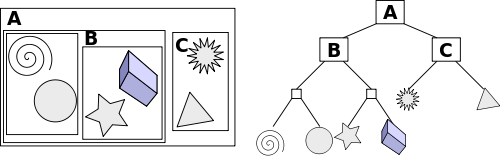
**Bounding Volume Hierarchy**

练孙鸿

1. **Introduction to BVH**

BVH(Boungding Volume Hierarchy)是一种用于加速射线碰撞检测的数据结构，并且是一种树(Tree)数据结构。场景中的所有物体都会计算挤一个轴对齐包围盒(AABB, Axis-Aligned Bounding Box)，每个场景对象都是绑定在BVH的叶结点上的。多个子节点的包围盒可以构成更大的、父节点的包围盒，使得任意层上的BVH节点的包围盒一定会包含其所有子节点所绑定的物体。那么，根节点的包围盒也必然包围着全场景的物体。



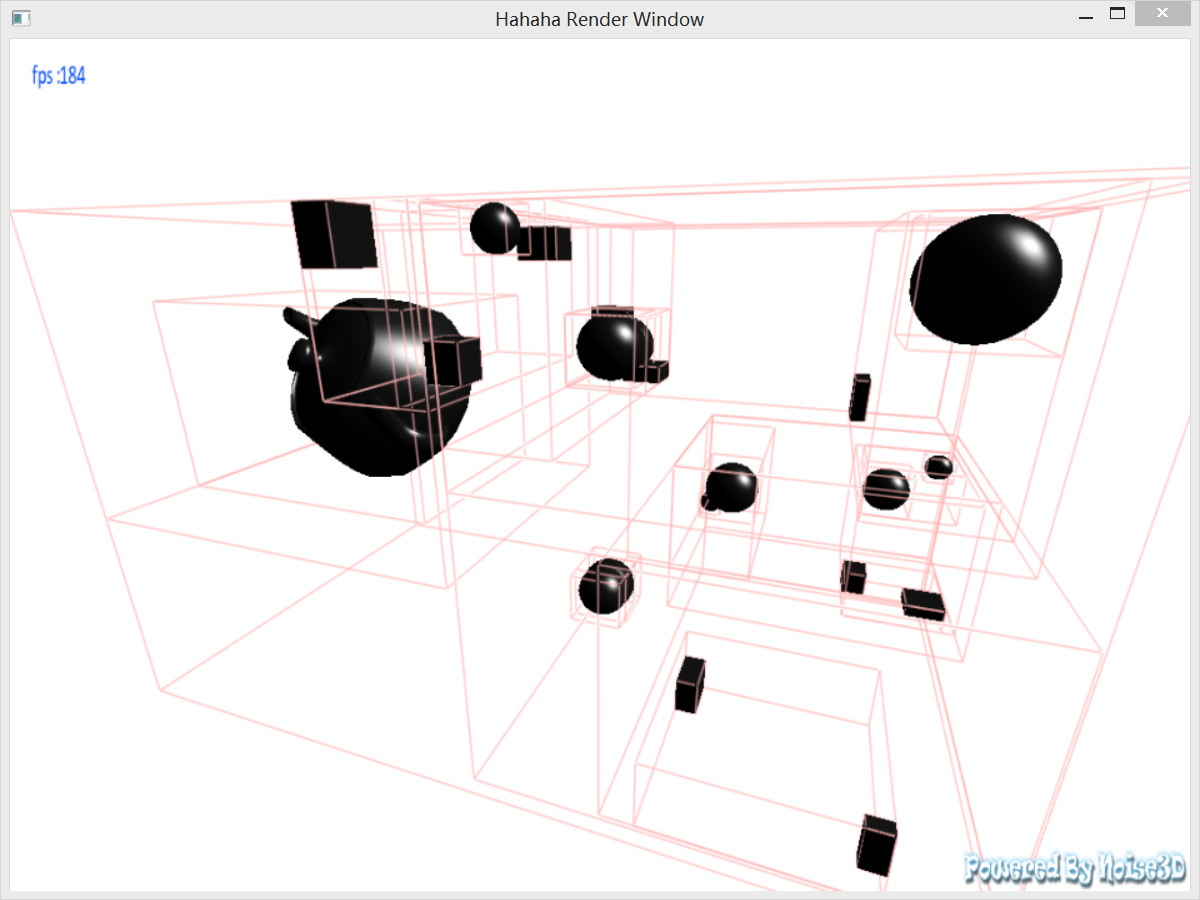
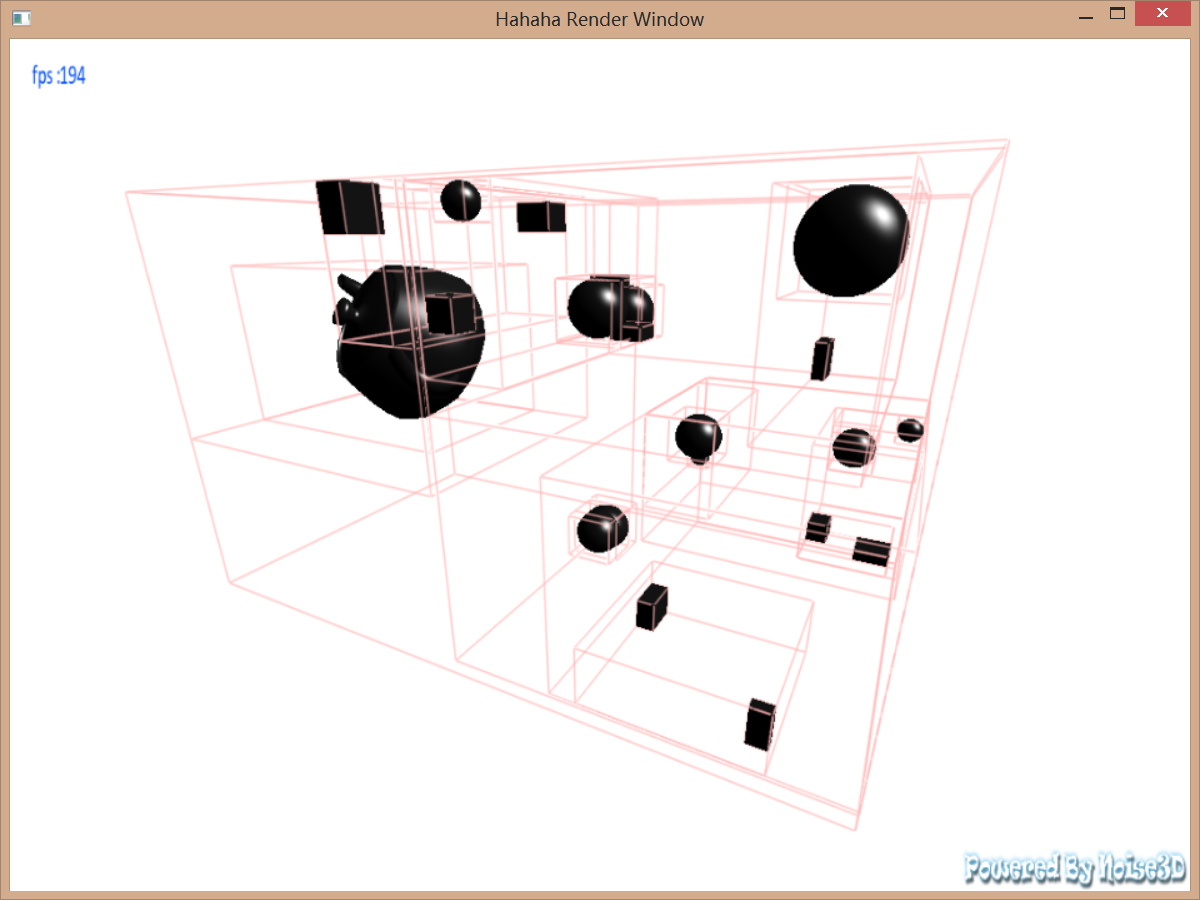
图：[1]BVH示意图

在光线追踪的时候，需要非常地频繁地检测射线与场景的交点。如果没有BVH，找出根射线与个场景物体的所有交点的渐进复杂度是，一般情况下远大于。但如果用平均代价（普通的构建方法的复杂度与构建一棵二叉搜索树类似）构造BVH之后，就可以用的渐进复杂度求出射线与场景的所有交点。可以用BVH实现碰撞检测加速的原因有：快速的Ray-AABB相交测试；BVH可以进行快速的搜索树剪枝(branch-pruning)，快速排除不可能与射线相交的物体以减少计算量。但要利用BVH进行加速之前，我们首先要基于具体的场景构建一棵BVH树。[1]构建BVH有多种方法，例如：自顶向下(Top-Bottom)，自底向上(Bottom-up)。**自顶向下法**的输入是场景的所有物体及其AABB，算法首先把输入物体的集合用某种方法分割成两个集合，然后递归地对分开的两个物体集合继续执行分割的操作，分到只剩单个不可分叶节点为止。虽然这个方法不保证能构造出一颗搜索高效、高度平衡的BVH树，但这是最快、最容易实现的，而且也是广泛使用的。**自底向上法**则是从叶节点开始两两合起来，但合并的方法比较复杂，本项目的实现不采用这种方法，具体实现可以参考[4]。本课题采用自顶向下的构建法，从BVH根节点的包围全场景的包围盒开始分割，递归地构建BVH。

1. **分割包围盒与物体**

在构建BVH时，某个父节点的包围盒所包围的物体需要用一定的方法进行分割、分类，分成两组或多组物体；每一组的物体及其包围盒构成的子节点。一般简单而高效分割方法有：按中点分割、按个数分割、表面积启发式分割(SAH, Surface Area Heuristic)、Morton Code编码分割。

中点分割法有很多种变体，但是都需要取某BVH节点AABB的某个笛卡尔坐标分量范围的中点。Blablabla介绍按质心分类，和Noise3D按AABB slabs来分类，累了先不写了。



**引用**

[1] Wikipedia, Bounding volume hierarchy. [DB/OL] <https://en.wikipedia.org/wiki/Bounding_volume_hierarchy>, 2019-3,

[2] Y. Gu, Y. He, K. Fatahalian and G Blelloch. [Efficient BVH Construction via Approximate Agglomerative Clustering](http://graphics.cs.cmu.edu/projects/aac/aac_build.pdf). HPG 2013.