余切加权进行拉普拉斯优化

摘要：

拉普拉斯网格优化与平滑是网格处理的经典算法，其一些基本概念可以作为神经网络预测3D mesh的一些约束，如平滑。

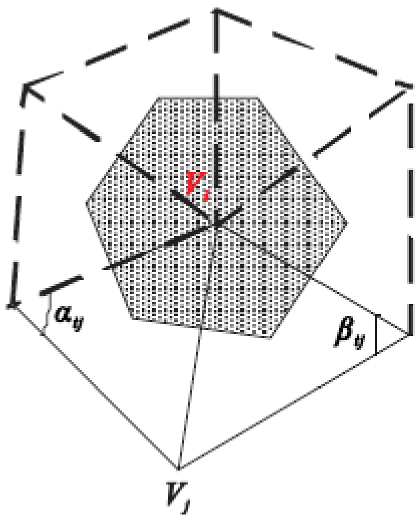
图的表示：

我们用 G=(V,E) G = (V , E) 来表示一个网格，其中V表示顶点由（x,y,z）三个坐标组成；E表示边由两个顶点组成。

拉普拉斯优化：

拉普拉斯优化的本质根据邻结点对每个顶点的位置进行优化，保证曲面看上去更加的光滑。每个邻结点都会对该目标点的位置做出一部分的贡献，贡献的大小由权重乘上到目标点空间距离组成。在拉普拉斯优化中权重可以 取均匀权重或者余切权重下面会进行具体的介绍。

拉普拉斯坐标定义为： $\delta_i = \sum_{\{i,j\}\in \mathbf{E}} w_{ij}(\mathbf{v}_j - \mathbf{v}_i) = \left[\sum_{\{i,j\}\in \mathbf{E}} w_{ij}\mathbf{v}_j\right] - \mathbf{v}_i,$ 其中 ∑{i,j}∈Ewi,j=1 ∑ { i , j } ∈ E w i , j = 1 ，并且 $w_{ij} = \frac{\omega_{ij}}{\sum_{\{i,k\}\in \mathbf{E}} \omega_{ik}}$ $\begin{aligned} \omega_{ij} &= 1, \\ \omega_{ij} &= cot\ \alpha + cot\ \beta, \end{aligned}$ 分别是均匀权值和余切权重。 α,β α , β 为：



[论文]在论文中我们为了给模型精细化，我们增加model的vertices数量，每一个新增的vertice是相邻的vertices的中点上，并且能够在法线方向上进行微调。论文中公式（3）s表示沿着法线方向的微小位移，优化的对象便是s。为了计算方便我们可以计算全局的拉普拉斯坐标值。（包括了model中原有的固定的vertices）。

上面的等式可以用矩阵表示： $\Delta_d = \mathbf{L}\mathbf{V}_d.$ $\Delta_d = [\delta_{1d}, \delta_{2d}, \ldots, \delta_{nd}]^T, d \in \{x,y,z\}$ 其中L是n×n的拉普拉斯矩阵： $\mathbf{L}_{ij} = \begin{cases} -1 & i = j \\ w_{ij} & (i,j) \in \mathbf{E} \\ 0 & \text{otherwise} \end{cases}$ ，优化与平滑

[论文]我们可以预先计算出一个Wi,j的权重的矩阵，权重矩阵大小（N*N），如果我们选择平均矩阵，即如果存在邻边权重设置为1/（目标度数），不相邻的权重为0。输入是N个vertices的三维坐标V（3 * N）。最终的结果是拉普拉斯坐标 = Vj（3*N） * Wj,i（N * N）- Vi（3 * N）（参考拉普拉斯坐标定义，第一个公式）。得到的结果是N个顶点的拉普拉斯坐标。拉普拉斯坐标与原点越近，则越平滑。（优化对象是可移动点的沿法向的位移s）

拉普拉斯网格平滑求解：

$$\left[\begin{array}{c|c} \multicolumn{2}{c}{\mathbf{L}_u} \\ \hline \mathbf{I}_{m\times m} & \mathbf{0} \end{array}\right]\mathbf{V}'_d = \left[\begin{array}{c} \mathbf{0} \\ \hline \mathbf{V}_{(1\ldots m)d} \end{array}\right]$$

Lu 表示利用均匀权值构建的拉普拉斯矩阵，V_d^' V_d^' 为要求解的平滑后的网格顶点。其中用了m个顶点作为约束。这样等式左边是一个（n+m）×n的矩阵乘以一个n×1的向量（x,y,z分开算），右边是一个（n+m）×1的向量。

拉普拉斯网格优化求解：

$$\left[\begin{array}{c|c} \multicolumn{2}{c}{\mathbf{L}_u} \\ \hline \mathbf{I}_{m\times m} & \mathbf{0} \end{array}\right]\mathbf{V}'_d = \left[\begin{array}{c} \Delta_{d,c} \\ \hline \mathbf{V}_{(1\ldots m)d} \end{array}\right]$$

与平滑不一样的是，等式右边向量的前n个元素不再为0，而是在余切权值下计算得来的拉普拉斯坐标。 约束顶点的权值选择。