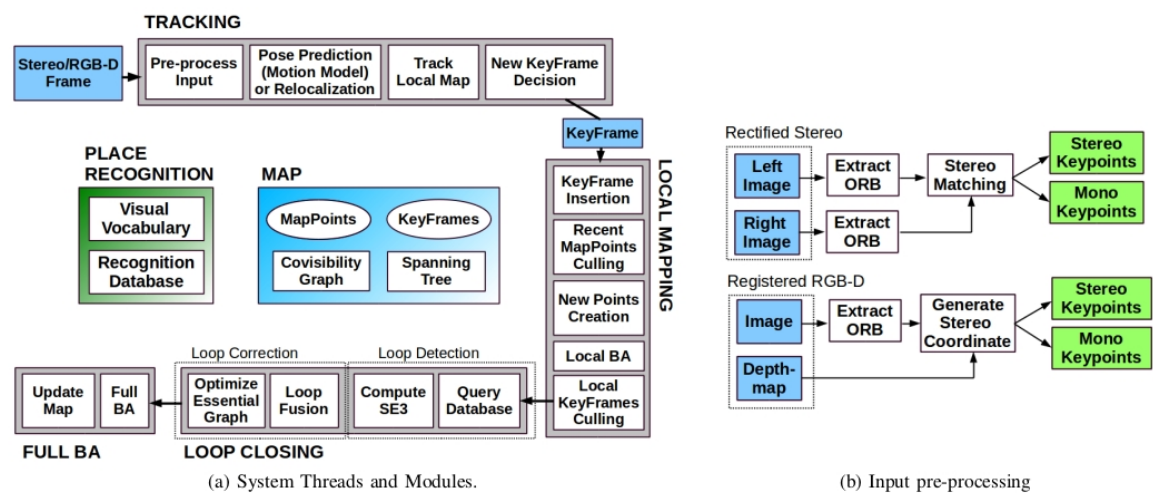


ORB-SLAM2学习笔记

ORB-SLAM2学习笔记

- 1. 系统框图
- 2. 数据结构
 - 2.1 MapPoint
 - 2.2 Frame
 - 2.3 KeyFrame
 - 2.4 Map
 - 2.5 数据结构维护
- 3. 代码流程
 - 3.1 主线程
 - 3.2 Tracking线程
 - 3.3 LocalMapping线程

1. 系统框图



2. 数据结构

2.1 MapPoint

属性	成员变量	类型	描述
公有	<code>mnId</code>	<code>long unsigned int</code>	地图点id
公有	<code>mnFirstKFid</code>	<code>long int</code>	第一个关键帧的id
公有	<code>mTrackProjX</code>	<code>float</code>	用于匹配

属性	成员变量	类型	描述
公有	<code>mTrackProjY</code>	<code>float</code>	用于匹配
公有	<code>mbTrackInView</code>	<code>bool</code>	用于匹配
公有	<code>mnTrackScaleLevel</code>	<code>int</code>	用于匹配
公有	<code>mTrackViewCos</code>	<code>float</code>	用于匹配
公有	<code>mnTrackReferenceForFrame</code>	<code>long unsigned int</code>	用于更新局部地图点
公有	<code>mnLastFrameSeen</code>	<code>long unsigned int</code>	用于检查可视
公有	<code>mnBALocalForKF</code>	<code>long unsigned int</code>	用于局部BA
公有	<code>mnFuseCandidateForKF</code>	<code>long unsigned int</code>	用于融合地图点
私有	<code>mWorldPos</code>	<code>cv::Mat</code>	地图点世界坐标
私有	<code>mpMap</code>	<code>Map*</code>	地图
私有	<code>mpRefKF</code>	<code>KeyFrame*</code>	参考关键帧
私有	<code>mObservations</code>	<code>map<KeyFrame*, size_t></code>	观测映射
私有	<code>nObs</code>	<code>int</code>	观测数量
私有	<code>mbBad</code>	<code>bool</code>	地图点bad标记, LocalMapping线程删除地图点和融合地图点时被设置
私有	<code>mpReplaced</code>	<code>MapPoint*</code>	用来替换的地图点
私有	<code>mnVisible</code>	<code>int</code>	可视计数
私有	<code>mnFound</code>	<code>int</code>	发现（被ORB关键点检测器检测到）计数
私有	<code>mNormalVector</code>	<code>cv::Mat</code>	平均观测方向向量

属性	成员变量	类型	描述
私有	<code>mDescriptor</code>	<code>cv::Mat</code>	描述子
私有	<code>mfMinDistance</code>	<code>float</code>	最小尺度不变距离
私有	<code>mfMaxDistance</code>	<code>float</code>	最大尺度不变距离

属性	成员函数	返回值	描述
公有	<code>MapPoint(const cv::Mat&, KeyFrame*, Map*)</code>		利用关键帧构造地图点
公有	<code>MapPoint(const cv::Mat&, Map*, Frame*, const int&)</code>		利用普通帧构造地图点
公有	<code>SetWorldPos(const cv::Mat&)</code>	<code>void</code>	设置地图点世界坐标
公有	<code>GetWorldPos()</code>	<code>cv::Mat</code>	获取地图点世界坐标
公有	<code>GetReferenceKeyFrame()</code>	<code>KeyFrame*</code>	获取参考关键帧
公有	<code>AddObservation(KeyFrame*, size_t)</code>	<code>void</code>	添加观测
公有	<code>EraseObservation(KeyFrame*)</code>	<code>void</code>	删除观测
公有	<code>GetObservations()</code>	<code>map<KeyFrame*, size_t></code>	获取所有观测
公有	<code>Observations()</code>	<code>int</code>	获取观测数量
公有	<code>GetIndexInKeyFrame(KeyFrame*)</code>	<code>int</code>	获取地图点某关键帧中的索引
公有	<code>IsInKeyFrame(KeyFrame*)</code>	<code>bool</code>	判断地图在某关键帧中是否被检测到
公有	<code>SetBadFlag()</code>	<code>void</code>	设置bad标记
公有	<code>isBad()</code>	<code>bool</code>	判断地图点是否为bad
公有	<code>Replace(MapPoint*)</code>	<code>void</code>	替换地图点

属性	成员函数	返回值	描述
公有	<code>GetReplaced()</code>	<code>MapPoint*</code>	获取用来替换的地图点
公有	<code>IncreaseVisible(int)</code>	<code>void</code>	增加可视计数
公有	<code>IncreaseFound(int)</code>	<code>void</code>	增加发现计数
公有	<code>GetFoundRatio()</code>	<code>float</code>	获取“发现/可视”计数比例
公有	<code>ComputeDistinctiveDescriptors()</code>	<code>cv::Mat</code>	获取描述子
公有	<code>GetDescriptor()</code>	<code>void</code>	计算平均观测方向向量, 最小和最大不变性距离
公有	<code>UpdateNormalAndDepth()</code>	<code>void</code>	计算平均观测方向向量, 最小和最大不变性距离
公有	<code>GetNormal()</code>	<code>cv::Mat</code>	获取平均观测方向向量
公有	<code>GetMinDistanceInvariance()</code>	<code>float</code>	获取最小不变性距离
公有	<code>GetMaxDistanceInvariance()</code>	<code>float</code>	获取最大不变性距离
公有	<code>PredictScale(const float&, KeyFrame*)</code>	<code>int</code>	预测距离对应的尺度
公有	<code>PredictScale(const float&, Frame*)</code>	<code>int</code>	预测距离对应的尺度

2.2 Frame

属性	成员变量	类型	描述
公有	<code>mnId</code>	<code>long unsigned int</code>	帧id
公有	<code>mTimeStamp</code>	<code>double</code>	帧时间戳
公有	<code>mpReferenceKF</code>	<code>KeyFrame*</code>	参考关键帧
公有	<code>mK</code>	<code>cv::Mat</code>	相机内参矩阵

属性	成员变量	类型	描述
公有	<code>mDistCoef</code>	<code>cv::Mat</code>	相机畸变系数
公有	<code>mTcw</code>	<code>cv::Mat</code>	相机位姿
公有	<code>mpORBExtractor</code>	<code>ORBExtractor*</code>	ORB特征提取器
公有	<code>mvKeys</code>	<code>vector<cv::KeyPoint></code>	原始关键点
公有	<code>mvKeysUn</code>	<code>vector<cv::KeyPoint></code>	去畸变的关键点
公有	<code>mDescriptors</code>	<code>cv::Mat</code>	描述子
公有	<code>mvpMapPoints</code>	<code>vector<MapPoint*></code>	地图点数组
公有	<code>mvbOutlier</code>	<code>vector<bool></code>	外点标记数组
公有	<code>N</code>	<code>int</code>	关键点数量
公有	<code>mpORBvocabulary</code>	<code>ORBVocabulary*</code>	ORB词袋
公有	<code>mBowVec</code>	<code>DBow2::BowVector</code>	词袋向量
公有	<code>mFeatVec</code>	<code>DBow2::FeatureVector</code>	特征向量
公有	<code>mGrid[FRAME_GRID_COLS] [FRAME_GRID_ROWS]</code>	<code>vector<size_t></code>	关键点在网格分布的索引数组
公有	<code>mnScaleLevels</code>	<code>int</code>	尺度数量
公有	<code>mfScaleFactor</code>	<code>float</code>	尺度倍数
公有	<code>mfLogScaleFactor</code>	<code>float</code>	对数尺度倍数
公有	<code>mvScaleFactors</code>	<code>vector<float></code>	尺度数组
公有	<code>mvInvScaleFactors</code>	<code>vector<float></code>	逆尺度数组

属性	成员变量	类型	描述
公有	<code>mvLevelSigma2</code>	<code>vector<float></code>	方差数组
公有	<code>mvInvLevelSigma2</code>	<code>vector<float></code>	逆方差数组

属性	成员函数	返回值	描述
公有	<code>Frame()</code>		构造函数
公有	<code>Frame(const Frame&)</code>		拷贝构造函数
公有	<code>Frame(const cv::Mat&, const double&, ORBExtractor*, ORBVocabulary*, cv::Mat&, cv::Mat&)</code>		单目图像帧构造函数
公有	<code>ComputeBoW()</code>	<code>void</code>	计算BoW特征
公有	<code>SetPose(const cv::Mat&)</code>	<code>void</code>	设置相机位姿
公有	<code>GetRotation()</code>	<code>cv::Mat</code>	获取相机旋转矩阵
公有	<code>GetTranslation()</code>	<code>cv::Mat</code>	获取相机平移向量
公有	<code>GetRotationInverse()</code>	<code>cv::Mat</code>	获取相机旋转矩阵的逆矩阵
公有	<code>GetCameraCenter()</code>	<code>cv::Mat</code>	获取相机中心坐标
公有	<code>isInFrustum(MapPoint*, float)</code>	<code>bool</code>	判断地图点是否在当前帧视野中

属性	成员函数	返回值	描述
公有	<code>IsInImage(const float&, const float&)</code>	<code>bool</code>	判断点是否在去畸变图像中
公有	<code>PosInGrid(const cv::KeyPoint&, int&, int&)</code>	<code>bool</code>	计算关键点的网格坐标
公有	<code>GetFeaturesInArea(const float&, const float&, const float&, const int, const int)</code>	<code>vector<size_t></code>	获取某个区域的内的所有关键点索引

2.3 KeyFrame

属性	成员变量	类型	描述
公有	<code>mnId</code>	<code>long unsigned int</code>	关键帧id
公有	<code>mTimeStamp</code>	<code>const double</code>	时间戳
公有	<code>mK</code>	<code>const cv::Mat</code>	相机内参矩阵
公有	<code>mvKeys</code>	<code>const vector<cv::KeyPoint></code>	原始关键点
公有	<code>mvKeysUn</code>	<code>const vector<cv::KeyPoint></code>	去畸变的关键点
公有	<code>mDescriptors</code>	<code>const cv::Mat</code>	描述子
公有	<code>N</code>	<code>const int</code>	关键点数量
公有	<code>mBowVec</code>	<code>DBow2::BowVector</code>	词袋向量
公有	<code>mFeatVec</code>	<code>DBow2::FeatureVector</code>	特征向量
公有	<code>mnScaleLevels</code>	<code>const int</code>	尺度数量
公有	<code>mfScaleFactor</code>	<code>const float</code>	尺度倍数
公有	<code>mfLogScaleFactor</code>	<code>const float</code>	对数尺度倍数

属性	成员变量	类型	描述
公有	<code>mvScaleFactors</code>	<code>const vector<float></code>	尺度数组
公有	<code>mvLevelSigma2</code>	<code>const vector<float></code>	方差数组
公有	<code>mvInvLevelSigma2</code>	<code>const vector<float></code>	逆方差数组
公有	<code>mnTrackReferenceForFrame</code>	<code>long unsigned int</code>	用于更新局部关键帧
公有	<code>mnFuseTargetForKF</code>	<code>long unsigned int</code>	用于融合地图点
公有	<code>mnBALocalForKF</code>	<code>long unsigned int</code>	用于局部BA
公有	<code>mnBAFixedForKF</code>	<code>long unsigned int</code>	用于局部BA
私有	<code>mTcw</code>	<code>cv::Mat</code>	相机位姿
私有	<code>mTwc</code>	<code>cv::Mat</code>	相机位姿逆矩阵
私有	<code>mOw</code>	<code>cv::Mat</code>	相机中心坐标
私有	<code>mpMap</code>	<code>Map*</code>	地图
私有	<code>mvpMapPoints</code>	<code>vector<MapPoint*></code>	地图点数组
私有	<code>mGrid</code>	<code>vector<vector<vector<size_t>>></code>	关键点在网格分布的索引数组
私有	<code>mConnectedKeyFrameWeights</code>	<code>map<KeyFrame*, int></code>	共视图映射
私有	<code>mvpOrderedConnectedKeyFrames</code>	<code>vector<KeyFrame*></code>	共视图关键帧数组
私有	<code>mvOrderedWeights</code>	<code>vector<int></code>	共视图权重数组
私有	<code>mpParent</code>	<code>KeyFrame*</code>	父关键帧
私有	<code>mvpChildrens</code>	<code>set<KeyFrame*></code>	子关键帧集合
私有	<code>mbBad</code>	<code>bool</code>	bad标记, LocalMapping线程删除关键帧时被设置

属性	成员函数	返回值	描述
----	------	-----	----

属性	成员函数	返回值	描述
公有	<code>KeyFrame(Frame&, Map*, KeyFrameDatabase*)</code>		构造函数
公有	<code>SetPose(const cv::Mat&)</code>	<code>void</code>	获取相机位姿
公有	<code>GetPose()</code>	<code>cv::Mat</code>	获取相机位姿
公有	<code>GetPoseInverse()</code>	<code>cv::Mat</code>	获取相机位姿的逆矩阵
公有	<code>GetCameraCenter()</code>	<code>cv::Mat</code>	获取相机中心坐标
公有	<code>GetRotation()</code>	<code>cv::Mat</code>	获取相机旋转矩阵
公有	<code>GetTranslation()</code>	<code>cv::Mat</code>	获取相机平移向量
公有	<code>ComputeBoW()</code>	<code>void</code>	计算BoW特征
公有	<code>AddConnection(KeyFrame*, const int&)</code>	<code>void</code>	共视图添加连接
公有	<code>EraseConnection(KeyFrame*)</code>	<code>void</code>	共视图删除连接
公有	<code>UpdateConnections()</code>	<code>void</code>	计算自身共视图
公有	<code>UpdateBestCovisibles()</code>	<code>void</code>	计算其他帧共视图
公有	<code>GetConnectedKeyFrames()</code>	<code>set<KeyFrame*></code>	获取所有共视帧
公有	<code>GetVectorCovisibleKeyFrames()</code>	<code>vector<KeyFrame*></code>	获取所有共视帧
公有	<code>GetBestCovisibilityKeyFrames(const int&)</code>	<code>vector<KeyFrame*></code>	获取权重最高的前N个共视帧
公有	<code>GetCovisiblesByWeight(const int&)</code>	<code>vector<KeyFrame*></code>	获取权重大于w的共视帧
公有	<code>GetWeight(KeyFrame*)</code>	<code>int</code>	获取共视图某帧权重
公有	<code>AddChild(KeyFrame*)</code>	<code>void</code>	添加子关键帧

属性	成员函数	返回值	描述
公有	<code>EraseChild(KeyFrame*)</code>	<code>void</code>	删除子关键帧
公有	<code>ChangeParent(KeyFrame*)</code>	<code>void</code>	改变父关键帧
公有	<code>GetChilds()</code>	<code>set<KeyFrame*></code>	获取所有子关键帧
公有	<code>GetParent()</code>	<code>KeyFrame*</code>	获取父关键帧
公有	<code>AddMapPoint(MapPoint*, const size_t&)</code>	<code>void</code>	添加地图点
公有	<code>EraseMapPointMatch(const size_t&)</code>	<code>void</code>	删除地图点
公有	<code>EraseMapPointMatch(MapPoint*)</code>	<code>void</code>	删除地图点
公有	<code>ReplaceMapPointMatch(const size_t&, MapPoint*)</code>	<code>void</code>	替换地图点
公有	<code>GetMapPoints()</code>	<code>set<MapPoint*></code>	获取所有有效地图点
公有	<code>GetMapPointMatches()</code>	<code>vector<MapPoint*></code>	获取所有地图点
公有	<code>GetMapPoint(const size_t&)</code>	<code>MapPoint*</code>	获取地图点
公有	<code>TrackedMapPoints(const int&)</code>	<code>int</code>	跟踪地图点，返回数量
公有	<code>GetFeaturesInArea(const float&, const float&, const float&)</code>	<code>vector<size_t></code>	获取某个区域的内的所有关键点索引
公有	<code>IsInImage(const float&, const float&)</code>	<code>bool</code>	判断点是否在去畸变图像中
公有	<code>SetBadFlag()</code>	<code>void</code>	设置bad标记
公有	<code>isBad()</code>	<code>bool</code>	判断关键帧是否为bad
公有	<code>ComputeSceneMedianDepth(const int)</code>	<code>float</code>	单目计算场景深度

2.4 Map

属性	成员变量	类型	描述
公有	<code>mMutexMapUpdate</code>	<code>mutex</code>	地图更新互斥锁，用于局部BA
公有	<code>mspKeyFrames</code>	<code>set<KeyFrame*></code>	关键帧集合
公有	<code>mvpMapPoints</code>	<code>set<MapPoint*></code>	地图点集合
公有	<code>mvpLocalMapPoints</code>	<code>vector<MapPoint*></code>	局部地图点数组，用于可视化

属性	成员函数	返回值	描述
公有	<code>Map()</code>		构造函数
公有	<code>clear()</code>	<code>void</code>	清空地图
公有	<code>AddKeyFrame(KeyFrame*)</code>	<code>void</code>	添加关键帧
公有	<code>EraseKeyFrame(KeyFrame*)</code>	<code>void</code>	删除关键帧
公有	<code>GetAllKeyFrames()</code>	<code>vector<KeyFrame*></code>	获取所有关键帧
公有	<code>KeyFramesInMap()</code>	<code>long unsigned int</code>	获取关键帧数量
公有	<code>AddMapPoint(MapPoint*)</code>	<code>void</code>	添加地图点
公有	<code>EraseMapPoint(MapPoint*)</code>	<code>void</code>	删除地图点
公有	<code>GetAllMapPoints()</code>	<code>vector<MapPoint*></code>	获取所有地图点
公有	<code>MapPointsInMap()</code>	<code>long unsigned int</code>	获取地图点数量
公有	<code>SetLocalMapPoints(const vector<MapPoint*>&)</code>	<code>void</code>	设置局部地图点数组
公有	<code>GetLocalMapPoints()</code>	<code>vector<MapPoint*></code>	获取局部地图点数组

2.5 数据结构维护

Map全局地图

- 维护所有关键帧集合(`set<KeyFrame*> mspKeyFrames`)

- 维护所有地图点集合(`set<MapPoint*> mspMapPoints`)

Tracking线程

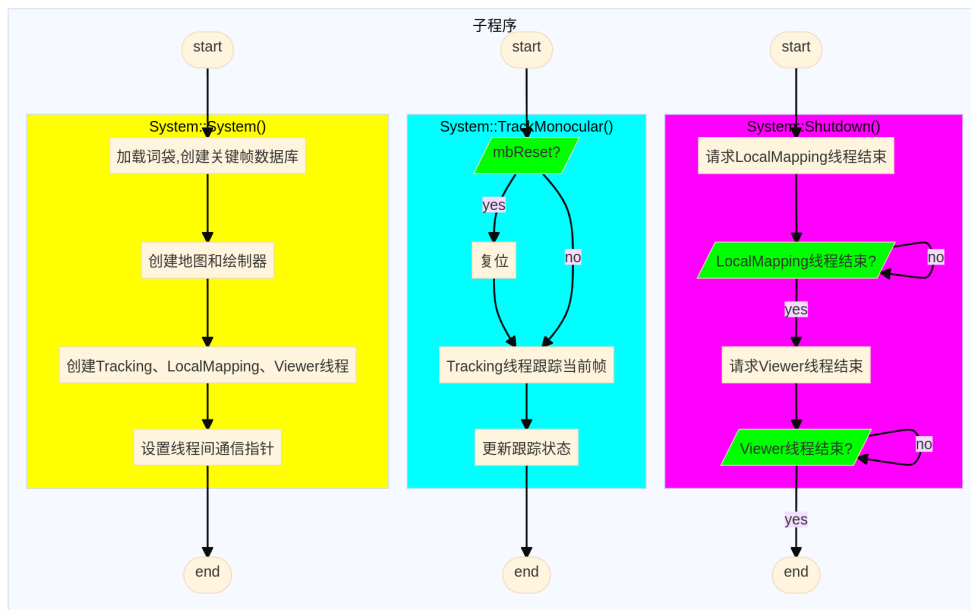
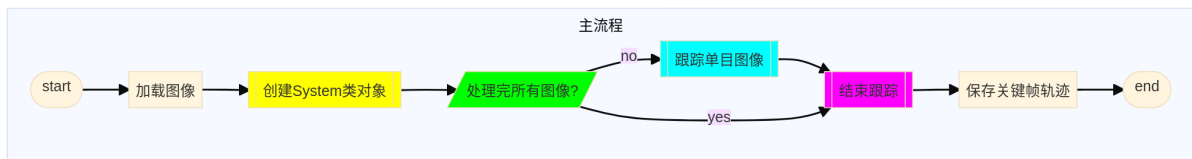
- 维护当前帧(`Frame mCurrentFrame`)
- 维护上一帧(`Frame mLastFrame`), 用于运动模型跟踪
- 维护参考关键帧(`KeyFrame* mpReferenceKF`), 用于跟踪参考关键帧
- 维护局部关键帧数组(`vector<KeyFrame*> mvpLocalKeyFrames`), 用于跟踪局部地图
- 维护局部地图点数组(`vector<MapPoint*> mvpLocalMapPoints`), 用于跟踪局部地图

LocalMapping线程

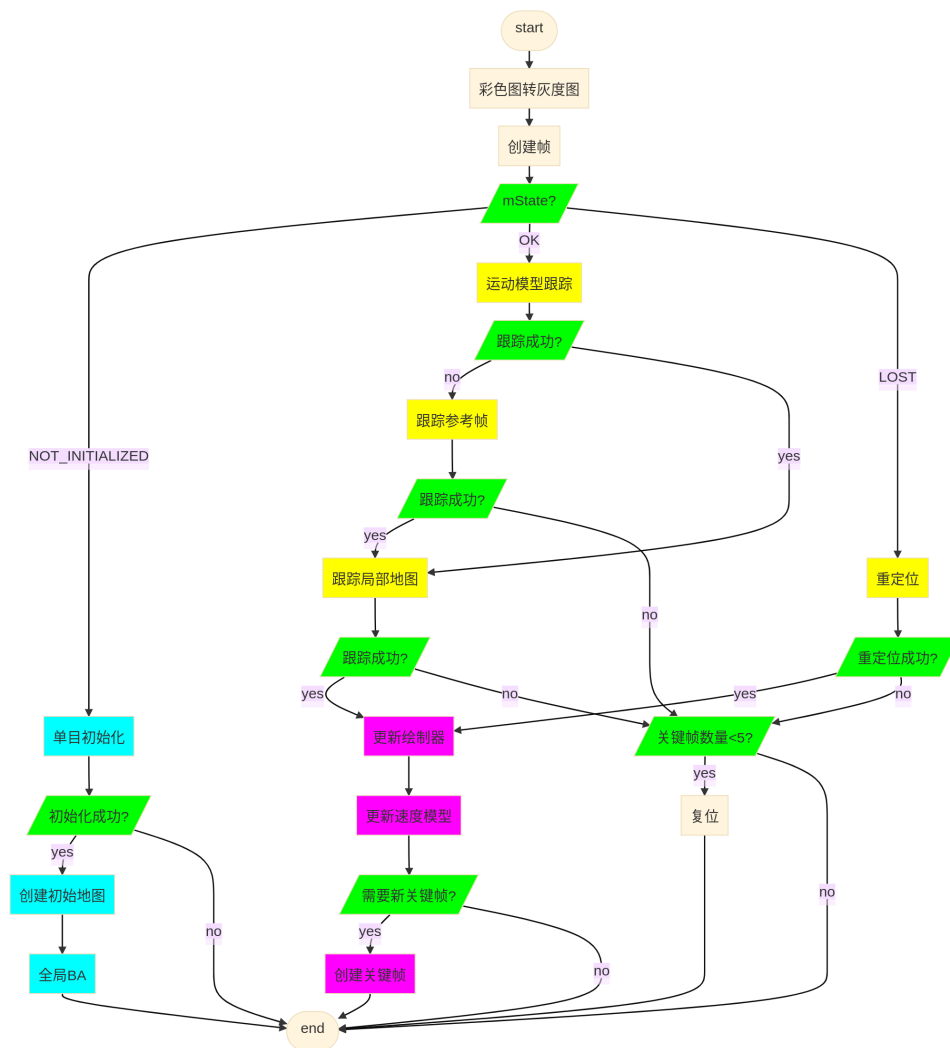
- 维护当前关键帧(`KeyFrame* mpCurrentKeyFrame`)
- 维护新建关键帧队列(`list<KeyFrame*> mInNewKeyFrames`), 用于与Tracking线程同步
- 维护新建地图点队列(`list<MapPoint*> mInRecentAddedMapPoints`)

3. 代码流程

3.1 主线程



3.2 Tracking线程



3.3 LocalMapping线程

