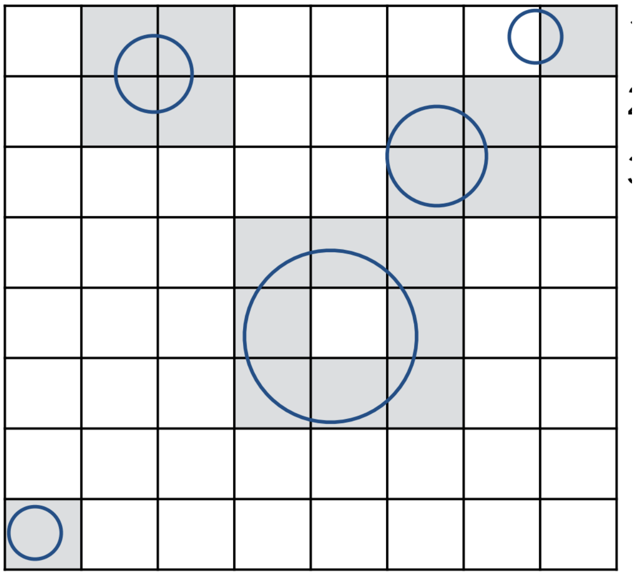
Lecture 14

Uniform Spatial Partitions (Grids)

Preprocess

- 找到 bounding box
- 创建 **网格**
- 存储物体与网格相覆盖的点

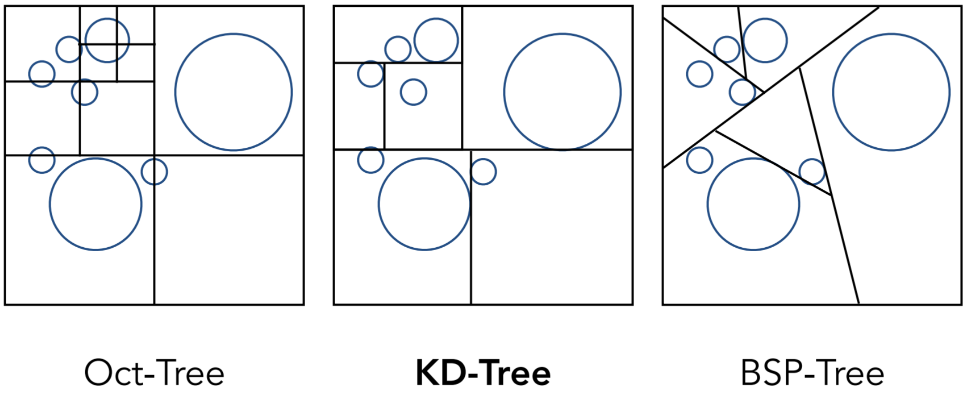


- **Ray-Intersection**
  光线与物体求交
- 遍历光线经历的每个 **网格**
- 如果含有物体，则与网格中含有的所有物体求交。

Grid Resolution 网格的分辨率（密度）会影响求交次数

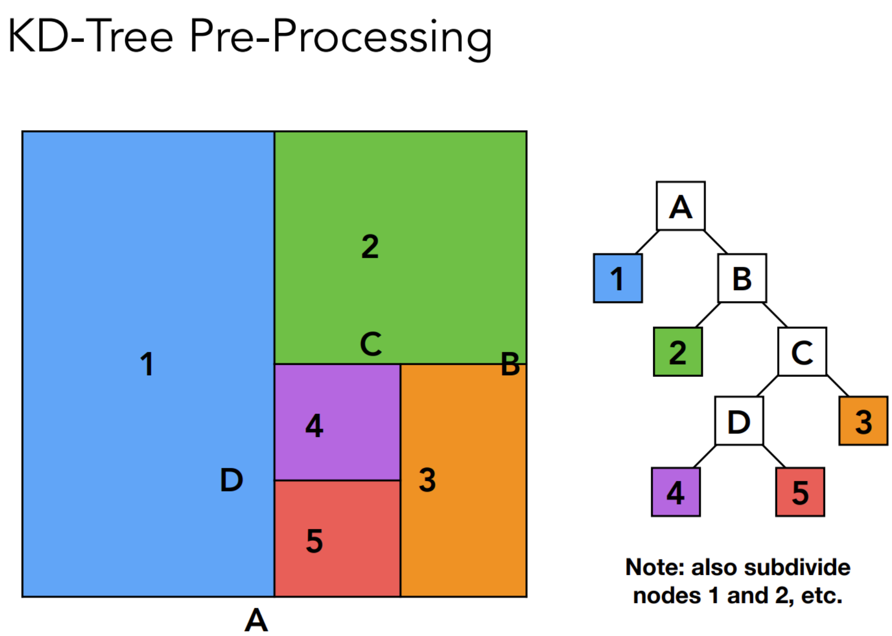
- **过少**：不会起加速作用
- **过多**：太多空白网格，造成浪费。
- 普遍来说：$cells = C * objs (C = 27\ in\ 3D)$

Spatial partition 空间划分



- Oct-Tree 八叉树：每次将空间划分为八个子空间（在三维中），在二维中自然每次划分成 **四个子空间** ，按一定规则收敛，**遇到已划分的格子中没有物体/物体较少** 停止划分
- KD-Tree 每次在坐标轴一个方向切一刀，在 x/y/z 轴三个方向上交替划分，以保证空间划分较为均匀
- BSP-Tree 较KD-Tree，切一刀的方向不限制在坐标轴方向，能将物体划为两份的任意方向即可。

KD-Tree



KD-Tree 节点

- 中间节点：记录 **划分位置，划分方向，子节点**
- 叶子节点：保存物体。

光线求交

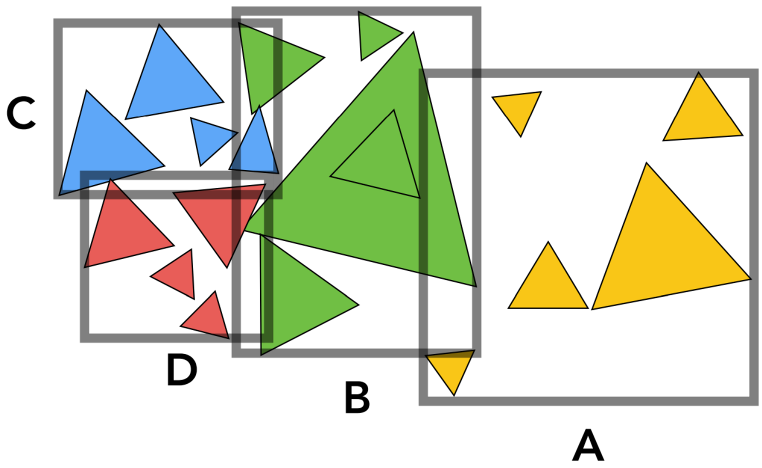
- 如果光线与 **叶子节点** 相交，则将光线与叶子节点中全部物体求交
- 如果光线与 **中间节点** 相交，将光线与该节点的子节点求交（递归）

KD-Tree的问题

- 不容易判断三角形与网格是否相交 *?*
- 一个物体存在于多个网格中，在不同网格中都会进行光线与该物体的求交，会导致计算浪费。

Bouding Volume Hierarchy (BVH)

相较于Kd-Tree 以空间将物体划分，BVH每次将物体分成两部分，不在意空间是否重复



具体流程 BVH 选定一个轴（x/y/z），一般是整个空间内物体占据最长距离的轴），按照所有物体的中位数进行划分。

- 寻找 bounding box
- 将集合分成两个子集
- 重新计算子集的 bounding box
- 回到第二步（如果满足条件，则停止划分）
- 在每个叶子节点中存储物体

- Heuristic #1: Always choose the longest axis in node
- Heuristic #2: Split node at location of **median** object

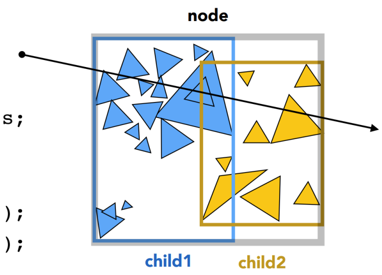
BVH中光线求交 伪代码如下

```
Intersect(Ray ray, BVH node) {
  if (ray misses node.bbox) return;

  if (node is a leaf node)
     test intersection with all objs;
     return closest intersection;

  hit1 = Intersect(ray, node.child1);
  hit2 = Intersect(ray, node.child2);

  return the closer of hit1, hit2;
}
```



Spatial vs Object

- 前者空间不重叠，后者物体不会划分到多个子节点中。

SAH(表面积启发式积分) BVH 中划分一共分三种策略

- 取**中点划分**
- 按**等量划分**
- **启发式划分** SAH

对于 **中点划分**，其步骤如下

- 选取最长轴两端物体的重心，并做线段取其中点，此处一分为二进行划分
- 问题：遇到物体 **分布不均匀** 的情况，划分效果很差。遇到包围盒 **重叠过多** 的请况，会导致光线常常对两棵子树求交，划分基本没有效果。

对于 **等量划分**，其步骤如下

- 选取最长轴能将左右分成均等个数的划分方式，此处一分为二。
- 问题：对于中点划分中分布不均匀的问题有所改善，但对于 **重叠** 问题，并不能起到作用

SAH

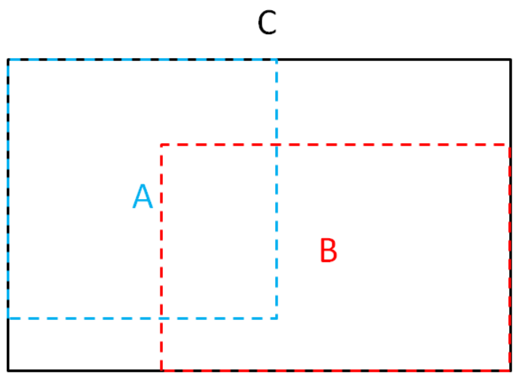
假设场景中有 n 个物体，此时不做划分，则光线需要对全部物体做求交，花费时间为 $t_1 + t_2 + ... + t_n$ ，我们此处假设每个物体性质一样，即求

交时间相同，此时花费时间为 n 。 再假设场景分为两块，A/B ，并且设定光线与 A 相交的概率为 $P(A)$ ， 同理对 B 为 $P(B)$,此时花费时间为,也可以看做花费时间的 **数学期望**

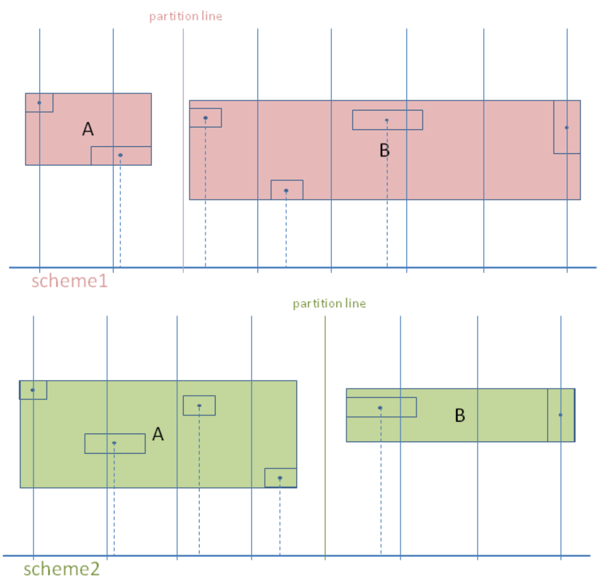
$$t_{all} = p(A) * (n\ in\ A) + p(B) * (n\ in\ B)$$

SAH 规定,一个子包围盒在其父包围盒被击中的前提下,被光线击中的概率为 $S_{子}/S_{父}$

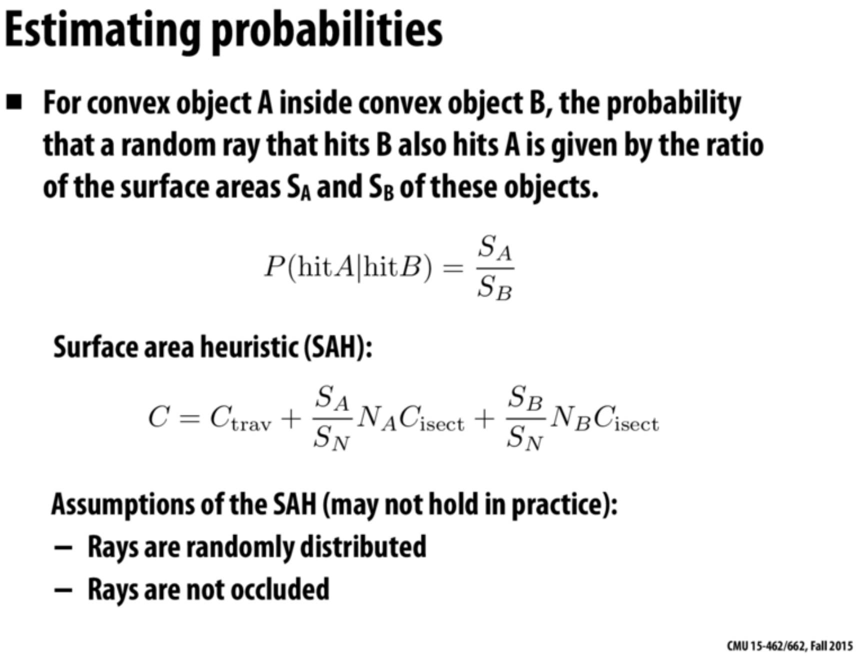
而两个区域 A/B 被击中概率之和为 $S(A)/S(C) + S(B)/S(C)$,此处注意:**包围盒会发生重叠,导致其结果大于S(C)** ,如图



SAH 在此之上,通过对长轴切分后花费的计算,遍历出一个最优切分结果,如下图有两种不同的切分方式

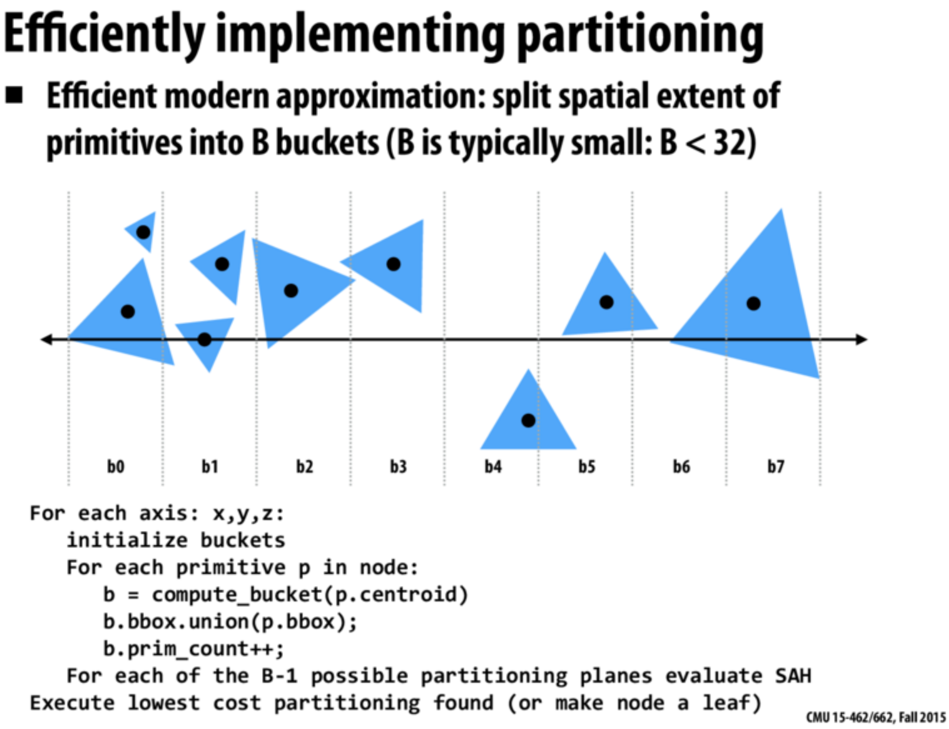


我们设定遍历所有包围盒的代价为 C_{trav} ,可得出 SAH 的花费公式



SAH 与 buckets

上文中 SAH 遍历每个物体进行划分,更好的方式是将节点的空间沿着跨度最长的坐标轴先 **划分为若干个桶(bucket)** , 划分只会出现在桶与桶之间的位置上,如图所示



根据物体个数指定 **桶的个数与位置**, 假设桶个数为 B ,所有的划分只存在 $B-1$ 种情况,会减少花费 **住:GAMES101作业中,我采用bucket划分的SAH反而更慢了...**

Basic radiometry 辐射度量学

基于 Blinn-Phong 模型得出的图像不够真实，在物理上也不算正确，需要重新考虑光线传播的过程。以下是几个基本概念

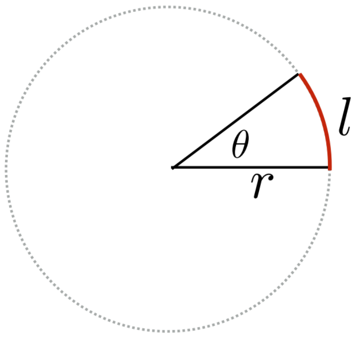
Radiant Energy Q 能量，指电磁辐射的总能量 单位：J

Radiant Flux ϕ 单位时间内的能量 单位：$Watt/lumen$

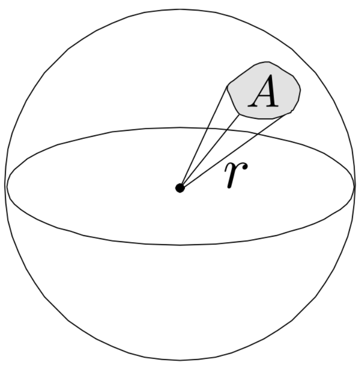
Radiant Intensity I 单位时间通过单位 **立体角** 的功率，表示为特定方向上的辐射通量

$$I(\omega) = \frac{d\phi}{d\omega}$$

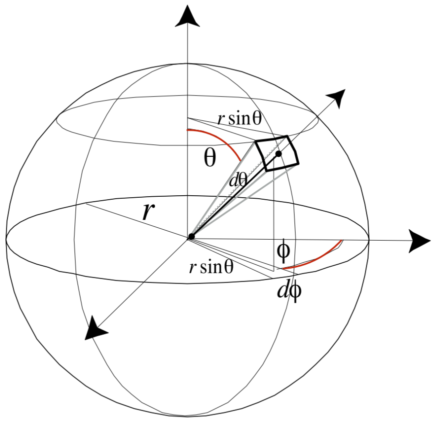
立体角 Solid angle 立体角是二维圆中弧度角在三维空间中的延申。二维中 $\theta = \frac{l}{r}$ ，圆的最大角度为 2π



三维中 $\Omega = \frac{A}{r^2}$ ，球的最大立体角为 4π



单位立体角 单位立体角即球面上单位面积对应的立体角

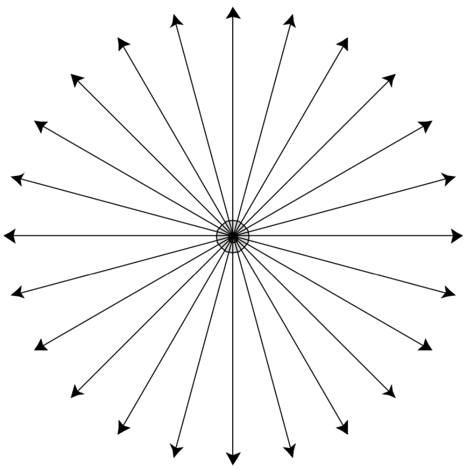


$$\begin{aligned} \mathrm{d}A &= (r\,\mathrm{d}\theta)(r\sin\theta\,\mathrm{d}\phi) \\ &= r^2\sin\theta\,\mathrm{d}\theta\,\mathrm{d}\phi \end{aligned}$$

$$\mathrm{d}\omega = \frac{\mathrm{d}A}{r^2} = \sin\theta\,\mathrm{d}\theta\,\mathrm{d}\phi$$

并且可对单位立体角对 θ 和 ϕ 两个角度进行积分，最后可得出 4π ，即球的表面积。

辐射通量 已知 Intensity 是单位立体角的辐射通量，则通过对单位立体角的积分可求出总的辐射通量



$$\begin{aligned} \Phi &= \int_{S^2} I\,\mathrm{d}\omega \\ &= 4\pi I \end{aligned}$$

$$I = \frac{\Phi}{4\pi}$$

reference

silde BVH with SAH (Bounding Volume Hierarchy with Surface Area Heuristic) Games101：作业6解析（含提高部分SAH