

---

# 实验四

## Minimal Surface & Mesh Parameterization

---

ID: 58 陈文博

March 18, 2020

### 1 实验要求

- \* 初步理解 \*.obj 数据 (\*.obj, \*.mtl)
  - 安装 MeshLab 查看三维数据文件
- \* 学习网格的数据结构及操作
  - 使用 MeshFrame 框架
  - 寻找非封闭网格曲面的边界
- \* 实现极小曲面与网格参数化
  - 极小曲面：边界固定，求解方程组
  - 参数化：边界映射到平面，求解方程组
- \* 巩固使用 Eigen 库求解稀疏线性方程组

## 2 开发环境

**IDE:** Microsoft Visual Studio 2019 community

**CMake:** 3.16.3

**Qt:** 5.14.1

**Eigen:** 3.3.7

**Assimp:** 5.0.1

**tinyxml2:** 8.0.0

**Others**

### 3 算法原理

#### 3.1 极小曲面 Minimal Surface

定义：平均曲率处处为 0 的曲面

算法：

由于曲面的平均曲率处处为 0，即：

$$H(v_i) = 0, \forall i \quad (3.1)$$

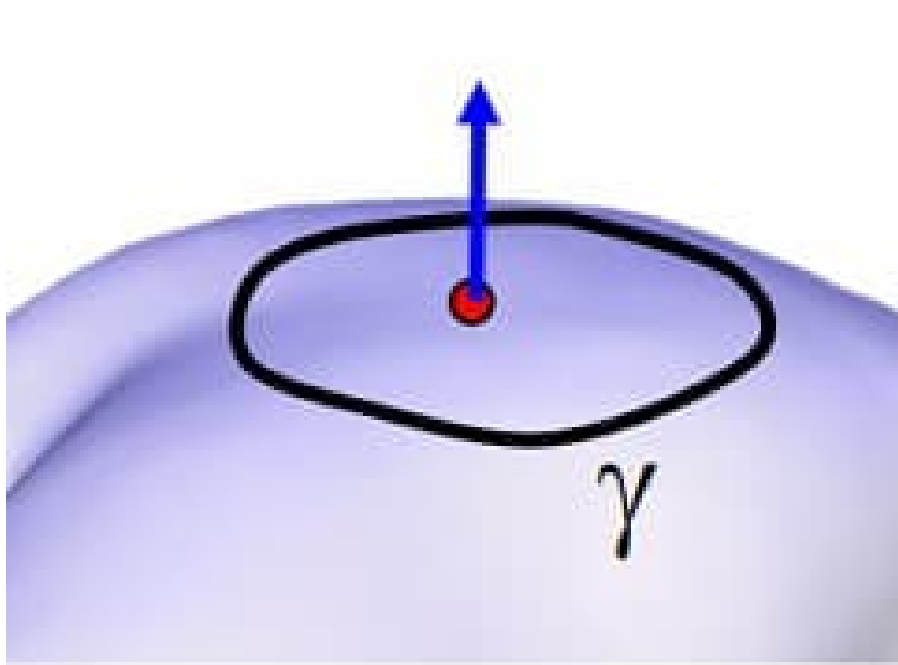


Figure 3.1: 曲率图示

如图，有：

$$\lim_{len(\gamma) \rightarrow 0} \frac{1}{len(\gamma)} \int_{v \in \gamma} (v_i - v) ds = H(v_i) \mathbf{n}_i \quad (3.2)$$

微分坐标为：

$$\delta_i = v_i - \frac{1}{d_i} \sum_{v \in N(i)} v = \frac{1}{d_i} \sum_{v \in N(i)} (v_i - v) = 0 \quad (3.3)$$

固定边界点，通过求解稀疏方程组可以得到最小平面的顶点坐标

### 3.2 网格参数化 Mesh Parameterization

将网格边界映射到凸多边形（如单位圆、正方形）上，对每个内部点生成关于其领域点  $N(i)$  的重心坐标  $(\lambda_{i1}, \lambda_{i2}, \dots, \lambda_{id_i})$ ，利用重心坐标得到：

$$v_i - \sum_{j \in N(i)} \lambda_{ij} v_j = 0, i = 1, \dots, n \quad (3.4)$$

其中  $n$  为内部点数量，通过求解该稀疏方程组，可得到曲面参数化后的坐标，对于不同的重心坐标求取方法，得到的参数化网格也不同。利用添加参数化后的纹理坐标连接纹理图像，可实现纹理映射，如下图，不同重心坐标纹理映射效果也不同：

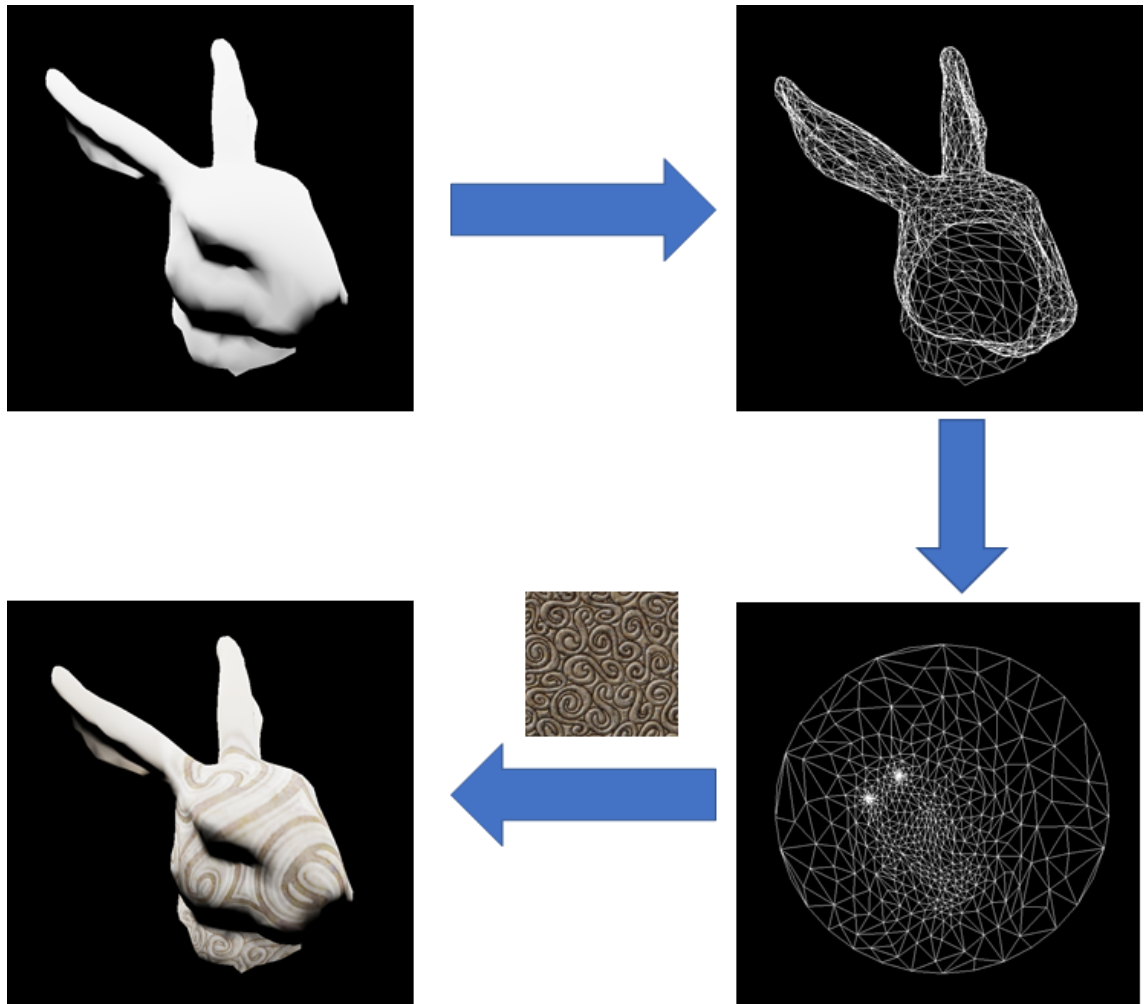


Figure 3.2: 纹理映射

### 3.2.1 Uniform

均匀重心坐标，只考虑内部点的度而不考虑与其邻接点的关系，即取重心坐标：

$$\lambda_{ij} = \frac{1}{d_i}, d_i = |N(i)|, j \in N(i) \quad (3.5)$$

每个邻接点权重相等

### 3.2.2 Cotangent

余切重心坐标，考虑了内部点与邻接点的位置关系，如下图所示，设某一内部点  $v$  顺次连接的相邻点为  $v_{i-1}$ ,  $v_i$ ,  $v_{i+1}$ ，设  $\beta_{i-1} = \angle vv_{i-1}v_i$  和  $\gamma_i = \angle vv_{i+1}v_i$

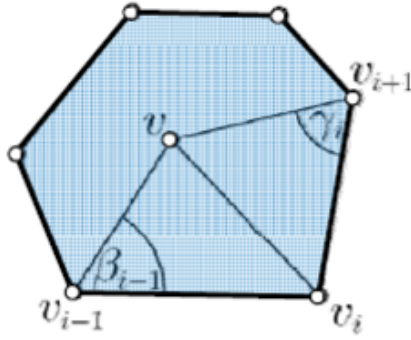


Figure 3.3: discrete harmonic coordinates

重心坐标如下：

$$\lambda_j = \frac{w_j}{\sum_{j \in N(v)} w_j} \quad (3.6)$$

$$w_j = \cot(\beta_{i-1}) + \cot(\gamma_i), j \in N(v) \quad (3.7)$$

## 4 设计难点与解决

### 4.1 UEngine 框架的使用

作业提供的 UEngine 框架提供了作业需要的模型加载渲染显示等基础功能，作业的极小曲面和网格参数化算法通过 `MinSurf` 和 `Parameterization` 两个类实现，同时在 `attribute` 中添加多个按钮以方便交互

### 4.2 度为 2 的边界顶点的问题

当网格中出现度为 2 的边界顶点时，若使用正方形作为参数化的固定边界模式可能会发生崩溃，例如测试模型中的 `bunny_head.obj`（如下图）

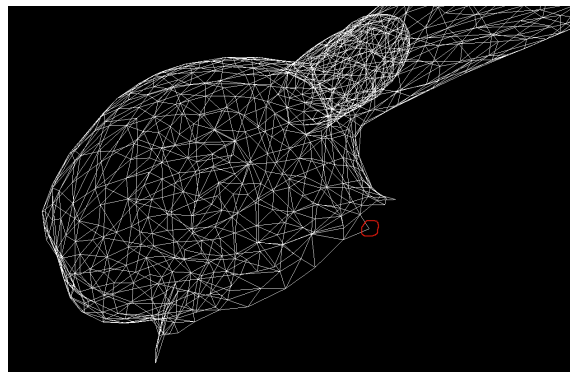


Figure 4.1: `bunny_head.obj` 中的特殊顶点

当展开到正方形边界时，可能会出现以下情况：

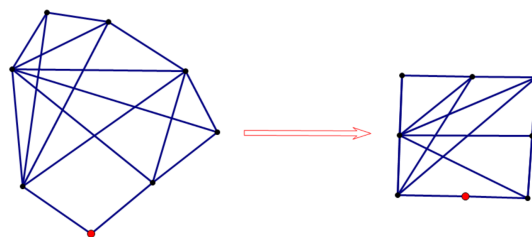


Figure 4.2: 特殊情况

这种情况将发生错误，实际编程中仍能够渲染出纹理贴图，但查看参数化网格图时会出现报错，由于时间原因暂无好的解决方法。

## 5 实验效果

### 5.1 极小曲面 Minimal Surface

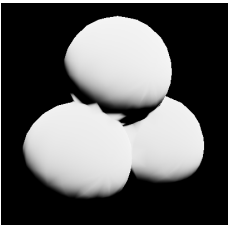
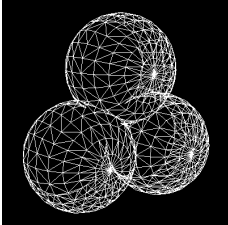
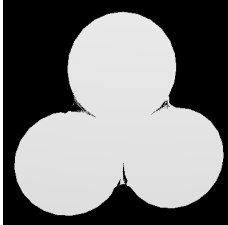
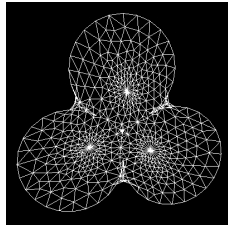
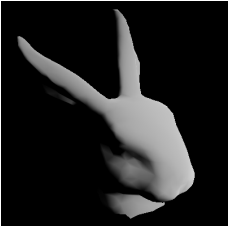
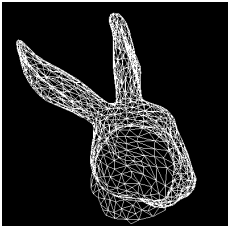
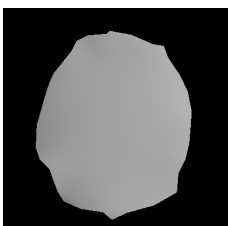
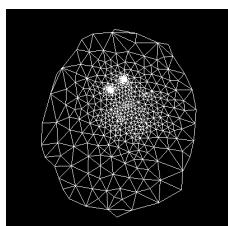
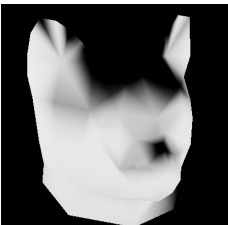
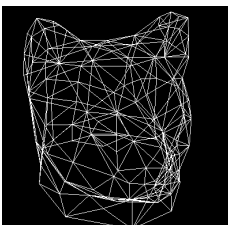
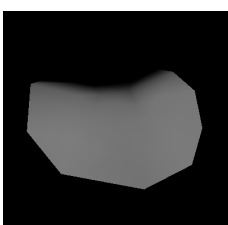
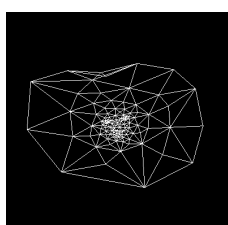
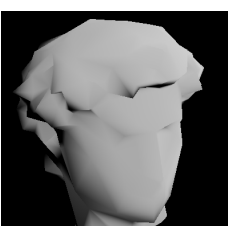
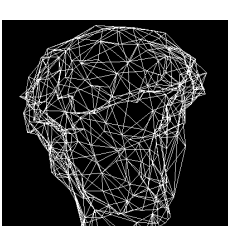

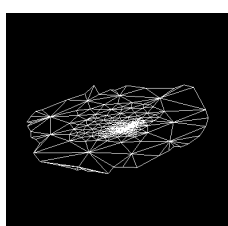
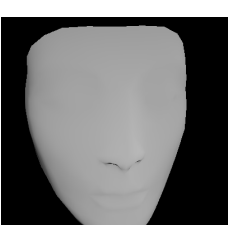
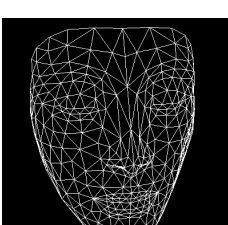
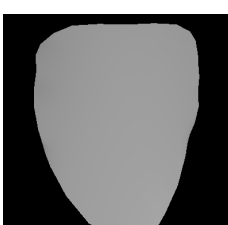
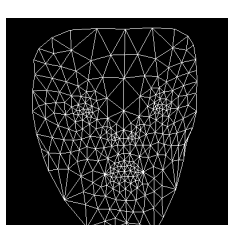
| 原曲面   | 原网格   | 极小曲面   | 极小曲面网格  |
|---|---|--|---|
|    |    |    |    |
|    |    |    |    |
|   |   |   |   |
|  |  |  |  |
|  |  |  |  |

Table 5.1: 极小曲面

## 5.2 网格参数化与纹理映射

### 5.2.1 *Uniform* 权重 & 单位圆边界

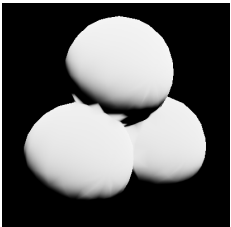
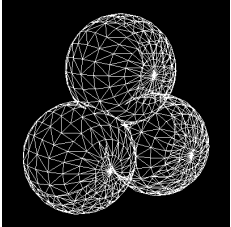
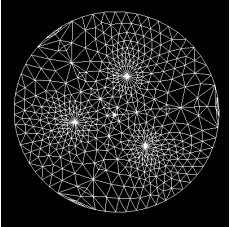


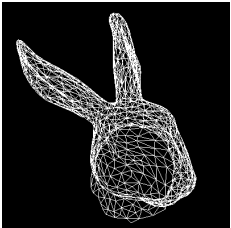
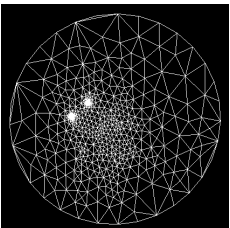
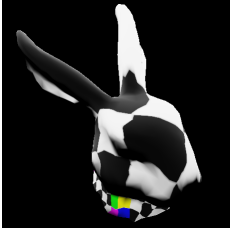
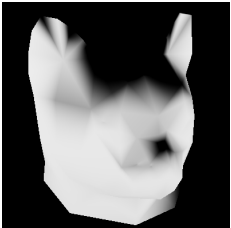
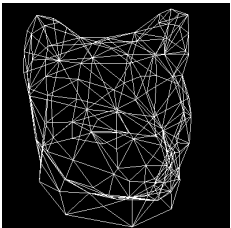
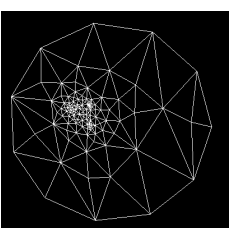
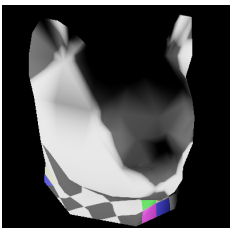
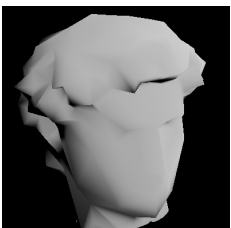
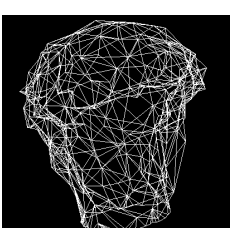
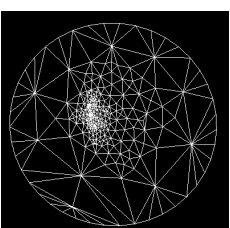
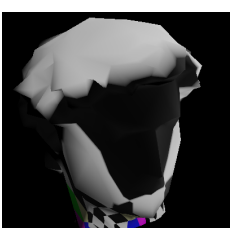
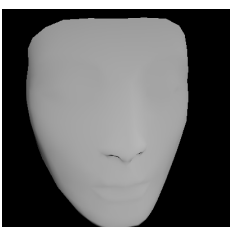
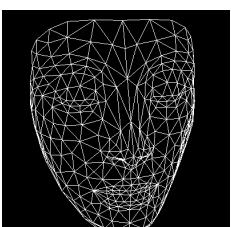
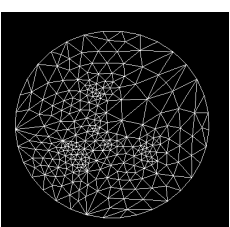
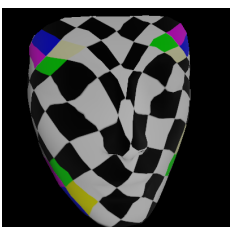
| 原曲面   | 原网格   | 参数化网格  | 纹理映射  |
|---|---|--|---|
|    |    |    |    |
|    |    |    |    |
|   |   |   |   |
|  |  |  |  |
|  |  |  |  |

Table 5.2: *Uniform* 权重，单位圆边界的网格参数化与纹理映射



### 5.2.2 Uniform 权重 & 正方形边界

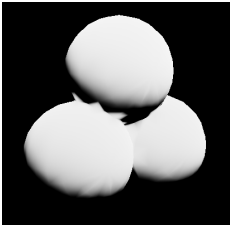
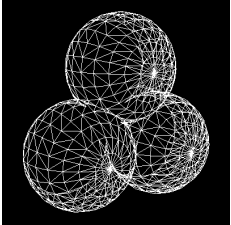
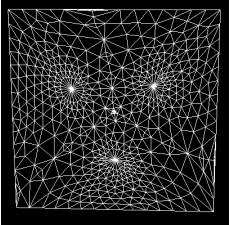


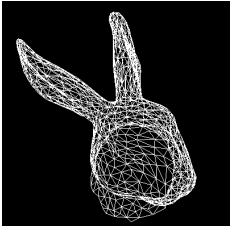
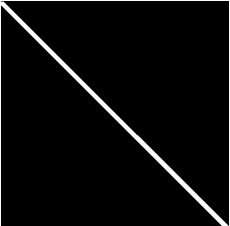

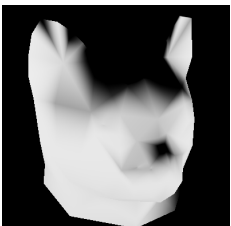
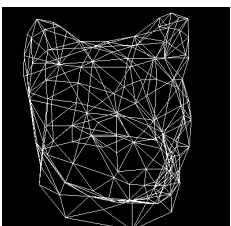
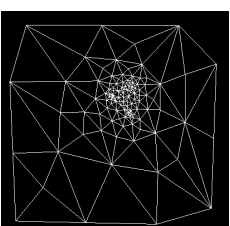
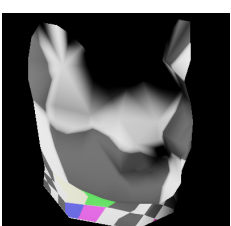
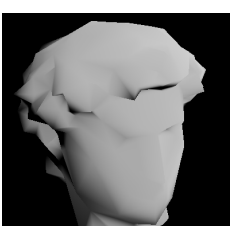
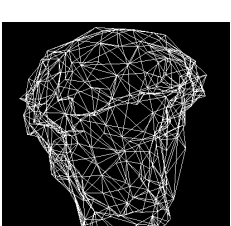
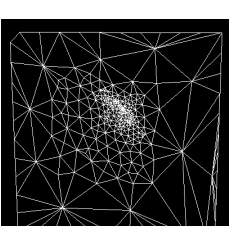
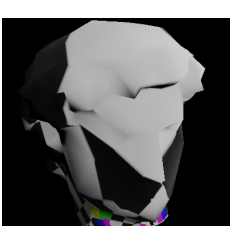
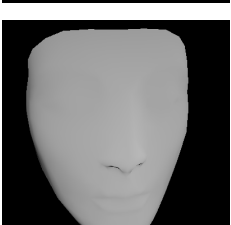
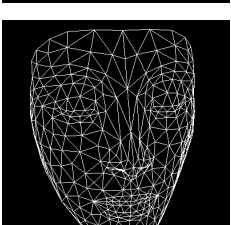
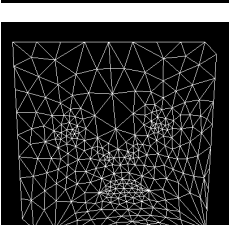
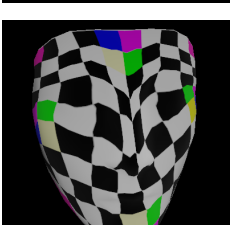
| 原曲面   | 原网格   | 参数化网格  | 纹理映射  |
|---|---|--|---|
|    |    |    |    |
|    |    |    |    |
|   |   |   |   |
|  |  |  |  |
|  |  |  |  |

Table 5.3: Uniform 权重，正方形边界的网格参数化与纹理映射

### 5.2.3 Cotangent 权重 & 单位圆边界

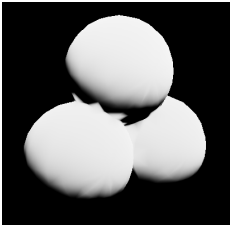
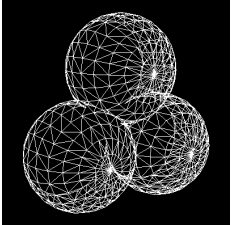
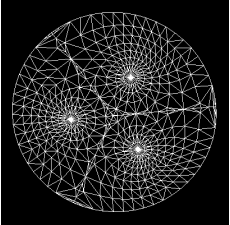


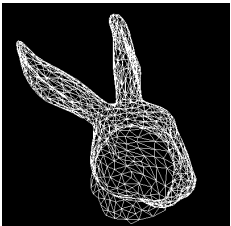
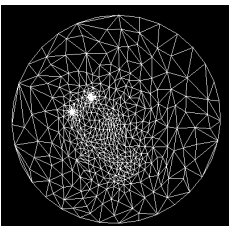
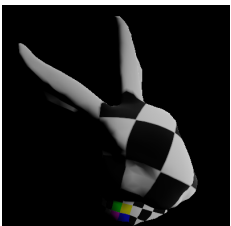
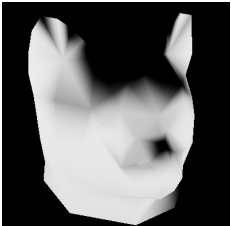
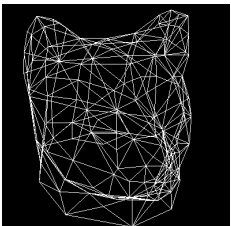
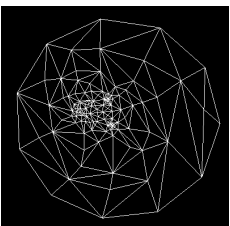
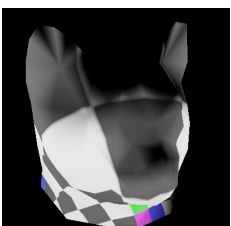
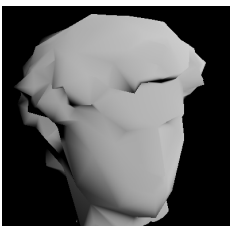
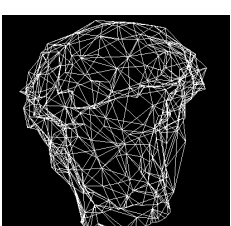
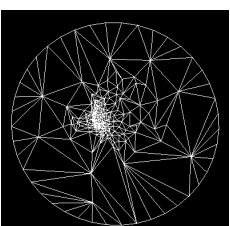
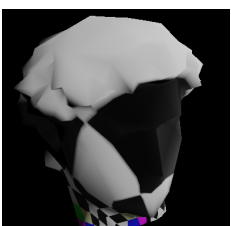
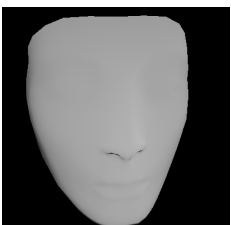
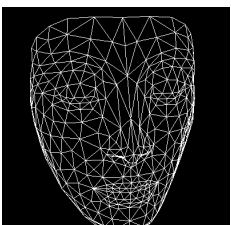
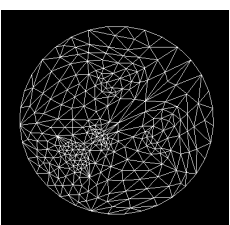
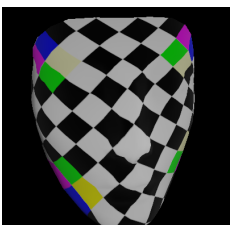
| 原曲面   | 原网格   | 参数化网格  | 纹理映射  |
|---|---|--|---|
|    |    |    |    |
|    |    |    |    |
|   |   |   |   |
|  |  |  |  |
|  |  |  |  |

Table 5.4: Cotangent 权重，单位圆边界的网格参数化与纹理映射

### 5.2.4 Cotangent 权重 & 正方形边界

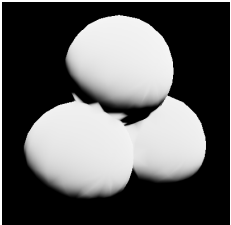
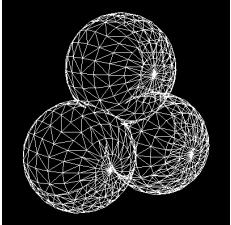
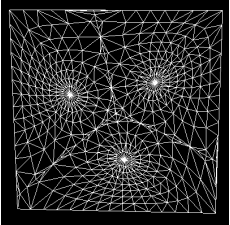


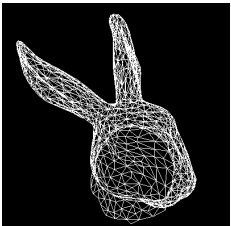
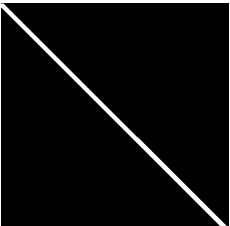
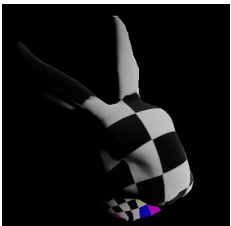
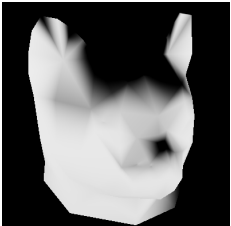
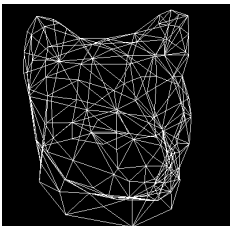
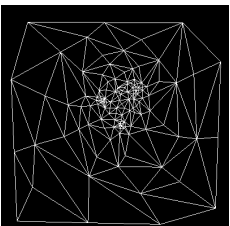
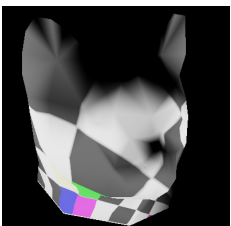
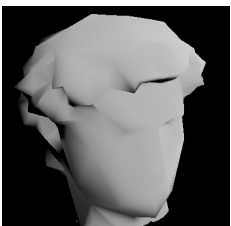
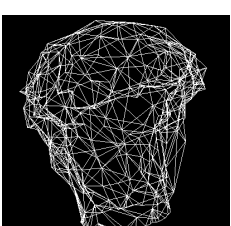
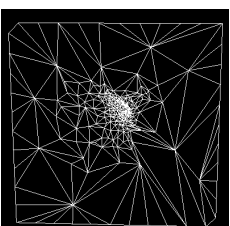
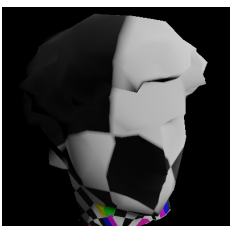
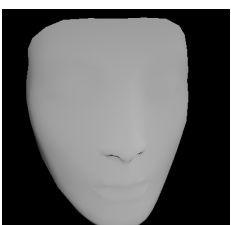
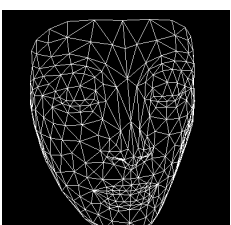
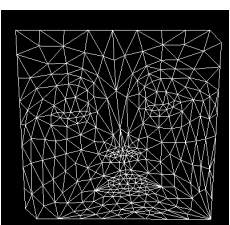
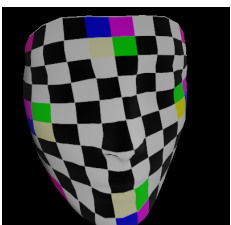
| 原曲面   | 原网格   | 参数化网格  | 纹理映射  |
|---|---|--|---|
|    |    |    |    |
|    |    |    |    |
|   |   |   |   |
|  |  |  |  |
|  |  |  |  |

Table 5.5: Cotangent 权重，正方形边界的网格参数化与纹理映射

## 6 总结

从实验结果可以看出使用 Uniform 权重时，曲面表面的纹理会有一定程度的变形，而使用 cotangent 权重时会改善很多。

曲面参数化的核心是求取重心坐标，除了作业中的 Uniform 和 Cotangent 方法之外，还有 Watchpress Coordinates、Mean Value Coordinates 以及论文 [1] 中提到的 shape preserving 方法，由于时间原因没来得及一一测试。

## REFERENCES

- [1] Michael S Floater et al. Parametrization and smooth approximation of surface triangulations. *Computer aided geometric design*, 14(3):231–250, 1997.