ORB-SLAM 模块逻辑 以stereo-kitti为例

main

根据数据集格式，获取帧时间戳和各帧对应的图片路径。构造system系统。进入读书循环，读取左右图片，然后调用system：：TrackStereo做跟踪。目前看跟踪线程就是主线程。

System 的构造函数

设置sensor类型

调用cv::FileStorage 解析参数配置文件

载入ORB字典

构造关键帧数据库

构造Atlas

构造FrameDrawer和MapDrawer用于绘图

构造Tracking对象

构造LocalMapping对象，新建线程启动run函数，设定参数

构造LoopClosing对象，新建线程启动run函数，

如果需要可视化，构造Viewer对象，新建线程启动run函数。

让各个模块互相拥有指针

将log输出设定为QUIET

System：：TrackStereo

检查传感器类型是否匹配

加锁，根据定位模式，设置跟踪线程和建图线程

加锁，根据设置确定是否重置跟踪线程

如果有IMU，加入IMU数据（通过调用GrabImuData）

调用GrabImageStereo处理本帧图像

根据跟踪结果设置当前跟踪状态

返回跟踪结果

Tracking的构造函数

解析参数（包括相机参数，ORB程序参数和IMU参数）和设置变量

Tracking：：GrabImageStereo

复制输入图片

将图片转换为灰度图

构建Frame对象（包含图片数据，相机参数和IMU数据，此时已经提取了ORB特征点，并做了左右图像特征点匹配，包括一次亚像素匹配；最后做了三角化，测量了特征点的深度。）

调用Track();

Tracking::Track()

等待绘图线程同步

根据需要重置地图

获取当前地图

处理帧时间错乱

处理跳帧（包括重置活动地图/创建新地图等）

设置IMU偏置

状态演化（NO\_IMAGES\_YET->NOT\_INITIALIZED）

IMU预积分

双目初始化

更新FrameDrawer;

Tracking::PreintegrateIMU

清理mvImuFromLastFrame

检查mlQueueImuInput的长度，没有数据，返回

从队列中获取数据

如果Imu数据点的时间在上一帧周期内很远，丢弃数据

取完数据跳出循环

数据不足，休息500微秒，接着取

构造一个Imu积分器

对Imu数据点做线性插值，送入积分器

完成后设置预积分完成标志

ImuTypes::IntegrateNewMeasurement()

使用迭代的方式计算三个预积分量，其中旋转矩阵积分一次就做一次svd分解，保证dR为正交矩阵。

迭代计算预积分对偏置的雅可比

更新预积分的协方差（协方差包括，原有噪声随着状态演化的传播和每次新加入测量带来的噪声））

Tracking::MonocularInitialization

Tracking::StereoInitialization()

如果关键点数少于500个，初始化失败。

ORBextractor，在提取器的构造函数中设置提取算法参数，使用（）运算符执行算法。相比于普通的ORB角点，orb-slam3的改进如下：

双阈值：先将图片分割为30\*30pixel的子块，在每个子块上分别使用高阈值提取角点，如果某个子块没有提取到角点，再改用低阈值。确保纹理较弱的位置也能提取到角点。

四叉树均匀化：使用四叉树将角点们隔离开来，在每个叶子节点上分别执行非极大值抑制，从而保护了响应度较低的角点，确保角点在图片中均匀分布。

需要注意的细节有：

给图片加上边界（BORDER\_REFLECT\_101），这样既能尽量应用所有的像素信息，也能避免边界上像素突变造成的干扰。

遗留问题

四叉树分裂时，每个节点的成员变量记录了当前时刻节点链表的头部的迭代器，但似乎没有用。