VINS-Mono 详解

马朝伟

目录

1 VINS-Mono 代码目录

VINS-Mono 代码目录概述：

文件目录概述

- ar_demo
- benchmark_publisher 发布数据集中参考值
- config 硬件、系统配置文件
- camera_model
  - calib
  - camera_models 各类相机模型
  - chessboard 用于检测棋盘格特征点
  - gpl
  - sparse_graph
  - intrinsic_calib.cc 相机矫正模块主函数
- feature_trackers
  - feature_tracker_node(main()函数,ROS接受图像的回调函数)
  - feature_tracker.c 特征点跟踪的具体实现
- support_files
- vins_estimator
  - src
    - factor 实现IMU、camera等残差模型，涉及了ceres优化库，Jacobian矩阵。
    - initial 系统初始化，外参标定，SFM
    - loop_closure 闭环检测，这里主要是利用DBOW2作者的一个demo程序
    - utility 相机显示，四元数等数据转换
    - estimator_node(main()函数)
      - 多线程 measurement_process、loop_detection、 pose_graph
    - feature_manager.cpp 特征点管理，三角化，关键帧操作
    - parameters.cpp 外部系统设置参数输入

该图截取自：

https://blog.csdn.net/u012871872/article/details/78128087

2 IMU 预积分

VIO 中，如果在世界坐标系中对 IMU 进行积分，积分项中包含体坐标系相对于世界坐标系的瞬时旋转矩阵。然而，在优化位姿时，关键帧时刻体坐标系相对于世界坐标系的旋转矩阵会发生变化，那么需要对 IMU 重新进行积分。预积分就是为了避免这种重复积分。IMU 预积分将参考坐标系改为前一帧的体坐标系，从而积出了两帧之间的相对运动。后文中所有的推导中的旋转采用了右乘形式(from local to global)，使用右乘形式的好处是，可以直接使用体轴角速度在体坐标系（IMU 坐标系）下的坐标，而左乘形式需要将角速度转换到世界坐标系下。

2.1 预积分

从 IMU 获取的加速度计测量信息 a_t 和陀螺仪测量信息 w_t：

$$a_t = a_{t(real)} + b_{a_t} + R_w^t g^w + n_a \tag{1}$$

$$w_t = w_{t(real)} + b_{w_t} + n_w \tag{2}$$

我觉得等式(1)中 $R_w^t g^w$ 之前的符号应该是减号，而不是加号。是否可能是这样，vins-mono 的论文中的 g^w 的方向与实际重力方向相反，是从地指向天，而不是我们习惯上以为的指向地心？？？？？

IMU 的测量信息是在本体坐标系（即 IMU 坐标）中获取的。理论上真实的加速度 $a_{t(real)}$ 受加速度偏置 b_{a_t} 和噪声 n_a 的影响，它们之和构成了最终的加速度测量值 a_t。理论上体轴的角速度 $w_{t(real)}$ 受陀螺仪偏置 b_{w_t} 和噪声 n_w 的影响，它们之和构成了最终的角速度测量值 w_t。

假设噪声 n_a 和 n_w 服从高斯分布：

$$\begin{aligned} n_a &\sim N(0, \sigma_a^2) \\ n_w &\sim N(0, \sigma_w^2) \end{aligned} \tag{3}$$

加速度偏置 b_{a_t} 和陀螺仪偏置 b_{w_t} 被建模为随机游走：

$$\dot{b}_{a_t} = n_{b_a}$$
$$\dot{b}_{w_t} = n_{b_w} \tag{4}$$

其中 n_{b_a} 和 n_{b_w} 服从高斯分布：

$$n_{b_a} \sim N(0, \sigma_{b_a}^2)$$
$$n_{b_w} \sim N(0, \sigma_{b_w}^2) \tag{5}$$

对于连续两个关键帧 b_k 和 b_{k+1}，它们对应的时刻分别为 t_k、t_{k+1}。可以根据 $[t_k, t_{k+1}]$ 时间间隔内 IMU 的测量值，对系统的位置、速度和旋转等状态进行传播（注意这里的四元数采用了实部在后，虚部在前的形式）：

$$p_{b_{k+1}}^w = p_{b_k}^w + v_{b_k}^w \Delta t_k + \iint_{t \in [t_k, t_{k+1}]} (R_t^w (a_t - b_{a_t} - n_a) - g^w) dt^2 \tag{6}$$

$$v_{b_{k+1}}^w = v_{b_k}^w + \int_{t \in [t_k, t_{k+1}]} (R_t^w (a_t - b_{a_t} - n_a) - g^w) dt \tag{7}$$

$$q_{b_{k+1}}^w = q_{b_k}^w \otimes \int_{t \in [t_k, t_{k+1}]} \frac{1}{2} q_t^{b_k} \otimes (w_t - b_{w_t} - n_w) dt$$
$$= q_{b_k}^w \otimes \int_{t \in [t_k, t_{k+1}]} \frac{1}{2} \Omega(w_t - b_{w_t} - n_w) q_t^{b_k} dt \tag{8}$$

其中：

$$\Omega(w) = \begin{bmatrix} -[w]_\times & w \\ -w^T & 0 \end{bmatrix}, [w]_\times = \begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix} \tag{9}$$

Δt_k 是 $[t_k, t_{k+1}]$ 之间的时间间隔，R_t^w 为 t 时刻从本体坐标系到世界坐标系的旋转矩阵，$q_t^{b_k}$ 为用四元数表示的 t 时刻从本体坐标系到世界坐标系的旋转。

从等式(6)(7)(8)可以看出，系统位置、速度和旋转等状态的传播需要关键帧 b_k 时刻的位置 $p_{b_k}^w$、速度 $v_{b_k}^w$ 和旋转 $q_{b_k}^w$，当这些起始状态发生改变时，就需要按照等式(6)(7)(8)重新进行状态传播。在基于优化的算法中，每个关键帧时刻的状态需要频繁调整，所以就需要频繁地重新积分，这样会浪费大量的计算资源。IMU 预积分就是为了避免这种计算资源上的浪费。

IMU 预积分的思路简单来说，就是将参考坐标系从世界坐标系 w 调整为第 k 个关键帧时刻的本体坐标系 b_k：

$$R_w^{b_k} p_{b_{k+1}}^w = R_w^{b_k}(p_{b_k}^w + v_{b_k}^w \Delta t_k - \frac{1}{2}g^w \Delta t_k^2) + \alpha_{b_{k+1}}^{b_k} \tag{10}$$

$$R_w^{b_k} v_{b_{k+1}}^w = R_w^{b_k}(v_{b_k}^w - g^w \Delta t_k) + \beta_{b_{k+1}}^{b_k} \tag{11}$$

$$q_w^{b_k} \otimes q_{b_{k+1}}^w = \gamma_{b_{k+1}}^{b_k} \tag{12}$$

其中：

$$\alpha_{b_{k+1}}^{b_k} = \iint_{t \in [t_k, t_{k+1}]} (R_t^{b_k}(a_t - b_{a_t} - n_a))dt^2 \tag{13}$$

$$\beta_{b_{k+1}}^{b_k} = \int_{t \in [t_k, t_{k+1}]} (R_t^{b_k}(a_t - b_{a_t} - n_a))dt \tag{14}$$

$$\gamma_{b_{k+1}}^{b_k} = \int_{t \in [t_k, t_{k+1}]} \frac{1}{2}\Omega(w_t - b_{w_t} - n_w)\gamma_t^{b_k}dt \tag{15}$$

积分项(13)(14)(15)中的参考坐标系变成了 b_k，可以理解为这时的积分结果为 b_{k+1} 对于 b_k 的相对运动量，即使在优化过程中对关键帧的位置、速度和旋转等状态进行调整，也不对积分项(13)(14)(15)产生任何影响，从而避免了重复积分。

2.2 连续时间状态方程

这一部分的内容可以参考 sola 的文章[1]。

用 $\alpha_t^{b_k}$、$\beta_t^{b_k}$、$\gamma_t^{b_k}$、b_{a_t} 和 b_{w_t} 表示系统真实状态，它们都是去除噪声后的理想状态，真实状态可以分解为标称状态和误差状态的组合。标称状态代表大信号，未去除噪声，用 $\hat{\alpha}_t^{b_k}$、$\hat{\beta}_t^{b_k}$、$\hat{\gamma}_t^{b_k}$、$\hat{b}_{a_t}$ 和 $\hat{b}_{w_t}$ 表示。误差状态表示小信号，用 $\delta\alpha_t^{b_k}$、$\delta\beta_t^{b_k}$、$\delta\theta_t^{b_k}$、δb_{a_t} 和 δb_{w_t} 表示，噪声带来的误差累积由误差状态的状态方程来描述。真实状态与标称状态、误差状态之间的关系由公式(16)(17)(18)(19)(20)描述。

$$\alpha_t^{b_k} = \hat{\alpha}_t^{b_k} + \delta\alpha_t^{b_k} \tag{16}$$

$$\beta_t^{b_k} = \hat{\beta}_t^{b_k} + \delta\beta_t^{b_k} \tag{17}$$

$$\begin{aligned} \gamma_t^{b_k} &= \hat{\gamma}_t^{b_k} \otimes \delta\gamma_t^{b_k} \\ &= \hat{\gamma}_t^{b_k} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}\delta\theta_t^{b_k} \end{bmatrix} \end{aligned} \tag{18}$$

$$b_{a_t} = \hat{b}_{a_t} + \delta b_{a_t} \tag{19}$$

$$b_{w_t} = \hat{b}_{w_t} + \delta b_{w_t} \tag{20}$$

2.2.1 真实状态(true-state)的状态方程

$$\dot{\alpha}_t^{b_k} = \beta_t^{b_k} \tag{21}$$

$$\dot{\beta}_t^{b_k} = R_t^{b_k}(a_t - b_{a_t} - n_a) \tag{22}$$

$$\dot{\gamma}_t^{b_k} = \frac{1}{2}\gamma_t^{b_k} \otimes (w_t - b_{w_t} - n_w)$$
$$= \frac{1}{2}\Omega(w_t - b_{w_t} - n_w)\gamma_t^{b_k} \tag{23}$$

$$\dot{b}_{a_t} = n_{b_a} \tag{24}$$

$$\dot{b}_{w_t} = n_{b_w} \tag{25}$$

2.2.2 标称状态(nominal-state)的状态方程

$$\dot{\hat{\alpha}}_t^{b_k} = \dot{\hat{\beta}}_t^{b_k} \tag{26}$$

$$\dot{\hat{\beta}}_t^{b_k} = \hat{R}_t^{b_k}(a_t - \hat{b}_{a_t}) \tag{27}$$

$$\dot{\hat{\gamma}}_t^{b_k} = \frac{1}{2}\hat{\gamma}_t^{b_k} \otimes (w_t - \hat{b}_{w_t})$$
$$= \frac{1}{2}\Omega(w_t - \hat{b}_{w_t})\hat{\gamma}_t^{b_k} \tag{28}$$

$$\dot{\hat{b}}_{a_t} = 0 \tag{29}$$

$$\dot{\hat{b}}_{w_t} = 0 \tag{30}$$

等式(29)(30)实际上表明了，在实际使用 IMU 信息进行积分时，加速度计和陀螺仪的偏置一般是保持不变的，只有在得到优化时，它们的值才会发生改变。

2.2.3 误差状态(error-state)的状态方程

先给出误差状态的状态方程，后面再逐一推导：

$$\delta\dot{\alpha}_t^{b_k} = \delta\beta_t^{b_k} \tag{31}$$

$$\delta\dot{\beta}_t^{b_k} = -\hat{R}_t^{b_k}\left[a_t - \hat{b}_{a_t}\right]_\times \delta\theta_t^{b_k} - \hat{R}_t^{b_k}\delta b_{a_t} - \hat{R}_t^{b_k}n_a \tag{32}$$

$$\delta\dot{\theta}_t^{b_k} = -\left[w_t - \hat{b}_{w_t}\right]_\times \delta\theta_t^{b_k} - \delta b_{w_t} - n_w \tag{33}$$

$$\delta\dot{b}_{a_t} = n_{b_a} \tag{34}$$

$$\delta\dot{b}_{w_t} = n_{b_w} \tag{35}$$

将误差状态的状态方程写成矩阵形式：

$$\begin{bmatrix} \delta\dot{\alpha}_t^{b_k} \\ \delta\dot{\beta}_t^{b_k} \\ \delta\dot{\theta}_t^{b_k} \\ \delta\dot{b}_{a_t} \\ \delta\dot{b}_{w_t} \end{bmatrix} = \begin{bmatrix} 0 & I & 0 & 0 & 0 \\ 0 & 0 & -\hat{R}_t^{b_k}\left[a_t - \hat{b}_{a_t}\right]_\times & -\hat{R}_t^{b_k} & 0 \\ 0 & 0 & -\left[w_t - \hat{b}_{w_t}\right]_\times & 0 & -I \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta\alpha_t^{b_k} \\ \delta\beta_t^{b_k} \\ \delta\theta_t^{b_k} \\ \delta b_{a_t} \\ \delta b_{w_t} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ -\hat{R}_t^{b_k} & 0 & 0 & 0 \\ 0 & -I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix} \begin{bmatrix} n_a \\ n_w \\ n_{b_a} \\ n_{b_w} \end{bmatrix} \tag{36}$$

$$= F_t \delta z_t^{b_k} + G_t n_t$$

公式(31)(34)(35)是显而易见的，因为这些公式都是线性关系。公式(31)可由公式(21)与公式(26)做差得到，公式(34)可由公式(24)与公式(29)做差得到，公式(35)可由公式(25)和公式(30)做差得到。公式(32)(33)的推导稍微复杂。

先进行等式(32)的推导：

$$R_t^{b_k} = \hat{R}_t^{b_k}(I + \left[\delta\theta_t^{b_k}\right]_\times) + O(\|\delta\theta_t^{b_k}\|^2) \tag{37}$$

为了表述上的方便，定义：

$$a = a_t - b_{a_t} - n_a \tag{38}$$

$$\hat{a} = a_t - \hat{b}_{a_t} \tag{39}$$

a 和 $\hat{a}$ 存在以下关系：

$$a = \hat{a} + \delta a \tag{40}$$

可以得到：

$$\begin{aligned} \delta a &= a - \hat{a} \\ &= (a_t - b_{a_t} - n_a) - (a_t - \hat{b}_{a_t}) \\ &= -(b_{a_t} - \hat{b}_{a_t}) - n_a \\ &= -\delta b_{a_t} - n_a \end{aligned} \tag{41}$$

将等式(39)代入等式(27)：

$$\dot{\hat{\beta}}_t^{b_k} = \hat{R}_t^{b_k}\hat{a} \tag{42}$$

根据(40)将等式(22)中等号右边括号内的项替换为 $\hat{a}+\delta a$：

$$\dot{\beta}_t^{b_k} = R_t^{b_k}(\hat{a}+\delta a) \tag{43}$$

将等式(37)代入等式(43)，并忽略 $\delta\theta_t^{b_k}$ 的二阶及二阶以上的项：

$$\begin{aligned}
\dot{\beta}_t^{b_k} &= \hat{R}_t^{b_k}(I+\left[\delta\theta_t^{b_k}\right]_\times)(\hat{a}+\delta a) \\
&= \hat{R}_t^{b_k}\hat{a} + \hat{R}_t^{b_k}\delta a + \hat{R}_t^{b_k}\left[\delta\theta_t^{b_k}\right]_\times\hat{a} + \hat{R}_t^{b_k}\left[\delta\theta_t^{b_k}\right]_\times\delta a
\end{aligned} \tag{44}$$

根据等式(17)两边求导可得：

$$\dot{\beta}_t^{b_k} = \dot{\hat{\beta}}_t^{b_k} + \delta\dot{\beta}_t^{b_k} \tag{45}$$

由将等式(42)等式(45)可得：

$$\begin{aligned}
\dot{\beta}_t^{b_k} &= \dot{\hat{\beta}}_t^{b_k} + \delta\dot{\beta}_t^{b_k} \\
&= \hat{R}_t^{b_k}\hat{a} + \delta\dot{\beta}_t^{b_k}
\end{aligned} \tag{46}$$

将等式(44)与等式(46)联立：

$$\hat{R}_t^{b_k}\hat{a} + \delta\dot{\beta}_t^{b_k} = \hat{R}_t^{b_k}\hat{a} + \hat{R}_t^{b_k}\delta a + \hat{R}_t^{b_k}\left[\delta\theta_t^{b_k}\right]_\times\hat{a} + \hat{R}_t^{b_k}\left[\delta\theta_t^{b_k}\right]_\times\delta a \tag{47}$$

化简上式得：

$$\delta\dot{\beta}_t^{b_k} = \hat{R}_t^{b_k}\delta a + \hat{R}_t^{b_k}\left[\delta\theta_t^{b_k}\right]_\times\hat{a} + \hat{R}_t^{b_k}\left[\delta\theta_t^{b_k}\right]_\times\delta a \tag{48}$$

忽略等式(48)中的二阶小量，并且调整叉乘项的顺序（$[a]_\times b = -[b]_\times a$）：

$$\delta\dot{\beta}_t^{b_k} = \hat{R}_t^{b_k}\delta a - \hat{R}_t^{b_k}\left[\hat{a}\right]_\times\delta\theta_t^{b_k} \tag{49}$$

再将等式(39)(41)代入等式(49)：

$$\begin{aligned}
\delta\dot{\beta}_t^{b_k} &= \hat{R}_t^{b_k}\delta a - \hat{R}_t^{b_k}\left[\hat{a}\right]_\times\delta\theta_t^{b_k} \\
&= -\hat{R}_t^{b_k}[a_t - \hat{b}_{a_t}]_\times\delta\theta_t^{b_k} - \hat{R}_t^{b_k}\delta b_{a_t} - \hat{R}_t^{b_k}n_a
\end{aligned} \tag{50}$$

至此，等式(32)推导完毕。

下面推导等式(33)：

为了表述上的方便，定义：

$$w = w_t - b_{w_t} - n_w \tag{51}$$

$$\hat{w} = w_t - \hat{b}_{w_t} \tag{52}$$

w 和 $\hat{w}$ 存在以下关系：

$$w = \hat{w} + \delta w \tag{53}$$

可以得到：

$$\begin{aligned}
\delta w &= w - \hat{w} \\
&= -(b_{w_t} - \hat{b}_{w_t}) - n_w \\
&= -\delta b_{w_t} - n_w
\end{aligned} \tag{54}$$

对等式(18)两边求导：

$$\dot{\gamma}_t^{b_k} = \dot{\hat{\gamma}}_t^{b_k} \otimes \delta\gamma_t^{b_k} + \hat{\gamma}_t^{b_k} \otimes \delta\dot{\gamma}_t^{b_k} \tag{55}$$

将等式(28)代入等式(55)：

$$\dot{\gamma}_t^{b_k} = \frac{1}{2} \hat{\gamma}_t^{b_k} \otimes \hat{w} \otimes \delta\gamma_t^{b_k} + \hat{\gamma}_t^{b_k} \otimes \delta\dot{\gamma}_t^{b_k} \tag{56}$$

将等式(18)代入等式(23)：

$$\dot{\gamma}_t^{b_k} = \frac{1}{2} \hat{\gamma}_t^{b_k} \otimes \delta\gamma_t^{b_k} \otimes w \tag{57}$$

联立等式(56)(57)得：

$$\frac{1}{2} \hat{\gamma}_t^{b_k} \otimes \delta\gamma_t^{b_k} \otimes w = \frac{1}{2} \hat{\gamma}_t^{b_k} \otimes \hat{w} \otimes \delta\gamma_t^{b_k} + \hat{\gamma}_t^{b_k} \otimes \delta\dot{\gamma}_t^{b_k} \tag{58}$$

化简等式(58)得：

$$\delta\dot{\gamma}_t^{b_k} = \frac{1}{2} \delta\gamma_t^{b_k} \otimes w - \frac{1}{2} \hat{w} \otimes \delta\gamma_t^{b_k} \tag{59}$$

由等式(18)可知：

$$\delta\gamma_t^{b_k} = \begin{bmatrix} 1 \\ \frac{1}{2}\delta\theta_t^{b_k} \end{bmatrix} \tag{60}$$

等式(60)实际上是对旋转误差的重新参数化，旋转误差本质上只有 3 个自由度，但是以四元数形式表示的误差 $\delta\gamma_t^{b_k}$ 有 4 个自由度，所以将其重新参数化为 $\delta\theta_t^{b_k}$，它们之间的转换关系由等式(60)给出。实际上，等式(60)中的等号应该是近似相等，等式右边仅保留了 $\delta\theta_t^{b_k}$ 的一阶项，高阶项被忽略，但是由于旋转误差是小量，所以仅仅保留一阶项可以使该等式近似成立。

对等式(60)等号两边同时求导：

$$\delta\ddot{\gamma}_t^{b_k} = \begin{bmatrix} 0 \\ \dfrac{1}{2}\delta\dot{\theta}_t^{b_k} \end{bmatrix} \tag{61}$$

等式(61)可改写为：

$$\begin{bmatrix} 0 \\ \delta\dot{\theta}_t^{b_k} \end{bmatrix} = 2\delta\ddot{\gamma}_t^{b_k} \tag{62}$$

将等式(59)代入等式(62)得：

$$\begin{aligned} \begin{bmatrix} 0 \\ \delta\dot{\theta}_t^{b_k} \end{bmatrix} &= 2\delta\ddot{\gamma}_t^{b_k} \\ &= \delta\gamma_t^{b_k} \otimes w - \hat{w} \otimes \delta\gamma_t^{b_k} \\ &= \begin{bmatrix} 0 & -w^T \\ w & -[w]_\times \end{bmatrix}\delta\gamma_t^{b_k} - \begin{bmatrix} 0 & -\hat{w}^T \\ \hat{w} & [\hat{w}]_\times \end{bmatrix}\delta\gamma_t^{b_k} \\ &= \begin{bmatrix} 0 & -(w^T - \hat{w}^T) \\ w - \hat{w} & -[w + \hat{w}]_\times \end{bmatrix}\delta\gamma_t^{b_k} \end{aligned} \tag{63}$$

将等式(60)代入等式(63)得：

$$\begin{bmatrix} 0 \\ \delta\dot{\theta}_t^{b_k} \end{bmatrix} = \begin{bmatrix} 0 & -(w^T - \hat{w}^T) \\ w - \hat{w} & -[w + \hat{w}]_\times \end{bmatrix}\begin{bmatrix} 1 \\ \dfrac{1}{2}\delta\theta_t^{b_k} \end{bmatrix} \tag{64}$$

再将等式(53)代入等式(64)得：

$$\begin{bmatrix} 0 \\ \delta\dot{\theta}_t^{b_k} \end{bmatrix} = \begin{bmatrix} 0 & -\delta w \\ \delta w & -[2\hat{w} + \delta w]_\times \end{bmatrix}\begin{bmatrix} 1 \\ \dfrac{1}{2}\delta\theta_t^{b_k} \end{bmatrix} \tag{65}$$

将等式(65)拆分为两个等式：

$$0 = --\delta w\frac{1}{2}\delta\theta_t^{b_k} \tag{66}$$

$$\delta\dot{\theta}_t^{b_k} = \delta w - [\hat{w}]_\times\delta\theta_t^{b_k} - \frac{1}{2}[\delta w]_\times\delta\theta_t^{b_k} \tag{67}$$

等式(66)没什么用处。将等式(67)中的二阶小量略去得：

$$\delta\dot{\theta}_t^{b_k} = \delta w - [\hat{w}]_\times\delta\theta_t^{b_k} \tag{68}$$

将等式(52)(54)代入等式(68)得：

$$\delta\dot{\theta}_t^{b_k} = -[w_t - \hat{b}_{w_t}]_\times\delta\theta_t^{b_k} - \delta b_{w_t} - n_w \tag{69}$$

至此，等式(33)推导完毕。

2.3 中值积分法下的预积分方程

2.3.1 标称状态

（似乎并不是严格意义上的中值积分，不知道为什么，但是大家都这么推）

$$\hat{\alpha}_{i+1}^{b_k} = \hat{\alpha}_i^{b_k} + \hat{\beta}_i^{b_k}\delta t + \frac{1}{2}\frac{R(\hat{\gamma}_i^{b_k})(w_i - \hat{b}_{w_i}) + R(\hat{\gamma}_{i+1}^{b_k})(w_{i+1} - \hat{b}_{w_{i+1}})}{2}\delta t^2 \tag{70}$$

$$\hat{\beta}_{i+1}^{b_k} = \hat{\beta}_i^{b_k} + \frac{R(\hat{\gamma}_i^{b_k})(w_i - \hat{b}_{w_i}) + R(\hat{\gamma}_i^{b_k})(w_{i+1} - \hat{b}_{w_{i+1}})}{2}\delta t \tag{71}$$

$$\hat{\gamma}_{i+1}^{b_k} = \hat{\gamma}_i^{b_k} \otimes \begin{bmatrix} 1 \\ \dfrac{1}{2}\dfrac{(w_i - \hat{b}_{w_i}) + (w_{i+1} - \hat{b}_{w_{i+1}})}{2}\delta t \end{bmatrix} \tag{72}$$

$$\hat{b}_{a_{i+1}} = \hat{b}_{a_i} \tag{73}$$

$$\hat{b}_{w_{i+1}} = \hat{b}_{w_i} \tag{74}$$

2.3.2 误差状态

中值积分方式下：

$$\delta\theta_{i+1}^{b_k} = \delta\theta_i^{b_k} + \delta\dot{\theta}_{i+\frac{1}{2}}^{b_k}\delta t \tag{75}$$

由等式(33)可得：（我觉得等式(76)并不是严格意义上中值积分法下中点处的导数，而是其近似）

$$\delta\dot{\theta}_{i+\frac{1}{2}}^{b_k} = -\left[\frac{(w_i - \hat{b}_{w_i}) + (w_{i+1} - \hat{b}_{w_{i+1}})}{2}\right]_\times \delta\theta_i^{b_k} - \frac{\delta b_{w_i} + \delta b_{w_{i+1}}}{2} - \frac{n_{w_0} + n_{w_1}}{2} \tag{76}$$

由等式(30)(35)可得：

$$\hat{b}_{w_{i+1}} = \hat{b}_{w_i} \tag{77}$$

$$\delta b_{w_{i+1}} = \delta b_{w_i} \tag{78}$$

等式(78)是我自己猜的，因为网上的推导直接这么替换的。

将等式(77)(78)代入等式(76)得：

$$\delta\dot{\theta}_{i+\frac{1}{2}}^{b_k} = -\left[\frac{w_i + w_{i+1}}{2} - \hat{b}_{w_i}\right]_\times \delta\theta_i^{b_k} - \delta b_{w_i} - \frac{n_{w_0} + n_{w_1}}{2} \tag{79}$$

由于 n_{w_0} 和 n_{w_1} 均为高斯噪声，加也行，减也行，但考虑到 vins-mono 的代码中使用

的是加，我们也改为使用加：

$$\delta\dot{\theta}^{b_k}_{i+\frac{1}{2}} = -\left[\frac{w_i + w_{i+1}}{2} - \hat{b}_{w_i}\right]_\times \delta\theta^{b_k}_i - \delta b_{w_i} + \frac{n_{w_0} + n_{w_1}}{2} \tag{80}$$

将等式(80)代入等式(75)得：

$$\delta\theta^{b_k}_{i+1} = (I - \left[\frac{w_i + w_{i+1}}{2} - \hat{b}_{w_i}\right]_\times \delta t)\delta\theta^{b_k}_i - \delta b_{w_i}\delta t + \frac{n_{w_0} + n_{w_1}}{2}\delta t \tag{81}$$

接下来计算 $\delta\beta^{b_k}_{i+1}$：

中值积分方式下：

$$\delta\beta^{b_k}_{i+1} = \delta\beta^{b_k}_i + \delta\dot{\beta}^{b_k}_{i+\frac{1}{2}}\delta t \tag{82}$$

由等式(32)可得：

$$\delta\dot{\beta}^{b_k}_{i+\frac{1}{2}} = -\frac{R(\hat{\gamma}^{b_k}_i)\left[a_i - \hat{b}_{a_i}\right]_\times \delta\theta^{b_k}_i + R(\hat{\gamma}^{b_k}_{i+1})\left[a_{i+1} - \hat{b}_{a_{i+1}}\right]_\times \delta\theta^{b_k}_{i+1}}{2} - \frac{R(\hat{\gamma}^{b_k}_i)\delta b_{a_i} + R(\hat{\gamma}^{b_k}_{i+1})\delta b_{a_{i+1}}}{2}$$
$$- \frac{R(\hat{\gamma}^{b_k}_i)n_{a_0} + R(\hat{\gamma}^{b_k}_{i+1})n_{a_1}}{2} \tag{83}$$

由于 n_{a_0} 和 n_{a_1} 均为高斯噪声，加也行，减也行，但考虑到 vins-mono 的代码中使用的是加，我们也改为使用加：

$$\delta\dot{\beta}^{b_k}_{i+\frac{1}{2}} = -\frac{R(\hat{\gamma}^{b_k}_i)\left[a_i - \hat{b}_{a_i}\right]_\times \delta\theta^{b_k}_i + R(\hat{\gamma}^{b_k}_{i+1})\left[a_{i+1} - \hat{b}_{a_{i+1}}\right]_\times \delta\theta^{b_k}_{i+1}}{2} - \frac{R(\hat{\gamma}^{b_k}_i)\delta b_{a_i} + R(\hat{\gamma}^{b_k}_{i+1})\delta b_{a_{i+1}}}{2}$$
$$+ \frac{R(\hat{\gamma}^{b_k}_i)n_{a_0} + R(\hat{\gamma}^{b_k}_{i+1})n_{a_1}}{2} \tag{84}$$

由等式(29)(34)得：

$$\hat{b}_{a_{i+1}} = \hat{b}_{a_i} \tag{85}$$

$$\delta b_{a_{i+1}} = \delta b_{a_i} \tag{86}$$

等式(86)是我猜的，因为网上的推导是直接这么替换的。

将等式(85)(86)代入等式(83)：

$$\delta\dot{\beta}^{b_k}_{i+\frac{1}{2}} = -\frac{R(\hat{\gamma}^{b_k}_i)\left[a_i - \hat{b}_{a_i}\right]_\times \delta\theta^{b_k}_i + R(\hat{\gamma}^{b_k}_{i+1})\left[a_{i+1} - \hat{b}_{a_i}\right]_\times \delta\theta^{b_k}_{i+1}}{2} - \frac{R(\hat{\gamma}^{b_k}_i)\delta b_{a_i} + R(\hat{\gamma}^{b_k}_{i+1})\delta b_{a_i}}{2}$$
$$+ \frac{R(\hat{\gamma}^{b_k}_i)n_{a_0} + R(\hat{\gamma}^{b_k}_i)n_{a_1}}{2} \tag{87}$$

将等式(81)代入等式(87)：

$$
\begin{aligned}
\delta\dot{\beta}^{b_k}_{i+\frac{1}{2}} = & -\frac{1}{2}R(\hat{\gamma}^{b_k}_i)\Big[a_i - \hat{b}_{a_i}\Big]_\times \delta\theta^{b_k}_i \\
& -\frac{1}{2}R(\hat{\gamma}^{b_k}_{i+1})\Big[a_{i+1} - \hat{b}_{a_{i+1}}\Big]_\times ((I - \Big[\frac{w_i + w_{i+1}}{2} - \hat{b}_{w_i}\Big]_\times \delta t)\delta\theta^{b_k}_i - \delta b_{w_i}\delta t + \frac{n_{w_0} + n_{w_1}}{2}\delta t) \\
& -\frac{R(\hat{\gamma}^{b_k}_i)\delta b_{a_i} + R(\hat{\gamma}^{b_k}_{i+1})\delta b_{a_i}}{2} \\
& +\frac{R(\hat{\gamma}^{b_k}_i)n_{a_0} + R(\hat{\gamma}^{b_k}_{i+1})n_{a_1}}{2}
\end{aligned}
\tag{88}
$$

将等式(88)代入等式(82)得：

$$
\begin{aligned}
\delta\beta^{b_k}_{i+1} = & \ \delta\beta^{b_k}_i \\
& -\frac{1}{2}(R(\hat{\gamma}^{b_k}_i)\Big[a_i - \hat{b}_{a_i}\Big]_\times \delta t + R(\hat{\gamma}^{b_k}_{i+1})\Big[a_{i+1} - \hat{b}_{a_i}\Big]_\times ((I - \Big[\frac{w_i + w_{i+1}}{2} - \hat{b}_{w_i}\Big]_\times \delta t)\delta t)\delta\theta^{b_k}_i \\
& -\frac{1}{2}(R(\hat{\gamma}^{b_k}_i) + R(\hat{\gamma}^{b_k}_{i+1}))\delta t\delta b_{a_i} \\
& +\frac{1}{2}R(\hat{\gamma}^{b_k}_{i+1})\Big[a_{i+1} - \hat{b}_{a_i}\Big]_\times \delta t^2 \delta b_{w_i} \\
& +\frac{1}{2}R(\hat{\gamma}^{b_k}_i)\delta t n_{a_0} \\
& +\frac{1}{2}R(\hat{\gamma}^{b_k}_{i+1})\delta t n_{a_1} \\
& -\frac{1}{4}R(\hat{\gamma}^{b_k}_{i+1})\Big[a_{i+1} - \hat{b}_{a_i}\Big]_\times \delta t^2 n_{w_0} \\
& -\frac{1}{4}R(\hat{\gamma}^{b_k}_{i+1})\Big[a_{i+1} - \hat{b}_{a_i}\Big]_\times \delta t^2 n_{w_1}
\end{aligned}
\tag{89}
$$

接下来计算 $\delta\alpha^{b_k}_{i+1}$：

中值积分方式下：

$$
\delta\alpha^{b_k}_{i+1} = \delta\alpha^{b_k}_i + \delta\dot{\alpha}^{b_k}_{i+\frac{1}{2}}\delta t
\tag{90}
$$

由等式(31)得：

$$
\begin{aligned}
\delta\dot{\alpha}^{b_k}_{i+\frac{1}{2}} &= \delta\beta^{b_k}_{i+\frac{1}{2}} \\
&= \delta\beta^{b_k}_i + \delta\dot{\beta}^{b_k}_{i+\frac{1}{2}}\frac{1}{2}\delta t
\end{aligned}
\tag{91}
$$

将等式(88)代入等式(91)得：

$$\delta\dot{\alpha}_{i+\frac{1}{2}}^{b_k} = \delta\beta_i^{b_k}$$

$$-\frac{1}{4}(R(\hat{\gamma}_i^{b_k})\left[a_i - \hat{b}_{a_i}\right]_{\times}\delta t + R(\hat{\gamma}_{i+1}^{b_k})\left[a_{i+1} - \hat{b}_{a_i}\right]_{\times}(I - \left[\frac{w_i + w_{i+1}}{2} - \hat{b}_{w_i}\right]_{\times}\delta t)\delta t)\delta\theta_i^{b_k}$$

$$-\frac{1}{4}(R(\hat{\gamma}_i^{b_k}) + R(\hat{\gamma}_{i+1}^{b_k}))\delta t\delta b_{a_i}$$

$$+\frac{1}{4}R(\hat{\gamma}_{i+1}^{b_k})\left[a_{i+1} - \hat{b}_{a_i}\right]_{\times}\delta t^2\delta b_{w_i}$$

$$+\frac{1}{4}R(\hat{\gamma}_i^{b_k})\delta t n_{a_0} \tag{92}$$

$$+\frac{1}{4}R(\hat{\gamma}_{i+1}^{b_k})\delta t n_{a_1}$$

$$-\frac{1}{8}R(\hat{\gamma}_{i+1}^{b_k})\left[a_{i+1} - \hat{b}_{a_i}\right]_{\times}\delta t^2 n_{w_0}$$

$$-\frac{1}{8}R(\hat{\gamma}_{i+1}^{b_k})\left[a_{i+1} - \hat{b}_{a_i}\right]_{\times}\delta t^2 n_{w_1}$$

将等式(92)代入等式(90)得：

$$\delta\alpha_{i+1}^{b_k} = \delta\alpha_i^{b_k}$$

$$+\delta\beta_i^{b_k}\delta t$$

$$-\frac{1}{4}(R(\hat{\gamma}_i^{b_k})\left[a_i - \hat{b}_{a_i}\right]_{\times}\delta t^2 + R(\hat{\gamma}_{i+1}^{b_k})\left[a_{i+1} - \hat{b}_{a_i}\right]_{\times}(I - \left[\frac{w_i + w_{i+1}}{2} - \hat{b}_{w_i}\right]_{\times}\delta t)\delta t^2)\delta\theta_i^{b_k}$$

$$-\frac{1}{4}(R(\hat{\gamma}_i^{b_k}) + R(\hat{\gamma}_{i+1}^{b_k}))\delta t^2\delta b_{a_i}$$

$$+\frac{1}{4}R(\hat{\gamma}_{i+1}^{b_k})\left[a_{i+1} - \hat{b}_{a_i}\right]_{\times}\delta t^3\delta b_{w_i}$$

$$+\frac{1}{4}R(\hat{\gamma}_i^{b_k})\delta t^2 n_{a_0}$$

$$+\frac{1}{4}R(\hat{\gamma}_{i+1}^{b_k})\delta t^2 n_{a_1}$$

$$-\frac{1}{8}R(\hat{\gamma}_{i+1}^{b_k})\left[a_{i+1} - \hat{b}_{a_i}\right]_{\times}\delta t^3 n_{w_0}$$

$$-\frac{1}{8}R(\hat{\gamma}_{i+1}^{b_k})\left[a_{i+1} - \hat{b}_{a_i}\right]_{\times}\delta t^3 n_{w_1} \tag{93}$$

$$
\begin{bmatrix} \delta\alpha_{i+1}^{b_k} \\ \delta\theta_{i+1}^{b_k} \\ \delta\beta_{i+1}^{b_k} \\ \delta b_{a_{i+1}} \\ \delta b_{w_{i+1}} \end{bmatrix} = \begin{bmatrix} I & f_{01} & \delta tI & -\frac{1}{4}(R(\hat{\gamma}_i^{b_k})+R(\hat{\gamma}_{i+1}^{b_k}))\delta t^2 & f_{04} \\ 0 & I-\left[\frac{w_i+w_{i+1}}{2}-\hat{b}_{w_i}\right]_\times \delta t & 0 & 0 & -\delta tI \\ 0 & f_{21} & I & -\frac{1}{2}(R(\hat{\gamma}_i^{b_k})+R(\hat{\gamma}_{i+1}^{b_k}))\delta t & f_{24} \\ 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & I \end{bmatrix} \begin{bmatrix} \delta\alpha_i^{b_k} \\ \delta\theta_i^{b_k} \\ \delta\beta_i^{b_k} \\ \delta b_{a_i} \\ \delta b_{w_i} \end{bmatrix}
$$

$$
+ \begin{bmatrix} \frac{1}{4}R(\hat{\gamma}_i^{b_k})\delta t^2 & v_{01} & \frac{1}{4}R(\hat{\gamma}_{i+1}^{b_k})\delta t^2 & v_{03} & 0 & 0 \\ 0 & \frac{1}{2}\delta tI & 0 & \frac{1}{2}\delta tI & 0 & 0 \\ \frac{1}{2}R(\hat{\gamma}_i^{b_k})\delta t & v_{21} & \frac{1}{2}R(\hat{\gamma}_{i+1}^{b_k})\delta t & v_{23} & 0 & 0 \\ 0 & 0 & 0 & 0 & \delta tI & 0 \\ 0 & 0 & 0 & 0 & 0 & \delta tI \end{bmatrix} \begin{bmatrix} n_{a_0} \\ n_{w_0} \\ n_{a_1} \\ n_{w_1} \\ n_{b_a} \\ n_{b_w} \end{bmatrix} \tag{94}
$$

其中：

$$
f_{01} = -\frac{1}{4}R(\hat{\gamma}_i^{b_k})\left[a_i-\hat{b}_{a_i}\right]_\times \delta t^2 - \frac{1}{4}R(\hat{\gamma}_{i+1}^{b_k})\left[a_{i+1}-\hat{b}_{a_i}\right]_\times (I-\left[\frac{w_i+w_{i+1}}{2}-\hat{b}_{w_i}\right]_\times \delta t)\delta t^2 \tag{95}
$$

$$
f_{04} = -\frac{1}{4}R(\hat{\gamma}_{i+1}^{b_k})\left[a_{i+1}-\hat{b}_{a_i}\right]_\times \delta t^2(-\delta t) \tag{96}
$$

$$
f_{21} = -\frac{1}{2}R(\hat{\gamma}_i^{b_k})\left[a_i-\hat{b}_{a_i}\right]_\times \delta t - \frac{1}{2}R(\hat{\gamma}_{i+1}^{b_k})\left[a_{i+1}-\hat{b}_{a_i}\right]_\times (I-\left[\frac{w_i+w_{i+1}}{2}-\hat{b}_{w_i}\right]_\times \delta t)\delta t \tag{97}
$$

$$
f_{24} = -\frac{1}{2}R(\hat{\gamma}_{i+1}^{b_k})\left[a_{i+1}-\hat{b}_{a_i}\right]_\times \delta t(-\delta t) \tag{98}
$$

$$
v_{01} = \frac{1}{4}(-R(\hat{\gamma}_{i+1}^{b_k})\left[a_{i+1}-\hat{b}_{a_i}\right]_\times \delta t^2)\frac{1}{2}\delta t \tag{99}
$$

$$
v_{03} = \frac{1}{4}(-R(\hat{\gamma}_{i+1}^{b_k})\left[a_{i+1}-\hat{b}_{a_i}\right]_\times \delta t^2)\frac{1}{2}\delta t = v_{01} \tag{100}
$$

$$
v_{21} = \frac{1}{2}(-R(\hat{\gamma}_{i+1}^{b_k})\left[a_{i+1}-\hat{b}_{a_i}\right]_\times \delta t)\frac{1}{2}\delta t \tag{101}
$$

$$
v_{23} = \frac{1}{2}(-R(\hat{\gamma}_{i+1}^{b_k})\left[a_{i+1}-\hat{b}_{a_i}\right]_\times \delta t)\frac{1}{2}\delta t = v_{21} \tag{102}
$$

等式(94)与 vins-mono 代码一致，vins-mono 对应代码在 integration_base.h 中，如下图所示：

```
MatrixXd F = MatrixXd::Zero(15, 15);
F.block<3, 3>(0, 0) = Matrix3d::Identity();
F.block<3, 3>(0, 3) = -0.25 * delta_q.toRotationMatrix() * R_a_0_x * _dt * _dt +
                      -0.25 * result_delta_q.toRotationMatrix() * R_a_1_x * (Matrix3d::Identity() - R_w_x * _dt) * _dt * _dt;
F.block<3, 3>(0, 6) = MatrixXd::Identity(3,3) * _dt;
F.block<3, 3>(0, 9) = -0.25 * (delta_q.toRotationMatrix() + result_delta_q.toRotationMatrix()) * _dt * _dt;
F.block<3, 3>(0, 12) = -0.25 * result_delta_q.toRotationMatrix() * R_a_1_x * _dt * _dt * -_dt;
F.block<3, 3>(3, 3) = Matrix3d::Identity() - R_w_x * _dt;
F.block<3, 3>(3, 12) = -1.0 * MatrixXd::Identity(3,3) * _dt;
F.block<3, 3>(6, 3) = -0.5 * delta_q.toRotationMatrix() * R_a_0_x * _dt +
                      -0.5 * result_delta_q.toRotationMatrix() * R_a_1_x * (Matrix3d::Identity() - R_w_x * _dt) * _dt;
F.block<3, 3>(6, 6) = Matrix3d::Identity();
F.block<3, 3>(6, 9) = -0.5 * (delta_q.toRotationMatrix() + result_delta_q.toRotationMatrix()) * _dt;
F.block<3, 3>(6, 12) = -0.5 * result_delta_q.toRotationMatrix() * R_a_1_x * _dt * -_dt;
F.block<3, 3>(9, 9) = Matrix3d::Identity();
F.block<3, 3>(12, 12) = Matrix3d::Identity();
//cout<<"A"<<endl<<A<<endl;

MatrixXd V = MatrixXd::Zero(15,18);
V.block<3, 3>(0, 0) =  0.25 * delta_q.toRotationMatrix() * _dt * _dt;
V.block<3, 3>(0, 3) =  0.25 * -result_delta_q.toRotationMatrix() * R_a_1_x  * _dt * _dt * 0.5 * _dt;
V.block<3, 3>(0, 6) =  0.25 * result_delta_q.toRotationMatrix() * _dt * _dt;
V.block<3, 3>(0, 9) =  V.block<3, 3>(0, 3);
V.block<3, 3>(3, 3) =  0.5 * MatrixXd::Identity(3,3) * _dt;
V.block<3, 3>(3, 9) =  0.5 * MatrixXd::Identity(3,3) * _dt;
V.block<3, 3>(6, 0) =  0.5 * delta_q.toRotationMatrix() * _dt;
V.block<3, 3>(6, 3) =  0.5 * -result_delta_q.toRotationMatrix() * R_a_1_x  * _dt * 0.5 * _dt;
V.block<3, 3>(6, 6) =  0.5 * result_delta_q.toRotationMatrix() * _dt;
V.block<3, 3>(6, 9) =  V.block<3, 3>(6, 3);
V.block<3, 3>(9, 12) = MatrixXd::Identity(3,3) * _dt;
V.block<3, 3>(12, 15) = MatrixXd::Identity(3,3) * _dt;
```

将等式(94)简写为：

$$\delta z_{i+1} = F \delta z_i + VN \tag{103}$$

其中：

$$\delta z_i = \begin{bmatrix} \delta \alpha_i^{b_k} \\ \delta \theta_i^{b_k} \\ \delta \beta_i^{b_k} \\ \delta b_{a_i} \\ \delta b_{w_i} \end{bmatrix} \tag{104}$$

$$\delta z_{i+1} = \begin{bmatrix} \delta \alpha_{i+1}^{b_k} \\ \delta \theta_{i+1}^{b_k} \\ \delta \beta_{i+1}^{b_k} \\ \delta b_{a_{i+1}} \\ \delta b_{w_{i+1}} \end{bmatrix} \tag{105}$$

$$F = \begin{bmatrix} I & f_{01} & \delta t I & -\frac{1}{4}(R(\hat{\gamma}_i^{b_k}) + R(\hat{\gamma}_{i+1}^{b_k}))\delta t^2 & f_{04} \\ 0 & I - \left[\frac{w_i + w_{i+1}}{2} - \hat{b}_{w_i}\right]_{\times} \delta t & 0 & 0 & -\delta t I \\ 0 & f_{21} & I & -\frac{1}{2}(R(\hat{\gamma}_i^{b_k}) + R(\hat{\gamma}_{i+1}^{b_k}))\delta t & f_{24} \\ 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & I \end{bmatrix} \tag{106}$$

$$V = \begin{bmatrix} \frac{1}{4}R(\hat{\gamma}_i^{b_k})\delta t^2 & v_{01} & \frac{1}{4}R(\hat{\gamma}_{i+1}^{b_k})\delta t^2 & v_{03} & 0 & 0 \\ 0 & \frac{1}{2}\delta t I & 0 & \frac{1}{2}\delta t I & 0 & 0 \\ \frac{1}{2}R(\hat{\gamma}_i^{b_k})\delta t & v_{21} & \frac{1}{2}R(\hat{\gamma}_{i+1}^{b_k})\delta t & v_{23} & 0 & 0 \\ 0 & 0 & 0 & 0 & \delta t I & 0 \\ 0 & 0 & 0 & 0 & 0 & \delta t I \end{bmatrix} \tag{107}$$

$$N = \begin{bmatrix} n_{a_0} \\ n_{w_0} \\ n_{a_1} \\ n_{w_1} \\ n_{b_a} \\ n_{b_w} \end{bmatrix} \tag{108}$$

由等式(103)可得雅可比矩阵和协方差矩阵的递推方程：

$$J_{i+1} = FJ_i \tag{109}$$

$$P_{i+1} = FP_iF^T + VQV^T \tag{110}$$

其中 Q 为噪声信号 N 的协方差矩阵，由于假设 N 的各个分量相互独立，所以 Q 为对角形矩阵：

$$Q = \begin{bmatrix} \sigma_{n_a}^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{n_w}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{n_a}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{n_w}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{n_{b_a}}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{n_{b_w}}^2 \end{bmatrix} \tag{111}$$

令 b_k、b_{k+1} 表示两个相邻的关键帧，两个关键帧之间的时间间隔内有一些列的 IMU 测量，i 对应其测量的离散时间点。初始时刻（即第关键帧 b_k 对应的时刻）系统状态的雅可比矩阵和协方差矩阵为：

$$J_{b_k} = I \tag{112}$$

$$P_{b_k} = 0 \tag{113}$$

根据初始值(112)(113)和递推方程(109)(110)可得关键帧 b_{k+1} 对应时刻，系统状态的雅可比矩阵 $J_{b_{k+1}}$ 和协方差矩阵 $P_{b_{k+1}}$。

$\alpha_{b_{k+1}}^{b_k}$、$\beta_{b_{k+1}}^{b_k}$ 和 $\gamma_{b_{k+1}}^{b_k}$ 相对于关键帧 b_k 时刻加速度计偏置误差 δb_{a_k} 和陀螺仪偏置误差 δb_{w_k} 的一阶近似可以表示为：

$$\alpha_{b_{k+1}}^{b_k} = \hat{\alpha}_{b_{k+1}}^{b_k} + J_{\delta b_{a_k}}^{\delta \alpha_{b_{k+1}}^{b_k}} \delta b_{a_k} + J_{\delta b_{w_k}}^{\delta \alpha_{b_{k+1}}^{b_k}} \delta b_{w_k} \tag{114}$$

$$\beta_{b_{k+1}}^{b_k} = \hat{\beta}_{b_k}^{b_k} + J_{\delta b_{a_k}}^{\delta \beta_{b_{k+1}}^{b_k}} \delta b_{a_k} + J_{\delta b_{w_k}}^{\delta \beta_{b_{k+1}}^{b_k}} \delta b_{w_k} \tag{115}$$

$$\gamma_{b_{k+1}}^{b_k} = \hat{\gamma}_{b_{k+1}}^{b_k} \otimes \begin{bmatrix} 1 \\ \frac{1}{2} J_{\delta b_{w_k}}^{\delta \theta_{b_{k+1}}^{b_k}} \delta b_{w_k} \end{bmatrix} \tag{116}$$

$J_{\delta b_{a_k}}^{\delta \alpha_{b_{k+1}}^{b_k}}$、$J_{\delta b_{w_k}}^{\delta \alpha_{b_{k+1}}^{b_k}}$、$J_{\delta b_{a_k}}^{\delta \beta_{b_{k+1}}^{b_k}}$、$J_{\delta b_{w_k}}^{\delta \beta_{b_{k+1}}^{b_k}}$、$J_{\delta b_{w_k}}^{\delta \theta_{b_{k+1}}^{b_k}}$ 的含义为其对应符号中，上标对下标的雅可比矩阵，它们都是 $J_{b_{k+1}}$ 的子块矩阵，可以从 $J_{b_{k+1}}$ 对应位置中获取。

当对加速度计偏置和陀螺仪的偏置发生（微小）改变时，就可以根据等式(114)(115)(116)对预积分项进行修正，避免了重复积分。

至此，可以写出预积分形式下 IMU 的测量模型：

$$\begin{bmatrix} \hat{\alpha}_{b_{k+1}}^{b_k} \\ \hat{\gamma}_{b_{k+1}}^{b_k} \\ \hat{\beta}_{b_{k+1}}^{b_k} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} R_w^{b_k}(p_{b_{k+1}}^w - p_{b_k}^w - v_{b_k}^w \Delta t_k + \frac{1}{2} g^w \Delta t_k^2) \\ (q_{b_k}^w)^{-1} \otimes q_{b_{k+1}}^w \\ R_w^{b_k}(v_{b_{k+1}}^w - v_{b_k}^w + g^w \Delta t_k) \\ b_{ab_{k+1}} - b_{ab_k} \\ b_{wb_{k+1}} - b_{wb_k} \end{bmatrix} \tag{117}$$

（不太明白(117)中第 4、5 行为什么这么写，可能是因为测量过程中默认后一帧 IMU

的偏置等于前一帧的偏置）

3 视觉处理前端

VINS-Mono 的视觉处理前端被封装为一个 ROS 节点 feature_tracker_node，该节点的话题订阅、话题发布等信息如下表所示：

	topic	type	note
Subscribe	image	sensor_msgs::ImageConstPtr	相机原始图
Publish	feature	sensor_msgs::PointCloud	跟踪的特征点，给后端优化用
Publish	feature_img	sensor_msgs::Image	跟踪特征点图片，输出给RVIZ，调试用

ROS 节点 feature_tracker_node 接收到图像数据后，会进入回调函数 img_callback()，每个相机都有一个 FeatureTracker 实例，即 TrackerData[i]。然后调用每个相机实例中的 readImage() 函数来提取和跟踪特征点，然后将所有相机的特征点融合到 feature_points(sensor_msgs::PointCloudPtr)中发布。

回调函数 img_callback()中有一部分代码逻辑是处理双目的。STEREO_TRACK 为双目开关控制，STEREO_TRACK 是在 parameters.h 中声明的一个 int 变量。STEREO_TRACK 置为 1，则相机 0 和相机 1 是双目相机，相机 0 调用 FeatureTracker 进行跟踪，相机 1 通过 LK 光流跟踪相机 0 中的特征点，处理逻辑和前后帧跟踪基本一致，区别仅在于：双目相机间的跟踪不再提取新的特征点，而前后帧跟踪可能需要提取新的特征点。

FeatureTracker 类中最主要的成员函数是 readImage()，这里涉及到图像的 3 个变量需要说明一下，以方便理解代码，代码中的 3 个 img：

prev_img   cur_img   forw_img

cur_img 和 forw_img 分别是光流跟踪的前后两帧，forw_img 才是真正的当前帧，cur_img 实际上是上一帧，prev_img 是上一次发布的帧。prev_img 的用处是：光流跟踪后用 prev_img 和 forw_img 根据 Fundamental Matrix 做 RANSAC 剔除 outlier，也就是 rejectWithF()函数。

readImage()的作用是对新来的图像使用光流法进行特征点跟踪，处理流程为：

1. 若控制参数 EQUALIZE 为真，则调用 cv::creatCLAHE()对输入图像做自适应直方图均衡；否则，不做处理。

2. 调用 cv::calcOpticalFlowPyrLK()进行光流跟踪，跟踪前一帧的特征点 cur_pts 得到 forw_pts，根据 status 把跟踪失败的点剔除（注意 prev, cur, forw, ids, track_cnt 都要剔除），而且还需要将跟踪到图像边界外的点剔除。

3. 如果不需要发布当前帧的数据，那么直接把当前帧 forw 的数据赋给上一帧 cur，然后在这一步就结束了。

4. 如果需要发布当前帧的数据，先调用 rejectWithF()对 prev_pts 和 forw_pts 做 RANSAC 剔除 outlier（实际上就是调用了 cv::findFundamentalMat()函数）。然后所有剩下的特征点的 track_cnt 加 1。

5. 在跟踪过程中，为了保持跟踪到的特征点在当前帧图像中均匀分布（避免特征点扎堆的现象），会调用 FeatureTracker 类中的 FeatureTracker::;setMask()函数，先对跟踪到的特征点 forw_pts 按照跟踪次数降序排列（认为特征点被跟踪到的次数越多越好），然后遍历这个降序排列，对于遍历的每一个特征点，在 mask 中将该点周围半径为 MIN_DIST 的区域设置为 0，在后续的遍历过程中，不再选择该区域内的点。

6. 由于跟踪过程中，上一帧特征点由于各种原因无法被跟踪，而且为了保证特征点均匀分布而剔除了一些特征点，如果不补充新的特征点，那么每一帧中特征点的数量会越来越少。所以，当前帧除了跟踪前一帧中的特征点，还会调用 cv::goodFeaturesToTrack()在 mask 中不为 0 的区域提取新的特征点。新提取的特征点通过 FeatureTracker::addPoints()函数 push 到 forw_pts 中，id 初始化为-1，track_cnt 初始化为 1。

4 系统初始化

VINS-Mono 的初始化采用松组合（loosely coupled）的方式获取初始值。首先用 SFM 求解滑动窗口内所有帧的位姿（以第一帧作为参考坐标系）和所有路标点的 3D 位置。然后将 SFM 的结果与 IMU 预积分的值进行对齐，实现对陀螺仪偏置的校正，再求解每一帧对应的速度，求解重力向量方向，恢复单目相机的尺度因子。需要注意的是，在初始化过程中，并没有对加速度计的偏置进行校正，这是因为重力是初始化过程中待求的量，而加速度计偏置与重力耦合，而且系统的加速度相对于重力加速度很小，所以加速度计偏置在初始化过程中很难观测，因此初始化过程中不考虑加速度计偏置的校正。

4.1 相机与 IMU 之间的相对旋转

相对旋转是相机与 IMU 之间外参的一部分。相机与 IMU 之间的旋转标定非常重要，偏差 $1°\sim2°$ 系统的精度就会变得很低。这部分内容参考[2]中的 V-A 部分。

通过视觉 SFM 获取了相邻两关键帧之间相机的相对旋转 $R_{c_{k+1}}^{c_k}$ ，通过 IMU 预积分获取了相邻两关键帧之间体坐标系（IMU 坐标系）的相对旋转 $R_{b_{k+1}}^{b_k}$ 。设 IMU 与相机之间的相对旋转为 R_c^b ，则对于任意 k ，等式(118)成立：

$$R_{b_{k+1}}^{b_k} R_c^b = R_c^b R_{c_{k+1}}^{c_k} \tag{118}$$

将等式(118)用四元数表示：

$$q_{b_{k+1}}^{b_k} \otimes q_c^b = q_c^b \otimes q_{c_{k+1}}^{c_k} \tag{119}$$

由等式(119)可得：

$$(Q_1(q_{b_{k+1}}^{b_k}) - Q_2(q_{c_{k+1}}^{c_k})) q_c^b = Q_{k+1}^k q_c^b = 0 \tag{120}$$

定义（注意四元数的定义采用了虚部在前，实部在后的形式）：

$$q = \begin{bmatrix} q_x \\ q_y \\ q_z \\ q_w \end{bmatrix}, q_v = \begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix}, [q_v]_\times = \begin{bmatrix} 0 & -q_z & q_y \\ q_z & 0 & -q_x \\ -q_y & q_x & 0 \end{bmatrix} \tag{121}$$

$$Q_1(q) = \begin{bmatrix} q_w I + [q_v] & q_v \\ -q_v^T & q_w \end{bmatrix} \tag{122}$$

$$Q_2(q) = \begin{bmatrix} q_w I - [q_v] & q_v \\ -q_v^T & q_w \end{bmatrix} \tag{123}$$

对于 N 对相对旋转的测量值，可得过约束的线性方程：

$$\begin{bmatrix} w_1^0 Q_1^0 \\ w_2^1 Q_2^1 \\ \vdots \\ w_N^{N-1} Q_N^{N-1} \end{bmatrix} q_c^b = Q_N q_c^b \tag{124}$$

（未完待续）

4.2 单目相机初始化

这一步就是常规的单目 SLAM 的初始化过程。

（待补充）

纯视觉初始化时，采用第一帧 c_0 时的相机坐标系作为参考坐标系，通过纯视觉 SFM 可获得滑动窗口中所有关键帧的位姿 $(\bar{p}_{b_k}^{c_0}, q_{b_k}^{c_0})$，其中 $\bar{p}_{b_k}^{c_0}$ 从 c_0 坐标系到 b_k 坐标系的位移向量（无绝对尺度信息）在 c_0 坐标系下的坐标，$q_{b_k}^{c_0}$ 为从 b_k 坐标系到 c_0 坐标系的旋转四元数。假设已知相机与 IMU 之间的外参 (p_c^b, q_c^b)，其中 p_c^b 为从本体坐标系到相应时刻相机坐标系的位移向量在相应时刻本体坐标系下的坐标，q_c^b 为从本体坐标系到相应时刻相机坐标系的旋转四元数。可按照等式(125)(126)完成从本体坐标系转换到 c_0 坐标系：

$$q_{b_k}^{c_0} = q_{c_k}^{c_0} \otimes (q_c^b)^{-1} \tag{125}$$

$$s\bar{p}_{b_k}^{c_0} = s\bar{p}_{c_k}^{c_0} - R_{b_k}^{c_0} p_c^b \tag{126}$$

其中参数 s 给视觉测量的位移赋予尺度信息。

4.3 SFM 与 IMU 预积分结果对齐

SFM 与 IMU 预积分结果对齐主要解决 3 个问题：

1. 陀螺仪偏置的校准；
2. 初始化速度、重力向量和尺度因子；
3. 重力向量的精调。

4.3.1 陀螺仪偏置的校正

这一部分的内容对应于 VINS-Mono 代码 initial_aligment.cpp 中的 solveGyroscopeBias()函数。

考虑滑动窗口中的连续两个关键帧 b_k 和 b_{k+1}，通过 SFM 可以获取两个关键帧时刻本体坐标系相对于参考相机坐标系的旋转 $q_{b_k}^{c_0}$ 和 $q_{b_{k+1}}^{c_0}$，通过 IMU 预积分可以获得两个关键帧时刻本体坐标系之间的相对旋转约束 $\gamma_{b_{k+1}}^{b_k}$，校正陀螺仪偏置的目标函数为：

$$\min_{\delta b_w} \sum_{k \in B} \left\| q_{b_{k+1}}^{c_0}{}^{-1} \otimes q_{b_k}^{c_0} \otimes \gamma_{b_{k+1}}^{b_k} \right\|^2 \tag{127}$$

其中：

$$\gamma_{b_{k+1}}^{b_k} = \hat{\gamma}_{b_{k+1}}^{b_k} \otimes \begin{bmatrix} 1 \\ \frac{1}{2} J_{\delta b_{w_k}}^{\delta\theta_{b_{k+1}}^{b_k}} \delta b_{w_k} \end{bmatrix} \tag{128}$$

B 表示滑动窗口中的所有关键帧，等式(128)为 IMU 预积分中推导的，$\gamma_{b_{k+1}}^{b_k}$ 关于陀螺仪偏置误差 δb_w 的一阶近似。

等式(127)中目标函数的最小值为单位四元数，所以可将等式(127)进一步改写为（这里的四元数采用实部在前，虚部在后的形式）：

$$
\begin{aligned}
q_{b_{k+1}}^{c_0}{}^{-1} \otimes q_{b_k}^{c_0} \otimes \gamma_{b_{k+1}}^{b_k} &= \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\
\hat{\gamma}_{b_{k+1}}^{b_k} \otimes \begin{bmatrix} 1 \\ \frac{1}{2} J_{\delta b_{w_k}}^{\delta\theta_{b_{k+1}}^{b_k}} \delta b_{w_k} \end{bmatrix} &= q_{b_k}^{c_0-1} \otimes q_{b_{k+1}}^{c_0} \otimes \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\
\begin{bmatrix} 1 \\ \frac{1}{2} J_{\delta b_{w_k}}^{\delta\theta_{b_{k+1}}^{b_k}} \delta b_{w_k} \end{bmatrix} &= \hat{\gamma}_{b_{k+1}}^{b_k}{}^{-1} \otimes q_{b_k}^{c_0-1} \otimes q_{b_{k+1}}^{c_0} \otimes \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}
\end{aligned}
\tag{129}
$$

只考虑虚部，则有：

$$J_{\delta b_{w_k}}^{\delta\theta_{b_{k+1}}^{b_k}}\delta b_{w_k} = 2(\hat{\gamma}_{b_{k+1}}^{b_k}{}^{-1}\otimes q_{b_k}^{c_0-1}\otimes q_{b_{k+1}}^{c_0})_{vec} \tag{130}$$

$(q)_{vec}$ 表示取四元数 q 的虚部。

将等式(130)中等号坐标转为正定矩阵：

$$J_{\delta b_{w_k}}^{\delta\theta_{b_{k+1}}^{b_k}T}J_{\delta b_{w_k}}^{\delta\theta_{b_{k+1}}^{b_k}}\delta b_{w_k} = 2J_{\delta b_{w_k}}^{\delta\theta_{b_{k+1}}^{b_k}T}(\hat{\gamma}_{b_{k+1}}^{b_k}{}^{-1}\otimes q_{b_k}^{c_0-1}\otimes q_{b_{k+1}}^{c_0})_{vec} \tag{131}$$

这样就可以使用 cholesky 分解求解矩阵(131)，获取目标函数达到最小的解 δb_w，从而完成对陀螺仪偏置的校正。

4.3.2 初始化速度、重力向量和尺度因子

这一部分的内容对应于 VINS-Mono 代码 initial_aligment.cpp 中的 LinearAlignment() 函数。

待优化的变量为：

$$\chi_I = \begin{bmatrix} v_{b_0}^{b_0} \\ v_{b_0}^{b_0} \\ \vdots \\ v_{b_0}^{b_0} \\ g^{c_0} \\ s \end{bmatrix} \tag{132}$$

将预积分项(10)(11)中的世界坐标系 w 换为 c_0 坐标系：

$$\alpha_{b_{k+1}}^{b_k} = R_{c_0}^{b_k}(p_{b_{k+1}}^{c_0} - p_{b_k}^{c_0} - v_{b_k}^{c_0}\Delta t_k + \frac{1}{2}g^{c_0}\Delta t_k^2) \tag{133}$$

$$\beta_{b_{k+1}}^{b_k} = R_{c_0}^{b_k}(v_{b_{k+1}}^{c_0} - v_{b_k}^{c_0} + g^{c_0}\Delta t_k) \tag{134}$$

$p_{b_{k+1}}^{c_0}$ 和 $p_{b_k}^{c_0}$ 可由视觉 SFM 获得：

$$p_{b_k}^{c_0} = s\bar{p}_{b_k}^{c_0} \tag{135}$$

$$p_{b_{k+1}}^{c_0} = s\bar{p}_{b_{k+1}}^{c_0} \tag{136}$$

将等式(135)(136)代入等式(133)：

$$\alpha_{b_{k+1}}^{b_k} = R_{c_0}^{b_k}(s(\bar{p}_{b_{k+1}}^{c_0} - \bar{p}_{b_k}^{c_0}) - v_{b_k}^{c_0}\Delta t_k + \frac{1}{2}g^{c_0}\Delta t_k^2) \tag{137}$$

将等式(134)(137)中的速度都转换到 c_0 坐标系下：

$$\alpha_{b_{k+1}}^{b_k} = R_{c_0}^{b_k}(s(\overline{p}_{b_{k+1}}^{c_0} - \overline{p}_{b_k}^{c_0}) - R_{b_k}^{c_0}v_{b_k}^{b_k}\Delta t_k + \frac{1}{2}g^{c_0}\Delta t_k^2) \tag{138}$$

$$\beta_{b_{k+1}}^{b_k} = R_{c_0}^{b_k}(R_{b_{k+1}}^{c_0}v_{b_{k+1}}^{b_{k+1}} - R_{b_k}^{c_0}v_{b_k}^{b_k} + g^{c_0}\Delta t_k) \tag{139}$$

想办法将等式(138)(139)转换成 $Hx = b$ 的形式，这样便于利用 cholesky 进行求解。

将等式(126)带入等式(138)：

$$
\begin{aligned}
&\alpha_{b_{k+1}}^{b_k} = R_{c_0}^{b_k}(s(\overline{p}_{b_{k+1}}^{c_0} - \overline{p}_{b_k}^{c_0}) - R_{b_k}^{c_0}v_{b_k}^{b_k}\Delta t_k + \frac{1}{2}g^{c_0}\Delta t_k^2) \Rightarrow \\
&\alpha_{b_{k+1}}^{b_k} = R_{c_0}^{b_k}((s\overline{p}_{c_{k+1}}^{c_0} - R_{b_{k+1}}^{c_0}p_c^b) - (s\overline{p}_{c_k}^{c_0} - R_{b_k}^{c_0}p_c^b) - R_{b_k}^{c_0}v_{b_k}^{b_k}\Delta t_k + \frac{1}{2}g^{c_0}\Delta t_k^2) \Rightarrow \\
&\alpha_{b_{k+1}}^{b_k} = R_{c_0}^{b_k}s(\overline{p}_{c_{k+1}}^{c_0} - \overline{p}_{c_k}^{c_0}) - R_{c_0}^{b_k}R_{b_{k+1}}^{c_0}p_c^b + p_c^b - v_{b_k}^{b_k}\Delta t_k + \frac{1}{2}R_{c_0}^{b_k}g^{c_0}\Delta t_k^2 \Rightarrow \\
&-v_{b_k}^{b_k}\Delta t_k + \frac{1}{2}R_{c_0}^{b_k}g^{c_0}\Delta t_k^2 + R_{c_0}^{b_k}(\overline{p}_{c_{k+1}}^{c_0} - \overline{p}_{c_k}^{c_0})s = \alpha_{b_{k+1}}^{b_k} + R_{c_0}^{b_k}R_{b_{k+1}}^{c_0}p_c^b - p_c^b \Rightarrow \\
&\begin{bmatrix} -I\Delta t_k & 0 & \frac{1}{2}R_{c_0}^{b_k}\Delta t_k^2 & R_{c_0}^{b_k}(\overline{p}_{c_{k+1}}^{c_0} - \overline{p}_{c_k}^{c_0}) \end{bmatrix}\begin{bmatrix} v_{b_k}^{b_k} \\ v_{b_{k+1}}^{b_{k+1}} \\ g^{c_0} \\ s \end{bmatrix} = \alpha_{b_{k+1}}^{b_k} + R_{c_0}^{b_k}R_{b_{k+1}}^{c_0}p_c^b - p_c^b
\end{aligned}
\tag{140}
$$

同样地也将等式(139)转换为矩阵形式：

$$
\begin{aligned}
&\beta_{b_{k+1}}^{b_k} = R_{c_0}^{b_k}(R_{b_{k+1}}^{c_0}v_{b_{k+1}}^{b_{k+1}} - R_{b_k}^{c_0}v_{b_k}^{b_k} + g^{c_0}\Delta t_k) \Rightarrow \\
&-v_{b_k}^{b_k} + R_{c_0}^{b_k}R_{b_{k+1}}^{c_0}v_{b_{k+1}}^{b_{k+1}} + R_{c_0}^{b_k}g^{c_0}\Delta t_k = \beta_{b_{k+1}}^{b_k} \Rightarrow \\
&\begin{bmatrix} -I & R_{c_0}^{b_k}R_{b_{k+1}}^{c_0} & R_{c_0}^{b_k}\Delta t_k & 0 \end{bmatrix}\begin{bmatrix} v_{b_k}^{b_k} \\ v_{b_{k+1}}^{b_{k+1}} \\ g^{c_0} \\ s \end{bmatrix} = \beta_{b_{k+1}}^{b_k}
\end{aligned}
\tag{141}
$$

联立等式(140)(141)：

$$
\begin{bmatrix} -I\Delta t_k & 0 & \frac{1}{2}R_{c_0}^{b_k}\Delta t_k^2 & R_{c_0}^{b_k}(\overline{p}_{c_{k+1}}^{c_0} - \overline{p}_{c_k}^{c_0}) \\ -I & R_{c_0}^{b_k}R_{b_{k+1}}^{c_0} & R_{c_0}^{b_k}\Delta t_k & 0 \end{bmatrix}\begin{bmatrix} v_{b_k}^{b_k} \\ v_{b_{k+1}}^{b_{k+1}} \\ g^{c_0} \\ s \end{bmatrix} = \begin{bmatrix} \alpha_{b_{k+1}}^{b_k} + R_{c_0}^{b_k}R_{b_{k+1}}^{c_0}p_c^b - p_c^b \\ \beta_{b_{k+1}}^{b_k} \end{bmatrix} \tag{142}
$$

理论上等式(142)中的等号应该成立，但是由于各种噪声与数值运算的误差，等号并不严格成立，将理论真值 $\alpha_{b_{k+1}}^{b_k}$、$\beta_{b_{k+1}}^{b_k}$ 替换为通过 IMU 测量值计算出的标称值 $\hat{\alpha}_{b_{k+1}}^{b_k}$、$\hat{\beta}_{b_{k+1}}^{b_k}$，得到约等式：

$$\begin{bmatrix} -I\Delta t_k & 0 & \frac{1}{2}R_{c_0}^{b_k}\Delta t_k^2 & R_{c_0}^{b_k}(\overline{p}_{c_{k+1}}^{c_0}-\overline{p}_{c_k}^{c_0}) \\ -I & R_{c_0}^{b_k}R_{b_{k+1}}^{c_0} & R_{c_0}^{b_k}\Delta t_k & 0 \end{bmatrix}\begin{bmatrix} v_{b_k}^{b_k} \\ v_{b_{k+1}}^{b_{k+1}} \\ g^{c_0} \\ s \end{bmatrix} \approx \begin{bmatrix} \hat{\alpha}_{b_{k+1}}^{b_k}+R_{c_0}^{b_k}R_{b_{k+1}}^{c_0}p_c^b-p_c^b \\ \hat{\beta}_{b_{k+1}}^{b_k} \end{bmatrix} \quad (143)$$

令：

$$H_{b_{k+1}}^{b_k}=\begin{bmatrix} -I\Delta t_k & 0 & \frac{1}{2}R_{c_0}^{b_k}\Delta t_k^2 & R_{c_0}^{b_k}(\overline{p}_{c_{k+1}}^{c_0}-\overline{p}_{c_k}^{c_0}) \\ -I & R_{c_0}^{b_k}R_{b_{k+1}}^{c_0} & R_{c_0}^{b_k}\Delta t_k & 0 \end{bmatrix} \quad (144)$$

$$\chi_{b_{k+1}}^{b_k}=\begin{bmatrix} v_{b_k}^{b_k} \\ v_{b_{k+1}}^{b_{k+1}} \\ g^{c_0} \\ s \end{bmatrix} \quad (145)$$

$$\hat{z}_{b_{k+1}}^{b_k}=\begin{bmatrix} \hat{\alpha}_{b_{k+1}}^{b_k}+R_{c_0}^{b_k}R_{b_{k+1}}^{c_0}p_c^b-p_c^b \\ \hat{\beta}_{b_{k+1}}^{b_k} \end{bmatrix} \quad (146)$$
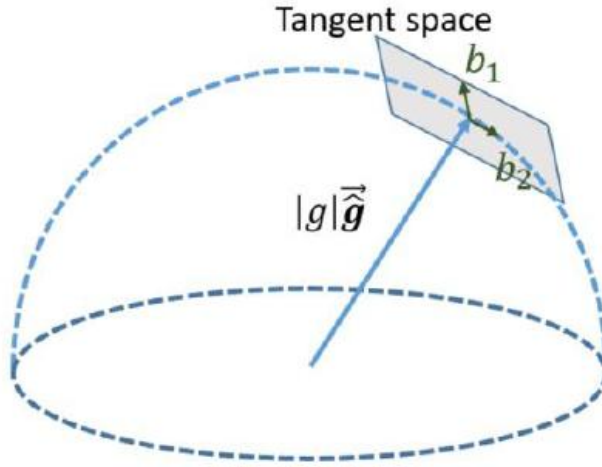
约等式(143)可简写为：

$$H_{b_{k+1}}^{b_k}\chi_{b_{k+1}}^{b_k}\approx\hat{z}_{b_{k+1}}^{b_k} \quad (147)$$

通过解线性最小二乘问题：

$$\min_{\chi_{b_{k+1}}^{b_k}}\sum_{k\in B}\left\|\hat{z}_{b_{k+1}}^{b_k}-H_{b_{k+1}}^{b_k}\chi_{b_{k+1}}^{b_k}\right\|^2 \quad (148)$$

即可优化 χ_I，即滑动窗口中所有关键帧的速度、重力向量在 c_0 坐标系中的分量列阵和单目尺度因子 s。

4.3.3 重力向量的精调

两自由度的重力矢量参数化

假设重力的模值固定，且其数值为已知量，这样重力向量的自由度就由 3 变成了 2。因此将重力向量重新参数化：

$$
\begin{aligned}
\hat{g} &= \|g\| \bar{\bar{g}} + w_1 b_1 + w_2 b_2 \\
&= \|g\| \bar{\bar{g}} + \begin{bmatrix} b_1 & b_2 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \\
&= \|g\| \bar{\bar{g}} + bw
\end{aligned}
\tag{149}
$$

$\bar{\bar{g}}$ 为重力向量的方向向量在 c_0 坐标系下的坐标，b_1、b_2 为重力向量正切空间的一对正交基。重新参数化后的重力向量由 w 控制。

将等式(149)代入等式(143)，重新整理：

$$
\begin{bmatrix}
-I\Delta t_k & 0 & \frac{1}{2} R_{c_0}^{b_k} \Delta t_k^2 b & R_{c_0}^{b_k} (\bar{p}_{c_{k+1}}^{c_0} - \bar{p}_{c_k}^{c_0}) \\
-I & R_{c_0}^{b_k} R_{b_{k+1}}^{c_0} & R_{c_0}^{b_k} \Delta t_k b & 0
\end{bmatrix}
\begin{bmatrix}
v_{b_k}^{b_k} \\
v_{b_{k+1}}^{b_{k+1}} \\
w \\
s
\end{bmatrix}
\approx
$$

$$
\begin{bmatrix}
\hat{\alpha}_{b_{k+1}}^{b_k} + R_{c_0}^{b_k} R_{b_{k+1}}^{c_0} p_c^b - p_c^b - \frac{1}{2} R_{c_0}^{b_k} \Delta t_k^2 \|g\| \bar{\bar{g}} \\
\hat{\beta}_{b_{k+1}}^{b_k} - R_{c_0}^{b_k} \Delta t_k \|g\| \bar{\bar{g}}
\end{bmatrix}
\tag{150}
$$

然后用上一节中同样的方法，解一个最小二乘问题，从而实现了重力的精调。

5 后端优化

5.1 状态向量

VINS-Mono 后端优化的状态向量包括滑动窗口内的 $n+1$ 个关键帧时刻 IMU 坐标系的位置、姿态、姿态（旋转）、加速度计偏置、陀螺仪偏置，Camera 到 IMU 的外参，$m+1$ 个 3D 路标点的逆深度：

$$
\begin{aligned}
\chi &= \left[x_0, x_1, \cdots, x_n, x_c^b, \lambda_0, \lambda_1, \cdots, \lambda_m \right] \\
x_k &= \left[p_{b_k}^w, v_{b_k}^w, q_{b_k}^w, b_a, b_g \right], k \in [0, n] \\
x_c^b &= \left[p_c^b, q_c^b \right]
\end{aligned}
\tag{151}
$$

优化过程中的误差状态量为：

$$
\begin{aligned}
\delta\chi &= \left[\delta x_0, \delta x_1, \cdots, \delta x_n, \delta x_c^b, \delta\lambda_0, \delta\lambda_1, \cdots, \delta\lambda_m \right] \\
\delta x_k &= \left[\delta p_{b_k}^w, \delta v_{b_k}^w, \delta q_{b_k}^w, \delta b_a, \delta b_g \right], k \in [0, n] \\
\delta x_c^b &= \left[\delta p_c^b, \delta q_c^b \right]
\end{aligned}
\tag{152}
$$

5.2 目标函数

$$
\min_{\chi}\left\{ \left\| r_P - H_P\chi \right\|^2 + \sum_{k \in B}\left\| r_B(\hat{z}_{b_{k+1}}^{b_k}, \chi) \right\|_{P_{b_{k+1}}^{b_k}}^2 + \sum_{(l,j) \in C}\left\| r_C(\hat{z}_l^{c_j}, \chi) \right\|_{P_l^{c_j}}^2 \right\}
\tag{153}
$$

其中 $P_{b_{k+1}}^{b_k}$ 为 IMU 预积分噪声项的协方差矩阵，$P_l^{c_j}$ 为视觉观测噪声的协方差矩阵。

目标函数中三项分别为：边缘化的先验信息、IMU 测量残差、视觉测量的残差。注意这三项使用的距离都是马氏距离。

根据高斯牛顿法求优化过程中最为核心的增量方程，以 IMU 测量残差为例，先将 $r_B(\hat{z}_{b_{k+1}}^{b_k}, \chi)$ 进行一阶泰勒展开：

$$
\begin{aligned}
\min_{\chi}\left\| r_B(\hat{z}_{b_{k+1}}^{b_k}, \chi) \right\|_{P_{b_{k+1}}^{b_k}}^2 &\approx \min_{\chi}\left\| r_B(\hat{z}_{b_{k+1}}^{b_k}, \chi) + H_{b_{k+1}}^{b_k}\delta\chi \right\|_{P_{b_{k+1}}^{b_k}}^2 \\
&= \min_{\chi}\left\{ (r_B + H_{b_{k+1}}^{b_k}\delta\chi)^T (P_{b_{k+1}}^{b_k})^{-1} (r_B + H_{b_{k+1}}^{b_k}\delta\chi) \right\} \\
&= \min_{\chi}\left\{ r_B^T (P_{b_{k+1}}^{b_k})^{-1} r_B + 2r_B^T (P_{b_{k+1}}^{b_k})^{-1} H_{b_{k+1}}^{b_k}\delta\chi + \delta\chi^T (H_{b_{k+1}}^{b_k})^T (P_{b_{k+1}}^{b_k})^{-1} H_{b_{k+1}}^{b_k}\delta\chi \right\}
\end{aligned}
\tag{154}
$$

其中 $H_{b_{k+1}}^{b_k}$ 是雅可比矩阵，不要被符号迷惑而错以为 $H_{b_{k+1}}^{b_k}$ 是 Hessian 矩阵。

求(154)中等号右边关于 $\delta\chi$ 的导数，并令其为 0，得到增量方程：

$$(H_{b_{k+1}}^{b_k})^T (P_{b_{k+1}}^{b_k})^{-1} H_{b_{k+1}}^{b_k} \delta\chi = -(H_{b_{k+1}}^{b_k})^T (P_{b_{k+1}}^{b_k})^{-1} r_B \tag{155}$$

那么可以写出目标函数(153)对应的增量方程：

$$(\Lambda_P + \sum (H_{b_{k+1}}^{b_k})^T (P_{b_{k+1}}^{b_k})^{-1} H_{b_{k+1}}^{b_k} + \sum (H_l^{c_j})^T (P_l^{c_j})^{-1} H_l^{c_j}) \delta\chi \\ = b_P + b_B + b_C \tag{156}$$

增量方程可进一步简化为：

$$(\Lambda_P + \Lambda_B + \Lambda_C)\delta\chi = b_P + b_B + b_C \tag{157}$$

其中 Λ_P、Λ_B、Λ_C 为（近似的）Hessian 矩阵。

5.3 IMU 测量约束

5.3.1 IMU 测量残差

根据 IMU 测量模型(117)可写出 IMU 测量的残差 $r_B(\hat{z}_{b_{k+1}}^{b_k}, \chi)$：

$$r_B(\hat{z}_{b_{k+1}}^{b_k}, \chi) = \begin{bmatrix} \delta\alpha_{b_{k+1}}^{b_k} \\ \delta\theta_{b_{k+1}}^{b_k} \\ \delta\beta_{b_{k+1}}^{b_k} \\ \delta b_a \\ \delta b_w \end{bmatrix}$$

$$= \begin{bmatrix} R_w^{b_k}(p_{b_{k+1}}^w - p_{b_k}^w - v_{b_k}^w \Delta t_k + \frac{1}{2} g^w \Delta t_k^2) - \hat{\alpha}_{b_{k+1}}^{b_k} \\ 2\left[(\hat{\gamma}_{b_{k+1}}^{b_k})^{-1} \otimes (q_{b_k}^w)^{-1} \otimes q_{b_{k+1}}^w \right]_{xyz} \\ R_w^{b_k}(v_{b_{k+1}}^w - v_{b_k}^w + g^w \Delta t_k) - \hat{\beta}_{b_{k+1}}^{b_k} \\ b_{ab_{k+1}} - b_{ab_k} \\ b_{wb_{k+1}} - b_{wb_k} \end{bmatrix} \tag{158}$$

其中 $[q]_{xyz}$ 表示提取四元数 q 的虚部，$\left[\hat{\alpha}_{b_{k+1}}^{b_k}, \hat{\beta}_{b_{k+1}}^{b_k}, \hat{\gamma}_{b_{k+1}}^{b_k} \right]^T$ 为关键帧 b_k 和关键帧 b_{k+1} 时间间隔内，仅仅使用含有噪声的加速度计和陀螺仪数据计算的预积分 IMU 测量项，$\delta\theta_{b_{k+1}}^{b_k}$ 是四元数误差的三维表示。

5.3.2 待优化变量

对于两帧之间的 IMU 测量残差，待优化变量为：

$$\left[p_{b_k}^w, \theta_{b_k}^w \right], \left[v_{b_k}^w, b_{a_k}, b_{w_k} \right], \left[p_{b_{k+1}}^w, \theta_{b_{k+1}}^w \right], \left[v_{b_{k+1}}^w, b_{a_{k+1}}, b_{w_{k+1}} \right] \tag{159}$$

其中 $\theta_{b_k}^w$、$\theta_{b_{k+1}}^w$ 分别为 $R_{b_k}^w$、$R_{b_{k+1}}^w$ 对应的李代数，而 $R_{b_{k+1}}^w$、$R_{b_k}^w$ 分别为 $q_{b_k}^w$、$q_{b_{k+1}}^w$ 对应的旋转矩阵。

按照(159)划分变量是为了和 VINS-Mono 代码保持一致。

5.3.3 雅可比矩阵

将两帧之间 IMU 预积分测量残差对应的求偏导对象，按照(159)分为四部分，相应的雅可比矩阵块为：$J[0]^{15\times6}$、$J[1]^{15\times9}$、$J[2]^{15\times6}$、$J[3]^{15\times9}$，它们分别对应 VINS-Mono 代码 imu_factor.h 中的 jacobians[0]、jacobians[1]、jacobians[2]、jacobians[3]。

求 $J[0]^{15\times6}$：

$$\begin{aligned}
\frac{\partial \delta\alpha_{b_{k+1}}^{b_k}}{\partial p_{b_k}^w} &= \frac{\partial(R_w^{b_k}(p_{b_{k+1}}^w - p_{b_k}^w - v_{b_k}^w \Delta t_k + \frac{1}{2}g^w \Delta t_k^2) - \hat{\alpha}_{b_{k+1}}^{b_k})}{\partial p_{b_k}^w} \\
&= -R_w^{b_k}
\end{aligned} \tag{160}$$

$$\frac{\partial \delta \alpha_{b_{k+1}}^{b_k}}{\partial \theta_{b_k}^{w}} = \frac{\partial (R_w^{b_k}(p_{b_{k+1}}^{w} - p_{b_k}^{w} - v_{b_k}^{w}\Delta t_k + \frac{1}{2}g^w\Delta t_k^2) - \hat{\alpha}_{b_{k+1}}^{b_k})}{\partial \theta_{b_k}^{w}}$$

$$= \lim_{\Delta \theta_{b_k}^{w} \to 0} \frac{(R_{b_k}^{w}\exp(\left[\Delta \theta_{b_k}^{w}\right]_{\times}))^T(p_{b_{k+1}}^{w} - p_{b_k}^{w} - v_{b_k}^{w}\Delta t_k + \frac{1}{2}g^w\Delta t_k^2) - (R_{b_k}^{w})^T(p_{b_{k+1}}^{w} - p_{b_k}^{w} - v_{b_k}^{w}\Delta t_k + \frac{1}{2}g^w\Delta t_k^2)}{\Delta \theta_{b_k}^{w}}$$

$$= \lim_{\Delta \theta_{b_k}^{w} \to 0} \frac{(R_{b_k}^{w}(I+\left[\Delta \theta_{b_k}^{w}\right]_{\times}))^T(p_{b_{k+1}}^{w} - p_{b_k}^{w} - v_{b_k}^{w}\Delta t_k + \frac{1}{2}g^w\Delta t_k^2) - (R_{b_k}^{w})^T(p_{b_{k+1}}^{w} - p_{b_k}^{w} - v_{b_k}^{w}\Delta t_k + \frac{1}{2}g^w\Delta t_k^2)}{\Delta \theta_{b_k}^{w}}$$

$$= \lim_{\Delta \theta_{b_k}^{w} \to 0} \frac{(R_{b_k}^{w}\left[\Delta \theta_{b_k}^{w}\right]_{\times})^T(p_{b_{k+1}}^{w} - p_{b_k}^{w} - v_{b_k}^{w}\Delta t_k + \frac{1}{2}g^w\Delta t_k^2)}{\Delta \theta_{b_k}^{w}}$$

$$= \lim_{\Delta \theta_{b_k}^{w} \to 0} \frac{-\left[\Delta \theta_{b_k}^{w}\right]_{\times} R_w^{b_k}(p_{b_{k+1}}^{w} - p_{b_k}^{w} - v_{b_k}^{w}\Delta t_k + \frac{1}{2}g^w\Delta t_k^2)}{\Delta \theta_{b_k}^{w}}$$

$$= \lim_{\Delta \theta_{b_k}^{w} \to 0} \frac{\left[R_w^{b_k}(p_{b_{k+1}}^{w} - p_{b_k}^{w} - v_{b_k}^{w}\Delta t_k + \frac{1}{2}g^w\Delta t_k^2)\right]_{\times}\Delta \theta_{b_k}^{w}}{\Delta \theta_{b_k}^{w}}$$

$$= \left[R_w^{b_k}(p_{b_{k+1}}^{w} - p_{b_k}^{w} - v_{b_k}^{w}\Delta t_k + \frac{1}{2}g^w\Delta t_k^2)\right]_{\times}$$

$$(161)$$

注意(161)中 $\theta_{b_k}^{w}$ 是 $R_{b_k}^{w}$ 对应的李代数，而不是 $R_w^{b_k}$ 对应的李代数，这里很容易弄混，而且这里李代数的求导采用的是右乘形式。

$$\frac{\partial \delta \theta_{b_{k+1}}^{b_k}}{\partial p_{b_k}^{w}} = \frac{\partial (2\left[(\hat{\gamma}_{b_{k+1}}^{b_k})^{-1} \otimes (q_{b_k}^{w})^{-1} \otimes q_{b_{k+1}}^{w}\right]_{xyz})}{\partial p_{b_k}^{w}}$$
$$= 0 \qquad (162)$$

$$\frac{\partial \delta\theta_{b_{k+1}}^{b_k}}{\partial \theta_{b_k}^{w}} = \lim_{\Delta\theta_{b_k}^{w} \to 0} \frac{2\left[(\hat{\gamma}_{b_{k+1}}^{b_k})^{-1} \otimes (q_{b_k}^{w} \otimes \begin{bmatrix} 1 \\ \dfrac{\Delta\theta_{b_k}^{w}}{2} \end{bmatrix})^{-1} \otimes q_{b_{k+1}}^{w}\right]_{xyz} - 2\left[(\hat{\gamma}_{b_{k+1}}^{b_k})^{-1} \otimes (q_{b_k}^{w})^{-1} \otimes q_{b_{k+1}}^{w}\right]_{xyz}}{\Delta\theta_{b_k}^{w}}$$

$$= \lim_{\Delta\theta_{b_k}^{w} \to 0} \frac{2\left[(\hat{\gamma}_{b_{k+1}}^{b_k})^{-1} \otimes (q_{b_k}^{w} \otimes \begin{bmatrix} 1 \\ \dfrac{\Delta\theta_{b_k}^{w}}{2} \end{bmatrix})^{-1} \otimes q_{b_{k+1}}^{w}\right]_{xyz} - 2\left[(\hat{\gamma}_{b_{k+1}}^{b_k})^{-1} \otimes (q_{b_k}^{w})^{-1} \otimes q_{b_{k+1}}^{w}\right]_{xyz}}{\Delta\theta_{b_k}^{w}}$$

$$= \lim_{\Delta\theta_{b_k}^{w} \to 0} \frac{2\left[(\hat{\gamma}_{b_{k+1}}^{b_k})^{-1} \otimes \begin{bmatrix} 1 \\ -\dfrac{\Delta\theta_{b_k}^{w}}{2} \end{bmatrix} \otimes (q_{b_k}^{w})^{-1} \otimes q_{b_{k+1}}^{w}\right]_{xyz}}{\Delta\theta_{b_k}^{w}}$$

$$= \lim_{\Delta\theta_{b_k}^{w} \to 0} \frac{2\left[R((q_{b_k}^{w})^{-1} \otimes q_{b_{k+1}}^{w})L((\hat{\gamma}_{b_{k+1}}^{b_k})^{-1})\begin{bmatrix} 1 \\ -\dfrac{\Delta\theta_{b_k}^{w}}{2} \end{bmatrix}\right]_{xyz}}{\Delta\theta_{b_k}^{w}}$$

$$= \lim_{\Delta\theta_{b_k}^{w} \to 0} \frac{-\left[R((q_{b_k}^{w})^{-1} \otimes q_{b_{k+1}}^{w})L((\hat{\gamma}_{b_{k+1}}^{b_k})^{-1})\right]_{3\times3}\Delta\theta_{b_k}^{w}}{\Delta\theta_{b_k}^{w}} \qquad (163)$$

$$= -\left[R((q_{b_k}^{w})^{-1} \otimes q_{b_{k+1}}^{w})L((\hat{\gamma}_{b_{k+1}}^{b_k})^{-1})\right]_{3\times3}$$

对中 $L(\bullet)$、$R(\bullet)$ 的含义的说明如下，对于四元数 $q = [x \quad y \quad z \quad s] = [w \quad s]$：

$$L(q) = \begin{bmatrix} [w]_{\times} & w \\ -w^T & 0 \end{bmatrix} + sI \qquad (164)$$

$$R(q) = \begin{bmatrix} -[w]_{\times} & w \\ -w^T & 0 \end{bmatrix} + sI \qquad (165)$$

$$L(q^{-1}) = \begin{bmatrix} [-w]_{\times} & -w \\ -(-w)^T & 0 \end{bmatrix} + sI = \begin{bmatrix} -[w]_{\times} & -w \\ w^T & 0 \end{bmatrix} + sI \qquad (166)$$

若只取(166)中虚部对应的 3×3 矩阵子块，则有：

$$L(q^{-1})_{3\times3} = -[w]_{\times} + sI = R(q)_{3\times3} \qquad (167)$$

同理可得：

$$R(q^{-1})_{3\times3} = L(q)_{3\times3} \qquad (168)$$

根据(167)(168)可将(163)进一步写为：

$$\frac{\partial \delta \theta^{b_k}_{b_{k+1}}}{\partial \theta^w_{b_k}} = -\left[R((q^w_{b_k})^{-1} \otimes q^w_{b_{k+1}}) L((\hat{\gamma}^{b_k}_{b_{k+1}})^{-1}) \right]_{3\times3} \tag{169}$$

$$= -\left[L((q^w_{b_{k+1}})^{-1} \otimes q^w_{b_k}) R(\hat{\gamma}^{b_k}_{b_{k+1}}) \right]_{3\times3}$$

$$\frac{\partial \delta \beta^{b_k}_{b_{k+1}}}{\partial p^w_{b_k}} = \frac{\partial (R^{b_k}_w (v^w_{b_{k+1}} - v^w_{b_k} + g^w \Delta t_k) - \hat{\beta}^{b_k}_{b_{k+1}})}{\partial p^w_{b_k}} \tag{170}$$

$$= 0$$

与(161)中 $\frac{\partial \delta \alpha^{b_k}_{b_{k+1}}}{\partial \theta^w_{b_k}}$ 的推导类似，可以推导出 $\frac{\partial \delta \beta^{b_k}_{b_{k+1}}}{\partial \theta^w_{b_k}}$：

$$\frac{\partial \delta \beta^{b_k}_{b_{k+1}}}{\partial \theta^w_{b_k}} = \frac{\partial (R^{b_k}_w (v^w_{b_{k+1}} - v^w_{b_k} + g^w \Delta t_k) - \hat{\beta}^{b_k}_{b_{k+1}})}{\partial \theta^w_{b_k}} \tag{171}$$

$$= \left[R^{b_k}_w (v^w_{b_{k+1}} - v^w_{b_k} + g^w \Delta t_k) \right]_{\times}$$

$$\frac{\partial \delta b_a}{\partial p^w_{b_k}} = \frac{\partial (b_{a_{k+1}} - b_{a_k})}{\partial p^w_{b_k}} \tag{172}$$

$$= 0$$

$$\frac{\partial \delta b_a}{\partial \theta^w_{b_k}} = \frac{\partial (b_{a_{k+1}} - b_{a_k})}{\partial \theta^w_{b_k}} \tag{173}$$

$$= 0$$

$$\frac{\partial \delta b_w}{\partial p^w_{b_k}} = \frac{\partial (b_{w_{k+1}} - b_{w_k})}{\partial p^w_{b_k}} \tag{174}$$

$$= 0$$

$$\frac{\partial \delta b_w}{\partial \theta^w_{b_k}} = \frac{\partial (b_{w_{k+1}} - b_{w_k})}{\partial \theta^w_{b_k}} \tag{175}$$

$$= 0$$

由(160)(161)(162)(169)(170)(171)(172)(173)(174)(175)可得：

$$J[0]^{15\times6} = \begin{bmatrix} \dfrac{\partial r_B}{\partial p^w_{b_k}} & \dfrac{\partial r_B}{\partial \theta^w_{b_k}} \end{bmatrix} = \begin{bmatrix} -R^{b_k}_w & \left[R^{b_k}_w (p^w_{b_{k+1}} - p^w_{b_k} - v^w_{b_k} \Delta t_k + \dfrac{1}{2} g^w \Delta t^2_k) \right]_{\times} \\ 0 & -\left[L((q^w_{b_{k+1}})^{-1} \otimes q^w_{b_k}) R(\hat{\gamma}^{b_k}_{b_{k+1}}) \right]_{3\times3} \\ 0 & \left[R^{b_k}_w (v^w_{b_{k+1}} - v^w_{b_k} + g^w \Delta t_k) \right]_{\times} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{176}$$

$J[0]^{15\times6}$ 对应的 imu_factor.h 中的代码片段为：

```
if (jacobians[0])
{
    Eigen::Map<Eigen::Matrix<double, 15, 7, Eigen::RowMajor>> jacobian_pose_i(jacobians[0]);
    jacobian_pose_i.setZero();

    jacobian_pose_i.block<3, 3>(O_P, O_P) = -Qi.inverse().toRotationMatrix();
    jacobian_pose_i.block<3, 3>(O_P, O_R) = Utility::skewSymmetric(Qi.inverse() * (0.5 * G * sum_dt * sum_dt + Pj - Pi - Vi * sum_dt));

    jacobian_pose_i.block<3, 3>(O_R, O_R) = -(Qj.inverse() * Qi).toRotationMatrix();

    Eigen::Quaterniond corrected_delta_q = pre_integration->delta_q * Utility::deltaQ(dq_dbg * (Bgi - pre_integration->linearized_bg));
    jacobian_pose_i.block<3, 3>(O_R, O_R) = -(Utility::Qleft(Qj.inverse() * Qi) * Utility::Qright(corrected_delta_q)).bottomRightCorner<3, 3>();

    jacobian_pose_i.block<3, 3>(O_V, O_R) = Utility::skewSymmetric(Qi.inverse() * (G * sum_dt + Vj - Vi));

    jacobian_pose_i = sqrt_info * jacobian_pose_i;

    if (jacobian_pose_i.maxCoeff() > 1e8 || jacobian_pose_i.minCoeff() < -1e8)
    {
        ROS_WARN("numerical unstable in preintegration");
        //std::cout << sqrt_info << std::endl;
        ROS_BREAK();
    }
}
```

求 $J[1]^{15\times9}$ ：

$$\frac{\partial\delta\alpha_{b_{k+1}}^{b_k}}{\partial v_{b_k}^w} = \frac{\partial(R_w^{b_k}(p_{b_{k+1}}^w - p_{b_k}^w - v_{b_k}^w\Delta t_k + \frac{1}{2}g^w\Delta t_k^2) - \hat{\alpha}_{b_{k+1}}^{b_k})}{\partial v_{b_k}^w} \tag{177}$$
$$= -R_w^{b_k}\Delta t_k$$

$\dfrac{\partial\delta\alpha_{b_{k+1}}^{b_k}}{\partial b_{a_k}}$ 、 $\dfrac{\partial\delta\alpha_{b_{k+1}}^{b_k}}{\partial b_{w_k}}$ 可以从(109)的大雅可比矩阵中获取。

由(109)可得：

$$J_{b_a}^\alpha = \frac{\partial\delta\alpha_{b_{k+1}}^{b_k}}{\partial\delta b_a} \tag{178}$$

$$J_{b_w}^\alpha = \frac{\partial\delta\alpha_{b_{k+1}}^{b_k}}{\partial\delta b_w} \tag{179}$$

由(178)可得：

$$\partial\delta\alpha_{b_{k+1}}^{b_k} = J_{b_a}^\alpha\partial\delta b_a$$
$$= J_{b_a}^\alpha\partial\delta(b_{a(true)} - b_a) \tag{180}$$
$$= J_{b_a}^\alpha\partial\delta b_{a(true)} - J_{b_a}^\alpha\partial\delta b_a$$

等式(180)中 $\delta b_a = \delta(b_{a(true)} - b_a)$ 的关系是我自己脑补的，是根据等式(19)得来的，不

知道对不对。但是根据(158)中的第四行又有 $\delta b_a = b_{ab_{k+1}} - b_{ab_k}$ ，不知道到底应该采用哪一个关系？？

由(180)可得：

$$\frac{\partial \delta \alpha_{b_{k+1}}^{b_k}}{\partial b_{a_k}} = -J_{b_a}^{\alpha} \tag{181}$$

同理可得：

$$\frac{\partial \delta \alpha_{b_{k+1}}^{b_k}}{\partial b_{w_k}} = -J_{b_w}^{\alpha} \tag{182}$$

$$\frac{\partial \delta \theta_{b_{k+1}}^{b_k}}{\partial v_{b_k}^w} = \frac{\partial (2 \left[(\hat{\gamma}_{b_{k+1}}^{b_k})^{-1} \otimes (q_{b_k}^w)^{-1} \otimes q_{b_{k+1}}^w \right]_{xyz})}{\partial v_{b_k}^w} \tag{183}$$
$$= 0$$

$$\frac{\partial \delta \theta_{b_{k+1}}^{b_k}}{\partial b_{a_k}} = \frac{\partial (2 \left[(\hat{\gamma}_{b_{k+1}}^{b_k})^{-1} \otimes (q_{b_k}^w)^{-1} \otimes q_{b_{k+1}}^w \right]_{xyz})}{\partial b_{a_k}} \tag{184}$$
$$= 0$$

$$\frac{\partial \delta \theta_{b_{k+1}}^{b_k}}{\partial b_{w_k}} = \frac{\partial (2 \left[(\hat{\gamma}_{b_{k+1}}^{b_k})^{-1} \otimes (q_{b_k}^w)^{-1} \otimes q_{b_{k+1}}^w \right]_{xyz})}{\partial b_{w_k}}$$

$$= \lim_{\Delta b_{w_k} \to 0} \frac{2 \left[(\hat{\gamma}_{b_{k+1}}^{b_k} \otimes \begin{bmatrix} 1 \\ \frac{1}{2} J_{b_w}^{\gamma} \Delta b_{w_k} \end{bmatrix})^{-1} \otimes (q_{b_k}^w)^{-1} \otimes q_{b_{k+1}}^w \right]_{xyz} - 2 \left[(\hat{\gamma}_{b_{k+1}}^{b_k})^{-1} \otimes (q_{b_k}^w)^{-1} \otimes q_{b_{k+1}}^w \right]_{xyz}}{\Delta b_{w_k}}$$

$$= \lim_{\Delta b_{w_k} \to 0} \frac{\left[\begin{bmatrix} 0 \\ -J_{b_w}^{\gamma} \Delta b_{w_k} \end{bmatrix} (\hat{\gamma}_{b_{k+1}}^{b_k})^{-1} \otimes (q_{b_k}^w)^{-1} \otimes q_{b_{k+1}}^w \right]_{xyz}}{\Delta b_{w_k}}$$

$$= \lim_{\Delta b_{w_k} \to 0} \frac{-\left[R((\hat{\gamma}_{b_{k+1}}^{b_k})^{-1} \otimes (q_{b_k}^w)^{-1} \otimes q_{b_{k+1}}^w) \right]_{3 \times 3} J_{b_w}^{\gamma} \Delta b_{w_k}}{\Delta b_{w_k}}$$

$$= -\left[R((\hat{\gamma}_{b_{k+1}}^{b_k})^{-1} \otimes (q_{b_k}^w)^{-1} \otimes q_{b_{k+1}}^w) \right]_{3 \times 3} J_{b_w}^{\gamma}$$

$$= -\left[L((q_{b_{k+1}}^w)^{-1} \otimes q_{b_k}^w \otimes \hat{\gamma}_{b_{k+1}}^{b_k}) \right]_{3 \times 3} J_{b_w}^{\gamma} \tag{185}$$

$$\frac{\partial \delta \beta_{b_{k+1}}^{b_k}}{\partial v_{b_k}^w} = \frac{\partial (R_w^{b_k}(v_{b_{k+1}}^w - v_{b_k}^w + g^w \Delta t_k) - \hat{\beta}_{b_{k+1}}^{b_k})}{\partial v_{b_k}^w} \tag{186}$$
$$= -R_w^{b_k}$$

$\dfrac{\partial \delta \beta_{b_{k+1}}^{b_k}}{\partial b_{a_k}}$、$\dfrac{\partial \delta \beta_{b_{k+1}}^{b_k}}{\partial b_{w_k}}$ 同理可从(109)的大雅可比矩阵中获取：

$$\frac{\partial \delta \beta_{b_{k+1}}^{b_k}}{\partial b_{a_k}} = -J_{b_a}^{\beta} \tag{187}$$

$$\frac{\partial \delta \beta_{b_{k+1}}^{b_k}}{\partial b_{w_k}} = -J_{b_w}^{\beta} \tag{188}$$

$$\begin{aligned} \frac{\partial \delta b_a}{\partial v_{b_k}^w} &= \frac{\partial (b_{a_{k+1}} - b_{a_k})}{\partial v_{b_k}^w} \\ &= 0 \end{aligned} \tag{189}$$

$$\begin{aligned} \frac{\partial \delta b_a}{\partial b_{a_k}} &= \frac{\partial (b_{a_{k+1}} - b_{a_k})}{\partial b_{a_k}} \\ &= -I \end{aligned} \tag{190}$$

$$\begin{aligned} \frac{\partial \delta b_a}{\partial b_{w_k}} &= \frac{\partial (b_{a_{k+1}} - b_{a_k})}{\partial b_{w_k}} \\ &= 0 \end{aligned} \tag{191}$$

$$\begin{aligned} \frac{\partial \delta b_w}{\partial v_{b_k}^w} &= \frac{\partial (b_{w_{k+1}} - b_{w_k})}{\partial v_{b_k}^w} \\ &= 0 \end{aligned} \tag{192}$$

$$\begin{aligned} \frac{\partial \delta b_w}{\partial b_{a_k}} &= \frac{\partial (b_{w_{k+1}} - b_{w_k})}{\partial b_{a_k}} \\ &= 0 \end{aligned} \tag{193}$$

$$\begin{aligned} \frac{\partial \delta b_w}{\partial b_{w_k}} &= \frac{\partial (b_{w_{k+1}} - b_{w_k})}{\partial b_{w_k}} \\ &= -I \end{aligned} \tag{194}$$

由(177)(181)(182)(183)(184)(185)(186)(187)(188)(189)(190)(191)(192)(193)(194)可得：

$$J[1]^{15 \times 9} = \begin{bmatrix} \dfrac{\partial r_B}{\partial v_{b_k}^w} & \dfrac{\partial r_B}{\partial b_{a_k}} & \dfrac{\partial r_B}{\partial b_{w_k}} \end{bmatrix} = \begin{bmatrix} -R_w^{b_k} \Delta t_k & -J_{b_a}^{\alpha} & -J_{b_w}^{\alpha} \\ 0 & 0 & -\left[L((q_{b_{k+1}}^w)^{-1} \otimes q_{b_k}^w \otimes \hat{\gamma}_{b_{k+1}}^{b_k}) \right]_{3\times 3} J_{b_w}^{\gamma} \\ -R_w^{b_k} & -J_{b_a}^{\beta} & -J_{b_w}^{\beta} \\ 0 & -I & 0 \\ 0 & 0 & -I \end{bmatrix} \tag{195}$$

$J[1]^{15 \times 9}$ 对应的 imu_factor.h 中的代码片段为：

```
if (jacobians[1])
{
    Eigen::Map<Eigen::Matrix<double, 15, 9, Eigen::RowMajor>> jacobian_speedbias_i(jacobians[1]);
    jacobian_speedbias_i.setZero();
    jacobian_speedbias_i.block<3, 3>(O_P, O_V - O_V) = -Qi.inverse().toRotationMatrix() * sum_dt;
    jacobian_speedbias_i.block<3, 3>(O_P, O_BA - O_V) = -dp_dba;
    jacobian_speedbias_i.block<3, 3>(O_P, O_BG - O_V) = -dp_dbg;


    jacobian_speedbias_i.block<3, 3>(O_R, O_BG - O_V) = -dq_dbg;

    Eigen::Quaterniond corrected_delta_q = pre_integration->delta_q * Utility::deltaQ(dq_dbg * (Bgi - pre_integration->linearized_bg));
    jacobian_speedbias_i.block<3, 3>(O_R, O_BG - O_V) = -Utility::Qleft(Qj.inverse() * Qi * corrected_delta_q).bottomRightCorner<3, 3>() * dq_dbg;


    jacobian_speedbias_i.block<3, 3>(O_V, O_V - O_V) = -Qi.inverse().toRotationMatrix();
    jacobian_speedbias_i.block<3, 3>(O_V, O_BA - O_V) = -dv_dba;
    jacobian_speedbias_i.block<3, 3>(O_V, O_BG - O_V) = -dv_dbg;

    jacobian_speedbias_i.block<3, 3>(O_BA, O_BA - O_V) = -Eigen::Matrix3d::Identity();

    jacobian_speedbias_i.block<3, 3>(O_BG, O_BG - O_V) = -Eigen::Matrix3d::Identity();

    jacobian_speedbias_i = sqrt_info * jacobian_speedbias_i;

    ROS_ASSERT(fabs(jacobian_speedbias_i.maxCoeff()) < 1e8);
    ROS_ASSERT(fabs(jacobian_speedbias_i.minCoeff()) < 1e8);
}
```

求 $J[2]^{15\times6}$：

$$\frac{\partial \delta \alpha_{b_{k+1}}^{b_k}}{\partial p_{b_{k+1}}^w} = \frac{\partial (R_w^{b_k}(p_{b_{k+1}}^w - p_{b_k}^w - v_{b_k}^w \Delta t_k + \frac{1}{2} g^w \Delta t_k^2) - \hat{\alpha}_{b_{k+1}}^{b_k})}{\partial p_{b_{k+1}}^w} \tag{196}$$

$$= R_w^{b_k}$$

$$\frac{\partial \delta \alpha_{b_{k+1}}^{b_k}}{\partial \theta_{b_{k+1}}^w} = \frac{\partial (R_w^{b_k}(p_{b_{k+1}}^w - p_{b_k}^w - v_{b_k}^w \Delta t_k + \frac{1}{2} g^w \Delta t_k^2) - \hat{\alpha}_{b_{k+1}}^{b_k})}{\partial \theta_{b_{k+1}}^w} \tag{197}$$

$$= 0$$

$$\frac{\partial \delta \theta_{b_{k+1}}^{b_k}}{\partial p_{b_{k+1}}^w} = \frac{\partial (2 \left[(\hat{\gamma}_{b_{k+1}}^{b_k})^{-1} \otimes (q_{b_k}^w)^{-1} \otimes q_{b_{k+1}}^w \right]_{xyz})}{\partial p_{b_{k+1}}^w} \tag{198}$$

$$= 0$$

$$\frac{\partial \delta \theta_{b_{k+1}}^{b_k}}{\partial \theta_{b_{k+1}}^w} = \lim_{\Delta \theta_{b_{k+1}}^w \to 0} \frac{2\left[(\hat{\gamma}_{b_{k+1}}^{b_k})^{-1} \otimes (q_{b_k}^w)^{-1} \otimes (q_{b_{k+1}}^w \otimes \begin{bmatrix} 1 \\ \dfrac{\Delta \theta_{b_{k+1}}^w}{2} \end{bmatrix})\right]_{xyz} - 2\left[(\hat{\gamma}_{b_{k+1}}^{b_k})^{-1} \otimes (q_{b_k}^w)^{-1} \otimes q_{b_{k+1}}^w\right]_{xyz}}{\Delta \theta_{b_{k+1}}^w}$$

$$= \lim_{\Delta \theta_{b_{k+1}}^w \to 0} \frac{\left[(\hat{\gamma}_{b_{k+1}}^{b_k})^{-1} \otimes (q_{b_k}^w)^{-1} \otimes (q_{b_{k+1}}^w \otimes \begin{bmatrix} 0 \\ \Delta \theta_{b_{k+1}}^w \end{bmatrix})\right]_{xyz}}{\Delta \theta_{b_{k+1}}^w}$$

$$= \lim_{\Delta \theta_{b_{k+1}}^w \to 0} \frac{\left[L((\hat{\gamma}_{b_{k+1}}^{b_k})^{-1} \otimes (q_{b_k}^w)^{-1} \otimes q_{b_{k+1}}^w)\right]_{3\times 3} \Delta \theta_{b_{k+1}}^w}{\Delta \theta_{b_{k+1}}^w}$$

$$= \left[L((\hat{\gamma}_{b_{k+1}}^{b_k})^{-1} \otimes (q_{b_k}^w)^{-1} \otimes q_{b_{k+1}}^w)\right]_{3\times 3}$$

$$\tag{199}$$

$$\frac{\partial \delta \beta_{b_{k+1}}^{b_k}}{\partial p_{b_{k+1}}^w} = \frac{\partial (R_w^{b_k}(v_{b_{k+1}}^w - v_{b_k}^w + g^w \Delta t_k) - \hat{\beta}_{b_{k+1}}^{b_k})}{\partial p_{b_{k+1}}^w} \tag{200}$$
$$= 0$$

$$\frac{\partial \delta \beta_{b_{k+1}}^{b_k}}{\partial \theta_{b_{k+1}}^w} = \frac{\partial (R_w^{b_k}(v_{b_{k+1}}^w - v_{b_k}^w + g^w \Delta t_k) - \hat{\beta}_{b_{k+1}}^{b_k})}{\partial \theta_{b_{k+1}}^w} \tag{201}$$
$$= 0$$

$$\frac{\partial \delta b_a}{\partial p_{b_{k+1}}^w} = \frac{\partial (b_{a_{k+1}} - b_{a_k})}{\partial p_{b_{k+1}}^w} \tag{202}$$
$$= 0$$

$$\frac{\partial \delta b_a}{\partial \theta_{b_{k+1}}^w} = \frac{\partial (b_{a_{k+1}} - b_{a_k})}{\partial \theta_{b_{k+1}}^w} \tag{203}$$
$$= 0$$

$$\frac{\partial \delta b_w}{\partial p_{b_{k+1}}^w} = \frac{\partial (b_{w_{k+1}} - b_{w_k})}{\partial p_{b_{k+1}}^w} \tag{204}$$
$$= 0$$

$$\frac{\partial \delta b_w}{\partial \theta_{b_{k+1}}^w} = \frac{\partial (b_{w_{k+1}} - b_{w_k})}{\partial \theta_{b_{k+1}}^w} \tag{205}$$
$$= 0$$

由(196)(197)(198)(199)(200)(201)(202)(203)(204)(205)可得：

$$J\left[2\right]^{15\times6}=\begin{bmatrix}\dfrac{\partial r_B}{\partial p_{b_{k+1}}^w}&\dfrac{\partial r_B}{\partial \theta_{b_{k+1}}^w}\end{bmatrix}=\begin{bmatrix}R_w^{b_k}&0\\0&\left[L((\hat{\gamma}_{b_{k+1}}^{b_k})^{-1}\otimes(q_{b_k}^w)^{-1}\otimes q_{b_{k+1}}^w)\right]_{3\times3}\\0&0\\0&0\\0&0\end{bmatrix}\tag{206}$$

$J\left[2\right]^{15\times7}$ 对应的 imu_factor.h 中的代码片段为：

```
if (jacobians[2])
{
    Eigen::Map<Eigen::Matrix<double, 15, 7, Eigen::RowMajor>> jacobian_pose_j(jacobians[2]);
    jacobian_pose_j.setZero();

    jacobian_pose_j.block<3, 3>(O_P, O_P) = Qi.inverse().toRotationMatrix();

    jacobian_pose_j.block<3, 3>(O_R, O_R) = Eigen::Matrix3d::Identity();

    Eigen::Quaterniond corrected_delta_q = pre_integration->delta_q * Utility::deltaQ(dq_dbg * (Bgi - pre_integration->linearized_bg));
    jacobian_pose_j.block<3, 3>(O_R, O_R) = Utility::Qleft(corrected_delta_q.inverse() * Qi.inverse() * Qj).bottomRightCorner<3, 3>();

    jacobian_pose_j = sqrt_info * jacobian_pose_j;

    ROS_ASSERT(fabs(jacobian_pose_j.maxCoeff()) < 1e8);
    ROS_ASSERT(fabs(jacobian_pose_j.minCoeff()) < 1e8);
}
```

求 $J\left[3\right]^{15\times9}$：

$$\frac{\partial \delta\alpha_{b_{k+1}}^{b_k}}{\partial v_{b_{k+1}}^w}=\frac{\partial(R_w^{b_k}(p_{b_{k+1}}^w-p_{b_k}^w-v_{b_k}^w\Delta t_k+\frac{1}{2}g^w\Delta t_k^2)-\hat{\alpha}_{b_{k+1}}^{b_k})}{\partial v_{b_{k+1}}^w}\tag{207}$$
$$=0$$

$\dfrac{\partial \delta\alpha_{b_{k+1}}^{b_k}}{\partial b_{a_{k+1}}}$、$\dfrac{\partial \delta\alpha_{b_{k+1}}^{b_k}}{\partial b_{w_{k+1}}}$ 可以从(109)得大雅可比矩阵中获取，而实际上根据等式(78)(86)，

可以认为连续两个关键帧 k、$k+1$ 之间 IMU 的约束与 $\delta b_{a_{i+1}}$、$\delta b_{w_{i+1}}$ 无关，仅与 δb_{a_i}、δb_{w_i}

有关，所以：

$$\frac{\partial \delta\alpha_{b_{k+1}}^{b_k}}{\partial b_{a_{k+1}}}=0\tag{208}$$

$$\frac{\partial \delta\alpha_{b_{k+1}}^{b_k}}{\partial b_{w_{k+1}}}=0\tag{209}$$

$$\frac{\partial \delta\theta_{b_{k+1}}^{b_k}}{\partial v_{b_{k+1}}^w}=\frac{\partial(2\left[(\hat{\gamma}_{b_{k+1}}^{b_k})^{-1}\otimes(q_{b_k}^w)^{-1}\otimes q_{b_{k+1}}^w\right]_{xyz})}{\partial v_{b_{k+1}}^w}\tag{210}$$
$$=0$$

$$\frac{\partial \delta \theta_{b_{k+1}}^{b_k}}{\partial b_{a_{k+1}}} = \frac{\partial (2 \left[(\hat{\gamma}_{b_{k+1}}^{b_k})^{-1} \otimes (q_{b_k}^w)^{-1} \otimes q_{b_{k+1}}^w \right]_{xyz})}{\partial b_{a_{k+1}}} \tag{211}$$
$$= 0$$

$$\frac{\partial \delta \theta_{b_{k+1}}^{b_k}}{\partial b_{w_{k+1}}} = \frac{\partial (2 \left[(\hat{\gamma}_{b_{k+1}}^{b_k})^{-1} \otimes (q_{b_k}^w)^{-1} \otimes q_{b_{k+1}}^w \right]_{xyz})}{\partial b_{w_{k+1}}} \tag{212}$$
$$= 0$$

$$\frac{\partial \delta \beta_{b_{k+1}}^{b_k}}{\partial v_{b_{k+1}}^w} = \frac{\partial (R_w^{b_k} (v_{b_{k+1}}^w - v_{b_k}^w + g^w \Delta t_k) - \hat{\beta}_{b_{k+1}}^{b_k})}{\partial v_{b_{k+1}}^w} \tag{213}$$
$$= R_w^{b_k}$$

$$\frac{\partial \delta \beta_{b_{k+1}}^{b_k}}{\partial b_{a_{k+1}}} = 0 \tag{214}$$

$$\frac{\partial \delta \beta_{b_{k+1}}^{b_k}}{\partial b_{w_{k+1}}} = 0 \tag{215}$$

$$\frac{\partial \delta b_a}{\partial v_{b_{k+1}}^w} = \frac{\partial (b_{a_{k+1}} - b_{a_k})}{\partial v_{b_{k+1}}^w} \tag{216}$$
$$= 0$$

$$\frac{\partial \delta b_a}{\partial b_{a_{k+1}}} = \frac{\partial (b_{a_{k+1}} - b_{a_k})}{\partial b_{a_{k+1}}} \tag{217}$$
$$= I$$

$$\frac{\partial \delta b_a}{\partial b_{w_{k+1}}} = \frac{\partial (b_{a_{k+1}} - b_{a_k})}{\partial b_{w_{k+1}}} \tag{218}$$
$$= 0$$

$$\frac{\partial \delta b_w}{\partial v_{b_{k+1}}^w} = \frac{\partial (b_{w_{k+1}} - b_{w_k})}{\partial v_{b_{k+1}}^w} \tag{219}$$
$$= 0$$

$$\frac{\partial \delta b_w}{\partial b_{a_{k+1}}} = \frac{\partial (b_{w_{k+1}} - b_{w_k})}{\partial b_{a_{k+1}}} \tag{220}$$
$$= 0$$

$$\frac{\partial \delta b_w}{\partial b_{w_{k+1}}} = \frac{\partial (b_{w_{k+1}} - b_{w_k})}{\partial b_{w_{k+1}}} \tag{221}$$
$$= I$$

由(207)(208)(209)(210)(211)(212)(213)(214)(215)(216)(217)(218)(219)(220)(221)可得：

$$J\left[3\right]^{15\times 9} = \begin{bmatrix} \dfrac{\partial r_B}{\partial v_{b_{k+1}}^w} & \dfrac{\partial r_B}{\partial b_{a_{k+1}}} & \dfrac{\partial r_B}{\partial b_{w_{k+1}}} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ R_w^{b_k} & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \tag{222}$$

$J\left[3\right]^{15\times 9}$ 对应的 imu_factor.h 中的代码片段为：

```cpp
if (jacobians[3])
{
    Eigen::Map<Eigen::Matrix<double, 15, 9, Eigen::RowMajor>> jacobian_speedbias_j(jacobians[3]);
    jacobian_speedbias_j.setZero();

    jacobian_speedbias_j.block<3, 3>(O_V, O_V - O_V) = Qi.inverse().toRotationMatrix();

    jacobian_speedbias_j.block<3, 3>(O_BA, O_BA - O_V) = Eigen::Matrix3d::Identity();

    jacobian_speedbias_j.block<3, 3>(O_BG, O_BG - O_V) = Eigen::Matrix3d::Identity();

    jacobian_speedbias_j = sqrt_info * jacobian_speedbias_j;

    ROS_ASSERT(fabs(jacobian_speedbias_j.maxCoeff()) < 1e8);
    ROS_ASSERT(fabs(jacobian_speedbias_j.minCoeff()) < 1e8);
}
```

在 imu_factor.h 中，残差还需要乘以一个信息矩阵 sqrt_info，这是因为目标函数中使用的是残差的马氏距离 $d = r^T P^{-1} r$，其中 P 是协方差矩阵，P^{-1} 就是对应的信息矩阵。而 Ceres 只接受最小二乘优化，即 $\min(e^T e)$，所以把 P^{-1} 做 LLT 分解，即 $P^{-1} = LL^T$，然后：

$$d = r^T P^{-1} r = r^T LL^T r = (L^T r)^T L^T r \tag{223}$$

令 $L^T r$ 作为新的残差，这样就可以使用 Ceres 进行优化求解了。马氏距离相当于一个残差加权，协方差大的加权小，加权小表示对该数据的信任程度越低，协方差小的加权大，加权大表示对该数据的信任程度越高。代码中 sqrt_info.setIdentity() 表示使用单位阵作为信息矩阵，实际上就是不加权。

5.3.4 协方差矩阵

(223) 中的协方差矩阵 P 可由等式 (110) 计算得到。

5.4 视觉测量约束

5.4.1 视觉测量残差

考虑第 i 帧图像中首次观测到的路标点 l，在第 j 帧图像对路标点 l 的视觉测量残差

定义为：

$$r_B(\hat{z}_l^{c_j}, \chi) = \begin{bmatrix} b_1 & b_2 \end{bmatrix}^T \cdot (\frac{p_l^{c_j}}{\left\| p_l^{c_j} \right\|} - \hat{\bar{p}}_l^{c_j}) \tag{224}$$

等式(224)与 VINS-Mono 论文中的公式稍有区别，区别在于括号内求残差时减法的顺序，论文中为 $\hat{\bar{p}}_l^{c_j} - \frac{p_l^{c_j}}{\left\| p_l^{c_j} \right\|}$，而 VINS-Mono 代码中为 $\frac{p_l^{c_j}}{\left\| p_l^{c_j} \right\|} - \hat{\bar{p}}_l^{c_j}$，这个减法的顺序并不影响优化结果。此处，我们与 VINS-Mono 代码中的习惯保持一致。

(224)中的 $\hat{\bar{p}}_l^{c_j}$ 由等式(225)计算得到：

$$\hat{\bar{p}}_l^{c_j} = \pi_c^{-1}(\begin{bmatrix} \hat{u}_l^{c_j} \\ \hat{v}_l^{c_j} \end{bmatrix}) \tag{225}$$

$\begin{bmatrix} \hat{u}_l^{c_j}, \hat{v}_l^{c_j} \end{bmatrix}$ 是第 j 帧图像对路标点 l 的观测值，即路标点 l 在第 j 帧图像中的像素坐标。π_c^{-1} 表示反投影操作，通过相机内参将路标点的像素坐标，转换为相应时刻相机坐标系下的一个单位向量。

(224)中的 $P_l^{c_j}$ 是路标点 l 在第 j 帧图像对应时刻相机坐标系下的可能坐标。$p_l^{c_j}$ 由等式(226)计算得到：

$$
\begin{aligned}
p_l^{c_j} = R_b^c(\\
R_w^{b_j}(\\
R_{b_i}^w(\\
R_c^b \frac{1}{\lambda_l} \pi_c^{-1}(\begin{bmatrix} \hat{u}_l^{c_i} \\ \hat{v}_l^{c_i} \end{bmatrix}) \\
+ p_c^b) \\
+ p_{b_i}^w - p_{b_j}^w) \\
- p_c^b)
\end{aligned}
\tag{226}
$$

其中 λ_l 是路标点 l 在第 i 帧图像中首次观测到时，它在相应时刻相机坐标系中的逆深度。

等式(226)的推导过程如下：

设路标点 l 在世界坐标系 w 中的坐标为 p_l^w，p_l^w 在第 i 帧图像中的像素坐标为：

$$\begin{bmatrix} \hat{u}_l^{c_i} \\ \hat{v}_l^{c_i} \end{bmatrix} = \lambda_l \pi_c (T_b^c T_w^{b_i} p_l^w) \tag{227}$$

其中 π_c 表示正向投影操作，通过相机内参将路标点在相机坐标系下单位向量化后的坐标，转换为相应像素坐标系下的像素坐标。

由等式(227)可得：

$$\begin{aligned} p_l^w &= T_{b_i}^w T_c^b \frac{1}{\lambda_l} \pi_c^{-1} \left(\begin{bmatrix} \hat{u}_l^{c_i} \\ \hat{v}_l^{c_i} \end{bmatrix} \right) \\ &= R_{b_i}^w (R_c^b \frac{1}{\lambda_l} \pi_c^{-1} \left(\begin{bmatrix} \hat{u}_l^{c_i} \\ \hat{v}_l^{c_i} \end{bmatrix} \right) + p_c^b) + p_{b_i}^w \end{aligned} \tag{228}$$

p_l^w 在第 i 帧图像对应时刻相机坐标系下的坐标为：

$$p_l^{c_j} = T_b^c T_w^{b_j} p_l^w \tag{229}$$

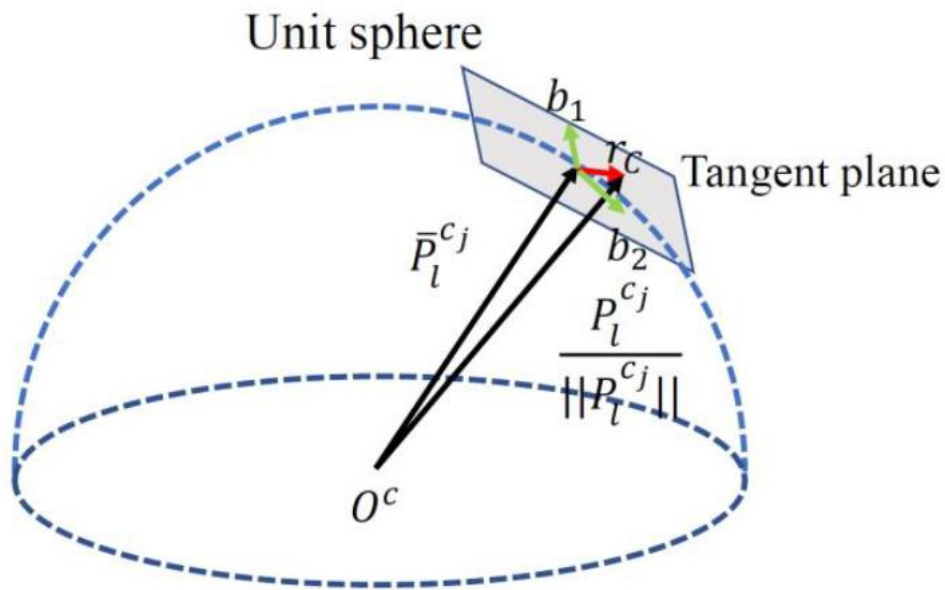由(229)可得：

$$\begin{aligned} p_l^w &= T_{b_j}^w T_c^b p_l^{c_j} \\ &= R_{b_j}^w (R_c^b p_l^{c_j} + p_c^b) + p_{b_j}^w \end{aligned} \tag{230}$$

将等式(228)代入等式(230)：

$$\begin{aligned} R_{b_i}^w (R_c^b \frac{1}{\lambda_l} \pi_c^{-1} \left(\begin{bmatrix} \hat{u}_l^{c_i} \\ \hat{v}_l^{c_i} \end{bmatrix} \right) + p_c^b) + p_{b_i}^w &= p_l^w = R_{b_j}^w (R_c^b p_l^{c_j} + p_c^b) + p_{b_j}^w \Rightarrow \\ p_l^{c_j} = R_b^c (R_w^{b_j} (R_{b_i}^w (R_c^b \frac{1}{\lambda_l} \pi_c^{-1} \left(\begin{bmatrix} \hat{u}_l^{c_i} \\ \hat{v}_l^{c_i} \end{bmatrix} \right) + p_c^b) &+ p_{b_i}^w - p_{b_j}^w) - p_c^b) \end{aligned} \tag{231}$$

至此，等式(226)推导完毕。

在单位球面上的视觉残差

VINS-Mono 将相机坐标系下路标点的坐标单位向量化，然后在单位球上计算重投影误差，所以视觉测量残差的自由度为 2。将初始的 3 维视觉测量残差 $\hat{\bar{p}}_l^{c_j} - \dfrac{p_l^{c_j}}{\left\|p_l^{c_j}\right\|}$ 投影到正切平面上，b_1、b_2 为正切平面上的任意一对正交基。

这一部分内容对应的 projection_factor.cpp 中的代码片段为：

```cpp
//视觉重投影的残差
bool ProjectionFactor::Evaluate(double const *const *parameters, double *residuals, double **jacobians) const
{
    TicToc tic_toc;
    Eigen::Vector3d Pi(parameters[0][0], parameters[0][1], parameters[0][2]);
    Eigen::Quaterniond Qi(parameters[0][6], parameters[0][3], parameters[0][4], parameters[0][5]);

    Eigen::Vector3d Pj(parameters[1][0], parameters[1][1], parameters[1][2]);
    Eigen::Quaterniond Qj(parameters[1][6], parameters[1][3], parameters[1][4], parameters[1][5]);

    Eigen::Vector3d tic(parameters[2][0], parameters[2][1], parameters[2][2]);
    Eigen::Quaterniond qic(parameters[2][6], parameters[2][3], parameters[2][4], parameters[2][5]);

    double inv_dep_i = parameters[3][0];

    Eigen::Vector3d pts_camera_i = pts_i / inv_dep_i;
    Eigen::Vector3d pts_imu_i = qic * pts_camera_i + tic;
    Eigen::Vector3d pts_w = Qi * pts_imu_i + Pi;
    Eigen::Vector3d pts_imu_j = Qj.inverse() * (pts_w - Pj);
    Eigen::Vector3d pts_camera_j = qic.inverse() * (pts_imu_j - tic);
    Eigen::Map<Eigen::Vector2d> residual(residuals);

#ifdef UNIT_SPHERE_ERROR
    residual =  tangent_base * (pts_camera_j.normalized() - pts_j.normalized());
#else
    double dep_j = pts_camera_j.z();
    residual = (pts_camera_j / dep_j).head<2>() - pts_j.head<2>();
#endif
```

5.4.2 待优化变量

对于视觉测量残差，待优化的变量为：

$$\left[p_{b_i}^w, \theta_{b_i}^w \right], \left[p_{b_j}^w, \theta_{b_j}^w \right], \left[p_c^b, \theta_c^b \right], \lambda_l \tag{232}$$

其中 $\theta_{b_i}^w$、$\theta_{b_j}^w$、θ_c^b 分别为 $R_{b_i}^w$、$R_{b_j}^w$、R_c^b 对应的李代数。

按照(232)划分变量是为了和 VINS-Mono 代码保持一致。

5.4.3 雅可比矩阵

记：

$$r_C = \begin{bmatrix} b_1 & b_2 \end{bmatrix}^T \cdot \frac{1}{\left\| p_l^{c_j} \right\|} \cdot r_C^{'} \tag{233}$$

$$
\begin{aligned}
r_C^{'} &= \left\| p_l^{c_j} \right\| (\frac{p_l^{c_j}}{\left\| p_l^{c_j} \right\|} - \hat{\bar{p}}_l^{c_j}) \\
&= p_l^{c_j} - \hat{p}_l^{c_j} \\
&= R_b^c (R_w^{b_j} (R_{b_i}^w (R_c^b \frac{1}{\lambda_l} \hat{\bar{p}}_l^{c_i} + p_c^b) + p_{b_i}^w - p_{b_j}^w) - p_c^b) - \hat{p}_l^{c_j}
\end{aligned} \tag{234}
$$

这一部分实际上求的是 $r_C^{'}$ 关于待优化变量的雅可比矩阵。

根据(232)对待优化变量的划分，将雅可比矩阵分为四部分，相应的雅可比矩阵块为：$J[0]^{3\times 6}$、$J[1]^{3\times 6}$、$J[2]^{3\times 6}$、$J[3]^{3\times 1}$，它们分别对应 VINS-Mono 代码 projection_factor.cpp 中的 jacobians[0]、jacobians[1]、jacobians[2]、jacobians[3]。

求 $J[0]^{3\times 6}$：

$$\frac{\partial r_C^{'}}{\partial p_{b_i}^w} = \frac{\partial (R_b^c (R_w^{b_j} (R_{b_i}^w (R_c^b \frac{1}{\lambda_l} \hat{\bar{p}}_l^{c_i} + p_c^b) + p_{b_i}^w - p_{b_j}^w) - p_c^b) - \hat{p}_l^{c_j})}{\partial p_{b_i}^w} \tag{235}$$

$$= R_b^c R_w^{b_j}$$

$$\frac{\partial r_C^{'}}{\partial \theta_{b_i}^w} = \frac{\partial (R_b^c (R_w^{b_j} (R_{b_i}^w (R_c^b \frac{1}{\lambda_l} \hat{\bar{p}}_l^{c_i} + p_c^b) + p_{b_i}^w - p_{b_j}^w) - p_c^b) - \hat{p}_l^{c_j})}{\partial \theta_{b_i}^w}$$

$$= \lim_{\Delta\theta_{b_i}^w \to 0} \frac{(R_b^c (R_w^{b_j} (R_{b_i}^w \exp([\Delta\theta_{b_i}^w]_\times))(R_c^b \frac{1}{\lambda_l} \hat{\bar{p}}_l^{c_i} + p_c^b) + p_{b_i}^w - p_{b_j}^w) - p_c^b) - \hat{p}_l^{c_j}) - (R_b^c (R_w^{b_j} (R_{b_i}^w (R_c^b \frac{1}{\lambda_l} \hat{\bar{p}}_l^{c_i} + p_c^b) + p_{b_i}^w - p_{b_j}^w) - p_c^b) - \hat{p}_l^{c_j})}{\Delta\theta_{b_i}^w}$$

$$= \lim_{\Delta\theta_{b_i}^w \to 0} \frac{(R_b^c (R_w^{b_j} (R_{b_i}^w (I + [\Delta\theta_{b_i}^w]_\times)(R_c^b \frac{1}{\lambda_l} \hat{\bar{p}}_l^{c_i} + p_c^b) + p_{b_i}^w - p_{b_j}^w) - p_c^b) - \hat{p}_l^{c_j}) - (R_b^c (R_w^{b_j} (R_{b_i}^w (R_c^b \frac{1}{\lambda_l} \hat{\bar{p}}_l^{c_i} + p_c^b) + p_{b_i}^w - p_{b_j}^w) - p_c^b) - \hat{p}_l^{c_j})}{\Delta\theta_{b_i}^w}$$

$$= \lim_{\Delta\theta_{b_i}^w \to 0} \frac{R_b^c (R_w^{b_j} R_{b_i}^w [\Delta\theta_{b_i}^w]_\times (R_c^b \frac{1}{\lambda_l} \hat{\bar{p}}_l^{c_i} + p_c^b) - p_c^b) - \hat{p}_l^{c_j}}{\Delta\theta_{b_i}^w}$$

$$= \lim_{\Delta\theta_{b_i}^w \to 0} \frac{R_b^c R_w^{b_j} R_{b_i}^w [\Delta\theta_{b_i}^w]_\times (R_c^b \frac{1}{\lambda_l} \hat{\bar{p}}_l^{c_i} + p_c^b) - R_b^c p_c^b - \hat{p}_l^{c_j}}{\Delta\theta_{b_i}^w}$$

$$= \lim_{\Delta\theta_{b_i}^w \to 0} \frac{R_b^c R_w^{b_j} R_{b_i}^w [\Delta\theta_{b_i}^w]_\times (R_c^b \frac{1}{\lambda_l} \hat{\bar{p}}_l^{c_i} + p_c^b)}{\Delta\theta_{b_i}^w} \tag{236}$$

$$= \lim_{\Delta\theta_{b_i}^w \to 0} \frac{-R_b^c R_w^{b_j} R_{b_i}^w [R_c^b \frac{1}{\lambda_l} \hat{\bar{p}}_l^{c_i} + p_c^b]_\times \Delta\theta_{b_i}^w}{\Delta\theta_{b_i}^w}$$

$$= -R_b^c R_w^{b_j} R_{b_i}^w [R_c^b \frac{1}{\lambda_l} \hat{\bar{p}}_l^{c_i} + p_c^b]_\times$$

由(235)(236)可得：

$$J[0]^{3\times6} = \left[\frac{\partial r_C^{'}}{\partial p_{b_i}^{w}} \quad \frac{\partial r_C^{'}}{\partial \theta_{b_i}^{w}} \right] = \left[R_b^c R_w^{b_j} \quad -R_b^c R_w^{b_j} R_{b_i}^{w} \left[R_c^b \frac{1}{\lambda_l} \hat{\bar{p}}_l^{c_i} + p_c^b \right]_{\times} \right] \tag{237}$$

$J[0]^{3\times6}$ 对应的 projection_factor.cpp 中的代码片段为：

```cpp
if (jacobians[0])
{
    Eigen::Map<Eigen::Matrix<double, 2, 7, Eigen::RowMajor>> jacobian_pose_i(jacobians[0]);

    Eigen::Matrix<double, 3, 6> jaco_i;
    jaco_i.leftCols<3>() = ric.transpose() * Rj.transpose();
    jaco_i.rightCols<3>() = ric.transpose() * Rj.transpose() * Ri * -Utility::skewSymmetric(pts_imu_i);

    jacobian_pose_i.leftCols<6>() = reduce * jaco_i;
    jacobian_pose_i.rightCols<1>().setZero();
}
```

求 $J[1]^{3\times6}$：

$$\frac{\partial r_C^{'}}{\partial p_{b_j}^{w}} = \frac{\partial (R_b^c (R_w^{b_j} (R_{b_i}^{w} (R_c^b \frac{1}{\lambda_l} \hat{\bar{p}}_l^{c_i} + p_c^b) + p_{b_i}^{w} - p_{b_j}^{w}) - p_c^b) - \hat{p}_l^{c_j})}{\partial p_{b_j}^{w}} \tag{238}$$

$$= -R_b^c R_w^{b_j}$$

$$\frac{\partial r_C^{'}}{\partial \theta_{b_j}^w} = \frac{\partial(R_b^c(R_w^{b_j}(R_{b_i}^w(R_c^b \frac{1}{\lambda_l}\hat{\bar{p}}_l^{c_i} + p_c^b) + p_{b_i}^w - p_{b_j}^w) - p_c^b) - \hat{p}_l^{c_j})}{\partial \theta_{b_j}^w}$$

$$= \lim_{\Delta\theta_{b_j}^w \to 0} \frac{\begin{aligned}&(R_b^c((R_{b_j}^w \exp(\left[\Delta\theta_{b_j}^w\right]_\times))^T(R_{b_i}^w(R_c^b \frac{1}{\lambda_l}\hat{\bar{p}}_l^{c_i} + p_c^b) + p_{b_i}^w - p_{b_j}^w) - p_c^b) - \hat{p}_l^{c_j}) - \\ &(R_b^c(R_w^{b_j}(R_{b_i}^w(R_c^b \frac{1}{\lambda_l}\hat{\bar{p}}_l^{c_i} + p_c^b) + p_{b_i}^w - p_{b_j}^w) - p_c^b) - \hat{p}_l^{c_j})\end{aligned}}{\Delta\theta_{b_j}^w}$$

$$= \lim_{\Delta\theta_{b_j}^w \to 0} \frac{\begin{aligned}&(R_b^c((R_{b_j}^w(I + \left[\Delta\theta_{b_j}^w\right]_\times))^T(R_{b_i}^w(R_c^b \frac{1}{\lambda_l}\hat{\bar{p}}_l^{c_i} + p_c^b) + p_{b_i}^w - p_{b_j}^w) - p_c^b) - \hat{p}_l^{c_j}) - \\ &(R_b^c(R_w^{b_j}(R_{b_i}^w(R_c^b \frac{1}{\lambda_l}\hat{\bar{p}}_l^{c_i} + p_c^b) + p_{b_i}^w - p_{b_j}^w) - p_c^b) - \hat{p}_l^{c_j})\end{aligned}}{\Delta\theta_{b_j}^w}$$

$$= \lim_{\Delta\theta_{b_j}^w \to 0} \frac{-R_b^c \left[\Delta\theta_{b_j}^w\right]_\times R_w^{b_j}(R_{b_i}^w(R_c^b \frac{1}{\lambda_l}\hat{\bar{p}}_l^{c_i} + p_c^b) + p_{b_i}^w - p_{b_j}^w)}{\Delta\theta_{b_j}^w}$$

$$= \lim_{\Delta\theta_{b_j}^w \to 0} \frac{-R_b^c \left[\Delta\theta_{b_j}^w\right]_\times R_w^{b_j}(R_{b_i}^w(R_c^b \frac{1}{\lambda_l}\hat{\bar{p}}_l^{c_i} + p_c^b) + p_{b_i}^w - p_{b_j}^w)}{\Delta\theta_{b_j}^w}$$

$$= \lim_{\Delta\theta_{b_j}^w \to 0} \frac{R_b^c \left[R_w^{b_j}(R_{b_i}^w(R_c^b \frac{1}{\lambda_l}\hat{\bar{p}}_l^{c_i} + p_c^b) + p_{b_i}^w - p_{b_j}^w)\right]_\times \Delta\theta_{b_j}^w}{\Delta\theta_{b_j}^w}$$

$$= R_b^c \left[R_w^{b_j}(R_{b_i}^w(R_c^b \frac{1}{\lambda_l}\hat{\bar{p}}_l^{c_i} + p_c^b) + p_{b_i}^w - p_{b_j}^w)\right]_\times \tag{239}$$

由(238)(239)可得：

$$J[1]^{3\times6} = \left[\begin{array}{cc}\frac{\partial r_C^{'}}{\partial p_{b_j}^w} & \frac{\partial r_C^{'}}{\partial \theta_{b_j}^w}\end{array}\right] = \left[\begin{array}{cc}-R_b^c R_w^{b_j} & R_b^c\left[R_w^{b_j}(R_{b_i}^w(R_c^b \frac{1}{\lambda_l}\hat{\bar{p}}_l^{c_i} + p_c^b) + p_{b_i}^w - p_{b_j}^w)\right]_\times\end{array}\right] \tag{240}$$

$J[1]^{3\times6}$ 对应的 projection_factor.cpp 中的代码片段为：

```
if (jacobians[1])
{
    Eigen::Map<Eigen::Matrix<double, 2, 7, Eigen::RowMajor>> jacobian_pose_j(jacobians[1]);

    Eigen::Matrix<double, 3, 6> jaco_j;
    jaco_j.leftCols<3>() = ric.transpose() * -Rj.transpose();
    jaco_j.rightCols<3>() = ric.transpose() * Utility::skewSymmetric(pts_imu_j);

    jacobian_pose_j.leftCols<6>() = reduce * jaco_j;
    jacobian_pose_j.rightCols<1>().setZero();
}
```

求 $J\left[2\right]^{3\times 6}$:

$$
\begin{aligned}
\frac{\partial r_C^{'}}{\partial p_c^b} &= \frac{\partial (R_b^c (R_w^{b_j} (R_{b_i}^w (R_c^b \frac{1}{\lambda_l} \hat{\bar{p}}_l^{c_i} + p_c^b) + p_{b_i}^w - p_{b_j}^w) - p_c^b) - \hat{p}_l^{c_j})}{\partial p_c^b} \\
&= R_b^c R_w^{b_j} R_{b_i}^w - R_b^c \\
&= R_b^c (R_w^{b_j} R_{b_i}^w - I)
\end{aligned}
\tag{241}
$$

$$\frac{\partial r_C^{'}}{\partial \theta_c^b} = \frac{\partial (R_b^c (R_w^{b_j}(R_{b_i}^w(R_c^b \frac{1}{\lambda_l}\hat{\bar{p}}_l^{c_i}+p_c^b)+p_{b_i}^w-p_{b_j}^w)-p_c^b)-\hat{p}_l^{c_j})}{\partial \theta_c^b}$$

$$= \lim_{\Delta\theta_c^b \to 0} \frac{((R_c^b \exp([\Delta\theta_c^b]_\times))^T (R_w^{b_j}(R_{b_i}^w(R_c^b \exp([\Delta\theta_c^b]_\times)\frac{1}{\lambda_l}\hat{\bar{p}}_l^{c_i}+p_c^b)+p_{b_i}^w-p_{b_j}^w)-p_c^b)-\hat{p}_l^{c_j})- (R_b^c(R_w^{b_j}(R_{b_i}^w(R_c^b \frac{1}{\lambda_l}\hat{\bar{p}}_l^{c_i}+p_c^b)+p_{b_i}^w-p_{b_j}^w)-p_c^b)-\hat{p}_l^{c_j})}{\Delta\theta_c^b}$$

$$= \lim_{\Delta\theta_c^b \to 0} \frac{((I-[\Delta\theta_c^b]_\times)R_b^c(R_w^{b_j}(R_{b_i}^w(R_c^b(I+[\Delta\theta_c^b]_\times)\frac{1}{\lambda_l}\hat{\bar{p}}_l^{c_i}+p_c^b)+p_{b_i}^w-p_{b_j}^w)-p_c^b)-\hat{p}_l^{c_j})- (R_b^c(R_w^{b_j}(R_{b_i}^w(R_c^b \frac{1}{\lambda_l}\hat{\bar{p}}_l^{c_i}+p_c^b)+p_{b_i}^w-p_{b_j}^w)-p_c^b)-\hat{p}_l^{c_j})}{\Delta\theta_c^b}$$

$$= \lim_{\Delta\theta_c^b \to 0} \frac{(R_b^c-[\Delta\theta_c^b]_\times R_b^c)(R_w^{b_j}R_{b_i}^w R_c^b[\Delta\theta_c^b]_\times \frac{1}{\lambda_l}\hat{\bar{p}}_l^{c_i}+R_w^{b_j}(R_{b_i}^w(R_c^b \frac{1}{\lambda_l}\hat{\bar{p}}_l^{c_i}+p_c^b)+p_{b_i}^w-p_{b_j}^w)-p_c^b)}{\Delta\theta_c^b}$$

$$= \lim_{\Delta\theta_c^b \to 0} \frac{R_b^c R_w^{b_j}R_{b_i}^w R_c^b[\Delta\theta_c^b]_\times \frac{1}{\lambda_l}\hat{\bar{p}}_l^{c_i}-[\Delta\theta_c^b]_\times R_b^c(R_w^{b_j}(R_{b_i}^w(R_c^b \frac{1}{\lambda_l}\hat{\bar{p}}_l^{c_i}+p_c^b)+p_{b_i}^w-p_{b_j}^w)-p_c^b)}{\Delta\theta_c^b}$$

$$= \lim_{\Delta\theta_c^b \to 0} \frac{-R_b^c R_w^{b_j}R_{b_i}^w R_c^b[\frac{1}{\lambda_l}\hat{\bar{p}}_l^{c_i}]_\times \Delta\theta_c^b + [R_b^c(R_w^{b_j}(R_{b_i}^w(R_c^b \frac{1}{\lambda_l}\hat{\bar{p}}_l^{c_i}+p_c^b)+p_{b_i}^w-p_{b_j}^w)-p_c^b)]_\times \Delta\theta_c^b}{\Delta\theta_c^b}$$

$$= -R_b^c R_w^{b_j}R_{b_i}^w R_c^b[\frac{1}{\lambda_l}\hat{\bar{p}}_l^{c_i}]_\times + [R_b^c(R_w^{b_j}(R_{b_i}^w(R_c^b \frac{1}{\lambda_l}\hat{\bar{p}}_l^{c_i}+p_c^b)+p_{b_i}^w-p_{b_j}^w)-p_c^b)]_\times$$

$$= -R_b^c R_w^{b_j}R_{b_i}^w R_c^b[\frac{1}{\lambda_l}\hat{\bar{p}}_l^{c_i}]_\times + [R_b^c R_w^{b_j}R_{b_i}^w R_c^b \frac{1}{\lambda_l}\hat{\bar{p}}_l^{c_i}]_\times + [R_b^c(R_w^{b_j}(R_{b_i}^w p_c^b+p_{b_i}^w-p_{b_j}^w)-p_c^b)]_\times$$

$$(242)$$

由(241)(242)得：

$$J[2]^{3\times6} = \begin{bmatrix} \dfrac{\partial r_C^{'}}{\partial p_c^b} & \dfrac{\partial r_C^{'}}{\partial \theta_c^b} \end{bmatrix} = \begin{bmatrix} R_b^c(R_w^{b_j}R_{b_i}^w-I) & \begin{matrix} -R_b^c R_w^{b_j}R_{b_i}^w R_c^b[\frac{1}{\lambda_l}\hat{\bar{p}}_l^{c_i}]_\times + [R_b^c R_w^{b_j}R_{b_i}^w R_c^b \frac{1}{\lambda_l}\hat{\bar{p}}_l^{c_i}]_\times \\ + [R_b^c(R_w^{b_j}(R_{b_i}^w p_c^b+p_{b_i}^w-p_{b_j}^w)-p_c^b)]_\times \end{matrix} \end{bmatrix}$$

$$(243)$$

$J[2]^{3\times6}$ 对应的 projection_factor.cpp 中的代码片段为：

```
if (jacobians[2])
{
    Eigen::Map<Eigen::Matrix<double, 2, 7, Eigen::RowMajor>> jacobian_ex_pose(jacobians[2]);
    Eigen::Matrix<double, 3, 6> jaco_ex;
    jaco_ex.leftCols<3>() = ric.transpose() * (Rj.transpose() * Ri - Eigen::Matrix3d::Identity());
    Eigen::Matrix3d tmp_r = ric.transpose() * Rj.transpose() * Ri * ric;
    jaco_ex.rightCols<3>() = -tmp_r * Utility::skewSymmetric(pts_camera_i) + Utility::skewSymmetric(tmp_r * pts_camera_i) +
                             Utility::skewSymmetric(ric.transpose() * (Rj.transpose() * (Ri * tic + Pi - Pj) - tic));
    jacobian_ex_pose.leftCols<6>() = reduce * jaco_ex;
    jacobian_ex_pose.rightCols<1>().setZero();
}
if (jacobians[3])
```

求 $J[3]^{3\times1}$ ：

$$\frac{\partial r_C^{'}}{\partial \lambda_l} = \frac{\partial (R_b^c(R_w^{b_j}(R_{b_i}^w(R_c^b\frac{1}{\lambda_l}\hat{\bar{p}}_l^{c_i}+p_c^b)+p_{b_i}^w-p_{b_j}^w)-p_c^b)-\hat{\bar{p}}_l^{c_j})}{\partial \lambda_l} \tag{244}$$

$$= -R_b^c R_w^{b_j} R_{b_i}^w R_c^b \frac{\hat{\bar{p}}_l^{c_i}}{\lambda_l^2}$$

由(244)可得：

$$J[3]^{3\times1} = \frac{\partial r_C^{'}}{\partial \lambda_l} = -R_b^c R_w^{b_j} R_{b_i}^w R_c^b \frac{\hat{\bar{p}}_l^{c_i}}{\lambda_l^2} \tag{245}$$

$J[3]^{3\times1}$ 对应的 projection_factor.cpp 中的代码片段为：

```
    if (jacobians[3])
    {
        Eigen::Map<Eigen::Vector2d> jacobian_feature(jacobians[3]);
#if 1
        jacobian_feature = reduce * ric.transpose() * Rj.transpose() * Ri * ric * pts_i * -1.0 / (inv_dep_i * inv_dep_i);
#else
        jacobian_feature = reduce * ric.transpose() * Rj.transpose() * Ri * ric * pts_i;
#endif
```

5.4.4 协方差矩阵

（这一部分还有疑问，不太确定这么算对不对）

视觉测量残差的协方差矩阵代表着视觉测量的不确定性，这种不确定性来自于相机将 3D 点投影到像素平面的过程。相机内参是我们对相机成像过程的一种建模，实际成像过程中，可以认为实际像素坐标由理论像素坐标和噪声两部分组成。假设 u、v 坐标方向的噪声相互独立，且噪声均为 0 均值，方差为 1.5（像素平方）的高斯噪声，对应到归一化相机平面上的协方差需要除以焦距 f：

$$\Sigma_{vis} = \frac{1.5}{f} I_{2\times2} \tag{246}$$

5.5 边缘化（Marginalization）

待补充！

6 参考文献

[1] SOLÀ J. Quaternion kinematics for the error-state Kalman filter[J]. 2017.

[2] YANG Z, SHEN S. Monocular Visual‑Inertial State Estimation With Online Initialization and Camera‑IMU Extrinsic Calibration[J]. IEEE Transactions on Automation Science & Engineering, 2017,14(1):39-51.

6 参考文献