

The Next Week

1. Overview

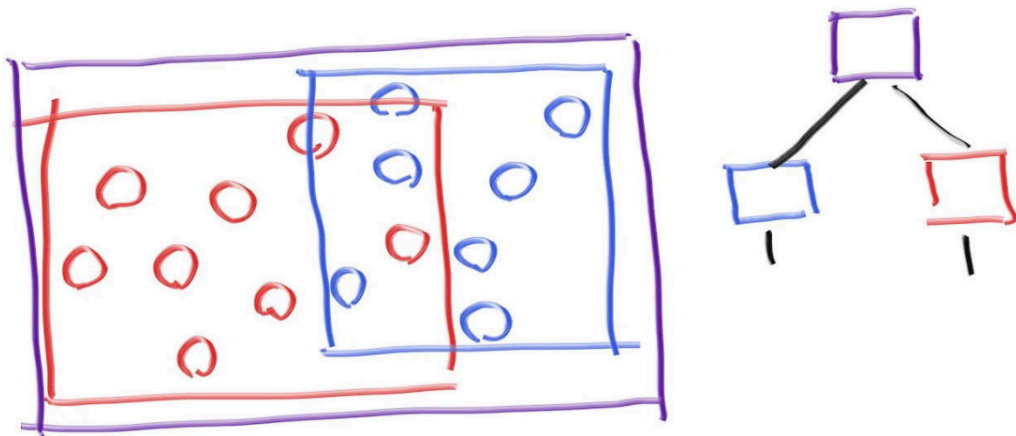
- 目标:
添加 texture, volume, rectangle, instance, light, BVH
- 难点:
BVH, Perlin textures

2. Motion Blur

- 现实中动态模糊是镜头打开持续曝光得到的
- SpaceTime Ray Tracing
将ray在镜头打开时间段内随机生成, ray记录时间
- camera 记录镜头打开和关闭时间
- 让球在镜头打开时移动
为运动球增加了一个类, 与静止球相比多了起点和终点位置 (也可以以用此类表示静止球, 只要让起点终点位置相同, 但会影响性能)
- 反射出的 ray 时间与 入射 ray 相同

3. BVH

- 用于加速 ray tracing
- 光追的主要瓶颈是 **ray-object intersection** 检测 (目前的暴力做法中, 每条直线都需要和每个物体判断是否相交, 很多判断可以避免)
- 需要将物体整理, 每次检测只判断部分物体
- 常用整理方法有两种:
 1. 按空间划分
 2. 按物体划分 (更常用)
- Key Idea:
将一堆物体包括在一个大的包围盒内, 先判断光线是否与包围盒碰撞, 之后再与其中的物体判断
 - 不是均匀划分空间, 每个物体只在一个包围体内, 但包围体之间可重叠



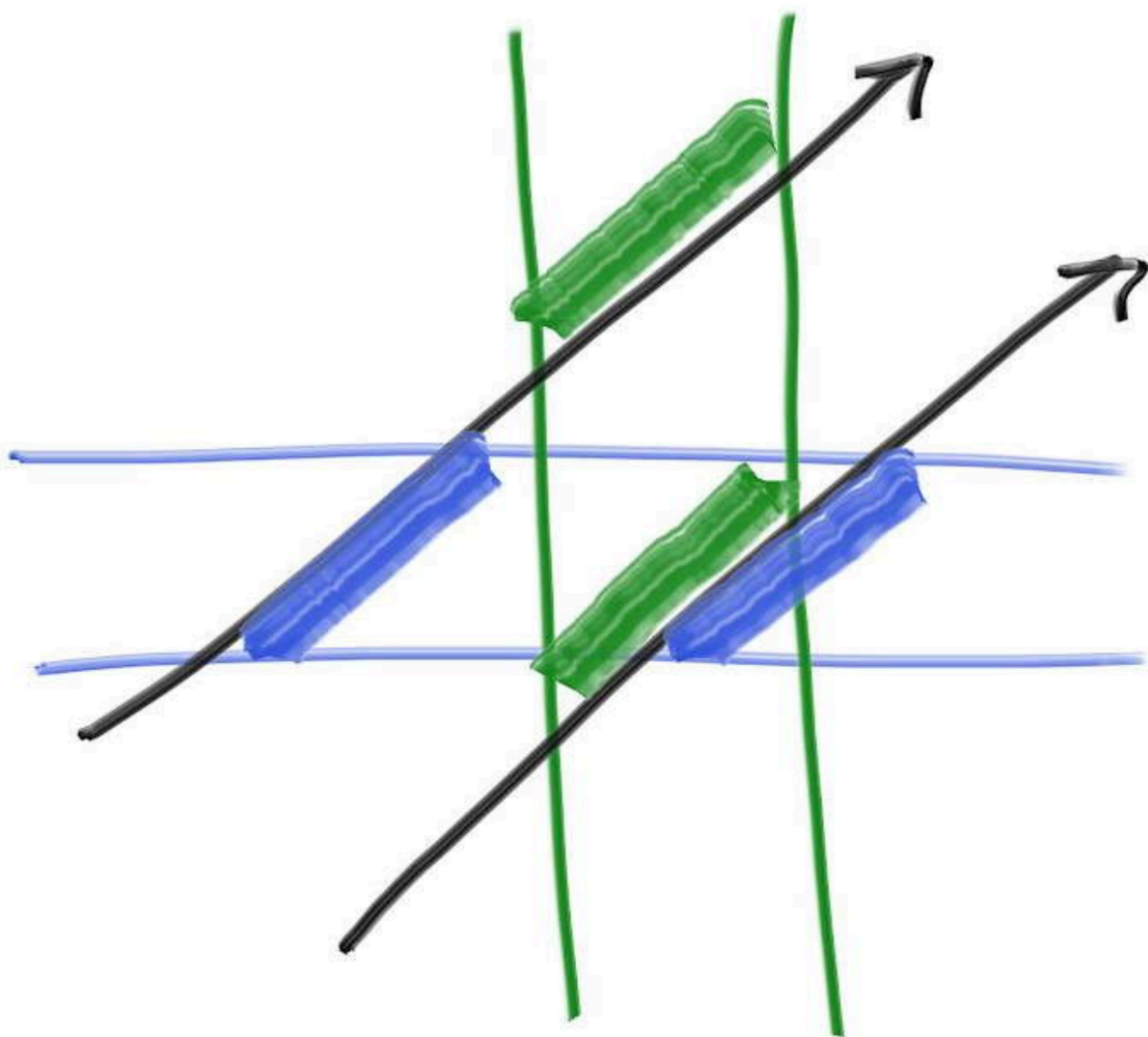
- 轴对齐包围盒(AABB)

判断与包围盒的碰撞时不需要记录碰撞点或法向量

- 如何判断 ray 与 box 是否相交: ray slab

将box各个轴向的平面方程代入ray方程, 找到解区间, 相交的充要条件是:

各个轴区间重合



上图是2d box 的例子, 3d情况下同样适用

$$ray: P(t) = A + tb$$

$$plane: x = x_0$$

$$x_0 = A_x + t_0 b_x$$

$$hit: t_0 = \frac{x_0 - A_x}{b_x}$$

- ray slab 还有缺陷:
 1. 对于沿轴反方向传播的ray, 会出现大小倒置的区间
 2. 计算中涉及除法, 可能得到无穷值
- 做特殊处理:
 1. 统一区间方向:

$$t_{x0} = \min\left(\frac{x_0 - A_x}{b_x}, \frac{x_1 - A_x}{b_x}\right)$$

$$t_{x1} = \max\left(\frac{x_0 - A_x}{b_x}, \frac{x_1 - A_x}{b_x}\right)$$

2. 除 0 问题之后解决 (填充?)

- 构造 包围盒

计算所有物体的包围盒, 并在其上建立树形结构

 1. 构造单个几何体包围盒

在 几何体基类 中添加相应函数
 2. 对 list 构建包围盒
- 在 包围盒基础上 构建 node, node用于组成树结构


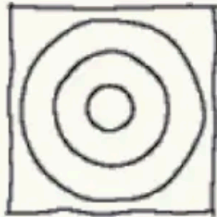
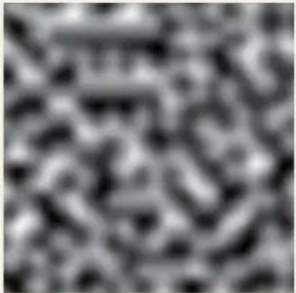
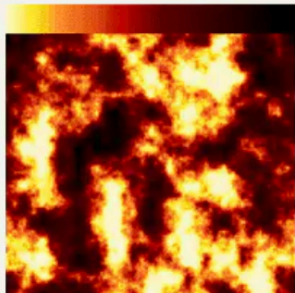
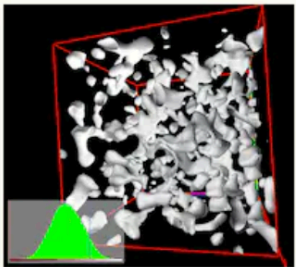

node 也 属于 hittable 物体,
- 构建 BVH 过程:
 1. 先整个场景一个完整的包围框
 2. 迭代分解:
 1. 随机选择一个轴,
 2. 将物体沿该轴排序
 3. 按顺序分为两半
 - 不要判断空指针 (影响性能), 当框里只有一个物体时, 就
- 并不是所有物体都有包围盒 (无限平面), 如果要加这些物体需要特殊处理

4. 静态纹理

- 单色纹理
要在 hit_record 中保存光线交点处的纹理坐标
- 球体的纹理坐标
在球坐标系下做纹理映射
- 在漫反射材质中用纹理替代颜色
- 构建checkered纹理（正弦函数）

5. 柏林噪声(Perlin Noise)

用于生成随机内容，可用于生成纹理

噪声维度	原始噪声（灰阶）	游戏应用
1		 使用噪声进行偏移，制造手绘效果
2		 制造火焰燃烧特效
3		 最普遍的运用莫过于生成游戏里的随机地形

- 特点：
 - 可重复性，给定种子生成的序列是一定的
 - 快
- 算法（2D）：
 1. 定义一个 grid, 在 grid 的每个顶点生成随机的梯度向量

2. 对输入图中的每个点，找到其所处的格内的四个顶点，获取其到各点的距离向量，再分别点乘
 3. 将点乘结果用缓和曲线加权相加得到最终结果
- 3D算法同理
 - 得到的图像可用线性插值平滑
 - 线性插值的结果会有网格感，可用 Hermitian 插值
 - 可以用 scale 的方法调整图像频率
 - 复合的噪声会更加复杂，可通过多次perlin实现
 - “纹理过程”的 hello world：大理石纹理：
在perlin噪声的基础上添加正弦规律

6. 纹理映射

- 由图片生成纹理

7. 光照

- 自发光材质
自带发光颜色，不反射
- 添加光照后，背景置为纯黑
- 添加矩形物体：
矩形没有厚度，在构造包围盒前要做padding
- 光源的亮度设为比1大的值，可以使光线在反射后仍足够亮，从而照亮场景
- 构造Cornell Box
用于模拟漫反射表面与光的交互

8. Instance

- 添加轴对齐的 box, 之后再rotate
- 光线追踪渲染中的旋转位移等操作，可以不用直接操作物体，而是反向操作光线实现

9. Volumes

烟雾

- 需要新添加次表面散射，即从烟雾内部进行光线散射
- 烟雾可以简化表示为 a surface that probabilistically might or might not be there at every point in the volume.
- 光线在volume中任意位置都有一定概率反射：

$$probability = C \cdot \Delta L$$

- C 表示volume 密度， δL 是光线在volumn中移动的距离
- todo: 处理凹多面体

- 烟雾可用于模拟丁达尔效应

10. final

- 次表面材质：通过在介质中填充 volume 实现