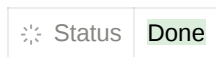


NICE-SLAM Mapping



主代码

程序共包含三个线程，分别对应如下三个函数：

1. `self.tracking`
2. `self.mapping`
3. `self.coarse_mapping`

其中 `self.mapping` 与 `self.coarse_mapping` 均对应 `Mapper` 类，功能对应 `run` 函数

Mapper

1. 关键帧选取
2. 采样
3. rendering

关键帧选取

Strategy 1: `global`

- Step 1: 获取需要选取的帧数=可参与迭代的帧数 - 2，其中 - 2指代最近一帧与当前帧
- Step 2: 随机选取
- Step 3: 添加最后一帧关键帧与当前帧的信息

Strategy 2: `overlap`

- Step 1: 根据当前帧的rgb, depth, pose等信息在空间中采样
- Step 2: 利用已知的keyframe的pose, 将采样投影到不同的keyframe上, 计算占据keyframe的比例
- Step 3: 根据计算出的比例选取关键帧
- Step 4: 添加最后一帧关键帧与当前帧的信息

Discussion: 为什么overlap的效果好？

采样

按照nerf的套路

Step 1: 根据图片大小, depth, rgb随机采样

Step 2: 根据内外参将采样点获取rays_o, rays_d

Step 3: 计算每个ray的范围[near, far]用于采样

模型训练

Step 1: 根据stage控制模型优化的部分

- 对于fine mapper, 先是middle stage, 再fine stage
- 每个stage的iter数量根据middle_iter_ratio和fine_iter_ratio来设置

Step 2: <https://github.com/cvg/nice-slam/blob/7af15cc33729aa5a8ca052908d96f495e34ab34c/src/utils/Renderer.py#L23>

Step 3: https://github.com/cvg/nice-slam/blob/7af15cc33729aa5a8ca052908d96f495e34ab34c/src/conv_onet/models/decoder.py#L312

Step 4: https://github.com/cvg/nice-slam/blob/7af15cc33729aa5a8ca052908d96f495e34ab34c/src/conv_onet/models/decoder.py#L262

- 获取grid中的feature, 如果是fine则为middle和fine的concat
- mlp
- 得到预测的occ

Step 5: render后的rgb和depth和真值计算loss, 并backward

Step 6: 将相机位姿更新回keyframe_dict中