目录

[Vins-Fusion 2](#_Toc23331)

[void sync\_process() 2](#_Toc14651)

[inputImage（time , image0 , image1） 2](#_Toc22320)

[IMU回调函数（imu\_callback） 4](#_Toc12233)

[InputIMU（acc，gyr） 4](#_Toc15619)

[特征回调（feature\_callback） 4](#_Toc1270)

[inputFeature（t , featureFrame） 4](#_Toc45)

[processMeasurements（） 4](#_Toc28548)

[initFirstIMUPose（accVector）初始化第一帧IMU数据 5](#_Toc12278)

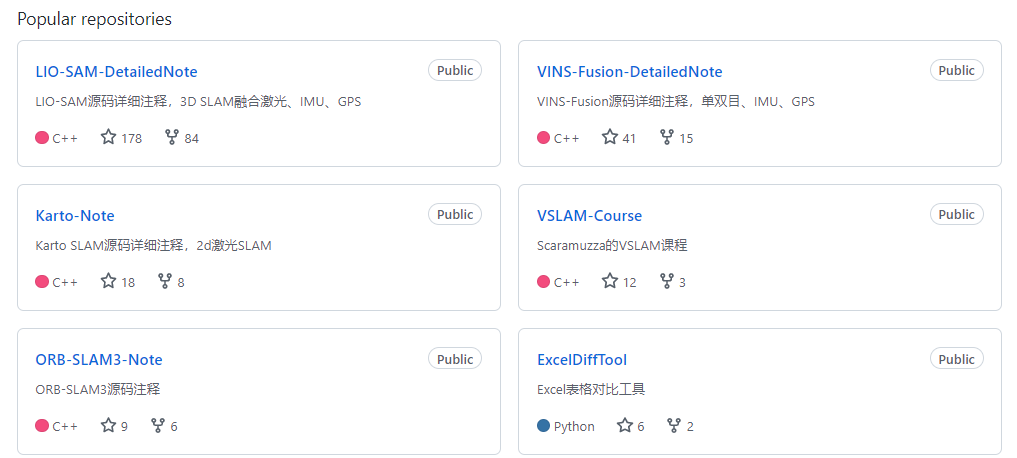
[processIMU 5](#_Toc28394)

[processImage（图像，t）https://zhuanlan.zhihu.com/p/270382090 6](#_Toc28191)

[回环优化 15](#_Toc17790)

[全局优化 16](#_Toc25814)

各框架源码注释：https://github.com/smilefacehh



Vins:https://blog.csdn.net/huanghaihui\_123/article/details/86518880?spm=1001.2101.3001.6650.11&utm\_medium=distribute.pc\_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7Edefault-11-86518880-blog-87357488.pc\_relevant\_antiscanv4&depth\_1-utm\_source=distribute.pc\_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7Edefault-11-86518880-blog-87357488.pc\_relevant\_antiscanv4&utm\_relevant\_index=13

Vins中相机和IMU对齐：<https://zhuanlan.zhihu.com/p/466221991>

Vins边缘化：<https://zhuanlan.zhihu.com/p/335242594>

Vins中Ceres（BA）：<https://zhuanlan.zhihu.com/p/488016175>  
vins回环：<https://blog.csdn.net/huanghaihui_123/article/details/87357488>

Evo工具使用：<https://blog.51cto.com/u_14411234/3127894>

# Vins-Fusion

## void sync\_process()

While{

If（双目）

如果两目的图像时间间隔不超过0.003秒，则取出两帧图像image0和image1。

### inputImage（time , image0 , image1）

inputImageCnt++（输入图像数量加1）

If (image1为空)

#### trackImage（t , img）

1.如果先前已经有了预测的特征点（predict\_pts），将predict\_pts赋值给当前的特征点 （cur\_pts）

1.1 进行光流预测

cv::calcOpticalFlowPyrLK(prev\_img, cur\_img, prev\_pts, cur\_pts, status, ...)

如果返回的status=1的个数（succ\_sum）小于10，则增加金字塔层数继续进行预测 （根据status变量是否为1来确定对应的点是否被追踪到）。

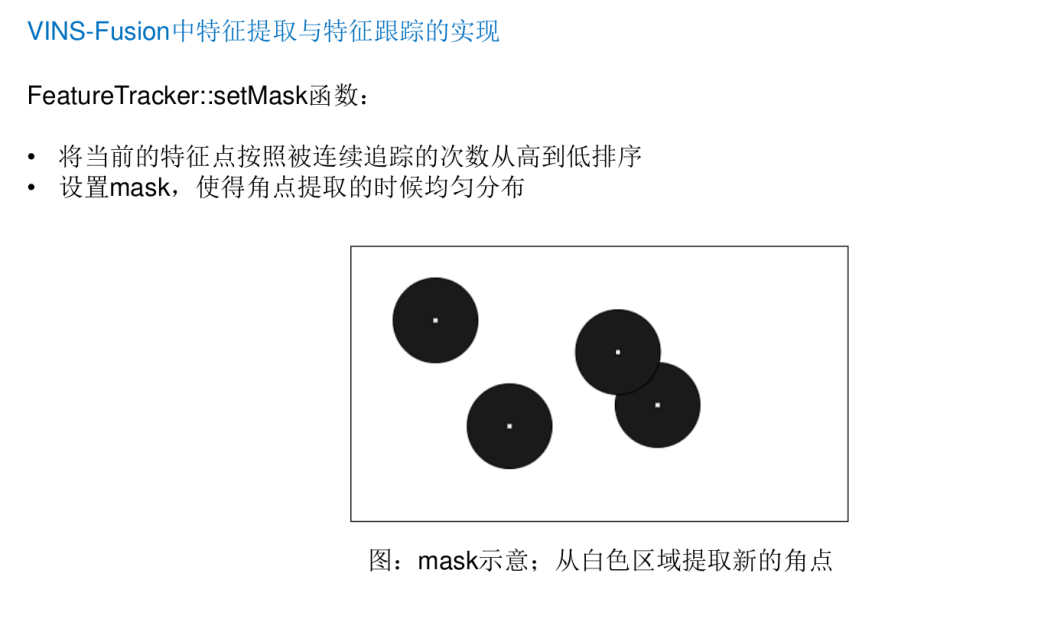
如果之前并没有预测的特征点，则直接设置3层金字塔进行预测。

2.if（FLOW\_BACK）

将当前图像与上一帧图像进行光流预测，将预测的点保存在reverse\_pts中

1. 去除外点（reduceVector），将丢失的点剔除

#### 4 . setMask()



#### 5 .提取特征点

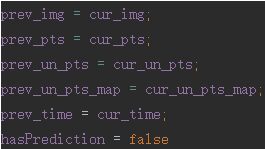
如果当前图像中特征点的数量少于配置文件中的数量（MAX\_CNT），则进行特征点 提取（goodFeatureToTrack），提取的特征点保存在n\_pts中

提取成功后，将提取的点存入cur\_pts中，id存入ids中，将1存入track\_cnt中

将cur\_pts中的点投影到归一化平面中，并去除畸变（cur\_un\_pts）。并计算他的运动 速度（pts\_velocity）。

如果右目图像不为空，则执行上述相同的操作。

将当前帧的数据全部赋值为上一帧



#### 构建特征帧

将去除畸变后的归一化平面的点的x,y,z（z=1）取出，将cur\_pts中的像素坐标x,y赋值给p\_u,p\_v，将速度pts\_velocity取出赋值，最后得到featureFrame

如果是双目，则对右目图像重复上述操作

最后返回featureFrame

将featureFrame和时间t存入featureBuf中，如果系统不是多线程的话，开始processMeasurements（）

如果是多线程的话，系统会在最开始读取配置文件参数，设置参数（setParameter）的时候就开始processMeasurements（）

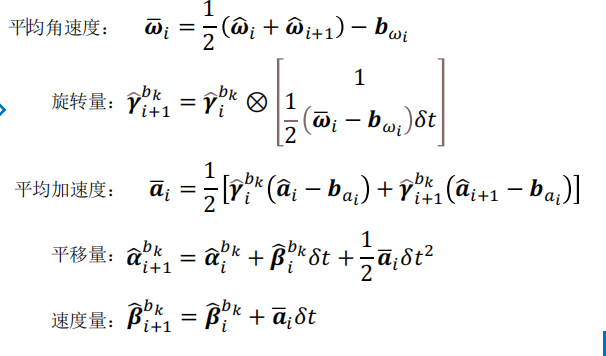
## IMU回调函数（imu\_callback）

读取加速度和角速度acc，gyr

### InputIMU（acc，gyr）

1. 将加速度和时间存入accBuf中，将角速度和时间存入gyrBuf中

#### fastPredictIMU（时间，加速度，角速度）



计算IMU预积分

latest\_time（时间）、latest\_Q（旋转量）、latest\_P（平移量）、latest\_V（速度量）、 latest\_acc\_0（上一帧的加速度）、latest\_gyr\_0（上一帧的角速度）。

## 特征回调（feature\_callback）

构建featureFrame

### inputFeature（t , featureFrame）

将featureFrame和时间一起存入featureBuf中

如果不是多线程，则进行processMeasurements（）

## processMeasurements（）

如果featureBuf不为空，取出featureBuf中的第一帧图像feature，将feature对应的时间+td（默认是0）赋值给curTime。

如果使用IMU，则进行

getIMUInterval（前一帧图像对应的时间，当前帧对应时间，加速度（accVector），角速度 （gryVector））

{这个函数主要是将在两帧图像之间的IMU的加速度和角速度分别保存在accVector和gryVector中，同时也会保存超过当前帧对应时间的后一帧IMU数据}

### initFirstIMUPose（accVector）初始化第一帧IMU数据

{

https://blog.csdn.net/huanghaihui\_123/article/details/103075107

计算accVector中所有加速度的平均值，将其转化成为旋转矩阵（R0）形式，在将R0转换成为欧拉角形式，取出偏航角y。。。。。

}

### processIMU

（当前IMU数据对应的时间（t），两个IMU之间的时间间隔（dt），加速度，角速度）

{

Push\_back（dt，加速度，角速度）

{

将dt，加速度，角速度存入dt\_buf，acc\_buf，gry\_buf中

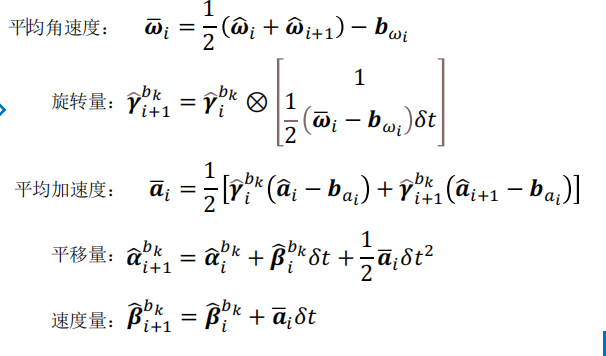
Propagate（dt，加速度，角速度）

{

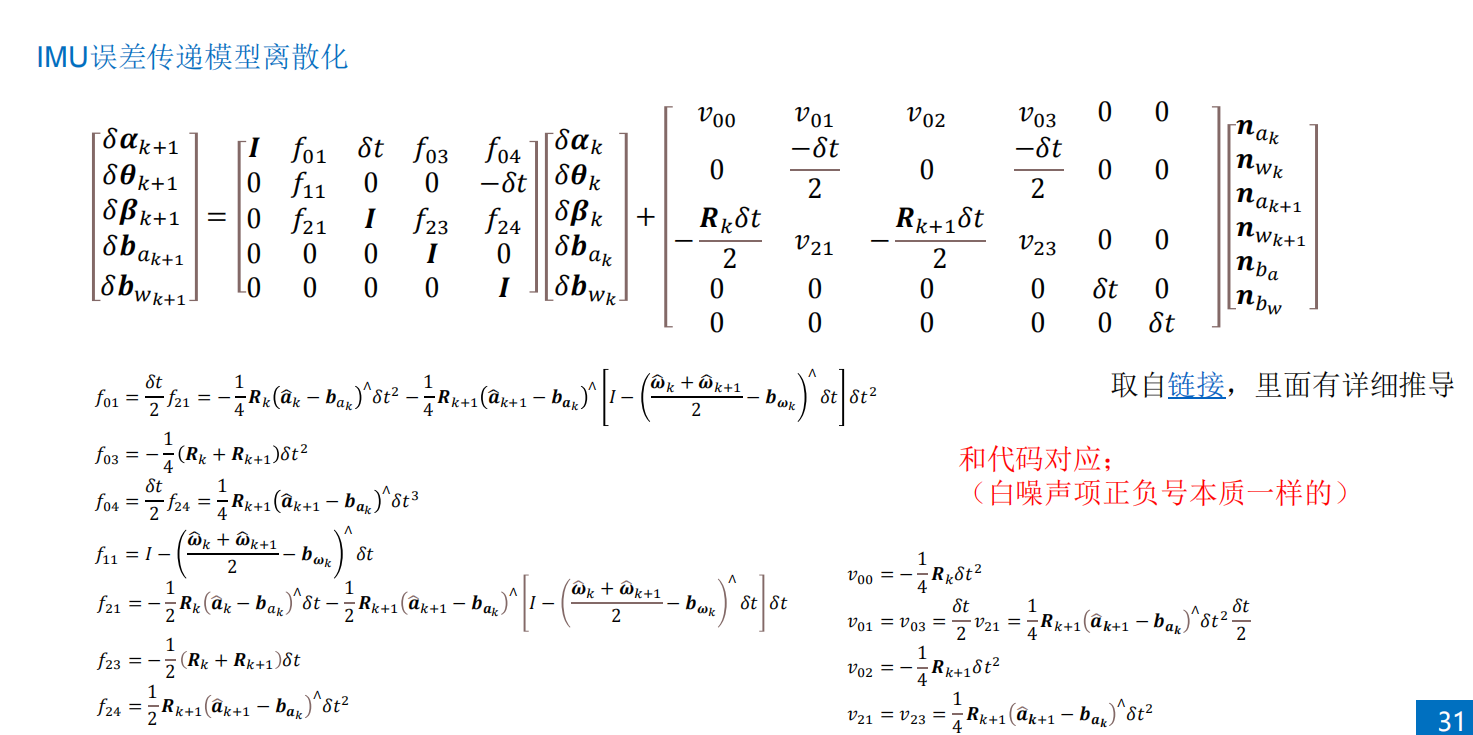
midPointIntegration（）

{

①IMU预积分



②计算雅克比矩阵（PPT第五章-视觉惯性里程计(中)）



}

将midPointIntegration得到的结果全部赋值进上一帧结果，dt累加

}

}

}

### processImage（图像，t）https://zhuanlan.zhihu.com/p/270382090

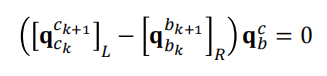
{

① addFeatureCheckParallax（）{根据设定的视差判断采取哪种边缘化策略}

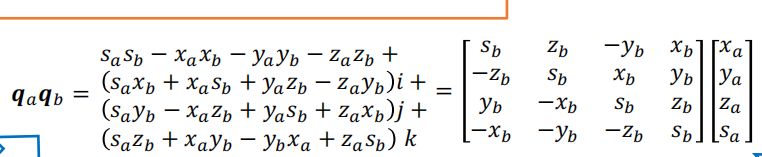
#### ② CalibrationExRotation（）计算相机和IMU之间的外参

{

1. solveRelativeR（通过PnP恢复出旋转矩阵，存入Rc中）
2. PPT第五章-视觉惯性里程计（中）（二）P14-16



PPT第四章-惯性传感器部分P31



构建L，R矩阵，然后再进行SVD分解，求出ric（相机到IMU之间的旋转（外参））

}

#### ③ 初始化（单目+IMU）initialStructure（）

{

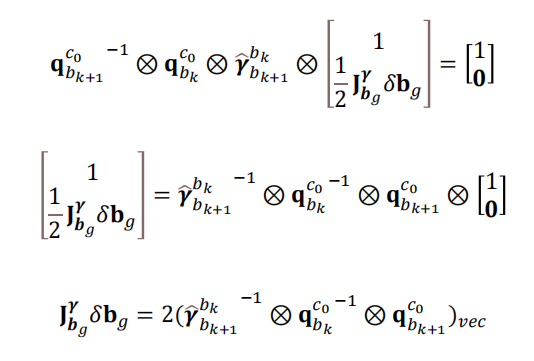
1. relativePose（）选择与最新帧有足够共视点，有足够的视差，并且能够解出relative\_R 和relative\_T的帧l
2. construct（）对11个关键帧做sfm
3. 对滑动窗口内的所有帧做sfm
4. VisualIMUAlignment视觉IMU联合初始化，计算bg、s、VS、g（视觉惯性对齐）

https://zhuanlan.zhihu.com/p/466221991

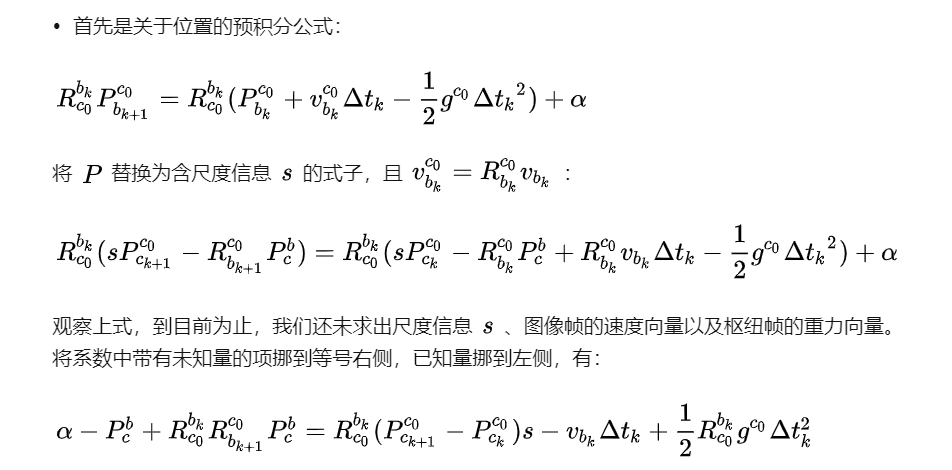
{

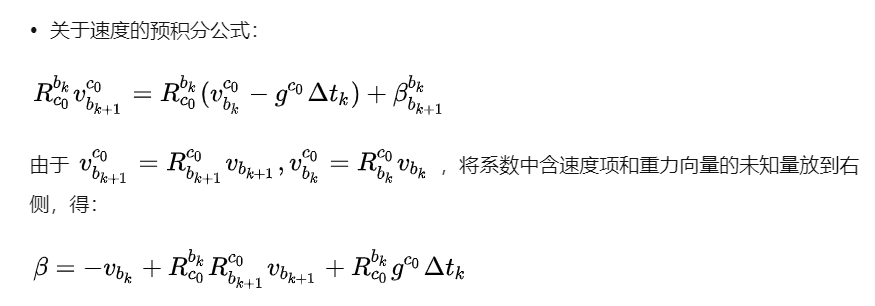
① solveGyroscopeBias（）初始化陀螺仪的bias

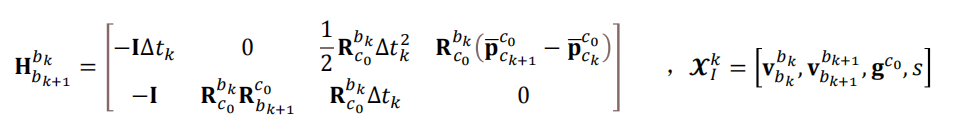
PPT第五章-视觉惯性里程计（中）（二）P18-19

② LinearAlignment（）PPT第五章-视觉惯性里程计（中）（二）P21

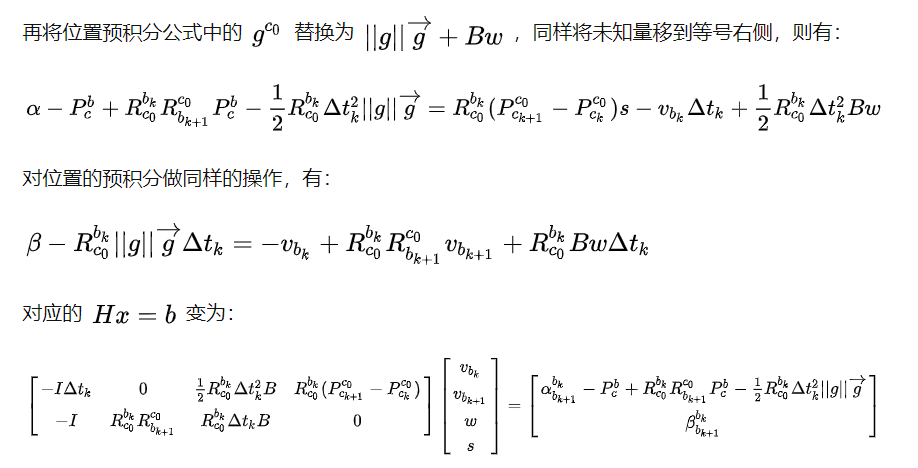






求解得到s，g

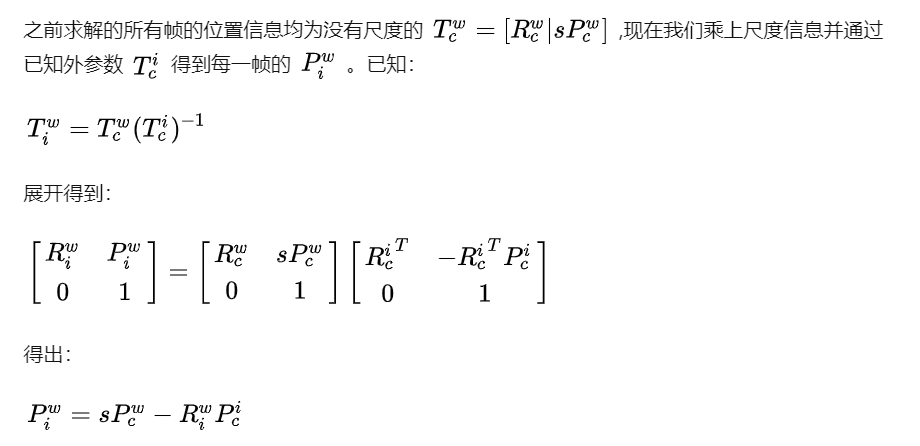
优化重力向量RefineGravity（）

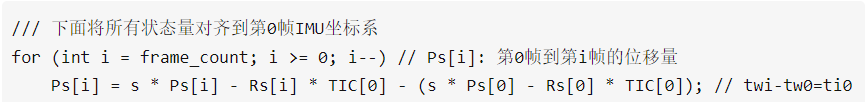




}

1. 更新状态，将初始化求出来的所有状态量对齐到第0帧IMU坐标系，同时要保证第 0帧的yaw=0。







Vw = Rwi\*Vi 将速度矢量由IMU系转到世界系

1. 重力对齐，最后再将滑窗里每一帧的位姿、速度矢量对齐到重力方向上，同时要保 证第0帧的yaw=0：

<https://blog.csdn.net/huanghaihui_123/article/details/103075107>

1. 三角化恢复深度

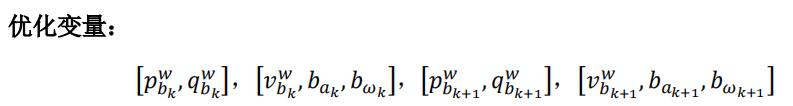
}

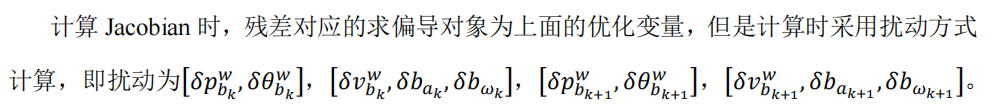
#### ④ optimization（）

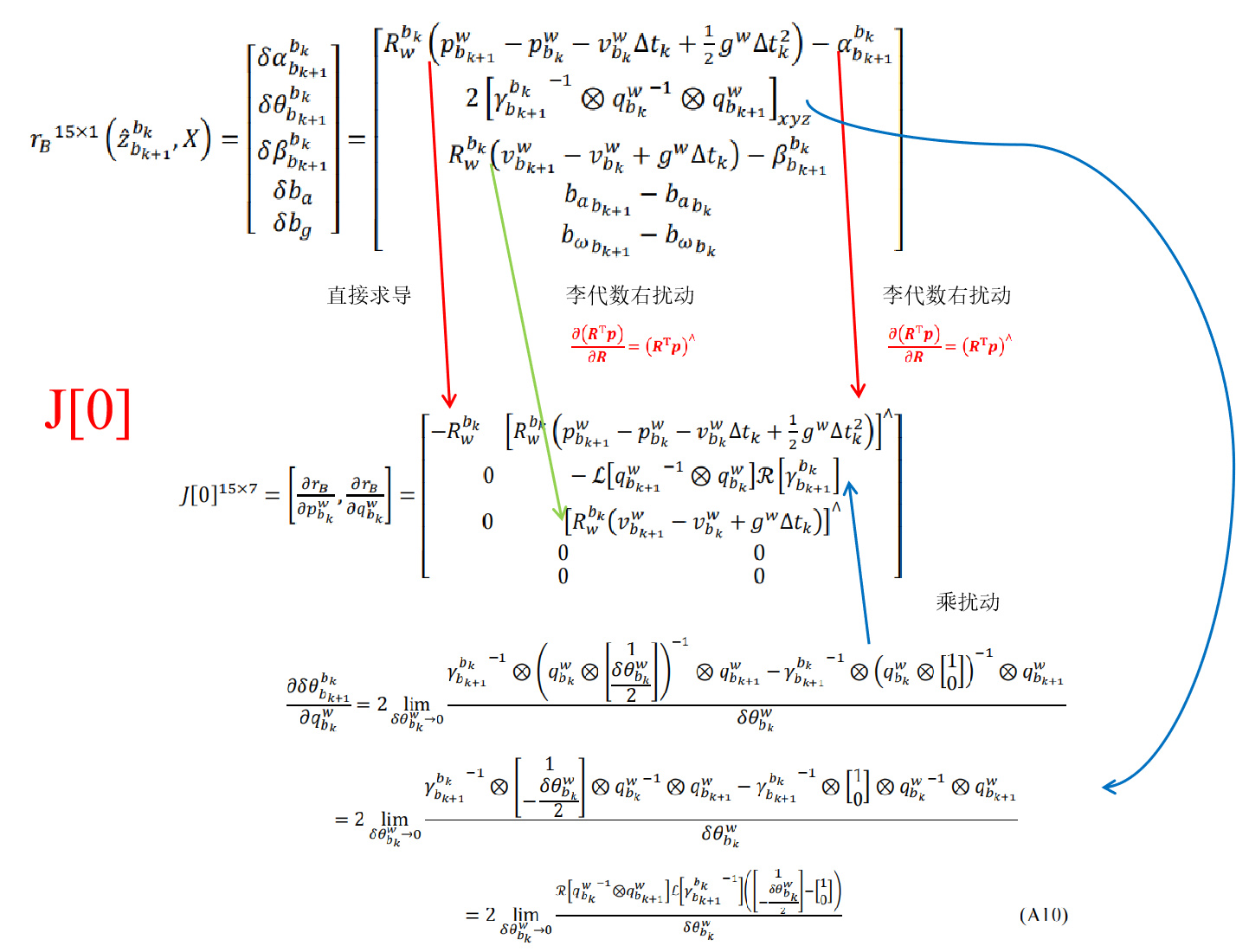
https://blog.csdn.net/weixin\_39578197/article/details/110712642?spm=1001.2101.3001.6650.3&utm\_medium=distribute.pc\_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7Edefault-3-110712642-blog-121726485.pc\_relevant\_multi\_platform\_whitelistv1&depth\_1-utm\_source=distribute.pc\_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7Edefault-3-110712642-blog-121726485.pc\_relevant\_multi\_platform\_whitelistv1&utm\_relevant\_index=6

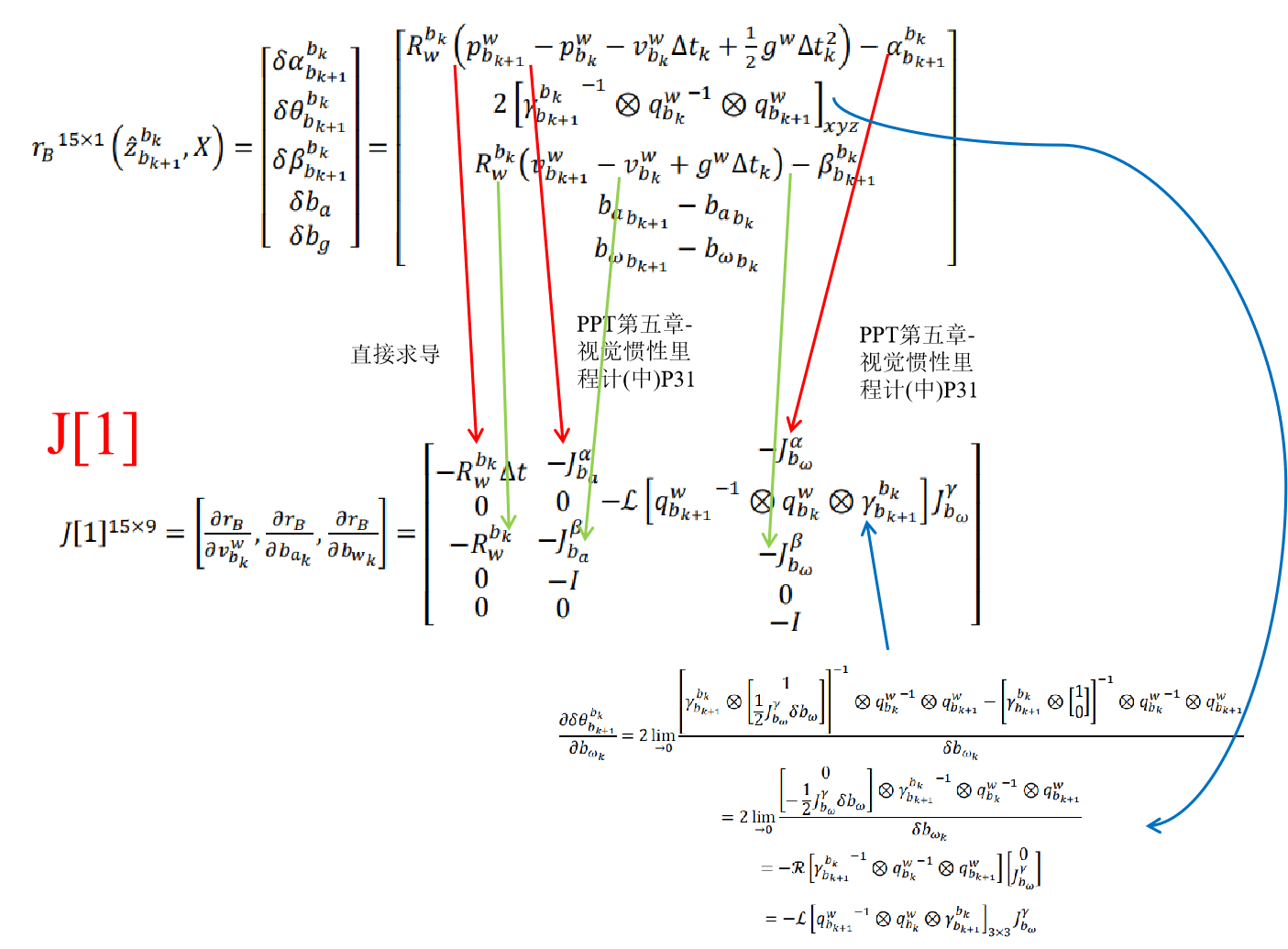
<https://www.jianshu.com/p/a9349370a8be/>

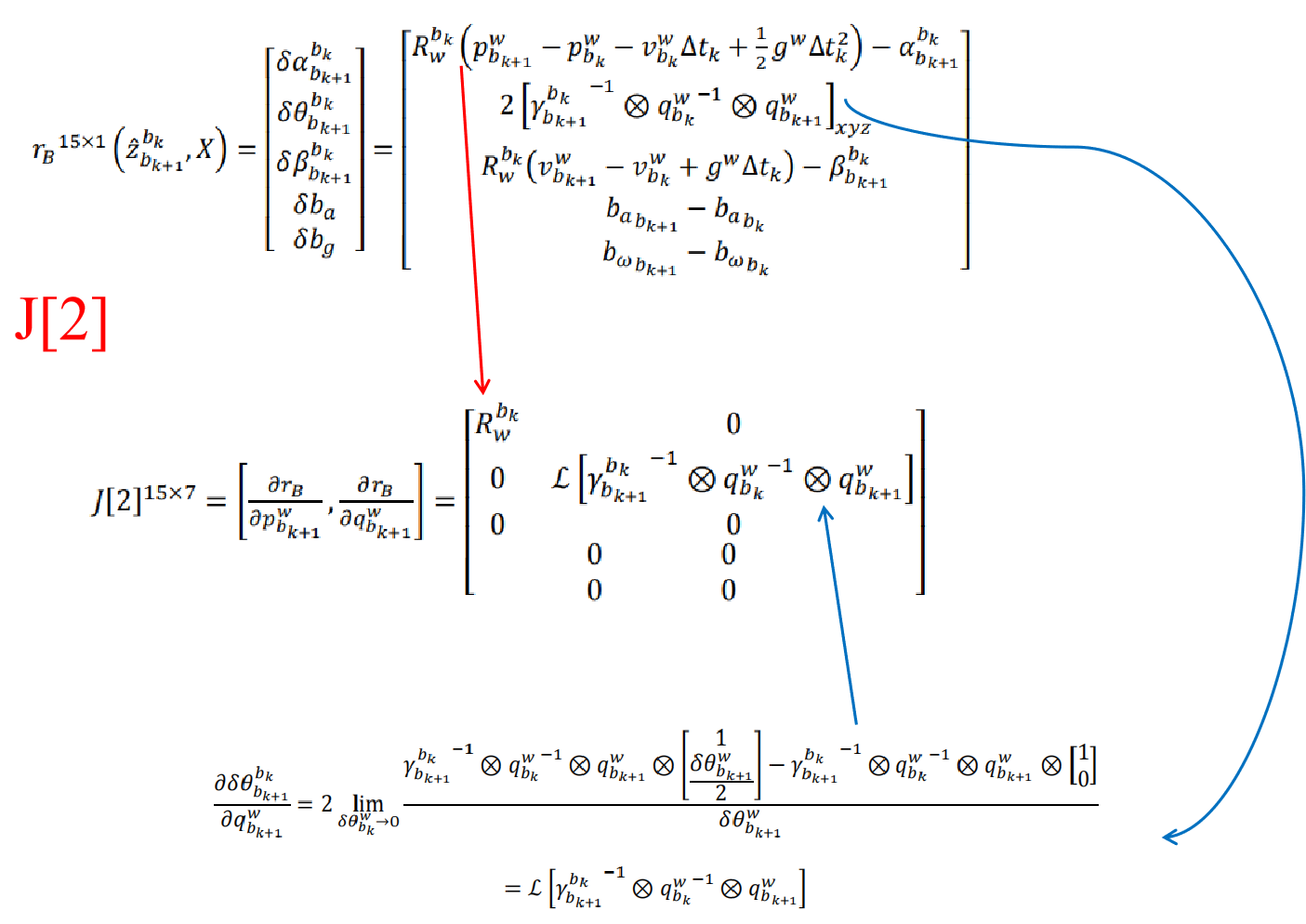
1. 添加待优化变量[p,q](7)，[speed,ba,bg](9)，添加相机与IMU的外参[p\_cb,q\_cb](7)，添加时间偏移，添加边缘化的残差
2. 添加IMU的residual。待优化变量分别为

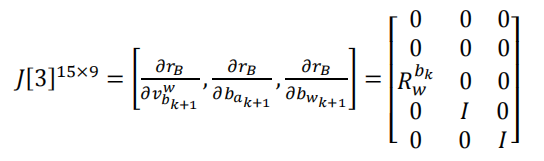




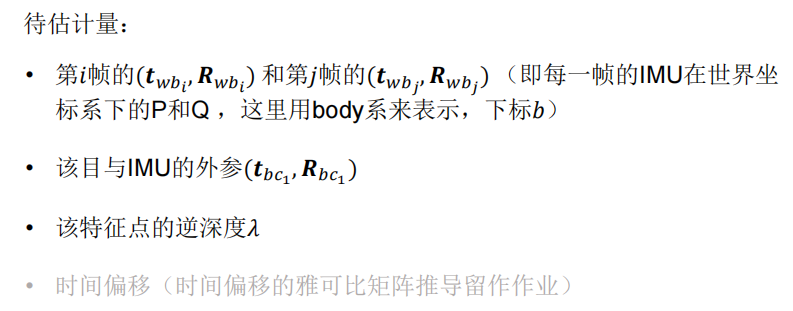


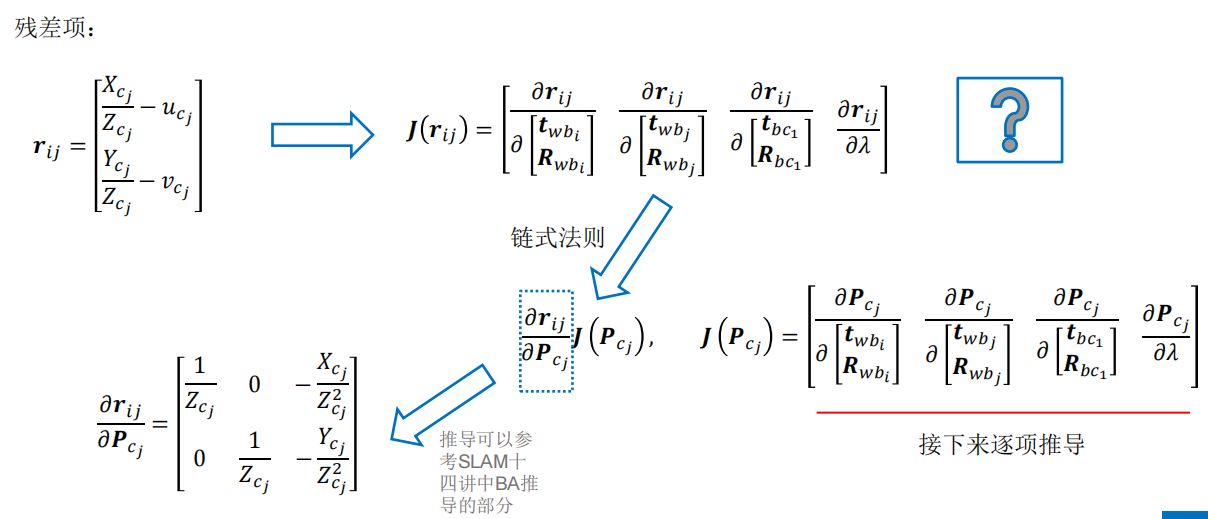


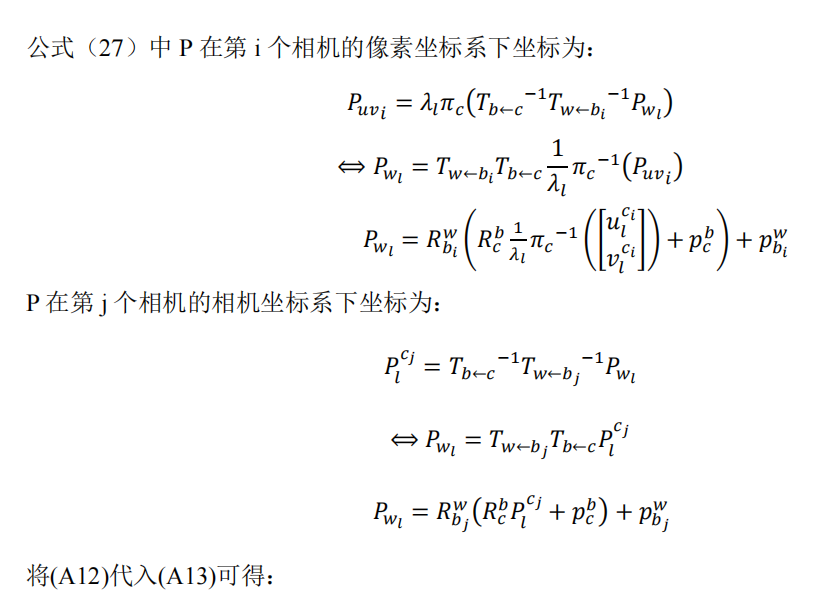


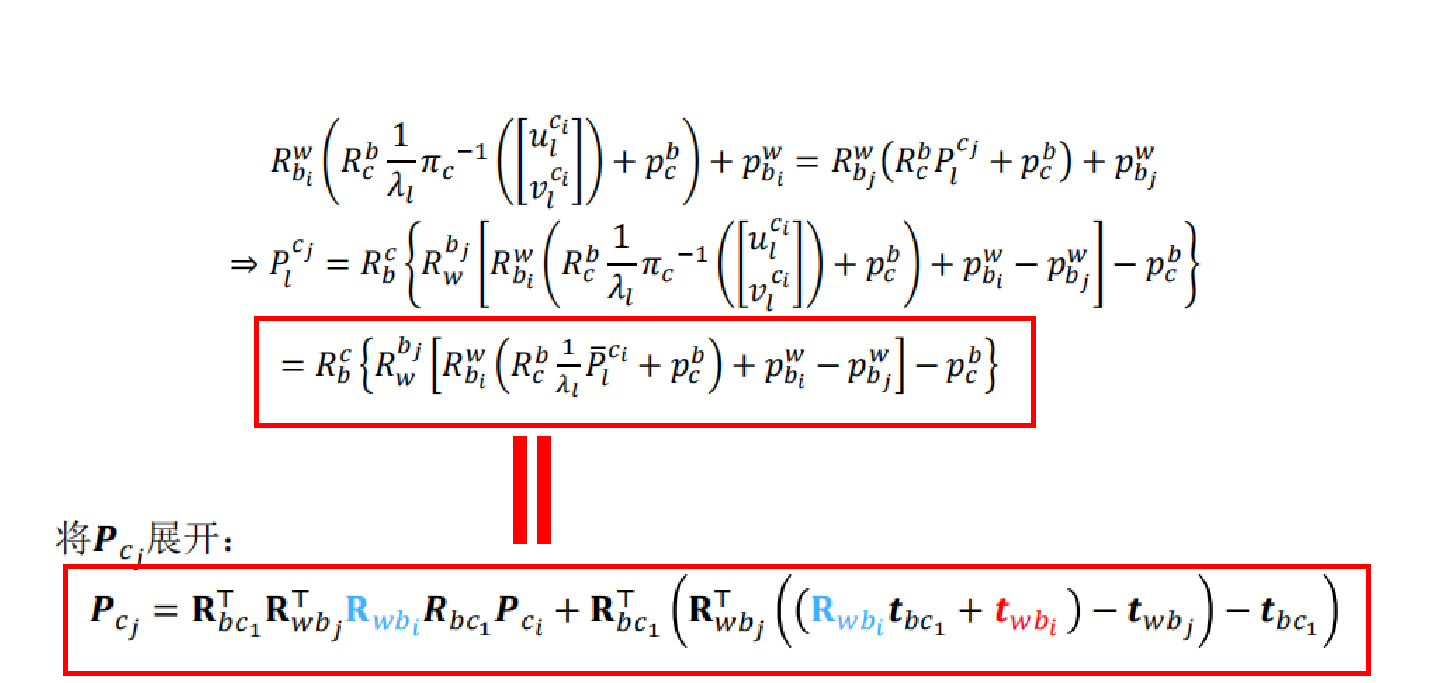


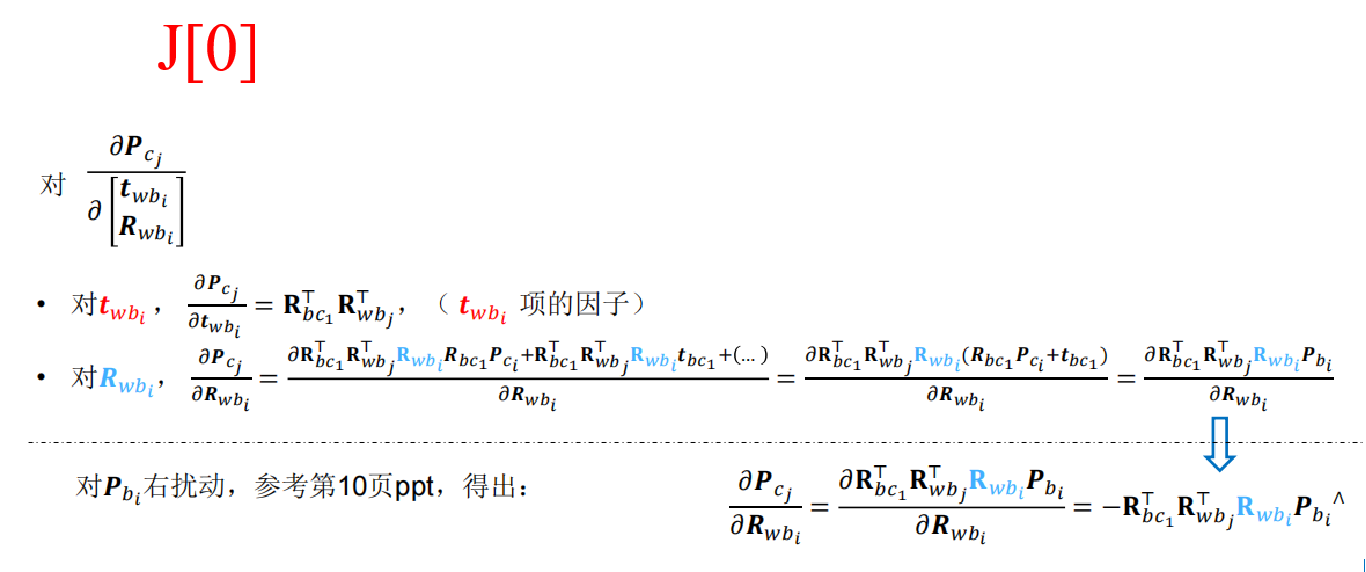
1. 添加视觉的residual（以ProjectionTwoFrameOneCamFactor为例）

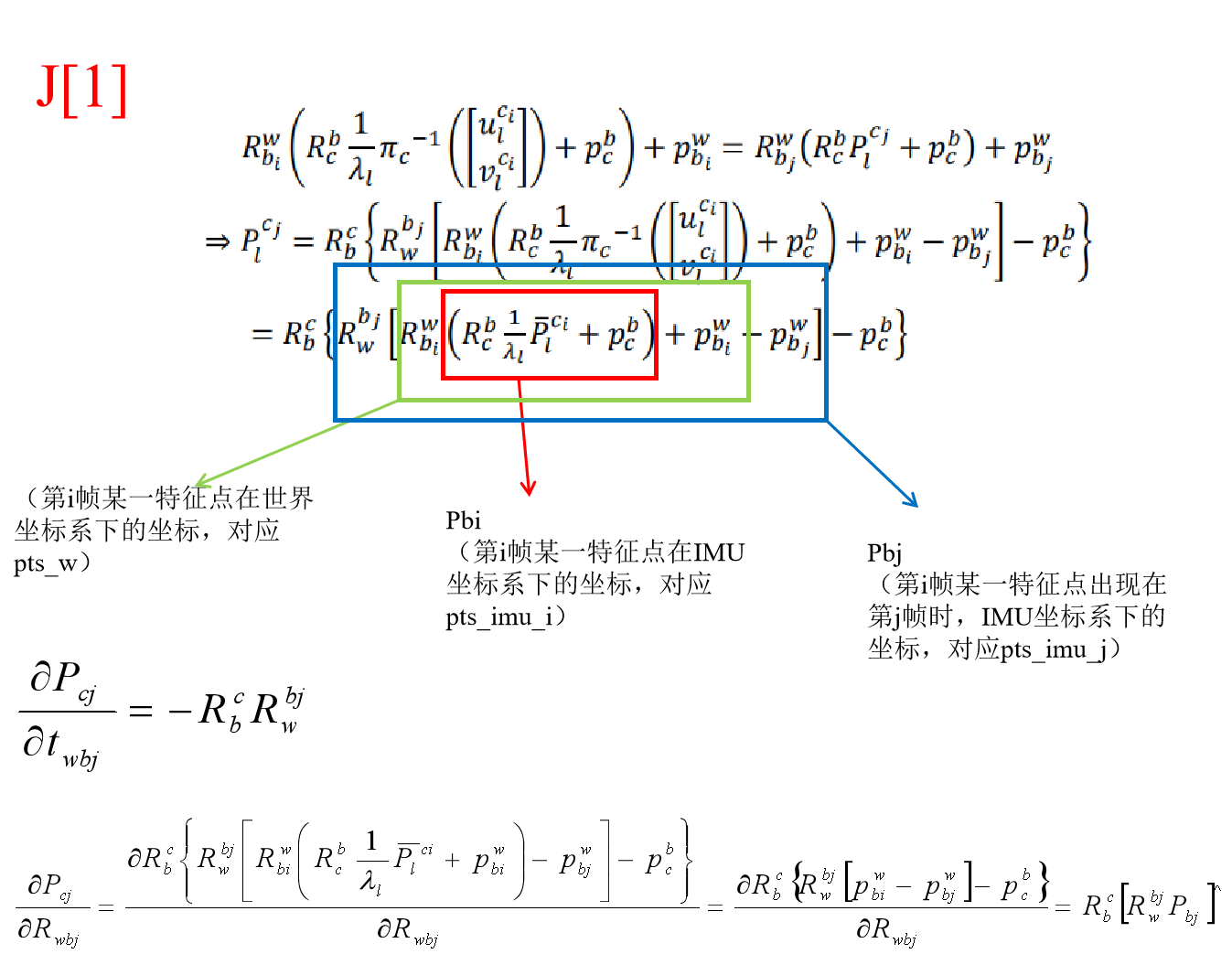


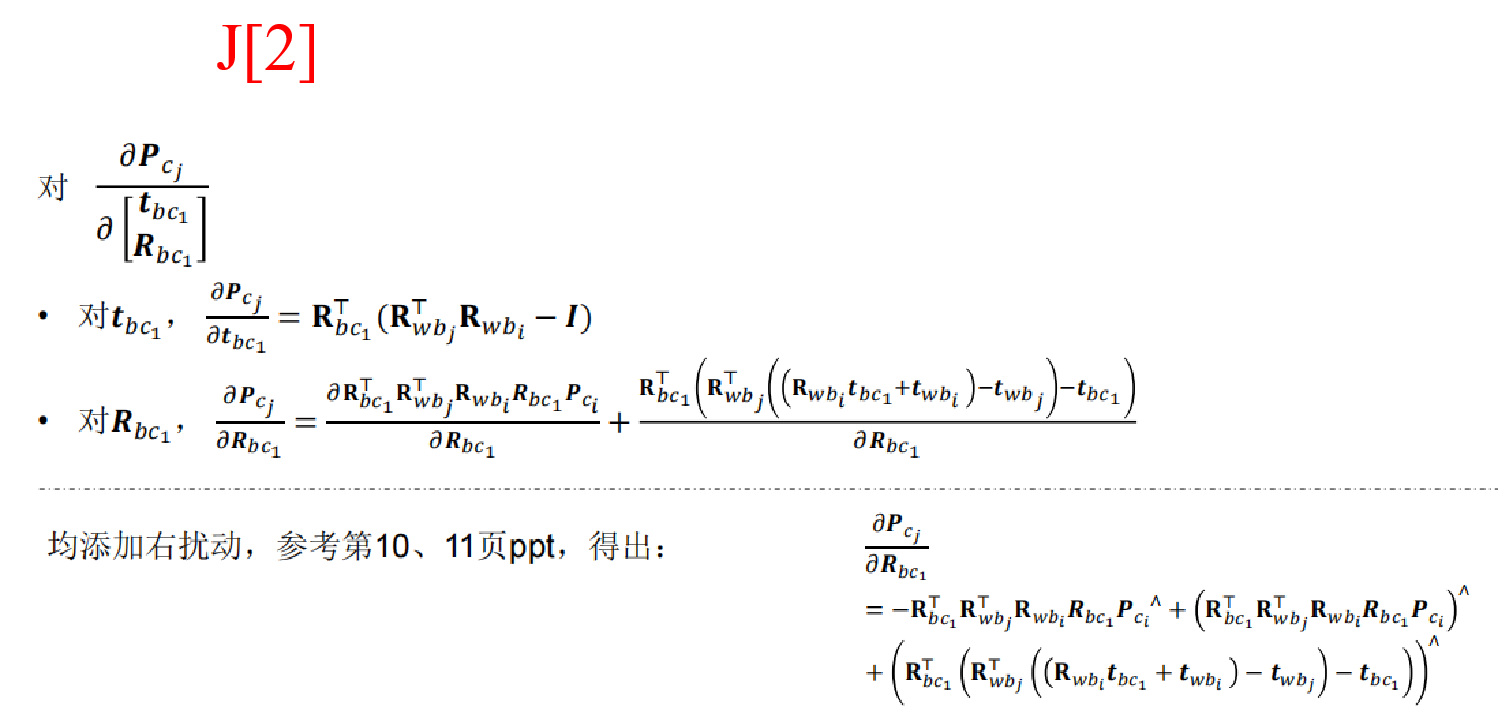


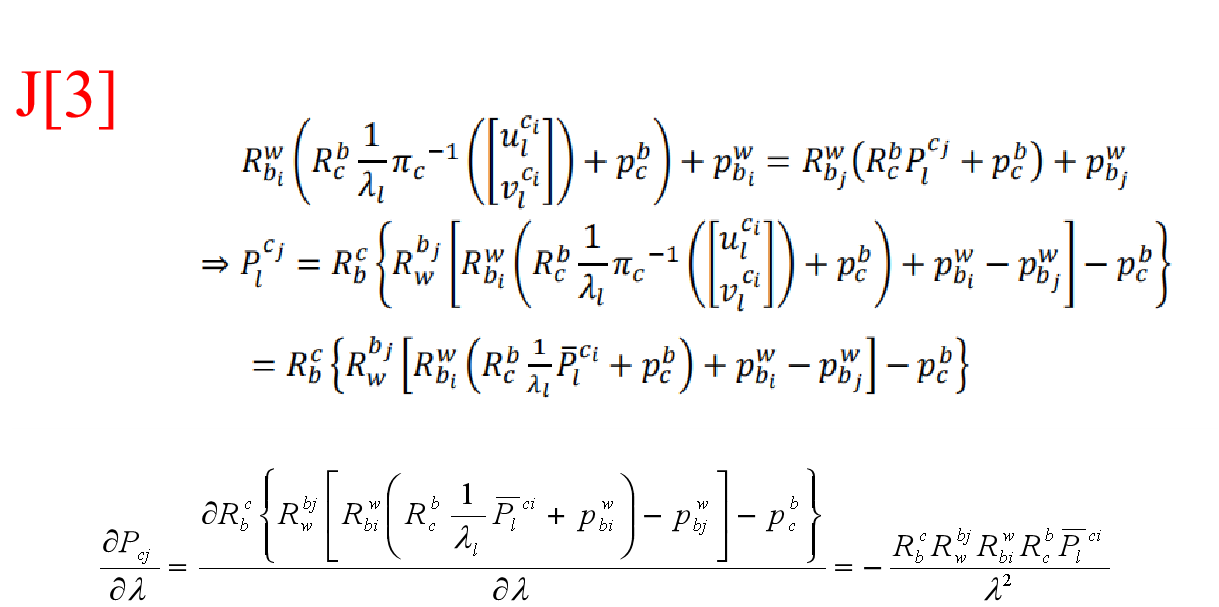


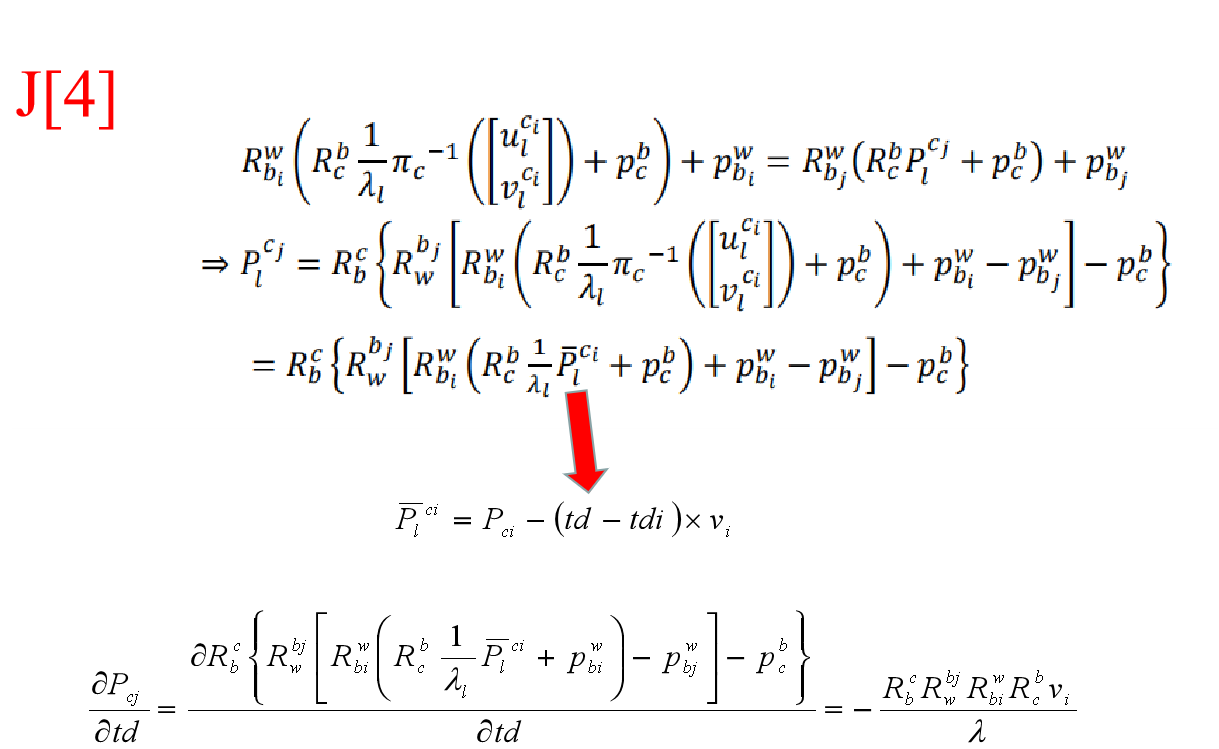












1. 添加完成待优化变量后开始进行优化
2. 开始构建边缘化残差项（边缘化最老帧），即将需要边缘化的factor放入marginalization\_info，并指出需要边缘化掉的变量：para\_Pose[0]、para\_SpeedBias[0]以及窗口内第零帧第一次观测到的特征点para\_Feature[feature\_index]。

<https://zhuanlan.zhihu.com/p/335242594>

<https://www.freesion.com/article/42241448279/>

<https://blog.csdn.net/HozenChe/article/details/125291285>

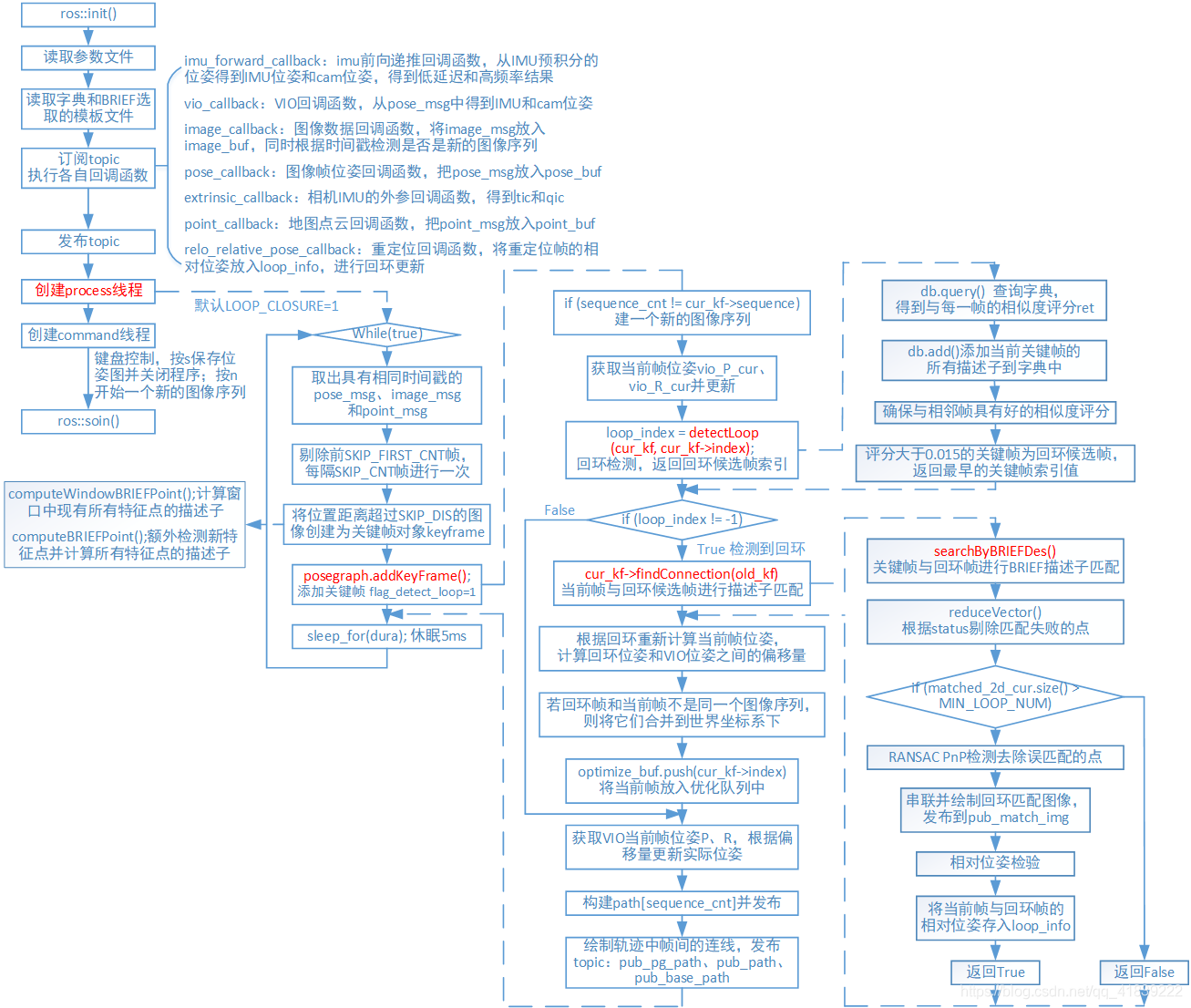
⑤ 更新状态

⑥ 滑窗slideWindow（）

}

## 回环优化

https://blog.csdn.net/xiaojinger\_123/article/details/119597747?utm\_medium=distribute.pc\_relevant.none-task-blog-2~default~baidujs\_baidulandingword~default-0-119597747-blog-87357488.pc\_relevant\_blogantidownloadv1&spm=1001.2101.3001.4242.1&utm\_relevant\_index=3



## 全局优化

https://blog.csdn.net/hltt3838/article/details/109725845?ops\_request\_misc=%257B%2522request%255Fid%2522%253A%2522165629167416782246472435%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request\_id=165629167416782246472435&biz\_id=0&utm\_medium=distribute.pc\_search\_result.none-task-blog-2~blog~first\_rank\_ecpm\_v1~rank\_v31\_ecpm-13-109725845-null-null.nonecase&utm\_term=vins&spm=1018.2226.3001.4450