MSCKF_VIO算法

1、MSCKF视觉惯性融合算法简介

惯性导航利用惯性测量单元(IMU)测量得到的角速度、加速度信息进行惯性导航解算得到运载体的位置、速度、姿态(含航向)等信息，具有实时性好、动态性能好等优点；但是由于其积分式特点，使得传感器和算法解算的误差会持续累积，导致长时间精度很低，特别是对于低端IMU。即使是军用战略性武器上使用的高精度惯导系统也在努力与别的信息进行融合，提高实际导航精度。视觉信息包含了丰富的三维场景信息，通过视觉跟踪可以对运载体的运动进行测量，视觉测量可以给出运动增量。惯性与视觉融合算法利用视觉观测与IMU预测联合得到重投影误差作为观测量来进行系统状态估计。分为滑动窗SWF与多状态约束卡尔曼滤波MSCKF两种技术方案。作为惯性导航专业背景的从业人员，比较青睐基于卡尔曼滤波的MSCKF算法。这里给出msckf_vio_mono算法的数学推导过程。这里的公式表达和推导过程按照惯性导航的习惯，对于非惯性导航专业的可能会感觉有点陌生。

2、MSCKF视觉前端

这里给出msckf_vio_mono代码里给出的视觉前端算法模型。

S1:创建图像金字塔

S2:提取图像FAST特征

S3:跟踪特征

S3.1>使用IMU积分计算先前帧和当前帧的相对旋转运动粗略估值C_p^c；

S3.2>整理并存储先前图像的特征点编号、坐标、存在时间，给出跟踪前特征信息；

S3.3>使用先前帧特征像素坐标、C_p^c、相机内参预测当前帧特征的像素坐标；

假定先前帧特征像素坐标x、当前帧对应特征像素坐标为x′，按照针孔模型其与世界系坐标X关系为

$$x = K\begin{bmatrix} I & 0 \end{bmatrix} X \tag{1a}$$

$$x' = KC_p^c \begin{bmatrix} I & -\tilde{C} \end{bmatrix} X \tag{1b}$$

当仅考虑旋转时，式(1b)可以重新列写为：

$$x' = KC_p^c X = KC_p^c K^{-1} x \tag{1c}$$

式(1c)给出了由先前帧特征像素坐标计算当前帧对应特征像素坐标的公式。其中K为相机内参，可预先标定获取，C_p^c由前一步IMU积分获取。其中K和C_p^c都是3x3矩阵，x和x′取齐次形式，即：

$x = \begin{bmatrix} x_1 & x_2 & 1 \end{bmatrix}^T$和$x' = \begin{bmatrix} x_1' & x_2' & 1 \end{bmatrix}^T$，$X = \begin{bmatrix} X_1 & Y_1 & Z_1 & 1 \end{bmatrix}$，在求取x′后做一次齐次化即可；

对于无人机这类应用，线速度一般不小，平移量应该不能忽略，比如对于fla飞行数据集，飞行速度从5m/s到17.5m/s，在视觉10Hz更新时，一个周期内的位移量为0.5m-1.75m，量级不小；对于消费级IMU实测的纯INS性能约0.2m/0.2s，这样在0.1s周期内INS漂移约0.1-0.2m，小于实际平移，INS积分平移补偿是有意义的，式(1a)、(1b)可以得到新考虑旋转和平移的运动补偿公式：

$$x' = KC_p^c\left[I \quad -\tilde{C}\right]X = K\left[C_p^c \quad t\right]K^{-1}x,$$

其中，

$$t = C_p^c\tilde{C}$$

世界系到图像系的转换关系为

$$X_{cam} = C_p^c X + t$$

MSCKF跑fla数据集合时运行一会就会出现算法跑飞，把这一公式改进加入代码之后。。。

对于LDS雷达来说也有类似的补偿公式：

$$\nu_{\mathrm{pre}}^b = C_L^b\nu_{pre}^L$$

$$\nu_{\mathrm{cur}}^b = C_p^c\nu_{pre}^b + t$$

根据前后时刻两个扫描点在投影，想要获取的是补偿之后的L系扫描点投影，

$$\nu_{\mathrm{pre}}^{\mathrm{L}} = C_L^{b^{-1}}\nu_{pre}^b$$

$$= C_L^{b^{-1}}C_c^p(\nu_{\mathrm{cur}}^b - t)$$

$$= C_L^{b^{-1}}C_c^p\nu_{\mathrm{cur}}^b - C_L^{b^{-1}}C_c^p t$$

$$= C_L^{b^{-1}}C_c^pC_L^b\nu_{cur}^L - C_L^{b^{-1}}C_c^p t$$

当把旋转和平移都放入一个矩阵表达时，上式就是

$$\nu_{\mathrm{pre}}^{\mathrm{L}} = C_L^{b^{-1}}C_c^p\nu_{\mathrm{cur}}^b = C_L^{b^{-1}}C_c^pC_L^b\nu_{\mathrm{cur}}^L$$

S3.4>使用 S1 构建的金字塔、S2 提取的特征和 S3.3 预测的特征位置x′，基于光流跟踪算法给出精确特征位置x′，并给出哪些点被跟踪，哪些点没有跟踪；

S3.5>对被跟踪的特征点做两点RANSAC，剔除外点；

S4:添加特征

由于每次跟踪会导致特征点数目下降，在跟踪之后需要添加新的特征。在当前特征之外再次提取特征。

S5:裁剪特征

去除同一区域内过多的特征。

3、MSCKF视觉惯性融合

3.1MSCKF简介

多状态约束卡尔曼滤波(MSCKF)是视觉/惯性融合的一种紧耦合组合导航算法。MSCKF其实是一种平滑算法，利用多个相机位姿对同一视觉特征的观测形成对多个位姿的几何约束。GNSS/INS紧耦合是多个卫星同时约束一个位姿，估计过程是滤波过程。深层次的原因是GNSS提供先验的卫星位置及伪距信息，但是视觉并无这些信息。视觉特征匹配才找到并跟踪“卫星”(特征点三维位置)，这一过程必然是时间顺序，不是实时的。三角化才能得到特征点位置，这是依赖视觉和INS的，进而观测量是含有状态的，这是非线性的估计；同时量测是帧间的，但状态是帧同步的。

MSCKF利用INS信息进行状态和协方差预测、跟踪两帧图像特征。特征点不加入状态向量中，仅在新的图像信息到来时将该帧的位姿信息加入状态向量中。状态向量包含INS和形成几何约束的若干个图像位姿信息的滑动窗口。在满足一定条件时进行边缘化，剔除旧的状态，保留新的状态，保持滑动窗的大小在一定限度内。当一直跟踪的特征点无法观测到或是滑动窗大小达到某一阈值时，利用三角测量和迭代优化算法计算该特征点的位置，得到一个精确的三维空间点位置。利用这些特征点信息对滑动窗内位姿的几何约束来估计INS的各种信息，估计可以采用EKF或是UKF。

3.2 MSCKF算法流程

MSCKF算法估计INS系b在全局导航系(世界系)n里的位姿。相机系C和INS固联，两者之间由包含旋转$\mathrm{C}_{\mathrm{c}}^{b}$和平移$\mathrm{p}_{c}^{b}$的外参表达。INS的采样率和更新速率都比较高，一般在100Hz以上；相机的测量和视觉算法的更新速率较低，一般在10-30Hz左右。在每一步INS更新时都更新协方差矩阵。当每一帧图像信息到来时由许多任务要做，包括增广当前状态向量，将当前相机的位姿添加到滑动窗中，同时更新协方差。特征点跟踪也要跟踪之前一致跟踪的特征点并检测这些被跟踪的特征点是否消失。如果特征点消失或是滑动窗已达制定上限阈值，就开始进行量测更新；并在下一帧图像到来时重新开始跟踪特征点。量测更新时通过三角测量和高斯牛顿迭代优化算法计算特征点的位置。由于是跟踪特征在多个相机位姿处的观测，处理会有延时，但是三角测量的结果很精确。预测的特征点的位置与量测值的重投影误差作为卡尔曼滤波量测误差，以此进行状态估计，估计出状态误差之后反馈。这是视觉/惯性卡尔曼滤波融合算法的流程。视觉与惯性的融合要求INS与视觉测量是同步的，每一帧图像产生时都有对应的INS测量值，每帧图像对应整数帧的INS信息。

3.3 INS状态向量和INS误差方程

与视觉融合的INS多是低成本MEMS惯导系统，其陀螺精度（零偏稳定性及重复性）为0.1°/s量级，加速度计精度为5mg量级或更差。由于陀螺精度太低，无法敏感到地球自转信息，因而没有必要采用完整而复杂的捷联惯导更新算法，可对其作大幅简化。

简化的捷联姿态更新算法为

$$\mathrm{C}_{b(m)}^{n} = C_{b(m-1)}^{n} C_{b(m)}^{b(m-1)} \tag{2}$$

其中

$$C_{b(m)}^{b(m-1)} = \left[\cos(\Delta\theta/2) \quad \Delta\vec{\theta}/\Delta\theta * \sin(\Delta\theta/2)\right]^{T} \tag{3}$$

$\mathrm{C}_{b(m)}^{n}$表示t_m时刻的姿态变换矩阵，$\mathrm{C}_{b(m)}^{b(m-1)}$是从$t_{m-1}$时刻到$t_m$时刻的姿态矩阵变化（记采样间隔$T_s = t_m - t_{m-1}$），$\boldsymbol{\Delta\theta}_m$是陀螺在时间段$[t_{m-1}, t_m]$内输出的角增量且$\Delta\theta_m = \left|\boldsymbol{\Delta\theta}_m\right|$。低精度陀螺一般采用角速率输出采样方式，只需简单地将其乘以采样间隔T_s即可近似变换为角增量。

对于中低速行驶的运载体，比如地速v<100m/s，可以忽略地球自转及地球曲率的影响，速度更新方程简化为

$$\mathrm{v}_{\mathrm{m}}^{n} = v_{m-1}^{n} + \Delta v_{sf(m)}^{n} + g^{n} T_s \tag{4}$$

其中

$$\Delta v_{sf(m)}^{n} = C_{b(m-1)}^{n}(\Delta v_m + 0.5 * \Delta\theta_m \times \Delta v_m) \qquad (5)$$

$\boldsymbol{v}_m^n$ 为 t_m 时刻的惯导速度，$\boldsymbol{C}_{b(m-1)}^n$ 为与四元数 $\boldsymbol{Q}_{b(m-1)}^n$ 对应的姿态阵，$\boldsymbol{\Delta v}_m$ 是加速度计在时间段 $[t_{m-1}, t_m]$ 内输出的比力增量，实际中也可采用比力输出乘以采样间隔进行近似。

一般应用中，低成本MEMS系统通常在小范围内运动，比如数百米，这时可以选择当地直角坐标系作为导航参考坐标系（n 系），导航起始点作为坐标原点（o），三坐标轴（ox_n、oy_n 和 oz_n）分别指向东向、北向和天向。在直角坐标系下，导航定位微分方程将变得非常简单，为 $\dot{\boldsymbol{p}}^n = \boldsymbol{v}^n$，对其离散化即得位置更新方程

$$\boldsymbol{p}_m^n = \boldsymbol{p}_{m-1}^n + \frac{\boldsymbol{v}_{m-1}^n + \boldsymbol{v}_m^n}{2} T_s \qquad (6)$$

其中，记 $\boldsymbol{p}_m^n = \begin{bmatrix} x_m & y_m & z_m \end{bmatrix}^{\mathrm{T}}$。

与上述简化导航算法相对应的低精度惯导系统误差方程，如下

$$\dot{\boldsymbol{\phi}} = -\boldsymbol{C}_b^n(\boldsymbol{\varepsilon}_r^b + \boldsymbol{w}_\varepsilon) \qquad (7a)$$

$$\delta\dot{\boldsymbol{v}}^n = \boldsymbol{f}_{sf}^n \times \boldsymbol{\phi} + \boldsymbol{C}_b^n(\nabla_r^b + \boldsymbol{w}_\nabla) \qquad (7b)$$

$$\delta\dot{\boldsymbol{p}}^n = \delta\boldsymbol{v}^n \qquad (7c)$$

其中，$\boldsymbol{w}_\varepsilon$ 和 $\boldsymbol{w}_\nabla$ 分别为陀螺角速率白噪声和加计比力白噪声，$\boldsymbol{\varepsilon}_r^b = \begin{bmatrix} \varepsilon_{rx}^b & \varepsilon_{ry}^b & \varepsilon_{rz}^b \end{bmatrix}^{\mathrm{T}}$ 和 $\nabla_r^b = \begin{bmatrix} \nabla_{rx}^b & \nabla_{ry}^b & \nabla_{rz}^b \end{bmatrix}^{\mathrm{T}}$ 分别为陀螺和加速度计一阶马尔科夫过程随机误差，如下

$$\dot{\varepsilon}_{ri}^b = -\frac{1}{\tau_{Gi}}\varepsilon_{ri}^b + w_{rGi} \quad \text{和} \quad \dot{\nabla}_{ri}^b = -\frac{1}{\tau_{Ai}}\nabla_{ri}^b + w_{rAi} \qquad i = x, y, z \qquad (8)$$

τ_{Gi} 和 τ_{Ai} 是相应的相关时间常数，w_{rGi} 和 w_{rAi} 是一阶马尔科夫过程激励白噪声。对于低精度的惯性器件，假设其时间相关误差模型为一阶马尔科夫过程是非常实用的：其一，与随机常值模型相比，一阶马尔科夫模型可在长时间组合滤波后避免滤波器过度收敛现象，过度收敛会导致滤波器抗干扰性能变差；其二，如果惯性器件误差中确实存在较大随机常值成分，可通过滤波器的惯性器件误差反馈校正，消除随机常值误差的影响；其三，与同时建立随机常值和一阶马尔科夫过程两种模型相比，仅使用后者有利于降低建模维数和滤波计算量。

选择惯导系统的姿态失准角 $\boldsymbol{\phi}$、速度误差 $\delta\boldsymbol{v}^n$、定位误差 $\delta\boldsymbol{p}^n$、陀螺相关漂移 $\boldsymbol{\varepsilon}_r^b$、加速度计相关偏值 ∇_r^b 为状态 χ（共15维）。

$$\chi = \begin{bmatrix} \phi^{\mathrm{T}} & \delta V^{n^T} & \delta p^{n^T} & \varepsilon_r^b & \nabla^b \end{bmatrix}^T \qquad (9)$$

系统状态空间模型为

$$\dot{\chi} = F\chi + GW \qquad (10)$$

其中,

$$\mathrm{F} = \begin{bmatrix} 0_{3x3} & 0_{3x3} & 0_{3x3} & -C_b^n & 0_{3x3} \\ f_{sf}^n \times & 0_{3x3} & 0_{3x3} & 0_{3x3} & C_b^n \\ 0_{3x3} & I & 0_{3x3} & 0_{3x3} & 0_{3x3} \\ 0_{3x3} & 0_{3x3} & 0_{3x3} & \beta_G & 0_{3x3} \\ 0_{3x3} & 0_{3x3} & 0_{3x3} & 0_{3x3} & \beta_A \end{bmatrix} \qquad (11a)$$

$$\beta_G = \begin{bmatrix} 1/\tau_{gx} & 0 & 0 \\ 0 & 1/\tau_{gy} & 0 \\ 0 & 0 & 1/\tau_{gz} \end{bmatrix}, \beta_A = \begin{bmatrix} 1/\tau_{ax} & 0 & 0 \\ 0 & 1/\tau_{ay} & 0 \\ 0 & 0 & 1/\tau_{az} \end{bmatrix} \quad (11b)$$

$$G = \begin{bmatrix} -C_b^n & 0_{3x3} & 0_{3x3} & 0_{3x3} \\ 0_{3x3} & C_b^n & 0_{3x3} & 0_{3x3} \\ 0_{3x3} & 0_{3x3} & 0_{3x3} & 0_{3x3} \\ 0_{3x3} & 0_{3x3} & I_{3x3} & 0_{3x3} \\ 0_{3x3} & 0_{3x3} & 0_{3x3} & I_{3x3} \end{bmatrix} \quad (12)$$

3.4 MSCKF状态向量

MSCKF误差状态向量包括两部分：INS误差状态χ_{ins}，相机位姿误差状态$\pi_{\text{i}} = (\phi_{C_i}^n, \delta p_{C_i}^n)$，列写如下：

$$\chi = \begin{bmatrix} \chi_{ins}^T & \phi_{C_1}^{n\,T} & \delta p_{C_1}^{n\,T} & \cdots & \phi_{C_N}^{n\,T} & \delta p_{C_N}^{n\,T} \end{bmatrix}^T \quad (13)$$

对于MSCKF全状态测量值来说$\tilde{X} = X + \chi$，这一点与一般视觉文档不一样，惯性导航采用这一微扰方式。

状态协方差矩阵

$$P_k = \begin{bmatrix} P_{II_k} & P_{IC_k} \\ P_{IC_k}^T & P_{CC_k} \end{bmatrix} \quad (14)$$

其中，

P_{II_k}是INS误差状态协方差，为15x15矩阵；

P_{IC_k} INS与相机状态互协方差，为15x(15+6N)矩阵；

P_{CC_k}是相机误差状态协方差，为6Nx6N矩阵；

加入相机状态就要进行状态协方差矩阵的增广。

3.5 INS状态更新

当采集到IMU数据时通过式(2)-(6)进行INS状态更新，得到当前时刻的运动学信息。并由式(11a)计算INS误差状态相关的矩阵。

3.6 相机状态增广

当一帧新图像数据到来时，要把当前相机的位姿误差$\pi_{\text{i}} = (\phi_{C_i}^n, \delta p_{C_i}^n)$添加到滑动窗中，同时也要增广更新协方差矩阵$P_k$。

使用 INS 状态更新的结果和相机相对于 INS 的外参C_c^b和p_c^b可以计算相机位姿。

$$p_c^n = p_b^n + C_b^n p_c^b \quad (15a)$$

$$C_c^n = C_b^n C_c^b \quad (15b)$$

由式（15）可以看出，相机姿态误差π_N相对于 INS 误差状态χ_{ins}的雅可比矩阵为只与位

姿的变化有关，列写如下：

$$\mathrm{J}_{\mathrm{N}} = \frac{\partial \pi_N}{\partial \chi_{ins}} = \begin{bmatrix} \frac{\partial p_C^n}{\partial \phi} & 0 & \frac{\partial p_C^n}{\partial \delta p} & \cdots & 0 \\ \frac{\partial \phi_C^n}{\partial \phi} & 0 & \frac{\partial \phi_C^n}{\partial \delta p} & \cdots & 0 \end{bmatrix} \tag{16}$$

下面给出式(16)中各个分量的详细推导过程。

对式(15a)做微扰可得：

$$\begin{aligned} \delta \mathrm{p}_{\mathrm{c}}^n &= \delta p_b^n + \delta C_b^n p_c^b \\ &= \delta p_b^n + (C_b^{n'} - C_b^n)\, p_c^b \\ &= \delta p_b^n + (C_n^{n'} - I) C_b^n p_c^b \\ &= \delta p_b^n + (I - \phi \times -I) C_b^n p_c^b \\ &= \delta p_b^n - (\phi \times) C_b^n p_c^b \\ &= \delta p_b^n + (C_b^n p_c^b) \times \phi \end{aligned} \tag{17a}$$

由式(17a)可得式(16)的第一行偏微分：

$$\frac{\partial p_c^n}{\partial \phi} = (C_b^n p_c^b) \times \tag{17b}$$

$$\frac{\partial p_c^n}{\partial \delta p} = \mathrm{I} \tag{17c}$$

对式(15b)做微扰可得：

$$\begin{aligned} \delta \mathrm{C}_{\mathrm{c}}^n &= \delta C_b^n \mathrm{C}_c^b \\ &= (\tilde{C}_b^n - C_b^n)\, \mathrm{C}_c^b \\ &= (C_n^{n'} - I) C_b^n \mathrm{C}_c^b \\ &= (I - \phi \times -I) C_b^n \mathrm{C}_c^b \\ &= -\phi \times C_b^n \mathrm{C}_c^b \\ &= -\phi \times C_c^n \\ &= C_c^n (\phi \times) \end{aligned} \tag{17d}$$

$$\begin{aligned} \delta \mathrm{C}_{\mathrm{c}}^n &= C_c^{n'} - C_c^n \\ &= (C_n^{n'}(C) - I) C_c^n \\ \text{又} &= (I - \phi_c^n \times -I) C_c^n \\ &= -\phi_c^n \times C_c^n \\ &= C_c^n (\phi_c^n \times) \end{aligned} \tag{17e}$$

由式(17d)、(17e)可得式(16)的第二行偏微分值：

$$\frac{\partial \phi_C^n}{\partial \delta p} = 0 \tag{17f}$$

$$\frac{\partial \phi_C^n}{\partial \phi} = \mathrm{I} \tag{17g}$$

由式(17b)、(17c)、(17f)、(17g)可以得到式(16)的各个分量，式(16)重新列写如下：

$$\mathrm{J_N} = \frac{\partial \pi_N}{\partial \chi_{ins}} = \begin{bmatrix} (C_b^n p_c^b)\times & 0 & \mathrm{I} & \dots & 0 \\ I & 0 & 0 & \dots & 0 \end{bmatrix} \tag{18}$$

3.7 量测模型

使用 IMU 数据做惯性导航解算得到的运动结果误差随时间逐渐累积，需要通过视觉测量进行修正。视觉提供了三维信息的二维观测，可以通过特征提取和跟踪的方式提供帧间的几何约束。如果在不同相机位姿都可观测到同一特征点，可以加强对各个位姿的几何约束。

通过求取的特征点在世界系下的三维坐标以及相机位姿的估计值可以得到经过视觉算法跟踪的特征点的量测位置 $\hat{\mathrm{z}}_i^j$，这一位置认为是精确的，这就是视觉特征点对窗口内位姿的几何约束。使用特征跟踪(预跟踪也用了 INS)得到的特征点位置为 z_i^j。通过预测值与量测值的差构造卡尔曼滤波的测量残差 $\mathrm{r}_i^{(j)}$--重投影误差。

$$\mathrm{r}_i^{(j)} = \mathrm{z}_i^{(j)} - \hat{\mathrm{z}}_i^{(j)} \tag{19}$$

式(19)的重投影误差作为卡尔曼滤波的量测量，用于估计系统状态向量，然后反馈修正。

3.7.1 特征点三维位置求取

假定一个特征点 $\mathrm{f_j}$ 在连续 N 帧图像中被观测到，每一个测量值 $\mathrm{z}_i^j = \begin{bmatrix} u_i^j & v_i^j \end{bmatrix}^T$，$i = 1...N$ 提供了特征 $\mathrm{f_j}$ 的像素坐标。

通过测量窗口内两帧(第一和最后一帧)图像间的特征跟踪和三角化得到特征点位置的初值 $\tilde{\mathrm{p}}_{\mathrm{f_j}}^n$。利用逆深度高斯牛顿迭代优化的方式得到特征点位置的迭代解 $\mathrm{p}_{\mathrm{f_j}}^n$。

对于每一个特征测量，可以给出经过内参校正的特征坐标：

$$\hat{\mathrm{z}}_i^j = \begin{bmatrix} \mathrm{u}_\mathrm{i}' & \mathrm{v}_\mathrm{i}' \end{bmatrix}^T = \begin{bmatrix} \frac{u_i - c_\mathrm{u}}{f_u} & \frac{v_i - c_v}{f_v} \end{bmatrix}^T \tag{20}$$

首先通过最小二乘算法利用前两个相机系 $\mathrm{C_1}$ 和 $\mathrm{C_2}$ 估计特征 $\mathrm{f_j}$ 在相机系 $\mathrm{C_1}$ 内的位置，

$$\hat{\mathrm{p}}_{C_1 f_j}^{C_1} = \begin{bmatrix} \hat{X}_{C_1}^{(j)} & \hat{Y}_{C_1}^{(j)} & \hat{Z}_{C_1}^{(j)} \end{bmatrix}^\mathrm{T} = \lambda \rho_{f_j}^{C_1} \tag{21}$$

其中，

$$\rho_{f_j}^{C_1} = \frac{1}{\sqrt{\mathrm{u}_i'^2 + v_i'^2 + 1}} \begin{bmatrix} \mathrm{u}_i' & v_i' & 1 \end{bmatrix}^T \tag{22}$$

式(22)表示 $\mathrm{C_i}$ 系中特征点 $\mathrm{f_j}$ 的单位向量。

特征 $\mathrm{f_j}$ 在两个相机系 $\mathrm{C_1}$ 和 $\mathrm{C_2}$ 内的特征分别应用式(21)可得：

$$\hat{\mathrm{p}}_{C_1 f_j}^{C_1} = \lambda_1 \rho_{f_j}^{C_1} \tag{23a}$$

$$\hat{\mathrm{p}}_{C_2 f_j}^{C_2} = \lambda_2 \rho_{f_j}^{C_2} \tag{23b}$$

式(23a)减去式(23b)可得：

$$\hat{\mathrm{p}}_{C_1 f_j}^{C_1} - \hat{\mathrm{p}}_{C_2 f_j}^{C_2} = \lambda_1 \rho_{f_j}^{C_1} - \lambda_2 \rho_{f_j}^{C_2}$$

其中，$\hat{\mathrm{p}}_{C_1 f_j}^{C_1} - \hat{\mathrm{p}}_{C_2 f_j}^{C_2} = \hat{p}_{C_1 C_2}^{C_1}$ 表示 C_2 系到 C_1 系的位移量在 C_1 系的投影。

$$\lambda_1 \rho_{f_j}^{C_1} - \lambda_2 \rho_{f_j}^{C_2} = \begin{bmatrix} \rho_{f_j}^{C_1} & - \rho_{f_j}^{C_2} \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix}$$

根据上述分析可得：

$$\begin{bmatrix} \rho_{f_j}^{C_1} & - \rho_{f_j}^{C_2} \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = \hat{p}_{C_1 C_2}^{C_1} \tag{24}$$

式(24)中，令

$$\mathrm{A} = \begin{bmatrix} \rho_{f_j}^{C_1} & - \rho_{f_j}^{C_2} \end{bmatrix}$$

$$\lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix}$$

$$b = \hat{p}_{C_1 C_2}^{C_1}$$

则式(24)就可以列写成最常见的：

$$A\lambda = b \tag{25}$$

可用最小二乘法求解,得

$$\lambda = (\mathrm{A}^T A)^{-1} A^T b$$

解出式(25)后可以把 λ 回代式(21),可得特征点在 C_1 系内得坐标，进而可以求出特征在相机系 C_i 的位置。

$$\hat{\mathrm{p}}_{C_\mathrm{i} f_j}^{C_i} = C_1^i \hat{\mathrm{p}}_{C_1 f_j}^{C_1} + \hat{p}_{C_\mathrm{i} C_1}^{C_i} \tag{26}$$

其中，

$\hat{\mathrm{p}}_{C_\mathrm{i} f_j}^{C_i}$ 表示特征 $\mathrm{f_j}$ 在 $\mathrm{C_i}$ 系内的位置；

C_1^i 表示相机系1与相机系i的姿态转换矩阵，可由INS联合获取；

$\hat{\mathrm{p}}_{C_1 f_j}^{C_1}$ 表示特征 $\mathrm{f_j}$ 在 C_1 系内的位置；

$\hat{p}_{C_\mathrm{i} C_1}^{C_i}$ 表示 C_1 系与 $\mathrm{C_i}$ 系的相对位置在 $\mathrm{C_i}$ 内的投影，也可由INS联合获取；

式(26)表示利用最小二乘和INS或状态估值可以得到特征点再各个相机系内的位置，这里的量会作为量测，其中包含了状态向量信息。在后面迭代优化会用视觉投影误差进一步修正特征位置，又

含有量测信息，可见形成的最终量测是高度非线性的。至此已经可以求出特征点 f_j 在各个相机系的初始位置估值，后面使用迭代优化的方式进一步求取更精确的值。

定义如下向量：

$$\hat{y} = \begin{bmatrix} \alpha & \beta & \gamma \end{bmatrix}^T = \frac{1}{\hat{Z}_{C_1}^{(j)}} \begin{bmatrix} \hat{X}_{C_1}^{(j)} & \hat{Y}_{C_1}^{(j)} & 1 \end{bmatrix}^T \tag{27}$$

再定义三个函数：

$$\begin{bmatrix} h_1(\hat{y}) \\ h_2(\hat{y}) \\ h_3(\hat{y}) \end{bmatrix} = \hat{C}_1^i \begin{bmatrix} \alpha \\ \beta \\ 1 \end{bmatrix} + \gamma \hat{p}_{C_i C_1}^{C_i} \tag{28}$$

式(28)表示参数化的特征位置。

特征测量误差可以重新列写如下：

$$e(\hat{y}) = \hat{z}_i^{(j)} - \frac{1}{h_3(\hat{y})} \begin{bmatrix} h_1(\hat{y}) \\ h_2(\hat{y}) \end{bmatrix} \tag{29}$$

式(29)表示特征点的预测与量测位置的残差，可用于迭代优化。迭代优化过程利用了滑动窗内所有的相机位姿，可以平滑误差，求出的特征点位置比较精确。

采用高斯牛顿迭代需要求取残差对参数的偏导 $J = \frac{\partial e}{\partial \hat{y}}$， 按照参考文献给出带信息矩阵 Ω 的增量估计公式：

$$J^T \Omega J \delta y = -J^T \Omega e(\hat{y}) \tag{30}$$

其中，

$$\Omega = diag\left\{ \frac{1}{\sigma_{1u'}^2}, \frac{1}{\sigma_{1v'}^2} \quad \cdots \quad \frac{1}{\sigma_{Nu'}^2}, \frac{1}{\sigma_{Nv'}^2} \right\}$$

按照式(30)迭代更新即可的最终的特征点位值估计。

然后可以计算出特征点在世界系的投影：

$$\hat{p}_{f_j}^n = C_i^n \hat{p}_{C_i f_j}^{C_i} + \hat{p}_{C_1}^n \tag{31}$$

3.7.2 量测量构造

估计出特征点的精确位置后可以构造最终的量测残差 $r_i^{(j)}$，式(19)重新列写如下：

$$r_i^{(j)} = z_i^{(j)} - \hat{z}_i^{(j)} \tag{32}$$

其中，

其中，

$$\hat{z}_i^{(j)} = \frac{1}{\hat{Z}_{C_i}^{(j)}} \begin{bmatrix} \hat{X}_{C_i}^{(j)} & \hat{Y}_{C_i}^{(j)} \end{bmatrix}^T \tag{33}$$

式(33)的特征点坐标 $\hat{p}_{C_i f_j}^{C_i}$ 可以由下式构造

$$\begin{aligned}\hat{\mathrm{p}}_{C_i f_j}^{C_i} &= \begin{bmatrix}\hat{\mathrm{X}}_{C_i}^{(j)} & \hat{\mathrm{Y}}_{C_i}^{(j)} & \hat{\mathrm{Z}}_{C_i}^{(j)}\end{bmatrix}^T \\ &= C_n^{C_i}(\hat{p}_{f_j}^n - \hat{p}_{C_i}^n)\end{aligned} \tag{34}$$

相机相对世界系的位姿 $\hat{p}_{C_i}^n$ 、 $C_n^{C_i}$ 可以由INS和状态估值计算

式(32)是非线性的，可以用相机位置和特征点位置对式(32)进行线性化，可以把式(32)重新列写如下：

$$\mathrm{r}_i^{(j)} \approx \mathrm{H}_{\chi \mathrm{i}}^{(\mathrm{j})}\tilde{\chi}_i + \mathrm{H}_{\mathrm{fi}}^{(\mathrm{j})}\tilde{p}_{f_j}^n + n_i^{(j)} \tag{35}$$

其中，

$\tilde{p}_{f_j}^n$ 和 $\tilde{\chi}_i$ 分别式特征点位置和误差状态向量；

$\mathrm{H}_{\mathrm{fi}}^{(\mathrm{j})}$ 是残差对特征点位置的雅可比矩阵；

$\mathrm{H}_{\chi \mathrm{i}}^{(\mathrm{j})}$ 是残差对误差状态向量的雅可比矩阵，只在相应相机状态处不为零；

利用特征点在 C_i 系的坐标可以得到链式法则

$$\mathrm{H}_{\chi \mathrm{i}}^{(\mathrm{j})} = \frac{\partial z_{\mathrm{i}}^{(\mathrm{j})}}{\partial \mathrm{p}_{\mathrm{f_j}}^{\mathrm{C_i}}} \bullet \frac{\partial \mathrm{p}_{\mathrm{f_j}}^{\mathrm{C_i}}}{\partial \chi_{C_i}} \tag{36a}$$

$$\mathrm{H}_{\mathrm{f_i}}^{(\mathrm{j})} = \frac{\partial z_{\mathrm{i}}^{(\mathrm{j})}}{\partial \mathrm{p}_{\mathrm{f_j}}^{\mathrm{C_i}}} \bullet \frac{\partial \mathrm{p}_{\mathrm{f_j}}^{\mathrm{C_i}}}{\partial \mathrm{p}_{\mathrm{f_j}}^{\mathrm{n}}} \tag{36b}$$

$\frac{\partial z_{\mathrm{i}}^{(\mathrm{j})}}{\partial \mathrm{p}_{\mathrm{f_j}}^{\mathrm{C_i}}}$ 可以由定义得到

$$\frac{\partial z_{\mathrm{i}}^{(\mathrm{j})}}{\partial \mathrm{p}_{\mathrm{f_j}}^{\mathrm{C_i}}} = \begin{bmatrix} 1 & 0 & -\dfrac{\hat{\mathrm{X}}_j^{C_i}}{\hat{\mathrm{Z}}_j^{C_i 2}} \\ 0 & 1 & -\dfrac{\hat{\mathrm{Y}}_j^{C_i}}{\hat{\mathrm{Z}}_j^{C_i 2}} \end{bmatrix}$$

$\frac{\partial \mathrm{p}_{\mathrm{f_j}}^{\mathrm{C_i}}}{\partial \chi_{C_i}}$ 可以由式(34)对相机状态求偏导得到

$$\frac{\partial \mathrm{p}_{\mathrm{f_j}}^{\mathrm{C_i}}}{\partial \chi_{C_i}} = \begin{bmatrix} \dfrac{\partial \mathrm{p}_{\mathrm{f_j}}^{\mathrm{C_i}}}{\partial \phi_{C_i}} & \dfrac{\partial \mathrm{p}_{\mathrm{f_j}}^{\mathrm{C_i}}}{\partial \delta p_{\mathrm{C_i}}^n} \end{bmatrix} = \begin{bmatrix} -C_n^{C_i}(p_{f_j}^n \times) & -C_n^{C_i} \end{bmatrix}$$

$$\frac{\partial \mathrm{p}_{\mathrm{j}}^{\mathrm{C_i}}}{\partial \mathrm{p}_{\mathrm{j}}^{\mathrm{n}}} = -C_n^{C_i}$$

根据上述公式即可计算出式(36)的各个分量。

$\tilde{p}_{f_j}^n$ 的计算使用相机位姿和迭代优化实现，因此 $\tilde{p}_{f_j}^n$ 的不确定性与相机位姿的有关。为了确保

$\tilde{p}_{f_j}^{n}$ 不影响残差的不确定性，残差方程投影到 H_{f}^{j} 的零空间

$$\begin{aligned} \mathrm{r}_0^{(j)} &= V^T r^{(j)} \\ &= V^T \mathrm{H}_{\chi}^{(\mathrm{j})} \tilde{\chi} + V^T n_i^{(j)} \\ &= \mathrm{H}_{\chi,0}^{(\mathrm{j})} \tilde{\chi} + V^T n_i^{(j)} \end{aligned} \tag{36}$$

此后就可以按照正常的卡尔曼滤波算法来处理了。