CS 201: Data Structures II
BVH Trees for Accelerated Ray Tracing

L1 - Group 4

Spring 2023

1 Group Members

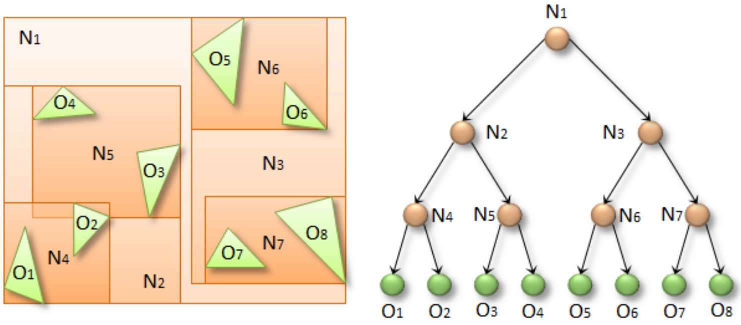
1. Mohammad Ali Naqvi
2. Hammad Sajid
3. Muhammad Khubaib Mukaddam
4. Muhammad Hussain

2 Data Structure

BVH Tree (Bounding Volume Hierarchy) is a binary tree used in computer graphics and computational geometry to efficiently organize and query objects in a 3D space. Each node in the tree contains:

- A bounding box that contains all the objects in the subtree rooted at that node.
- A pointer to the left and right child nodes.
- Two iterators start and end pointing to the beginning and the end of a list of actual scene primitives.

In a Bounding Volume Hierarchy, all geometric objects are wrapped in bounding volumes that form the leaf nodes of the tree. These nodes are then grouped as small sets and enclosed in larger sets. These in turn are also grouped recursively and included in other larger boundary volumes, resulting in a tree structure with a single boundary volume at the top of the tree. This is precisely the reason why BVH are used for coliision detection and ray tracing in computer graphics [1].



The image above shows a simple BVH tree that traces the triangles bounded by a box that is again bounded by a box in a hierarchal order.

3 Application

There are many applications of BVH trees, specifically in computer graphics. These include, but are restricted to:

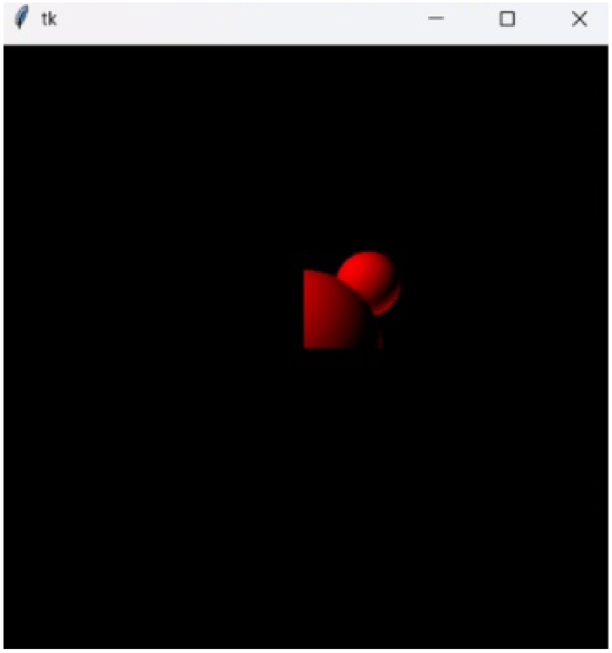
- Collision detection: Since BVH trees are used to efficiently organize and query objects in a 3D space, they are used in collision detection algorithms. For example, in the case of a game, the BVH tree is used to detect collisions between the player and the objects in the game quickly and efficiently.
- Ray tracing: Ray tracing is a technique used to generate an image by tracing the path of light as pixels in an image plane and simulating the effects of its encounters with virtual objects. BVH trees are used to accelerate ray tracing algorithms by reducing the number of intersections that need to be calculated while also compute the color and intensity of the light at that point.
- CAD/CAM: BVH trees can be used in computer-aided design (CAD) and computer-aided manufacturing (CAM) applications to quickly determine whether a tool or object will collide with a surface during machining operations.

Out of these applications, we will be using BVH trees to accelerate rar tracing algorithm. In the next section we will discuss the functionality of our interface.

4 Functionality of the Interface

There are several options for providing interface to visualize the traced image. The image visualization will be done using tkinter or matplotlib module in python. The approach we will be utilizing in our BVH tree will be from the bottom up. We will use simple bounding boxes to enclose the objects in the scene and then group them in a hierarchal order. To avoid complexity, our scene will consist of only spheres. We will construct a simple scene and construct the BVH tree uon the scene. The interface will show the traced sphere on the GUI (Graphical User Interface). The GUI will be static visualizer beacuse an intercative realtime visualizer requires advanced Computer

Graphics concepts and is out of the scope of the course. The image attached below gives a brief idea of how the interface will look like and how the sphere will be traced (This GUI was created through ChatGPT and is for demonstration purpose only and most likely may not be used in the final project) [2].



The output would be an image which would show the image produced by the BVH accelerated ray tracer.

The image above shows the GUI that will be used to display the traced sphere. This is not the most accurate depiction of the GUI, but it gives a general idea of how the GUI will look like.

The way we will be tracing the sphere by using the ray tracing algorithm is that as soon as the ray intersects with any of the pixel of the sphere, the color of that pixel will be changed to the color of the sphere. If there is no collision then the pixel will remain black (this is what the image above depicts in the background). These spheres will be placed in a hierarchal order and the ray will be traced from the bottom of the tree to the top. The bottom of the tree will consist of these spheres while their parent nodes will be the bounding boxes that enclose these spheres, the parent nodes of these bounding boxes will be additional bounding boxes. In this way, the image will be traced in a hierarchal order. This will include a lot of mathematical calculations that we will be doing in the backend.

5 Datasets

There isn't a dataset that we have decided to use for this project. In our project, we will pre-define the spheres and their positions and then we will construct the BVH tree using these spheres.

6 Work Distribution

The work distribution is tentative and is based on the project flow.

Item	Activity	Name
1	Building BVH Heirarchy	Muhammad Khubaib Mukaddam
2	Constructing BVH Tree	Hammad Sajid
3	Integrating BVH into RayTracer	Muhammad Hussain
4	Building RayTracer and	Muhammad Ali Naqvi

Ofcourse, these tasks above are dependent on one another and cannot be done independent of others concurrently.

References

[1] Efficient BVH Construction via Approximate Agglomerative Clustering by Y. Gu, Y. He, K. Fatahalian and G Blelloch, http://graphics.cs.cmu.edu/projects/aac/aac_build.pdf ,last accesed on 03/04/2023.

[2] Conversation with ChatGPT, OpenAI, https://chatgpt.com/, last accesed on 03/04/2023.