

## (Task 3) Bounding Volume Hierarchy

Nick Sharp edited this page 11 days ago · 3 revisions

In this task you will implement a bounding volume hierarchy that accelerates ray-scene intersection. All of this work will be in the `BVHAccel` class in `bvh.cpp`.

The starter code constructs a valid BVH, but it is a trivial BVH with a single node containing all scene primitives. A `BVHNode` has the following fields:

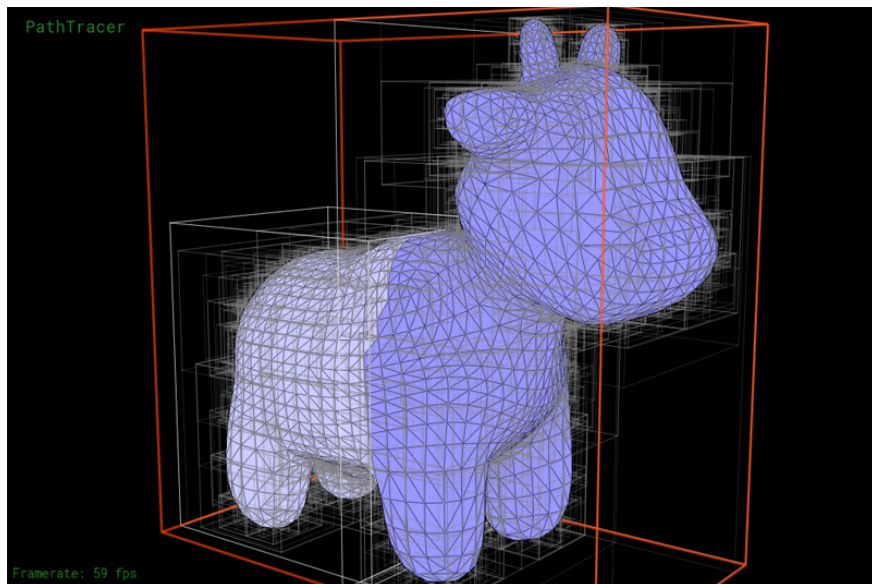
- `BBox bb` : the bounding box of the node (bounds all primitives in the subtree rooted by this node)
- `int start` : start index of primitives in the BVH's primitive array
- `size_t range` : range of index in the primitive list (number of primitives in the subtree rooted by the node)
- `BVHNode* l` : left child node
- `BVHNode* r` : right child node

The `BVHAccel` class maintains an array of all primitives in the BVH (primitives). The fields `start` and `range` in the `BVHNode` refer the range of contained primitives in this array. The primitives in this array are not initially in any particular order, and you will likely want to rearrange the order as you build the BVH.

### Step 1: BVH Construction

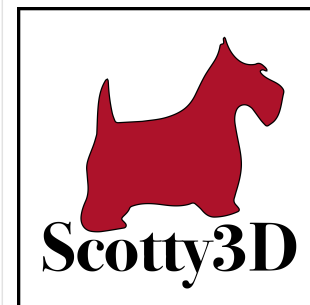
Your job is to construct a BVH using the [Surface Area Heuristic](#) discussed in class. Tree construction should occur when the `BVHAccel` object is constructed.

We have implemented a number of tools to help you debug the BVH. Press the `v` key to enter BVH visualization mode. This mode allows you to directly visualize a BVH as shown below. The current BVH node is highlighted in red. Primitives in the left and right subtrees of the current BVH node are rendered in different colors. Press the `<` or `>` keys to descend to child nodes of the mesh. Press `?` to move the parent of the current node.



Another view showing the contents of a lower node in the BVH:

► Pages 33



[Home](#)

[Project Philosophy](#)

[Build Instructions](#)

#### User Guide

- [Overview](#)
- [MeshEdit](#)
- [PathTracer](#)
- [Animation](#)

#### Developer Manual

[Overview](#)

#### MeshEdit

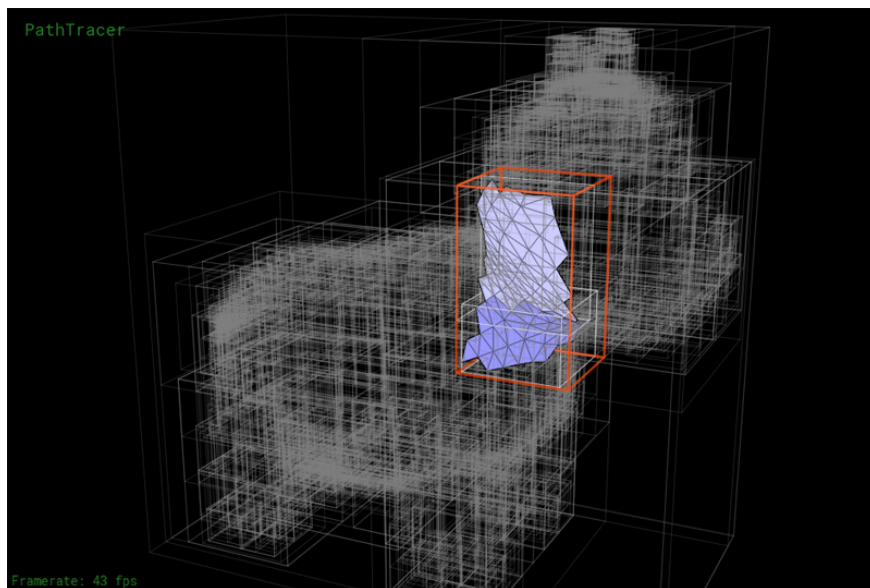
[MeshEdit Overview](#)

- [Halfedge Mesh](#)
- [Local Mesh Operations](#)
  - [Tutorial: Edge Flip](#)
  - [Beveling](#)
- [Global Mesh Operations](#)
  - [Triangulation](#)
  - [Subdivision](#)
    - [Linear Subdivision](#)
    - [Catmull-Clark Subdivision](#)
    - [Loop Subdivision](#)
  - [Simplification](#)
  - [Isotropic Remeshing](#)

#### PathTracer

[PathTracer Overview](#)

- [Task 1: Camera Rays](#)
- [Task 2: Intersecting Primitives](#)
- [Task 3: BVH](#)



- [Task 4: Shadow Rays](#)
- [Task 5: Path Tracing](#)
- [Task 6: Materials](#)
- [Task 7: Environment Light](#)

Notes:

- [Ray-triangle intersection](#)
- [Dielectrics and Transmission](#)
- [Importance Sampling](#)

### Animation

[Animation Overview](#)

Clone this wiki locally

<https://github.com/cmu462>



## Step 2: Ray-BVH Intersection

Implement the ray-BVH intersection routines required by the Primitive interface. You may wish to consider the node visit order optimizations we discussed in class. Once complete, your renderer should be able to render all of the test scenes in a reasonable amount of time.

[15-462 Computer Graphics](#) | [CMU Graphics](#) | [CMU Computer Science](#)

**Carnegie Mellon University**