



三维点云处理第七次作业



主讲人 Lorenzo



- 分为两步：
- 第一步先找出那些“邻居位置变化较大”的点，作为候选点集
- 第二步再对候选点集做NMS，作为最后的关键点集

参数：

- r_1 ：用每个点 r_1 范围内的点来计算weighted covariance matrix，对应代码中的 `neighbors_salient_radius`
- r_2 ：用于NMS，定义需要被抑制的范围，对应代码中的 `non_max_radius`
- γ_{21}, γ_{32} ：用于判断每个点是否为关键点
- `iss_count`：关键点数量的上限

找出邻居位置变化较大的点，作为候选点集

- 对于每个点 p_i ：
 - 计算weighted covariance matrix
 - 计算weighted covariance matrix的特征值： $\lambda_i^1, \lambda_i^2, \lambda_i^3$
 - 根据条件 $\frac{\lambda_i^2}{\lambda_i^1} < \gamma_{21}$ and $\frac{\lambda_i^3}{\lambda_i^2} < \gamma_{32}$ 判断 p_i 是否为关键点，如果是，则将 i 收集到 `cand_idx` 里面

计算covariance matrix

- 计算covariance matrix

- 遍历点 p_i 的每个邻居 p_j ：
 - 计算 p_j 对 p_i 的权重，即 w_j
 - 将 w_j 累加到 `wsum`
 - 计算 $w_j(p_i - p_j)(p_i - p_j)^T$ ，累加到 `numerator`。注意 p_i 及 p_j 都是column vector
- `numerator`与`wsum`的商即为weighted covariance matrix

```
"""
calculate covariance matrix
"""
numerator = np.zeros((3,3))
wsun = 0
for j in neighbors_salient_radius[i]:
    # TODO: calculate the weight of p_j
    wj = 1
    # TODO: update numerator
    numerator += np.zeros((3,3))
    wsum += wj
cov = numerator / wsum
```

计算特征值

计算weighted covariance matrix的特征值： $\lambda_i^1, \lambda_i^2, \lambda_i^3$

```
"""  
calculate its eigenvalues  
"""  
# TODO: use "cov" to calculate eigenvalues  
eigenvalues, eigenvectors = 0, 0  
  
# TODO: sort eigenvalues in decreasing order  
lambda1, lambda2, lambda3 = 0, 0, 0  
lambda3_dict[i] = lambda3
```

关键点判断

根据条件 $\frac{\lambda_i^2}{\lambda_i^1} < \gamma_{21}$ and $\frac{\lambda_i^3}{\lambda_i^2} < \gamma_{32}$ 判断 p_i 是否为关键点，如果是，则将 i 收集到 `cand_idx` 里面

```
"""  
decide whether to keep the point  
"""  
  
# TODO: decide whether to keep pi  
keep = True  
  
if keep:  
    cand_idx.add(i)
```

对候选点集做NMS，作为最后的关键点集

第二步对 `cand_idx`s 做NMS，找出关键点集 `iss_idx`s。

- 对于 `cand_idx`s 里的点，再做NMS
 - 从 `cand_idx`s 中选取 λ_3 最大者，令它为点 `cand_idx`，并将它放入关键点集 `iss_idx`s
 - 查找 `cand_idx` 在 r_2 范围内的邻居，从 `cand_idx`s 中删除
 - 迭代进行上述两步，直到 `cand_idx`s 为空或关键点的数量 `len(iss_idx)` 大于等于 `iss_count`

选取 λ_3 最大者，放入关键点集

从 `cand_idx`s 中选取 λ_3 最大者，令它为点 `cand_idx`，并将它放入关键点集 `iss_idx`s

```
"""
find the cand_idx with max lambda3
"""
# TODO: find the cand_idx with max lambda3, using lambda3_dict
cand_idx = 0

"""
add the point into key points set
"""
iss_idx.append(cand_idx)
```


从候选点集删除关键点的邻居

查找 `cand_idx` 在 r_2 范围内的邻居，从 `cand_idx`s 中删除

```
"""  
delete its neighbors from candidates set  
"""  
# TODO: delete its neighbors from candidates set, using neighbors_non_max  
cand_idx = cand_idx
```

- weight的计算：应为 p_j 在 r_1 邻域内邻居数量的倒数，与距离无关
- covariance matrix的计算：直接调用np.cov，未加权
- p^{best} ：应该从未拜访的点挑选 λ_3 最大者当作 p^{best} ，而非随机选取
- 关键点的判断：未使用课件上提到的 γ_{21} 及 γ_{32}





深蓝学院
shenlanxueyuan.com

感谢各位聆听 !
Thanks for Listening

