

I implemented the acceleration structure of a BVH (Bounding Volume Hierarchy). This structure divides the objects in the scene based on their bounding boxes. It generates a binary tree. It's non-terminal nodes represent the bounding box containing children nodes. It's terminal nodes represent the objects themselves. The tree creates the binary tree by sorting the objects based on the center point of their bounding boxes. Each level of the tree sorts these center points by an alternating axis (x, then y, then z). Then, it splits the sorted object list in half and generates a bvh for each half that are then set as the left and right child nodes.

When the hit() function is called on the BVH, it is first determined if the ray hits the top-level BVH bounding box. If it does hit BVH's bounding box, then it checks the children BVH bounding boxes. If it hits either of these, it will continue to traverse down the nodes and determine if the children BVHs / objects were hit. Once it determines if the left and right child nodes were hit, it compares the parametric t-values to determine which side was hit by the ray first. This is then the material used in the shade record.

I want to go back and make the Triangle Mesh itself implemented with a BVH to store the individual triangles. Currently, the acceleration structure only helps for the area of the image where the mesh bounding box is not hit. So, when a ray does hit the mesh's bounding box, every triangle in the mesh is traversed. This could obviously be fixed using the BVH, but I have not implemented this yet.

Render	BVH Used?	Render Time (seconds)
Cow mesh	No	212.77
Cow mesh	Yes	42.4238
5^3 spheres (125)	Yes	2.858
11^3 spheres (1331)	Yes	4.723
22^3 spheres (10648)	Yes	13.127

CODE CITATIONS:

Nearly all of the code I wrote in the ray tracer was heavily taken from the Ray Tracing From the Ground Up book by Kevin Suffern. However, I re-typed all of this code by hand / line-by-line. I did this to understand the code better and make it more similar to my coding style / deleted things I thought were unnecessary. I also used a few other websites for topics that were not covered well in Suffern's book.

Most of the code: Ray Tracing From the Ground Up: Kevin Suffern

Most of the code (Fork of Kevin Suffern's code):

https://github.com/tfiner/RTFGU/blob/f1d6c164afeb8bd38b74171582b06d9d30cc7e25/src/geo_obj/BBox.h

OBJ Parser: <https://stackoverflow.com/questions/21120699/c-obj-file-parser>

BVH: https://graphics.stanford.edu/~boulos/papers/efficient_notes.pdf

BVH:

<https://www.scratchapixel.com/lessons/advanced-rendering/introduction-acceleration-structure/bounding-volume-hierarchy-BVH-part1>