homework 7

Global Laplacian Smoothing

上次作业的内容是局部的拉普拉斯平滑。局部的laplacian平滑缺点在于只考虑局部，而且需要多次迭代，每次处理一个点，耗时长。局部的laplacian平滑实际上做的事情是在缩短下面向量的长度：

$$v_i - \sum_{j \in \text{1-ring}} w_j v_j$$

这个向量也就是离散形式下的平均曲率流，当每个顶点的平均曲率流长度为0时，也就得到了极小曲面。仔细观察可以发现，其实上式很容易组织成矩阵的形式：

$$L\mathbf{v} = \sigma$$

其中$\mathbf{v}$是所有顶点构成的向量，而这里的L是由权重组成的，是一个稀疏矩阵，只有在网格中有边连接的部分在矩阵中才是非0项；σ的每一项则是：$\sigma_i = v_i - \sum_{j \in \text{1-ring}} w_j v_j$：

$$\begin{pmatrix} l_{1,1} & l_{1,2} & \cdots & l_{1,n} \\ l_{2,1} & l_{2,2} & \cdots & l_{2,n} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ l_{n,1} & l_{n,2} & \cdots & l_{n,n} \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ \vdots \\ v_n \end{pmatrix} = \begin{pmatrix} \sigma_1^{(x)} \\ \sigma_2^{(x)} \\ \vdots \\ \vdots \\ \sigma_n^{(x)} \end{pmatrix}$$

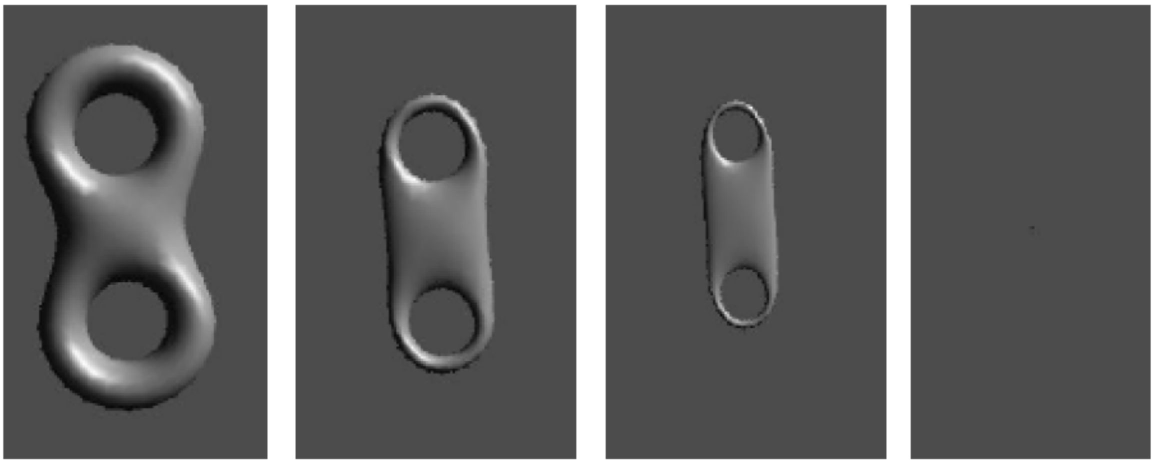
其中：

$$l_{i,j} = \begin{cases} 1 & i = j \\ -w_{i,j} & j \in \text{1-ring}(i) \\ 0 & \text{otherwise} \end{cases}$$

如果我们想要求极小曲面的顶点坐标，只要让所有的非临界点的$\sigma_i = \mathbf{0}$，对上式方程求解$\mathbf{v}$即可，就可以得到新的顶点坐标。

这里需要注意的点是，如果我们不固定任何点，也就是直接令$\sigma = \mathbf{0}$，那么一个天然解就是：$\mathbf{v} = \mathbf{0}$，网格对象也就消失了。这个在某种程度上解释了之前为什么对封闭

曲面求极小曲面会出现下面的现象：



固定边界点的方法就是让边界点对应的$\sigma_i = v_i$。

我们可以添加一个系数λ，乘在非边界顶点的σ_i上，可以用来控制光滑程度。当$\lambda = 0$时，也就可以得到极小曲面：

$$L\mathbf{v} = \lambda\sigma$$

Mesh Parameterization

之前学参数化，一直在做的是将曲线进行参数化，本征维度为1。对于曲面来说，参数化本征维度为二，也就是映射到(u, v)平面上。实际上，在学会极小曲面之后，映射到二维就值差一步了，只要将边界点固定到平面上就可以。这里需要注意的是，一般需要将边界点映射到一个凸多边形上，理论证明这样的映射方式，三角形一定不会发生反转（flip）。

整个算法步骤如下：

- 检测边界
- 将边界映射到正方形边界或圆边界 (凸边界)
- 构建稀疏方程组
- 求解稀疏方程组 更新顶点坐标 连接纹理图像，更新显示

这里主要要做的是第二步，其他几步在其他作业中已经包含了。第二步中，如果将边界点映射到一个凸多边形上？首先要考虑的问题是，在读取网格的时候，边界点读取进来的顺序可能是任意的，但是实际上这些边界点是有连接关系的，所以第一步是找到有序的边界点。这个利用HalfEdge很容易做到。步骤如下：

1. 找到任意一个边界边，将该边的末尾顶点加入待查找的有序点中

2. 遍历与这个末尾点连接的所有边，直到找到下一个边界边
3. 回到第一步，直到遇到最初的边界边，算法结束

这样我们就得到了有序的边界点集：$\{v_1, ..., v_n\}$

将这些点映射到凸多边形上（我选择的是圆），方法也有多种。我的做法是对这些点进行参数化。因为首先边界是曲线，所以之前学过的Uniform，或者Foley参数化都可以，这样可以得到各个点的参数$t_i \in [0, 1]$。接下来只要按照参数映射到多边形上即可，对于圆的话，可以直接根据参数求得角度：$\theta_i = 2\pi \cdot t$，再根据角度求得坐标值（其中r为半径）：

$$(x, y) = (r \cdot \cos\theta, r \cdot \sin\theta)$$

作业完成情况

我在尝试用games102学到的内容来制作一个C++几何处理的框架 ，名称为Dragon（https://github.com/MyEvolution/Dragon）。目前这个框架已经完成了可视化的部分的制作（使用OpenGL+ImGui），能比较方便地展示这次作业的情况。这个框架是在Ubuntu系统上测试的。本次作业完成情况如下：

1. 实现了全局拉普拉斯平滑
2. 实现了网格参数化

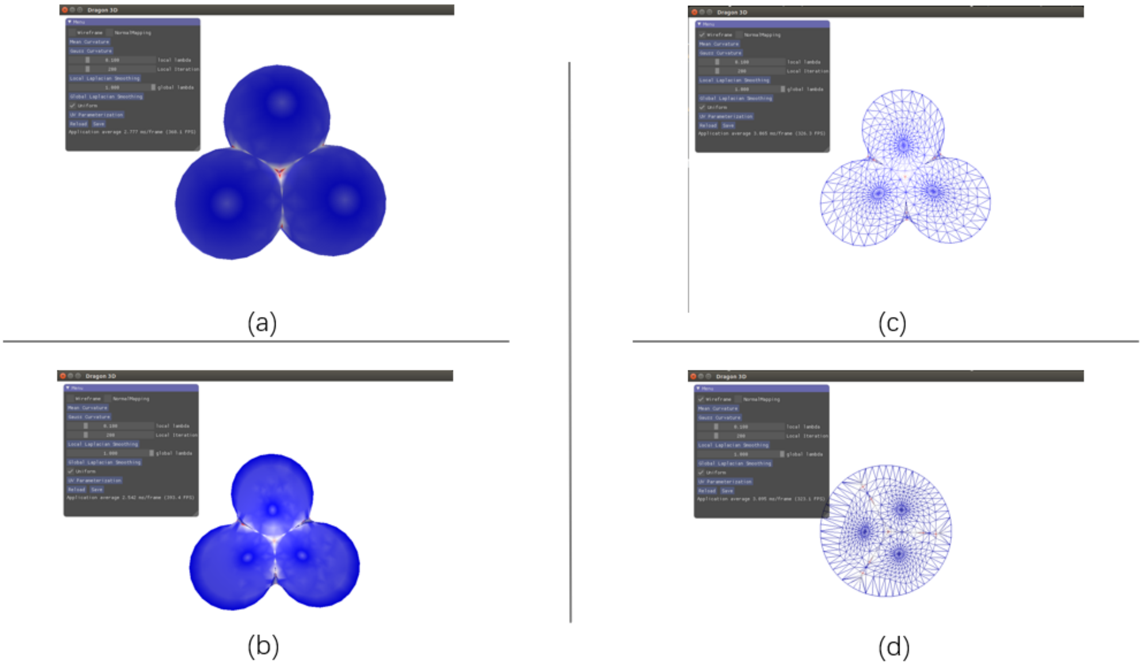


Fig 1. Balls (a)原模型 （b）全局极小平面 （c）全局极小平面的wireframe显示 （d）曲面参数化的wireframe显示

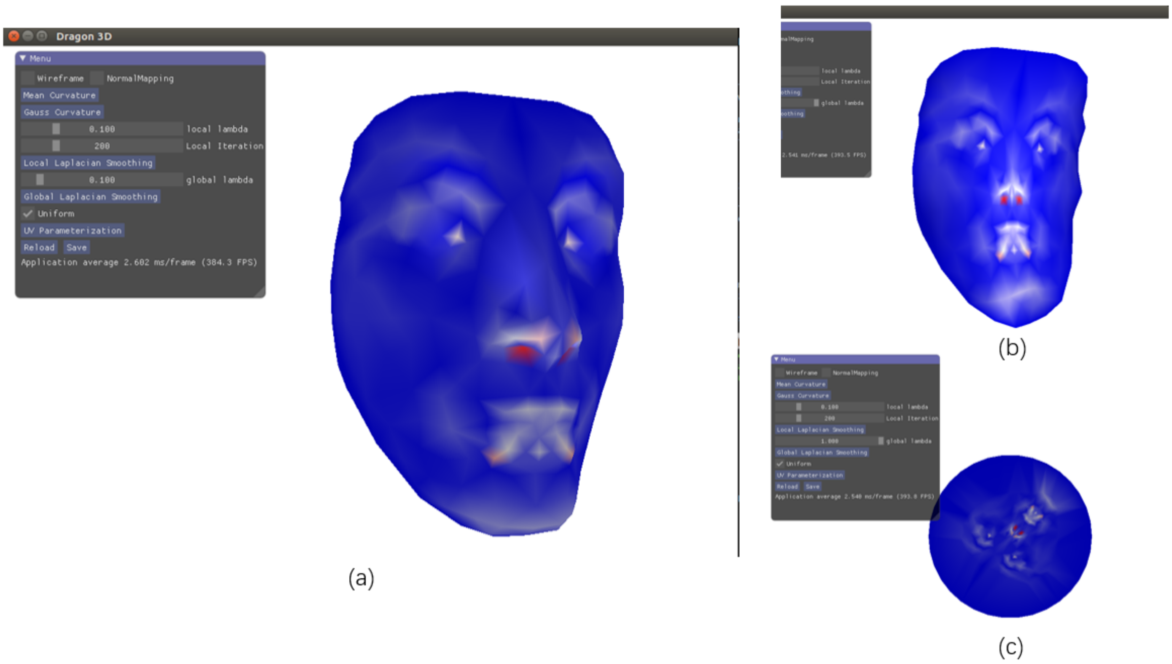


Fig 2. Face (a)原模型 （b）全局极小平面 （c）曲面参数化

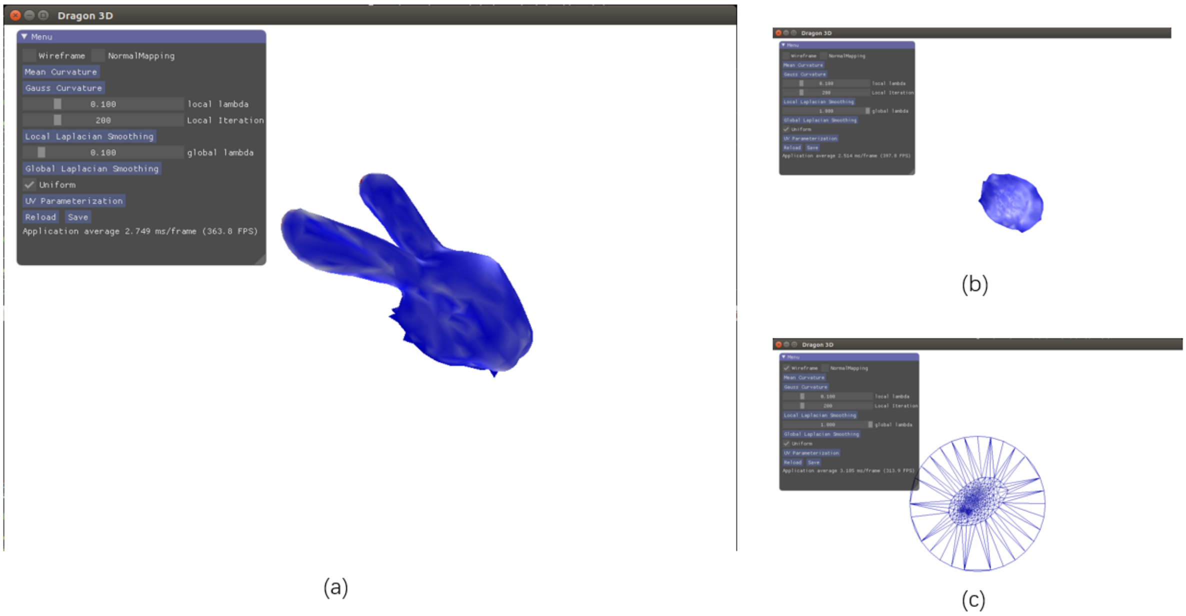


Fig 3. Bunny head (a)原模型 （b）全局极小平面 （c）曲面参数化后的wireframe显示

此外，我将之前的高斯曲率与平均曲率，局部极小曲面也整合到本框架中了，也增添了显示wireframe，显示法向量等功能。详细内容可以查看视频。

以上是本次作业的内容，谢谢老师与助教。