Accelerated ray tracing

Brian Curless
CSE 557
Autumn 2017

Faster ray-polyhedron intersection

Let's say you were intersecting a ray with a triangle mesh:

- intersect the ray with each triangle
- return the intersection with the smallest t-value.

Q: How might you speed this up?

Bounding Volume Hierarchies (BVHs)

We can generalize the idea of bounding volume acceleration with bounding volume hierarchies (BVHs).

Intersect with largest BV... ...
then intersect with children...
...
until you reach the leaf nodes - the primitives.

Key: build balanced trees with tight bounding volumes.
Bounding Volume Hierarchies (BVHs)

How do you build a tree?

1. Bottom up: start with individual primitives and gradually cluster them into a tree.
2. Top down: start with one bounding volume around all the primitives and recursively split into two.

Recommendation: go with top down – easier to do, works well.

For top down, how to decide where to split?

Choose a splitting axis and then follow one of these heuristics:

- Find the median of the bounding box centers along the axis and split at that location
- Find the midpoint of the parent bounding box and split there
- Find the split that minimizes the Surface Area Heuristic (SAH) cost:
  \[ N_{\text{left}} \cdot \text{SurfaceArea}(V_{\text{left}}) + N_{\text{right}} \cdot \text{SurfaceArea}(V_{\text{right}}) \]

Then move on to the next axis and repeat.

Uniform spatial subdivision

Another approach is uniform spatial subdivision.

Idea:

- Partition space into cells (voxels)
- Associate each primitive with the cells it overlaps
- Trace ray through voxel array using fast incremental arithmetic to step from cell to cell

Q: Given a 10^6 triangle football stadium with a 10^6 triangle teapot on one of the seats, would a single uniform spatial subdivision be a good idea?

Non-uniform spatial subdivision: octrees

Another approach is non-uniform spatial subdivision.

One version of this is octrees:

Non-uniform spatial subdivision: k-d trees

Another non-uniform subdivision is k-d (k-dimensional) trees:

If the planes can be non-axis aligned, then you get BSP (binary space partitioning) trees.

Various combinations of these ray intersections techniques are also possible.

[Image credits: Wikipedia.]