

Homework3

1. 使用任意软件对 bunny 模型进行网格简化，得到最终面片数量分别在 100~1000 和 1000~10000 的两个简化模型 (12')

(1) 使用 blender 导入 bunny 后，可以查看其面片数量接近 7w。

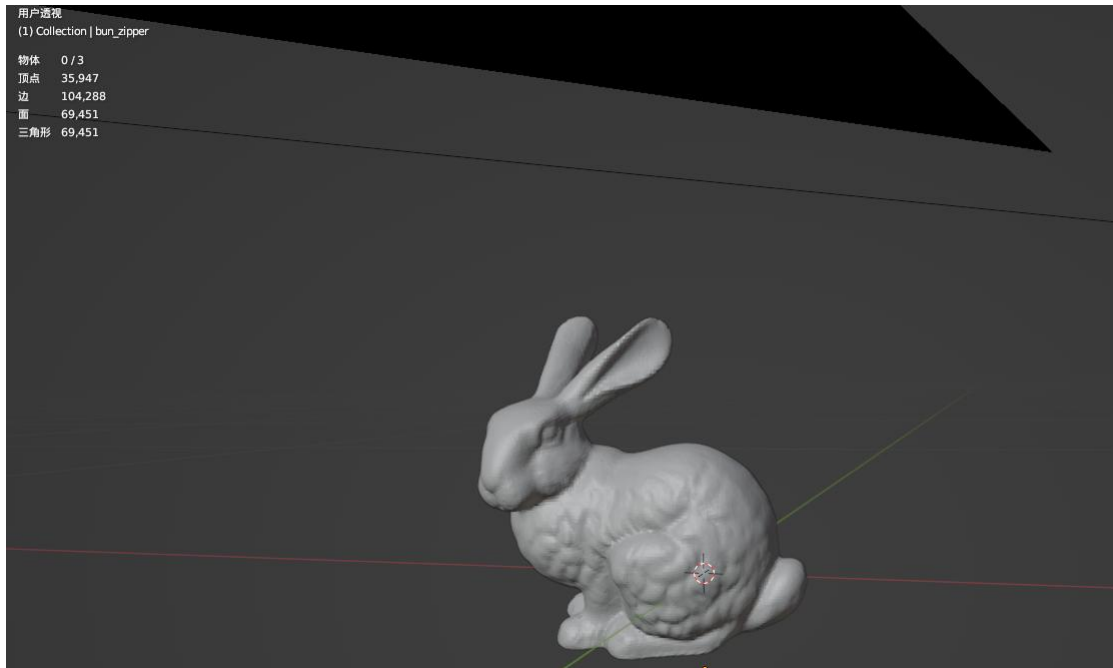


图 1 兔子面片数

(2) 在 blender 编写 python 使用顶点聚类算法进行减面，通过 cluster_size 控制减面程度，簇越大代表减面程度越大。代码如下：

```
import bpy
import bmesh
import mathutils

def vertex_clustering(obj, cluster_size=0.1):
    # 确保处于编辑模式
    bpy.context.view_layer.objects.active = obj
    bpy.ops.object.mode_set(mode='EDIT')

    # 获取 bmesh 对象
    bm = bmesh.from_edit_mesh(obj.data)

    clusters = {}
    cluster_centers = {}

    # 遍历所有顶点
```

```

for v in bm.verts:
    # 根据簇大小计算簇的 key
    key = (round(v.co.x / cluster_size), round(v.co.y / cluster_size), round(v.co.z /
cluster_size))

    if key not in clusters:
        clusters[key] = []
        cluster_centers[key] = mathutils.Vector((0, 0, 0))

    clusters[key].append(v)
    cluster_centers[key] += v.co

# 合并顶点
for key, verts in clusters.items():
    if len(verts) > 1:
        center = cluster_centers[key] / len(verts)
        for v in verts:
            v.co = center

# 移除重复顶点并更新 bmesh
bmesh.ops.remove_doubles(bm, verts=bm.verts, dist=0.0001)
bmesh.update_edit_mesh(obj.data)

# 返回对象到对象模式
bpy.ops.object.mode_set(mode='OBJECT')

# 获取当前选中的所有网格对象
selected_mesh_objects = [obj for obj in bpy.context.selected_objects if obj.type == 'MESH']

# 设定簇大小，值越大，减面程度越大
cluster_size = 0.005

# 对每个选中的网格对象应用顶点聚类算法
for obj in selected_mesh_objects:
    vertex_clustering(obj, cluster_size=cluster_size)
    print(f"Vertex clustering simplification completed for object: {obj.name}")

print("Vertex clustering simplification completed for all selected mesh objects.")

```

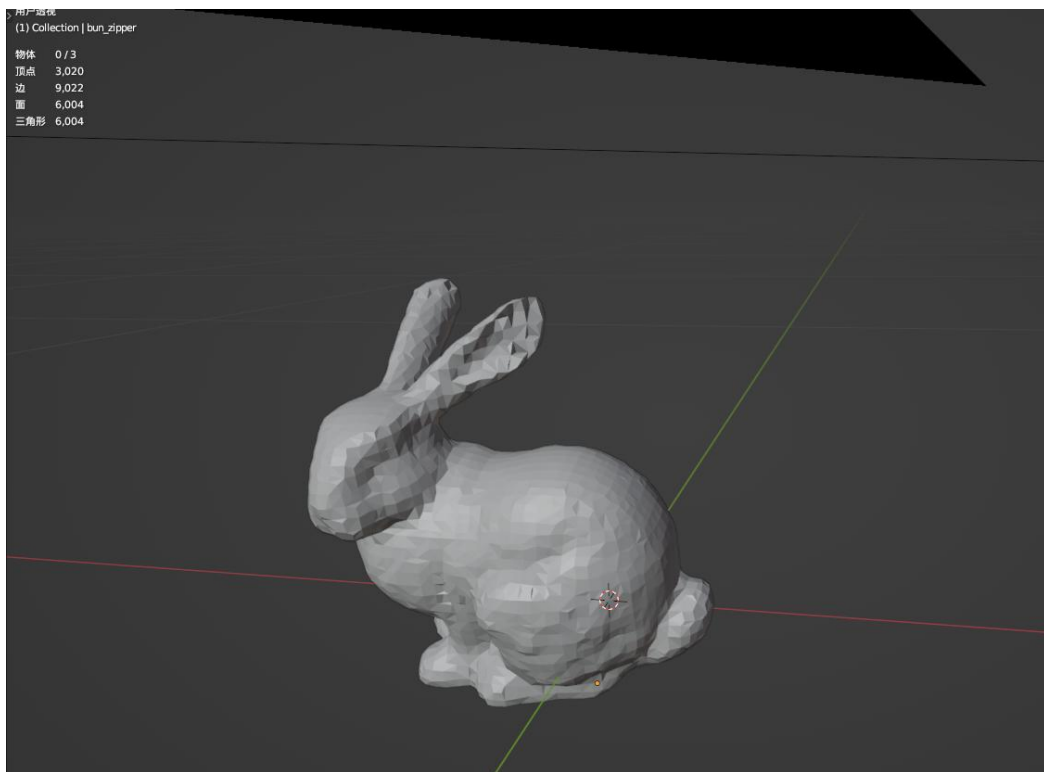


图 2 cluster_size = 0.005 面片 6000

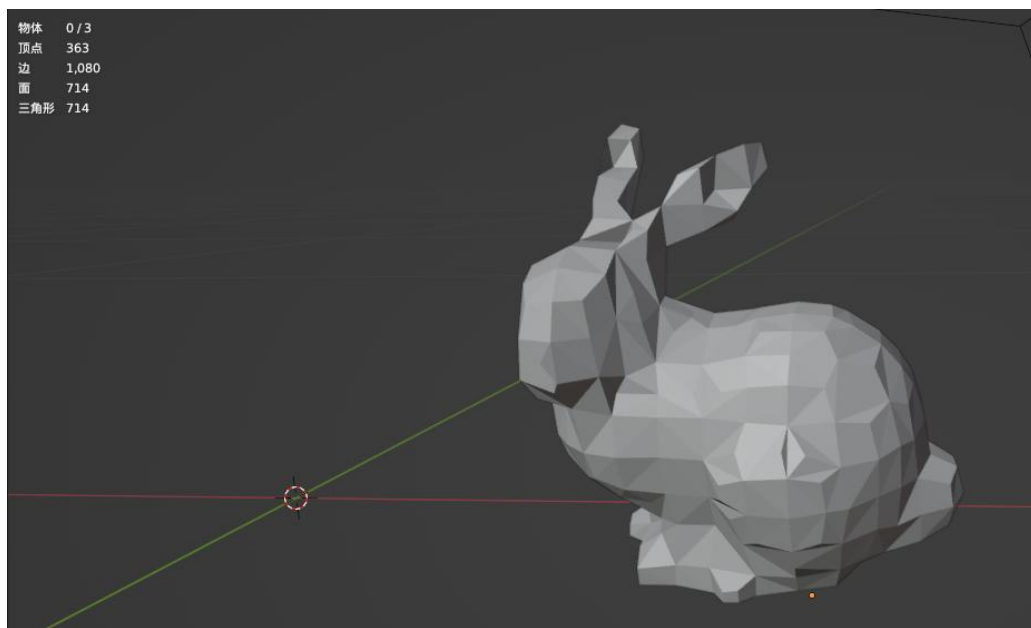


图 3 cluster_size = 0.015 面片 714

2. 使用任意软件对这两个模型进行渲染，你可以采用不同的材质和光照，对比效果 (6)
 - (1) 使用自定义材质，如下图：

