

# USTC\_CG HW4 MinSurMeshPara

张继耀,PB20000204

2023 年 6 月 18 日

## 目录

<b>1</b>	<b>问题介绍</b>	<b>1</b>
1.1	主要目的	1
1.2	实验内容	2
<b>2</b>	<b>算法设计</b>	<b>2</b>
2.1	极小曲面	2
2.2	参数化	3
<b>3</b>	<b>结果展示</b>	<b>3</b>
3.1	程序界面	3
3.2	实验结果	4
3.2.1	测试样例	4
3.2.2	两种参数化的对比	4
<b>4</b>	<b>总结与讨论</b>	<b>5</b>

## 1 问题介绍

### 1.1 主要目的

- 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

## 1.2 实验内容

- 极小化曲面类:MinSurf.h 和 MinSurf.cpp, 在其中完成极小曲面对应的算法。
- 参数化类:Paramaterize.h 和 Paramaterize.cpp, 在其中完成网格参数化算法, 并要求实现
  - Uniform weight
  - Cotangent weight
  - 显示纹理映射结果
- 在UEngine中添加功能, 主要有
  - 求给定边界的极小曲面
  - 非封闭网格曲面的参数化(圆形边界和正方形边界,两种权重的选取)
  - 显示纹理映射

## 2 算法设计

### 2.1 极小曲面

考虑曲面的微分坐标, 即

$$\delta_i = \nu_i - \sum_{\nu \in N(j)} \omega_j \nu_j$$

这里 $N(i)$ 是和 $\mu_i$ 相连的所有顶点的下标,  $\omega_j$ 为权重, 在求极小曲面的时候取了 $\omega_j = \frac{1}{d_j}$ , 这里 $d_j$ 是顶点的度。

为了方便起见, 可直接规定 $\sum \omega_i = 1$ 。对于极小曲面, 有 $\delta_i = 0$ 成立。而边界上的点是固定不动的, 于是我们得到线性方程组:

$$\begin{cases} Lx = \delta = 0 \\ x_i = \nu_i \text{ on Boundary} \end{cases}$$

其中 $L = I - D^{-1}A$ , 且有

$$A_{ij} = \begin{cases} 1, i \in N(j) \\ 0, otherwise \end{cases}$$

$$D_{ij} = \begin{cases} d_i, i = j \\ 0, otherwise \end{cases}$$

于是这个问题就变成了与HW3中类似的问题。还是利用Eigen库来求解方程组。

## 2.2 参数化

参数化与极小曲面相比，只多了一步就是将边界固定到平面凸多边形(例如正方形或圆形)上。只需将边界点按顺序均匀的分布在边界上即可。

定义嵌入:

$$\begin{cases} Wx = b_x \\ Wy = b_y \end{cases}$$

其中

$$w_{ij} = \begin{cases} < 0, (i, j) \in E \\ -\sum_{j \neq i} \omega_{ij}, (i, i) \\ 0, otherwise \end{cases}$$

考虑如下两种权重:

- Uniform weight
  - $\omega_j = 1$
- Cotangent weight
  - $\omega_j = (\cot \alpha + \cot \beta)$

剩下的步骤就与极小曲面的一样了。

## 3 结果展示

### 3.1 程序界面

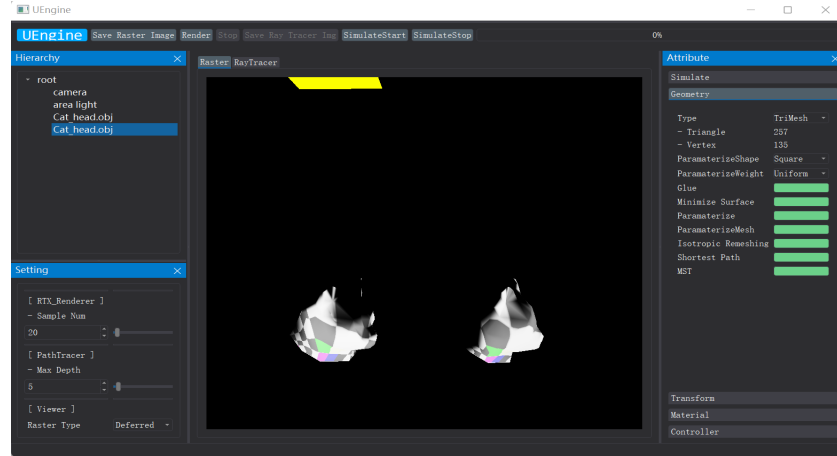


图 1: 程序界面

程序界面如上所示

- 添加了两个按钮ParamaterizeShape和ParamaterizeWeight，用于选择参数化时边界的形状和选择的权重
- Paramaterize时完成参数化，ParamaterizeMesh时显示输出的结果

## 3.2 实验结果

### 3.2.1 测试样例

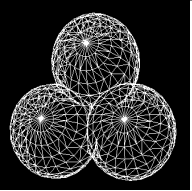
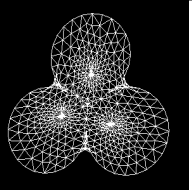


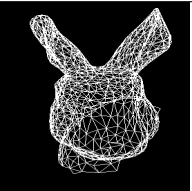
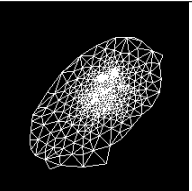
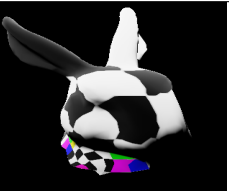
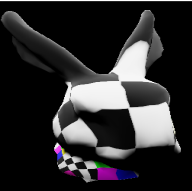
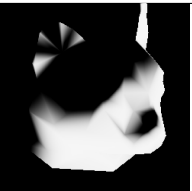
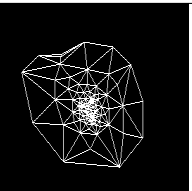
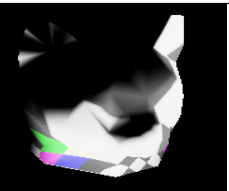
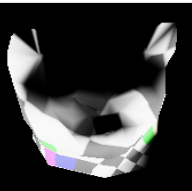
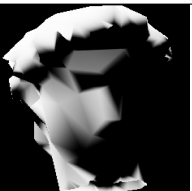
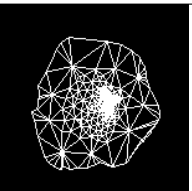
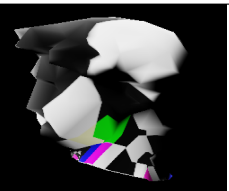
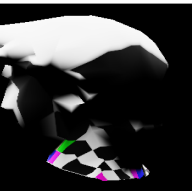
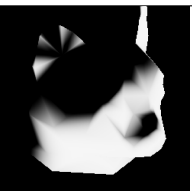
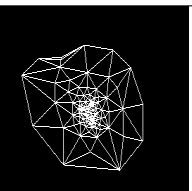
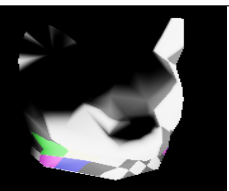
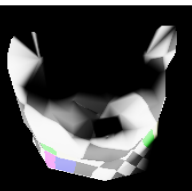
使用方法 测试例子	原图	极小曲面	纹理(Uniform参数化)	纹理(CoTan参数化)
Ball				
Bunny Head				
Cat Head				
David				
Face				

表 1: 主要结果

### 3.2.2 两种参数化的对比

以下是测试两种参数化在映射到圆形边界或正方形边界的例子，第一个表格是对Ball做的测试

人脸的例子可能会更明显一些。可以看出Cot方式基本能看出清晰的轮廓，Uni方式就要差一些。

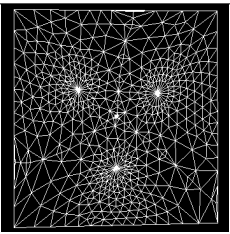
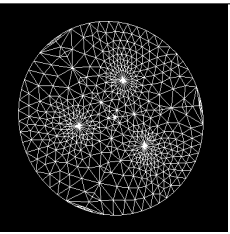
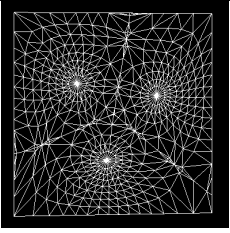
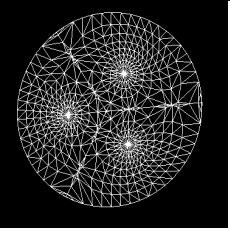
边界形状 使用方法	正方形	圆形
Uni		
Cot		

表 2: 对Ball做的测试结果

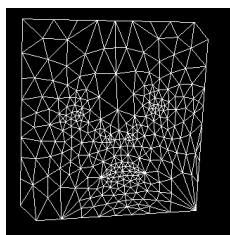


图 2: Uni方式

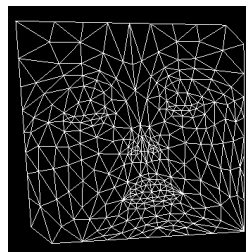


图 3: Cot方式

## 4 总结与讨论

- 通过选取不同的权重和边界条件，参数化的结果也是不同的。可以看到，Cot方式一般会比Uni方式要好一些。通常来说，Cot方式产生的图像纹理更平整、光滑。
- 正方形边界有时会出现缺角的情况，需要特殊处理。
- 在写参数化时，可以调用极小曲面已有的函数来得到结果，代码会更简洁一些。