

目录

1. Tracking::Track()	1
1.1 StereoInitialization()	1
1.2 MonocularInitialization()	1
1.3 TrackReferenceKeyFrame ()	1
1.4 TrackWithMotionModel ()	2
1.4.1 SearchByProjection ()	2
1.5 Relocalization ()	3
1.5.1 DetectRelocalizationCandidates (mCurrentFrame)	3
1.5.2 SearchByBoW(pKF , mCurrentFrame , vvpMapPointMatches[i])	4
1.6 TrackLocalMap()	4
2. LocalMapping::Run()	5
2.1 ProcessNewKeyFrame ()	5
2.2 SearchInNeighbors ()	6
2.2.1 ComputeDistinctiveDescriptors ()	6
3. LoopClosing::Run()	7
3.1 DetectLoop()	7
3.1.1 DetectLoopCandidates (mpCurrentKF, minScore)	7
3.1.2	7
3.2 ComputeSim3()	8
3.2.1 SearchBySim3(mpCurrentKF, pkF, vpMapPointMatches, s, R, t, 7.5)	8
3.2.2 OptimizeSim3(mpCurrentKF,pKF,vpMapPointMatches,gScm,10,mbFixScale)	8
3.2.3SearchByProjection (mpCurrentKF,mScw,mvpLoopMapPoints,mvpCurrentMatchedPoints,10)	9
3.3 CorrectLoop()	9
3.3.1 SearchAndFuse (CorrectedSim3)	10
3.3.2 OptimizeEssentialGraph()	10
3.3.3 RunGlobalBundledAdjustment (nloopKF)	11

ORB-SLAM2 自带摄像头: https://blog.csdn.net/zhangqian_shai/article/details/88406981

G201: <https://mp.weixin.qq.com/s/j9h9IT14jCu-VvEPHQhtBw>

G202 (顶点): <https://mp.weixin.qq.com/s/12V8iloLwVRPahE36OIPcw>

G203 (边): <https://mp.weixin.qq.com/s/etFYWaZ6y4XPiXrfqCm53Q>

G20 : <https://zhuanlan.zhihu.com/p/36889150>

1. Tracking::Track()

① 地图初始化

1.1 StereoInitialization()

- A. 特征点数大于 500, 设置初始位姿为单位矩阵, 将当前帧 (第一帧) 构建为初始关键帧, 并将其添加进局部地图中。
- B. 将特征点构造地图点, 通过反投影变换获取该特征点在世界坐标系下的 3D 坐标。

1.2 MonocularInitialization()

- A. 创建单目初始器, 特征点数目大于 100,。
用当前帧 (mCurrentFrame) 更新上一帧 (mLastFrame), 将上一帧 (mLastFrame) 的特征点保存在 mvbPrevMatched 中。
- B. 进行 ORB 特征点匹配(SearchForInitialization), 匹配关系保存在 mvIniMatches 中, 返回匹配点数 nmatches (要求要大于 100)。
- C. 通过 H 或 F 模型进行初始化。得到两帧之间的相对运动 (Rcw, tcw), 匹配关系存储在 mvIniMatches 中, 三角化后的空间点集合保存在 mvIniP3D 中。
- D. 将初始化第一帧作为世界坐标系 (单位矩阵), 求出当前帧 (mCurrentFrame) 的变换矩阵 Tcw。
- E. 将三角化后的 3D 点包装成地图点存入 KeyFrame 和 Map 中。

② 进入 SLAM 模式

- A. 检查并更新上一帧被替换的地图点。
- B. 如果运动模型 (mVelocity) 为空或当前帧紧靠上一帧, 则用参考关键帧跟踪

1.3 TrackReferenceKeyFrame ()

- a. 计算当前帧的词袋模型
 - b. 通过词袋加速当前帧 (mCurrentFrame) 与参考帧 (mpReferenceKF) 之间的特征点匹配, 将匹配关系存储在 vpMapPointMatches 中。返回匹配点数目 (nmatches)。nmatches 小于 15, 认为跟踪失败。
 - c. 用上一帧的位姿 (mLastFrame) 作为当前帧 (mCurrentFrame) 的位姿初始值。
 - d. 对当前帧 (mCurrentFrame) 的位姿进行 BA 优化 (PoseOptimization ())。
 - e. 对优化后的匹配点剔除外点 (3D), 统计匹配的内点 (3D) 数目 (nmatchesMap++), nmatchesMap 大于 10, 则认为跟踪成功。
- C. 如果不满足 B 中的条件, 则进行恒速模型跟踪

1.4 TrackWithMotionModel ()

- a. 更新上一帧的位姿 (UpdateLastFrame ())
 - 获取上一帧 (mLastFrame) 的参考关键帧 (pRef), 计算出参考关键帧到上一帧的位姿变换 (Tlr), 计算上一帧在世界坐标系下的位姿。
 - 对于双目或者 RGB-D。获取上一帧 (mLastFrame) 有效的特征点 (深度大于 0), 将他的深度值和 ID 存入 vDepthIdx 中。并从小到大排序。
 - 遍历 vDepthIdx, 判断其每个 ID 所对相应的特征点是否在上一帧中的地图点中, 或者并没有被观测到, 便新增地图点 (bCreateNew = true)
 - 新增地图点 (pNewMP), 将其添加进上一帧 (mLastFrame) 的地图点 (mvpMapPoints) 中。
- b. 根据恒速模型 (mVelocity) 得到当前帧 (mCurrentFrame) 的位姿
- c. 进行投影匹配

1.4.1 SearchByProjection ()

SearchByProjection(mCurrentFrame, mLastFrame, th, mSensor==System::MONOCULAR)

- 建立旋转直方图
源代码: `const float factor = 1.0f / HISTO_LENGTH` 改为:
`const float factor = HISTO_LENGTH / 360.0f`
- 获取当前帧相对于上一帧的平移向量 tlc, 判断当前帧是前进还是后退 (bForward, bBackward)。
- 将上一帧的地图点投影到当前帧, 获得其像素坐标
 1. 获取上一帧 (mLastFrame) 的地图点 (pMP) 在世界坐标系下的坐标 (x3Dw), 然后再经过当前帧的旋转和平移 (Rcw, tcw) 计算出其在当前帧相机坐标系下的坐标 (x3Dc)。再将其投影, 获得其像素坐标 u, v。
 2. 计算上一帧 (mLastFrame) 地图点对应的二维特征点在金字塔的层级 (nLastOctave), 设置搜索半径 radius。
- 获取当前帧搜索范围内的所有特征点 (2D) ID (vIndices2)
GetFeaturesInArea (u, v, radius, nLastOctave)
获取上一帧 (mLastFrame) 的地图点 (pMP) 的描述子 dMP
- 1. 遍历 vIndices2 中的特征点所对应的地图点, 如果存在, 则退出循环。
 2. 对于双目或 RGB-D 相机, 计算出当前特征点在右图中的像素坐标 (ur), 计算误差 er, 误差大于 radius, 退出。
 3. 计算当前帧和上一帧之间的描述子 (d, dMP) 距离, 获得最小描述子距离 (bestDist) 及其对应的 ID (bestIdx2)。
 4. 当 bestDist 小于设定的阈值 (TH_HIGH), 将上一帧中最佳 ID (bestIdx2) 对应的地图点 (pMP) 保存进当前帧的地图点中, 匹配数量 (nmatches) 加 1。
- 计算匹配点旋转角度差坐在的直方图

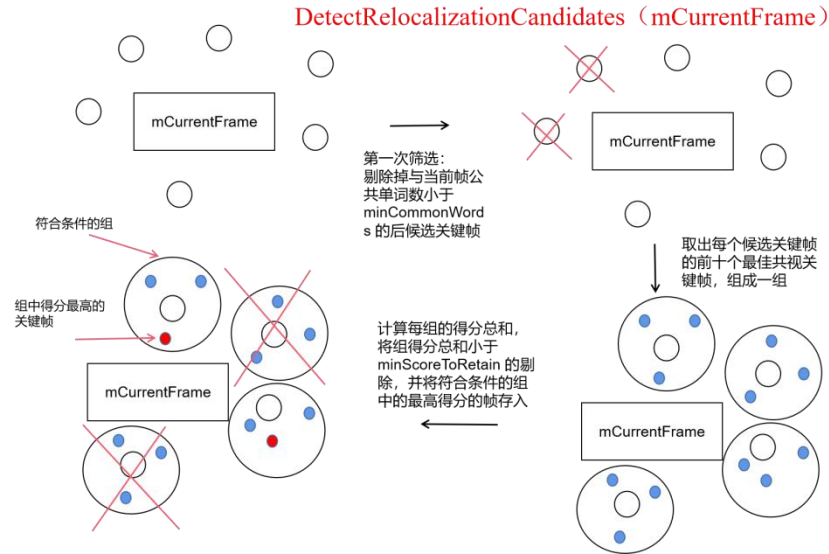
- 进行旋转一致性检查，删除不一致的匹配点，最终返回 `nmatches`。
 - d. 对当前帧进行 BA 优化 `PoseOptimization()`。
 - e. 对优化后的匹配点剔除外点（3D），统计匹配的内点（3D）数目（`nmatchesMap++`），`nmatchesMap` 大于 10，则认为跟踪成功。（d，e 步和 `TrackReferenceKeyFrame()` 中的一样）。
- D. 如果上述跟踪不成功，进行重定位跟踪

1.5 Relocalization（）

- a. 计算当前帧的词袋
- b. 用词袋找到与当前帧相似的候选关键帧，返回 `vpCandidateKFs`

1.5.1 DetectRelocalizationCandidates（mCurrentFrame）

- 找出与当前帧具有共同 `wordId` 的所有关键帧（`lkFs`），遍历 `lkFs`，对没有标记的关键帧（`pKFi`）进行重定位候选帧标记（`mnRelocQuery`），将其（`pKFi`）存入 `lkFsSharingWords` 中，重定位单词数（`mnRelocWords`）加 1。
- 遍历 `lkFsSharingWords`，统计关键帧组（`lkFs`）中与当前帧（`mCurrentFrame`）具有的最多公共单词的数量（`maxCommonWords`），设定最小单词数 $\text{minCommonWords} = \text{maxCommonWords} * 0.8$ 。
- 再次遍历 `lkFsSharingWords`，取出所有候选关键帧（`pKFi`），对满足公共单词数满足 `minCommonWords` 要求的候选关键帧，利用词袋模型计算与当前帧（`mCurrentFrame`）的得分 `si`，并将得分和符合条件的候选关键帧（`si`，`pKFi`）共同存入 `lScoreA`
- `ndMatch` 中。
- 遍历 `lScoreAndMatch`，取出其中的关键帧 `pKFi`，获取它 `pKFi` 的前十个最佳共视关键帧（`vpNeighs`），遍历 `vpNeighs`，将符合条件的关键帧得分全部总和存入 `accScore` 中，统计得到该组中得分最高的关键帧存入 `pBestKF`，将最高分数存入 `bestScore` 中。最后将总分和最佳关键帧（`accScore`，`pBestKF`）存入 `lAccScoreAndMatch` 中。统计各组中总分最高的组得分（`bestAccScore`）。
- 设置阈值 $\text{minScoreToRetain} = 0.75 * \text{bestAccScore}$ ，遍历 `lAccScoreAndMatch`，取出总得分 `si` 和总得分大于阈值 `minScoreToRetain` 的组中最佳关键帧 `pKFi`，将 `pKFi` 存入 `vpRelocCandidates` 中。



c. 遍历 vpCandidateKFs, 将候选关键帧 (pKF) 和当前帧进行词袋匹配

1.5.2 SearchByBoW(pKF, mCurrentFrame, vvpMapPointMatches[i])

- 获取关键帧 (pKF) 的地图点 vpMapPointsKF, 取出该关键帧的词袋特征向量 vFeatVecKF, 构建旋转直方图。
- 取出词袋中该节点下的特征点索引, vIndicesKF (pKF 的) 和 vIndicesF (mCurrentFrame 的)。
- 遍历 vIndicesKF, 取出索引所对应的特征点的地图点 (pMP), 计算该特征点的描述子 (dKF)。
- 遍历 vIndicesF, 计算相应的特征点的描述子 (dF), 计算 dKF 和 dF 的距离 dist。记录最佳距离和次佳距离 bestDist1, bestDist2。
- 进行匹配地图点点筛选, 将合格的存入 vpMapPointMatches。
- 删除误匹配的的点, 返回匹配成功的点数 (nmatches)。

d.

- 进行 EPnP 计算位姿, 返回 Tcw
- Iterate (5, bNoMore, vbInliers, nInliers)
- 对位姿 Tcw 进行 BA 优化, 返回内点个数 nGood
- 如果内点个数小于 50, 进行投影匹配 (SearchByProjection 见 1.4.1) 生成新的匹配
-

③ 仅定位模式。

④ 局部地图跟踪, 更新 mState

1.6 TrackLocalMap()

a. 更新局部地图 (更新局部关键帧和局部地图点)

b. SearchLocalPoints ()

- 遍历当前关键帧 (mCurrentFrame) 的地图点 pMP, 更新观测, 进行标记, 在后面的搜索匹配中不投影
- 遍历局部地图点, 判断该地图点是否在当前帧的观测范围内,

- nToMatch 加 1。
 - nToMatch 大于 0，进行投影匹配 SearchByProjection()（见 1.4.1），获得更多的匹配点。
 - c. BA 优化位姿
 - d. 更新当前还帧的地图点被观测程度，并统计跟踪到的地图点数目（mnMatchesInliers）。
 - e. 根据跟踪匹配数和重定位情况判断是否跟踪成功
- ⑤ 更新显示线程中的图像、特征点、地图点等信息。
- ⑥ 如果跟踪成功，更新很俗运动模型 mVelocity。
- ⑦ 清空观测不到的点
- ⑧ 删除临时地图点 mlpTemporalPoints
- ⑨ 插入关键帧 CreateNewKeyFrame（）
- ⑩ 删除外点
- ⑪ 如果跟踪失败，重置系统
- ⑫ 记录位姿 Tcr（由参考关键帧到当前帧的位姿变换），将 Tcr 存入 mlRelativeFramePoses 中。
- ⑬ Tracking 线程结束

2. LocalMapping::Run()

- ① 不接受关键帧
- ② 处理列表中的关键帧

2.1 ProcessNewKeyFrame（）

- A. 取出关键帧队列（mlNewKeyFrame，追踪线程向局部地图线程插入的关键帧队列）中第一帧作为当前关键帧（mpCurrentKeyFrame），然后将取出的关键帧从队列删除。
- B. 计算当前关键帧（mpCurrentKeyFrame）的词袋向量。
- C. 取出当前关键帧（mpCurrentKeyFrame）的所有地图点存入 vpMapPointMatches 中。遍历 vpMapPointMatches，取出地图点 pMP，如果地图点不在当前帧的观测中，则为添加观测，更新平均观测方向和观测距离。否则，将 pMP 存入 mlpRecentAddedMapPoints 中。
- D. 更新当前关键帧（mpCurrentKeyFrame）的连接关系
- E. 将该关键帧插入地图中
- ③ 删除质量不好的地图点 MapPointCulling（）
- ④ 当前关键帧和相邻关键帧三角化产生新的地图点
CreateNewMapPoints（）
- ⑤ 检查并融合当前关键帧和相邻关键帧中重复的地图点

2.2 SearchInNeighbors ()

- A. 获取与当前关键帧共识程度最高的前 `nn` (单目 20, 其他 10) 帧, 存入 `vpNeighKFs` 中。
- B.
 - 遍历 `vpNeighKFs`, 取出其中的关键帧 `pKFi` (一级相连关键帧) 存入 `vpTargetKFs` 中。
 - 获取与一级相连关键帧 `pKFi` 相连的共识程度最高的前五帧二级相连关键帧存入 `vpSecondNeighKFs` 中。
 - 遍历 `vpSecondNeighKFs`, 取出其中的关键帧 `pKFi2`, 将其也存入 `vpTargetKFs` 中。
- C.
 - 获取当前关键帧的地图点存入 `vpMapPointMatches` 中,
 - 遍历 `vpTargetKFs`, 取出其中的关键帧 `pKFi`。将 `vpMapPointMatches` 匹配到关键帧中进行融合。Fuse (`pKFi`, `vpMapPointMatches`), 大致步骤可参考 1.4.1 SearchByProjection ()。
- D. 遍历 `vpTargetKFs`, 取出其中的关键帧 `pKFi`, 获取 `pKFi` 的地图点存入 `vpMapPointsKFi` 中, 遍历其中 (`vpMapPointsKFi`) 的地图点 `pMP`, 将其存入存储融合点的向量中 (`vpFuseCandidates`)。然后再次进行融合。
- E. 重新获取当前关键帧的地图点存入 `vpMapPointMatches` 中, 遍历 `vpMapPointMatches` 中的地图点 `pMP`, 找出其中 (`pMP`) 的最佳描述子, 更新 `pingju8n` 观测方向和观测距离。

2.2.1 ComputeDistinctiveDescriptors ()

- a. 获取该地图点所有有效的观测关键帧信息, 存入 `observations` 中
 - b. 遍历 `observations`, 取出其对应的关键帧 `pKF`, 同时将其对应的描述子向量存入 `vDescriptors` 中
 - c. 计算两两描述子之间的距离, 存入 `Distances` 中。
 - d. 求第 `i` 个描述子与其他所有描述子之间的距离, 并从小到大进行排序, 取出其中的中值 `median`。通过循环求出所有描述子之间的最小中值 `BestMedian` 和其对应的索引 `BestIdx`。将最小中值描述子对应的索引存入 `mDescriptor` 中
- F. 更新当前帧和其他帧的连接关系。
- ⑥ 局部 BA, `LocalBundleAdjustment (mpCurrentKeyFrame, mbAbortBA, mpMap)`, 优化了当前关键帧的位姿和 `LocalKeyFrames` 能观测到的所有地图点的位置
 - ⑦ 删除冗余关键帧, 90%的地图点能被其他关键帧观察到, 则判定为冗余关键帧 `KeyFrameCulling ()`。
 - ⑧ 将当前关键帧插入闭环检测队里 (`mpLoopCloser`)
 - ⑨ 结束线程

3. LoopClosing::Run()

3.1 DetectLoop()

- ① 提取关键帧队列中第一帧，赋值为当前帧（`mpCurrentKF`），然后从队列删除
- ② 检测距离上次闭环时间
- ③ 找出当前帧的连接关键帧，并计算出当前关键帧相似度得分最低的连接关键帧作为最低分阈值（`minScore`）
- ④ 在所有的关键帧中找出闭环候选帧组 `vpCandidateKFs`（闭环候选帧不与当前帧连接）

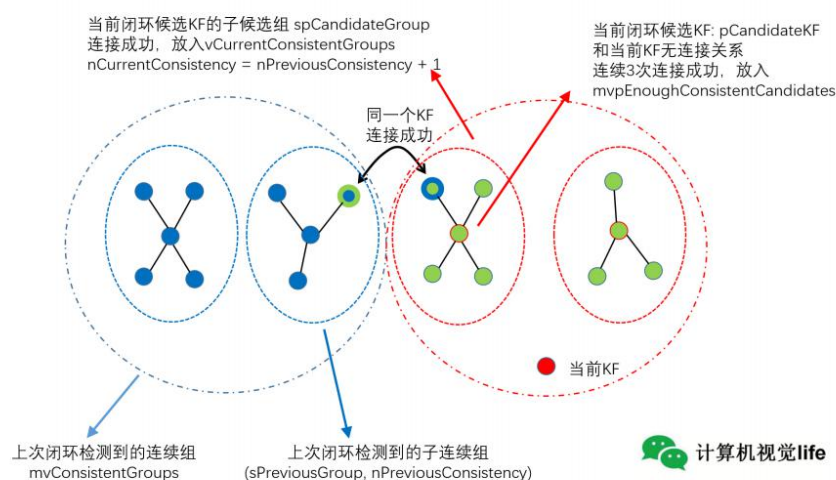
3.1.1 DetectLoopCandidates (`mpCurrentKF`, `minScore`)

取出与当前关键帧相连的所有关键帧，存储到 `spConnectedKeyFrames` 中

- A. 找出和当前关键帧具有公共单词的所有关键帧（存储到 `lkFsSharingWords`），不包括与当前帧相连的关键帧，并统计当前帧和闭环候选帧具有相同的 `words` 个数
- B. 统计闭环候选关键帧中与当前帧具有公共单词最多的单词数（`maxCommonWords`），让最小单词数（`minCommonWords`）等于 0.8 倍最大单词数
- C. 遍历上述的候选关键帧，挑选出共有单词数大于 `minCommonWords`，且匹配得分大于 `minScore` 的候选关键帧，并将它存入 `lScoreAndMatch`（符合条件的候选关键帧）
- D. 取出与 `lScoreAndMatch` 相连的 10 个关键帧，存入 `vpNeighbors` 中
计算最佳得分（`bestScore`）和累计得分（`accScore`）并将得分最高的帧赋给 `pBestKF`，再将 `accScore` 和 `pBestKF` 存入 `lAccScoreAndMatch`，将所有租的最佳累计得分保存在 `bestAccScore`，`minScoreToRetain = 0.75 * bestAccScore`
- E. 将候选帧组总得分大于 `minScoreToRetain` 的组中的最高得分的帧放入 `vpLoopCandidates` 中

3.1.2

对应函数 `DetectLoop()`



- A. 遍历 3.1.1 中得到的 `vpCandidateKFs`, 取出与其相连的关键帧构成一个子候选组

- (spCandidateGroup(包括自己))。
- B. 获取上一次回环检测的子连续组 (sPreviousGroup), 检测 sPreviousGroup, spCandidateGroup 是否有相同的关键帧。
- C. 如果有相同的关键帧, 则表示两个自候选组相连, 记录连续次数的 nCurrentConsistency + 1, 如果 nCurrentConsistency >= 3, 达到闭环条件, 就将此候选关键帧放入.mvpEnoughConsistentCandidates 中, 同时将此组的相应信息保存进上次闭环检测到的连续组 (mvConsistentGroups) 中

3.2 ComputeSim3()

- ① 遍历候选关键帧集, 初步筛选出与当前关键帧匹配点数大于 20 的候选关键帧, 并为每一个候选帧构建一个 Sim3Solver, 并将他们存储在 vpSim3Solver 向量组中, 记录保留下来的关键帧数量 (nCandidates)
- ② 对每个候选帧使用 Sim3Solver 迭代匹配, 如果计算出 Sim3 变换, 进行优化匹配取出 Sim3Solver 匹配后的内点集合, 保存在 vpMapPointMatches 中
求解出候选帧到当前帧的 R、t、s

3.2.1 SearchBySim3(mpCurrentKF, pkF, vpMapPointMatches, s, R, t, 7.5)

- A. 计算内参, sim3 的逆, 以及关键帧的地图点 (vpMapPoints1, vpMapPoints2)
- B. 记录已经匹配的地图点 (向量 vbAlreadyMatched1, vbAlreadyMatched2)
- C. 将 pkF1 的地图点投影到 pkF2 中的相机坐标系下坐标 p3Dc2, 然后再转换为像素坐标
- D. 预测投影点在金字塔的层数, 并计算搜索半径
- E. 获取该区域内的所有候选匹配特征点 (vIndices), 计算描述子距离, 找出最佳匹配点。保存在 vnMatch1
- F. 后续同上, 寻找 pkF2 中的特征点在 pkF1 中的最佳匹配 vnMatch2
- G. 一致性检查, 只有当 lpkF1 和 pkF2 中的两次投影的点是同一个点时才算匹配成功, 将最后匹配成功的点数保存在 nFound 中, 并返回

3.2.2 OptimizeSim3(mpCurrentKF, pkF, vpMapPointMatches, gScm, 10, mbFixScale)

- A. g2o 配置操作
- B. 设置顶点, 配置相机内参, 添加顶点
- C. 添加 pkF1 和 pkF2 的匹配点作为顶点 (但是不优化 (->setFixed(true)))
- D. 添加边 (正向投影, 从比黄候选关键帧到当前关键帧的投影)
同理, 添加边 (反向投影)
- E. 迭代优化
- F. 用卡方检验删除误差大的边
- G. 将优化后的结果赋值给 g2oS12 (其实就是 gScm), 返回匹配的内点个数 (nInliers)。

如果内点数 (nInliers) 大于等于 20, 将闭环候选关键帧存入 mpMathedKF 中, 将 vpMapPointMatches 存入.mvpCurrentMatchedPoints 中

- ③ 取出与候选关键帧相连的共视关键帧, 将他们一起存入 vpLoopConnectedKFs 中, 遍历 vpLoopConnectedKFs 中的所有帧的地图点, 将有效的地图点存入

- ④ mvpLoopMapPoints 中，将当前帧（mpCurrentKF）的 Id 赋值给 mnLoopPointForKF，将闭环候选关键帧及其连接关键帧的所有地图点投影到当前关键帧中，进行投影匹配。

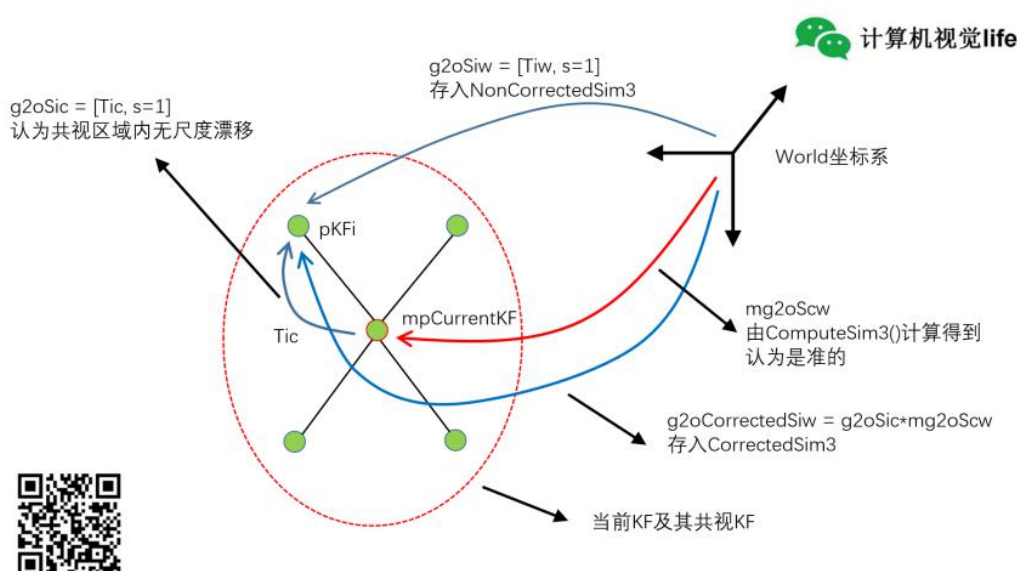
3.2.3 SearchByProjection

（mpCurrentKF, mScw, mvpLoopMapPoints, mvpCurrentMatchedPoints, 10）

投影匹配，参考 3.2.1 SearchBySim3，返回新的匹配成功的点的个数。并将新的匹配成功的添加进 mvpCurrentMatchedPoints 中。

- ⑤ 计算匹配成功的地图点个数（nTotalMatches），如果 nTotalMatches 大于等于 40，保留当前闭环候选帧，删掉其他的闭环候选关键帧，返回 true。否则，闭环不可靠，删除所有闭环候选帧，返回 false。

3.3 CorrectLoop()



- ① 更新当前帧的连接关系

- ②

- A. 选出当前关键帧的连接关键帧，并将他们一起存入 mvpCurrentConnectedKFs 中

遍历 mvpCurrentConnectedKFs，获取当前帧（mpCurrentKF）到连接帧（pKFi）的位姿变换（Tic, Ric, tic）。

获取当前帧到其共识关键帧的 Sim3 变换，g2oCorrectedSiw（认为是准确的），并将 g2oCorrectedSiw 存入 CorrectedSim3 中。g2oSiw（认为是不准确的），并将 g2oSiw 存入 NonCorrectedSim3 中。

- B. 遍历 CorrectedSim3，取出当前关键帧的连接关键帧（pKFi）以及其 Sim3 变换（g2oCorrectedSiw）。同时取出 pKFi 对应 NonCorrectedSim3 的 Sim3 变换（g2oSiw）。

获取 pKFi 的对应地图点（vpMPsi），遍历 vpMPsi，获取其世界坐标（P3Dw）、并将其转换为 Eigen 形式（eigP3Dw）。

对于 eigP3Dw, 首先通过 g2oSiw 获取其在 pKFi 下的坐标。然后再通过 g2oCorrectedSiw 再次获取其在水系坐标系下的坐标 (eigCorrectedP3Dw)。记录矫正该地图点的关键帧 (mpCurrentKF) Id, 以及当前地图点所在的连接关键帧 (pKFi) 的 Id。更新平均观测。

C. 获取 g2oCorrectedSiw 的 R, t, 求出变换矩阵 correctedTiw, 更新 pKFi 矫正后的位姿, 更新连接关系

③ 获取.mvpCurrentMatchedPoints (投影匹配成功) 中的地图点 pLoopMP, 以及在同样的索引下当前关键帧 mpCurrentKF 对应的地图点 pCurMP。如果 pCurMP 存在, 就用 pLoopMP 将其替换, 不存在的话, 就将 pLoopMP 添加进去

④

3.3.1 SearchAndFuse (CorrectedSim3)

A. 遍历 CorrectedPosesMap (即 CorrectedSim3), 取出关键帧 pKF (矫正后的与当前关键帧相连的关键帧), sim3 变换 g2oScw、cvScw。

B.

Fuse (pKF, cvScw,.mvpLoopMapPoints,4, vpReplacePoints)

a. 取出 Scw(即 cvScw)的旋转矩阵, 尺度, 计算得到 Rcw, tcw, Ow。

取出 pKF 的地图点, 存入 spAlreadyFound 中, 获取已经闭环匹配上的地图点 (.mvpLoopMapPoints) 数量 nPoints,

b. 遍历所有地图点, 获取地图点在相机坐标系下的坐标 p3Dc。

c. 获取地图点 (闭环候选关键帧的地图点) 投影在 pKF (当前关键帧的相连关键帧) 上的像素坐标

d. 投影匹配操作同 3.2.1 SearchBySim3

e. 替换或新增地图点, 记录需要替换的地图点存入 vpReplacePoints, 返回替换或新增的地图点数目 nFused。

C. 替换需要替换的地图点。(替换是用.mvpLoopMapPoints 中的地图点替换 pKF 中的地图点)

⑤

A. 遍历当前关键帧相连的关键帧组 (.mvpCurrentConnectedKFs), 得到与当前关键帧相连关键帧的相连关键帧 (当前关键帧的二级相连关键帧), 存入 vpPreviousNeighbors。

B. 更新一级相连关键帧 (pKFi) 的连接关系, 将连接关系存入 LoopConnections 中。

C. 首先, 将 LoopConnections 中的二级连接关系 vpPreviousNeighbors 删除。然后, 再将 LoopConnections 中的原来的当前关键帧连接关系 (.mvpCurrentConnectedKFs) 删除掉。

⑥

3.3.2 OptimizeEssentialGraph()

OptimizeEssentialGraph(mpMap ,mpMatchedKF, mpCurrentKF , NonCorrectedSim3, CorrectedSim3 , LoopConnections, mbFixScale)

- A. 构造优化器
- B. 添加顶点，优先使用调整后的 Sim3，否则使用追踪时的位姿。对闭环候选关键帧不进行优化
- C. 添加第一种边，闭环时因为地图点调整而出现的的关键帧间的新连接关系
- D. 添加第二种边，跟踪时形成的边，即生成树的边（有父关键帧）
添加第三种边，闭环匹配形成的边，当前帧和闭环匹配帧之间的连接关系
- 添加第四种边，共视程度超过 100 的关键帧也作为边进行优化
- E. 开始 g2o 优化
- F. 将优化后的位姿更新进关键帧中
- G. E, F 得到优化后的位姿后，地图点根据参考帧优化前后的相对关系调整自己的位置。

⑦ 添加当前帧和闭环匹配帧之间的边

⑧ 进行全局 BA

3.3.3 RunGlobalBundledAdjustment (nloopKF)

A.

`GlobalBundledAdjustment (mpMap , 10 , mbStopGBA , nloopKF , false)`

`BundledAdjustment(vpKFs , vpMP , nIterations , pbStopFlag , nLoopKF , bRobust)`

- a. 初始化 g2o 优化器
- b. 向优化器中添加顶点
添加关键帧位姿作为优化器顶点
添加地图点作为顶点（当顶点为三维点时，需要进行边缘化，即：
`setMarginalized (true)`）
- c. 向优化器中添加投影边。。。。。
- d. 获取优化后的结果
- B. 遍历并更新全局地图中的所有 `spanning tree` 中的关键帧
- C. 遍历每一个地图点，并用更新的关键帧位姿来更新地图点位置