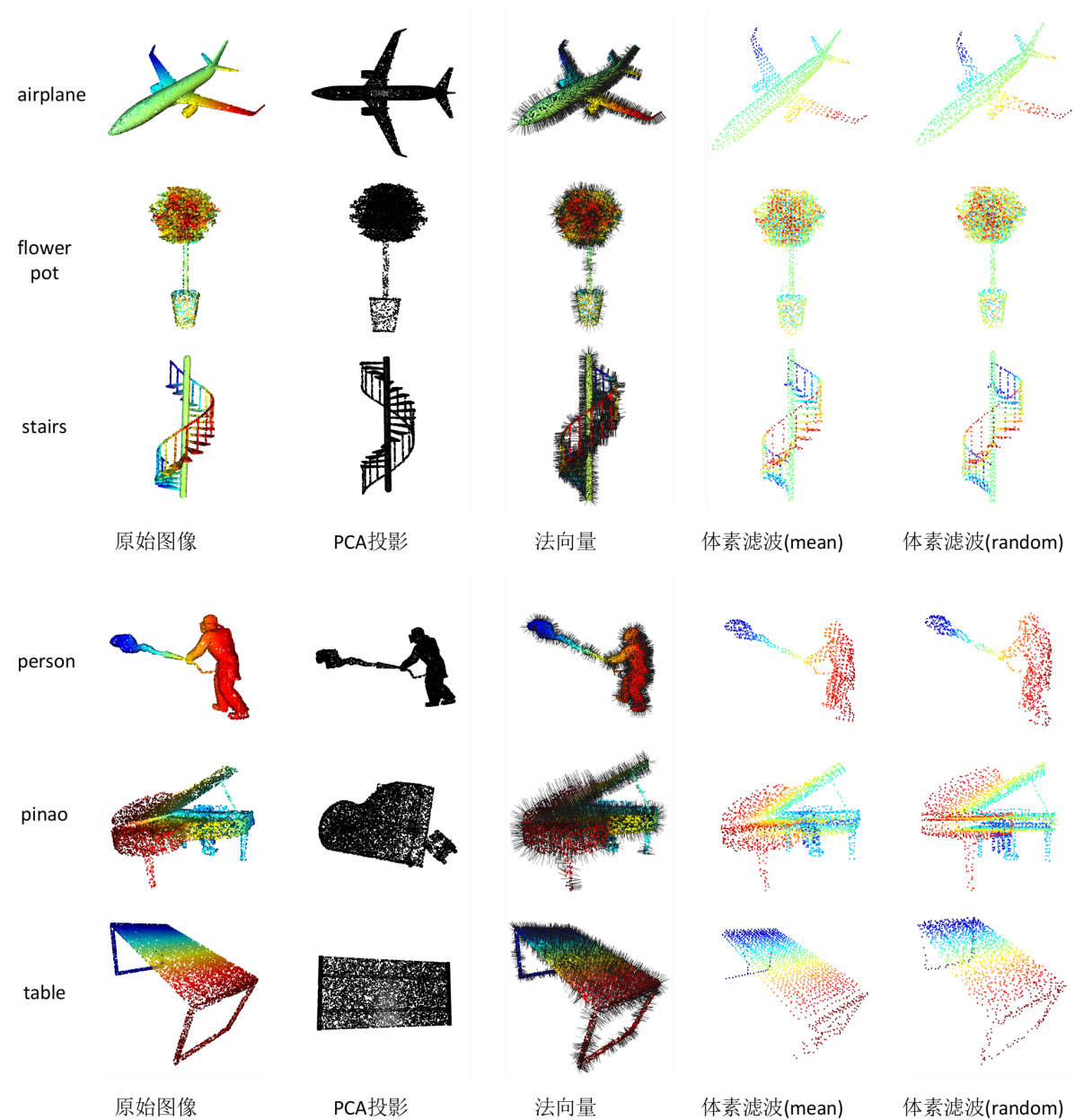


三维点云处理 - 作业 - 第一章

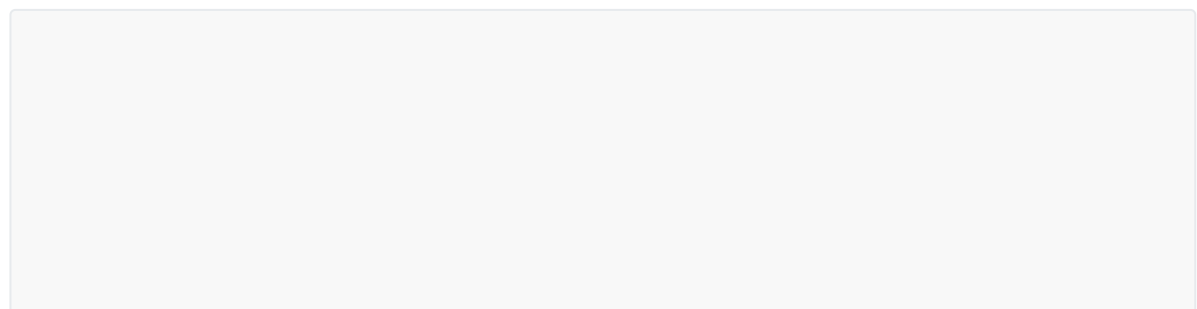
0. 结果展示

在40个类别中挑选了六个比较有代表性的进行展示, 全部40个种类的随机个体展示在附件 `results` 文件夹中. 从左到右依次是: 原始点云 -- PCA投影 -- 法向量可视化 -- 体素滤波(mean方法) -- 体素滤波(random方法).



1. PCA

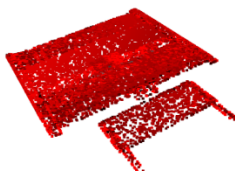
算法核心代码:



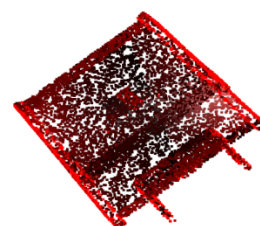
numpy 中计算协方差矩阵可以调用 `corrcoef` 和 `cov` 两个函数. `cov` 即使标准的协方差计算, 而 `corrcoef` 计算的是相关系数, 也可以看作是协方差, 但是剔除了变量量纲的影响. 相关系数R与协方差矩阵C的关系如下: $R_{ij} = \frac{C_{ij}}{\sqrt{C_{ii} * C_{jj}}}$. 因此计算出来的特征值和特征向量也不一样, PCA投影也不相同, 如下图所示:



原始图像



PCA投影(cov)



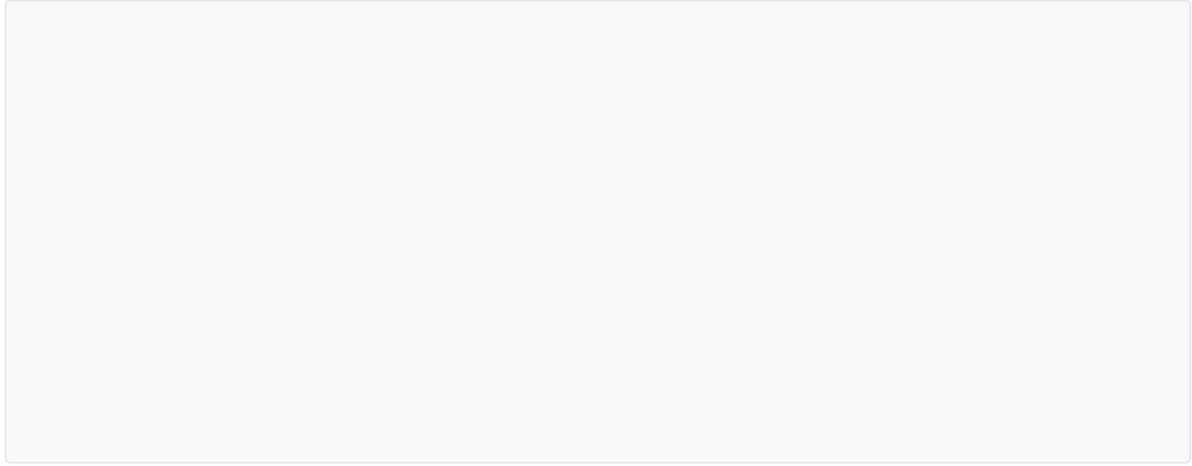
PCA投影(corrcoef)

可以看出这两种方法计算的结果大致一样, 但是又有所不同, 这可能是因为相关系数计算经过了方差的标准化, 是一种特殊的协方差.

注意: `cov` 与 `corrcoef` 函数的输入X的每一行代表一个变量, 每一列代表所有这些变量的单一观察. 所以应该传入 `3xN` 的矩阵. 如果没有做转置, 最终计算的结果将是 `NxN` 的, 不仅结果不正确, 而且会非常慢.

2. 法向量计算

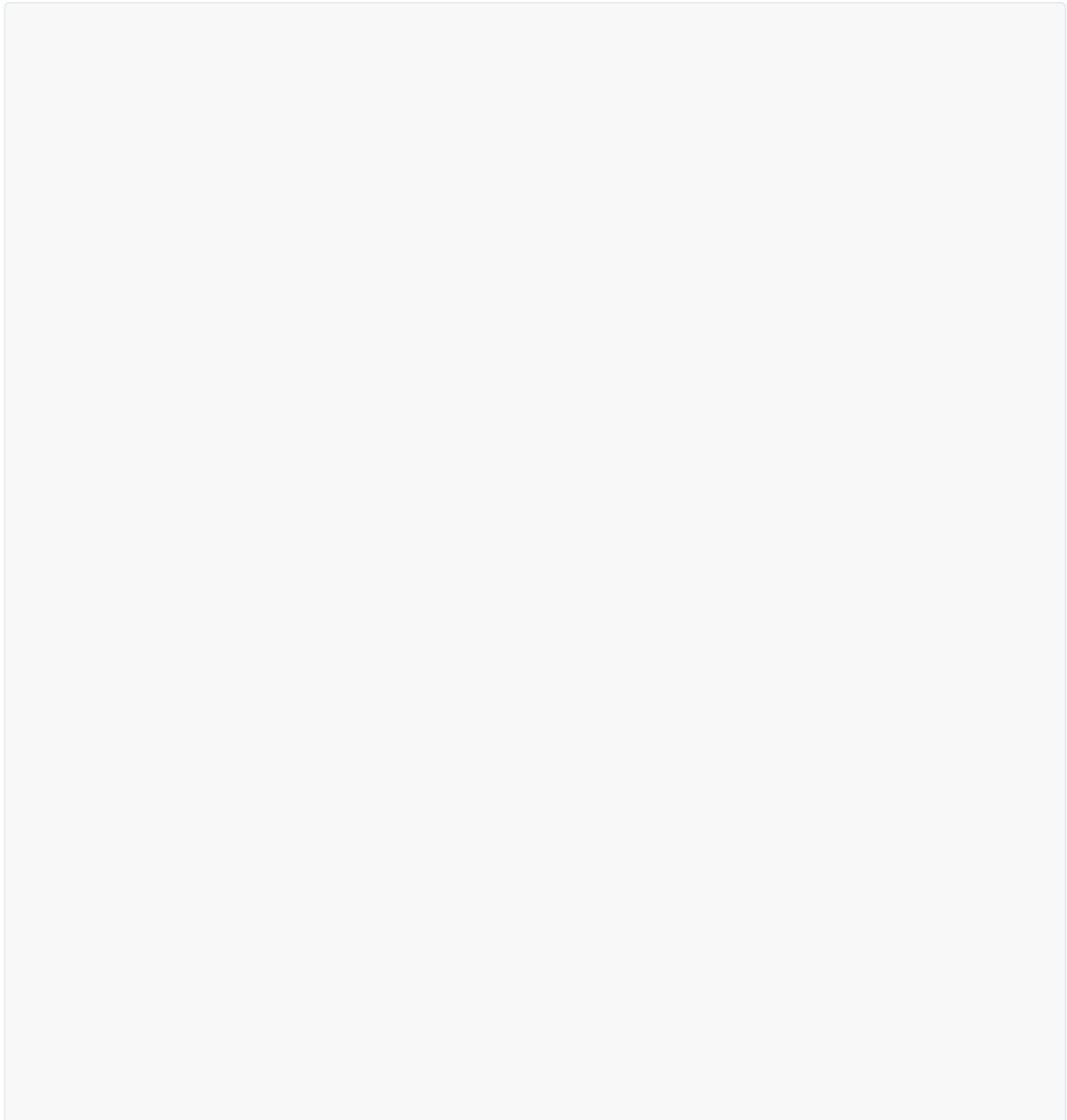
算法核心代码:



法向量计算时邻域的选择需要随着物体的特性调整, `search_knn_vector_3d` 选取的是距离目标点距离最近的k个点组成邻居, 对于形状平整, 边缘较少的区域, k值可以设置大一些, 有利于排除噪声的干扰;对于形状复杂, 表面变化频繁的区域, 可以设置小一点的k值, 以得到更加精确的法向量.

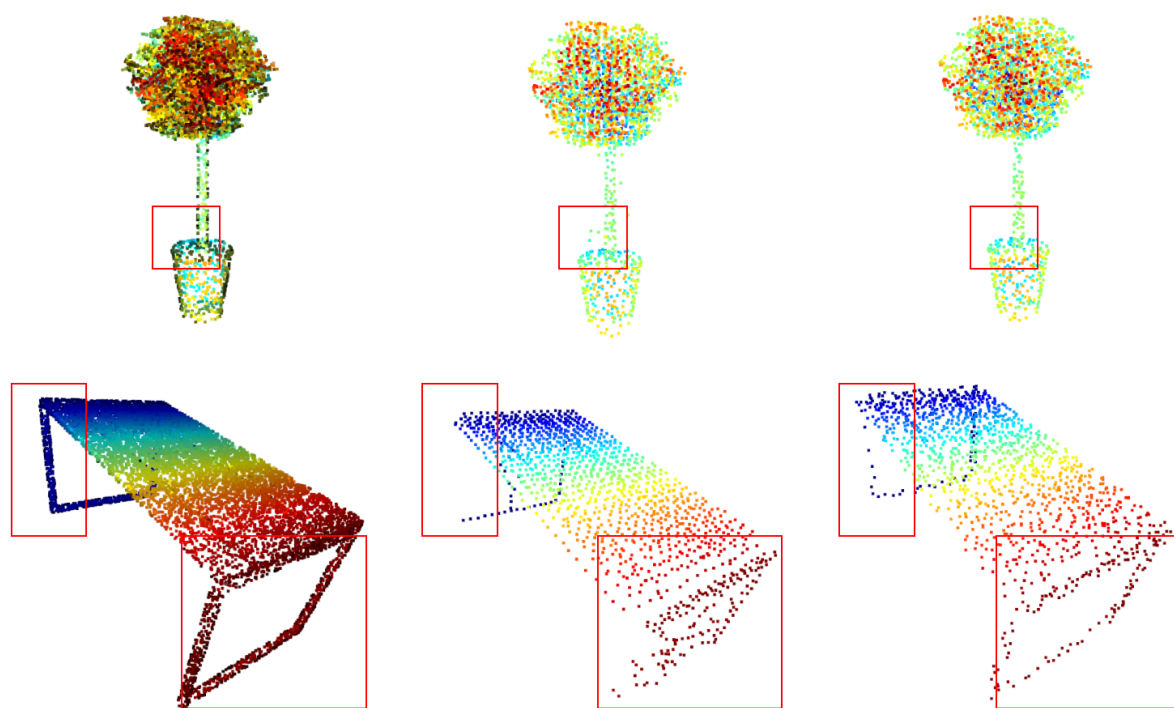
3. 体素滤波

算法核心代码:



实现的体素滤波有两种下采样方法, 一种是将落在同一voxel内的点做平均生成新的点输出, 另一种是在voxel内的点中随机选择一个点输出. 由于计算平均点输出的方法创造了新的点, 因此对数据的修改比较大, 效果没有随机取点好.

下图展示了这两种方法的结果对比, 可以发现mean方法在规则连续的区域效果比较好, 但是在角点, 细长以及边框区域效果很差.



原始图像

体素滤波(mean)

体素滤波(random)