# GAMES301 曲面参数化-HW1

Author: 彭博

## 概述

本次作业基于matlab实现了Tutte参数化，根据权重不同分为均匀权重的参数化和Floater权重的参数化两部分。参数化结果可以直接运行`hw1.m`来进行查看(保存参数化后的网格需要使用gptoolbox的`writeOBJ`函数)。

## 基于均匀权重的参数化

经典Tutte参数化使用了均匀权重，即对于网格内部的顶点$v_i$其参数化后的坐标满足相邻顶点的线性组合：

$$v_i = \frac{1}{|N(v_i)|}\sum_{v_j \in N(v_i)} \lambda_{ij} v_j \Rightarrow |N(v_i)| \cdot v_i - \sum_{v_j \in N(v_i)} \lambda_{ij} v_j = 0$$

$$\lambda_{ij} = 1$$

而对于边界上的顶点则可以手动将其映射到一个凸多边形(如圆或正方形)上。此时参数化的过程等价于求解线性方程组：

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

其中系数矩阵$\mathbf{L}$为一个稀疏矩阵，其每一行对应网格上每一个顶点参数化坐标需要满足的方程；$\mathbf{b}$同样为一个稀疏矩阵，仅在边界点上取凸多边形上给定的坐标。

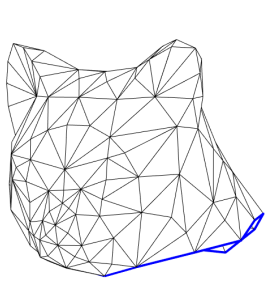
均匀权重的参数化的实现可参见`tutte.m`文件，其主要流程包括寻找并放置边界点、构造系数矩阵$\mathbf{L}$以及求解稀疏矩阵方程等。其中构造构造系数矩阵的核心代码摘录如下：

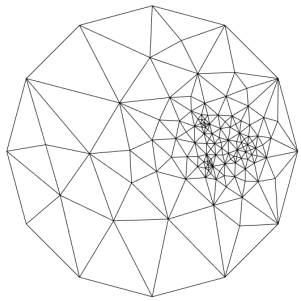
```
%% uniform Laplacian L=D-A
I = []; J = []; value = [];
D = zeros(1, nV);

for e = 1:nE
    vi = E(e, 1); vj = E(e, 2);

    if ~ismember(vi, B)
        I = [I, vi];
        J = [J, vj];
        value = [value, -1];

        D(vi) = D(vi)+1;
    end

    if ~ismember(vj, B)
        I = [I, vj];
        J = [J, vi];
        value = [value, -1];

        D(vj) = D(vj)+1;
```

```
    end
end

%% add diagnoal element
I = [I, 1:nV];
J = [J, 1:nV];
value = [value, D];

%% add boundary constraints
I = [I, B];
J = [J, B];
value = [value, ones(1,nB)];

%% LHS
L = sparse(I, J, value, nV, nV);
```

构造稀疏系数矩阵时会对网格的所有边进行遍历，如果边对应的顶点不在边界上则为系数矩阵的对应行添加元素，而对于边界上的顶点则需要在系数矩阵的对角元上添加元素。

均匀权重参数化结果可参见下图。





# 基于Floater权重的参数化

Floater权重的参数化过程与均匀权重参数化非常类似，唯一区别在于如何计算内部顶点与其邻域的组合系数$\lambda_{ij}$。计算$\lambda_{ij}$时需要考虑$v_i$与其邻域的几何关系，具体来说首先需要把三维网格上的1-ring按照保持角度和长度的方式投影到平面上，然后对于每个相邻点$p_j$寻找一个包含$p_i$的平面三角形$\Delta p_j p_r p_{r+1}$，最后把$p_i$在这个三角形上的重心坐标作为三角形对应顶点的系数。完成遍历后把顶点$v_j$所有三角形上重心坐标相加就得到了所需系数$\lambda_{ij}$。

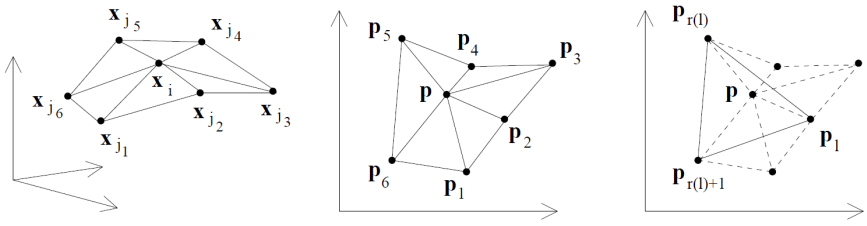



Fig. 5. The subtriangulation $\mathcal{S}_i(\mathcal{G}_i,\mathbf{X}_i)$ and a local parametrization $\mathcal{P}_i$.

Floater权重的参数化的实现可参见`floater.m`文件，其构造系数矩阵的代码摘录如下：

```
%% find half-edges
[HE, NEXT, TWIN, VHE] = findHE(V, F);

%% Floater weighted Laplacian L
I = []; J = []; value = [];
```

```
D = zeros(1, nV);

for vi=1:nV
    if ~ismember(vi, B)
        %% find one half-edge starting from vi
        heIdx = VHE(vi);

        %% find one-ring
        ring = findOneRing(heIdx, HE, NEXT, TWIN);
        nRing= length(ring);

        %% project to plane
        [P, ~, Theta] = project2Plane(V, vi, ring);

        %% add weights
        for pjIdx=1:nRing
            %% find a valid triangle on the plane
            [pjIdx, pkIdx, plIdx] = find2DTriangle(pjIdx, Theta);
            vj = ring(pjIdx); vk = ring(pkIdx); vl = ring(plIdx);

            %% find barycentric coordinates
            pj = P(pjIdx, :); pk = P(pkIdx, :); pl = P(plIdx, :);
            [bj, bk, bl] = findBaryCententricCoordinate(pj, pk, pl);

            %% add to sparse matrix
            I     = [    I, vi, vi, vi];
            J     = [    J, vj, vk, vl];
            value = [value,-bj,-bk,-bl];

            D(vi) = D(vi)+1;
        end
    end
end
```

为了方便对内部顶点的1-ring进行遍历，这里编写了`findHE`函数来实现一个简单的半边数据结构，其返回值分别为网格的所有半边、指向下个半边的指针、半边的对偶以及每个顶点到半边的指针。构造系数矩阵时首先根据内部顶点$v_i$寻找一条半边，然后利用`findOneRing`函数得到其所有相邻的顶点并使用`project2Plane`函数将它们投影到二维平面上。

接下来对每个相邻顶点$p_j$进行遍历，通过`find2DTriangle`函数来寻找哪些顶点可以构成包含$p_i$的平面三角形，并使用`findBaryCententricCoordinate`函数来计算相应的重心坐标。最后把相应的坐标添加到稀疏矩阵即可。相关辅助函数的定义和实现可参考相应的`.m`文件。

Floater权重参数化结果可参见下图：