

1. 计算外参数（旋转部分） ric

1. 通过当前帧和上一帧的匹配点对和分解基础矩阵找到相机的旋转，记录为Rc[i]
2. 利用上一次计算出来的ric，将imu所得到的旋转移移到camera上,记录为Rc_g[i]
3. 对比Rc[i]和Rc_g[i]之间的夹角，理论上如果ric是正确的话，那么这个值应该是0。实际上不太可能，我们将计算出来的这个值的导数当做权重，记录为huber。之后会用到。
4. 通过一下公式，我们可以得到

$$q_c \otimes q_i^c = q_i^c \otimes q_i$$
$$(L(q_c) - R(q_i))q_i^c = 0$$

期中的L和R分别为左乘和右乘矩阵，理论上每多一帧，都能够多一条方程，我们的方程不断累起来，同时用前面的huber作为系数，最终解最小二乘方程就可以得到 q_i^c 。

5. 取个逆就是 q_c^i 了。

为了保证这个过程比较合理，作者至少迭代了WINDOW_SIZE且保证了奇异值足够小的时候才输出。虽然我没有办法证明这种方法一定会收敛，但是从直觉上来说，应该是会逐渐收敛的才对。

```
bool InitialEXRotation::CalibrationExRotation(vector<pair<Vector3d, Vector3d>>
corres, Quaterniond delta_q_imu, Matrix3d &calib_ric_result)
```

2. 开始初始化

PS：只要有ric，我们就可以完成初始化了。

1. 初始化的时候，要保证有足够多的帧，至少要沾满滑动窗口
2. 先检查一下所有帧（包括被认为是second new的帧）的imu的加速度方差是否足够大，太小的话就丢弃。论文提到，初始化的时候默认imu的bias为0，实际上不可能，当方差很大的时候，就可以忽略不计了。

也就是说滑动窗口里面的帧是all frame的子集。

这里说一下MARGIN_SECOND_NEW和MARGIN_OLD

大致意思可以认为如果最新一帧和上一帧的视差太小的话，那就是太接近了，完全可以认为是没有的帧，这里就定义为MARGIN_SECOND_NEW,此时的处理方式是，用最新的这一帧去代替上一帧，然后这一帧加入all frame。

如果是MARGIN_OLD的话，那就是正常的滑动窗口了，直接删掉滑动窗口里面最旧的那一帧，同时删掉all frame里面最老的那一帧到滑动窗口准备删掉的那一帧（包括这一帧）。

3. 在滑动窗口里面检查一下与当前最新帧的视差和特征点都足够多的匹配帧，将这一帧记为l.同样通过F矩阵的方式，得到最新帧和l帧之间的相对位姿。
4. 然后根据这个初始的关系，不断用三角测量的方式加上Pnp和以上一帧的位姿为初值，不断的求解出滑动窗口里面的所有位姿。最终我们可以得到 T_{ci}^{cl}
5. 在求出滑动窗口里面的所有 T_{ci}^{cl} 之后，我们又通过Pnp的方式求得了除了滑动窗口外的所有帧的 T_{ci}^{cl} 。然后在这里执行了一步比较重要的操作，将 T_{ci}^{cl} 分别写成 R_{bi}^{cl} 和 t_{ci}^{cl}

这里注意，旋转和平移的参考帧不一样，这很重要，直接影响了我后面的卡壳了很久很久。

这里说下为啥会弄一个不一样的参考帧的个人理解：因为我们没有办法得到tic，我们只知道ric。

6. 到目前位置，我们已经得到了所有的 R_{bi}^{cl} ，和 R_{bi}^{bi-1} ，根据论文的意思，所有的误差均由IMU的bias引起，我们可以得到这个表达式

$$J_{b_w}^{\gamma T} 2(\hat{\gamma}^{-1} \otimes q_j^i)_{xyz} = J_{b_w}^{\gamma T} J_{b_w}^{\gamma} \delta b_w$$

同样，我们一个帧就会多一条表达式，这里不是用最小二乘的方法去求解的，而是通过直接线性法，直接左边求和，右边求和，直接求解 δb_w ，同时将这个增量更新到 b_w 上，然后重新以计算一下预积分。

7. 然后再根据论文的公式(18)(19)求速度、尺度和重力。公式和论文一模一样，然后用的依然是最小二乘的方法求解超定方程。这里说一下论文没有说得很详细的refineGravity。实际上做的事情就是，在上面求出来的重力粗值 g_0 ，重新写成

$$g = \text{normalizaed}(\text{normalized}(g_0) * G * lxly * dg) * G$$

的形式，这里的G指的是重力加速度的大小，lxly是一组和 g_0 正交的单位基底。将上面求解的时候用到的g，重新用这个替代进去，稍微变下形，待更新的量就变成了dg，不断更新到g里面，最终会收敛，代码里面的实现是固定迭代4次。

PS：这个过程中用到tic,但是如果没有的话会当成0。

8. 初始化所需要的参数我们已经全部求出来了。接下来将滑动窗口里面的帧都记录为关键帧，同时，用三角测量的方式重新对滑动窗口里面的特征点进行三角测量，然后再更新上尺度。
9. 将每个预积分重新更新。
10. 将 P_{ci}^{cl} 转化到 $R_{b0}^{cl} * P_{bi}^{b0}$

$$\begin{aligned} P_{bi}^{cl} - R_{ci}^{cl} P_c^i &= P_{c0}^{cl} - R_{c0}^{cl} P_c^i + R_{b0}^{cl} P_{bi}^{b0} \\ R_{b0}^{cl} P_{bi}^{b0} &= P_{bi}^{cl} - R_{ci}^{cl} P_c^i - (P_{c0}^{cl} - R_{c0}^{cl} P_c^i) \end{aligned}$$

这一条画图比较好证明一点。

$$V_{bi}^{b_i} \text{转化到 } R_{b0}^{cl} * V_{bi}^{b0} = R_{b0}^{cl} * (R_{bi}^{b0} * V_{bi}^{bi}) = R_{bi}^{cl} * V_{bi}^{bi}$$

$$R_{b0}^{cl} * R_{bi}^{b0} = R_{bi}^{cl}, \text{所以R不用做处理。}$$

这个东西就很重要了，卡壳了我好久。

网上的资料都瞎写，看了一个人的博客，基本都是乱解释，然后别的博客就跟着瞎抄，偶尔有人在评论区指出错误，但是却没有一个人给出正确的是什么样的东西。**基本都是看着像论文里面的公式14，就强行带入，欺骗自己。**我根据代码一步步跟踪和推测作者的心思，花了一天时间终于想明白了作者的意思是什么了。**一开始也严重的给我带偏了，甚至我开始怀疑代码有问题了。**

首先，理论上一点都没错，大家都是在同一个frame下面，前面多乘一个一样的旋转，一点都不影响使用。但是可以让代码变得稍微简洁一丢丢。可以看到上面的推导，只要增加这么一个小小的东西，所有的转化都变得很简洁，我估计这是作者这样写的主要原因，就是不太好理解就是了。

11. 我们已经求得了重力加速度的方向，我们希望重力加速度指向正上方。

网上很多人说什么重力明明是向下的，但是作者论文里却非要认为是向上的，所以要指向上方，我认为这是错误的。作者这么选是因为IMU读数读出来重力就上朝上的，而不是书写习惯的问题。

我们首先计算从当前的重力 \hat{g} 到(0,0,1)的转换关系，得到一个旋转矩阵R，注意理论上存在无数个。然后将这个矩阵表示为

$$R = R_Z R_Y R_X$$

的形式，我们去掉 R_Z ，此时

$$R = R_Y R_X$$

我们再将 $R_2 = RRs[0]$, 一样的将这个矩阵表示为

$$\begin{aligned} R_2 &= RRs[0] = R_2 = R_{Z2} R_{Y2} R_{X2} \\ \Rightarrow Rs[0] &= R_X^T R_Y^T R_{Z2} R_{Y2} R_{X2} \end{aligned}$$

的形式, 我们再去掉 R_{Z2} , 此时

$$R = R_{Z2}^T R_Y R_X$$

, 此时我们再

$$\begin{aligned} RRs[0] &= R_{Z2}^T R_Y R_X Rs[0] \\ &= R_{Z2}^T R_Y R_X R_X^T R_Y^T R_{Z2} R_{Y2} R_{X2} \\ &= R_{Z2}^T R_{Y2} R_{X2} \\ &= R_{Y2} R_{X2} \end{aligned}$$

这个矩阵, 其实就是最终展示出来有一个约束, 即第一帧的X轴会在Z-X平面上。同时, 重力加速度经过这个R转化后的朝向也是z轴正方向。

注意这里的表达式, $Rs[0] = R_{b0}^{cl}, \hat{g} = g^{cl}$

当 \hat{g} 和 $Rs[0]$ 的z轴平行的时候, $Rs[0]$ 的结果是 $Rz(0)$ 或者 $Rz(180)$

当 \hat{g} 和 $Rs[0]$ 的y轴平行的时候, $Rs[0]$ 的结果是 $Rx(90)$ 或者 $Ry(180)Rx(90)$

当 \hat{g} 和 $Rs[0]$ 的x轴平行的时候, $Rs[0]$ 的结果是 $Ry(-90)$ 或者 $Rx(\theta)Ry(-90)$

也就是说, 除非 \hat{g} 和 $Rs[0]$ 的x轴平行有无限多的位姿外, 其余位姿都是有限的。

会发现, $Rs[0]$ 变成了yaw为0的角度。

$$g = Rg = R_{Z2}^T R_Y R_X \hat{g} = R_{Z2}^T (0, 0, 1)^T G = (0, 0, 1)^T G$$

此时的重力加速度也确实指向了正上方。

至于这里整这么多的有的没的, 其实也不能够保证说相机的 $Rs[0]$ 的xy的指向如何, 只能够重力加速度指向正上方。对于一个姿态分解成 $R_Y R_X$ 的意义是什么, 我也不太能够确定。