1. 摘要
   1. 此项目是同事遗留交接给本人，最初效果是在windows平台上可以运行，输出相机position与rotation信息。本项目在研发阶段也由我协助进行相机标定以及Camera模块的开发。本项目基于ORB\_SLAM框架。目前完成了前后端，还缺少回环检测与建图方面的工作，在小范围，物体静止且没有户外景物与玻璃反光的条件下，还是能够完成任务。
2. Pipline：
   1. 双目匹配：进入slam线程后在左右相机进行orb特征点匹配，当特征点匹配数量高于阈值时，进行undistort 和3d点生成。
   2. 前后帧匹配：上一帧特征点与当前帧特征点进行匹配，用搜索到的匹配点对相机位姿进行优化（假定匹配得到的orb特征点的3d位置不变，G2O 优化vertex为相机位姿，edges是当前帧重投影误差（上一帧orb点经过相机位移和旋转后）重投影到当前帧对应的像素坐标与当前帧检测出的匹配点像素坐标的距离））。
   3. 补充点：当前后帧匹配的点数目低于阈值时，需要进行场景内特征点的补充，这时会再次运行双目匹配获得新的一批特征点。
3. 算法优化：
   1. Smooth
      1. 用kalman filter 来smooth 相机position信息。
   2. Imu 数据来约束G2O的optimize。
      1. 开启imu\_reader。
      2. 双目匹配成功后将imu读数做一个旋转变换置为（1，0，0，0）（起始点x，y为（0，0），朝向12点钟方向）。之后每次用到读数都需要变换。
      3. 在原有g2o优化项中增加一条边，为四元数转旋转矩阵后与imu读数的欧式距离。并且基于边的数量（前后帧匹配到的点的数量）乘以系数增加权重。
   3. 工程化
      1. Jetson nano 移植。
      2. 与手势项目的合并
         1. 合并之后的项目就完全实现了在虚拟环境下同时拥有手势识别和6Dof效果。
      3. 编写获取数据/状态接口。
      4. api文档与so/dll。
      5. 异常状态划分与反馈log。