想来想去，这个中医项目找到个能写论文的point真是不容易。说单纯做个general mesh simplification algorithm 又太难了，怎么想都比较trivial。

所以打算做个voxel model to mesh LOD的东西，先用基于MeshSlicer的Voxelizer（也就是我的mc 3d打印机）来产生实验数据（这个voxelizer可以在论文的实验数据部分提及一下，说下体素模型怎么来的，毕竟上网能下载的大部分的模型都是三角形网格）。

算法大致思想：**用不同大小的三维filter来对voxel model进行滤波/降采样**，voxel的是用0/1两个数来标记当前位置是否有方块，**但是经过down-sample以后可以是小数**，这样就可以用来做MC的线性插值了。得到更低分辨率的三维栅格之后，就可以用之前实现的Marching Cube来生成等值面。

注意：在MC进行三角形生成的时候要在voxel的六个边界面上填充0，不然处于边界的等值面可能无法生成（然后产生洞洞）：

例如：

000000000

001111000

111111111

111111111

000111100

000000000

看如上二维版MC也就是MarchingSquare，看第三四行的左右边，是无法生成等值边的

因为如果要用Marching Cube来生成三角形网格模型的话：在MC的相邻voxel/cube的等值面必须要连续，不然就不能做到几乎封闭---> 不同等值面与同一条棱的交点必须是唯一的---->因为down-sample这个过程，新的sample就会被很多原来附近的samples影响，所以相邻mc voxel在同一条棱上产生的交点有可能不一样。之前有点贪心，想要在mc算法生成等值面的时候就参考大于8个顶点来决定等值点的线性插值系数，这样子就可能在一条棱上生成不同位置的等值点，于是造成不连续性。 为了避免这种情况还是先生成降采样的所有顶点，然后再用普通的MC生成等值面**（但是要加上特殊的线性插值）**。

三维box Filter的长宽高会影响分辨率。

可以知道这不是topology preserving的，有些小洞可能会被down-sample这个过程吃掉

本论文描述的过程的使用场景

点云

|

v

三角形网格--->体素模型--->三角形网格LOD序列

^

|

CT切片组

现在优化了一波之后终于速度还算可以了，只是一个 简化质量-时间 相权衡的网格简化方法，**体素化分辨率越高质量越好，耗时越长**。分辨率与简化质量与具体模型的复杂度有联系。

这个算法耗费内存比较多，集中在体素模型的储存上。

算法可能只能应用于闭合多边形上

模型体素化的时候必须要被填充，不然表面极有可能会被过大的采样步长跳过