图形学实验1: 光线投射 实验报告

计81 肖光烜 2018011271

光线投射算法逻辑

将屏幕想象成相机的成像平面，每个像素发射一条射线和场景中的物体求交并找到最近焦点，之后计算该点处由光源照射影响产生的局部光强。最终将所有光强加和根据phong模型计算得到该点颜色。

算法主逻辑与提供代码一致，只在细节上进行了些许改动，遂不赘述。

讨论与参考

除了主循环的代码是从作业说明抄的以外，其他每一行代码都是我自己写的和调试的，没有参考任何同学/网上的代码。在实现过程中，仅仅与助教在题意理解上有过简短的交流，并未讨论具体实现方法。

遇到的问题

浮点数误差

由于在本次作业中所有实数均用浮点数存储，因此由于多次计算累计产生的误差最终可能产生无法预料的奇怪效果。在本次试验中就是在图像中会出现孤立的噪点，具体问题体现于：

```
    // 计算CH的长度
    float CH = sqrt(fabs(OC.squaredLength() - OH * OH));
```

以上代码出现在射线与球求交计算OH的步骤中，原始的代码为：

```
    // 计算CH的长度
    float CH = sqrt(OC.squaredLength() - OH * OH);
```

两行代码的唯一差别在于，正确代码在开根内部先求了一遍绝对值。这是由于，当射线非常接近圆心时，OC与OH近似相等。然而OC经过多次的浮点数计算操作，最终在开根内部反而会造成负数开根的状况，导致此处算出无效值。在根号内部加入绝对值就解决了这一问题。

```
    // 平行
    if (fabs(cos)<1e-6) return false;
```

以上代码出现于射线与平面&三角形求交的步骤中，cos表示射线与平面法向量夹角的余弦值。当cos等于0时，射线与平面平行，直接返回不相交。然而此处由于浮点数误差，cos几乎不可能等于0，于是要加上绝对值，并小于某特定小数（1e-6为float浮点数误差极限）。

漏抄加号使兔子变暗

```
// 计算局部光强
finalColor +=
    hit.getMaterial()->Shade(camRay, hit, L, lightColor);
```

主循环中此句的+=我起初漏写+，只写成=。这在单独光源的环境中没有问题，然而对于多光源的两个兔子图，就会造成图像较黑的问题。经过漫长的分析我终于找出了这个问题。

In Triangle Test 的边界问题

In Triangle Test 为检测一个点是否包含于一个三角形的函数。

这个问题与浮点数误差问题类似，也是会在图像上出现噪点。我起初的代码实现为：

```
bool inTriangle(const Vector3f& p) {
    return Vector3f::dot(Vector3f::cross((b - p), (c - p)), normal) > 0 &&
        Vector3f::dot(Vector3f::cross((c - p), (a - p)), normal) > 0 &&
        Vector3f::dot(Vector3f::cross((a - p), (b - p)), normal) > 0;
}
```

这种写法没有考虑射线与三角形相交于边界的情况，当射线与三角形相交与边界会导致误判点不在三角中，产生噪点。

正确的写法为：

```
bool inTriangle(const Vector3f& p) {
    return Vector3f::dot(Vector3f::cross((b - p), (c - p)), normal) >= -1e-6 &&
        Vector3f::dot(Vector3f::cross((c - p), (a - p)), normal) >= -1e-6 &&
        Vector3f::dot(Vector3f::cross((a - p), (b - p)), normal) >= -1e-6;
}
```

将大于号改为大于等于，同时考虑到浮点数误差引入小数1e-8.

使用说明

使用命令行打开到CMakeLists.txt所在目录，输入以下命令：

```
bash run_all.sh
```

Bug

目前没有发现未解决的Bug。

对本次作业的建议

这次作业做起来体验非常好，主要是助教小哥哥人很nice，习题课把所有算法都喂到嘴边了，然后加微信答疑也很耐心~

唯一需要修改的地方：

- 透视相机模型的配图，angle标注具有误导性（我个人认为标错了）。