

PA1 Report

计 81 包涵 2018011289

2020 年 3 月 30 日

1 算法实现

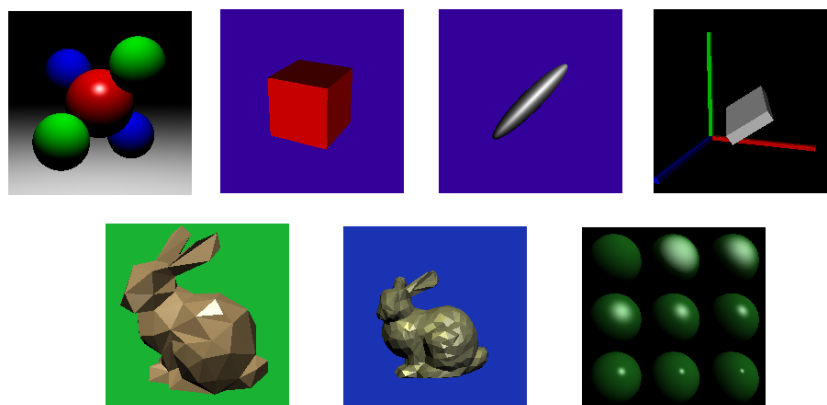
- PerspectiveCamera 类: 按照作业文档中的方法, 把相机坐标系中的方向转换为世界坐标系的坐标即可。
- Object3D:
 - Group 类: 使用 vector 存储 objects; 在 intersect 函数中, 遍历所有的 object 并求交, 把 t 值最小的存储并返回。
 - Plane 类: 构造函数存储平面的法向量和 d 值; intersect 函数直接求解直线和平面的参数方程。
 - Sphere 类: intersect 函数用课件中的方法计算球心到直线的距离, 然后计算弦长和 t 值。
 - Triangle 类: 构造函数中计算法向量, 虽然 mesh.cpp 里已经算了; intersect 函数先算直线和平面的交点, 然后用课件中的方法 (Barycentric Technique) 判断点是否在三角形内。
- Material 类: Shade 函数实现了作业文档中的 Phong 模型公式, 计算了局部的光强。
- main.cpp: 实现程序逻辑如下
 - 从 SceneParser 中读取 Group, Camera 和 NumLights; 从 Camera 中读取画布尺寸。
 - 遍历画布的每个像素, 生成光线与 Group 求交, 如果相交, 再遍历光源, 计算该点的光强; 不相交则把颜色设成背景。

2 遇到的问题

1. 是否需要 normalize 的问题：一开始在 Ray 的构造函数中把输入的参数都 normalize 了，发现不对，应当由构造函数的调用方保证参数的正确性。
2. PerspectiveCamera 的构造函数：使用了未经初始化的 direction 参数，导致一部分图片尺寸有问题，改为使用 `this->direction` 就解决了问题。

3 编译运行与结果

运行 `run_all.sh` 即可编译运行，结果如下



4 其他

没有与同学讨论过。目前应该没有 bug，但是如果有时间，要实现 mesh 求交的加速。