pixel
use common subtree elimination for compactness
careful: can't apply usual truncation criteria!

dynamic scene? repopulate objects to subdivision
treat it as cache; clear valid bit
must also clear entries involving this object

multiprocessor architecture:
ray cache; fill with predictive queries
eject with LRU or geometric criterion