

# 第 7 讲 VINS 初始化和 VIO 系统

贺一家，高翔，崔华坤

2019 年 7 月 28 日

## ① VIO 相关知识回顾

## ② VINS 鲁棒初始化

估计外参数旋转  $\mathbf{q}_{bc}$

估计陀螺仪 bias

估计重力向量，速度和尺度

优化重力向量  $\mathbf{g}^{c0}$

对齐导航世界坐标系  $w$

## ③ VINS 系统

VINS 系统流程

VINS 滑动窗口优化

## ④ 作业

## Section 1

# VIO 相关知识回顾

# IMU 预积分技术

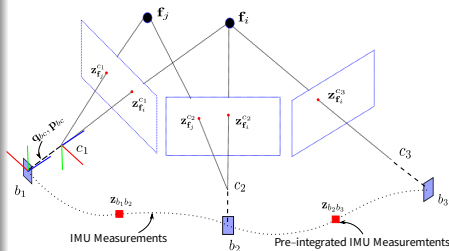
## IMU 传感器模型

$$\begin{aligned}\tilde{\omega}^b &= \omega^b + \mathbf{b}^g + \mathbf{n}^g \\ \tilde{\mathbf{a}}^b &= \mathbf{q}_{bw}(\mathbf{a}^w + \mathbf{g}^w) + \mathbf{b}^a + \mathbf{n}^a\end{aligned}\quad (1)$$

## IMU 预积分

将一段时间内的 IMU 数据直接积分起来就能得到两时刻  $i, j$  之间关于 IMU 的测量约束，即 **预积分量**：

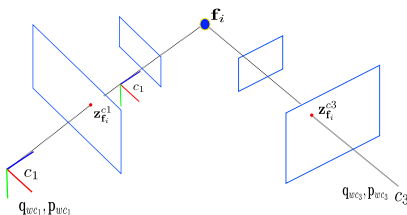
$$\begin{aligned}\alpha_{b_i b_j} &= \int \int_{t \in [i, j]} (\mathbf{q}_{b_i b_t} \mathbf{a}^{b_t}) \delta t^2 \\ \beta_{b_i b_j} &= \int_{t \in [i, j]} (\mathbf{q}_{b_i b_t} \mathbf{a}^{b_t}) \delta t \\ \mathbf{q}_{b_i b_j} &= \int_{t \in [i, j]} \mathbf{q}_{b_i b_t} \otimes \begin{bmatrix} 0 \\ \frac{1}{2} \omega^{b_t} \end{bmatrix} \delta t\end{aligned}\quad (2)$$



# 视觉几何基础

## 视觉技术

- ① 已知俩图像：特征点提取 (fast)，匹配 (光流，特征描述子)。
- ② 已知俩图像特征匹配点：利用对极几何约束 (E 矩阵，H 矩阵)，计算两图像之间的 pose (update to scale)。
- ③ 已知相机 pose, 已知特征点二维坐标：通过三角化得到三维坐标。
- ④ 已知 3d 点，2d 特征点：通过 Perspective-n-Point(PnP) 求取新的相机 pose。





## Section 2

# VINS 鲁棒初始化

# 视觉和 IMU 之间的联系

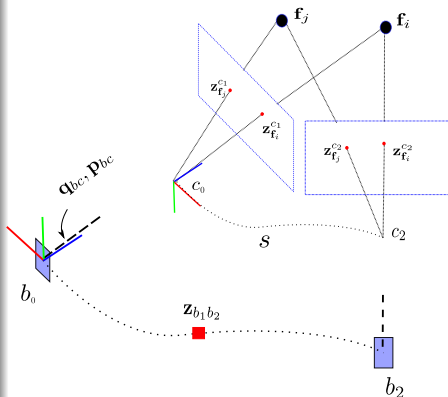
## 几何约束

考虑相机坐标系  $c_0$  为世界坐标系，则利用外参数  $\mathbf{q}_{bc}, \mathbf{t}_{bc}$  构建等式

$$\begin{aligned}\mathbf{q}_{c_0 b_k} &= \mathbf{q}_{c_0 c_k} \otimes \mathbf{q}_{bc}^{-1} \\ s\bar{\mathbf{p}}_{c_0 b_k} &= s\bar{\mathbf{p}}_{c_0 c_k} - \mathbf{R}_{c_0 b_k} \mathbf{p}_{bc}\end{aligned}\quad (3)$$

其中， $s$  为尺度因子， $\bar{\mathbf{p}}$  表示非米制单位的轨迹。等式(3)等价于

$$\begin{aligned}\bar{\mathbf{p}}_{c_0 b_k} &= \bar{\mathbf{p}}_{c_0 c_k} - \frac{1}{s} \mathbf{R}_{c_0 b_k} \mathbf{p}_{bc} \\ \bar{\mathbf{p}}_{c_0 c_k} &= \frac{1}{s} \mathbf{R}_{c_0 b_k} \mathbf{p}_{bc} + \bar{\mathbf{p}}_{c_0 b_k}\end{aligned}\quad (4)$$





# 视觉 IMU 对齐流程

## 估计流程

- ① 旋转外参数  $\mathbf{q}_{bc}$  未知, 则先估计旋转外参数.
- ② 利用旋转约束估计陀螺仪 bias.

$$\mathbf{q}_{c_0 b_k} = \mathbf{q}_{c_0 c_k} \otimes \mathbf{q}_{bc}^{-1}$$

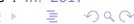
- ③ 利用平移约束估计重力方向, 速度, 以及尺度初始值.

$$s\bar{\mathbf{p}}_{c_0 b_k} = s\bar{\mathbf{p}}_{c_0 c_k} - \mathbf{R}_{c_0 b_k} \mathbf{p}_{bc}$$

- ④ 对重力向量  $\mathbf{g}^{c_0}$  进行进一步优化.
- ⑤ 求解世界坐标系  $w$  和初始相机坐标系  $c_0$  之间的旋转矩阵  $\mathbf{q}_{wc_0}$ , 并将轨迹对齐到世界坐标系.

以 VINS-mono 方法为基础进行讲解<sup>1,2</sup>.

<sup>1</sup>Zhenfei Yang and Shaojie Shen. "Monocular visual-inertial state estimation with online initialization and camera-IMU extrinsic calibration". In: *IEEE Transactions on Automation Science and Engineering* 14.1 (2016), pp. 39–51.

<sup>2</sup>Tong Qin and Shaojie Shen. "Robust initialization of monocular visual-inertial estimation on aerial robots". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 4225–4232. 

# 利用旋转约束估计外参数旋转 $\mathbf{q}_{bc}$

相邻两时刻  $k, k+1$  之间有：IMU 旋转积分  $\mathbf{q}_{b_k b_{k+1}}$ ，视觉测量  $\mathbf{q}_{c_k c_{k+1}}$ 。则有：

$$\mathbf{q}_{b_k b_{k+1}} \otimes \mathbf{q}_{bc} = \mathbf{q}_{bc} \otimes \mathbf{q}_{c_k c_{k+1}} \quad (5)$$

上式可写成：

$$\left( [\mathbf{q}_{b_k b_{k+1}}]_L - [\mathbf{q}_{c_k c_{k+1}}]_R \right) \mathbf{q}_{bc} = \mathbf{Q}_{k+1}^k \cdot \mathbf{q}_{bc} = \mathbf{0} \quad (6)$$

其中， $[\cdot]_L, [\cdot]_R$  表示 left and right quaternion multiplication。

将多个时刻线性方程(6)累计起来，并加上鲁棒核权重得到：

$$\begin{bmatrix} w_1^0 \cdot \mathbf{Q}_1^0 \\ w_2^1 \cdot \mathbf{Q}_2^1 \\ \vdots \\ w_N^{N-1} \cdot \mathbf{Q}_N^{N-1} \end{bmatrix} \mathbf{q}_{bc} = \mathbf{Q}_N \cdot \mathbf{q}_{bc} = \mathbf{0} \quad (7)$$

其中：

$$w_{k+1}^k = \begin{cases} 1, & r_{k+1}^k < \text{threshold} \\ \frac{\text{threshold}}{r_{k+1}^k}, & \text{otherwise} \end{cases} \quad (8)$$

由旋转矩阵和轴角之间的关系  $\text{tr}(\mathbf{R}) = 1 + 2 \cos \theta$ ，能得到角度误差  $r$  的计算为：

$$r_{k+1}^k = \text{acos} \left( \left( \text{tr} \left( \hat{\mathbf{R}}_{bc}^{-1} \mathbf{R}_{b_k b_{k+1}}^{-1} \hat{\mathbf{R}}_{bc} \mathbf{R}_{c_k c_{k+1}} \right) - 1 \right) / 2 \right) \quad (9)$$

公式(7)的求解同样采用 SVD 分解，即最小奇异值对应的奇异向量。

具体代码见：[initial\\_ex\\_rotation.cpp](#) 函数 `CalibrationExRotation()`。

# 基于旋转约束的 Gyroscope Bias

如果外参数  $\mathbf{q}_{bc}$  已标定好，利用旋转约束，可估计陀螺仪 bias:

$$\arg \min_{\delta \mathbf{b}^g} \sum_{k \in B} \left\| 2 \left[ \mathbf{q}_{c_0 b_{k+1}}^{-1} \otimes \mathbf{q}_{c_0 b_k} \otimes \mathbf{q}_{b_k b_{k+1}} \right]_{xyz} \right\|^2 \quad (10)$$

其中,  $B$  表示所有的图像关键帧集合, 另有预积分的一阶泰勒近似:

$$\mathbf{q}_{b_k b_{k+1}} \approx \hat{\mathbf{q}}_{b_k b_{k+1}} \otimes \left[ \begin{array}{c} 1 \\ \frac{1}{2} \mathbf{J}_{bg}^{\mathbf{q}} \delta \mathbf{b}^g \end{array} \right] \quad (11)$$

公式(10)为普通的最小二乘问题, 求取雅克比矩阵, 构建正定方程  $\mathbf{H}\mathbf{X} = \mathbf{b}$  即可以求解。

具体代码见: `initial_alignent.cpp` 函数 `solveGyroscopeBias()`。

# 初始化速度、重力和尺度因子

## 需要估计的变量

$$\mathcal{X}_I = [\mathbf{v}_0^{b_0}, \mathbf{v}_1^{b_1}, \dots, \mathbf{v}_n^{b_n}, \mathbf{g}^{c_0}, s]^\top \quad (12)$$

其中， $\mathbf{v}_k^{b_k}$  表示  $k$  时刻 body 坐标系的速度在 body 坐标系下的表示。  
 $\mathbf{g}^{c_0}$  为重力向量在第 0 帧相机坐标系下的表示。 $s$  表示尺度因子，将视觉轨迹拉伸到米制单位。

## 回顾：预积分量约束

世界坐标系  $w$  下有

$$\begin{aligned} \alpha_{b_i b_j} &= \mathbf{q}_{b_i w} (\mathbf{p}_{w b_j} - \mathbf{p}_{w b_i} - \mathbf{v}_i^w \Delta t + \frac{1}{2} \mathbf{g}^w \Delta t^2) \\ \beta_{b_i b_j} &= \mathbf{q}_{b_i w} (\mathbf{v}_j^w - \mathbf{v}_i^w + \mathbf{g}^w \Delta t) \end{aligned} \quad (13)$$

将世界坐标系  $w$  换成相机初始时刻坐标系  $c_0$  有

$$\begin{aligned}\alpha_{b_k b_{k+1}} &= \mathbf{R}_{b_k c_0} \left( s \left( \bar{\mathbf{p}}_{c_0 b_{k+1}} - \bar{\mathbf{p}}_{c_0 b_k} \right) + \frac{1}{2} \mathbf{g}^{c_0} \Delta t_k^2 - \mathbf{R}_{c_0 b_k} \mathbf{v}_k^{b_k} \Delta t_k \right) \\ \beta_{b_k b_{k+1}} &= \mathbf{R}_{b_k c_0} \left( \mathbf{R}_{c_0 b_{k+1}} \mathbf{v}_{k+1}^{b_{k+1}} + \mathbf{g}^{c_0} \Delta t_k - \mathbf{R}_{c_0 b_k} \mathbf{v}_k^{b_k} \right)\end{aligned}\quad (14)$$

将公式(3)代入公式(14)进行简单整理有：

$$\begin{aligned}\alpha_{b_k b_{k+1}} &= s \mathbf{R}_{b_k c_0} \left( \bar{\mathbf{p}}_{c_0 c_{k+1}} - \bar{\mathbf{p}}_{c_0 c_k} \right) \\ &\quad - \mathbf{R}_{b_k c_0} \mathbf{R}_{c_0 b_{k+1}} \mathbf{p}_{bc} + \mathbf{p}_{bc} + \frac{1}{2} \mathbf{R}_{b_k c_0} \mathbf{g}^{c_0} \Delta t_k^2 - \mathbf{v}_k^{b_k} \Delta t_k\end{aligned}\quad (15)$$

将待估计变量放到方程右边，有：

$$\hat{\mathbf{z}}_{b_{k+1}}^{b_k} = \begin{bmatrix} \hat{\alpha}_{b_k b_{k+1}} - \mathbf{p}_{bc} + \mathbf{R}_{b_k c_0} \mathbf{R}_{c_0 b_{k+1}} \mathbf{p}_{bc} \\ \hat{\beta}_{b_k b_{k+1}} \end{bmatrix} = \mathbf{H}_{b_{k+1}}^{b_k} \chi_I^k + \mathbf{n}_{b_{k+1}}^{b_k} \quad (16)$$

公式(16)中：

$$\mathcal{X}_I^k = [\mathbf{v}_k^{b_k}, \mathbf{v}_{k+1}^{b_{k+1}}, \mathbf{g}^{c_0}, s]^\top$$

$$\mathbf{H}_{b_{k+1}}^{b_k} = \begin{bmatrix} -\mathbf{I}\Delta t_k & \mathbf{0} & \frac{1}{2}\mathbf{R}_{b_k c_0}\Delta t_k^2 & \mathbf{R}_{b_k c_0}(\bar{\mathbf{p}}_{c_{k+1}}^{c_0} - \bar{\mathbf{p}}_{c_k}^{c_0}) \\ -\mathbf{I} & \mathbf{R}_{b_k c_0}\mathbf{R}_{c_0 b_{k+1}} & \mathbf{R}_{b_k c_0}\Delta t_k & \mathbf{0} \end{bmatrix} \quad (17)$$

公式(16)转化成线性最小二乘问题对状态量进行求解：

$$\min_{\mathcal{X}_I} \sum_{k \in \mathcal{B}} \left\| \hat{\mathbf{z}}_{b_{k+1}}^{b_k} - \mathbf{H}_{b_{k+1}}^{b_k} \mathcal{X}_I^k \right\|^2 \quad (18)$$

具体代码见：initial\_aligment.cpp 函数 LinearAlignment()。

# 优化重力向量 $\mathbf{g}^{c0}$

疑问：为什么需要优化重力向量

利用公式(16)求解重力向量  $\mathbf{g}^{c0}$  过程中，并没有加入模长限制  $\|\mathbf{g}^{c0}\| = 9.81$ 。三维变量  $\mathbf{g}^{c0}$  实际只有**两个自由度**。

## 重力向量参数化

三维向量自由度为 2，可以采用球面坐标进行参数化：

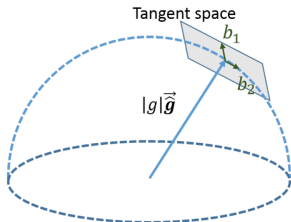
$$\hat{\mathbf{g}}^{c0} = \|\mathbf{g}\| \cdot \hat{\mathbf{g}}^{c0} + w_1 \vec{b}_1 + w_2 \vec{b}_2 \quad (19)$$

其中， $w_1, w_2$  为待优化变量

$$\vec{b}_1 = \begin{cases} (\hat{\mathbf{g}}^{c0} \times [1, 0, 0]), & \hat{\mathbf{g}}^{c0} \neq [1, 0, 0]^T \\ (\hat{\mathbf{g}}^{c0} \times [0, 0, 1]), & \text{otherwise} \end{cases}$$

$$\vec{b}_2 = \hat{\mathbf{g}}^{c0} \times \vec{b}_1$$

(20)



图：两自由度的重力矢量参数化



将公式 (19) 代入公式(16), 待优化变量变为:

$$\mathcal{X}_I^k = \begin{bmatrix} \mathbf{v}_k^{b_k} \\ \mathbf{v}_{k+1}^{b_{k+1}} \\ \mathbf{g}^{c0} \\ s \end{bmatrix} \rightarrow \begin{bmatrix} \mathbf{v}_k^{b_k} \\ \mathbf{v}_{k+1}^{b_{k+1}} \\ \mathbf{w}^{c0} \\ s \end{bmatrix} \quad (21)$$

公式(16)中的观测方程变为:

$$\hat{\mathbf{z}}_{b_k b_{k+1}}^{b_k} = \begin{bmatrix} \alpha_{b_k b_{k+1}} - \mathbf{p}_{bc} + \mathbf{R}_{b_k c0} \mathbf{R}_{c0 b_{k+1}} \mathbf{p}_{bc} - \frac{1}{2} \mathbf{R}_{b_k c0} \Delta t_k^2 \|g\| \cdot \hat{\mathbf{g}}^{c0} \\ \beta_{b_k b_{k+1}} - \mathbf{R}_{b_k c0} \Delta t_k \|g\| \cdot \hat{\mathbf{g}}^{c0} \end{bmatrix} \quad (22)$$

采用最小二乘对  $\mathcal{X}_I$  进行重新优化。

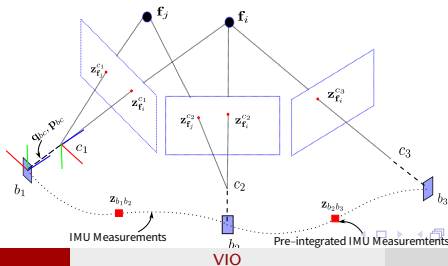
# 将相机坐标系对齐世界坐标系

## 对齐流程

- 1 找到  $c_0$  到  $w$  系的旋转矩阵  $\mathbf{R}_{wc_0} = \exp([\theta \mathbf{u}])$

$$\mathbf{u} = \frac{\hat{\mathbf{g}}^{c_0} \times \hat{\mathbf{g}}^w}{\|\hat{\mathbf{g}}^{c_0} \times \hat{\mathbf{g}}^w\|}, \quad \theta = \text{atan2}(\|\hat{\mathbf{g}}^{c_0} \times \hat{\mathbf{g}}^w\|, \hat{\mathbf{g}}^{c_0} \cdot \hat{\mathbf{g}}^w) \quad (23)$$

- 2 把所有  $c_0$  坐标系下的变量旋转到  $w$  下。
- 3 把相机平移和特征点尺度恢复到米制单位。
- 4 至此，完成了系统初始化过程。



# VINS 初始化拓展

## 疑问

- ① 加速度 bias 为何没有估计？
- ② 平移外参数  $p_{bc}$  为何没有初始化？

## 其他初始化方法

- ① 静止初始化：直接用加速度测量重力方向，初始速度为 0。
- ② 除 vins-mono 外的其他运动初始化方案<sup>a, b</sup>。

<sup>a</sup> Janne Mustaniemi et al. "Inertial-based scale estimation for structure from motion on mobile devices". In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE. 2017, pp. 4394–4401.

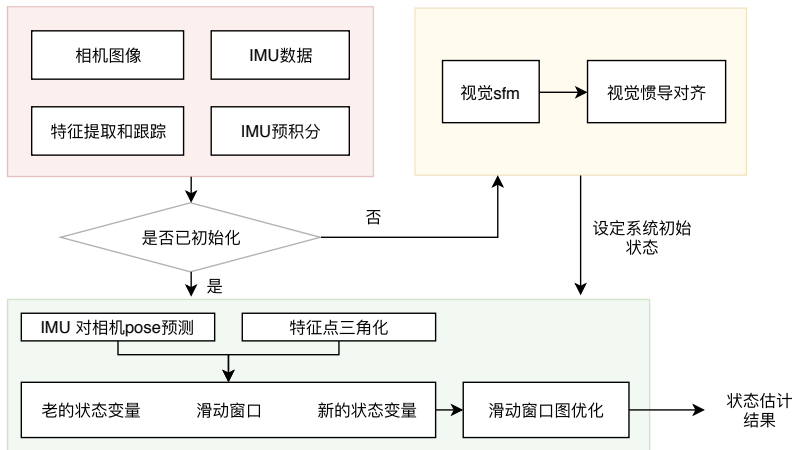
<sup>b</sup> Javier Domínguez-Conti et al. "Visual-Inertial SLAM Initialization: A General Linear Formulation and a Gravity-Observing Non-Linear Optimization". In: 2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR). IEEE. 2018, pp. 37–45.

## Section 3

# VINS 系统

## VINS 系统三大块

- ① 前端，数据处理：特征提取匹配，imu 积分
- ② 初始化：系统初始状态变量（重力方向，速度，尺度等等）
- ③ 后端：滑动窗口优化



## 后端滑动窗口优化

VINS 系统优化的状态变量为：

$$\begin{aligned}
 \mathcal{X} &= [\mathbf{x}_0, \mathbf{x}_1, \cdots \mathbf{x}_n, \mathbf{x}_c^b, \lambda_0, \lambda_1, \cdots \lambda_m] \\
 \mathbf{x}_k &= [\mathbf{p}_{wb_k}, \mathbf{v}_k^w, \mathbf{q}_{wb_k}, \mathbf{b}_k^a, \mathbf{b}_k^g], k \in [0, n] \\
 \mathbf{x}_{bc} &= [\mathbf{p}_{bc}, \mathbf{q}_{bc}]
 \end{aligned} \tag{24}$$

通过最小化滑动窗口中的残差项来估计系统的状态变量：

$$\min_{\mathcal{X}} \left\{ \underbrace{\|\mathbf{r}_p - \mathbf{J}_p \mathcal{X}\|^2}_{\text{先验}} + \underbrace{\sum_{k \in \mathcal{B}} \left\| \mathbf{r}_B \left( \hat{\mathbf{z}}_{b_{k+1}}^{b_k}, \mathcal{X} \right) \right\|_{\Sigma_{b_k b_{k+1}}}^2}_{\text{IMU 残差}} + \underbrace{\sum_{(l,j) \in \mathcal{C}} \rho \left( \left\| \mathbf{r}_C \left( \hat{\mathbf{z}}_l^{c_j}, \mathcal{X} \right) \right\|_{\Sigma_l^{c_j}}^2 \right)}_{\text{视觉残差}} \right\} \tag{25}$$

注意：其中鲁棒核函数  $\rho(\cdot)$  仅处理视觉 outlier.

## Section 4

# 作业

## 作业

- ① 将第二讲的仿真数据集（视觉特征，imu 数据）接入我们的 VINS 代码，并运行出轨迹结果。
  - 仿真数据集无噪声
  - 仿真数据集有噪声（不同噪声设定时，需要配置 vins 中 imu noise 大小。）

