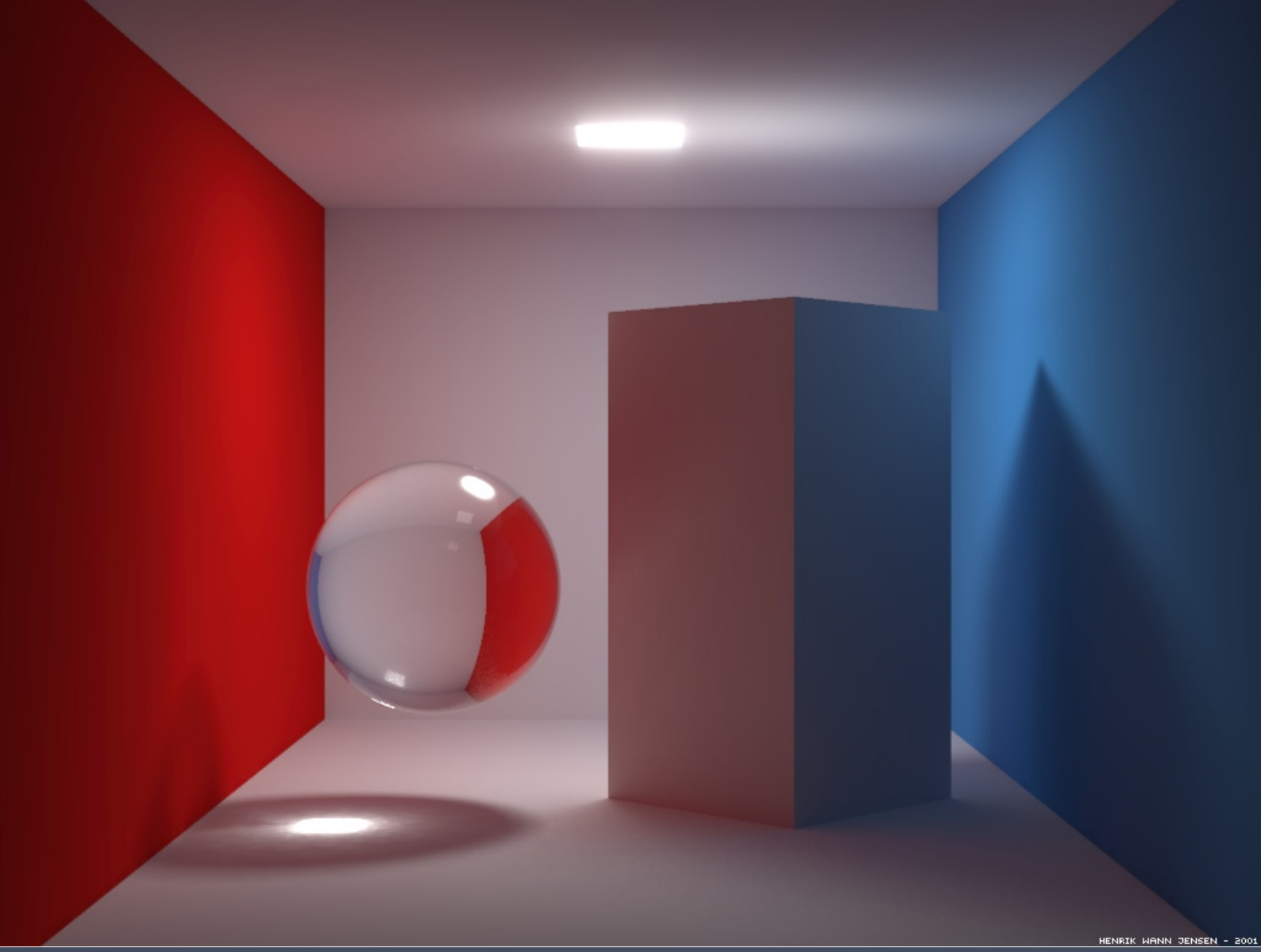

西安交通大学计算机图形学实验文档

局部光照和阴影（渲染部分）

作者：罗思源
组织：计算机图形学课题组
时间：October 10, 2022

目录

序

这部分实验将在一个独立的 OpenGL 工程中实现局部光照和阴影的简单渲染实验。

整体的逻辑见下：

参考 main 函数下的 init() 函数：

- 读取着色器 (fshader:fragment shader 和 vshader:vertex shader)
- 设置光源位置
- 设置物体旋转
- 设置材质
- 传递顶点数据

再参考 main 函数下的 display() 函数：

- 计算相机矩阵
- 传递物体变换矩阵
- 传递 MVP 矩阵（model, view, projection）
- 将材质和光源传递给着色器
- 绘制
- 绘制阴影部分

本文档分为三章：

- **设置相机**
- **实现光照模型和着色**
- **整体框架和其他功能**

该文档的说明较为简略，主要是描述实验需要完成的内容以及相关算法，具体的任务可见于代码注释。

第 1 章 设置相机

1.1 绘制场景与模型

创建 OpenGL 绘制窗口，然后读入三维场景文件（可以使用我们提供的几何体的 *.off 文件）并绘制。为了和后期的阴影颜色区分，可以将窗口背景色设置为灰色。

读取模型的函数在 TriMesh.cpp 中的 readOff() 函数中。遵循 off 文件的读取格式，依次读取顶点数，面片数，边数。并将其储存在 glm 库支持的结构体中即可。

1.2 任务：设置相机

需要补全 Camera.cpp 中对应的函数，实际功能是通过 MVP 矩阵变换，实现从三维空间场景到二维屏幕空间的映射关系。

- updateCamera() 函数（设置相机的位置和方向）
- lookAt() 函数（计算 View 矩阵）
- ortho() 函数（计算正交投影矩阵）
- perspective() 函数（计算透视投影矩阵）

- updateCamera() 函数：通过定义三个变量 eye, at 和 up，来设置摄像机的位置和方向。同时确定摄像机的坐标。需要补充 eye（摄像机相对位置）的 at（朝向）和 up（默认朝上方向），来设置摄像机参数。
- lookAt() 函数：补全摄像机的观察矩阵，即 MVP 矩阵中的 view 矩阵。该函数的输入为 camera 的三个参数，eye, at 和 up。通过这三个参数来计算出 view 矩阵即可。
- ortho() 函数：补全摄像机的正交投影矩阵，即实现将空间场景的物体投影到正方形屏幕空间的功能，输入参数是长方体空间的边界坐标，输出为一个 4x4 的矩阵。
- perspective() 函数：补全摄像机的透视投影矩阵，即实现透视的功能，输入四个参数，输出为一个 4x4 的矩阵。

上述的这些方法实现之后，则会得到一个具有透视关系的场景，接下来需要去做局部光照和着色。

第 2 章 实现光照模型和着色

2.1 任务：Blinn-Phong 光照模型

在完成摄像机控制以及 MVP 矩阵变换的条件下，我们便可以将三维场景中的物体投影在计算机屏幕上了。接下来需要考虑光照对物体的影响，本实验采取较为简单的一种局部光照模型，Blinn-Phong 光照模型 [1]。分别计算环境光，漫反射和镜面反射三种光照，并将三者相加即可。实验要求在 fshader.glsl 中实现 Blinn-Phong 光照模型，包括计算环境光分量，计算系数和不同漫反射分量，最终再合并三个分量，修正透明度。

下面简单介绍这三种光照：

- **环境光**（整体环境中的光，可以表示出整个物体的轮廓）
- **漫反射**（不考虑观察者位置，只根据光源方向以及法线向量来计算光照强度，可以表示出物体的体积感）
- **镜面反射**（考虑观察者位置，计算反射方向和观察者之间的夹角来修正光强，可以表示出物体的高光）

最终将三种光照相加，就可以得到一个比较符合物理的局部光照模型。

而在代码的实现中，也需要在计算完光照强度后，将光照强度相加并存储在 fcolor 中。

而注意 Blinn-Phong 模型则需要在 Phong 模型的基础上引入半角向量的概念来代替镜面反射的夹角来提升计算速度。

2.2 任务：着色

实现完光照模型后，需要再通过着色来增加质感和真实度。着色的核心便是更新法线向量。将原始光照模型中以面片的单位的光照系统更改为以面片中的点为单位的光照系统，来增强真实感。本次实验要求实现的 phong shading 便是利用顶点法向与重心坐标来插值出每个点的法线向量，顶点的法线向量的求解以及重心坐标我们已经在 TriMesh.cpp 中实现，仅需要实现法线向量的变换。

第 3 章 整体框架和其他功能

3.1 工程框架

工程目录介绍:

```
rendering-lab
├── include
│   └── Angel.h(环境检查)
│   └── Camera.h(摄像头位置与视角)
│   └── CheckError.h(环境检查)
│   └── TriMesh.h(顶点相关信息)
├── shaders
│   └── fshader.glsl(fragment shader): 需要做phong shading
│   └── vshader.glsl(vertex shader)
├── Camera.cpp(各种视角的具体实现)
├── InitShader.cpp(初始化shader)
├── main.cpp(实现各种基本交互功能和gui框架)
├── TriMesh.cpp(实现各种变换以及数据传递)
└── CMakeLists.txt(Cmake配置文件)
```

3.2 交互方式

在运行完程序后，也可以通过鼠标和键盘来控制摄像头和光源的位置。

通过”u/U” 控制摄像头横向旋转。

通过”i/I” 控制摄像头上下移动。

通过”o/O” 控制摄像头远近移动。通过 “1/2/3” 来切换模型参数。

通过鼠标左键可以设定光源位置。

参考文献

[1] Wiki. *Phong Shading*. URL: https://en.wikipedia.org/wiki/Phong_shading.