

Homework 9 Path Tracing

华南理工大学 曾亚军

一 实验目的

- 完成 Path Tracing 渲染算法。
- 环境光贴图和重要性采样。
- 搭建自己的场景并渲染。

二 算法主要内容

2.1 渲染方程

$$L_o(\mathbf{p}, \omega_o) = L_e(\mathbf{p}, \omega_o) + \int_{\mathcal{H}^2(\mathbf{n}(\mathbf{p}))} f_r(\mathbf{p}, \omega_i, \omega_o) L_i(\mathbf{p}, \omega_i) \cos \theta_{\omega_i, \mathbf{n}(\mathbf{p})} \omega_i. \quad (1)$$

对每条光路，我们都要求解上述渲染方程。其中， L 表示辐射度，反应了物体在该点和该方向上的辐射能量。 $L_o(\mathbf{p}, \omega_o)$ 是在点 \mathbf{p} 处以出射方向 ω 的辐射度。 $L_e(\mathbf{p}, \omega_o)$ 反应了物体自发光，只有光源该项才会非零。上式等号右边第二项是由场景中所有其他可见物体对该点的贡献，记为 $L_r(\mathbf{p}, \omega_o)$ 。

由递归，可得

$$\begin{aligned} L_r(\mathbf{p}, \omega_o) &= \int_{\mathbf{p}, \omega_o, \omega_i} (L_e(\mathbf{p}', -\omega_i) + L_r(\mathbf{p}', -\omega_i)) \\ &= \int_{\mathbf{p}, \omega_o, \omega_i} L_e(\mathbf{p}', -\omega_i) + \int_{\mathbf{p}, \omega_o, \omega_i} L_r(\mathbf{p}', -\omega_i), \end{aligned} \quad (2)$$

其中 \mathbf{p}' 为从点 \mathbf{p} 往 $-\omega_i$ 方向与场景物体的交点。将将

- $\int_{\mathbf{p}, \omega_o, \omega_i} L_e(\mathbf{p}', -\omega_i)$ 称为直接光，记作 $L_{\text{dir}}(\mathbf{p}, \omega_o)$
- $\int_{\mathbf{p}, \omega_o, \omega_i} L_r(\mathbf{p}', -\omega_i)$ 称为间接光，记作 $L_{\text{indir}}(\mathbf{p}, \omega_o)$

通过积分变换，直接光部分可以转化为如下积分：

$$L_{\text{dir}}(\mathbf{x} \rightarrow \mathbf{z}) = \int_A f_r(\mathbf{y} \rightarrow \mathbf{x} \rightarrow \mathbf{z}) L_e(\mathbf{y} \rightarrow \mathbf{x}) G(\mathbf{x} \leftrightarrow \mathbf{y}) A(\mathbf{y}), \quad (3)$$

其中， $G(\mathbf{x} \leftrightarrow \mathbf{y})$ 为几何传输项，反应传输效率，

$$G(\mathbf{x} \leftrightarrow \mathbf{y}) = V(\mathbf{x} \leftrightarrow \mathbf{y}) \frac{|\cos \theta_{\mathbf{x}, \mathbf{y}}| |\cos \theta_{\mathbf{y}, \mathbf{x}}|}{\|\mathbf{x} - \mathbf{y}\|^2}. \quad (4)$$

间接光可直接由递归得到。并通过轮盘赌控制间接光的递归次数。

2.2 蒙特卡罗方法

通过蒙特卡洛算法，可通过如下公式近似积分的值：

$$\begin{aligned} L_{\text{dir}}(\mathbf{x} \rightarrow \mathbf{z}) &\approx \sum_{i=1}^{N_e} \sum_{j=1}^{N_i} \frac{f_r(\mathbf{y}_i^{(j)} \rightarrow \mathbf{x} \rightarrow \mathbf{z}) L_e(\mathbf{y}_i^{(j)} \rightarrow \mathbf{x}) G(\mathbf{x} \rightarrow \mathbf{y}_i^{(j)})}{p(\mathbf{y}_i^{(j)})} \\ L_{\text{indir}}(\mathbf{p}, \boldsymbol{\omega}_o) &\approx \sum_{k=1}^N \frac{f_r(\mathbf{p}, \boldsymbol{\omega}_i^{(k)}, \boldsymbol{\omega}_o) L_r(\mathbf{p}^{(k)}, -\boldsymbol{\omega}) \cos \theta_{\boldsymbol{\omega}_i, \mathbf{n}(\mathbf{p})}}{p(\boldsymbol{\omega}_i^{(k)})} \end{aligned} \quad (5)$$

此外，在环境光的重要性采样中，通过每个像素的灰度值（归一化）才表示该点的采样概率。并通过积分变换转化为半球面方向的概率：

$$p(\boldsymbol{\omega}_i) = \frac{wh}{2\pi^2 \sin \theta} p_{\text{img}}(i, j) \quad (6)$$

2.3 别名法

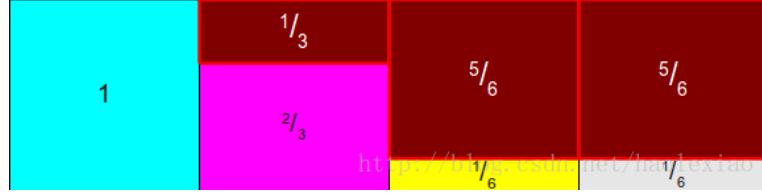


图 1: alias method

另外，别名法（Alias method），可以时间复杂性为 $O(N)$ 来构造别名表，从而实现每次采样时间复杂性为 $O(1)$ 。由于采样次数非常多，因为采取 Alias 别名法可以加快算法采样效率。别名法的实现很简单，先生成概率表 $prob$ 和 $accept$ ，分别表示该像素点的概率和采样接受概率 ($accept(i) = N * prob(i)$)。并创建两个队列 $small$ 和 $large$ ，将 $accept$ 小于 1 和大于 1 的序号分别进队处理。队列构建完成后，每次分别去两个队列的队首元素，进行填充和记录。

三 结果与讨论

3.1 初步结果

首先，从低采样开始进行测试，下图中展示了 $spp = 2$ 下的直接光、间接光对渲染结果的影响。

从图 3.2 中，可以比较清晰地看见直接光和间接光对渲染结果的影响。在图 3.2 的 (e) 和 (f) 中分别对比了均匀采样的初步结果和重要性采样的初步结果。结果显示，采取重要性采样，噪声数量会有所下降，且整体颜色要比均匀采样的渲染结果要稍微暗一些。

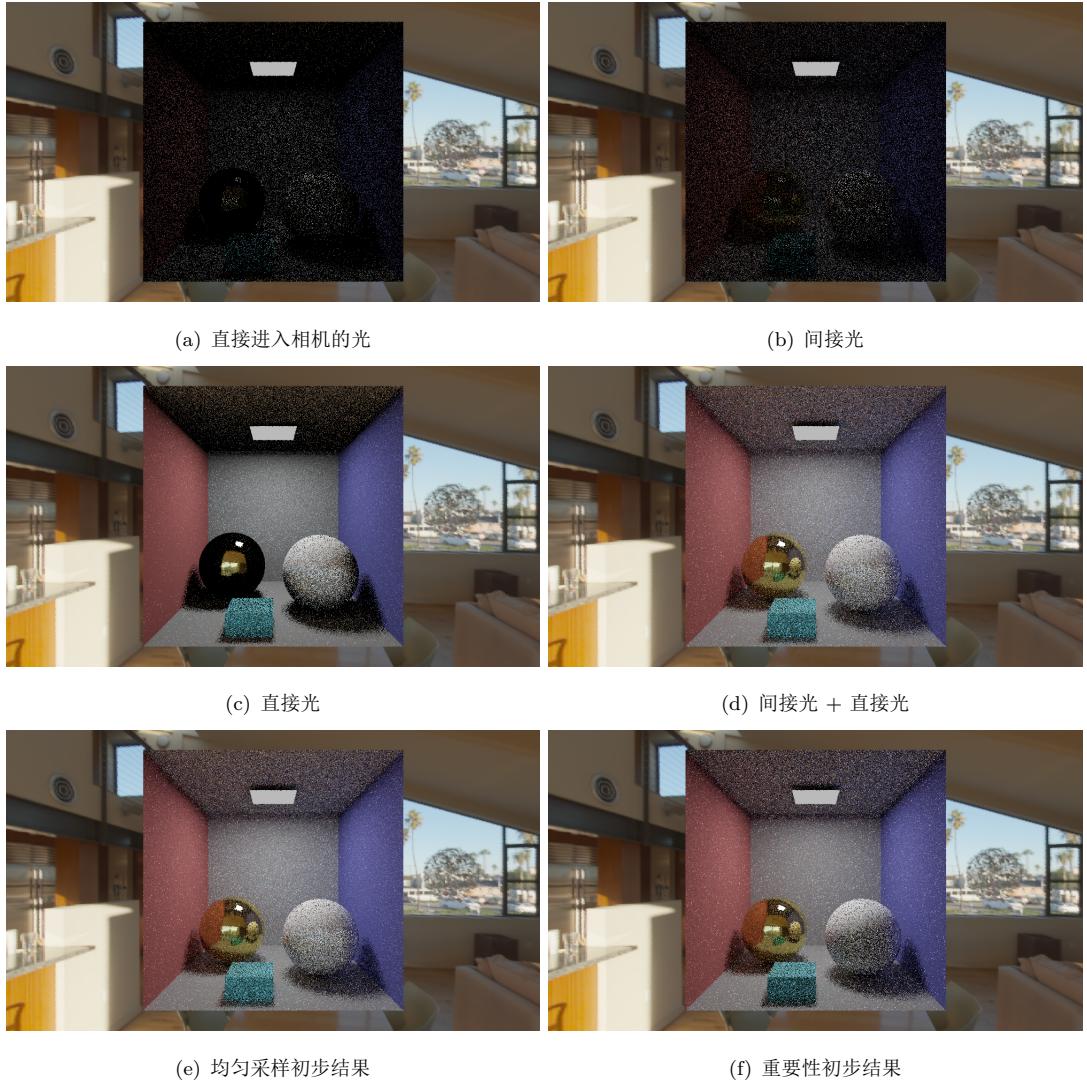


图 2: spp=2 下渲染结果

3.2 去噪

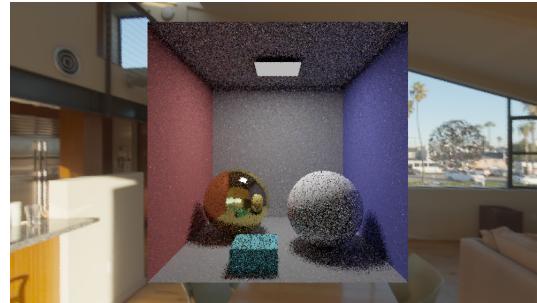
我们考虑对结果进行去噪。由之前参数化和去噪贴图的经验,我首先考虑了将图像的每一个像素点视为三维空间的一个点,而图像周围的八个点视为其邻接关系。并使用拉普拉斯坐标 $\delta_i = p_i - \frac{1}{N(i)} \sum_{j \in N(i)} p_j$ 。当像素点偏离中心一定程度时,认为是噪声点,取其中心坐标作为当前像素值。其结果如下图 (b) 去噪方法 1 中显示。噪声总体数目减少,但由一部分噪声反而被扩大化了。且不明确偏离程度到底是多少被认定为噪声才是适合的。

因而我们考虑采取另外一种去噪方法,同样是将图像的每一个像素点视为三维空间的一个点,以周围八个点计算重心和方差,若当前像素点偏离中心值的平方距离超过 $3\delta^2$ 时,认为时噪声点,用重心替代。其结果如下图去噪方法 2 中可以看到,尽管不能完全去除噪声,但还是能实现比较好的去噪结果。



(a) 初步结果

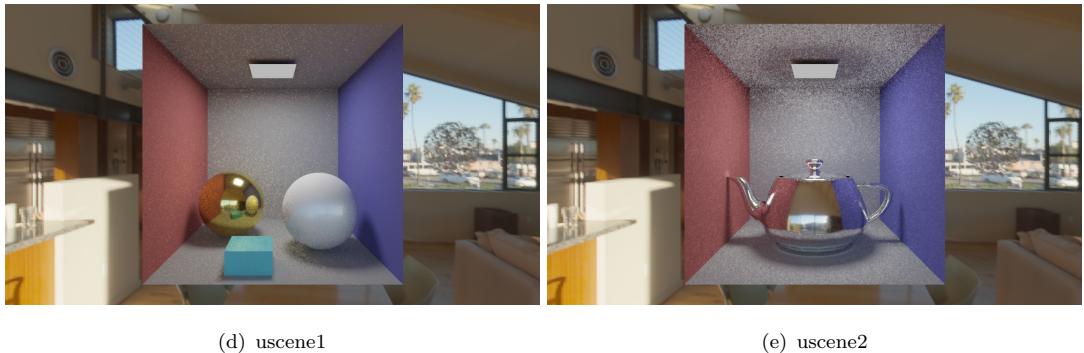
(b) 去噪方法 1



(c) 去噪方法 2

事实上，将图像像素点视为三维点云的化，有很多去除离群噪声的方法。另外一种简单的思路 (Liu, S.J., 2012) 是取每个点的最近邻的 k 个点，计算这 k 个点的重心，并以此构成一个新的点云。并用该店在新的点云中计算最近邻再作比较。但在本项目中，项目框架较大，引入 ANN 库时 cmake 处发生错误，因而最后没有实现这种去噪方法。

3.3 采样率更高的结果和其他场景



(d) uscene1

(e) uscene2

四 注意事项

- Alias 别名法处应使用更高精度的数据类型，如 `double`。否则，由于数据点个数和所有点 `rgf` 值之和都是比较大的数，在归一化的过程中会丢失精度，造成较大误差。
- 程序中光线的方向采样可能会采样到法向对应的下半平面，对此应当不断重新采样，直到采样到上半平面的值。

五 收获和感悟

这次作业的代码量也不多，核心就在于去理解好光线追踪的渲染方程。包括整一个代码框架，其实在课程录屏里面已经讲解得非常详细了。因此实现起来没有什么难度。实现出来会有一定的成就感，毕竟在学习这门课程之前，根本就没有接触过跟光线相关的处理。以前一直会觉得，这是非常非常高深的事情，到现在却发现，自己可以触碰了。而且很多论文也能轻松看懂了。

到这个作业为止，其实已经学习了很多东西了。从数字图像的处理，到三维网格的极小化、参数化，再有物理模型的仿真，到现在的光线和阴影。学到的东西非常多，内容也比较广泛，但其实也相当于各个方面涉猎一点。日后还需要进一步地读各种文献，了解更多出色的思路和想法，并提出自己的想法。而在这门课程的学习里面，我确实感受到了从事计算机图形学的乐趣，不过这个也很难说清楚，也许是实现算法的成就感，又或者是感受到了图像和网格的几何美？还是被这个过程中读过的论文那些优秀的想法所折服？也可能都有。相信未来走计算机图形学这条路的我，也会像现在一样充满好奇和热情。

参考文献

Liu, S.J., Chan, K., Wang, C.C.L., 2012. Iterative Consolidation of Unorganized Point Clouds. In: IEEE computer graphics and applications 32, pp. 70-83.