CG作业4报告

李惟乐 PB23050929

内容

算法原理

1. 固定边界求解极小曲面

顶点的微分坐标为：

$$\boldsymbol{\delta}_i = \boldsymbol{v_i} - \sum_{j \in N(i)} w_j \boldsymbol{v_j},$$

在固定边界点坐标取极小曲面的时候，令$\boldsymbol{\delta}_i = \mathbf{0}$，可以得到如下的线性方程组

$$v_i - \frac{1}{d_i} \sum_{k \in N_{\text{int}}(i)} v_k = \frac{1}{d_i} \sum_{j \in N_{\text{bnd}}(i)} v_j, \quad \text{for all interior } i.$$

其中 $N_{int}(i)$ 指的是在 v_i 附近的在曲面边界内的点， $N_{bnd}(i)$ 则为在该点附近的，在曲面边界上的点。边界内的点有待求解，在边界上的点为定值。因此按如下逻辑构建矩阵A和向量b：

对所有的非边界点进行遍历，对每一个边界点，再遍历它的所有邻居。若邻居为边界点，则根据其坐标值和权重，分别更新x, y, z三个方向的 b ，程序如下：

```
if (neighbor.is_boundary())
{
    const auto& position = halfedge_mesh->point(neighbor);
    B_x(i) += weight * position[0];
    B_y(i) += weight * position[1];
    B_z(i) += weight * position[2];
}
```

图像融合算法把图像看作 $\mathbb{R}^2 \to \mathbb{R}^3$ 的映射）求解的是这样一个问题：

而如果邻居并非边界点，则其为待解变量，为其构造一个矩阵上对应的三元组

```
triplet_list.push_back(Triplet<double>(i, Mirror[neighbor.idx()], -weight));
```

以此即可得到待解矩阵，解出写回即可。

2. 修改边界条件得到平面参数化

在上面求解极小曲面的基础上，如果将边界映射到多边形上，则可以得到对应的参数化表达。

圆形

对边界点按顺序进行遍历，通过以下公式得到每个点的对应角度：

$$\theta = 2\pi \frac{total\_length\_now}{perimeter}$$

其中total_length_now指的是到该点的累计边界长度，perimeter为边界总周长。之后令 $(x, y, z) = (cos\theta, sin\theta, 0)$ 即可得到边界点对应的归一化圆盘映射点。

正方形

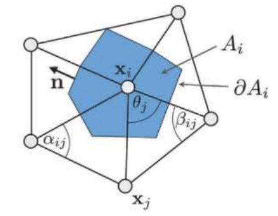
同样对边界点进行遍历，将点分为4份，比如33个点，分为8,8,8,9四份（有公共点），各个点的对应坐标如下：

- First Edge: $x = 0$, $y = \frac{total\_length\_now}{side\_length}$
- Second Edge: $y = 1$, $x = \frac{total\_length\_now}{side\_length}$
- Third Edge: $x = 1$，$y = 1 - \frac{total\_length\_now}{side\_length}$
- Fourth Edge: $y = 0$，$x = 1 - \frac{total\_length\_now}{side\_length}$

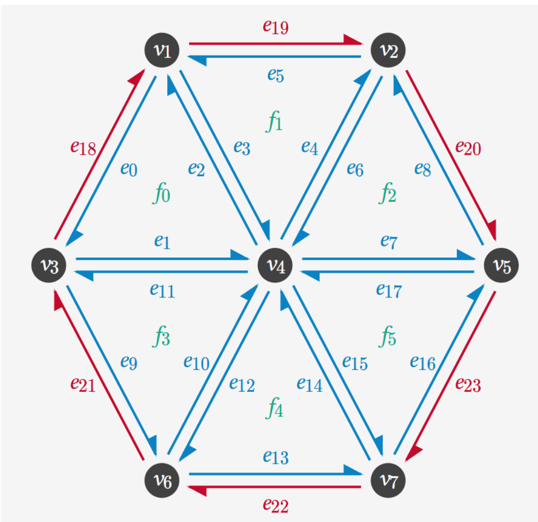
3. 不同的参数化

共实现了两种不同的参数化

- Uniform weights: $w_j = 1$;
- Cotangent weights: $w_j = \cot\alpha_{ij} + \cot\beta_{ij}$;



这里讲一下余切权重的实现，以下图为例



比如需要计算 v_4 到 v_6 的权重，则首先找到连接两个顶点的两条半边，记为 $link1$ 和 $link2$ ，则角α对应的两个向量（半边为）：

```
alpha_edge1 = link_edge1.next().opp();
alpha_edge2 = link_edge1.next().next();
```

同理，角β对应的两个半边为：

```
beta_edge1 = link_edge2.next().opp();
beta_edge2 = link_edge2.next().next();
```

则利用向量的点乘与叉乘模之商，可得到两个cot值，进而得到权重。

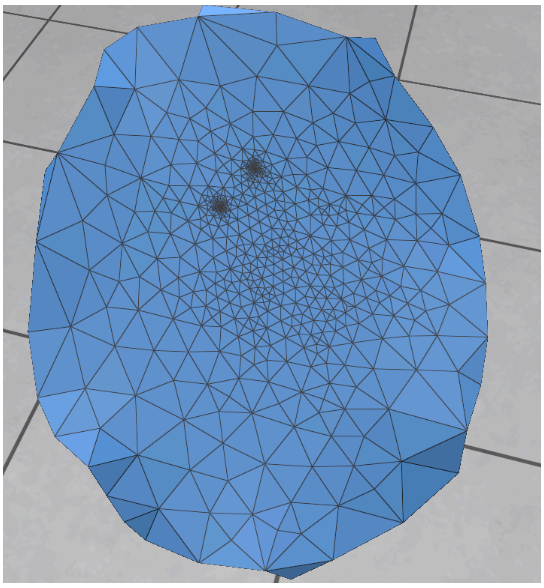
另外最终的权重值需要归一化，即

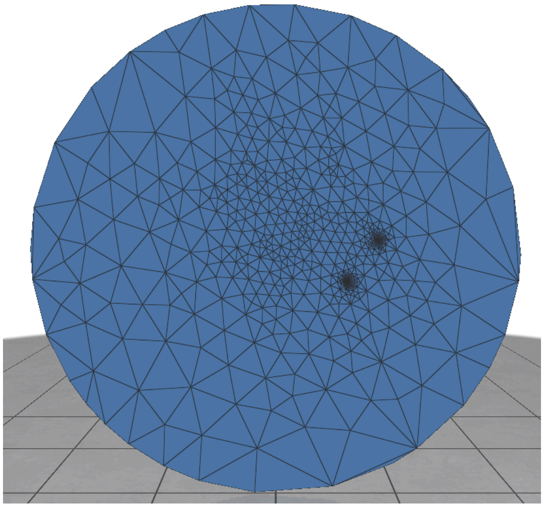
$$w_{j\_Final} = \frac{w_j}{\sum\limits_k w_k}.$$

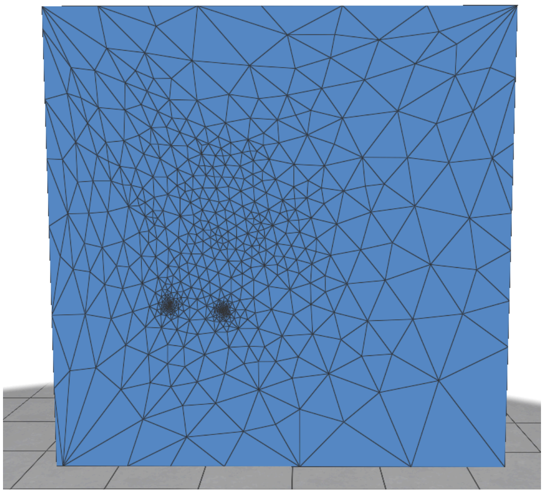
作业中将权重计算作为一个基类，派生出均匀权重和余切权重两个子类。

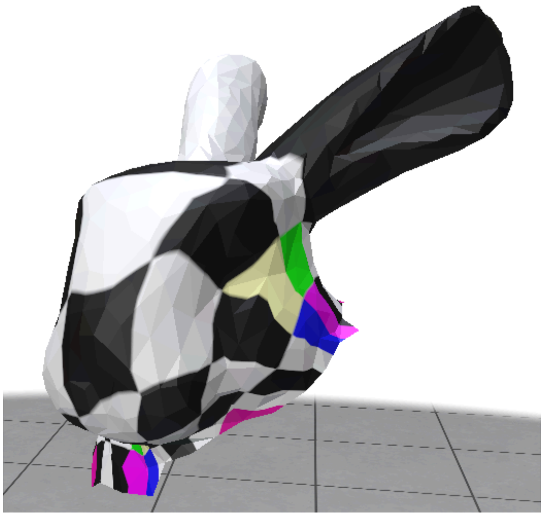
结果

均匀权重：

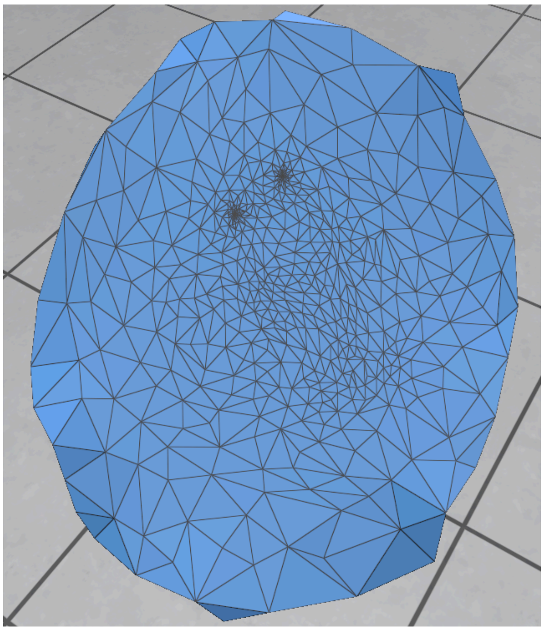


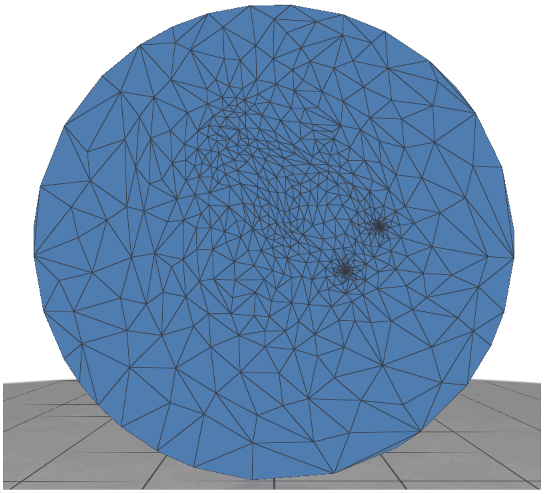


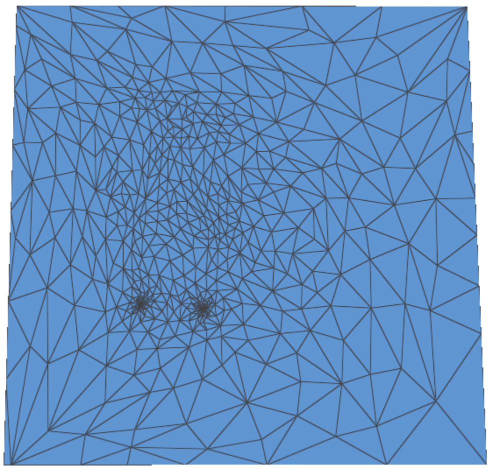


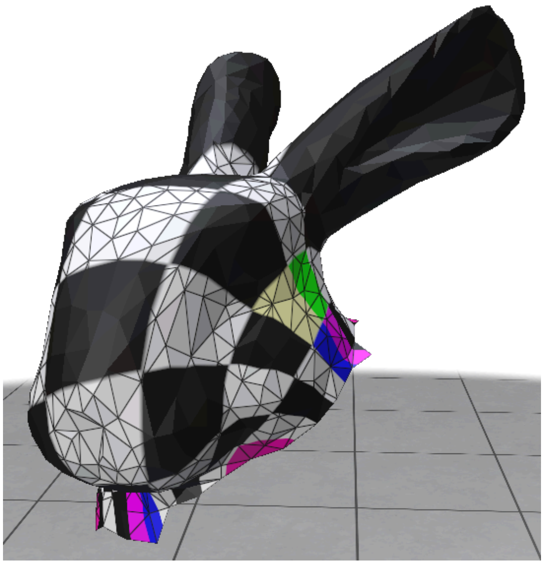


余切权重：









后续扩展

在编写程序过程中，出现过以下问题：

- 权未进行归一化处理，导致出现异常结果
- 权设置时用的参数误用为了参数化后的坐标，应为参数化前的坐标

目前该程序还可以进行如下改进：

- 添加新的权重子类shape-preserving weights