

回环检测

概述

有关回环检测的详细叙述可以参考《高翔SLAM》相关章节，或网上丰富的开源资料

<https://zhuanlan.zhihu.com/p/45573552>

简单来说，回环检测可以通过检测回环，从而对位姿进行矫正，消除累计误差实现全局优化。

具体到代码实现，可以分成两个环节：检测回环和回环矫正（全局优化），下面将分别讲述一下我对这两方面的理解。

检测回环

回环检测的第一步当然是检测回环，对于新出现的每一个关键帧，我们会将它与过去出现的关键帧进行比较，如果相似度超过某一阈值，我们认为出现了回环。

目前来说，检测回环的方法主要是“词袋”模型，如ORB-SLAM。具体原理可以看网上的博客，我在此阐述一下自己的理解。

动机

如何比较两帧之间的相似度？一个显而易见的想法是进行“特征匹配”，但是，如果基于特征点对两帧进行特征匹配的话，在SLAM运行期间，需要储存每一帧的上百个特征点的信息，占用大量内存，是不现实的。因此，出现了“词袋”模型，使用一串二进制数据来表征一帧的特征，从而实现帧与帧之间的比较。

流程

- 建立词典：词典由“单词”组成。我们对所有图像的特征点放在一起，进行“聚类”，最终得到的每一个聚类中心便是单词。可以说，每一个单词表征了一类特征点。值得一提的是，为了便于查找，词典一般会组织成K叉树的形式
- 特征表示：我们用一个长度等于单词数的二进制向量来表征一帧的特征。对于每一帧，我们将它的每一个特征点按照K叉树查找对应的单词。如果当前帧中存在某一单词，该单词在二进制向量中对应的位“置1”，否则置0，相当于one-hot向量的叠加。
- 相似度比较：比较两帧之间二进制向量的相似度时，有一些细节，参考相关资料。

代码实现

DBoW3库特别完善，从构建词典到相似度计算都有相关函数

回环矫正

回环矫正，即检测到回环后的处理。以ORB-SLAM3为例，它的回环检测方法比较经典，代码实现开源，网上也有许多讲解，可参考 [链接1](#)、[链接2](#)、[链接3](#)

- 先取出当前帧的共视关键帧，以当前帧的位姿为准，通过位姿传播矫正共视的局部关键帧的位姿
- 检查当前帧的地图点与经过闭环匹配后该帧的地图点是否存在冲突，对冲突的进行替换或填补
- 将闭环帧的局部地图点s投影到当前关键帧及其共视帧中，进行匹配，融合，新增或替换当前关键帧及其共视帧中的地图点

- 位姿图优化
- 全局BA

在我们vslam上的改进和应用

难点

和传统SLAM，如ORB-SLAM相比，vslam的输入数据为BEV分割图，因此没有适合用来建立词典的特征点。有一些工作，如Bag of Semantic Visual Words（2019IROS），提出了一种语义词袋的方法，根据分割图建立词典，可以试试。

在我的代码实现中，先省去了使用“词袋”模型的检测回环步骤，也由此避开了特征点的问题，“人工”选择当前帧和回环匹配帧，从而只进行回环矫正。

简化版代码实现

在林泽夫学长的avp-slam代码基础上，加入了回环矫正，代码见<https://github.com/ucasmjc/vslam/blob/main/src/loopclosing.cpp>。

- processing()函数处理数据，生成局部关键帧的点云和位姿
- correctloop()对人工选择的当前帧和回环帧进行回环矫正和位姿图优化。
- OptimizeEssentialGraph()为位姿图优化的函数，使用g2o库进行图优化。图的顶点为每个关键帧的位姿，边（约束条件）有3种，分别是当前帧与其共识帧之间、相邻关键帧之间和当前帧和回环帧之间。

问题

这代码主要针对“轨迹不闭合”的情况，但是我们的imu轨迹是闭合的，而且icp的结果轨迹也是闭合的（针对最后一帧和第一帧），所以没有回环矫正的意义。

代码中有许多“魔数”，如要直接使用，需根据代码逻辑调整。