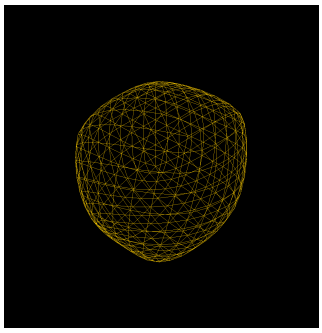
Lab2 报告

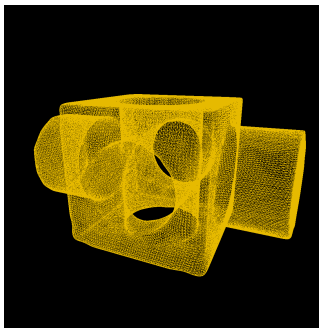
叶茂 2200017852

1 Loop Mesh Subdivision

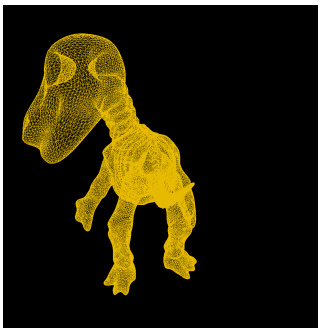
首先更新原有顶点，根据原有顶点及其周围顶点加权算出更新后顶点的坐标 $v = (1-n*u)v + \sum_{i=1}^{n} u * v_i$(当 $n = 3$ 时 $u = \frac{3}{16}$，否则 $u = \frac{3}{8n}$)。随后为每条边创造新的顶点，若边不位于边界，则由该半边的两个顶点和"跨过" 这条半边的两个顶点加权得到新顶点坐标；若边位于边界，我假设"不存在" 的跨过这条半边的顶点位于半边中点，随后同理计算新顶点坐标 $e_p = \frac{3}{8}(v_0 + v_2) + \frac{1}{8}(v_1 + v_3)$。最后进行插入面索引的工作，确保四个子面顶点顺序和原面的顶点顺序一致即可。



(a) cube



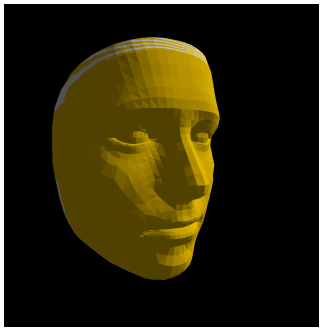
(b) block

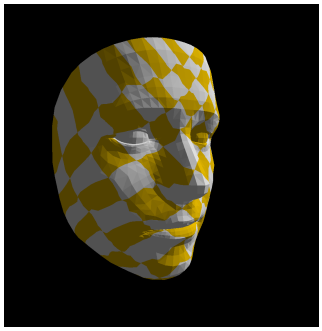


(c) dinosaur

2 Spring-Mass Mesh Parameterization

首先需要找出网格的边界点，在这里我遍历所有点，一旦遇到一个边界点，即可通过`BoundaryNeighbors()`函数逐一获取所有边界点的坐标。随后将所有边界点映射到 $[0, 1]^2$ 的边界上。对于每个不在边界上的点。采用 $Gauss - Seidel$ 迭代法，使用平均系数加权其所有邻居顶点坐标得到更新后的坐标。

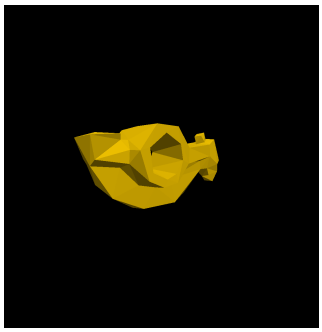

(a) 迭代 0 次


(b) 迭代 500 次


(c) 迭代 1000 次

3 Mesh Simplification

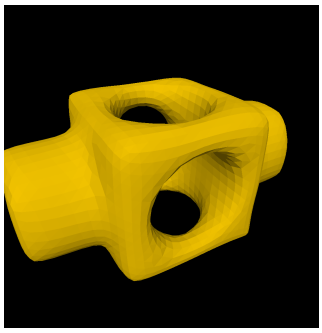
首先实现计算代价矩阵 Q 的函数: 获取面上三个顶点的坐标,利用`glm::cross()`函数计算该面的法向量，随后使用`glm::normalize()`将法向量归一化，代入任一顶点坐标即可得到该面的标准方程，根据 Q 的定义即可计算出 Q。对于`MakePair`函数，利用极值点梯度为 0 的性质，得到论文中的矩阵 $q = \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ q_{31} & q_{32} & q_{33} & q_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$ 计算该矩阵的行列式的绝对值，若小于 0.001，则认为其不可逆，以 0.0001 为步长遍历边上每一点，计算其代价 $v^T Q v$, 代价最小的点即为返回值；否则，该矩阵可逆，计算其逆矩阵，则 $q^-1 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$ 即为目标点，计算其代价即可。进行完边的坍缩操作后，需要对每个面和每个顶点的代价矩阵进行更新，计算新矩阵与原矩阵的差值对应更新即可。最后需要更新`pairs`数组，仅有顶点的邻居顶点所在环上的边需要进行更新操作，利用两重循环遍历每条边重新计算即可。


(a) rocker

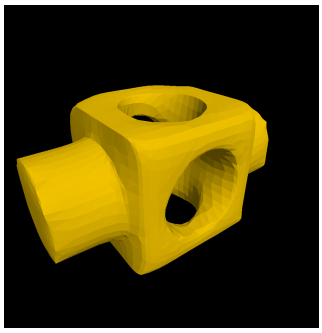

(b) arma

4 Mesh Smoothing

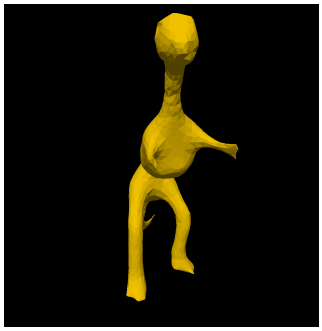
首先实现计算 cot 值的函数，对于给定的三个顶点 (包含角点)，计算角的两条边的内积，除以两条边的模长，即可得到 cos 值，算出 sin 值即可算出 cot 值。随后对于每个顶点，若`useUniformWeight`为真，则使用平均系数加权其每个邻居顶点；否则，计算每个邻居顶点的权重，对于每个 cot 值，小于 0.1f 的设为 0，大于 10.0f 的设为 10.0f。



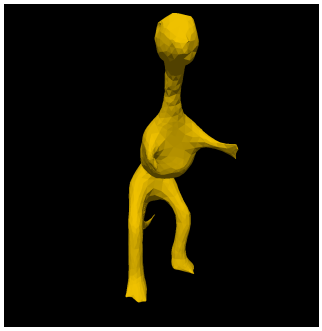
(a)
block(UniformWeight)



(b)
block(Cotangent)



(a) dinosaur(UniformWeight)

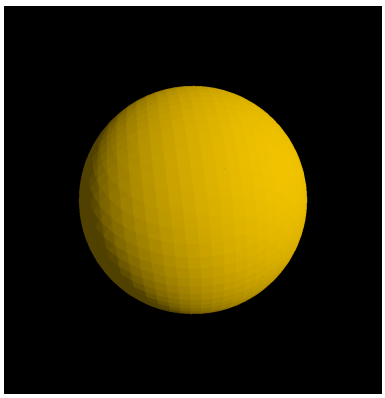


(b) dinosaur(Cotangent)

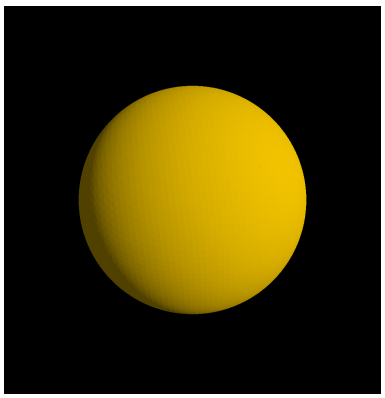
5 Marching Cubes

遍历网格结构的每个小立方体的起始顶点，对于小立方体的 8 个顶点，使用给定的`sdf`函数计算其到面的距离，设定一个`unsigned char`类型变量`idx`，若第 i 个顶点的 sdf 值大于 0，则通过位运算将`idx`第 i 位设为 1。利用该变量即可索引该小立方体的拓扑结构。对于小立方体上有 mesh 顶点的边，通过该边的两个端点的坐标差值得到 mesh 顶点的近似坐标。随后在`output.Positions`数组中查找该顶点是否存在，若存在则获取其下标，若不存在则将其加入数组后获取下标。对于一个面的三个顶点都做此操作即

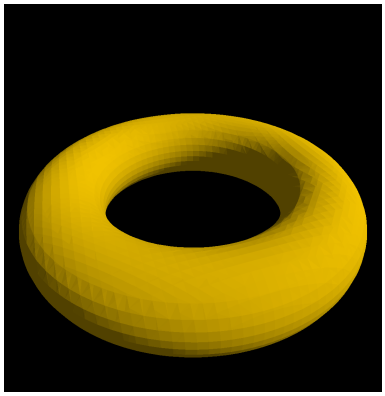
可得到三个索引，确定三个顶点的顺序，将它们加入`output.Indices`数组即可。



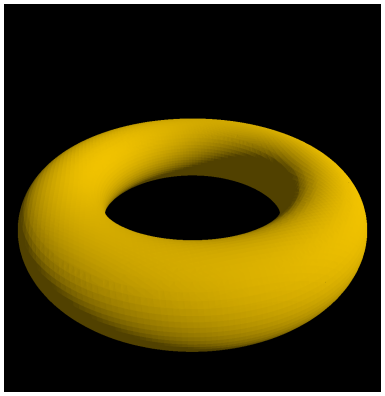
(a)
sphere(resolution=50)



(b)
sphere(resolution=100)



(a)
torus(resolution=50)



(b)
torus(resolution=100)