

Homework9 Program Report

张景浩 PB20010399

2023.5.20

1 问题描述

完成 OpenGL 的纹理映射，使用作业 5 中参数化结果作为纹理坐标，以 brick2.jpg 图像作为纹理，给作业 6 中的曲面添加纹理。

2 算法原理

2.1 OpenGL 纹理映射

我们使用 OpenGL 库中相关函数完成纹理映射。通过参数化映射确定曲面顶点在平面中的纹理坐标，在光栅化阶段通过对纹理坐标进行插值计算，得到曲面中每个对应像素点需要的纹理像素。步骤如下：

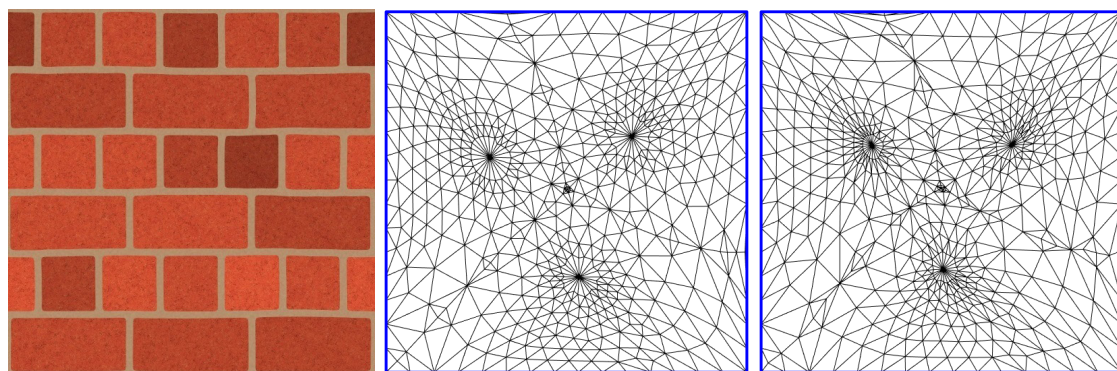
1. 创建一个纹理对象并加载纹理数据
2. 为顶点数据增加纹理坐标 (参数化坐标)
3. 在着色器中使用纹理坐标从纹理中取样得到像素颜色

在具体实现中，我们使用 `MyMesh::upload` 函数传入网格顶点坐标和纹理坐标，使用 `MyImage::MyImage` 和 `GLTexture::setImage` 传入纹理图像，最后将 `GLMesh::showTexture` 的值修改为 `true` 完成纹理映射。

3 代码实现

本次实验使用 visual studio 进行编译，代码框架来自 assignment5 和 assignment6，算法具体的实现详见 `main.cpp` 文件。纹理坐标使用作业 5 的代码完成预处理。

4 测试结果

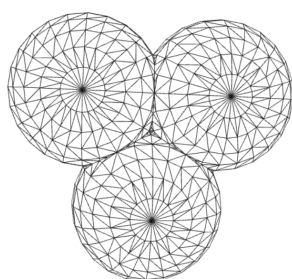


纹理

均匀权重参数化纹理坐标

floater 权重参数化纹理坐标

图 1: 完成纹理映射的准备



网格曲面

均匀权重参数化纹理映射

floater 权重参数化纹理映射

图 2: 纹理映射结果

5 总结

通过观察实验结果，我们发现通过 OpenGL 的纹理映射可以很好地将纹理图像贴到曲面之上，并且可以明显地看出不同地参数化坐标对于纹理映射结果的影响。对于具有保形性质的 floater 权重参数化而言，得到的纹理映射结果相比均匀权重参数化保持了墙面纹路在球体曲面上的弯曲，更加符合预期参数化效果。