

## 0.1 编译环境

程序在 ubuntu 18.04 上编译，使用 c++ 编写，通过 cmake 构建工程，cmake 的版本为 3.10.2，编译器使用的是 g++，g++ 的版本为 7.3.0，使用的 IDE 为 emacs，同时程序通过 OpenMP 加速。

程序使用：直接运行二进制文件即可

如：./bin/tMAIN

程序运行结束后会生成一张 PPM 图片，图片内容即为渲染结果

## 0.2 程序描述

程序建立几个关键类，通过这些关键类来进行渲染计算。KdTree 类用来加速求交计算，Camera 类用来保存渲染结果，Object 类及其派生类用于保存物体的材质信息，例如反射率、折射率等。并在该类的方法中实现了 BRDF 函数，Scene 类主要用于场景的多次反射折射计算。

程序依赖于第三方库，分别是 tiny\_obj\_loader 用来读取 obj 文件和材质，和 hit\_bounding\_box 用来计算光线和 bounding box 的求交。相关源码就可以从网上下载。

经过不断优化之后，程序的流程图已经各个函数的耗时情况如下。

## 0.3 结果展示

## 0.4 程序不足

程序使用了简单的随机采样，导致效率较低，同时不能很好地模拟耀辉现象。



图 1: 低采样率下结果, sample = 1



图 2: 高采样率下结果, sample = 10

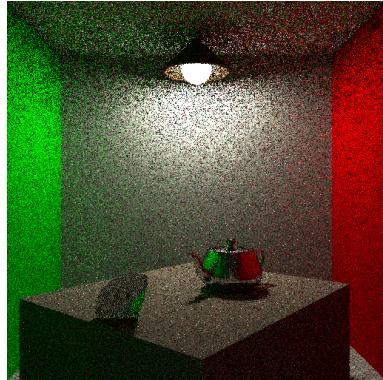


图 3: 低采样率下结果, sample = 1



图 4: 高采样率下结果, sample = 20

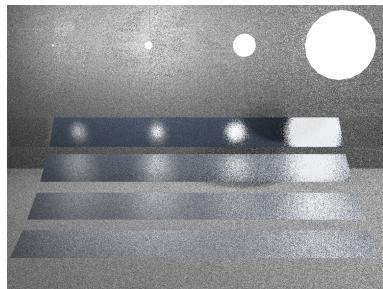


图 5: 低采样率下结果, sample = 1

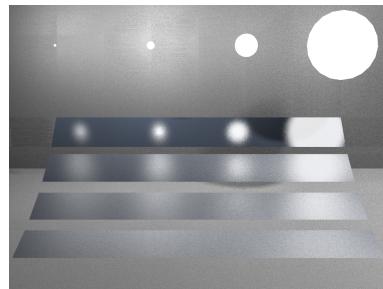


图 6: 高采样率下结果, sample = 20