

| 学号 | 姓名 | 论文规范性 (10) | 问题分析与调研 (30) | 方案创新性 (20) | 实验结果分析与讨论 (40) | 结课论文总成绩 (100) |
|----------|-----|------------|--------------|------------|----------------|---------------|
| 21301103 | 石昊原 | 8 | 21 | 17 | 36 | 82 |

没有参考文献及现有工作梳理



图形学课程结题（论文）

光线追踪调研实验

学 院： 软件学院

专 业： 软件工程

学生姓名： 石昊原

学 号： 21301103

指导教师： 吴雨婷

北京交通大学

2024 年 6 月

1 引言

本论文是针对现代图形学渲染技术中的离线光线追踪方向进行的调研、应用及分析。论文首先从技术介绍与发展现状进行介绍。

1.1 技术介绍

在光线跟踪中，每一个光线的路径由多重直线组成，几乎总是包含从原点到场景的反射，折射和阴影效应。每一束光线的直线部分的位置和方向总是在不断变化，因此每一条光线都要用一个数学方程式来表示，定义光线的空间路径为时间的函数，根据光线在到达屏幕前经过的场景中的目标的色素或颜色来分配给每一束光线一种颜色。

1.2 发展现状

相比于实时光线追踪，离线光追技术已经相对成熟，因为其能够生成高仿真的图像（模拟现实中的光照效果，如阴影、反射、折射等），所以被应用于动画电影和特效的制作中。

而实时光线追踪，随着计算机硬件性能的提升，实时光线追踪技术开始在游戏中得到广泛应用，尤其以 NVIDIA 的 GeForce RTX 系列显卡为主，为通用实时光线追踪技术提供硬件条件。

本次实验由于时间关系，将只进行离线的实时光线追踪技术尝试。

2 方法描述

为了实现现代图形学光线追踪渲染的简单流程模拟，本次实验包含 Whitted 光线追踪、光线追踪加速和渲染方程三部分。

2.1 Whitted 光线追踪

Whitted 光追只考虑了直接光照，即光源照射到物体，光线被反射或折射进入摄像机，并未考虑间接光照的影响，即别的物体反射的光打到另一个物体上再进入人眼，所以它不能很好的模拟全局光照的效果，但这是理解光线追踪的基础。

2.1.1 光线性质研究

不考虑光的波动性，在 Whitted-Style 光追中，光线的定义简化为三条：

- (1) 光是沿直线传播的。
- (2) 光线之间不会碰撞（即光线碰撞不会对两条光线造成任何影响）。
- (3) 光从光源发出，经过层层反射折射，最终进入摄像机，并且光路是可逆的。

2.1.2 算法过程

由于光路的可逆性，算法通过摄像机对每一个像素发出一道射线，当光线射中物体时，将该点连线到光源，判断是否在阴影中，如果没有被遮挡，就对这一点进行着色。反射或折射多次时，每个弹射点都计算一个着色值，都对最终像素值做出贡献，最终像素色值是各个弹射点色值的加权和。

2.1.3 光线与物体求交

定义光线为 $r(t) = O + td$, O 为探测光线的出发点, d 为单位向量, t 为时间。

与球求交：

球的表达式为 $(P - C)^2 - R^2 = 0$ ，联立得到交点

$$(O + td - C)^2 - R^2 = 0$$

$$(d^2)t^2 - 2(O - C)d + (O - C)^2 - R^2 = 0$$

设 $a = d^2, b = -2(O - C)d, c = (O - C)^2 - R^2$ ，简化为

$$at^2 + bt + c = 0$$

根据求根公式可知，

$$t = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}, t \geq 0$$

与隐式表面求交：

定义隐式表面为 $p: f(p) = 0$ ，联立得到交点

$$f(O + td) = 0$$

与显式表面求交：

在图形学中，模型通常是由多个三角形平面组成的，所以与显式表面求交有重要意义。

与三角形求交分为两个步骤，首先进行光线与三角形所在的平面求交，定义平面由法线 \vec{N} 与点 p' 构成，这样定义的原理是跟法线向量垂直的面有无数个，再额外给定某个点，即可唯一确定一个面。由于平面内所有线段与平面法线垂直对的关系，平面内所有的点 p 都满足 $(p - p')\vec{N} = 0$ 。

联立得到交点 $(O + td - p')\vec{N} = 0$

$$t = \frac{(p' - O)\vec{N}}{d\vec{N}}, t \geq 0$$

之后判断该点是否位于三角形内，使用向量叉乘法判断点与所有边方向是否一致。

也可以使用 Möller-Trumbore 射线-三角形求交算法，三角形内向量 \overrightarrow{AP} 、 \overrightarrow{AB} 、 \overrightarrow{AC} 线性相关，如 Figure1，所以可以写成如下形式

$$\begin{aligned}\overrightarrow{AP} &= u\overrightarrow{AB} + v\overrightarrow{AC} \\ \Rightarrow P - A &= u(B - A) + v(C - A) \\ \Rightarrow P &= (1 - u - v)A + uB + vC\end{aligned}$$

联立方程， $O + td = (1 - u - v)A + uB + vC$ ，存在三个未知数 t 、 u 、 v 。

解三维方程可以得到三个位置量

$$\begin{bmatrix} t \\ u \\ v \end{bmatrix} = \frac{1}{\vec{S}_1 * \vec{E}_1} \begin{bmatrix} \vec{S}_2 * \vec{E}_2 \\ \vec{S}_1 * \vec{S} \\ \vec{S}_2 * \vec{D} \end{bmatrix}$$

其中 $\vec{E}_1 = \vec{P}_1 - \vec{P}_0$ ， $\vec{E}_2 = \vec{P}_2 - \vec{P}_0$ ， $\vec{S} = \vec{O} - \vec{P}_0$ ， $\vec{S}_1 = \vec{D} \times \vec{E}_2$ ， $\vec{S}_2 = \vec{S} \times \vec{E}_1$ 。

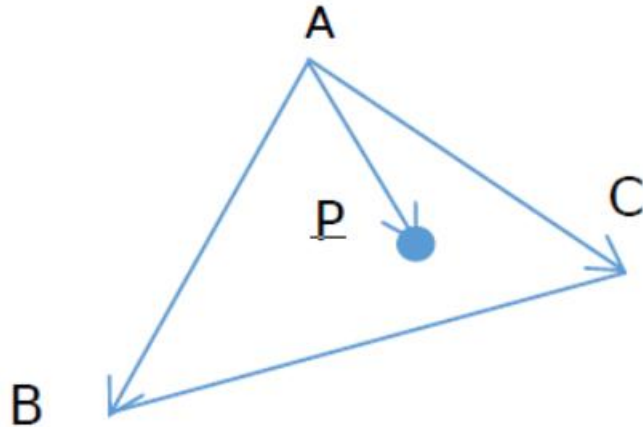


Figure.1

2.2 加速光线追踪

对象划分（层次包围盒） Bounding Volume Hierarchy(BVH)是一种不划分空间，而是划分物体的划分方式，也是目前最常用的求交加速划分策略。

首先定义包围盒对象，用一个简单的形状（长方体）框出这个物体，光线如果碰不到包围盒，必然碰不到包围盒内的模型。这里通常使用 Axis-Aligned Bounding Box (AABB 包围盒，轴对齐包围盒)，因为这样平面的法向更加工整，计算量更小。我们通常认为，一个盒子其实就是 6 个平面围住的一块公共区域。AABB 包围盒是轴对齐的，即面跟 xoy 或 xoz 或 yoz 任一平面平行。

在 3D 中，对三个面解出光线与其交点，将问题简化为 2D 形式，定义光线方程为 $r(t) = O + td$ ，与 $x = x_0$ 平面相交联立得 $x_0 = O_x + td_x$ ，计算得出

$$t = \frac{x_0 - O_x}{d_x}$$

同理对任意一个平面都可以计算出 t_{min} 和 t_{max} ，回到 3D 情景，分别对三个面求出 t_{min} 和 t_{max} ，定义光线进入包围盒的时间为 $\max(t_{min})$ ，定义光线离开包围盒的时间为 $\min(t_{max})$ ，若 $\max(t_{min}) < \min(t_{max})$ 且 $\min(t_{max}) \geq 0$ ，说明光线与包围盒相交。如图 Figure2。

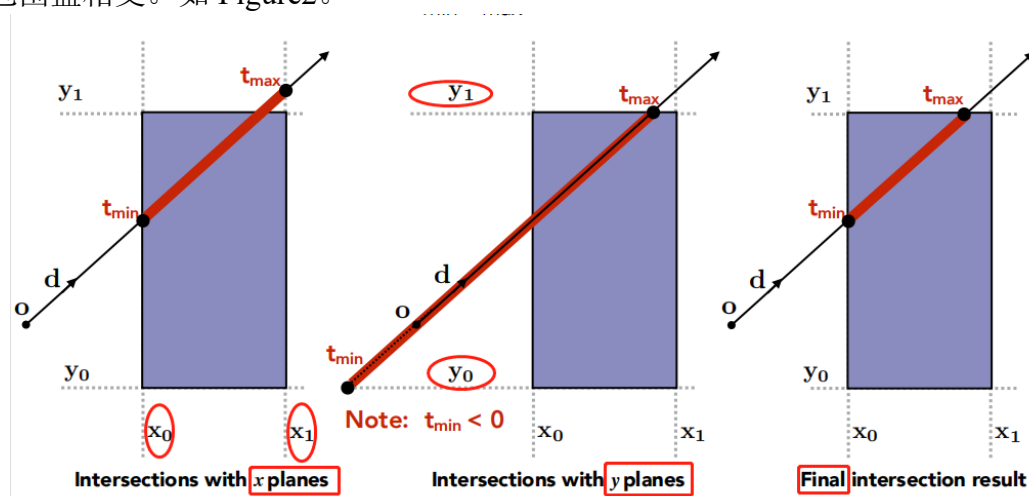


Figure.2

2.3 渲染方程

1986 年 James Kajiya 老先生提出伟大的“渲染方程 (The Rendering Equation)”，渲染方程从 1986 年指导了整个渲染领域至今。我们现在做的一切工作都是为了逼近或者直接求解渲染方程，以此达到照片级真实感。

2.3.1 辐射度量学

辐射度量学是为得到更好的更真实的光照计算效果(Whitted 光线追踪、Blinn-phong 着色计算是不真实的)。使用辐射度量学可以准确的测量空间中的光照的系统方法，定义了一系列属性，但是依然是在几何光学中的描述，不涉及光的波动性、互相干扰等。辐射度量学能够精确描述光、材质和表面，是路径追踪体系的基础。

在物理中，Radiant Energy 辐射能 Q 、Radiant Flux(Power) 辐射通量 Φ 、Radiant Intensity 辐射强度 I 、Irradiance 辐照度 E 和 Radiance 辐亮度 L 可以准确定义光照，以便我们描述光在空间中的属性。

定义如下：

Radiant Energy 辐射能：电磁波中电场能量和磁场能量的总和，也叫电磁波的能量；

Radiant Flux(Power) 辐射通量 (辐射功率)：单位时间内产生的、反射的、接收的能量；

Radiant Intensity 辐射强度：单位立体角上，产生的、反射的、接收的辐射通量；

Irradiance 辐照度，符号： E ：物体单位投影面积上接收到的能量；

Radiance 辐亮度，符号： L ：单位投影面积收到/发出的单位立体角上的辐射通量。

2.3.2 双面反射分布函数

单位面积 dA 接收到入射光后发生反射，其本质就是自身变为新的光源，向空间进行辐射。单位面积 dA 接收到 w_i 方向来的辐亮度，即 $L(w_i)$ 或 $dE(w_i)$ ，之后辐射到 w_r 方向上（反射出去无数多条，能量会被均分，而其中特定的一条分到的能量就是一个微元），辐射出去的辐亮度为 $dL_r(x, w_r)$ ，如 Figure3。

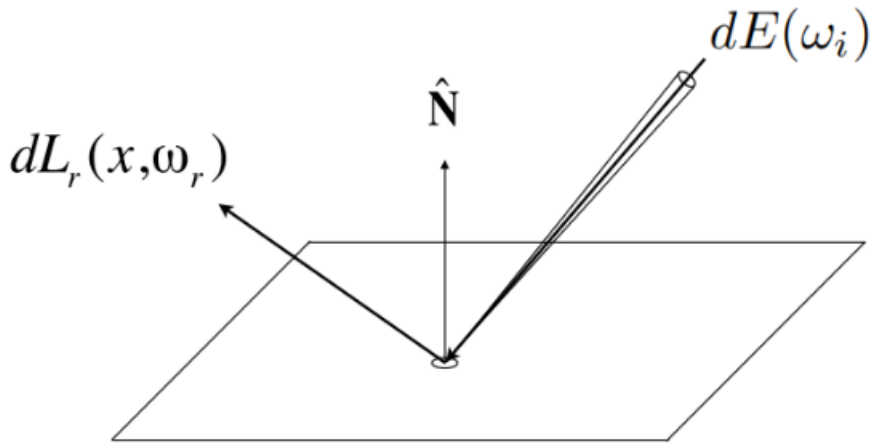


Figure.3

双面反射分布函数 Bidirectional Reflectance Distribution Function(BRDF) 描述了入射光线经过某个表面反射后在各个可能的出射方向上能量分布，如 Figure4。若已知入射方向 w_i 以及出射方向(自变量) w_r ，能得到值域在 $[0,1]$ 的浮点数返回值，该返回值为反射光线能量与入射光线能量的比值。

$$f_r(w_i \rightarrow w_r) = \frac{dL_r(w_r)}{dE_i(w_i)} = \frac{dL_r(w_r)}{L_i(w_i)\cos\theta_i dw_i}$$

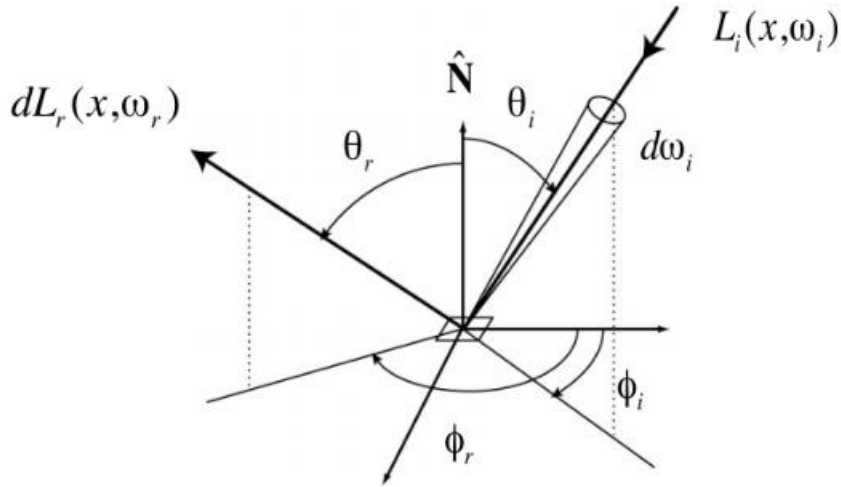


Figure.4

BRDF 描述一个表面如何把它收集的能量反射到另一个方向去，可以理解为定义了物体的材质。

2.3.3 反射方程

一个面元往某个方向反射光线的强度，需要考虑各条入射光线的贡献。

$$\int_{H^2} f_r(p, w_i \rightarrow w_r) L_i(p, w_i) \cos\theta_i dw_i$$

$f_r(p, w_i \rightarrow w_r)$ 为 BRDF 项，返回值值域[0,1]；

$L_i(p, w_i) \cos \theta_i dw_i$ 为面元对特定方向 w_i 接收到的辐照度。

反射方程可以理解为 BRDF（物体的材质）乘上面元对特定方向 w_i 接收到的辐照度，得到反射到目标方向的辐亮度。

但是物体表面的光线传播，要考虑物体自发光，在反射方程中加上一项： $L_e(p, w_o)$ ，即物体的自发光项。

渲染方程最终如下：

$$L = L_e(p, w_o) + \int_{H^2} f_r(p, w_i \rightarrow w_r) L_i(p, w_i) \cos \theta_i dw_i$$

简写为

$$l(u) = e(u) + \int I(v) K(u, v) dv$$

即

$$L = E + KL$$

简单理解为出射方向的强度 = 自发光强度 + K*入射光强度；K 直接理解成一个反射操作符。

通过级数展开，得到最终描述光线追踪的渲染方程

$$L = E + KL + K^2L + K^3L + \dots$$

2.4 路径追踪

渲染方程只有寥寥几行，但却不容易解出。计算机图形场景很复杂，数十亿三角形在今天并不罕见。但由于没有办法直接解出渲染方程，Kajiya 提出了第二个关键创新。

Kajiya 证明了统计技术可以用来解出渲染方程。即使没有直接解出该方程，也有能够沿着单个光线的路径求解。如果能够沿着足够多的光线路径解出该方程，就能准确模拟场景中的照明，生成逼真的图像。

2.4.1 直接光照

路径追踪的本质就是解渲染方程。

$$\begin{aligned} L &= L_e(p, w_o) + \int_{H^2} f_r(p, w_i \rightarrow w_r) L_i(p, w_i) \cos \theta_i dw_i \\ &= L_e(p, w_o) + \int_{H^2} f_r(p, w_i, w_r) L_i(p, w_i) (nw_i) dw_i \end{aligned}$$

蒙特卡洛方法中 $\int_a^b f(x)dx \approx \frac{1}{N} \sum_{k=1}^N \frac{f(X_k)}{p(X_k)}$,

在 L 中, $f(X_k) = f_r(p, w_i, w_r) L_i(p, w_i)(nw_i)$, 带入后使用蒙特卡洛积分求解:

$$L = \int f_r(p, w_i, w_r) L_i(p, w_i)(nw_i) dw_i$$

$$\approx \frac{1}{N} \sum_{i=1}^N \frac{L_i(p, w_i) f_r(p, w_i, w_o)(nw_i)}{p(w_i)}$$

按照某种自己定的 PDF, 采样 N 个不同入射方向, 对每个入射方向 w_i 的确定方式: 从 p 点向该方向发出射线 r, 判断如果击中了光源, 则计算此方向贡献的 L_o , 并且累加到之前的 L_o 上。

由于低采样时噪点太多, 而高采样时性能又会造成浪费, 所以存在一种优化方法, 即直接采样光源的表面。想要直接采样光源表面, 需要找到 dA 与 dw 的关系进行积分换元。单位立体角中, $dw = dA/r^2$, 所以 dw 更新为

$$dw = \frac{dA \cos\theta'}{\|x' - x\|^2}$$

最终渲染方程变为

$$L_o(p, w_o) = \int_A f_r(x, w_i, w_o) L_i(x, w_i) \frac{\cos\theta \cos\theta'}{\|x' - x\|^2} dA$$

2.4.2 全局光照

全局光照本质上就是直接光照+间接光照, 而间接光照的本质是递归的直接光照求和, 即将反射点作为新的光源来看, 进行直接光照的计算, 最终相加得到全局光照。

我们规定, 着色点随机发出 N 条光线, 对这 N 个选中的方向, 跟踪一条光线, 如果光线打到光源, 终止这个方向的着色, 如果光线打到另一个物体, 那么递归。由于光线会弹射无数次, 我们采用俄罗斯轮盘赌 (Russian Roulette), 概率 P 继续发射光线, 着色的结果加上 L_o/P , 概率(1-P)不发射光线, 着色的结果不变, 可以保证最终着色结果的期望与真实 L_o 一致:

$$E = P * \left(\frac{L_o}{P}\right) + (1 - P) * 0 = L_o$$

3 实验设置

3.1 Whitted 光线追踪

在 Whitted 光线追踪中，最重要的是找到光线与物体的交点，程序找到交点后直接进行着色并返回结果。

实验流程：添加两颗球体，材质分别为透明（反射与折射）与粗糙（漫反射），遍历像素生成光线，找到交点，执行着色。

3.2 加速结构

实验选用 BVH 加速结构，即在 3.1 Whitted 光线追踪的基础上，在寻找光线与物体交点前首先判断光线是否与包围盒相交，适用于物体数量增多、模型变得更加复杂时来加速求交过程。

3.3 路径追踪

基于 3.1 与 3.2 的实验内容基础上，修改了渲染逻辑，实现路径追踪；修改了 Object, Sphere, Triangle, TriangleMesh, BVH，添加了 area 属性与 Sample 方法，以实现光源按面积采样，并在 Scene 中添加了采样光源的接口 sampleLight；修改了 Material 并在其中实现了 sample, eval, pdf 三个方法用于 Path Tracing 变量的辅助计算。

4 实验结果与分析

4.1 Whitted 光线追踪

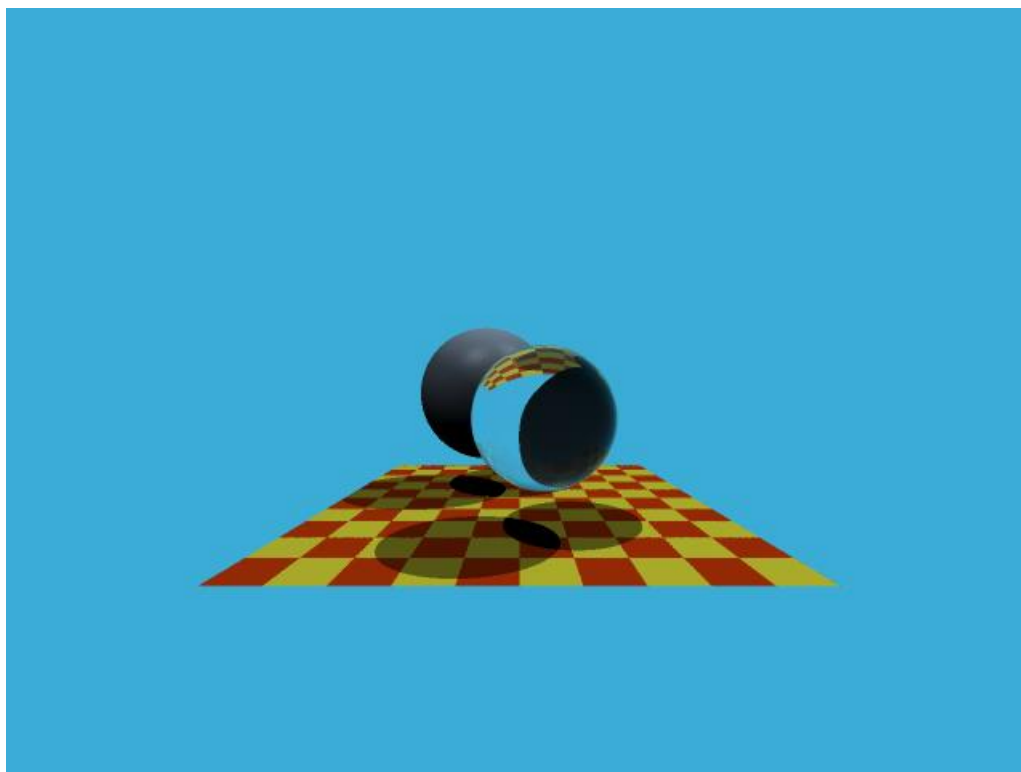


Figure.5

Whitted 光线追踪可以很好地模拟光线的反射、折射和阴影等效果。如 Figure5。

4.2 加速结构

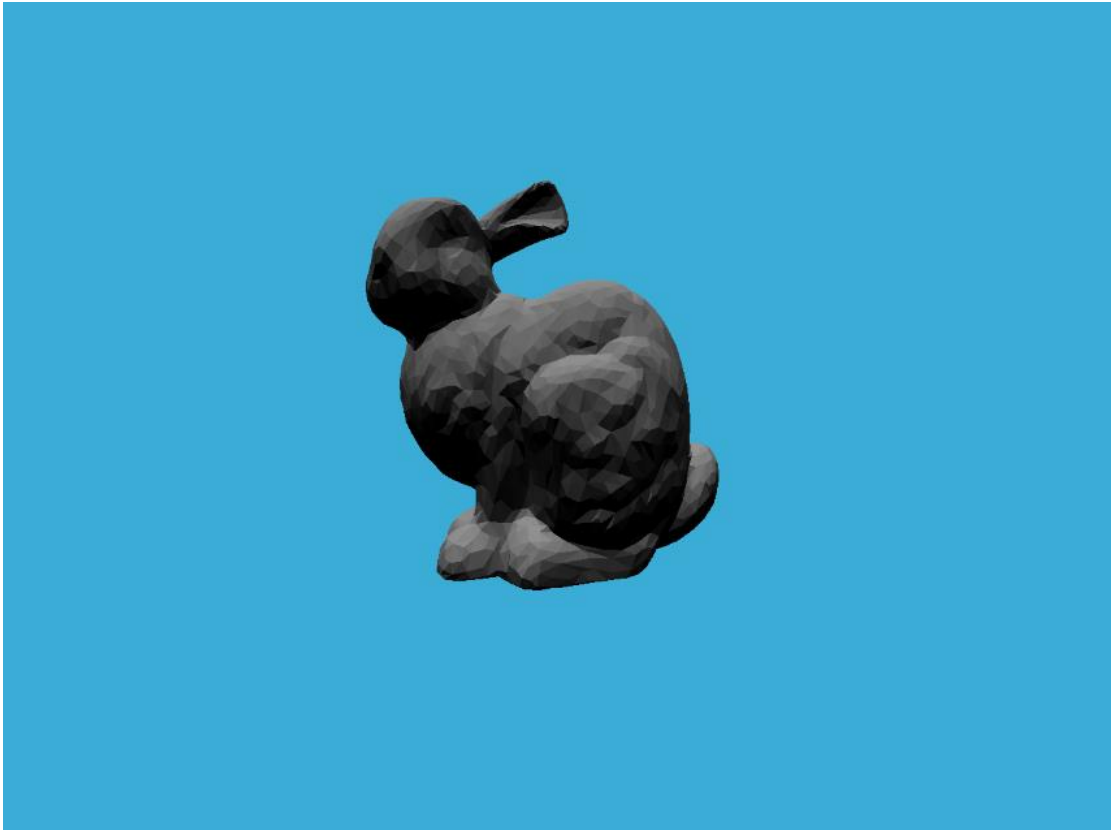


Figure.6

使用 BVH 加速结构后，比较复杂的模型也可以很快地实现光线求交，节省渲染时间，如 Figure6。

4.3 路径追踪



Figure.7

成功实现路径追踪，可以看到即使没有被光源直接照射的平面也会进行着色，证明成功实现了全局光照，效果如 Figure7。

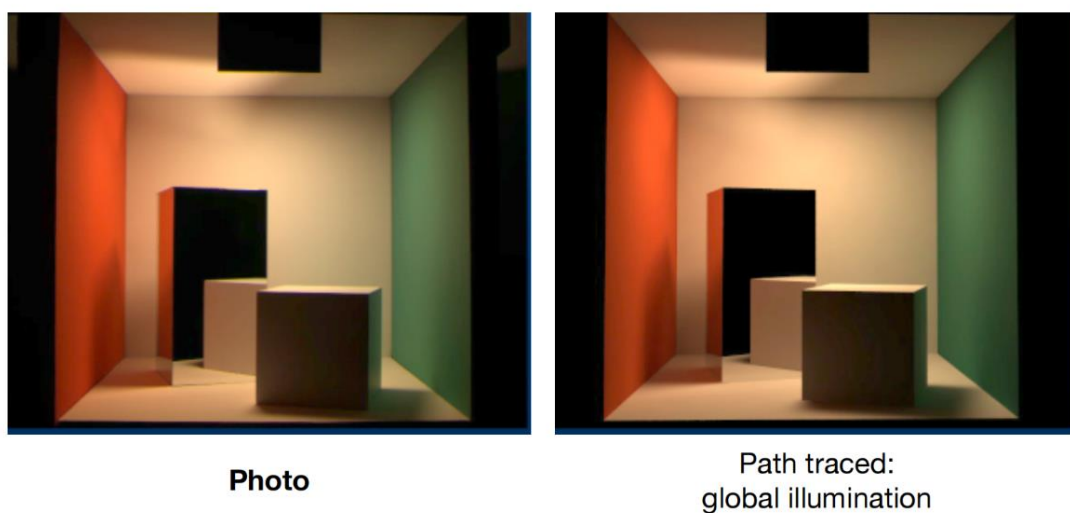


Figure.8

相比 games101 官方提供的效果图（Figure8），实验结果存在大量的噪点，猜测可能是因为采样率设置过低导致的。

5 结论

本次实验内容使用了 Games101 课程的 5、6、7 节进行实现，并成功跑通了离线光线追踪，实现了 Whitted 光线追踪、BVH 加速结构与路径追踪。总结下来，现代图形学光线追踪的简单流程即发射探测光线、计算光线与包围盒是否相交、计算光线与三角形是否相交、俄罗斯赌盘随机发射光线、渲染方程递归计算辐照度、Gamma 计算颜色、最终汇总着色。在本次实验中，实现 Whitted 光线追踪与 BVH 加速结构较为简单，而路径追踪的学习较难，涉及到物理、概率论、微积分、编码等多个学科知识，但总归是浅显地接触到了现代正在使用的前沿技术，系统性地学习了现代图形追踪的基础理论知识。