

规则形体（下）： 空间分割表示

华中科技大学软件学院 万琳



0

实体模型的三类表示

- ◆ **边界表示 (Boundary Representation, B-reps)** , 即用一组曲面 (含平面) 来描述物体 , 这些曲面将物体分为内部和外部。边界表示具体又包括多边形表面模型和扫描表示两种。
- ◆ **构造实体几何表示 (Constructive Solid Geometry , CSG)** , 它将实体表示成立方体、长方体、圆柱体、圆锥体等基本体素的组合 , 可以采用并、交、差等运算构造新的形体。
- ◆ **空间分割表示 (Space-Partitioning)** , 用来描述物体的内部性质 , 将包含一物体的空间区域划分成一组小的、非重叠的、连续实体 (通常是立方体) 。



提纲

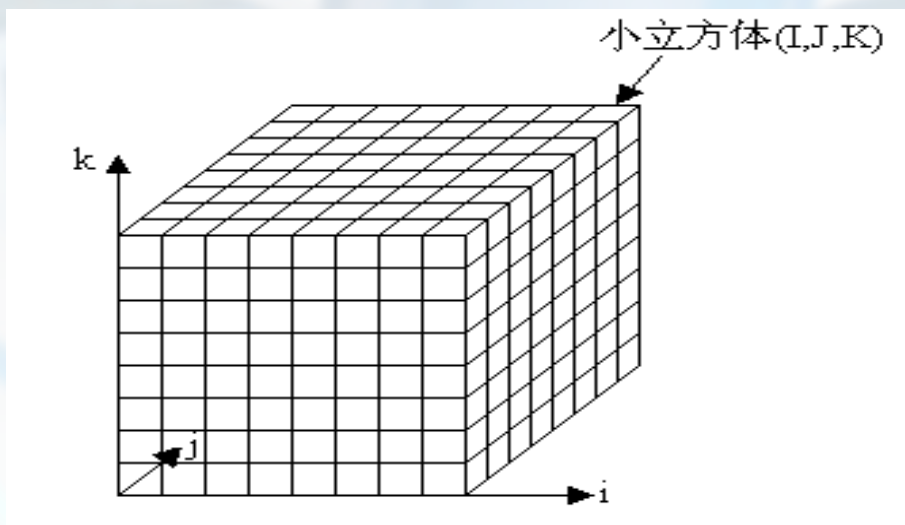
- ① 空间位置枚举表示
- ② 八叉树
- ③ BSP树

1

空间位置枚举表示

将包含实体的空间分割为大小相同、形状规则（正方形或立方体）的体素，然后，以体素的集合来表示图形对象。

➤ 用三维数组 $P[I][J][K]$ 表示物体，数组中的元素与单位小立方体——对应



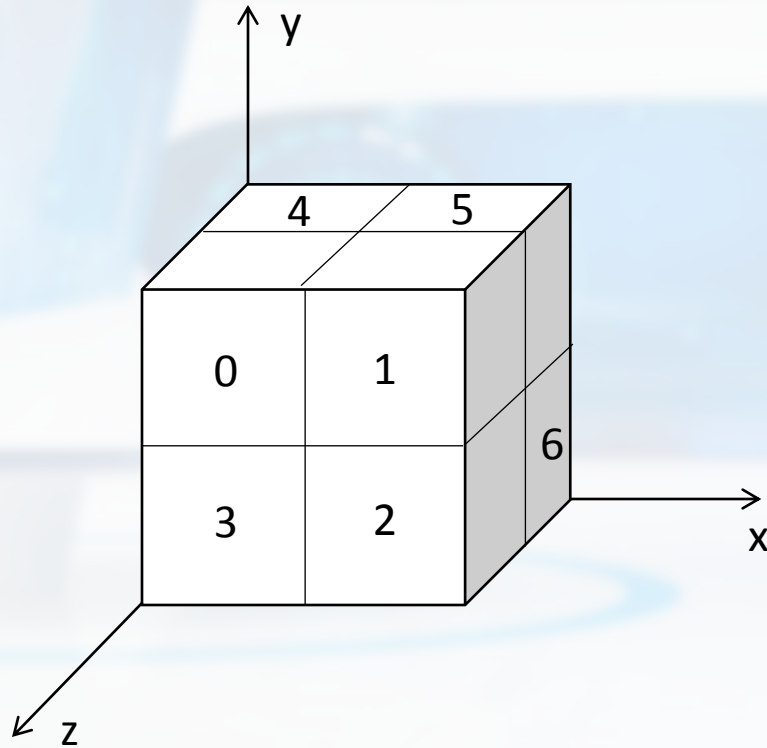
➤ 当 $P[I][J][K] = 1$ 时，表示对应的小立方体被物体占据

➤ 当 $P[I][J][K] = 0$ 时，表示对应的小立方体没有被物体占据

2

八叉树

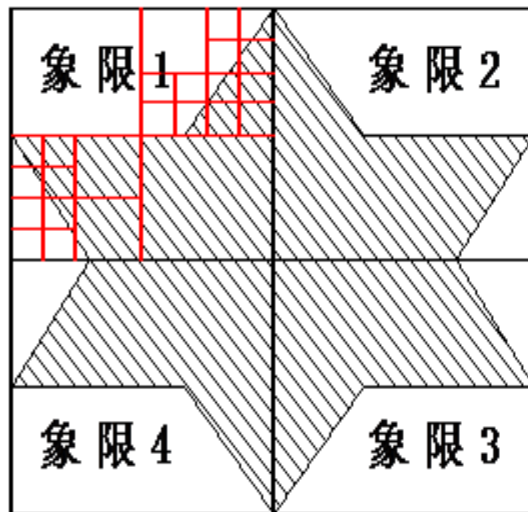
八叉树 (octrees) 又称为分层树结构，它对空间进行自适应划分，采用具有层次结构的八叉树来表示实体。



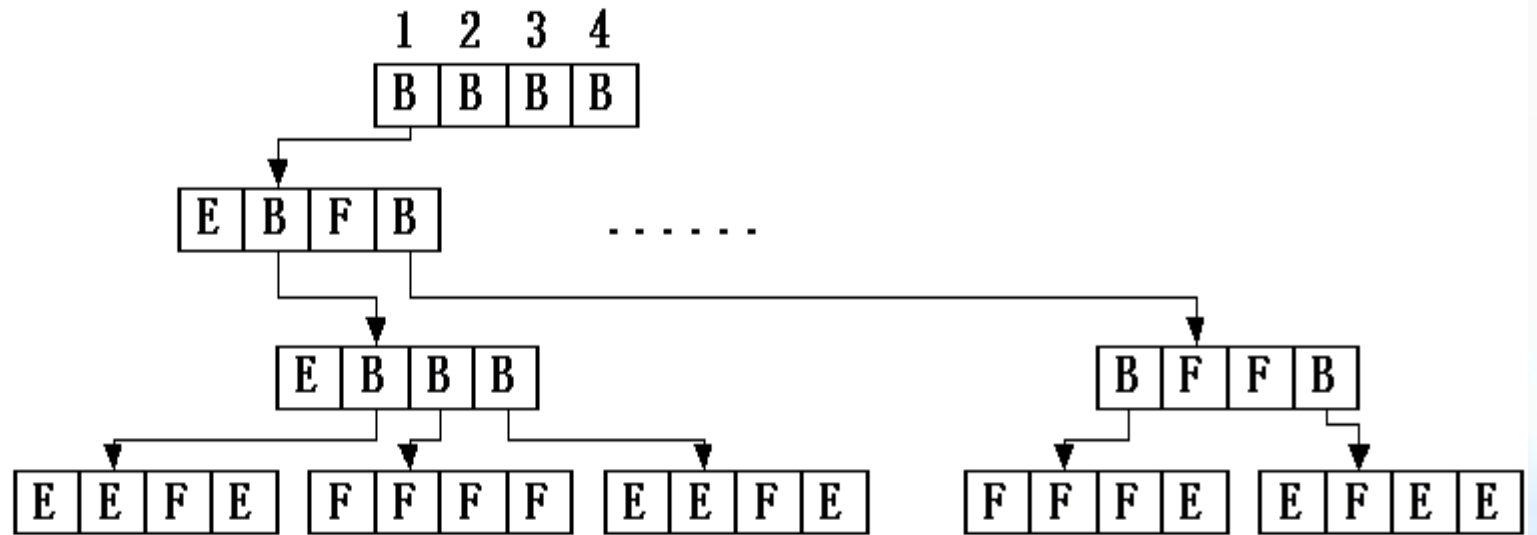
2

八叉树

四叉树



(a)

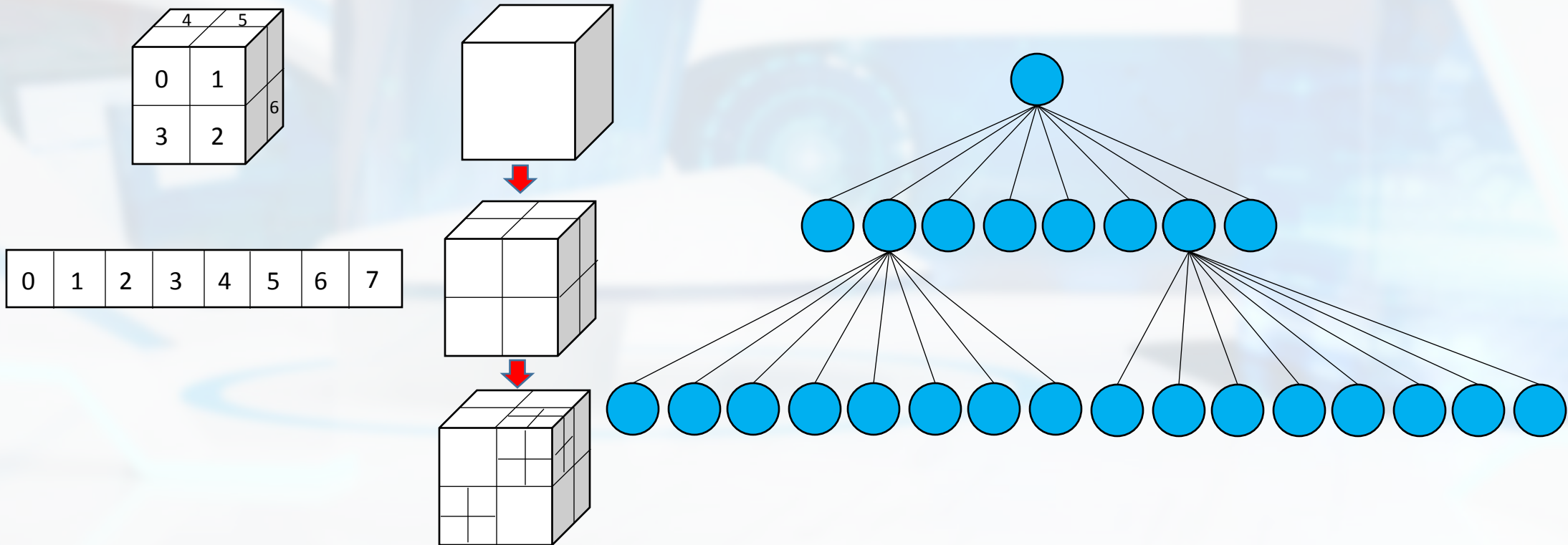


(b)

B:Boundary(边界) E:Empty(空) F:Full(满)

2 八叉树

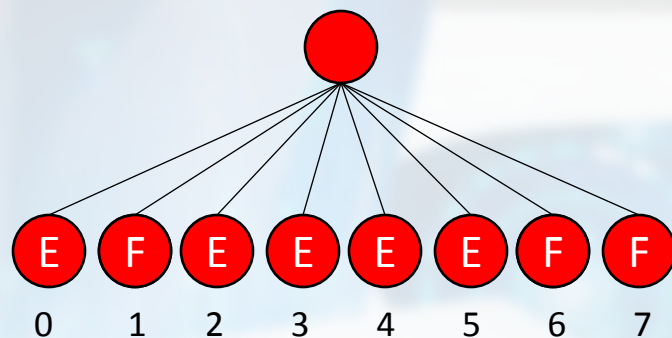
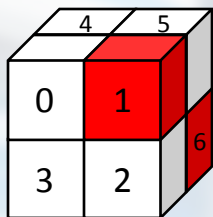
八叉树 (octrees) 又称为分层树结构，它对空间进行自适应划分，采用具有层次结构的八叉树来表示实体。



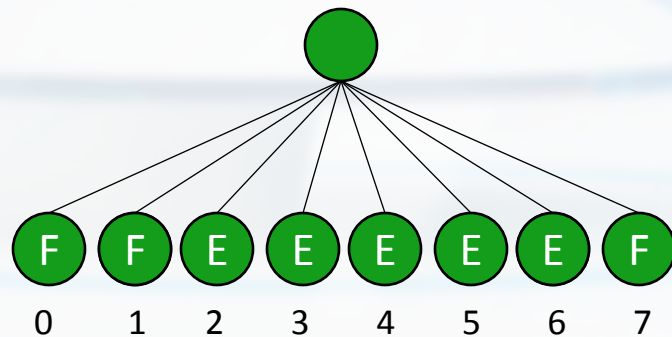
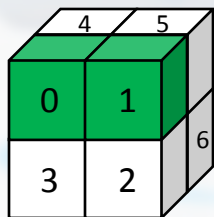
2 八叉树

基于八叉树的集合运算：

形体A



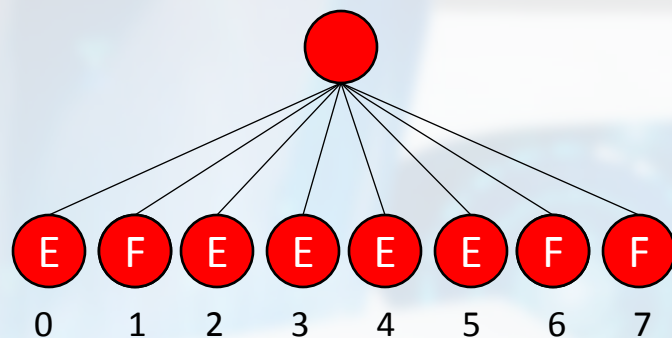
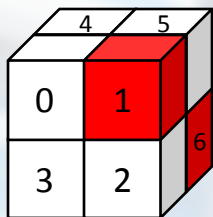
形体B



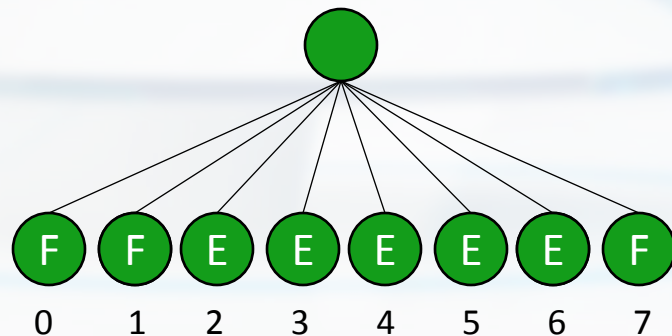
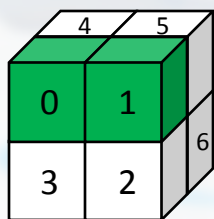
2 八叉树

基于八叉树的集合运算：

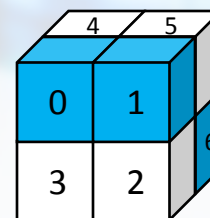
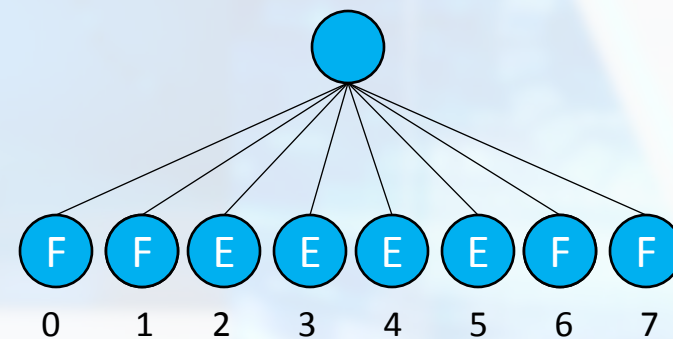
形体A



形体B



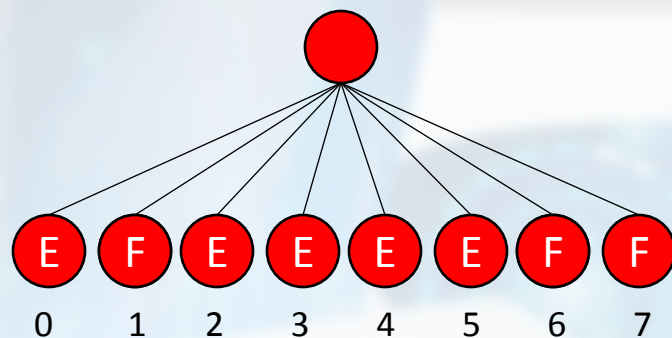
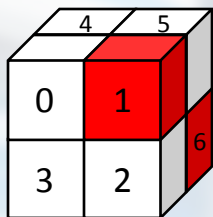
并运算



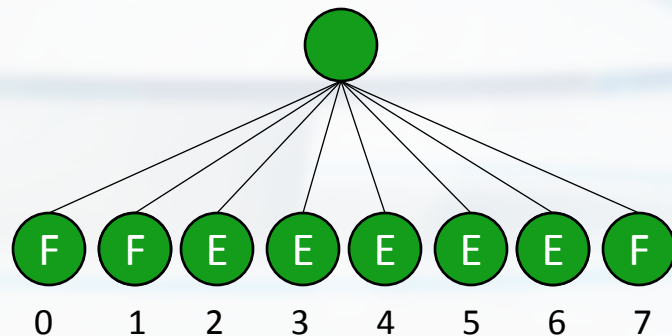
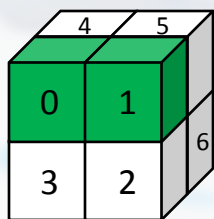
2 八叉树

基于八叉树的集合运算：

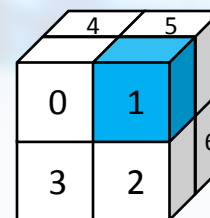
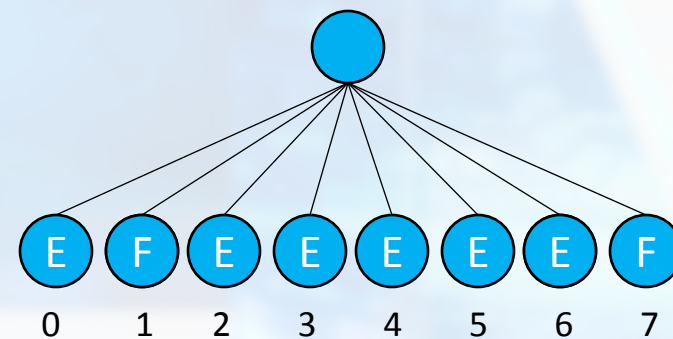
形体A



形体B



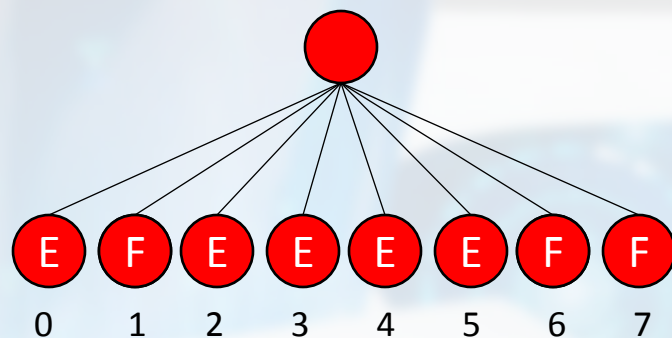
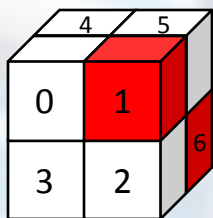
交运算



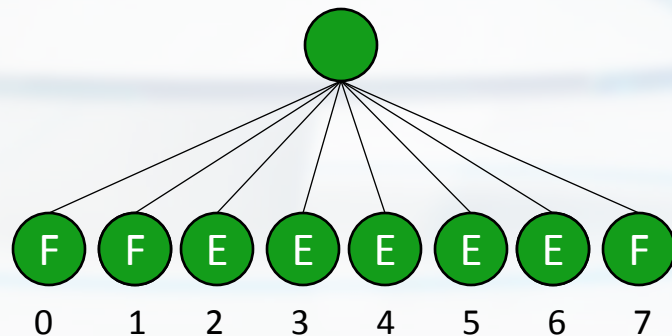
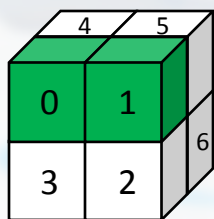
2 八叉树

基于八叉树的集合运算：

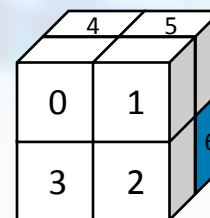
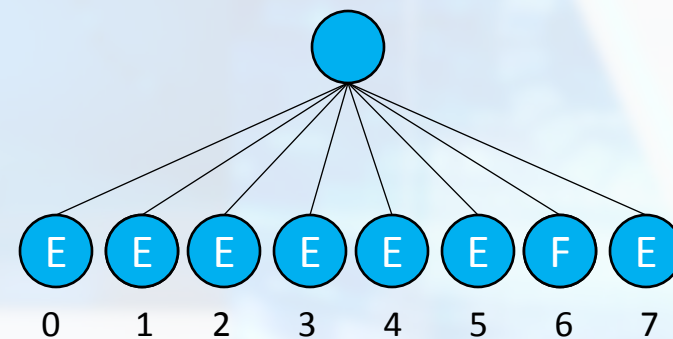
形体A



形体B



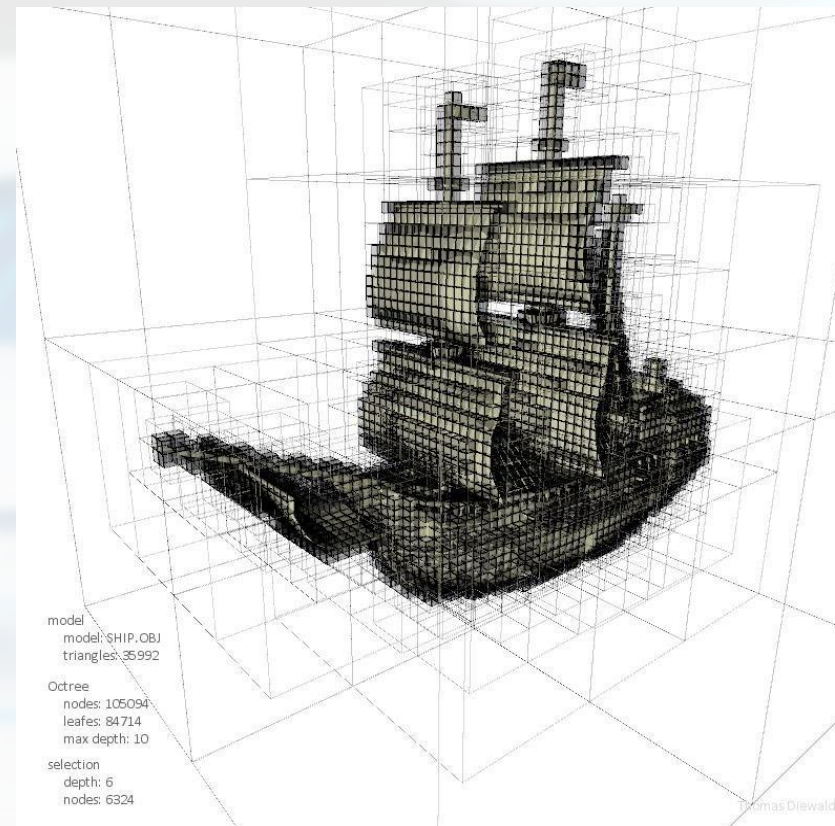
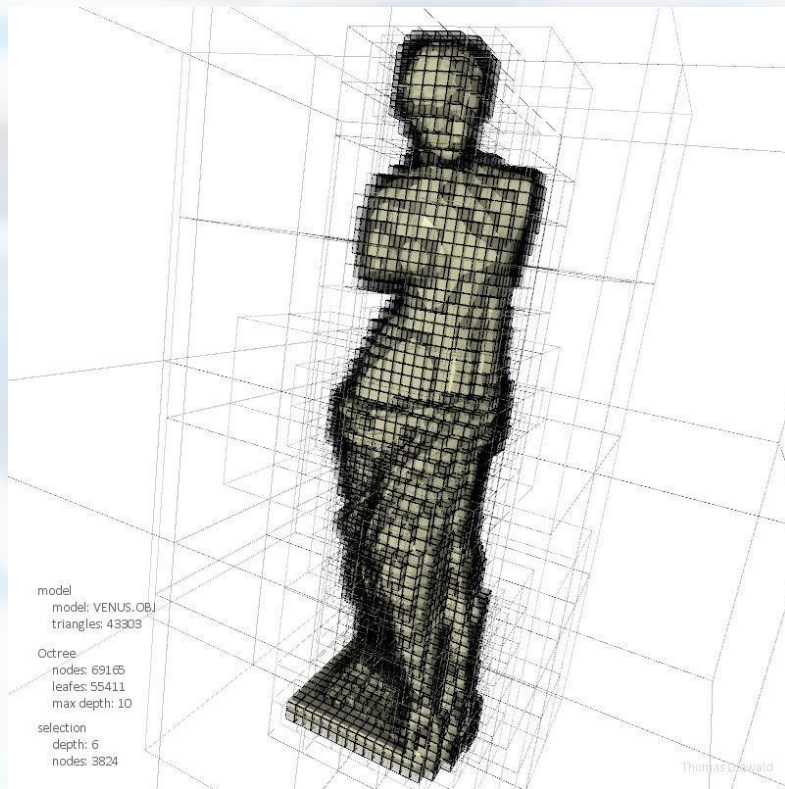
差运算
A-B



2

八叉树

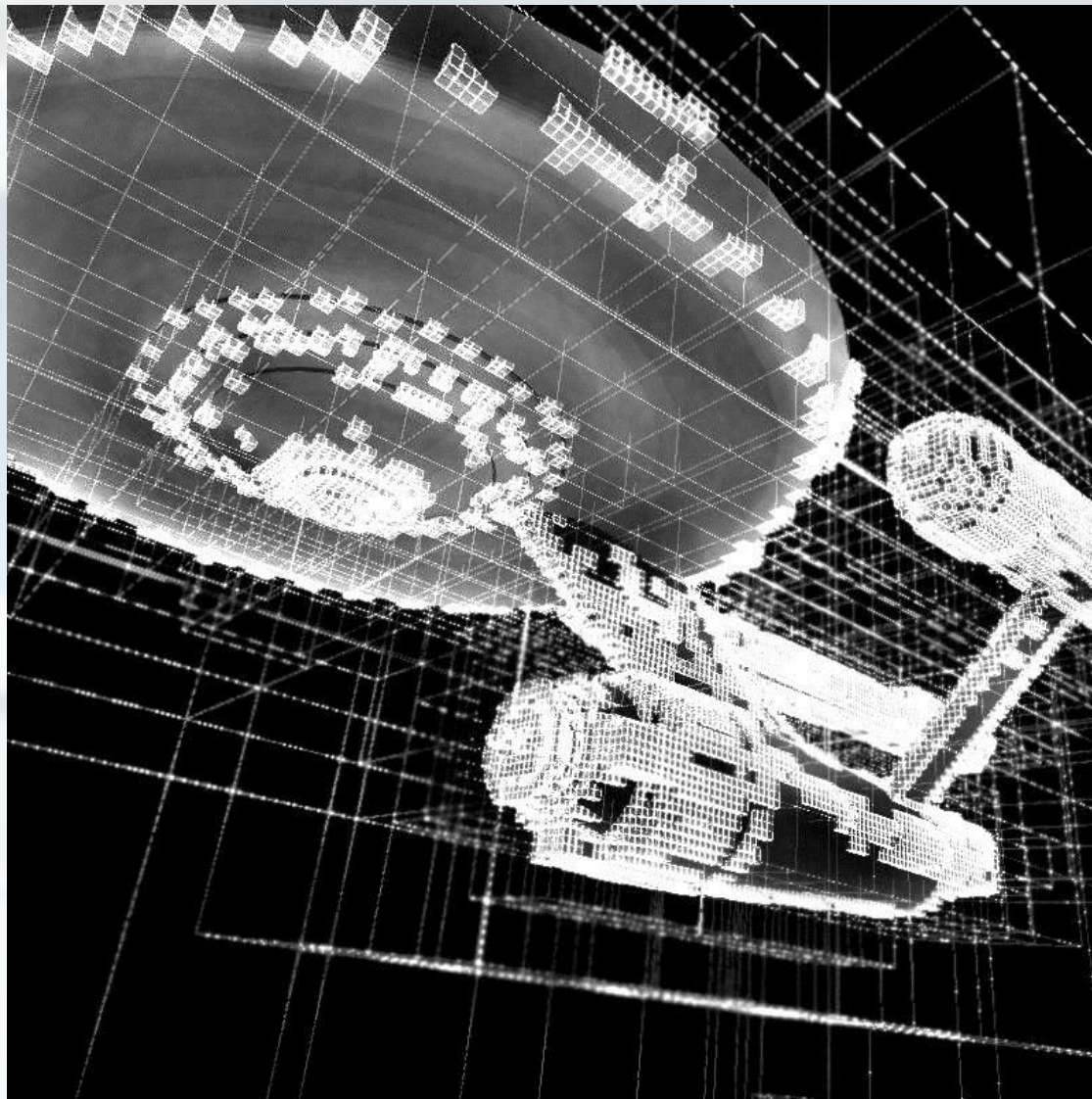
八叉树实例



2

八叉树

八叉树实例



2

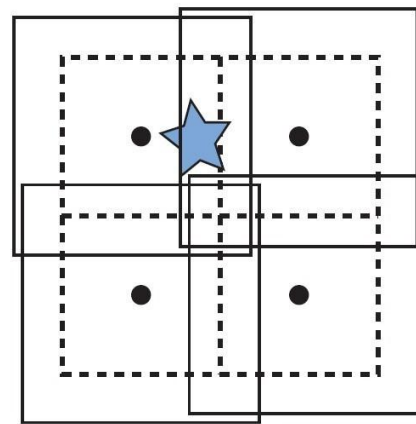
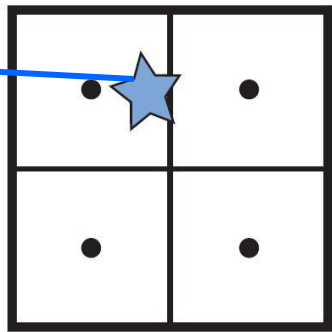
八叉树

松散八叉树思想：

松散八叉树的基本思想和普通八叉树一样，但是每个长方体的大小选中比较宽松。

如果一个普通长方体的边长为 l ，那么可以用 kl 来代替，其中 $k > 1$ 。

星形物体刺穿了一个普通八叉树的一个分割平面



$k=1.5$

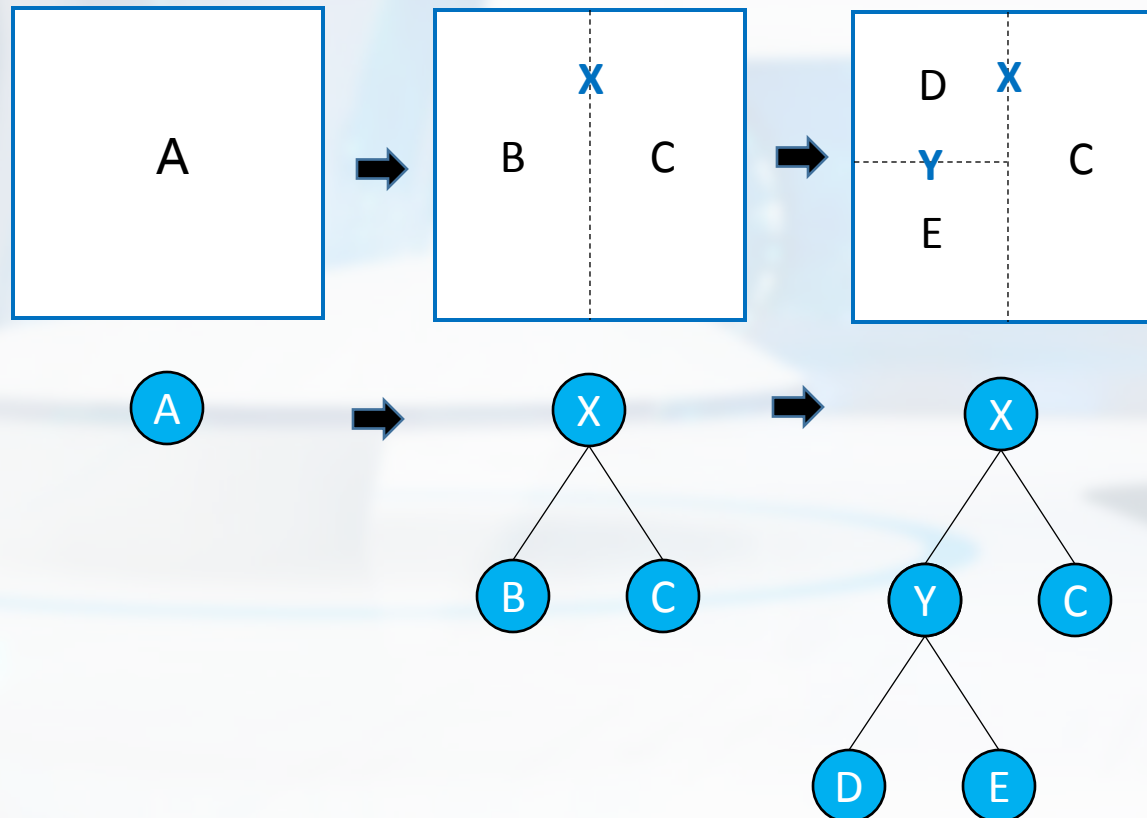
黑色的原点表示长方形的中心点（第一次细分）。

右图所示为一个 $k=1.5$ 的松散八叉树，也就是将长方体放大了 50%，如果将这些长方体稍微移动，就可以保证区分出它们。这样，这个星型多边形就完全位于左上角的长方形之中。

3

BSP树

二叉空间分割 (Binary Space Partitioning , BSP) 方法每次将一实体用任一位置和任一方向的平面分为二部分。

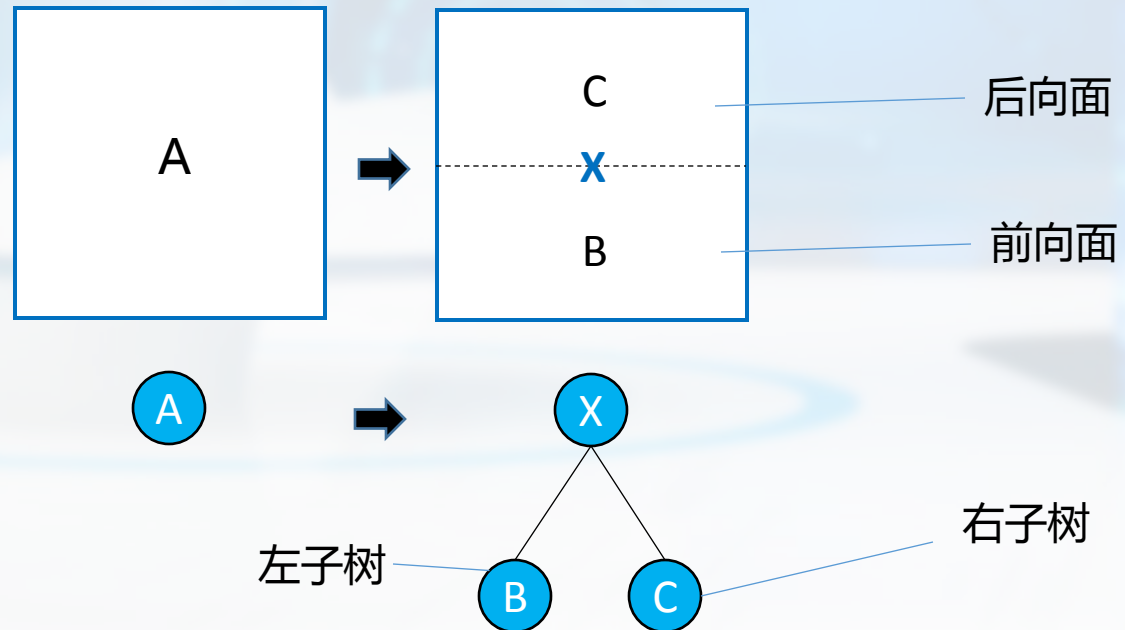


3

BSP树

与八叉树比较

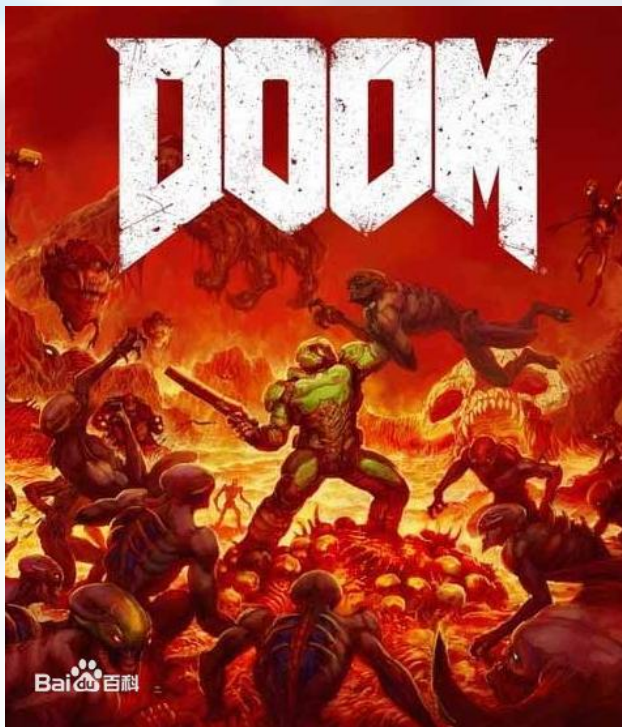
- ◆ 自适应分割：BSP树可以减少场景树的深度，减少搜索时间；
- ◆ 有向超平面：有效地识别前向面和后向面。



3

BSP树

最早应用BSP树的商业游戏《DOOM》

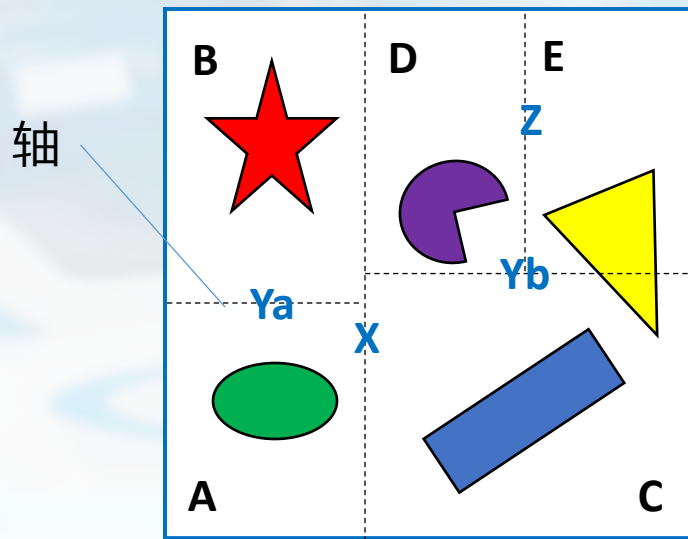


3

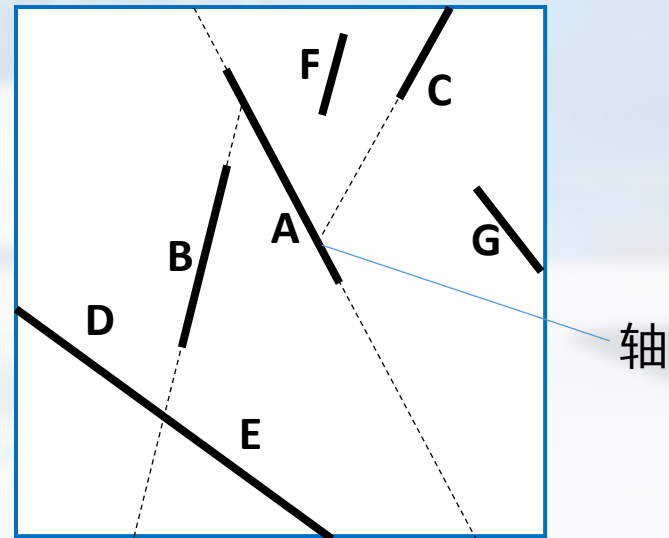
BSP树

两个类别：

- ◆轴对齐的BSP树
- ◆多边形对齐的BSP树



轴对齐的BSP树



多边形对齐的BSP树

3

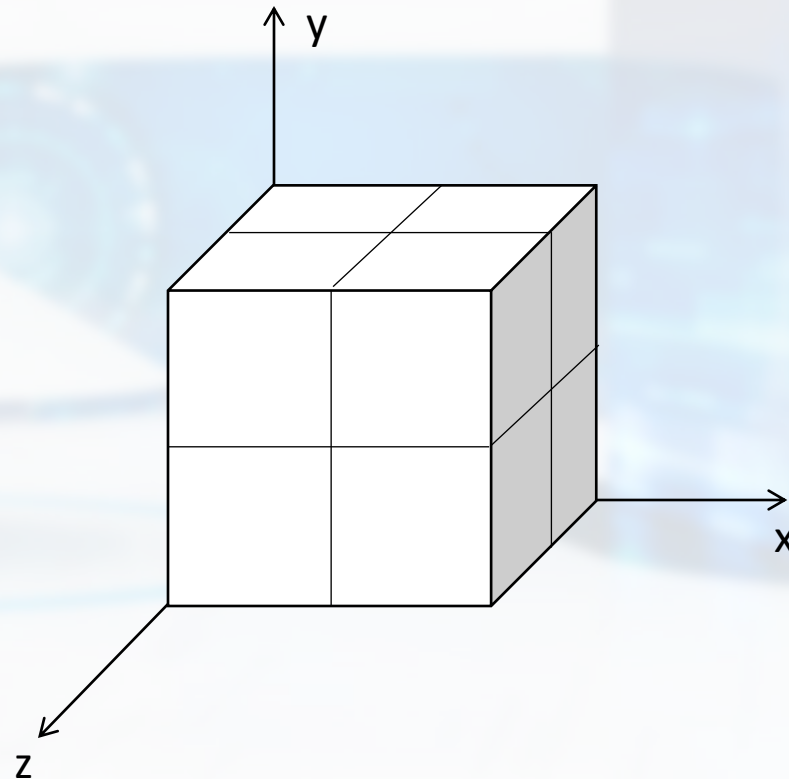
BSP树

◆轴对齐的BSP树

创建轴对齐BSP树的过程

step1 : 将整个场景包围在一个AABB (轴对齐包围盒, Axis-Aligned Bounding Box) 中

step2 : 以递归的方式将这个包围盒分为若干更小的盒子



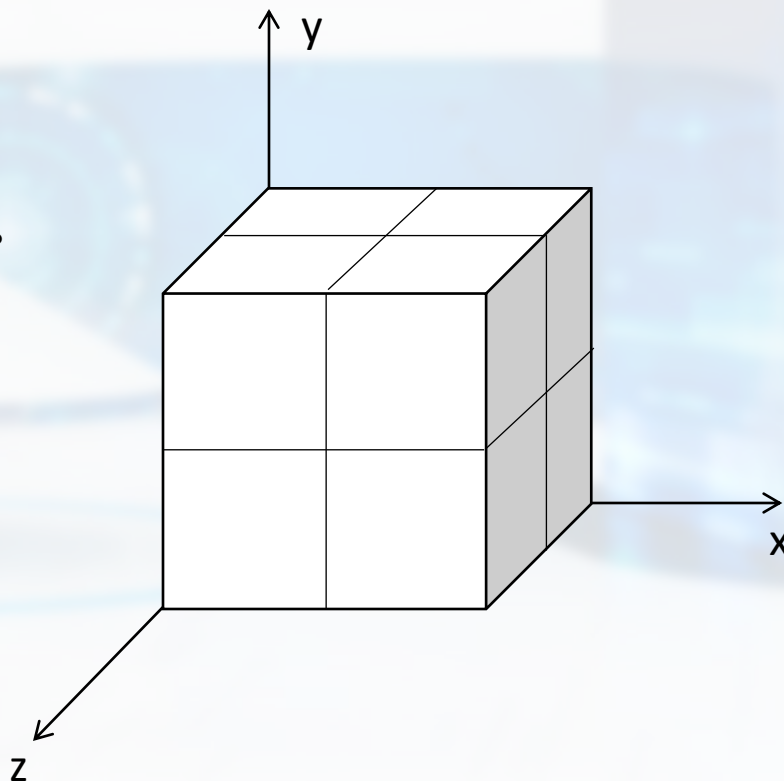
3

BSP树

◆轴对齐的BSP树

分割包围盒的方法：

方法一：轴进行循环，即在根节点沿着x轴对盒子进行分割，然后沿y轴对其子盒子进行分割，最后沿z轴对其孙盒子进行分割。



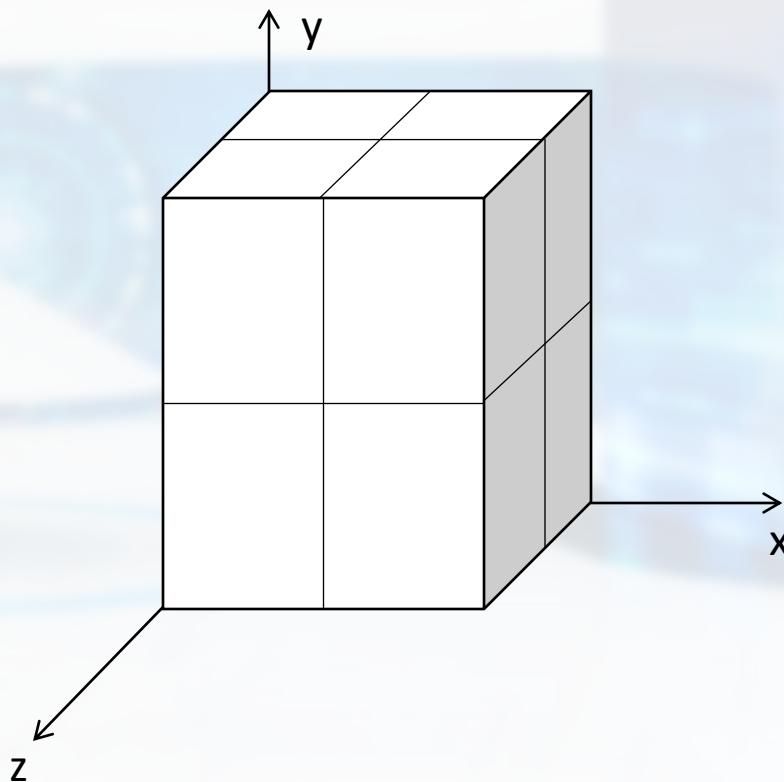
3

BSP树

轴对齐的BSP树

分割包围盒的方法：

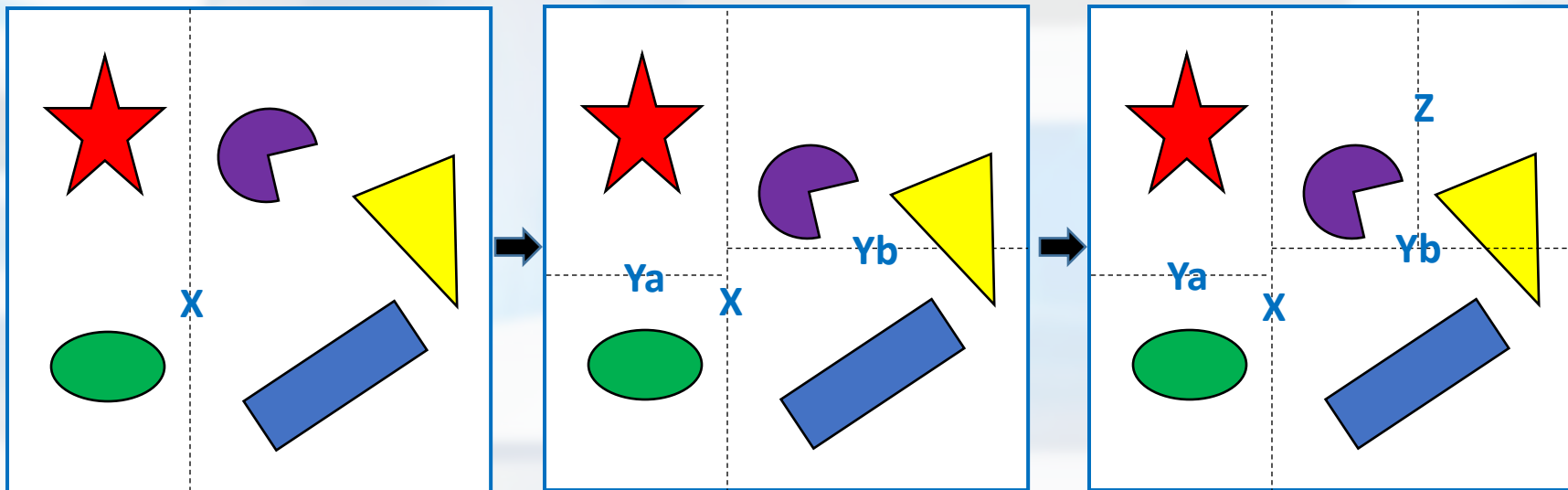
方法二：找到盒子的最长边，沿着这条边的方向对盒子进行分割。



3

BSP树

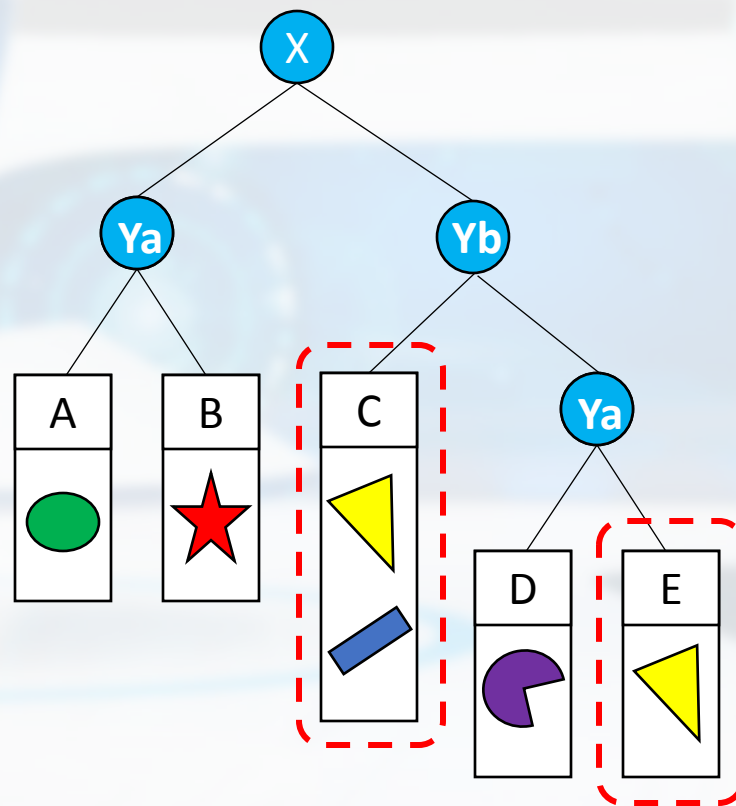
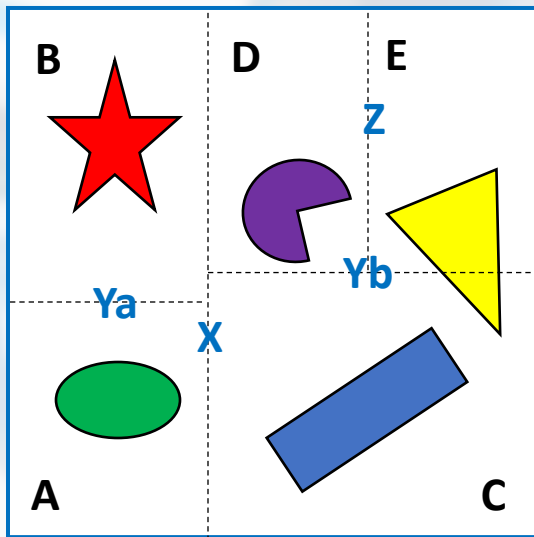
轴对齐的BSP树



3

BSP树

轴对齐的BSP树



3

BSP树

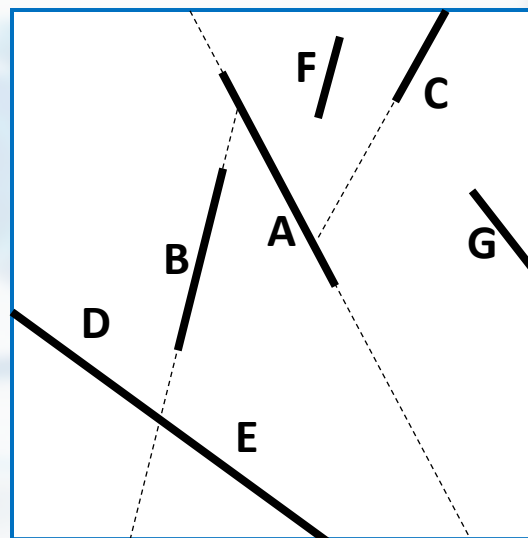
◆多边形对齐的BSP树

创建轴对齐BSP树的过程

Step1：在根节点处，选取一个多边形，
用这个多边形所在的平面将场景中剩余的
多边形分为两组

step2：以递归的方式将所有的多边形放入
对应的BSP树中

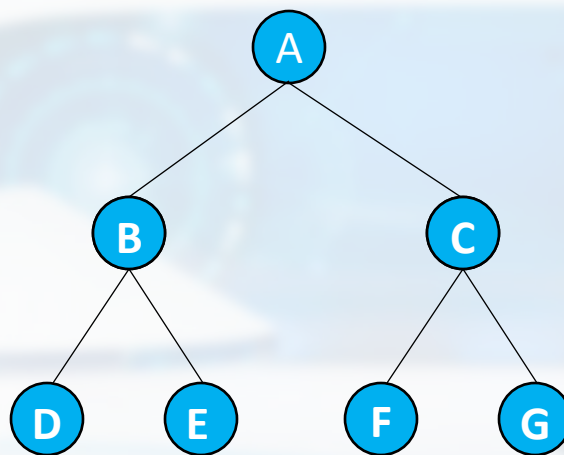
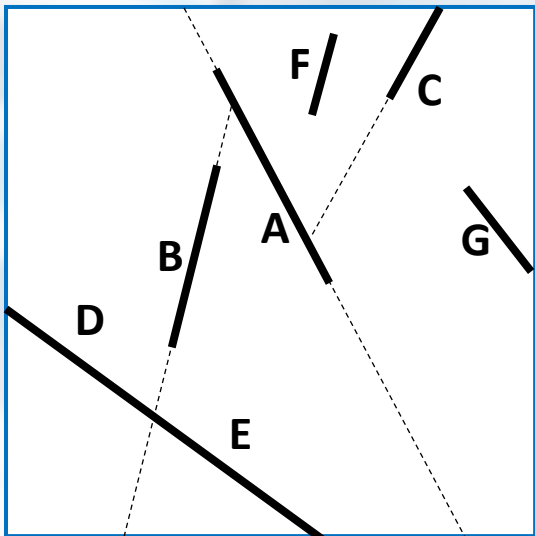
说明：一个场景的BSP树只需要计算一次



3

BSP树

多边形对齐的BSP树

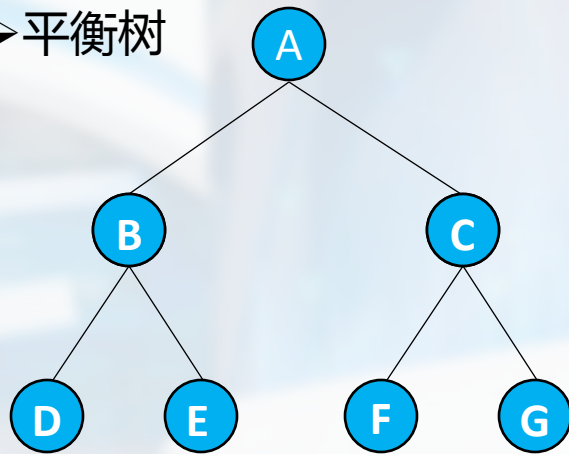


3

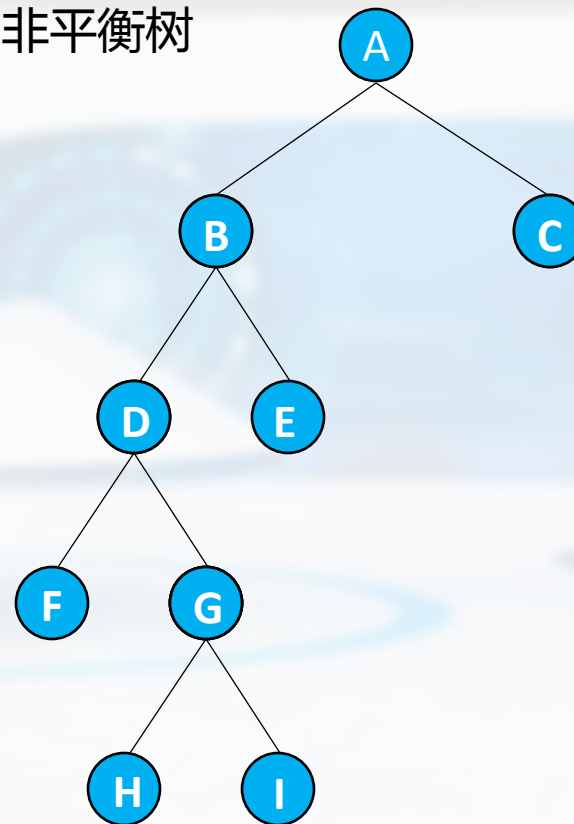
BSP树

平衡树和非平衡树

➤平衡树



➤非平衡树

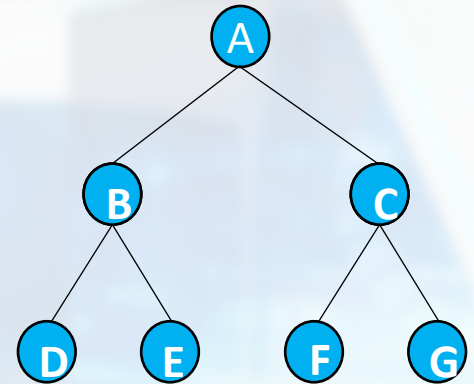
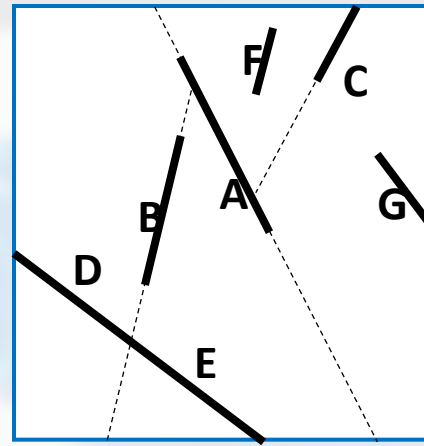


3

BSP树

多边形对齐的BSP树的应用：

- ◆深度测试：轴对齐只能对应粗排序，多边形对齐对应相对视点的排序，可以与之后的画家算法配合进行深度测试
- ◆相交测试
- ◆碰撞检测





谢谢

软件学院 万琳