视觉重定位

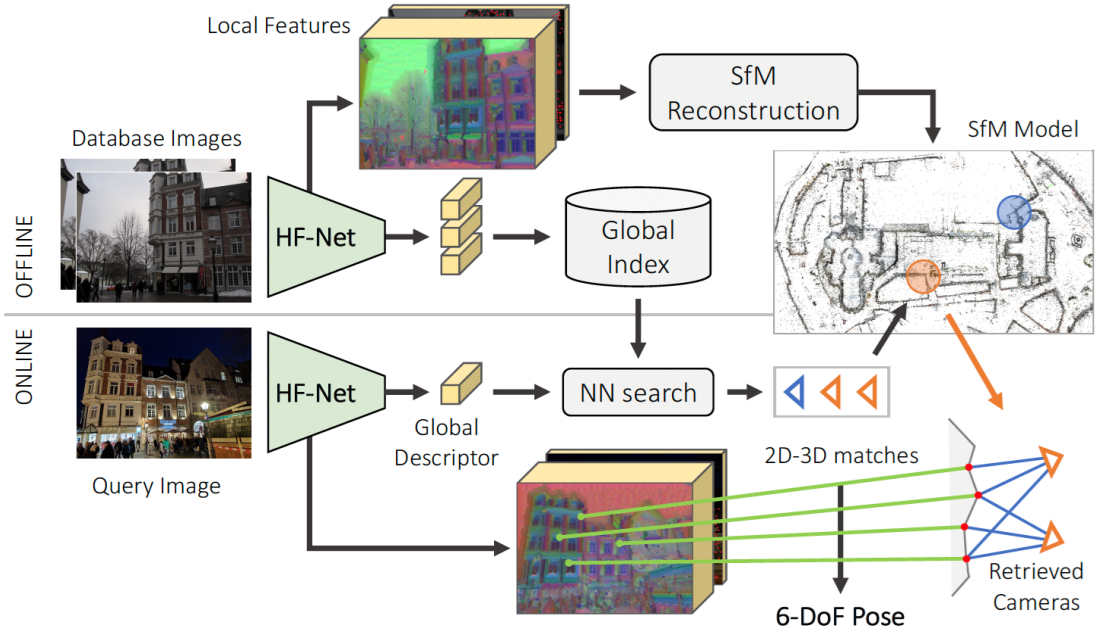

图 1. HLoc：使用图像检索与特征匹配进行层次重定位

背景介绍：

视觉重定位，是指根据当前相机观测的视觉信息来恢复相机在世界坐标系中的位置和姿态的过程。其本质是将当前观测信息与已有的重建模型进行匹配，从而估计其在世界坐标系下的位姿。

在深度学习的背景下，重定位算法在不同数据集上均取得了比较好的效果，目前最具有代表性的框架是 HLoc (hierarchical localization) [1]，流程如图 1 所示，主要包含如下几个步骤：1）图像检索， 2）特征检测与提取， 3）特征匹配， 4）估计位姿。离线重建的算法步骤和重建的点云模型格式是基于经典的重建算法 COLMAP[2]。

本次大作业项目是基于 OpenXRLab XRLocalization[3]的项目代码，该项目中图像检索/特征提取/特征匹配，各个模块都定义清晰，方便添加不同方法和进行扩展。

任务要求：

基本项：

1）以 OpenXRLab XRLocalization 算法代码为基础，添加**至少一种**图像检索/特征检测/特征匹配方法，并在视觉定位数据集上进行测试和分析，对比使用不同算法的效果区别

2)使用**至少一种**视觉定位数据集[4]进行测试分析,包括但不限于,Aachen Day-Night, InLoc, Extended CMU Seasons, RobotCar Seasons, 4Seasons, Cambridge Landmarks, and 7-Scenes。

3）展示不同方法的对比结果，定位姿态的 Recall (All conditions: (0.25m, 2°) / (0.5m, 5°) / (5m, 10°))，并且与 XRLocalization 中的 baseline 方法对比，包括 a. NetVLAD + SIFT + NN b.NetVLAD + SuperPoint + SuperGlue。

4) **注意：**本次大作业目的并不是要求在数据集上的定位精度超过 (NetVLAD + SuperPoint + SuperGlue)，而是对于视觉定位领域的认识与动手实践。因此，最后考核成绩会更加注重完成作业的工作量以及面对作业中各个模块所出现的问题的分析与解决办法。

加分项：

1）在两个以上定位数据集上进行测试对比。

2）自己完成两个以上的方法扩展，并且与 baseline 进行分析对比，并进行消融实验分析效果。

3）给出更多的定性和定量的对比结果以及原因分析。分析自己集成的方法为什么效果好，为什么效果不好。包括但不限于，a) 检索的 Recall 示例图以及对应的特征图可视化结果，分析检索特征提取的区域；b) 特征点检测以及特征点匹配的可视化结果；c）各个模块运行时间占比，耗时分析。

4）自定义展示的 AR Demo。在重建场景中放置最简单的立方体，将立方体渲染到指定的 query 图像中（可以简单的混合），可以通过贯穿 AR 物体的空间抖动来直观地感受到定位的准确度。

作业说明：

1）最后考核会参考作业所完成的工作量，不要只是简单把开源方法加进去，而没有对视觉重定位这个系统进行有效的分析和对比，作业需要有自己的思考和见解。

2）部分图像检索方法可供参考，Patch-NetVLAD[5], TransVPR[6], SARE[7], OpenIBL[8]。

3）部分特征检测方法可供参考，SIFT[9], R2D2[10], D2Net[11], SuperPoint[12]。

4）部分图像匹配方法可供参考，LoFTR[13], COTR[14], ECO-TR[15], SuperGlue[16]。

5）添加的方法不限于上述方法，上述所示方法基本都是具有开源代码，并且部分方法已经实现在 OpenXRLab XRLocalization 或者 HLoc 系统中，请选择没有实现在 XRLab 以及 HLoc 中的方法。

参考链接：

[1] HLoc: https://github.com/cvg/Hierarchical-Localization

[2] 重建算法 COLMAP: https://colmap.github.io/

[3] OpenXRLab xrlocalization: https://github.com/openxrlab/xrlocalization

[4] 视觉定位测试数据集平台: https://www.visuallocalization.net/benchmark/

[5] Patch-NetVLAD: https://github.com/QVPR/Patch-NetVLAD

[6] TransVPR: https://github.com/RuotongWANG/TransVPR-model-implementation

[7] SARE: https://github.com/Liumouliu/deepIBL

[8] OpenIBL: https://github.com/yxgeee/OpenIBL

[9] SIFT: https://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf

[10] R2D2: https://github.com/naver/r2d2

[11] D2Net: https://github.com/mihaidusmanu/d2-net

[12] SuperPoint: https://github.com/rpautrat/SuperPoint

[13] LoFTR: https://zju3dv.github.io/loftr/

[14] COTR: https://github.com/ubc-vision/COTR

[15] ECO-TR: https://dltan7.github.io/ecotr/

[16] SuperGlue: https://github.com/magicleap/SuperGluePretrainedNetwork