

Re-Tiling and PPS

李虎森

摘 要

本文介绍了一种网格重新三角化的方法，该方法基于 Re-Tiling 方法 [1]，但在每个新加入点确定最终的坐标时，利用网格生成的一种流形曲面，进行了调整，让得到的三角网格更加合理。

关键词：网格三角化，Re-Tiling，流形曲面

1 引言

论文在第二节，简单介绍了 Re-Tiling 方法 [1]，并提出了该方法的不足之处。为了做进一步改进，在第三节，引进了一种根据网格计算光滑流行曲面的一种方法，简记为 PPS 方法 [2]。第四节根据 Re-Tiling 和 PPS 两种方法，对一些网格进行了测试。第五节，对方法做了一个小结。

2 网格重新三角化——Re-Tiling

该方法的主要目的是，根据输入的原始网格，生成出新的网格，新网格的拓扑结构和原始网格相同，顶点数和三角面片数增多或者变少，让原始网格变得更加稀疏或者稠密。

主要的思路是，先在网格上（即三角形内部）随机插入 n （目标定点个数）个顶点，将插入点和原始网格顶点一起，重新三角化。然后计算每个顶点所收到来自其他插入点的“排斥力”，排斥力的大小和“排斥半径” r 有关，当插入点之间距离大于 r 时，排斥力为零。随后每个插入点根据所受的排斥力，在原始网格上移动。接着重复计算排斥力、在网格上移动，直到达到迭代次数。最后，移除原始顶点（能否移除有相应判别准则），并且将顶点的邻域多边形，重新三角化。这样得到一个新的网格。关于如何在网格上随机插入 n 个点，“排斥力”的计算和移除原始顶点时的判别准则等详细内容，参见文献 [1]。

下面介绍 Re-Tiling 算法流程：

输入：封闭的三角网格，目标顶点个数 n ，排斥力半径 r 。

- (1) 在网格上随机插入 n 个顶点；
- (2) 计算每个插入点之间的排斥力；
- (3) 根据每个插入点所收到的排斥力，将其在原始网格上移动；
- (4) 循环步骤 (2) (3) 达到迭代次数；

(5) 移除原始顶点。

输出：顶点数为 n 的新网格，拓扑结构与原始网格相同。

3 Re-Tiling 算法优化

3.1 Re-Tiling 算法不足

通过第二节的算法步骤，我们可以看到，插入的新顶点只在原始网格上移动，所以当算法第四步结束时，如果有一个原始三角形上超过三个点，那么就出现下面这样的情形：

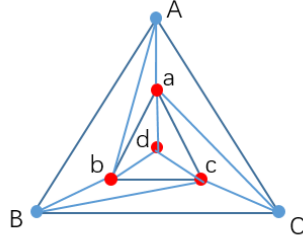


图 1

图中 A,B,C 是原网格上的点，a,b,c,d 是插入的点。此时由于 a,b,c,d 在同一个平面，这样的点是我们不希望的。

所以下面介绍一种给 a,b,c,d 找一个新的空间坐标，让他们即不在同一个平面，又能体现原始网格的形状。而新的位置将在，由网格生成的流形曲面上去找，这种流形曲面记做 PPS[2]。下面简单介绍 PPS 的构造，具体构造流程可以参考文献 [2]。

3.2 PPS 流形面构造

PPS 流形面，由初始网格生成，它近似于三维网格曲面，与网格曲面的拓扑结构和三维形状是接近的，区别在于 PPS 流形是有一定光滑性的曲面。下面来介绍如何根据空间网格构造对应的 PPS 流形。

流形，顾名思义是由多个同胚于欧式空间的“区域”组成，即由多个类似于 (U, φ) 多个 *chart* 确定，其中 U 是流形上“区域”， $\varphi(U) \in \mathbb{R}^n$ ，为方便我们称之为欧式参数域，设为 V 。所以在构建 PPS 流形时，需要先明确组成流形曲面的“区域” U ，以及与它们同胚的欧式参数域 V 。

在构建 PPS 流形时，每个顶点对应一个参数域，参数域是欧式空间 \mathbb{R}^2 上半径为 $|\cos(\pi/m_u)|$ 的圆，所以，只要确定了光滑映射 φ^{-1} ，那么我们的流形曲面就可以由 $\varphi^{-1}(V)$ 组成。

$$\varphi^{-1}(p) = \theta_u(p) = \sum_{v \in J_u(p)} \omega_{vu}(p) \times (\psi_v \circ \varphi_{vu}(p)) \quad (*)$$

其中， $p \in V = \Omega_u$ 。这里 φ^{-1} 具体构造参见文献 [2]。

4 Re-Tiling 和 PPS

这一节将具体介绍，怎么将利用 PPS 优化 Re-Tiling 算法。

4.1 Re-Tiling 优化

首先执行 Re-Tiling 算法流程的步骤 (1)——(4)，在移除原始顶点之前，将插入点对应到参数域 V 上，然后利用 3.2 中公式 (*), 计算出每个插入点对应到 PPS 流形中的坐标，将 PPS 流形上的坐标作为每个插入点的新坐标，这样由于流形曲面的光滑性和表面凹凸性，每个插入点就可以有一个更合理的空间坐标，就能较好的避免“四点共面”的情形。

4.2 效果展示

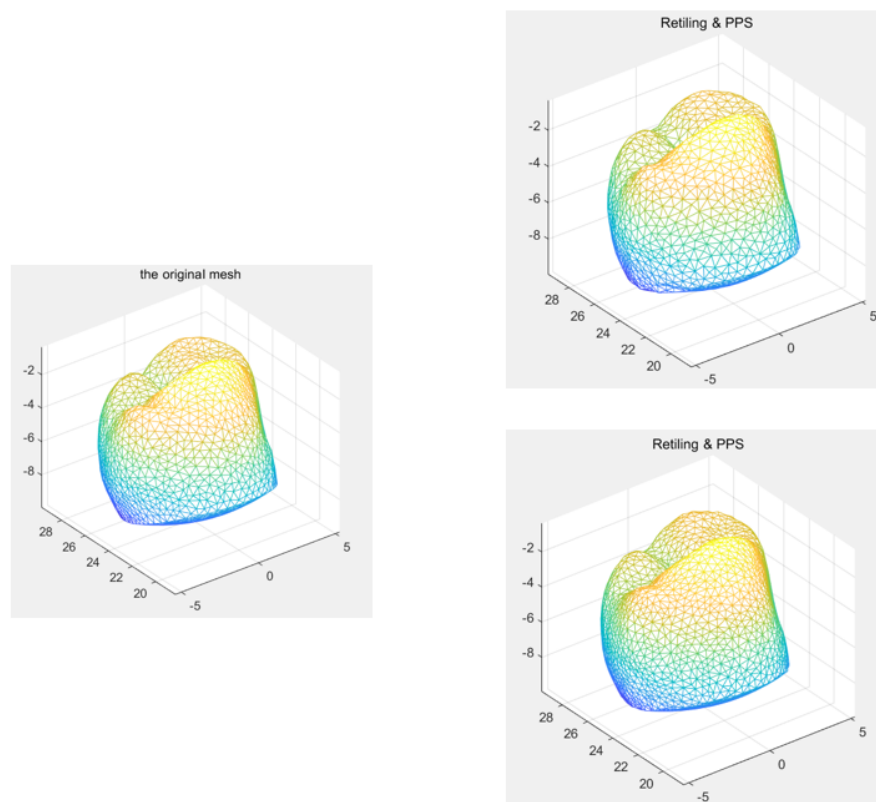


图 2

模型名称	顶点数	面数	目标顶点数	目标面数	耗时（单位：秒）
1	1967	3930	1458	2912	17s
			2424	4844	19s
2	19483	38962	14666	29328	222s
			24369	48788	261s

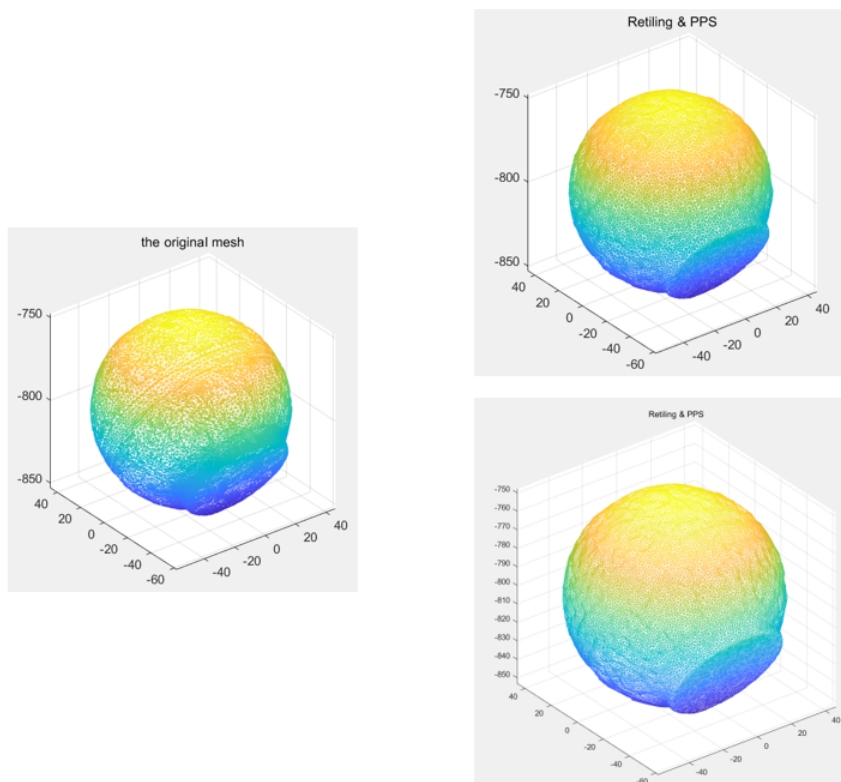


图 3

5 小结

点云到网格

在点云生成网格前，我们需要有一个初始网格，这个初始网格代表了点云所表示的物体的拓扑结构。下面是我们算法的流程：

Re-Tiling and PPS

输入：封闭的三角网格，目标顶点个数 n ，排斥力半径 r 。

- (1) 在网格上随机插入 n 个顶点；
- (2) 计算每个插入点之间的排斥力；
- (3) 根据每个插入点所收到的排斥力，将其在原始网格上移动；
- (4) 循环步骤 (2) (3) 达到迭代次数；
- (5) 计算出插入点在 PPS 流形上的位置，流形上的位置，更新现在的位置。
- (6) 移除原始顶点。

输出：顶点数为 n 的新网格，拓扑结构与原始网格相同。

6 参考文献

- [1].Greg Turk, Re-Tiling Polygonal Surfaces.
- [2].Marcelo Siqueira, Dianna Xu, A new construction of smooth surfaces from triangle meshes using parametric pseudo-manifolds.