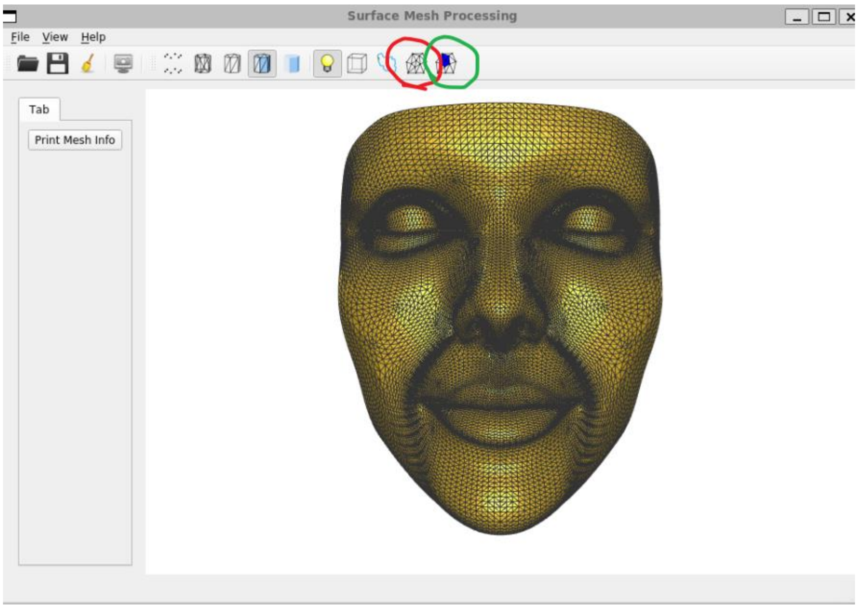
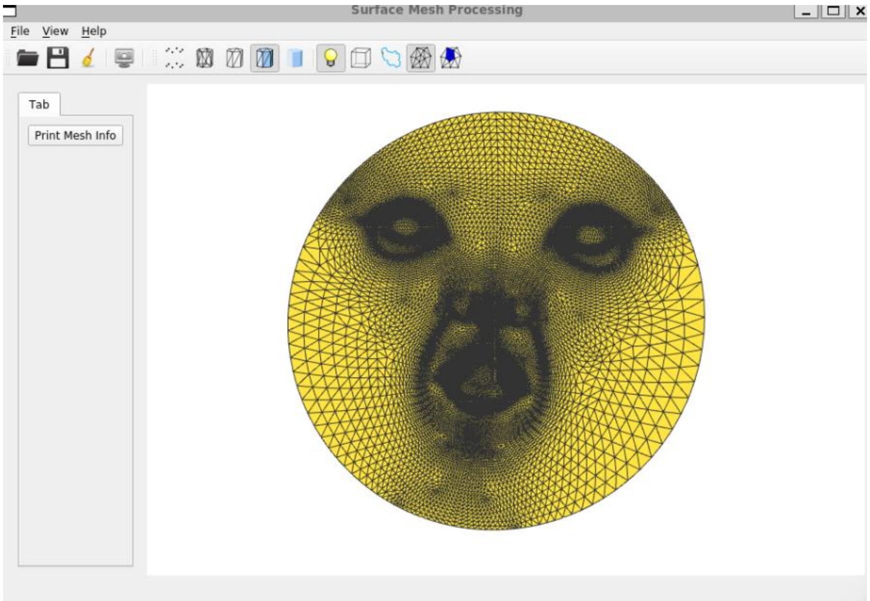
编译和运行

编译按照标准的 cmake 程序
mkdir build && cd build && cmake .. && make

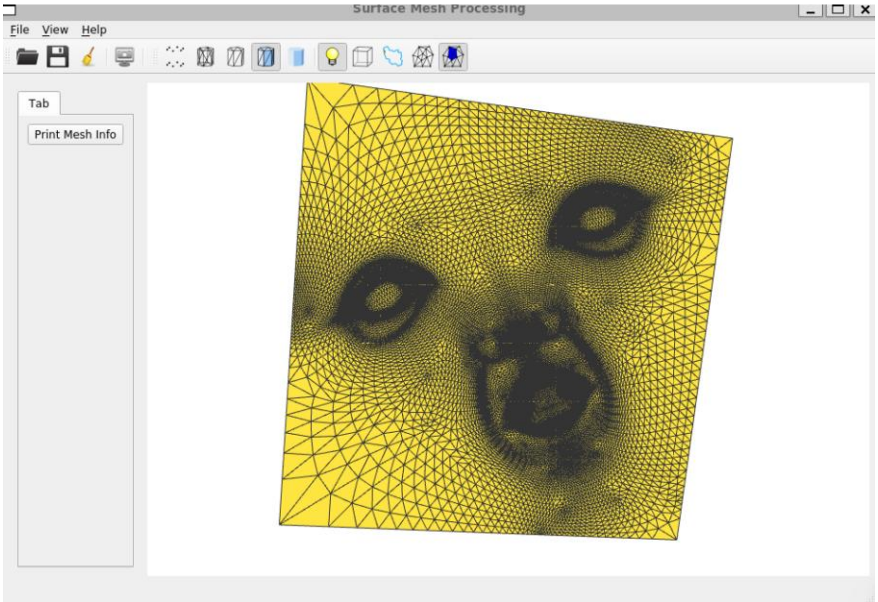
运行 SurfaceFrameworkCmake，加载 face.obj，效果如下



红圈对应按钮把边界映射成标准圆盘。



绿圈对应按钮把边界映射成正方形。



代码和原理

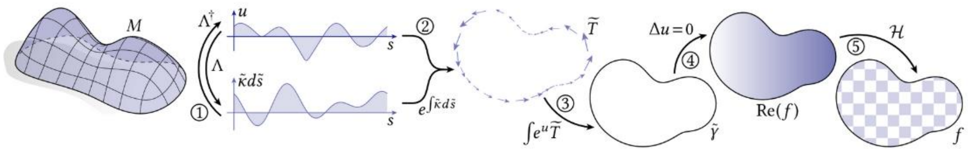


Fig. 7. Overview of the basic BFF algorithm. ① Given a surface M and either target scale factors u or target curvature density $\tilde{\kappa} d\tilde{s}$ along the boundary, the complementary quantity is obtained via the Dirichlet-to-Neumann map Λ. ② Curvature density is integrated to obtain unit tangents $\tilde{T}$. ③ Integrating rescaled tangents $e^{u}\tilde{T}$ yields the target boundary curve $\tilde{\gamma}$. ④ The real component of $\tilde{\gamma}$ is extended harmonically. ⑤ The Hilbert transform $\mathcal{H}$ provides the imaginary coordinate, and hence the final flattening $f : M \to \mathbb{C}$.

代码主要参考论文上图的流程和开源实现（https://github.com/GeometryCollective/boundary-first-flattening）

主要算法如下：

1. 根据指定边界（标准圆盘面和正方形）确定尺度因子 u。（算法 1）
2. 根据尺度因子计算边界，通过优化得到闭合曲线。（算法 2）
3. 根据边界曲线得到整个曲面的映射。（算法 3）

算法 1 是整个论文的核心。
给定目标边界的离散测地曲率（对应论文中曲率在对偶边上的积分）求解尺度因子。因为尺度因子和离散测地曲率需要满足 Cherrier 方程，通过求解 Cherrier 方程可以得到尺度因子。对于圆盘面（测地曲率为 1，因为对偶边长未知所以离散测地曲率）需要迭代得到离散测地曲率和尺度因子。对于正方形边界，有四个顶点的离散测地曲率为 90°，其余为零。

根据比例因子u和曲率的关系得到Yamabe方程，带边界的情况由Cherrier(1984)给出

$$\begin{cases} \Delta u = K - e^{2u}\tilde{K} & \text{on } M \\ \frac{\partial u}{\partial n} = \kappa - e^{u}\tilde{\kappa} & \text{on } \partial M \end{cases}$$

BFF论文中给出了分别在单位三角形和边上积分得到了离散形式

$$\Delta u dA = K dA - \widetilde{K} d\tilde{A}$$

$$\frac{\partial u}{\partial n} ds = \kappa ds - \tilde{\kappa} d\tilde{s}$$

积分得到

$$\begin{cases} Au = \Omega - \tilde{\Omega} \\ h = k - \tilde{k} \end{cases}$$

其中h是$\frac{\partial u}{\partial n}$在原对偶边$e_i$上

Cherrier方程

已知目标高斯曲率$\tilde{\Omega}$和测地曲率κ求解比例因子u

$$\begin{bmatrix} \mathrm{A_{II}} & \mathrm{A_{IB}} \\ \mathrm{A_{IB}^T} & \mathrm{A_{BB}} \end{bmatrix} \begin{bmatrix} \mathrm{u}_I \\ \mathrm{u}_B \end{bmatrix} = \begin{bmatrix} \Omega_{\mathrm{I}} \\ -\mathrm{h} \end{bmatrix}$$

这里$\mathrm{A}_{ij} = -\frac{1}{2}\left(\cot\beta_p^{ij} + \cot\beta_q^{ij}\right)$

推导$\frac{\partial u}{\partial n}$在对偶边上的积分。

给定Disk边界，如何计算离散测地曲率$\tilde{k}$。我们知道Disk的测地密度为$1/2\pi$，但是由于不知道缩放因子u，所以也不知道对偶边长度，所以需要用迭代算法求解。

1. 初始u=0，计算$l_{ij}^* = e^{(u(i)+u(j))/2} l_{ij}$，$l_{ij}^*$是目标边界的顶点i和j的边长。
2. 计算顶点的i的对偶边长$e_i = 0.5 * (l_{i-1i}^* + l_{ii+1}^*)$
3. 得到测地曲率$\tilde{k}_i = e_i/(2\pi * L)$
4. 根据测地曲率计算缩放因子u，重复这个过程直到收敛。

算法 2
由于离散误差的存在，算法 1 得到的边界可能不是封闭曲线。
通过求解下面优化问题得到封闭的曲线边长 ltilde。
下面优化问题有两个等式约束，可以引入两个对偶变量，通过拉格朗日乘子进行求解。

$$\min_{\tilde{\ell}:B\to\mathbb{R}} \frac{1}{2}\sum_{ij\in\partial M} {\ell_{ij}}^{-1}|\tilde{\ell}_{ij} - \ell_{ij}^*|^2 \quad \text{s.t.} \quad \sum_{ij\in\partial M} \tilde{\ell}_{ij}\widetilde{T}_{ij} = 0.$$

算法 3
因为共形映射的实部 u 和虚部 v 分别为调和映射，所以这部分原理已知边界调和权重的 Tutte 算法一致。

主要算法代码

算法 1：`convertNeumannToDirichlet`

算法 2：`closeLengths`

算法 3：`extendHarmonic`