Assignment 6: Acceleration

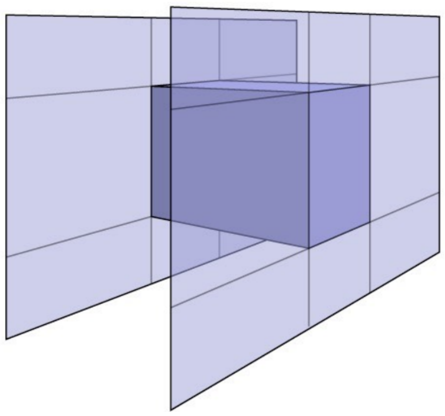
Mengzhu Wang

January 28, 2024

1 Ray Intersection With Box

Quick way to avoid intersections is to bound complex object with a simple volume. Object is fully contained in the volume. If it doesn't hit the volume, it doesn't hit the object. So we can test Bounding Volumes first, then test object if it hits.

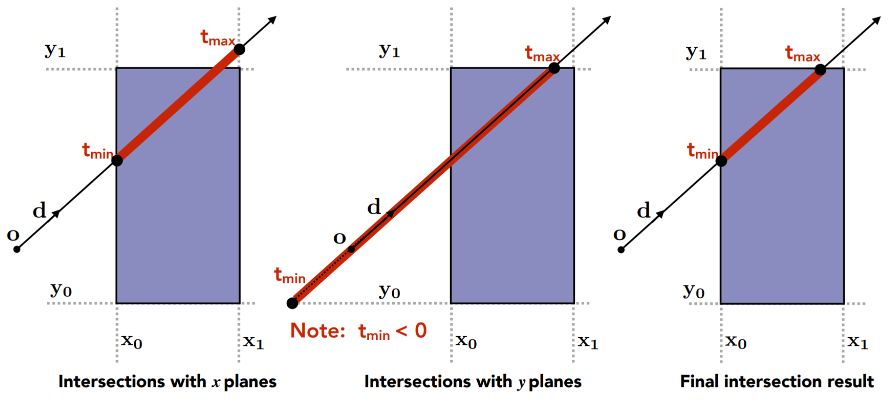
Box is the intersection of 3 pairs of slabs. Specially, we often use an ***Axis-Aligned Bounding Box (AABB)***. Any side of the bounding box is along either x, y or z axis.



Take 2D as an example, we can compute intersections with slabs and take intesection of t_{min}/t_{max} intervals. A 3D box is similar with three pairs of infinitely large slabs. The key ideas are:

- The ray enters the box only when it enters all pairs of slabs.
- The ray exits the box as long as it exits any pair of slabs.

Therefore, we have $t_{enter} = \max(t_{min})$ and $t_{exit} = \min(t_{max})$ for the 3D box. And in summary, ray and AABB intersect iff $t_{enter} < t_{exit} \quad \&\& \quad t_{exit} >= 0$.

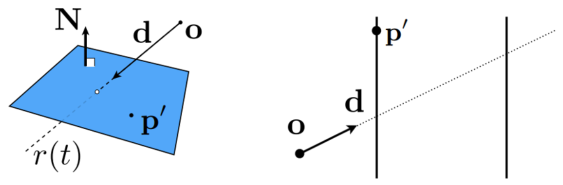

y1
tmax
tmin
d
o
y0
x0
x1
Intersections with x planes
y1
tmax
d
o
tmin
y0
Note: tmin < 0
x0
x1
Intersections with y planes
y1
tmax
tmin
d
o
y0
x0
x1
Final intersection result

In addition, axis-aligned can save computation. General intersection's equation is

$$t = \frac{(\mathbf{p}' - \mathbf{o}) \cdot \mathbf{N}}{\mathbf{d} \cdot \mathbf{N}} \tag{1}$$

It needs 3 subtractions, 6 multiplies and 1 division. While for slabs perpendicular to x-axis, the equation can be transformed as nelow with only 1 subtraction and 1 division.

$$t = \frac{\mathbf{p}'_x - \mathbf{o}_x}{\mathbf{d}_x} \tag{2}$$



2 Bounding Volume Hierarchy (BVH)

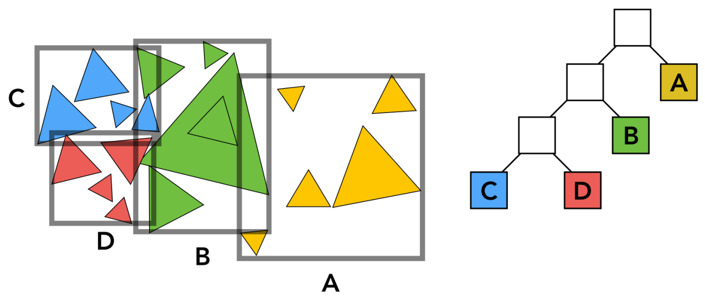
2.1 BVH Building

The steps of building BVHs

- Find bounding box
- Recursively split set of objects in two subsets
- Recompute the bounding box of the subsets
- Stop when necessary
- Store objects in each leaf node

The data structure for BVHs

- Internal nodes
  - Bounding box
  - Children: pointers to child nodes
- Left nodes
  - Bounding box
  - List of objects



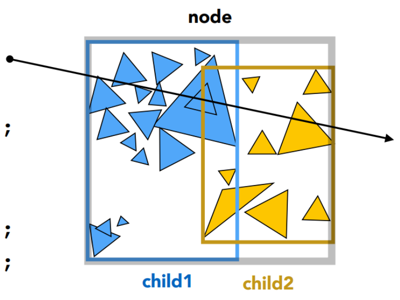
2.2 BVH Traversal

```
Intersect(Ray ray, BVH node) {
  if (ray misses node.bbox) return;

  if (node is a leaf node)
     test intersection with all objs;
     return closest intersection;

  hit1 = Intersect(ray, node.child1);
  hit2 = Intersect(ray, node.child2);

  return the closer of hit1, hit2;
}
```


node
child1
child2