山东大学 计算机科学与技术 学院

计算机图形学 课程实验报告

学号：202200130119	姓名：于斐	班级：学堂计机 22
实验题目：平滑旋转		
实验学时： 4	实验日期： 2024 年 11 月 23 日	

实验目的：

通过编程实现基于四元数的 3D 模型平滑旋转，掌握四元数在 3D 计算中的优势及其在姿态插值中的应用。通过设置模型的初始和终止姿态，并沿给定路径实现平滑的位姿过渡，深化对 3D 空间变换、插值计算及其视觉效果的理解。结合交互界面的开发，培养对复杂 3D 模型控制的综合设计能力，为计算机图形学领域的学习奠定技术基础。

实验步骤与内容：

实验环境：OpenGL 4.6 及 GLFW，GLM 等附属库。

实验步骤：

0. 与实验 1 完全相同：项目使用 CMake 管理，原则上任何支持 CMake 的编辑器均可使用。所有未包含在项目文件中的库均使用 CMake 的 FetchContent 导入，不需额外手动安装任何库。

   程序自动生成到 dist 目录下，但运行时 pwd 需包含 assets。因此需要在根目录运行 dist/Renderer 或将 assets 拷贝到 dist 下。

   a) 建立 OpenGL 窗体及 argument parser，config parser。前者代码位于 Renderer::init()中，在 src/renderer/renderer.h，renderer.cpp 下可以找到。后者代码位于 src/utils 中。使用 ImGUI 构建交互选单，用于控制程序运行时的行为。相关代码位于 src/gui/ 下。

   b) 构建 Model 类用于从 .obj 中导入模型。构建 Shader 类用于加载 Shader Program。构建 Camera 类，用于变换相机。

1. 构建物体变换方式。项目中构建了 Motion 类，支持 vec3 形式的 translation，scale 和 quaternion 形式的 rotation。

   相关数据存储为 Motion 类。定义可在 src/animation/animation.h 中找到。

```
struct Motion {
    glm::vec3 translation;
    glm::quat rotation;
    glm::vec3 scale;

    void reset() {
        translation = glm::vec3(0.0f, 0.0f, 0.0f);
        rotation = glm::quat(1.0f, 0.0f, 0.0f, 0.0f);
        scale = glm::vec3(1.0f, 1.0f, 1.0f);
    }
};
```

2. 构建 Animation 类，其中主要包括 std::vector<std::pair<Motion, float>> 用于存储关键帧。同时实现一个计时函数和插值函数。给定当前时间，可以通过相邻的两个关键帧插值出当前结果。

```cpp
class Animation {
public:
    Animation();
    ~Animation();

    void addKeyframe(Motion keyframe, float time);
    void update(float deltaTime);
    void reset();
    void clearKeyframes();
    void removeKeyframe(int index);
    std::vector<std::pair<Motion, float>>& getKeyframes();

    Motion getCurrentMotion();
    float getCurrentTime() { return currentTime; }
    int getCurrentKeyframe() { return currentKeyframe; }
    void setCurrentTime(float time);

    float getMaxTime() { return maxTime; }
```

对计时函数，renderer 在处理输入时有使用 glfwGetTime() 计算 delta time between frames，因此直接使用 update 函数将其传入 Animation 类即可。

对插值函数，程序首先找到当前时间所处在的两个相邻关键帧。

对 translation 和 scale，使用线性插值方法，直接数值计算比例并混合两个关键帧对应的 motion。

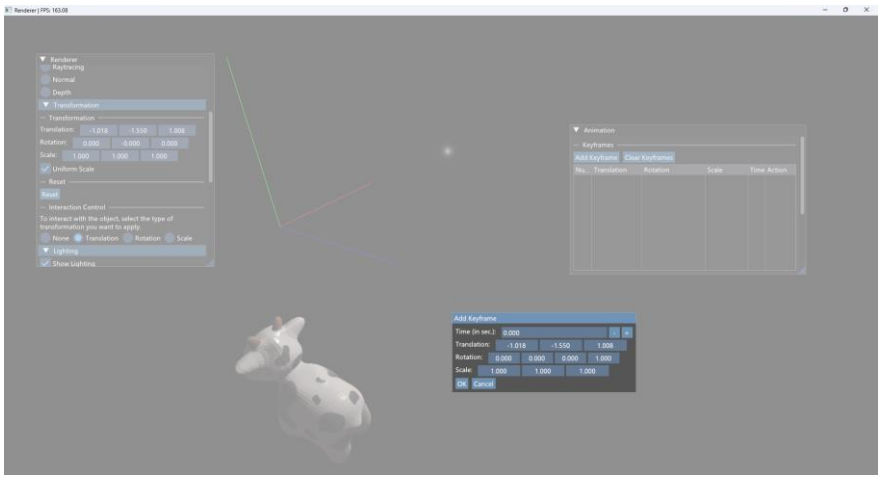
对 rotation，使用球面线性插值方法 slerp。对于两个 rotation quaternion $\mathbf{q}_1, \mathbf{q}_2$，相对时间 $t \in [0,1]$，球面线性插值函数可以被如下定义：

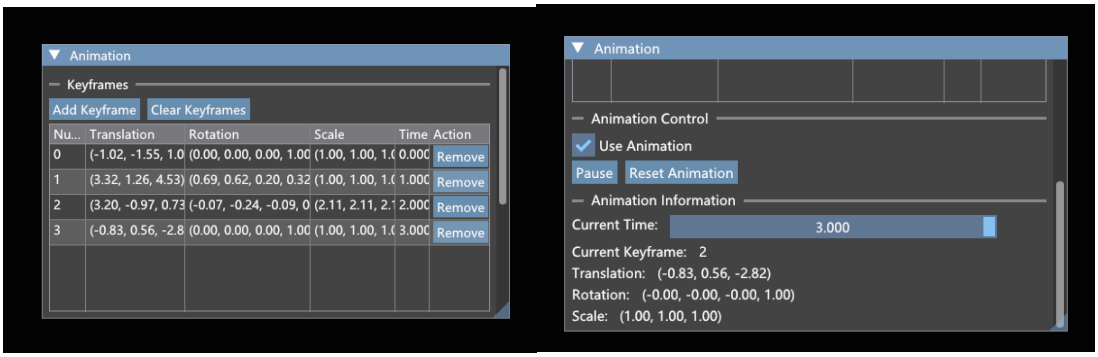
$$\mathbf{q}(t) = \frac{\sin\big((1-t)\theta\big)}{\sin\theta}\mathbf{q_1} + \frac{\sin(t\theta)}{\sin\theta}\mathbf{q_2}$$

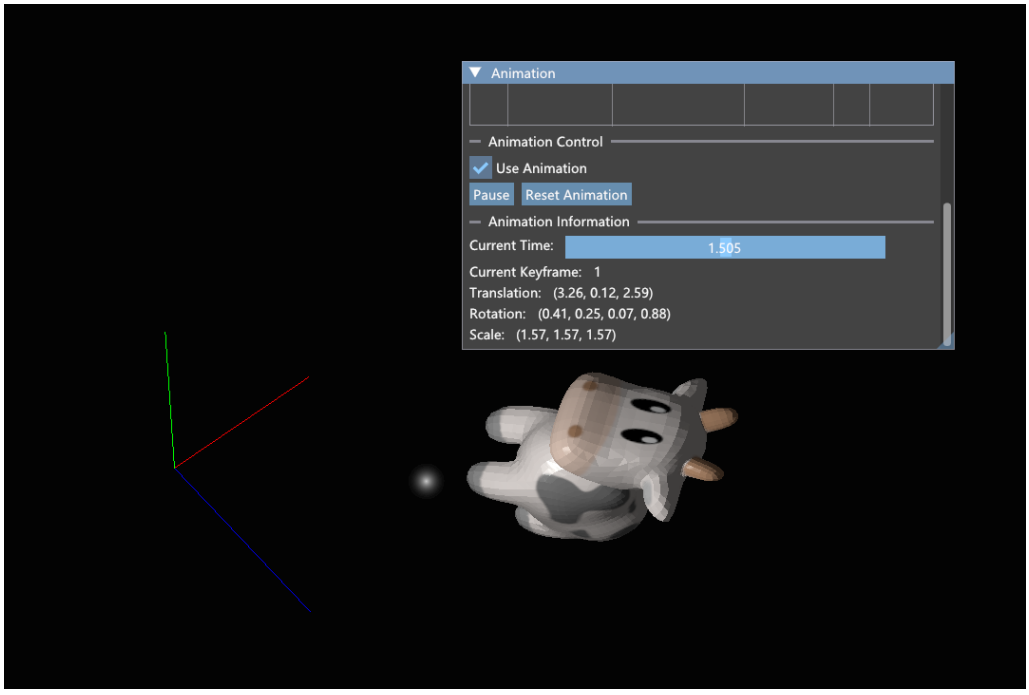
其中 $\theta = \arccos(\mathbf{q_1} \cdot \mathbf{q_2})$。

三者混合后即可得到平滑结果。

3. Demo







实验总结：

通过本次实验，成功实现了 3D 模型的平滑旋转和姿态过渡，深入理解了四元数在 3D 图形学中的重要性。实验过程中，通过正确定义模型的起始和终止姿态，掌握了姿态表示的方法及其与四元数的关系。进一步，通过实现沿路径的平滑旋转和平移，体会到四元数在插值计算中的平滑性和高效性，解决了传统欧拉角表示中的万向节锁问题。结合交互界面的设计与实现，增强了对用户友好操作的理解和实践能力。