

🔗 1. 八叉树Octree扩展库介绍

本章节学习八叉树Octree目的，是为了实现漫游的碰撞检测功能，比如遇到障碍物被挡住、比如爬坡和上楼梯。

你可以打开本节课的工厂漫游案例体验测试。

本节课内容会比较多，不过大部分只是作为了解和扩展学习，你只需要你掌握下面3行代码即可。

```
// 引入八叉树扩展库
import { Octree } from 'three/examples/jsm/math/Octree.js';
const worldOctree = new Octree();
// 分割模型，生成八叉树节点
worldOctree.fromGraphNode(模型对象);
```

js

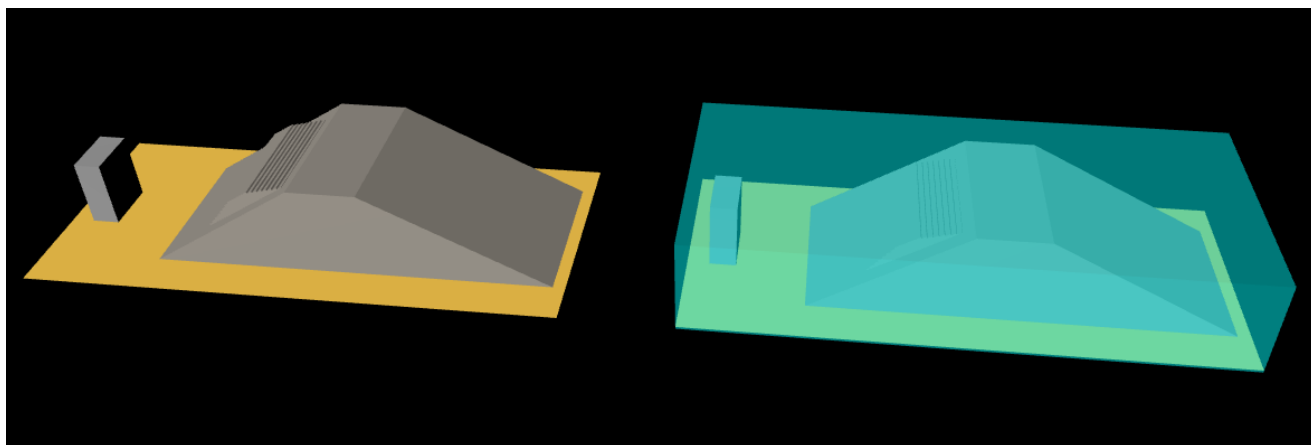
八叉树基本原理解释

下面给大家简单介绍下[八叉树](#)🔗 概念，初学者不要求记住具体细节，先有个印象就行。

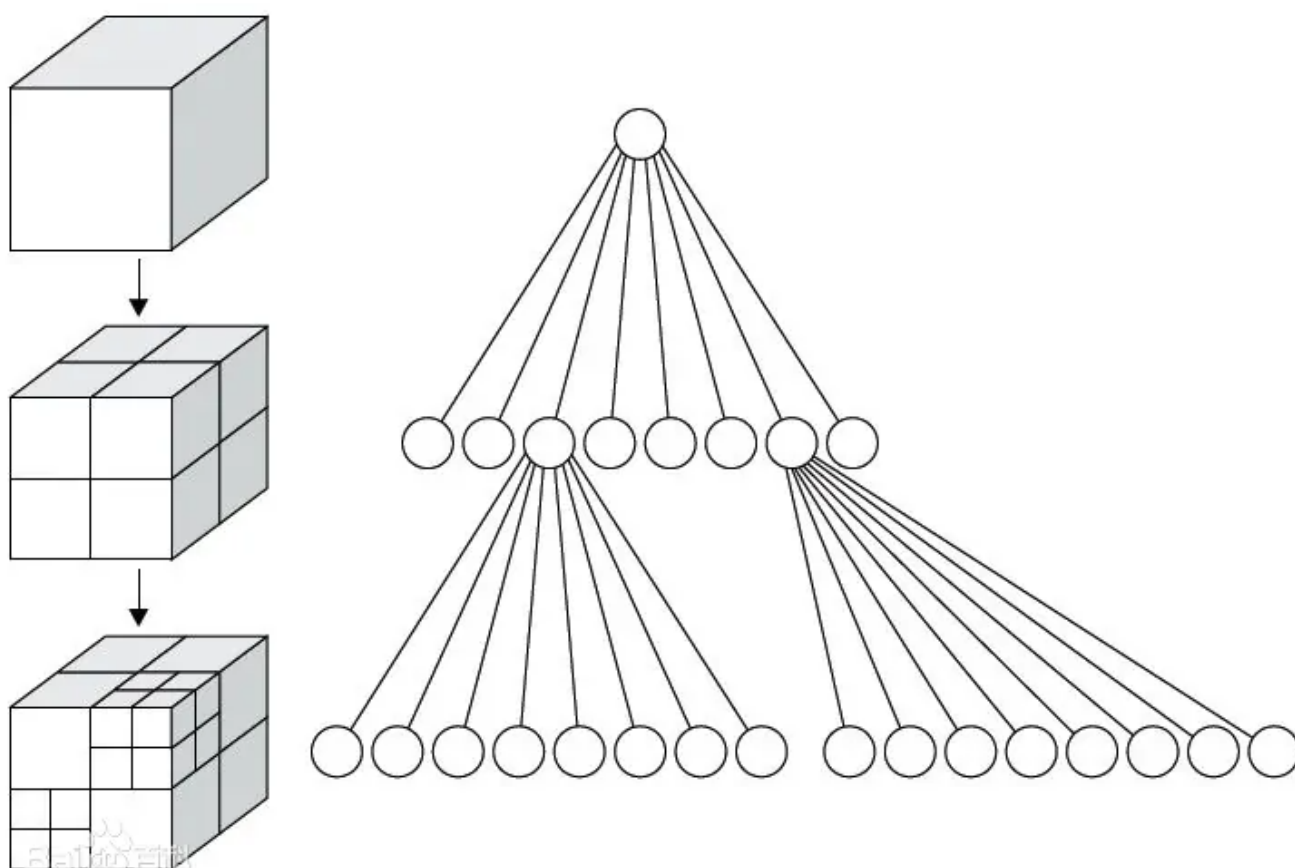
通过前面基础内容[2.3. 网格模型\(三角形概念\)](#)🔗 的学习，大家都知道网格模型Mesh本质是由三角形构成，三角形由顶点构成，这些三角形和自己的顶点数据分布在3D空间中。

```
const geometry = new THREE.BoxGeometry(50,50,50);
console.log('顶点位置数据',geometry.attributes.position);
console.log('三角形顶点索引数据',geometry.index);
```

js

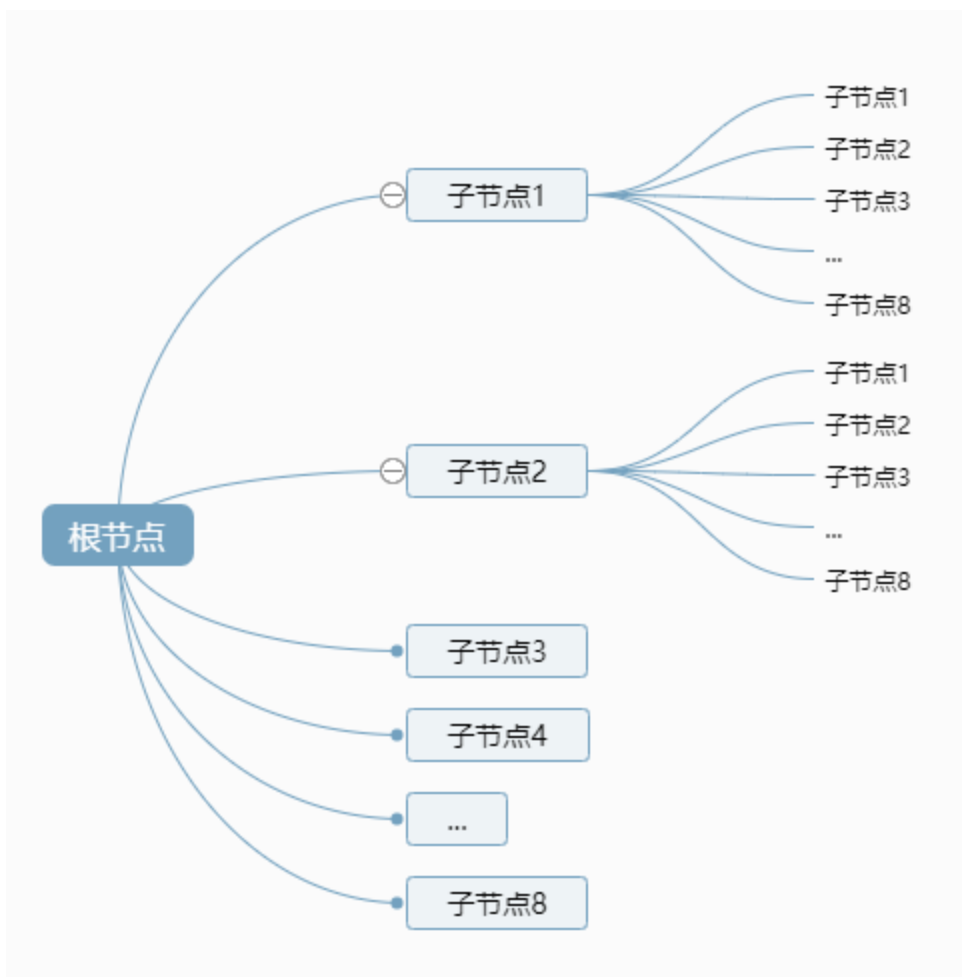


如果整个3D模型用一个长方体空间来表示，在三维空间xyz三个方向，都分割一次，这样就可以得到8个小的长方体子空间。



一个3D模型的三角形(顶点)分布在三维空间中，如果你用一个长方体来表示整个3d场景，当你分割为8个子空间的时候，每个子空间可以包含对应的三角形(顶点)数据。

每个子空间如果三角形(顶点)数量比较多，还可以继续分割，具体分割规则，你可以自定义，比如你可以规定，一个子空间包含的三角形数量只要大于8个就继续分割。这样一个个子空间可以构成一个树结构，整体来看，每个节点，分叉出来八个子节点。



项目引入 Octree.js

Three.js在目录 `/examples/jsm/math/` 下提供了一个八叉树相关的扩展库 `Octree.js`。npm 安装threejs情况下，Octree.js扩展库引入路径。

```
// 引入/examples/jsm/math/目录下八叉树扩展库
import { Octree } from 'three/examples/jsm/math/Octree.js';
```

js

在 `.html` 文件中，你也可以配置为其它任意路径引入方式

```
<script type="importmap">
  {
    "imports": {
      "three": "../..../three.js/build/three.module.js",
      "three/addons/": "../..../three.js/examples/jsm/"
    }
  }
</script>
<script type="module">
import { Octree } from 'three/addons/math/Octree.js';
```

html

```
</script>
```

生成八叉树 `.fromGraphNode()`

实例化一个八叉树对象。

```
const worldOctree = new Octree();
```

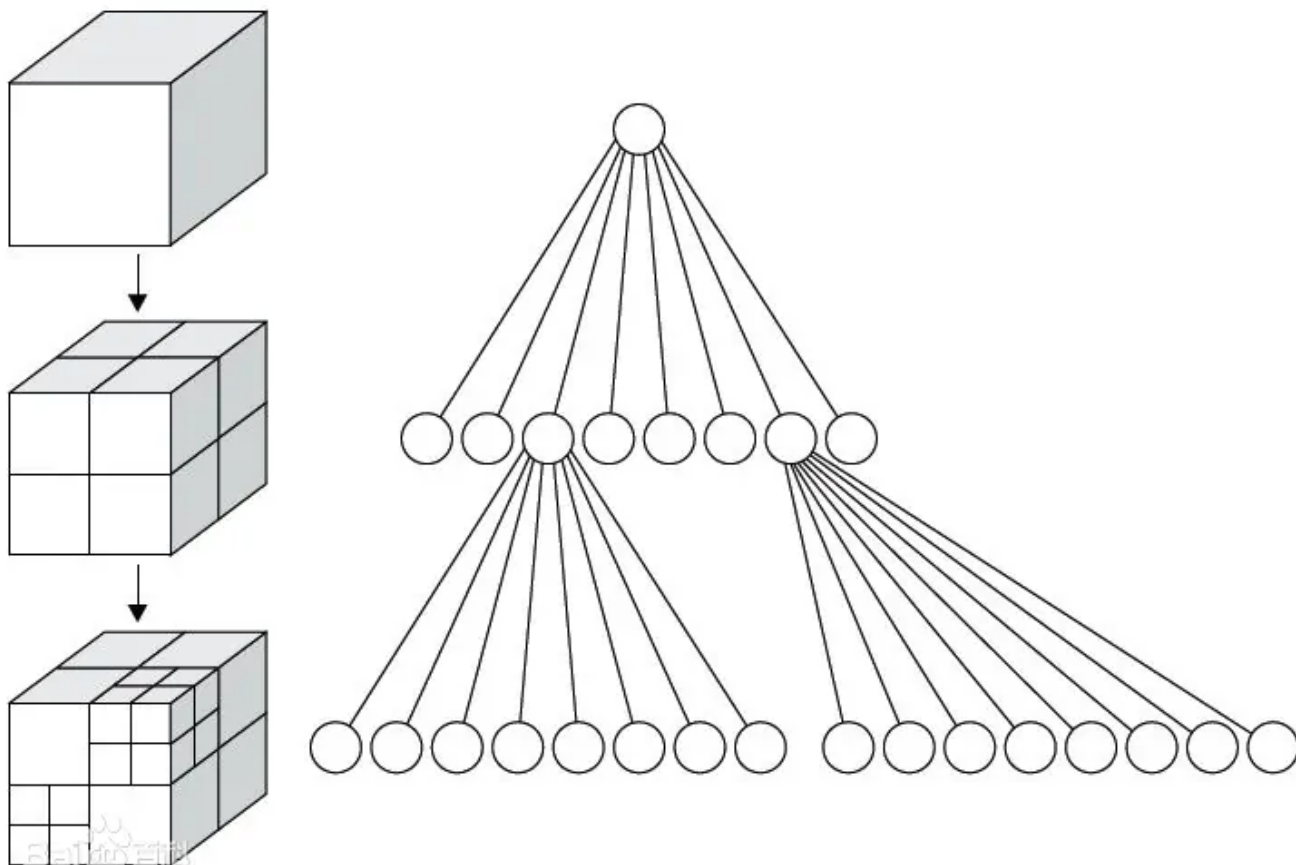
js

`.fromGraphNode()` 的参数是模型对象，比如一个mesh，或者多个mesh构成的层级模型。

```
const gltf = await loader.loadAsync("../地形.glb");  
worldOctree.fromGraphNode(gltf.scene);
```

js

执行 `.fromGraphNode()` 会对模型进行分割，分割为一个一个的小的长方体空间，构成一个八叉树。



执行 `.fromGraphNode()` 会把一个3D模型，分割为8个子空间，每个子空间都包含对应的三角形或者说顶点数据，每个子空间还可以继续分割。

具体分割规则非常复杂，不要求掌握，如果你有兴趣可以阅读 `Octree.js` 的源码，比如 `Octree.js` 会根据三角形数量决定是否分割一个子空间，比如一个子空间包含的三角形数量

小于等于8个就不在分割，当然你也可以修改规则，作为初学者，也不要求记住，先有个印象就行。

Octree.js文件中部分源码截取

```
if ( len > 8 && level < 16 ) {  
    subTrees[ i ].split( level + 1 );  
}
```

js

浏览器控制台打印八叉树

浏览器控制台打印八叉树，查看分割的结果(不要求掌握,过一遍即可)。

```
console.log('查看八叉树结构', worldOctree);
```

js

- `.box` 属性是包围盒Box3，描述当前分割的子空间位置和尺寸
- `.subTrees` 属性表示八叉树的子节点，类似threejs层级模型的 `children` 属性
- 查看叶子结点(最后一层没有子对象的节点) `.triangles` 属性,可以看到包含的三角形数据

OctreeHelper 可视化八叉树

Three.js在目录 `/examples/jsm/helpers/` 下提供了一个可视化八叉树相关的扩展库

`OctreeHelper.js` 。

```
import { OctreeHelper } from 'three/examples/jsm/helpers/OctreeHelper.js';
```

js

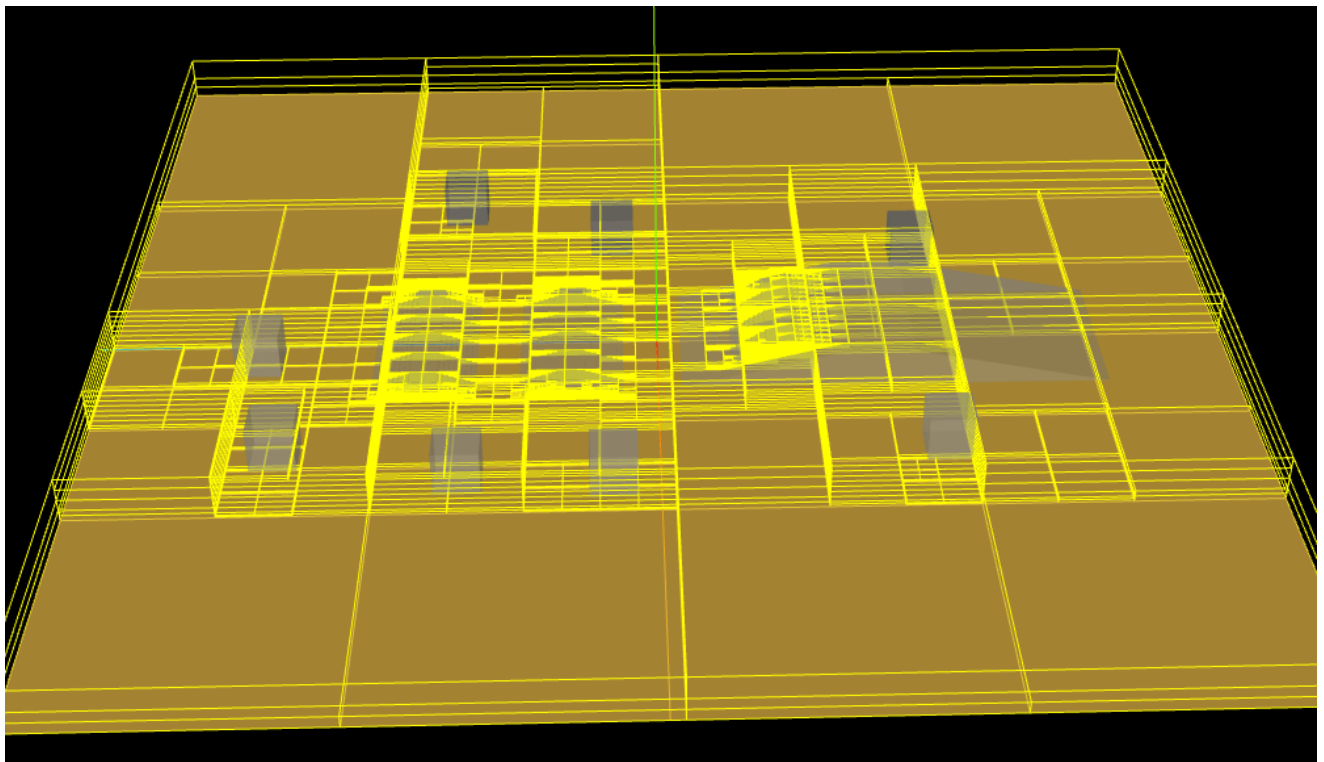
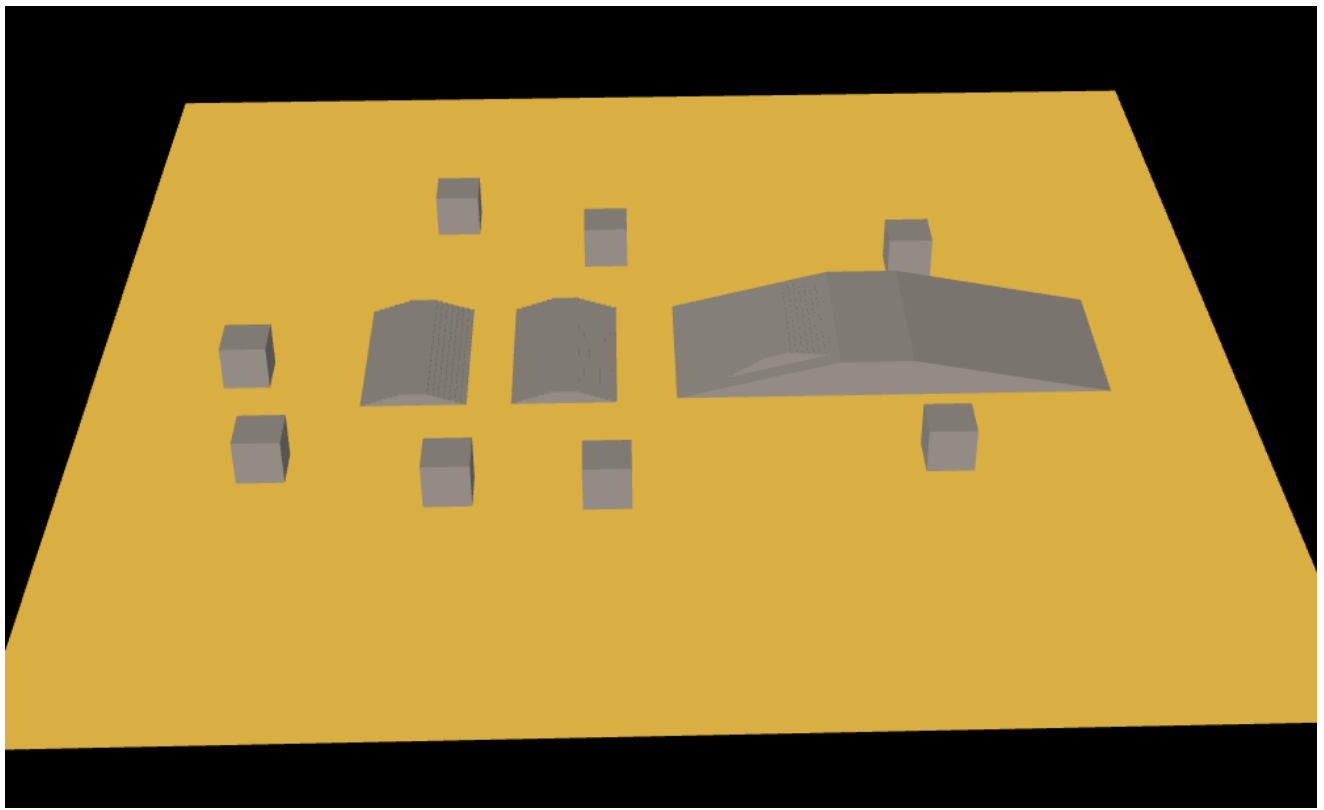
//课程案例源码里面配置的路径

```
import { OctreeHelper } from 'three/addons/helpers/OctreeHelper.js';
```

js

```
const helper = new OctreeHelper( worldOctree );  
scene.add( helper );
```

js



← 13. 骨骼动画与运动状态关联

2. 八叉树与胶囊Capsule交叉计算→

