Main

读取相机配置文件，orb字典

初始化tracker 对于第一帧mstate为NO\_IMAGES\_YET

tracker.run

调用GrabImage方法。 将图像转换为灰度图，初始化frame:首先在构建图像金字塔，然后在每一层上提取关键点和描述子，对提取的关键点进行去畸变处理，对于第一帧，对图像的四个角进行去畸变处理，对图像构建64\*48的grid，将提取的特征点按照位置分配到不同的cell中去

如果mstate是NO\_IMAGES\_YET，mstate=NOT\_INITIALIZED。如果当前帧提取的特征点数大于100，才进行初始化。将当前帧mInitialFrame和上一帧mLastFrame均设置为第一帧，初始化初始化器，得到当前帧的关键点，方差设置为1，mstate设置为INITIALIZING 当前帧的位姿为空。 循环再进行GrabImage 得到新帧

初始化

如果mState是INITIALIZING, 得到当前帧关键点描述子，grid等。如果当前帧特征点数小于100，mState=NOT\_INITIALIZING；否则对于第一帧中的每一个关键点xy，先在对应尺度上的当前帧xy附近的窗口大小中得到可能匹配的关键点，然后利用描述子之间的距离，得到最优匹配的关键点。如果检查旋转的话，就只考虑那些匹配关键点角度差异最类似的那些匹配。最后得到成功匹配的关键点对数。如果小于100，mstate设置为NOT\_INITIALIZED，返回；否则，对于两帧的匹配特征点，循环选取200组，每组8对匹配关键点，分别计算单应矩阵和基础矩阵(基础矩阵和单应矩阵都是3\*3 和Rt形成的变换矩阵要区分!!!不一样)

单应矩阵：(单应矩阵是一个直接描述像素坐标之间转换的矩阵)对两帧的关键点进行归一化处理(得到的T是什么 是相似变换因子)，迭代200次，每次利用8对关键点对计算单应矩阵：对归一化的8个特征点对，按照slambook中的方法，svd分解，得到单应矩阵(svo中调用了cv库，但是原理一样，和slambook159介绍的相同)因为直接线性变换 不具有相似变换不变性，会受到相机坐标系尺度，定向，原点的不同选择有影响，所以先把图像通过相似变换因子使得图像的形心在原点，到原点的平均距离为根号2，再计算单应矩阵，最后通过T’-1\*H\*T恢复单应矩阵。

对于每次迭代得到单应矩阵，分别将第一帧和当前帧的所有匹配关键点向另外一方进行重投影，将误差在阈值范围内的关键点对记录下来。并取得当前单应矩阵的分数。得到最佳的score的单应矩阵作为结果。

基础矩阵：和单应矩阵不是很一样，这里得到了相似变换因子后没有求逆而是求了转置。。。

利用八点法，但是这里没有使用归一化坐标，直接利用的是像素坐标...进行奇异值分解，得到基础矩阵，强制令其的奇异值两个相等，另外一个为零。然后利用重投影，将第一帧和当前帧的所有匹配关键点向另外一方进行重投影，将误差在阈值范围内的关键点对记录下来。并取得当前单应矩阵的分数。得到最佳的score的基础矩阵作为结果。(这里和单应矩阵不太一样... 这里没有求逆，直接求的转置...难道基础矩阵是正交阵??? 计算重投影误差也有点奇怪)

根据单应矩阵和基础矩阵的分数比率，适应性的选择使用单应矩阵还是基础矩阵

如果是单应矩阵:得到由单应矩阵可能的8中Rt分解(n是什么？和svo对比一下，应该差不多)对于每一组Rt分解：

得到摄像机光心的位置，对于满足单应矩阵误差的关键点对，利用分解得到的Rt构建投影矩阵，进行三角化(利用两帧的位姿和对应关键点的像素坐标 KTX = x根据两个帧 得到四组等式(本来有六组，但是有依赖关系) svd进行求解)得到地图点相对于第一帧的三维坐标。 如果三维坐标是无穷大，vbGood设置为false. 根据相机参数，将三维点分别投影到第一帧和第二帧，计算重投影误差。将满足误差范围的，保存视差角度(-RT\*t)，相对于第一帧的地图点三维坐标，并得到最大的视差角度，返回符合所有条件的地图点个数。

对于六组rt分解，找到三角化后满足重投影误差最多的，(如果次好的方案小于最好方案的3/4 地图点在两帧的视差角度都小于1.0 满足重投影误差的地图点大于50 而且满足条件的点数大于总数的90%)

如果是基础矩阵: 得到四组可能的Rt分解。 如果没有特别突出的解，则拒接该结果，否则通过重投影误差，得到最优的rt分解。

现在我们得到了从第一帧到第二帧的位姿变换。删选两帧匹配的地图点，留下根据变换矩阵和分解的rt，满足重投影误差的匹配关键点。

对于第一帧和当前帧，新建关键帧(保存位姿和光心位置)，对这两个关键帧分别计算词袋。将这两个关键帧加入地图。对于匹配关键帧前一部分生成的相对于第一帧的地图点三维坐标，新建地图点，将地图点分别加入这两个关键帧，建立关键帧关键点与地图点的对应。建立地图点与这两个关键帧的对应。通过地图点mObservations变量，得到可以在哪一个关键帧中看到，对应哪一个关键点。得到平均描述子距离最小的地图点关联的关键帧，将地图点的描述子mDescriptor设置为该关键帧对应关键点的描述子。更新地图点的法向(能够看到它的关键帧光心与地图点方向的平均)和深度(根据地图点对应在当前帧的level得到地图点对应的最大最小深度范围????)。当前帧的关键点与地图点建立联系。将地图点加入地图。对于第一个关键帧和当前关键帧，更新共视图。(对于关键帧，找到共视地图点大于15的其他关键帧，将两个关键帧进行权重连接，权重为共视点的个数。如果没有超过15个共视点的关键帧，将共视点最多的关键帧与其相连)，(如果该帧还没有建立生成树，将父亲节点设置为与第一帧共视点最多的帧,孩子设置为自己(每一个关键帧都有一个父亲节点))

对于地图中的所有关键帧 和地图中的地图点，进行全局BA，优化关键帧的位姿和地图点的三维坐标，更新地图点的法向和深度

得到第一个关键帧地图点深度的中位数，如果深度中位数小于零或者第二个关键帧对应的地图点数目小于100，初始化失败，重置返回

对第二个关键帧的位姿中的位移向量进行scale,乘以在第一帧中深度中位数的倒数，将地图点也乘以中位数的倒数(这里和svo是一样的) 将第一第二关键帧加入localMapping线程的mvpLocalKeyFrames，设置上一帧为当前帧，上一个关键帧为当前关键帧。将第一第二关键帧加入局部关键帧组mvpLocalKeyFrames中。将地图中的地图点加入局部地图点组mvpLocalMapPoints中，参考关键帧为第二关键帧，设置地图的参考地图点 mstate=working

其他帧

调用GrabImage方法。 将图像转换为灰度图，初始化frame:首先在构建图像金字塔，然后在每一层上提取关键点和描述子，对提取的关键点进行去畸变处理，对图像构建64\*48的grid，将提取的特征点按照位置分配到不同的cell中去

如果mstate是working而且不需要重定位

如果没有运动模型或者目前关键帧个数小于4，或者当前帧在刚刚重定位的帧2帧之内，通过跟踪上一帧来恢复位姿；否则，根据运动模型来更新位姿，如果运动模型更新失败，则通过跟踪上一帧来恢复位姿

跟踪上一帧来恢复位姿：

当前帧的位姿估计为上一帧的位姿

（找到初始对应，优化位姿，对于不在误差范围的关键点，利用位姿重投影找到更好的关键点位置，利用更好的关键点匹配得到更好的位姿。）

首先得到一个匹配关键点的尺度，在这个尺度下，遍历上一帧关联的地图点，对于地图点对应的关键点，在关键点周围一个窗口范围内，找到和该关键点一个level的关键点集合，对于这个集合，按照描述子的距离，找到最佳的匹配特征点，vpMapPointMatches2保存找到的关键点与地图点的对应，vnMatches21保存找到的关键点索引与前一帧对应关键点索引的对应。如果前后两帧匹配的关键点少于10个，在更大的level范围内查找匹配。当前帧的mvpMapPoints保存最终匹配的关键点与地图点的对应。

如果匹配点大于10，对于当前帧的位姿估计，还有当前帧关键点与地图点的联系，进行poseoptimization(其实就是利用当前帧的位姿，对地图点进行重投影，计算误差，优化当前帧的位姿),返回在误差范围内的关键点个数，对于不在误差范围内的地图点，将关键点相连的地图点设置为null。 对于这部分不在误差范围内的地图点，将地图点按照当前帧的位姿(优化后)重投影到当前帧，在该位置附近一个窗口，按照地图点在上一帧中对应的关键帧的level得到一系列候选。利用描述子距离，得到最匹配的关键点，重新建立关键点与此地图点的联系；如果匹配点小于10，直接利用上一帧的位姿，通过重投影，得到区域，描述子匹配，得到更多匹配的关键点。

有了更准确的特征点对应关系，再进行poseOptimization,根据重投影误差优化当前帧的位姿。这一次对于离群的关键点，将其对应的地图点设置为空。如果匹配上的点大于10个，则认为成功，转入TrackLocalMap否则就要进行重定位了。。。

根据运动模型来更新位姿：

一开始的velocity是单位阵，当前帧的位姿为velocity\*上一帧的位姿。Velocity更新：上上帧的位姿的逆乘以上一帧的位姿。

遍历上一帧的地图点，得到在第一帧中的三维坐标，利用当前帧位姿估计投影到当前帧，在当前帧一定窗口大小内通过描述子距离得到地图点在当前帧的最佳匹配，如果匹配点小于20，直接返回。否则进行poseOptimization,根据地图点的重投影误差优化当前帧的位姿，对于离群点，将当前帧对应关键点的地图点设置为空。如果此时匹配点大于10，直接进行trackLocalMap

重定位：

计算当前帧的词袋，

如果没有强制重定位，对于当前帧，根据词袋，共享单词等来获得候选的重定位关键帧。

否则将上一个关键帧共视点最多的9个关键帧作为候选关键帧，以及这上一个关键帧。对于候选的N个重定位关键帧，要求解N个pnpSolver

(使用词袋来搜索匹配点，为什么不用前面类似的方法呢？？？)首先取出关键帧和当前帧的特征向量（注意这里的特征向量是根据节点id排序好的特征向量（用DBOW库中的特征向量），便于我们进行搜索），然后遍历关键帧的每一个特征向量，在当前帧的同一节点下搜索其匹配点（计算同一节点特征向量对应描述子的距离最小匹配作为其匹配点）。最后进行匹配筛选（剔除误匹配）。如果匹配的点小于15,将其从候选重定位关键帧中删除。

否则对于候选关键帧和当前帧的匹配特征点，已知三维坐标和在当前帧中的像素坐标，利用pnp求解当前帧的位姿。对于外点，将对应的地图点设置为null,否则将关键点与地图点对应。接下来进行poseOptimization优化当前帧的位姿。将外点对应的地图点设置为空。如果匹配点小于50，对于在pnp求解过后候选重投影关键帧中没有成功投影在当前帧中的地图点，利用当前帧位姿重投影+描述子比较得到匹配点，如果加上之前的内点>50，进行poseOptimization优化当前帧位姿 如果大于30小于50，对于在pnp+poseOptimization失败的地图点中找匹配点，如果大于50，poseOptimization,对外点对应的地图点置空。如果内点大于50，则说明重定位成功，不需要再遍历其他的候选重定位帧了。

TrackLocalMap

更新参考关键帧：遍历当前帧关联的地图点，对于该地图点的共视关键帧，keyframeCounter将关键帧对应的索引+1。参考关键帧组mvpLocalKeyFrames清空。对于所有共视关键帧，加入mvpLocalKeyFrames(应该是整个地图的局部关键帧)，设置共视关键帧的mnTrackReferenceForFrame为当前帧id.对于每个共视关键帧，将其有最多共视点的10个关键帧也加入mvpLocalKeyFrames。mpReferenceKF保存有最多共视点的关键帧

更新局部地图：mvpLocalMapPoints清空，对于所有的局部关键帧，得到关键帧关联的地图点，将这些地图点加入局部地图，将地图点的mnTrackReferenceForFrame设置为当前帧

对于当前帧关联的地图点，mvVisible++ mnLastFrameSeen设置为当前帧。因为局部地图更新了，将局部地图中的地图点(没有在当前帧中对应的)，按照当前帧的位姿投影到当前帧，得到在当前帧中的像素坐标.和三维坐标，需要判断深度在地图点的深度范围之内,地图点在关键帧中的平均方向与当前帧的方向夹角要小于阈值等 如果满足了这些条件，记录一下地图点在当前帧对应的像素，视线夹角，level,地图点mvVisible++.对于满足这些条件的地图点，在当前帧uv一定范围内(与视线夹角和level有关)通过描述子匹配得到对应在当前帧的关键点。 使用扩充了的地图点与当前帧的对应poseOptimization，通过重投影误差优化当前帧的位姿，得到符合该位姿的关键点数。如果小于30，或者在重定位帧mMaxFrames之内且小于50，返回失败。

如果成功返回，得到当前帧较准确的位姿，判断是否将当前帧作为关键帧(从上一次关键帧插入，过去了足够多的帧，localmapping线程空转，而且当前帧符合位姿的关键点数大于15且小于共视点最多关键帧关键点数的90%)，如果是，新建关键帧，插入到localMapping线程。将最后的离群点对应的地图点设置为空。Mstate为working

否则 mstate = lost 如果地图的关键点小于5，重置。

LocalMapping.run

如果tracking插入了关键帧，localMapping开始工作，不接受新的关键帧

对于该关键帧，计算词袋，遍历该关键帧的地图点，更新地图点的可视变量mObservations(关于该关键帧，index+1),更新地图点的法向和深度，更新这个地图点共视的所有关键帧在改关键点的描述子与其他关键帧该点的描述子的距离大小。对于当前关键帧，更新共视图，将当前关键帧加入地图。

对mlpRecentAddedMapPoints新加进来的地图点进行检查。根据一定条件进行删除。

得到与当前关键帧共视点最多的20个关键帧，遍历每一个关键帧，计算候选关键帧到当前关键帧的基础矩阵F(相当于一个逆过程 之前是由匹配点得到基础矩阵，分解得到rt,现在已知rt,需要更多的匹配点，就要先算出基础矩阵，利用基础矩阵搜索匹配点)，基于词袋，对于当前帧中没有地图点对应的关键点，根据描述子距离，得到不大于最小距离两倍的候选关键点，根据基础矩阵判断候选关键点与当前帧的关键点是否满足极线约束(基础矩阵的逆和转置问题)(这里的极线约束 更像是投影误差???)，得到与当前帧特征点匹配的参考帧关键点，进行三角化，得到新的地图点(根据两帧的位姿和匹配的像素点的归一化坐标，得到六组等式，拿到4组，svd分解，得到匹配点相对于第一帧的三维坐标 ref:https://blog.csdn.net/qq\_30356613/article/details/80587649 和初始化的时候的三角化思路相同) 将三角化得到的地图点投影到当前帧和参考帧，如果重投影满足误差，以及到光心的比率，则三角化成功，新建地图点，添加地图点与两帧对应关键点的联系，更新地图点的描述子，法向和深度，将地图点加入地图。

对于当前关键帧，找到共视点最多的20个关键帧，对于这些关键帧，找到共视点最多的5个关键帧。

对于这些关键帧，将当前关键帧的地图点与它们进行融合： 遍历当前帧的地图点，投影到参考帧，在投影点附近一个区域，利用描述子得到最匹配的关键点，如果该关键点已经有匹配的地图点，进行替换，否则将其与地图点相关联。

对于这些关键帧，将它们关联的地图点与当前帧进行融合。

更新连接信息。

进行局部ba: 得到当前关键帧共视点最多的15个关键帧，将这些关键帧和它们关联的地图点作为优化对象。将这些地图点所有共视关键帧也加入优化，但是位姿固定。优化与当前帧共视点最多的15个关键帧的位姿和相关地图点的三维位置。迭代五次，对于离群点，删除关键点与地图点的对应。再迭代10次。处理离群点，得到优化结果。

关键帧剔除

将当前关键帧加入mpLoopCloser

LoopClosing::Run

如果队列中有关键帧，得到当前关键帧。获得当前关键帧共视点最多的15个关键帧，得到词袋差异的最小值。删选不在共视图中相连的关键帧，经过单个，和联合分数的删选，得到一系列候选回环关键帧。然后与之前出现的回环关键帧组找重合，如果有重合，将重合的次数也加入mvConsistentGroups。

对于每个候选回环关键帧，利用描述子匹配，得到当前帧关键点在候选回环 在候选回环关键帧中寻找匹配点，将匹配点对应的地图点与当前帧的关键点进行对应。利用得到的对应，计算sim3相似变换。(每次利用三组三维坐标(本应该是同一个地图点，但是现在是两个，分别在参考帧和当前帧中的三维点)，利用重投影误差计算离群点，得到对于一组候选回环关键帧最优的相似变换)，此时得到了s,R,t,和内点。对于当前帧中的这些外点，利用计算得到的相似变换重新在参考帧中得到匹配点；对于参考帧对应的外点，同理在当前帧中找到最佳的匹配，如果双向匹配的是同一个点，则扩充在内点中。利用optimizeSim3进一步优化相似矩阵(优化回环关键帧和当前关键帧的相似矩阵)，得到外点。如果内点数量满足阈值，则对于候选关键帧，利用相似变换，得到更新后的位姿(但还没有更新)。不用再处理其他的候选回环关键帧了。

此时得到了回环关键帧 以及利用相似变换得到的候选关键帧应该的位姿。

得到回环关键帧共视点最多的15个关键帧以及所有相关的地图点。遍历这些地图点，利用需要更新的位姿，投影到当前帧，在当前帧中找到对应的匹配点。

进行回环矫正：

遍历当前关键帧组(共视点最多的15个关键帧)利用得到的相似变换，矫正当前关键帧组的位姿。遍历除了当前帧的当前关键帧组KF，得到与KF关联的地图点，利用相似矩阵，更新地图点的坐标。更新地图点。

地图点融合：

对于前面计算sim3时候得到的对应，将当前关键帧对应的地图点换成回环关键帧对应的地图点。如果当前帧中没有地图点对应，新生成地图点，与当前帧建立联系。(也就是两帧基本上一样了。)

利用当前关键帧组矫正后的位姿，重投影:遍历这些关键帧，将它们关联的地图点，利用矫正后的位姿将地图点重投影到自己，在一定范围内找到更合适的特征点，如果这个特征点有对应的地图点，进行替换，否则进行添加

对于当前关键帧组，得到共视点最多的15个关键帧，更新连接，对于当前关键帧组的每一个，得到除了共视点最多的15个关键帧和当前关键帧组的有共视点的关键帧集合，optimizeEssentialGraph(建立各种连接 固定相对位姿，优化相对于第一帧的绝对位姿),*优化得到所有关键帧的相似矩阵，得到位姿，以及地图点的位置*