**深 圳 大 学 实 验 报 告**

**课程名称：­ 计算机图形学**

**实验项目名称： 实验三 光照与阴影**

**学院： 计算机与软件**

**专业： 计算机科学与技术**

**指导教师： 熊卫丹**

**报告人： 杜良衡 学号：2022150255 班级： 01B**

**实验时间： 2024.11.20**

**实验报告提交时间： 2024年 11 月 23 日**

**教务部制**

|  |
| --- |
| 实验目的与要求：   1. 掌握OpenGL三维场景的读取与绘制方法，理解光照和物体材质对渲染结果的影响，强化场景坐标系转换过程中常见矩阵的计算方法，熟悉阴影的绘制方法。 2. 创建OpenGL绘制窗口，读入三维场景文件并绘制。 3. 设置相机并添加交互，实现从不同位置/角度、以正交或透视投影方式观察场景。 4. 实现Phong光照效果和物体材质效果。 5. 自定义投影平面（为计算方便，推荐使用y=0平面），计算阴影投影矩阵，为三维物体生成阴影。 6. 使用鼠标点击（或其他方式）控制光源位置并更新光照效果，并同时更新三维物体的阴影。 |
| 实验过程及内容：  **1. 绘制场景、模型**  参考实验2.2内容来读入三维模型。TreMesh.cpp中的read\_off函数实现了读取OFF文件中三维模型的信息。在main.cpp的main函数中调用该函数来读取指定的OFF文件，此处为保存三维球体的sphere.off文件。      为了和后期的阴影颜色区分，在display函数中利用glClearColor函数将窗口背景色设置为灰色。    display函数中会根据物体的变换矩阵来绘制三维模型，其中着色器isShadow设置为0是表示正常绘制的颜色，用来与阴影作区分。    **2. 设置相机**  参考实验3.1，先在lookAt函数中先初始化一个单位矩阵c作为相机观察矩阵。    接着需要根据观察平面法向量VPN和观察正向向量VUP计算得到的3个向量n、u和v来更新相机的观察变换矩阵，也就是相机的局部坐标系。先计算出VPN再归一化得到向量n，再通过VUP和VPN生成与它们都垂直的方向向量u，最后计算得到VUP在照相机胶片平面上的投影v。这里的函数normalize和cross分别表示向量的归一化与叉乘。    然后按照下图更新相机的观察矩阵c      因为在最开始，需要将相机从坐标原点移动到视点，所以要更新观察矩阵的平移部分，这里通过将原始矩阵加上第四个维度，即齐次坐标的方式来实现。    在updateCamera函数中根据角度与距离半径更新计算相机eye的位置，用于lookAt函数。根据下图利用三角函数与距离半径radius计算相机的x、y、z坐标，注意要将角度upAngle和rotateAngle转为弧度。      接着用计算得到的坐标更新相机位置，并设置相机的VUP方向与世界坐标系y方向相同(方向朝上)，相机的参考点（at）指向被观察物体的中心。这样就能保证相机围绕被观察物体旋转。    在计算正交投影的ortho函数中初始化一个单位矩阵c作为正交投影矩阵。    在OpenGL中使用的正交投影是定义在一个平行六面体的视景体中。如下图所示，该六面体由六个参数决定，分别为左右裁剪平面（left和right），上下裁剪平面（top和bottom），远近裁剪平面（near和far）。    而在OpenGL渲染过程中，需要将定义的正交投影视景体通过平移和旋转变换到标准视景体中。因此，需要按照下图所示矩阵来更新正交投影矩阵c，并返回最终得到的矩阵。      在计算透视投影的perspective函数中先初始化一个单位矩阵c作为透视投影矩阵。    对于透视投影也需要设置视景体来裁剪三维物体。在OpenGl中可以通过视域（FoV，Field of View）来定义视景体，其由视角（Field of View），投影平面长宽比（aspect）和远近裁剪平面（near和far）决定。    而透视投影同样也需要执行投影规范化过程。按照下图所示矩阵来更新透视投影矩阵c，并返回最终得到的矩阵。      **3. 添加光照和材质效果**  参考实验3.4，来实现Phong光照效果。在computeTriangleNormals函数中，先调用了resize函数给存储面片法向量的容器face\_normals分配了一个和faces一样大的空间。接着遍历所有三角面片，对于每个三角面片，获取其三个顶点。    然后根据以下公式来计算该面片的法向量，使用cross函数来实现向量的叉乘。再调用normalize函数将法向量归一化，并将最后的值存储到face\_normals容器中。      在computeVertexNormals函数中，会先判断面片法向量是否已计算，没有的话就会调用刚才完成的computeTriangleNormals函数来计算。接着为vertex\_normals容器分配一个和vertex\_positions一样大的空间，用来存储顶点法向量，并初始化法向量为0。    然后遍历所有三角面片。对于给定顶点所在面片的法向量，顶点的平均法向量为法向量的和。因此先累加面的法向量，再对累加法向量归一化即可得到顶点法向量。      计算好法向量后，与顶点坐标类似，需要将其数据传递给着色器。为此增加了一个法向量变量vNormal以及记录法向量变量位置的nLocation。在main.cpp的bindObjectAndData函数中，通过glBufferSubData函数将法向量数据更新到顶点缓冲区对象中。    再仿照顶点坐标vPosition的写法，从顶点着色器中初始化顶点的法向量。这段代码先调用glGetAttribLocation函数获取顶点着色器中法向量的位置索引，接着以nLocation作为参数，启用法向量属性数组，以便渲染时传递给着色器。最后设置了顶点属性指针，指定要修改的顶点属性为nLocation，还包括是否需要归一化、属性之间偏移量等参数信息。    在fshader.glsl的main函数中，需要计算Phong反射模型涉及到的四个向量并归一化。如下图所示，p为三维物体表面上一点，要计算的四个向量中，N表示p点法向量，L是从点p指向光源位置的向量，V是从p点指向相机的向量，R是沿着L方向入射光线按照反射定律的出射方向。需要注意计算中要保证这些向量都已经归一化。    另外需要注意，要在相机坐标系下来计算所有向量，因为在相机坐标系下，原点就是相机位置/眼睛位置。因为在vshader文件中顶点坐标和光源坐标已经转换到相机坐标系，所以此处只需将法向量进行转换即可。    而利用转换后的顶点坐标、光源坐标和法向量，再结合归一化向量的函数normalize以及依据入射向量和法向量计算反射向量的函数reflect，就能够求得所需的四个归一化向量。    再依据以下公式，计算漫反射分量和镜面反射分量。其中反射分量的计算是通过反射系数乘上光源以及环境的反射光参数。Dot和max函数作用分别是向量点积和取两者最大值。        由于这里假设衰减系数，所以将这三个部分的颜色分量累加就形成了最终在三维物体表面上每个顶点的颜色，颜色相加后最后一维的透明度需要进行修正。      在main.cpp的init函数中初始化光源的位置以及各反射参数。    选用翠绿色的材质参数进行绘制。    找到init函数中对应设置材质的部分，材质由环境光、漫反射光、镜面反射光、高光系数参数组成。修改该部分的各参数。    **4. 添加阴影效果**  在main.cpp的display函数中，先创建3个float类型的变量来分别存储光源位置light\_position的x、y、z坐标。    假定光源位置在，物体由三角形表示，投影平面上的黑色三角形区域即为阴影，称之为阴影多边形。假设三角形任意一个顶点坐标为，投影到投影平面之后的坐标为，因为该点在 平面上，所以 。根据比例关系可得如下公式：  求解可得，  同理可得和。  为了能够方便地通过矩阵表示出投影关系，将所有坐标设置在齐次坐标系下，投影关系就能表示成如下公式。    最后可得到：  由此，仿照上述公式中的矩阵来初始化阴影投影矩阵shadowProjMatrix，并更新三角形的物体变换矩阵modelMatrix。    接着将isShadow变量设置为1，表示绘制阴影，再传递矩阵数据给着色器中的统一变量。最后绘制三角形的阴影以及投影平面。    **5. 交互控制光源位置并更新阴影**  参考实验2.1，使用鼠标点击控制光源位置并更新光照效果，并同时更新三维物体的阴影。    另外需要为材质material设置交互：通过点击数字键1~9增减反射系数，0增减高光指数。  首先，可以仿照已完成的通过数字键1~3控制环境光系数部分来实现数字键4~6控制漫反射系数diffuse。先接受反射系数并取其x、y或z坐标轴上的分量，接着进行增减，同时保证增减后的值合理，用max、min函数实现。更新后设置回对象中。    用同样的方式实现数字键7~9控制镜面反射系数specular。    然后实现数字键0增减高光系数。该事件内用float类型的shininess接收高光指数，进行增减后调用clamp确认是否在指定范围内，保证数据合理，再设置回对象中。    **6. 最终效果展示**  注意需要点击i键来调整向上角度后才能看到阴影 |

|  |
| --- |
|  |

深圳大学学生实验报告用纸

|  |
| --- |
| 实验结论：  通过这次实验，我掌握了如何读取OFF文件中三维模型的信息，还有光照和物体材质对渲染结果的影响。深入了解了Phong光照模型的工作原理，包括环境光、漫反射光和镜面反射光的计算方法。通过设置不同的光照参数和物体材质属性，显著改变渲染结果，使物体表面呈现出不同的视觉效果。这不仅增强了对光照和材质的理解，还提高了我在实际项目中应用这些知识的能力。  为实现从不同位置和角度观察物体的效果，我掌握了模型矩阵、视图矩阵和投影矩阵的计算方法，并了解了它们在场景坐标系转换中的作用。还实现了使用鼠标点击控制光源位置的功能，以及通过键盘来调整反射参数，使光照呈现不同效果。  实验中，我也遇到了不少问题，例如阴影的绘制。我发现生成阴影总是与物体有重合。经过多次调试，最终通过修改光源位置与物体的旋转位移使得阴影能够正确地呈现与变换。  这次实验中，我不仅巩固了理论知识，还通过实际操作加深了对OpenGL和计算机图形学的理解，能够利用代码实现越来越复杂的图形效果呈现与变换。 |
| 指导教师批阅意见：  成绩评定：  指导教师签字：  年 月 日 |
| 备注： |

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后10日内。