



点云作业第七讲-----ISS

主讲人 杨雨辰



1. 使用radius NN 得到n个初始关键点,

threshold阈值:每个radius内的lamda大于某个数值

```
for i in range(len(nearest_idx)):
    eigvals.append(compute_cov_eigval(data[nearest_idx[i]]))
eigvals = np.asarray(eigvals) # 求解每个点在各自radius范围内的la
#print(eigvals)
lamda3_threshold = np.median(eigvals,axis=0)[2]*4 # 所有lamda
#print(lamda3_threshold)
```

为了突出特征点将阈值设置为所有 λ_3 的4倍
`compute_cov_eigal`计算后的三组特征值的最小值 > 每个radius 里
的最小的特征值

2. 计算有重叠的关键点群

```
unvisited = T # 未访问集合
while len(T):
    unvisited_old = unvisited #更新访问集合
    core = list(T)[np.random.randint(0, len(T))]
    # 从关键点T中随机选取一点作为关键点
    # np.random.randint 在所选范围内取一个随机整数
    unvisited = unvisited - set([core])
    visited = []
    visited.append(core)
```

2. 计算有重叠的关键点群

将初始关键点的索引设置为未访问集合, 用`np.random`在初始关键点的索引范围内取一个随机整数设置为关键点. 并将关键点放入访问集合.

然后遍历所有初始关键点, 如果某点是在为访问集合并且是这个关键点的邻域的话将这个点放入集合S. 将S放入从未访问集合中去除, 放入以访问集合. 避免重复计算.

2. 计算有重叠的关键点群

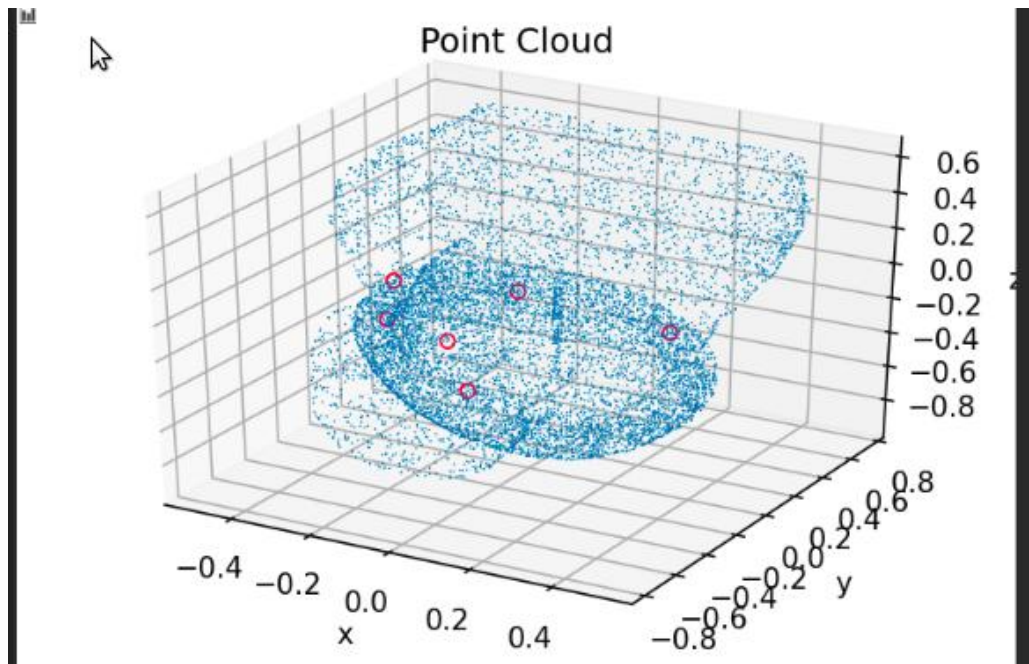
```
while len(visited): # 遍历所有初始关键点
    new_core = visited[0]
    if new_core in T:
        S = unvisited & set(nearest_idx[new_core])
        #print(S)
        visited += (list(S))
        unvisited = unvisited - S
    visited.remove(new_core) # new core 已做检测, 去掉new_core
    cluster = unvisited_old - unvisited # 有重叠(IOU)的关键点群
    T = T - cluster #去掉该类对象里面的核心对象, 差集
```

3. NMS

获取每个关键点的lamda3, 使用argmax返回lamda3的最大值的索引

```
#3. NMS 非极大抑制, 求解一个关键点群lamda3最大为关键点
cluster_lamda3 = []
for i in list(cluster):
    cluster_lamda3.append(eigvals[i][2]) # 获取每个关键点的lamda3
cluster_lamda3 = np.asarray(cluster_lamda3)
NMS_OUTPUT = np.argmax(cluster_lamda3) #argmax返回axis维度的最大值的索引
feature.append(list(cluster)[NMS_OUTPUT]) # 添加到 feature 特征点数组
return feature
```


效果图



效果图 使用深度相机获取的点云测试ISS, 发现ISS对噪点非常敏感.

