

计算机图形学

第九次实验报告

Path tracing

PB20000264

韩昊羽

一．实验要求

- 阅读框架，完成 path tracing 基本算法（直接光，间接光）
- 阅读 alias 相关知识，完成环境光重要性采样
- 自己构建新的场景
- 修改参数观察变化

二．操作环境

IDE: Microsoft Visual Studio 2019 community

CMake: 3.23.1

三．功能实现

3.1 实验原理 path tracing

渲染方程为：

$$L_o(p, w_o) = L_e(p, w_o) + \int_{\mathcal{H}^2(n(p))} f_r(p, w_i, w_o) L_i(p, w_i) \cos\theta_{w_i, n(p)} d\omega_i$$

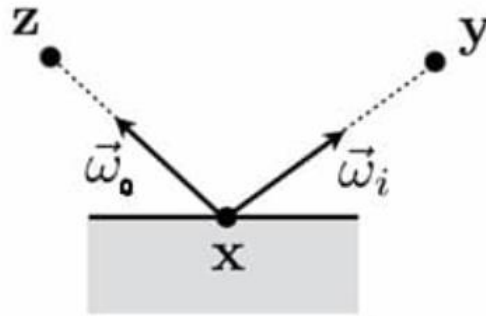
拆分直接光和间接光，并对 L_r 进行递归得到

$$L_r(p, w_o) = \int_{p, w_i, w_o} L_e(p', -w_i) + \int_{p, w_i, w_o} L_r(p', -w_i)$$

但这样需要处理积分，于是我们采用蒙特卡隆采样法，即为了求解积分，我们构造一个随机变量让其期望和积分值相同，最后对随机变量进行多次采样即可做到。

计算直接光 $L_{dir}(p, w_o) = \int_{p, w_i, w_o} L_e(p', -w_i)$ 的时候采用对光源采样的方法计算积分，对于采样后的光源 sample_light,

$$L_{dir}(p, w_o) = \sum_{i=1}^{N_e} L_i * f_r * \cos\theta * \frac{\cos\theta'}{l^2 * pdf}$$



其中， N_e 为采样的光源数量， L_i 为对应的光源强度， f_r 是 BRDF 对应参数值， θ 为 w_i 和物体法向的夹角， θ' 为 $-w_i$ 和光源法向的夹角（因为方向相反所以需要负号）， l 为 x 到 y 距离， pdf 为采样概率。

对于间接光 $L_{indir}(p, w_o) = \int_{p, w_i, w_o} L_r(p, -w_i)$ 需要迭代，对方向 w_i 进行采样

$$L_{indir}(p, w_o) = L_{new} * f_r * \frac{\cos\theta}{pdf} / pdf_RR$$

其中， L_{new} 是迭代的时候 w_i 方向碰到的第一个物体的光照强度值，通过迭代实现， f_r 是 BRDF 对应参数值， θ 为 w_i 和物体法向的夹角， pdf 为采样概率。 pdf_RR 称为 Russian roulette，即设置存活概率，只要存活才能继续迭代，否则迭代终止。

最后， $L_r(p, w_o) = L_{dir}(p, w_o) + L_{indir}(p, w_o)$

整体来看，path tracing 的算法为

```
PathTracer()::Trace(p, wo)
{
    // direct illumination
    for all lights (area lights, environment lights)
        Randomly sample the light x' with probability pdf_light;
        Intersect the ray(p, x'-p) with the scene;
        if ray(p, x'-p) does not hit any object // x' is visible for p
            L_dir = L_l * f_r * cos θ * cos θ' * |x' - p|^2 / pdf_light;

    // Indirect illumination
    L_indir = 0;
    Test Russian Roulette with probability pdf_RR;
    Randomly sample the hemisphere toward wi with pdf_hemi;
    Intersect the ray r(p, wi) with the scene;
    if ray r hit a non-emitting object at q
        L_indir = Trace(q, -wi) * f_r * cos θ / pdf_hemi / pdf_RR;

    return L_dir + L_indir;
}
```

3.2 改进方法 环境光重要性采样

采用别名法，构建概率表 U 和别名表 K 从而在 $O(1)$ 时间内进行采样。构建概率表时，按照环境贴图对应像素点的光强来按比例计算概率。构建别名表时，先按照概率表的情况进行一次划分：

- 如果 $U_i = 1$ （浮点计算时要处理误差）设置 $K_i = i$ 。事实上此时 K_i 的值永远也不会用到

- 如果 $U_i < 1$ ，把 i 划分到缺少 (less) 的一组
- 否则，划分到多余 (more) 的一组

然后重复以下步骤：

- 从 more 和 less 中各抽出一个 (一定有多就有少)，设为 i, j 。
- 设置 $K_j = i$
- 更改 U_i 的值 $U_i = U_i + U_j - 1$
- 从 less 中剔除 j ，然后按情况分类 i

直到结束。

在采样的过程中，从 $[0, 1)$ 中随机采样 x ，令 $i = [nx] + 1$ ， $y = nx + 1 - i$ ，如果 $y < U_i$ ，返回 i 否则返回 K_i 。

3.3 实际操作

在处理 $\cos\theta$ 的时候要考虑是正面还是背面，因为正面 $\cos\theta$ 的值一定是正的，所以一律 abs 处理。

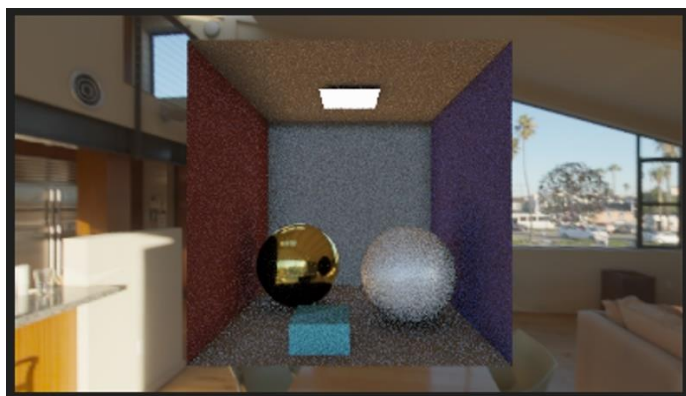
实际操作时，除了直接光和间接光，还需要单独处理表面绝对光滑的情形。这时直接衡量出射方向有没有光源就行

处理别名表的时候使用队列，同时因为采样函数是在 $[0, 1]$ 之间，同时下标为 $[0, n-1]$ 范围，所以实际上 i 不需要加一，反而需要检测是否超出下标。

四 . 难点难题

4.1 rayf3 参数问题

一开始在初始化 rayf3 向量的时候，我都使用了四个参数的构造方法，但发现这样环境光的渲染不对



发现金属球右上角也已经渲染上了颜色，但实际上应该只有中间有光，发现不能为环境光指定 $tmin$ 和 $tmax$ ，要不会计算得到和厚度为 0 的平边不相交，所以环境光需要使用两个参数的构造函数。

4.2 光线反向问题

如果直接对区域光检测是否碰到，会导致区域光后面的区域也被渲染上光，所以需要
对射线和法向的夹角进行检测。

同样的，在对间接光采样的时候也会出现 w_i 在背面的情况，需要调整 $w_i = w_i - 2 * (w_i \cdot n) * w_i$ 。

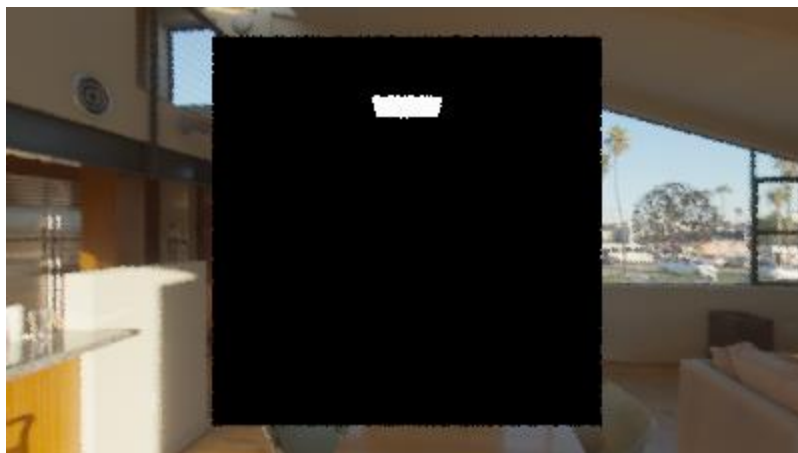
4.3 采样失败问题

在进行间接光采样的时候，有时候会采样失败传出 $w_i = 0$ 导致在 normalize 的时候报错。我还未找出 sample 函数的问题，只能对 w_i 进行一个检测，如果出现上述情况直接终止迭代。

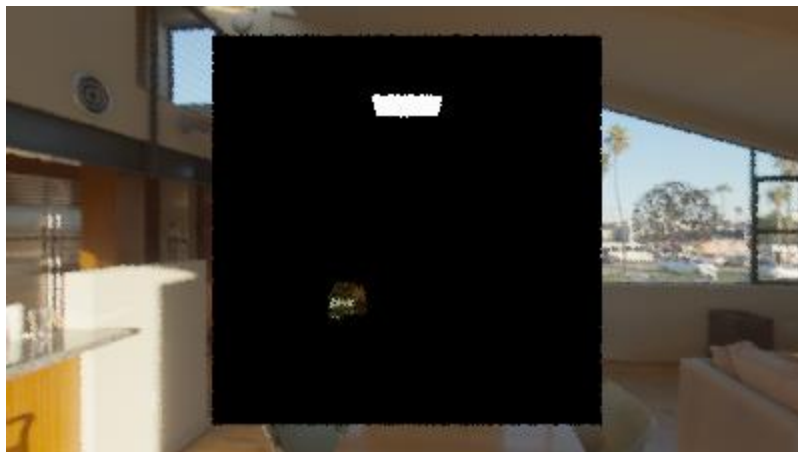
五 . 实验结果

比较：spp 为 2，width 为 400

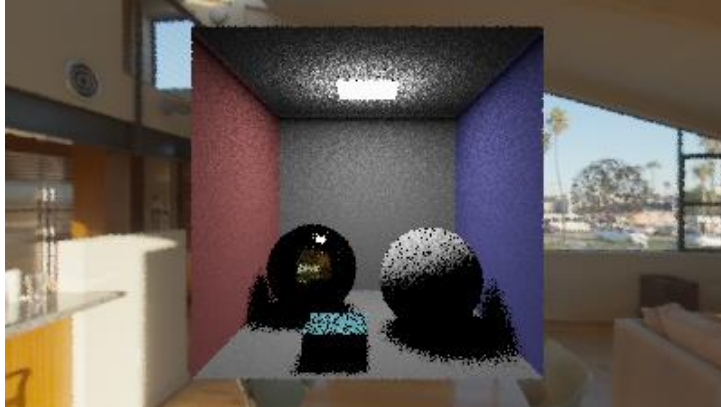
只有直接灯光：



有环境光：



环境光+平面光：

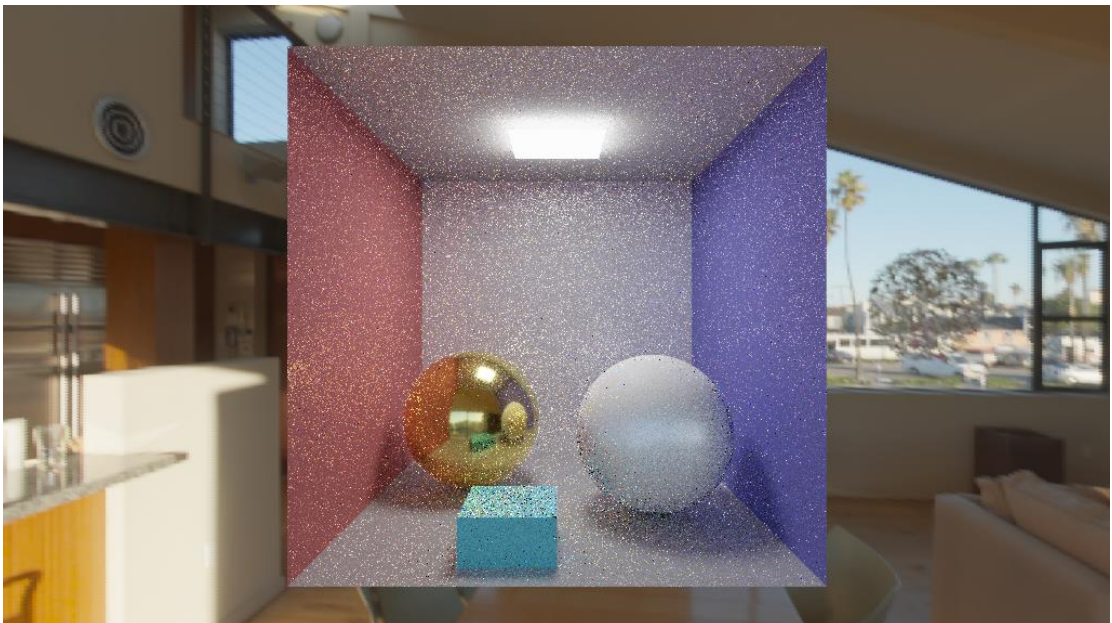


直接光+间接光：

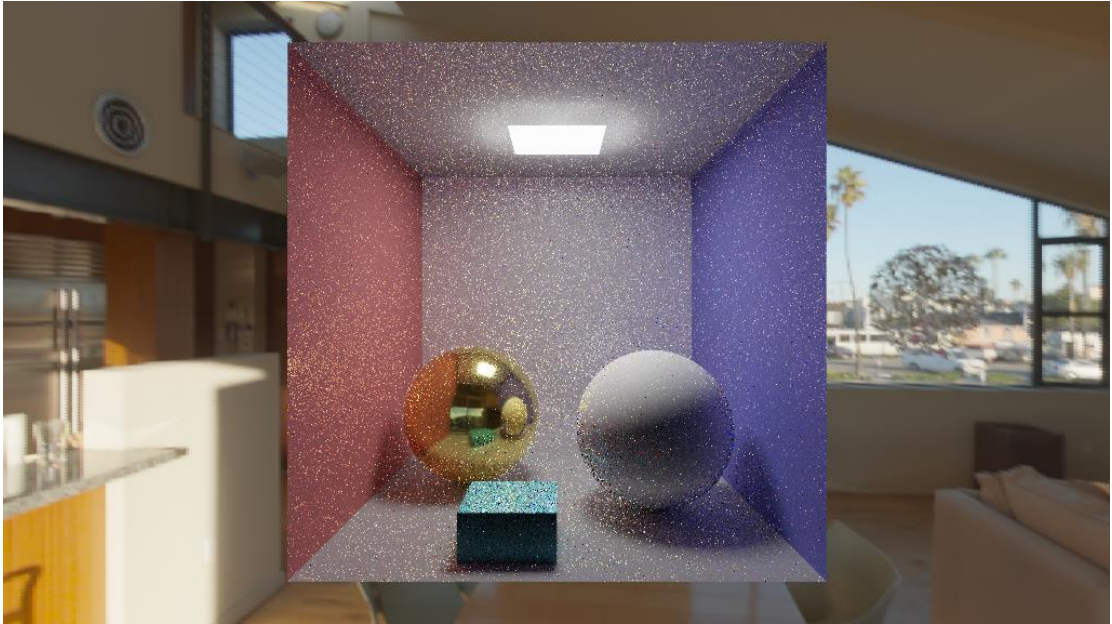


最终成品： spp 为 100， width 为 1024

未处理与光源相交时角度：



处理与光源相交时角度：



自制场景：



六．问题与展望

6.1 未解决的问题

- 在生成图片的时候没法调整摄像机的位置
- 生成的图片噪声太大，spp 很高也没有什么改观
- 运算时间太长，一张图片生成的成本太高

6.2 future work

- 增加改变摄像机位置的组件
- 加快算法，去除噪声（仿照 laplace 坐标）
- 对绝对反射材质的判定
- 更好的 UI 界面