

实验报告(G3)

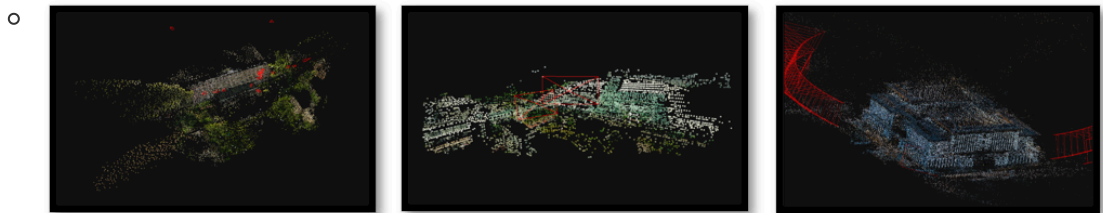
一、基础实现

1.1 场景重建

1.1.1 基础实现

- 重建方式与流程
 - 数据集的获取
 - 数据集来自于：拍摄的照片，录制或航拍视频抽帧，3D云平台提供的场景。
 - 特征点提取与匹配
 - 特征点提取方法(extractor): `disk`、`r2d2`、`superpoint_max`、`sift` 等。
 - 特征点匹配方法(matcher): 如 `Disk+lightglue`、`Superpoint+lightglue`、`NN-ratio`、`NN-superpoint` 等。
 - 模型重建
 - 利用 Colmap 实现 3D 点云与相机位姿信息的重建。

- 结果展示



- 除此之外，我们对于玉泉校门，基图与蒙民伟楼等校园区域进行重建。
- 同时，我们也尝试了对于组合建筑——基图、月牙楼与小剧场，进行重建，但由于数据集的影响，并未呈现出较好效果。
- 功能实现位于 `sfm_demo.ipynb` 文件之下。

1.1.2 对比探究

- 完成任务
 - 利用不同的特征点提取与匹配的方法，重建不同的建筑，对比其重建效果。
 - 功能实现位于 `comparison.ipynb` 文件之下。

- 对比结果展示



- 对比判断标准

- 重建点云稠密与稀疏
- 重建点云的 3D 结构
- 重建建筑的位置准确性

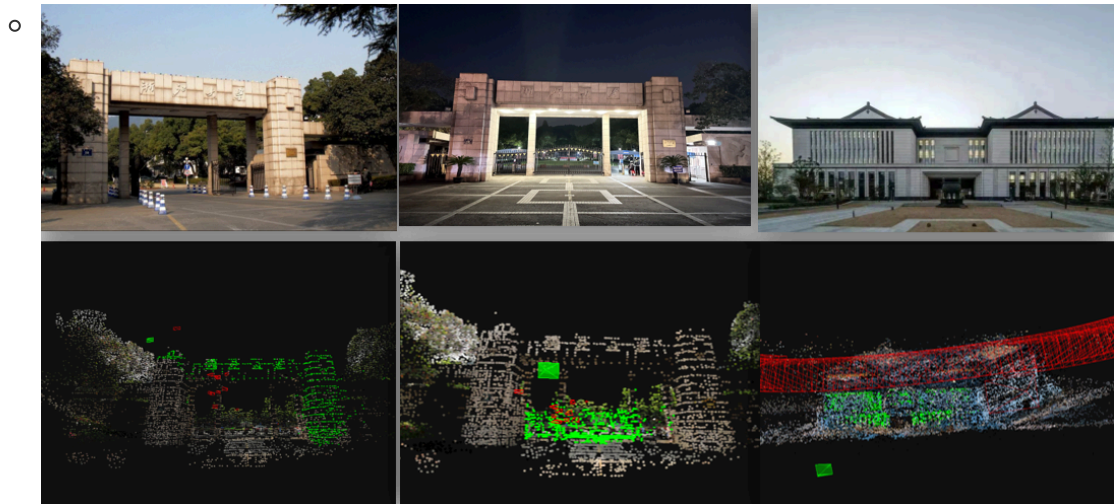
- 重建建筑的表面质量

1.2 图像定位

- 定位方式与流程

- 待定位图像的获取：拍摄的图片
- 关键帧的查找
 - 待定位的图像，数据集中检测到的近邻点。
- 特征点的匹配
 - 待定位图像，查找后的关键帧，局部特征点的匹配。
- 6DoF 位姿定位
 - 相机位姿与匹配特征点，3D 点云中的呈现。
- 功能实现位于 `Localization.ipynb` 文件之下。

- 定位结果



- 尝试不同视角，不同时间以及不同建筑的定位。

二、进阶探索

- 进阶探索中，我们尝试利用时序信息提高定位的精度。

- 问题呈现

- 在进行定位时，该模型容易受到建筑对称性影响，而单张图片难以提供更多定位信息，从而导致最终定位精度下降。
- 时序视频能够提供更多信息，利用时序信息，能够在一定程度上提升定位精度。
- 本次拓展探究选择主图作为探究对象。

2.1 图像拼接

2.1.1 方法与结果

- 利用前后帧进行拼接

- 在一段时序视频中，对于发生定位错误的某一帧，原因在于特征点匹配较少。

- 通过前后帧的提取与拼接，能够给该待定位帧提供更多的信息，从而实现更好匹配，提升定位精度。
- 代码尝试位于 `image_stitch_local` 文件之下。
- 帧的选择与拼接结果

◦



◦



2.1.2 方法漏洞与缺陷

- 利用该方法实现对于定位不准的帧进行精度提升，在测试过程中未能成功实现。
- 主要有以下几点原因：
 - 理论上
 - 拼接后的图像改变原有图像的信息，即使能够纠正定位，其依旧存在误差。
 - 实践上
 - 拼接效果差，难以稳定拼接图像；拓展信息少，特征匹配效果不明显；无法真正提升精度。

2.2 相机位姿矫正

2.2.1 方法与流程

2.2.1.1 找出错误帧

- 基于相邻两帧的相机位姿变换较小，我们首先找到相邻但是位姿变换较大的帧。
- 函数实现位于 `Localization.ipynb` 文件下 `check_pose_consistency()` 函数之下。

2.2.1.2 估计错误位姿

- 我们采用简单的几何均值来纠正错误的相机位姿，定义于 `fix()` 函数中。纠正算法定义如下：

- 第 i 帧错误，则分为两种情况：

如果 $(i+1)$ 帧正确：

$$\text{第 } i \text{ 帧} = (\text{第 } i-1 \text{ 帧} + \text{第 } i+1 \text{ 帧}) / 2$$

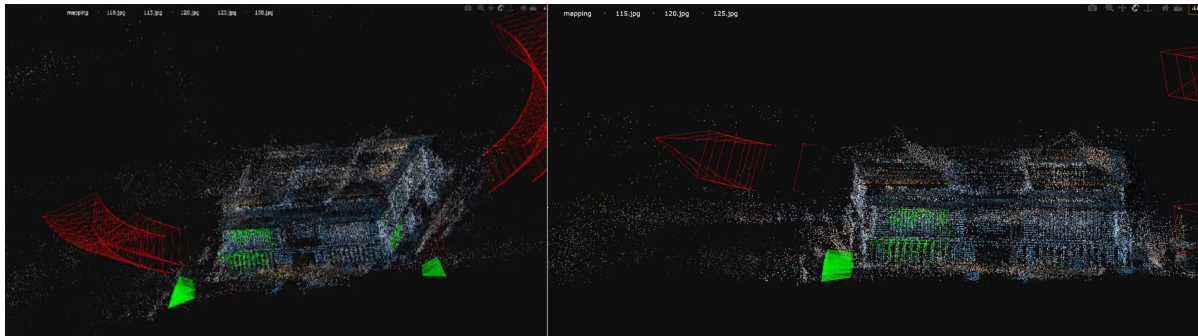
如果 $(i+1)$ 帧错误：

$$\text{第 } i \text{ 帧} = (\text{第 } i-1 \text{ 帧} * 2) - \text{第 } i-2 \text{ 帧}$$

- 纠错过程主要分为以下两步：
 - 初始化相机位姿的大致位置

- 通过 RANSAC 方式，对于每一帧，先计算前10帧相机位姿平均位置，再用前10帧的相机位姿进行一一比较，找出错误帧。
- 使用前10帧中没有问题的相机位姿重新估计相机位姿平均位置。
- 功能实现位于 `Localization.ipynb` 文件之下。
- 遍历修正所有帧
 - 对于每一帧，与相机中心位置比对，如果偏离较多，则使用 `fix()` 函数进行纠正。
 - 功能实现位于 `Localization.ipynb` 文件之下。

2.2.2 结果对比展示



- 可以看到，在没有被纠正前，第24帧的相机位姿明显偏离了预期，在纠正后，符合预期。

三、总结与反思

3.1 总结

- 项目原理：
 - 通过论文阅读，demo运行，理解 Hierarchical-Localization 的实现原理与方式。
- 基本实现：
 - 数据集获取：通过不同的途径获取数据。
 - 特征点提取与匹配：尝试不同extractor和matcher，比较结果。
 - 模型重建：对校园多个场景进行重建，进行结果分析。
 - 图像定位：展示定位流程与图像定位结果。
- 拓展探究：
 - 问题提出：探究方向选择的原因，以及存在的不足。
 - 方法尝试与探究：前后帧拼接、前后帧相机位姿纠错。

3.2 反思

- Hloc 实现对于 large scale 场景的定位，在小尺度上的定位表现效果较差。
- 未能最终得到 extractor 与 matcher 的最优匹配。
- 未对于定位精度进行量化，只做到主观判断。
- 对于“正确”定位的图像，未能提升精度，未探究出更合适的解决方案。