

# 实验 1: 三个小球的简单光线追踪

2021 年 10 月 20 号和 10 月 27 日进行本实验的验收  
2021 年 11 月 10 号晚 10 点之前提交最终实验报告

## 任务

### 通过简单的 ray-tracing 来实现对三种不同材质球体的渲染

#### 1. 实验简介

本实验旨在通过实现一个简单的光线追踪算法，让同学们对渲染有个初步理解。同学们需要利用学过的微积分知识与高中物理知识，理解不同光线在不同材质上的作用方式（即光线在物体表面的吸收，反射和折射等等现象），并且可以用程序实现出来。

本次实验是通过软件模拟光线的作用方式，所以不需要使用 opengl 库。

#### 2. 实验内容

1. **环境配置**：本实验需要同学们提前在 Linux 设备中安装好 cmake 与一个现代化的 C++ 编译环境。建议使用 linux 环境完成作业。
2. **编译命令**：请在命令行打开该目录后输入

```
1    mkdir build
2    cd build
3    cmake ..
```

如果你使用的是 Mac/Linux 系统，接下来使用 `make` 命令即可。

3. 任务要求：

3.1 本次实验要求实现三种不同的材质的光线追踪算法：漫反射材质、金属材质、玻璃材质。

**漫反射材质**：漫反射材质不仅仅接受其周围环境的光线，还会在散射时使光线变成自己本身的颜色。光线射入漫反射材质后，其反射方向是随机的。表面越暗，吸收就越有可能发生。我们可以使用任意的算法生成随机的反射方向，就能让其看上去像一个粗糙不平的漫反射材质。比如 lambertian 反射器。

**金属材质**：金属材质带有光泽，介于漫反射材质和镜面材质中间，随机的反射并不太适用于金属材质。我们需要找到能够在视觉上表示出金属材质的某种规律。

**玻璃材质**：玻璃材质有时也称为透明材质，这种实现不能通过仅仅考虑反射来实现，需要引入对折射光线角度和能量的变化来体现。

本次实验需要按照要求对下列 material.cpp 文件中的函数进行进行补全

```

1  Vec3 random_in_unit_sphere() {
2      // 随机在单位球上采样光线
3  }
4
5  Vec3 random_sample_hemisphere(const Vec3 &normal) {
6      // 随机在单位半球上采样光线
7  }
8
9  Vec3 reflect(
10     Vec3 &v /* 输入光线方向 */
11     ,Vec3 &n /* 物体表面处的法线 */) {
12     //返回反射光线的位置
13 }
14
15 Vec3 refract(Vec3 &v, Vec3 &n, float ior) {
16     /**
17      * 输入:
18      * v: 输入光线方向
19      * n: 物体表面处的法线
20      * ior: 物体的折射率
21      *
22      * 输出:
23      * 返回一个折射后的光线
24      *
25      * */
26 }
27
28 bool Material::scatter(const Ray &r, hit_record &rec, Ray &scattered,
29                        Vec3 &attenu) {
30     /**
31      * 输入:
32      * r: 光线的方向
33      * rec: 光线与物体相交的点的信息
34      *
35      * 输出:
36      * scattered: 光线与物体作用之后的反射方向
37      * attenu: 能量衰减
38      *
39      * 返回值:
40      * bool: 表示计算出来的散射光线是否合法,
41      *
42      * */
43     switch (type) {

```

```

44     case Diffuse: {
45         // 漫反射材质代码
46         return ture;
47         break;
48     }
49
50     case Metal: {
51         //金属材质代码
52         return ture;
53         break;
54     }
55
56     case Dielectric: {
57         //玻璃材质代码
58         return true;
59         break;
60     }
61 }
62 }

```

在 main.cpp 中补全 ray\_color() 函数，实现光线的递归。

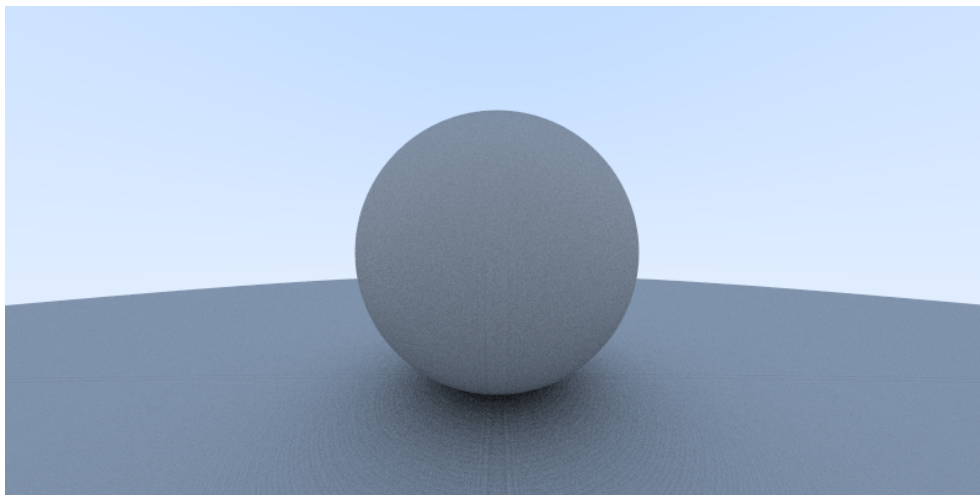
3.2 使用多线程提高在多核计算机上的运行速度，可以使用已有的多线程库，注意数据依赖。

### 3. Reference

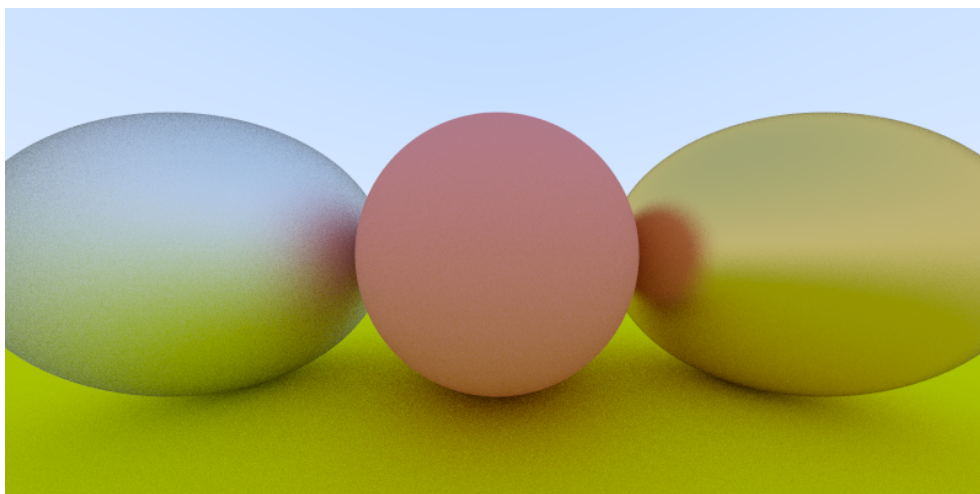
1. 初始代码框架正常运行后显示的图片



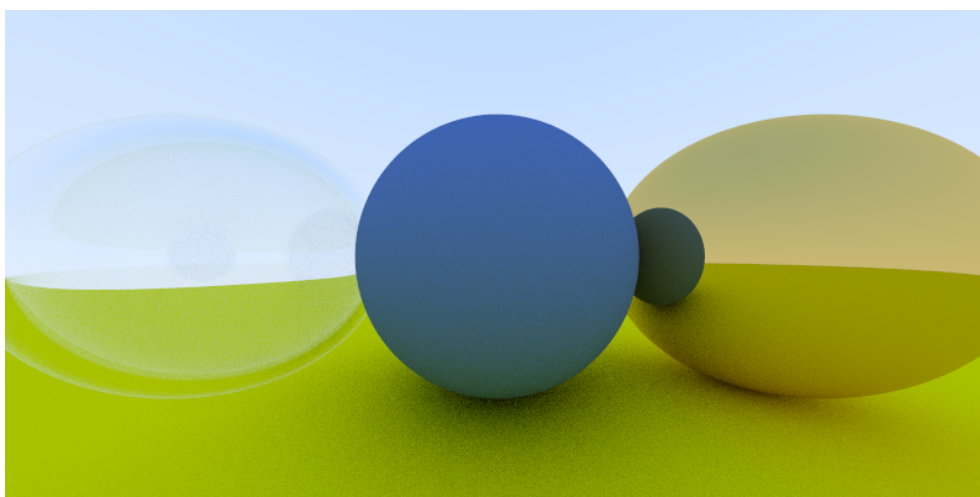
2. 实现漫反射材质后的示例图片 (参数设置请参考文档链接，下同)



3. 实现金属材质后的示例图片



4. 实现透明材质后的示例图片



上述小球和材质的参数可以从这个文档中 [Important Reference](#)找到。

#### 4. 提交要求

1. 补全代码并且要求能够正常运行，将运行结果保存为 png 格式
2. 请将所有代码以及运行结果全部打包，且书写实验报告，将代码与实验报告打包压缩为.zip/.rar 格式并提交。
3. 实验报告中的内容包括但不限于1. 实验环境; 2. 每一种材质的详细说明和必要的数学公式以及实验结果; 3. 带有注释的核心代码; 4. 心得体会
4. 建议使用 *LaTeX* 来写实验报告。但无论使用什么软件，要求提交作业的格式是 “PDF”。
5. 如果对于该实验有任何疑问，请联系助教。
6. 禁止抄袭，发现抄袭代码或者报告，则抄袭和被抄袭的人本次作业均为 0 分。