

# 2020-Fall 高等计算机图形学 课程大作业 光线追踪

2018012480 蒋颺

2021 年 1 月 10 日

## 简述

实现了基于光子映射的光线追踪引擎，支持景深、软阴影与纹理映射。同时利用OpenMP进行CPU并行加速，利用层次包围盒与K-D树进行复杂网格求交加速，同样利用K-D树进行光子图碰撞点搜索加速。结果如下：



**Fig. 1.** 光子映射效果图。分辨率为1600x900，采用有限大矩形有向光源照明，总共释放4000000光子，最大迭代深度为20。可以清晰看出刀面、玻璃杯的反射，玻璃杯底的辉映效果以及面光源采样形成的软阴影。

## 代码结构

所有实现基于PA1的代码框架。以下仅对特殊改动做说明：

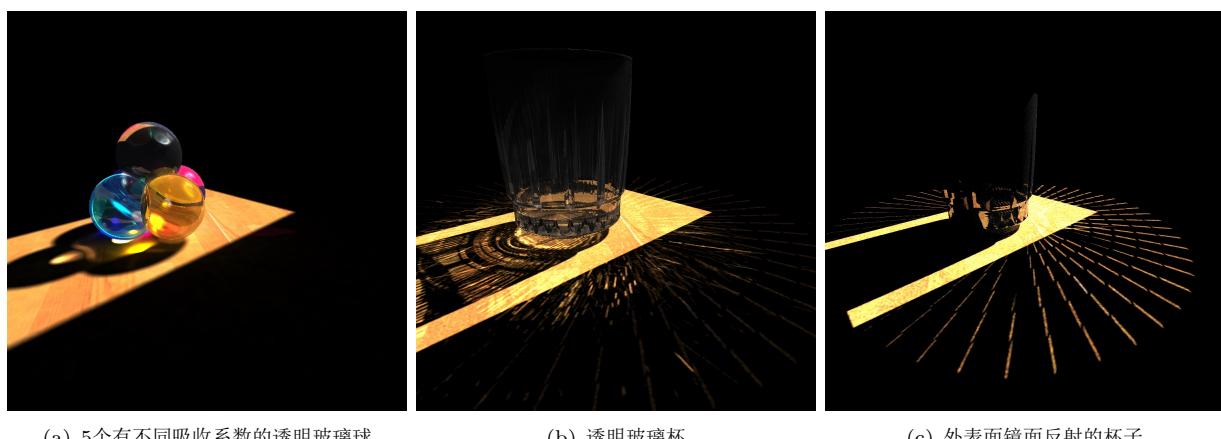
- `include/`
  - `camera.hpp`: 相机类，包含景深参数

- **hit.hpp**: 碰撞类，记录碰撞点位置、法向量和材质，若有纹理同时记录纹理坐标 $\langle u, v \rangle$
  - **light.hpp**: 光源类，包含点光源、有向光源、面光源、矩形有向光源。所有光源类支持光子映射中的光子释放方法。同时，对于面光源与矩形有向光源，支持碰撞检测与软阴影的面光源采样
  - **material.hpp**: 材质类，记录漫反射、镜面反射、折射相关的参数。若有纹理则记录纹理相关参数（映射规则、缩放系数等）
  - **mesh.hpp**: 三角网格类，支持obj、mtl文件的读入，并使用层次包围盒与K-D树加速求交过程
  - **photon.hpp**: 光子与光子图类，使用K-D树加速碰撞点搜索
  - **photonMapping.hpp**: 光子映射核心类，包含正向构建光子图与反向追踪的方法。正向构建光子图与反向追踪皆使用OpenMP进行并行加速
  - **plane.hpp**: 支持有限大四边形求交
- **src/**
    - **mesh.cpp**: 网格、材质读入及层次包围盒方法的具体实现
    - **main.cpp**: 主程序，包含各类初始化、场景文件读入、景深实现与图片存储

所有代码可以通过运行**run\_all.sh**脚本编译并产生**bin/PM**可执行文件。

## 渲染算法：光子映射

代码位于**photonMapping.hpp**与**photon.hpp**中，实现了基本的光子映射算法，构建单张光子图并统计场景中碰撞点的光通量。可以比较好地表现辉映与焦散效果：



**Fig. 2.** 有限大矩形有向光投影图。(a) 总共释放8000000光子，最大迭代深度为50。(b)(c)总共释放4000000光子，最大迭代深度为20。可以看出光线透过玻璃球、玻璃杯产生的辉映与焦散效果。

基础光子映射只生成事先确定光子数量的单张光子图，并通过预先给定的采样数量与采样距离进行光通量统计，这导致了需要反复尝试才能生成一张效果较好的图片的问题。可以通过渐进式光子映射，即迭代产生多张光子图的方法来解决这个问题，但受时间所限，并没有在本次大作业中实现渐进式光子映射。

## 景深

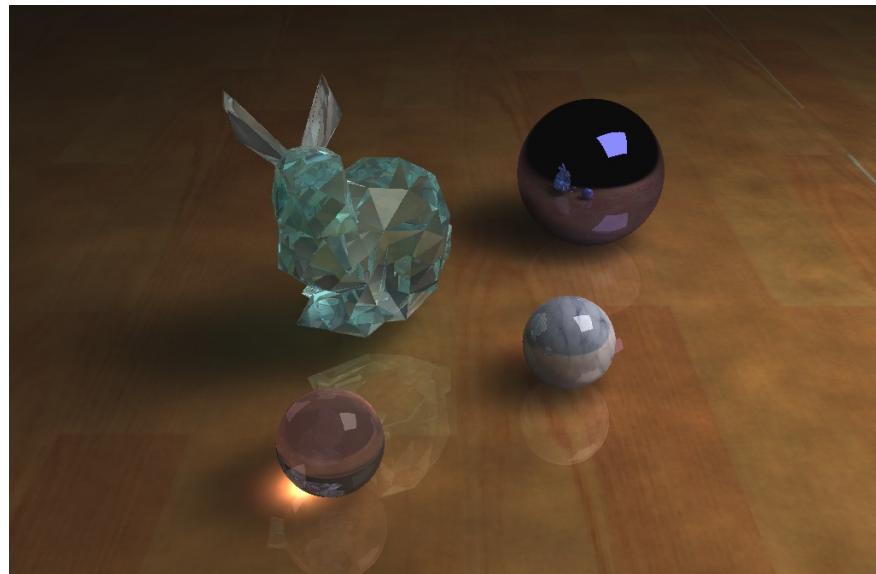
代码位于**main.cpp**与**camera.hpp**中，通过模拟有限大小透镜与焦平面，多次微小扰动后采样取平均实现。效果如下：



**Fig. 3.** 景深效果图。焦平面位于Bunny的脑袋附近，模拟透镜大小0.3，每个像素点采样128次取平均。

## 纹理

代码位于**material.hpp**,**plane.hpp**与**sphere.hpp**中，通过**material**类中记录的纹理映射方向与放缩大小得到物体对应点的颜色。效果参见Fig.1中桌布纹理、兔子照片与下图：



**Fig. 4.** 纹理效果图。具体为无限大平面的木地板纹理与漫反射球的大理石纹理。

## 软阴影

代码位于**light.hpp**中，通过对有限大光源发光位置的随机采样获得软阴影效果。效果如下：



**Fig. 5.** 软阴影效果图。通过对有限大矩形有向光源发光位置的随机采样，得到杯子投影与光源投影边界的软阴影。

## 加速算法

- 层次包围盒，位于**mesh.hpp**与**mesh.cpp**中，对三角面片K-D树中每一颗子树创建包围盒，自上而下通过是否与包围盒相交进行搜索。
- K-D树，位于**mesh.hpp**与**photon.hpp**中，对相交三角面片与碰撞点临近光子的查找理论可以达到 $O(\log n)$ 的时间复杂度。
- OpenMP，位于**main.cpp**与**photonMapping.hpp**中，对构建光子图与反向追踪统计光通量进行并行加速。