深蓝学院
shenlanxueyuan.com

ORB-SLAM2源码解析

主讲人 吴博 许可 高成强

东北大学

机器人算法工程师

预备知识

视觉VO与重定位

局部优化

全局闭环

代码串讲

总结

预备知识

视觉VO与重定位

局部优化

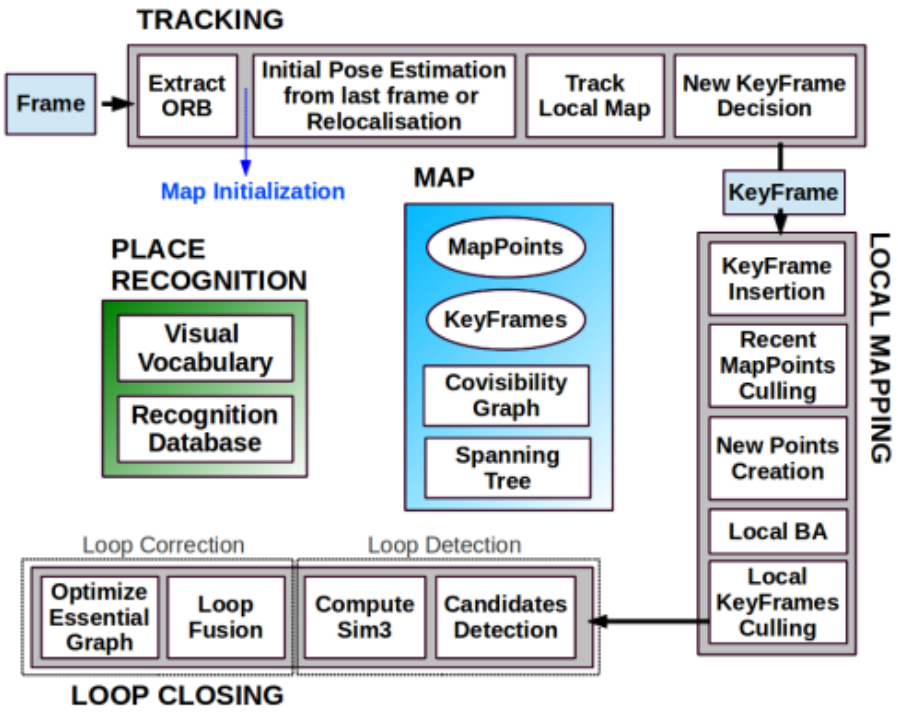
全局闭环

代码串讲

总结

整体软件结构

主要线程：



主要逻辑文件：

System.cpp --> class System

Initializer.cpp --> class Initializer

Tracking.cpp --> class Tracking

LocalMapping.cpp --> class LocalMapping

LoopClosing.cpp --> class LocalClosing

Viewer.cpp --> class Viewer

主要类：

class Frame

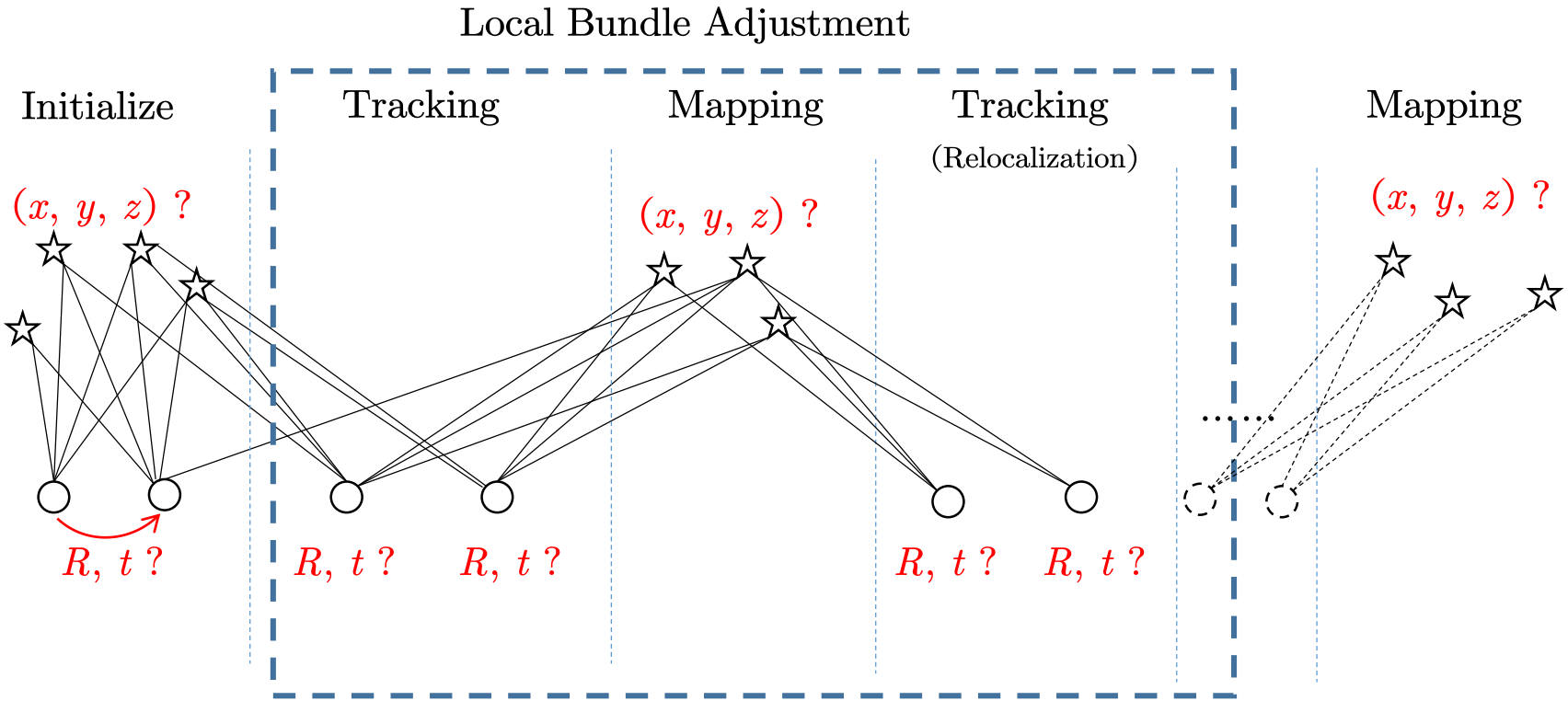
class KeyFrame

class KeyFrameDatabase

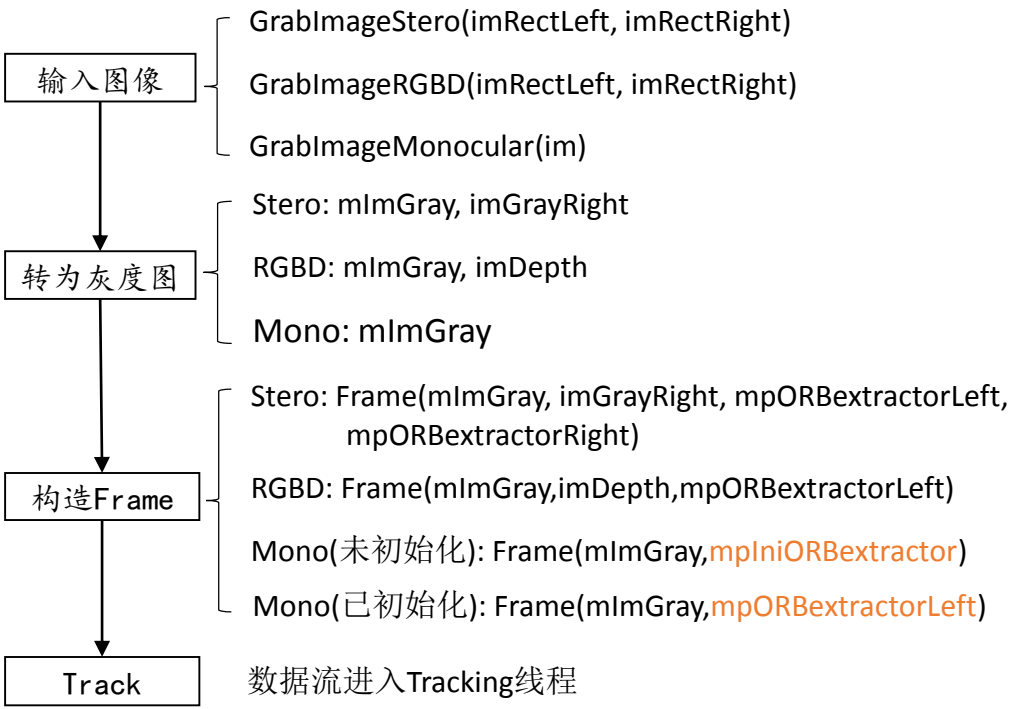
class Map

class Optimizer

最简单的VO流程


Local Bundle Adjustment
Initialize
Tracking
Mapping
Tracking
(Relocalization)
Mapping
$(x,\ y,\ z)$?
$(x,\ y,\ z)$?
$(x,\ y,\ z)$?
$R,\ t$?
$R,\ t$?
$R,\ t$?
$R,\ t$?
$R,\ t$?

视觉跟踪与建图　　主线程System入口



双目构造Frame需要左右相机找到匹配特征点：

step1：为左目每个特征点建立一个带状区域搜索表，限定搜索区域。（已提前极线矫正）

step2：通过描述子进行特征点匹配，得到每个特征最佳匹配点scaledR0

step3：通过SAD滑窗得到匹配修正量bestincR

step4：(bestincR, dist)(bestincR－1, dist)(bestR＋1, dist)三个点拟合抛物线，得到亚像素修正量deltaR

step5：最终匹配点位置为：scaleduR0＋bestincR＋deltaR

注：mpIniORBextractor相比mpORBextractorLeft提取的特征点多一倍

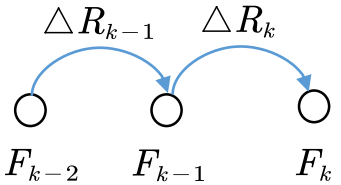
视觉跟踪与建图　　Tracking线程

Frame

- 初始化
  - SteroInitialization() 见之后ppt
  - MonocularInitialization()见之后ppt
- 相机位姿跟踪
  - mbOnlyTracking(false)
    - 位姿跟踪
      - TrackWithMotionModel()
      - TrackReferenceKeyFrame()
    - Relocalization() 见之后ppt
  - mbOnlyTracking(true)：同时跟踪与定位，不插入关键帧，局部地图不工作
- 局部地图跟踪
  - UpdateLocalMap()
    - UpdateLocalKeyFrames() 见之后ppt
    - UpdateLocalPoints() 见之后ppt
  - SearchLocalPoints()：获得局部地图与当前帧的匹配，见之后ppt
  - PoseOptimization()：最小化投影误差优化位姿
- 是否生成关键帧
  - NeedNewKeyFrame
    - 很长时间没有插入关键帧、局部地图空闲、跟踪快要跪
    - 跟踪地图MapPoints的比例比较少
- 生成关键帧
  - CreateNewKeyFrame
    - KeyFrame(mCurrentFrame,mpMap,mpKeyFrameDB)
    - 对于双目或RGBD摄像头构造一些MapPoints, 为MapPoints添加属性

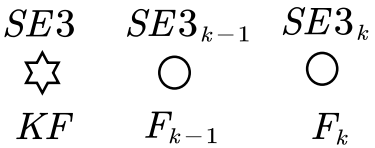
注： mbOnlyTracking默认为false，用户可通过运行界面选择仅跟踪定位模式

TrackWithMotionModel



恒速模型：假设 $\triangle R_k \approx \triangle R_{k-1}$

TrackReferenceKeyframe



跟踪参考帧模型： $SE3_k \approx SE3_{KF}$

对于双目或RGBD，可直接恢复一些3D点。
因此单帧即可创建Mappoints，KeyFrame等等

对于单目，通过时序上两帧观测恢复H和F模型、分解、以及三角化恢复3D点，并通过得分的方式选择选择一个更好的模型结果进行初始化

Fundamental模型评分

$$\text{scoreF} = \sum_{i=0}^{N} \rho(T_F - ||x' F x||^2/\sigma^2)$$

$$\rho(x) = \begin{cases} 0 & x \leq 0 \\ x & \text{else} \end{cases} \quad T_F = 3.84$$

Homography模型评分

$$\text{scoreH} = \sum_{i=0}^{N} \rho(T_H - ||x' - Hx||^2/\sigma^2)$$

$$\rho(x) = \begin{cases} 0 & x \leq 0 \\ x & \text{else} \end{cases} \quad T_H = 5.99$$

Fundamental模型与Homography模型选择

$$R_H = \frac{s_H}{s_H + s_F}, \; R_H \geq 0.45$$

UpdateLocalKeyFrames

能观测到当前帧mCurrentFrame的Mappoints的关键帧--->keyframeCounter

除去无效的关键帧---> mvpLocalKeyFrames

- mvpLocalKeyFrames中与其它（共视）连接最多的关键帧作为参考帧mpReferenceKF

共视关键帧也添加进来

- mvpLocalKeyFrames中每一帧共视比较好的10帧也添加进来
- 自己的子关键帧
- 自己的父关键帧

UpdateLocalPoints

收集mvpLocalKeyFrames中每帧关键帧的Mappoints --->mvLocalMapPoints

SearchLocalPoints

标记当前帧的Mappoints，这些Mappoints不同搜索匹配

遍历mvpLocalMappoints判断是否在mCurrentFrame视野内

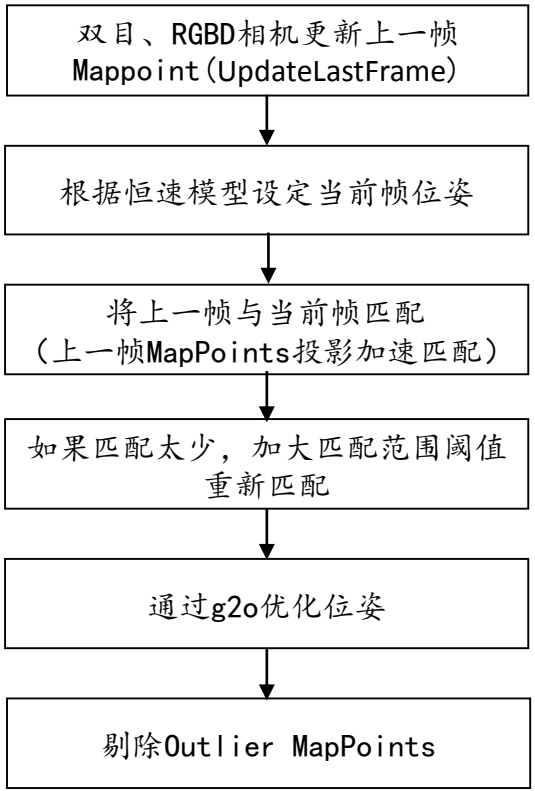
- 如果在视野内，则该mappoint被观测的统计值加1

当前帧mCurrentFrame通过投影匹配mvpLocalMappoints

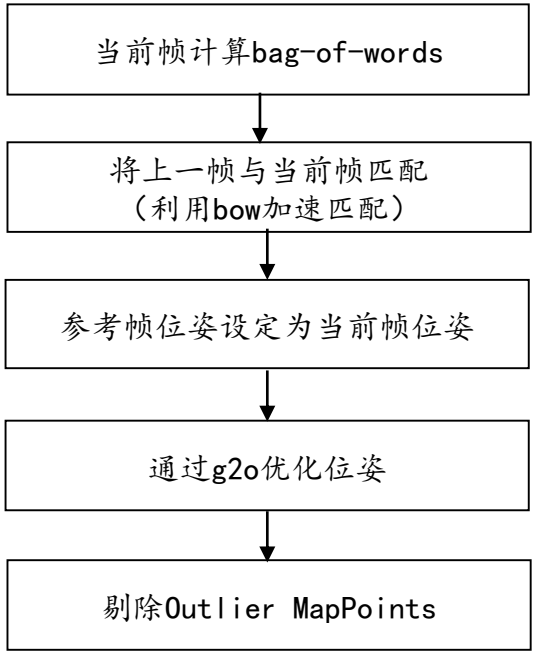
- 调用SearchByProject

Tracking线程

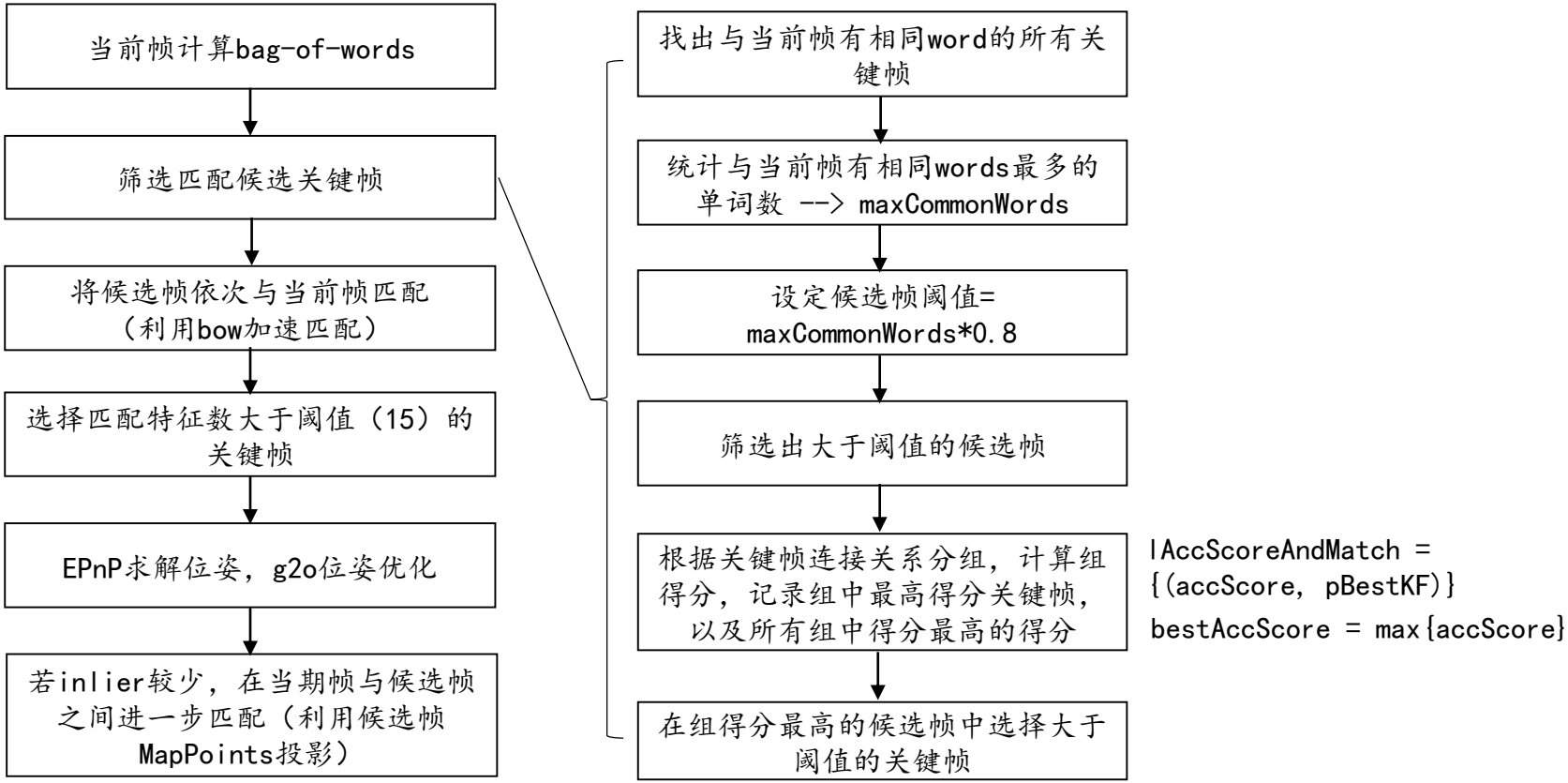
TrackWithMotionModel



TrackWithReferenceKeyframe



Tracking线程 Relocalization



预备知识

视觉VO与重定位

局部优化

全局闭环

代码串讲

总结

局部优化


将LocalMapping线程设繁忙
SetAcceptKeyFrames 线程状态同步
检查是否有新的关键帧
CheckNewKeyFrames 注意多线程内存访问
处理新的关键帧
ProcessNewKeyFrame-->UpdateConnections
剔除质量不佳的Mappoints
MapPointCulling
产生新的Mappoints
CreateNewMapPoints
当前帧与邻域帧Mappoints融合
SearchInNeighbors
局部BundleAdjustment优化
LocalBundleAdjustment
剔除冗余关键帧
KeyFrameCulling
关键帧加入闭环检测队列
InsertKeyFrame
将LocalMapping线程设为空闲
SetAcceptKeyFrames

局部优化

UpdateConnections

1. 取出当前关键帧mpCurrentKeyFrame所有的MapPoints--->mvpMapPoints
2. 取出能观测到这些MapPoints的关键帧--->observations
3. 统计每个关键帧观测到这些MapPoints的个数--->Kfcounter<KeyFrame*, int>
4. 建立共视有向连接图

- 如果这些关键帧中观测到MapPoints的个数大于阈值，则与当前关键帧建立连接 ---> vPairs
- 得到共视关系最好的关键帧 --->pKFmax
- 如果这些关键帧中观测到MapPoints的个数均小于阈值，则将pKFmax与当前帧建立连接---> vPairs
- 对vPairs按照共视的的程度进行排序
- vParis中共视关系最强的关键帧作为mpCurrentKeyFrame的父关键帧，同时mpCurrentKeyFrame作为该帧的子关键帧

局部优化

MapPointCulling

作用：剔除mlpRecentAddedMapPoints中不好的MapPoints

mlpRecentAddedMapPoints来自于：

- 对于双目会在创建关键帧时新建一些MapPoints
- 通过CreateNewMapPoints函数，关键帧之间三角化恢复一些MapPoints

剔除MapPoint的标准为（观测一致性，从被创建开始，未被连续可靠观测到）：

- mnFound / mnVisible < 25%
- 从MapPoint被创建开始，到现在已经过了不小于2帧关键帧，但是该MaPoints被观测到的关键帧数小于阈值（单目为2，双目为3）
- 从MapPoint被创建开始，到现在已经超过帧关键帧

局部优化

CreateNewMapPoints

1. 取出与当前关键帧mpCurrentKeyFrame共视关系比较好的关键帧 ---> vpNeighKFs，（单目20，双目10）
2. 对vpNeighKFs中每一个关键帧pKF2，判断视差角是否足够

- 对于双目，相机光心距离不小于双目基线
- 对于单目，相机光心距离 / pKF2中MapPoints中值深度 >= 0.01

3. 计算mpCurrentKeyFrame与pKF2之间Fundamental矩阵 $F_{12} = K_1^{-T}[t]_\times R_{12} K_2^{-1}$
4. 通过SearchForTriangulation函数查找特征匹配（借助F矩阵剔除Outlier）---> vMatchedIndices
5. 对匹配特征点DLT法三角化恢复3D点（对于双目，关键帧视差角大时用三角化法恢复3D点，反之则直接反投影）检查3D点有效性：a. 恢复3D点 b. 检测3D点是否在相机前方 c. 重投影误差。d. 3D点到俩相机距离之比和特征点金字塔层数之比接近
6. 将3D点构造为MapPoint并mlpRecentAddedMapPoints进行一致性观测检查

局部优化

SearchInNeighbors

1. 取出与当前关键帧一级共视的关键帧与二级共视（共视关键帧的共视关键帧）的关键帧
2. 将当前帧的MapPoints分别与一级二级相邻帧(的MapPoints)进行融合。matcher.Fuse函数

- 投影当前帧的MapPoints到相邻关键帧pKFi中，并判断是否有重复的MapPoints
- 如果MapPoint能匹配关键帧的特征点，并且该点有对应的MapPoint，那么将两个MapPoint合并
- 如果MapPoint能匹配关键帧的特征点，并且该点没有对应的MapPoint，那么为该点添加MapPoint

3. 将一级二级相邻帧的MapPoints分别与当前帧（的MapPoints）进行融合
4. 更新与其它帧的连接关系

局部优化

KeyFrameCulling： 从当前共视关键帧中剔除冗余关键帧

1. 取出与当前关键帧共视的所有关键帧vpLocalKeyFrames
2. 对于vpLocalKeyFrames中每个关键帧判断是否是冗余关键帧

- 取出该关键帧对应的MapPoints---> vpMapPoints
- 如果vpMapPoints中有90%的MapPoints可被至少3帧尺度近似的关键帧观测到，则为冗余关键帧

预备知识

视觉VO与重定位

局部优化

全局闭环

代码串讲

总结

全局闭环优化

mlploopKeyFrameQueue

队列中取一帧

mpcurrentKF

判断距离上一次闭环检测是否超过10帧

计算当前帧与相连关键帧的Bow最低得分

mpcurrentKF
minscore

检测得到闭环候选帧

vpLoopCandidate
s

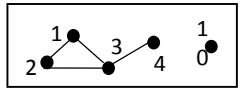
检测候选帧连续性

1、三个阈值都是计算获得，鲁邦性好

minscore mincommons minscoreToRetain

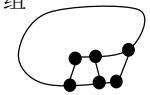
2、通过分组可以将单独得分很高的无匹配关键帧剔除

分组示意图：

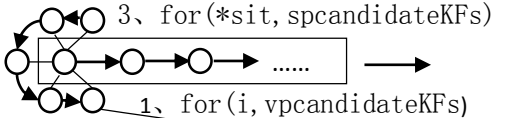


如图：1、2、3、4、10都是闭环候选帧。
节点1：与2、3相连，1与2、3分为一组
节点2：与1、3相连，2与1、3分为一组
节点3：与1、2、4相连，3与1、2、4分为一组
节点4：与3相连，4与3分为一组
节点10：10自己单独一组

连续性检测示意图：

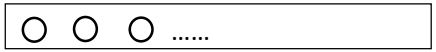


3、for(*sit,spcandidateKFs)

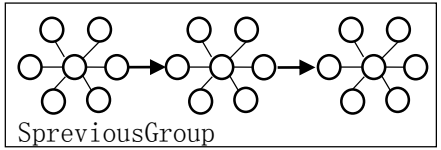


1、for(i,vpcandidateKFs)

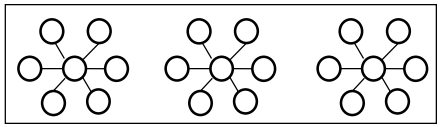
Result:



mvpEnoughConsistentcandidates



2、for(iG,mvConsistentGroup)



vcurrentconsistentGroup

(pKF,minscore)

找出与当前帧有公共单词的关键帧，但不包括与当前帧相连的关键帧

lKFsharingwords

统计候选帧中与pKF具有共同单词最多的单词数

maxcommonwords

得到阈值
mincommons=0.8*maxcommonwords

maxcommonwords
mincommons
minscore

筛选共有单词大于mincommons且Bow得分大于minscore的关键帧

lscoreAndMatch

将存在相连的分为一组，计算组最高得分bestAccScore，同时得到每组中得分最高的关键帧

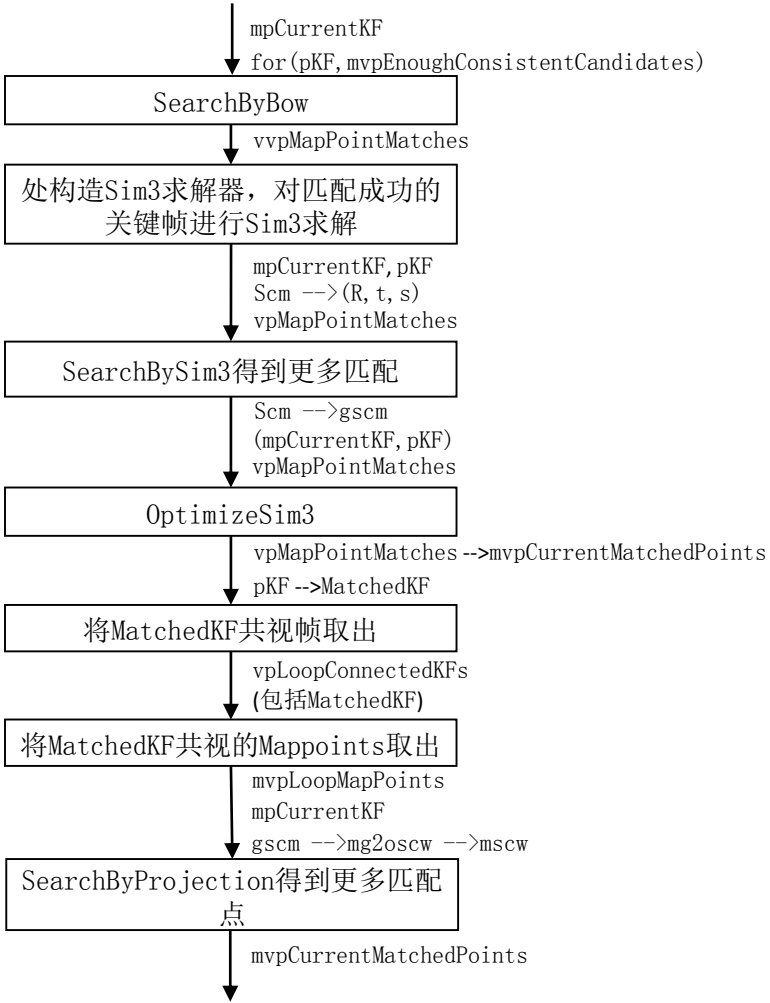
lsAccScoreAndMatch
bestAccScore

得到阈值minScoreToRetain
=0.75*bestAccScore

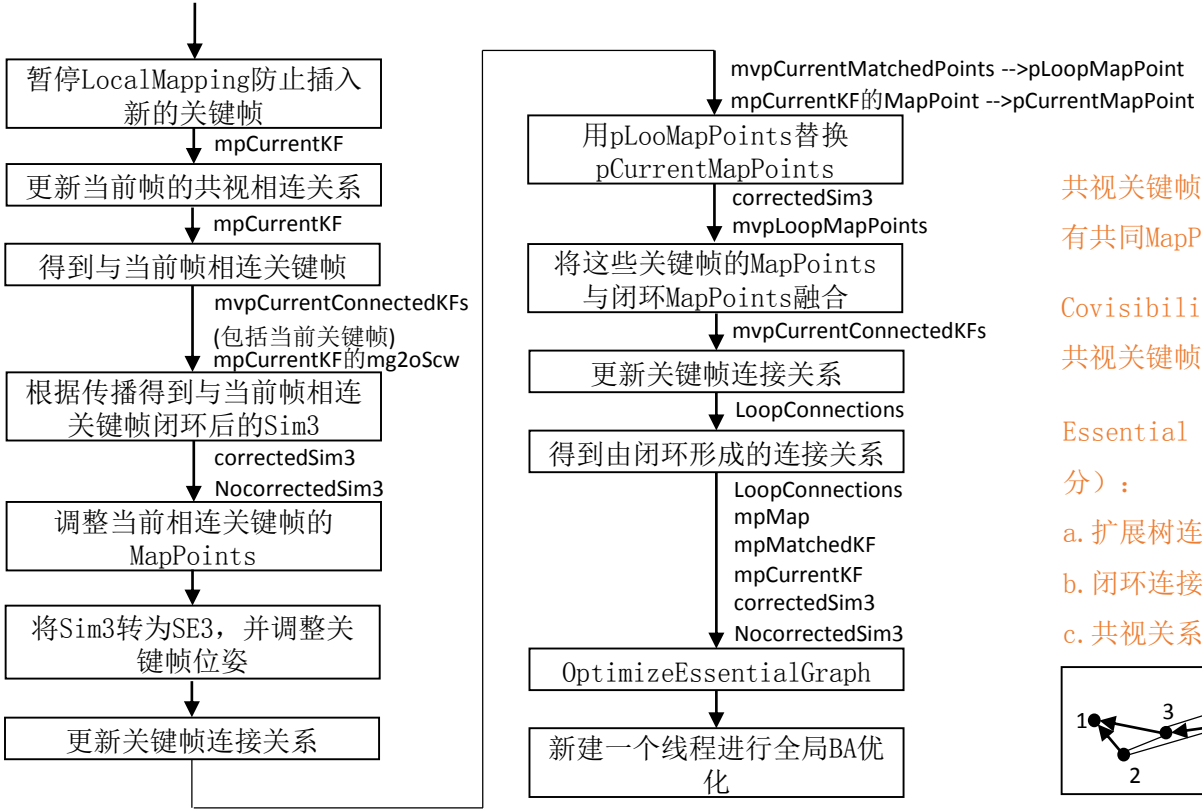
lsAccScoreAndMatch
minScoreToRetain

vpLoopCandidates

全局闭环优化


mpCurrentKF
for(pKF,mvpEnoughConsistentCandidates)
SearchByBow
vvpMapPointMatches
处构造Sim3求解器，对匹配成功的关键帧进行Sim3求解
mpCurrentKF,pKF
Scm -->(R,t,s)
vpMapPointMatches
SearchBySim3得到更多匹配
Scm -->gscm
(mpCurrentKF,pKF)
vpMapPointMatches
OptimizeSim3
vpMapPointMatches-->mvpCurrentMatchedPoints
pKF-->MatchedKF
将MatchedKF共视帧取出
vpLoopConnectedKFs
(包括MatchedKF)
将MatchedKF共视的Mappoints取出
mvpLoopMapPoints
mpCurrentKF
gscm -->mg2oscw -->mscw
SearchByProjection得到更多匹配点
mvpCurrentMatchedPoints

全局闭环优化



共视关键帧：

有共同MapPoints观测的关键帧

Covisibility Graph：

共视关键帧及它们的连接关系构成的图

Essential Graph（包含3部分）：

a. 扩展树连接关系

b. 闭环连接关系

c. 共视关系非常好的连接关系

g2o优化

GlobalBundleAdjustemnt与LocalBundleAdjustment：同时优化3D点和位姿

3D-2D 最小化重投影误差 e = (u,v) - project(Tcw*Pw)

Vertex: g2o::VertexSE3Expmap()，即当前帧的Tcw

g2o::VertexSBAPointXYZ()，MapPoint的mWorldPos

Edge: g2o::EdgeSE3ProjectXYZ()，BaseBinaryEdge

Vertex：待优化当前帧的Tcw

Vertex：待优化MapPoint的mWorldPos

measurement：MapPoint在当前帧中的二维位置(u,v)

InfoMatrix: invSigma2(与特征点所在的尺度有关)

g2o优化

PoseOptimization：只优化位姿

3D-2D 最小化重投影误差 e = (u,v) - project(Tcw*Pw)

只优化Frame的Tcw，不优化MapPoints的坐标

Vertex: g2o::VertexSE3Expmap()，即当前帧的Tcw

Edge: g2o::EdgeSE3ProjectXYZOnlyPose()，BaseUnaryEdge

- Vertex：待优化当前帧的Tcw
- measurement：MapPoint在当前帧中的二维位置(u,v)
- InfoMatrix: invSigma2(与特征点所在的尺度有关)

Edge: g2o::EdgeStereoSE3ProjectXYZOnlyPose()，BaseUnaryEdge

- Vertex：待优化当前帧的Tcw
- measurement：MapPoint在当前帧中的二维位置(ul,v,ur)
- InfoMatrix: invSigma2(与特征点所在的尺度有关)

g2o优化

OptimizeEssentialGraph在成功进行闭环检测后，全局BA优化前进行

Vertex: g2o::VertexSim3Expmap，Essential graph中关键帧的位姿

Edge: g2o::EdgeSim3()，BaseBinaryEdge

Vertex：关键帧的Tcw，MapPoint的Pw

measurement：经过CorrectLoop函数步骤2，Sim3传播校正后的位姿

InfoMatrix: 单位矩阵

g2o优化

OptimizeSim3在筛选闭环候选帧时用于位姿Sim3优化

Vertex: g2o::VertexSim3Expmap()，两个关键帧的位姿

g2o::VertexSBAPointXYZ()，两个关键帧共有的MapPoints

Edge: g2o::EdgeSim3ProjectXYZ()，BaseBinaryEdge

Vertex：关键帧的Sim3，MapPoint的Pw

measurement：MapPoint在关键帧中的二维位置(u,v)

InfoMatrix: invSigma2(与特征点所在的尺度有关)

Edge: g2o::EdgeInverseSim3ProjectXYZ()，BaseBinaryEdge

Vertex：关键帧的Sim3，MapPoint的Pw

measurement：MapPoint在关键帧中的二维位置(u,v)

InfoMatrix: invSigma2(与特征点所在的尺度有关)

谢谢大家！