



深蓝学院
shenlanxueyuan.com

第八章作业提示

主讲人 会打篮球的猫



- ① 基础：请仿照 Eq.(12)-(14)，推导当 vins 中对特征采用逆深度参数时，基于特征匀速模型的重投影误差计算形式。

设重投影误差是从 i 时刻图像投到 j 时刻图像。则使用逆深度的重投影过程可以简单描述为：

- ① 图像 i 上的观测 z_i 由相机内参转到归一化平面；
- ② 归一化平面坐标除以逆深度得到相机 i 系下的三维坐标 P^i ；
- ③ 根据相机 i 的位姿 R_i^w 和 p_i^w ，得到世界坐标系下的坐标 P^w ；
- ④ 根据相机 j 的位姿 R_j^w 和 p_j^w ，得到相机 j 系下的坐标 P^j ；
- ⑤ 将 P^j 除以 z 轴大小，并通过相机内参得到在图像 j 上预测的像素坐标；
- ⑥ 图像 j 上的观测 z_j 减去预测像素坐标，即得到重投影误差。

基础题

- ① 基础：请仿照 Eq.(12)-(14)，推导当 vins 中对特征采用逆深度参数时，基于特征匀速模型的重投影误差计算形式。

不考虑时间延迟，使用逆深度的重投影误差计算形式。

从 i 到 j 时刻构建重投影误差（无时间延迟）

$$X^{ci} = \frac{1}{\lambda_i} \cdot k^T \cdot \begin{bmatrix} z_i \\ 1 \end{bmatrix} \quad \dots \lambda_i \text{ 为逆深度, } k^T \text{ 表示由内参反投影}$$

$$X^w = R_{ci}^w \cdot X^{ci} + p_{ci}^w$$

$$X^{cj} = R_{wj}^c \cdot (X^w - p_{cj}^w)$$

$$e = z_j - \pi(X^{cj}) \quad \dots \pi(\cdot) \text{ 表示从相机坐标系三维坐标转化到像素坐标的一系列操作}$$

$$\therefore \begin{cases} e = z_j - \pi \left(R_{wj}^c \cdot (R_{ci}^w \cdot \frac{1}{\lambda_i} \cdot k^T \cdot \begin{bmatrix} z_i \\ 1 \end{bmatrix} + p_{ci}^w - p_{cj}^w) \right) \\ \begin{cases} z_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} & z_j = \begin{bmatrix} u_j \\ v_j \end{bmatrix} \end{cases} \end{cases}$$

基础题

- ① 基础：请仿照 Eq.(12)-(14)，推导当 vins 中对特征采用逆深度参数时，基于特征匀速模型的重投影误差计算形式。

考虑时间延迟，使用逆深度的重投影误差计算形式。

$$\begin{aligned} & \text{当考虑时间延迟时, } z_i \rightarrow z_i(t_d) \quad z_j \rightarrow z_j(t_d) \\ & \therefore \left\{ \begin{aligned} e &= z_j(t_d) - \pi \left[R_{ci}^g \cdot (R_{ci}^w \cdot \frac{1}{\lambda_i} \cdot k^{-1} \cdot \begin{bmatrix} z_i(t_d) \\ 1 \end{bmatrix}) + p_{ci}^w - p_{cj}^w \right] \\ z_i(t_d) &= \begin{bmatrix} u_i \\ v_i \end{bmatrix} + t_d \cdot V_i & z_j(t_d) &= \begin{bmatrix} u_j \\ v_j \end{bmatrix} + t_d \cdot V_j \\ k \text{ 时刻速度 } V_k &: V_k = \left(\begin{bmatrix} u_{k+1} \\ v_{k+1} \end{bmatrix} - \begin{bmatrix} u_k \\ v_k \end{bmatrix} \right) / (t_{k+1} - t_k) \end{aligned} \right. \end{aligned}$$

与无延时形式几乎相同。观测值中增加了延时项。

- ② 提升：阅读论文^a，总结基于 B 样条的时间戳估计算法流程，梳理论文公式。

Unified Temporal and Spatial Calibration for Multi-Sensor Systems

Paul Furgale, Joern Rehder, and Roland Siegwart

Abstract—In order to increase accuracy and robustness in state estimation for robotics, a growing number of applications rely on data from multiple complementary sensors. For the best performance in sensor fusion, these different sensors must be spatially and temporally registered with respect to each other. To this end, a number of approaches have been developed to estimate these system parameters in a two stage process, first estimating the time offset and subsequently solving for the spatial transformation between sensors.

In this work, we present on a novel framework for jointly estimating the temporal offset between measurements of different sensors and their spatial displacements with respect to each other. The approach is enabled by continuous-time batch estimation and extends previous work by seamlessly incorporating time offsets within the rigorous theoretical framework of maximum likelihood estimation.

Experimental results for a camera to inertial measurement unit (IMU) calibration prove the ability of this framework to accurately estimate time offsets up to a fraction of the smallest measurement period.

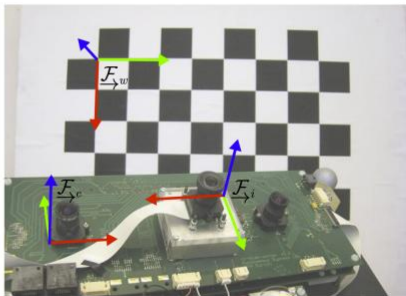


Fig. 1. This paper derives a unified framework for temporal and spatial calibration of multi-sensor systems such as this visual-inertial sensor used in the experiments. \mathcal{F}_w marks the inertial frame attached to a static calibration pattern, while \mathcal{F}_c and \mathcal{F}_i show the camera and the IMU frame respectively.

此论文的主要工作就是多传感器之间的时空标定，同时估计不同传感器之间的时间延时（time offset）以及外参。整个方法基于最大似然估计的理论框架，通过连续时间下的批量估计实现。

② 提升：阅读论文^a，总结基于 B 样条的时间戳估计算法流程，梳理论文公式。

A. Estimating Time Offsets using Basis Functions

随时间变化的状态量可以表示为有限个已知的基函数的加权和：

$$\Phi(t) := [\phi_1(t) \ \dots \ \phi_B(t)], \quad \mathbf{x}(t) := \Phi(t)\mathbf{c}, \quad (1)$$

如果要估计 \mathbf{x} ，则需要估计系数向量 \mathbf{c} 。当我们要从测量数据估计时间延时的时候，有如下残差形式：

$$\mathbf{e}_j := \mathbf{y}_j - \mathbf{h}(\mathbf{x}(t_j + d)), \quad (2)$$

d 就是未知的时间延时， \mathbf{y} 为观测值， $\mathbf{h}(\cdot)$ 表示预测函数。

接下来使用上述的基函数代替 \mathbf{x} ，则上式就可以写成如下形式：

$$\mathbf{e}_j = \mathbf{y}_j - \mathbf{h}(\Phi(t_j + d)\mathbf{c}), \quad (3)$$

然后就是进行线性化，从某一（工作）点进行一阶泰勒展开，对时间求雅克比矩阵，则有：

$$\mathbf{e}_j \approx \mathbf{y}_j - \mathbf{h}(\Phi(t_j + \bar{d})\mathbf{c}) - \mathbf{H}\dot{\Phi}(t_j + \bar{d})\mathbf{c}\Delta d, \quad (4)$$

其中容易看出，因为是嵌套形式，所以 \mathbf{H} 是 $\mathbf{h}(\cdot)$ 对 \mathbf{x} 的求导。且 $\Phi(t)$ 是对时间可导的。

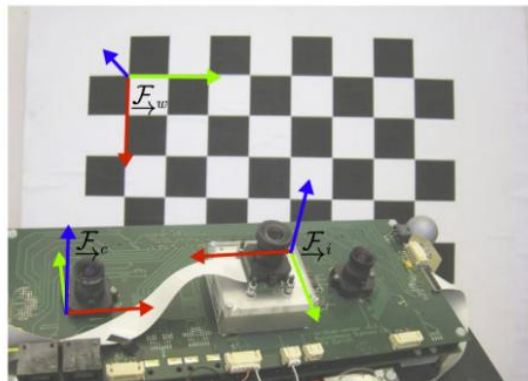
作者也是简单评价了上述过程，有两点好处，首先它允许在最大似然的严格理论框架下对延时进行估计；其次该问题在连续时间下评估非常方便。

② 提升：阅读论文^a，总结基于 B 样条的时间戳估计算法流程，梳理论文公式。

对时间延迟的通用（general）处理过程讲完以后，作者就将其带到了具体的例子中进一步说明，也就是相机和 IMU 的标定，对过程进行了详细的说明。

B. An Example: Camera/IMU Calibration

标定的目的是为了获得传感器之间的相对旋转、平移和时间延迟。通过在一个固定的标定板前面晃动传感器来采集数据，如下图所示：



简单描述一下涉及到的坐标系。有标定板所在的惯性世界系（W 系），整个估计也就是相对于 W 系的。在 IMU 系下测得角速度和加速度，相机坐标系位于光心，z 轴向前。这些也都是我们熟知的非常常规的坐标系定义。

B1、所要估计的参数包括：（1）重力方向，表示在 W 系下；

- （2）相机与 IMU 之间的变换，也就是外参；
- （3）相机与 IMU 之间的时间延迟；
- （4）IMU 的 pose；
- （5）加速度计和陀螺仪的零偏。

② 提升：阅读论文^a，总结基于 B 样条的时间戳估计算法流程，梳理论文公式。

B2、关于时变状态的参数化：也就是使用 B 样条曲线来描述，前文对时间的处理方法已经是初步引入了。

在此处，首先说明的是变换矩阵：

$$\mathbf{T}_{w,i}(t) := \begin{bmatrix} \mathbf{C}(\varphi(t)) & \mathbf{t}(t) \\ \mathbf{0}^T & 1 \end{bmatrix}, \quad (6)$$

包括旋转和平移，参数化为 6X1 的样条，使用三自由度的旋转和三自由度的平移。其中， $\varphi(t) := \Phi_{\varphi}(t)\mathbf{c}_{\varphi}$ 、 $\mathbf{t}(t) := \Phi_t(t)\mathbf{c}_t$ 。

W 系下的速度和加速度则表示为：

$$\mathbf{v}(t) = \dot{\mathbf{t}}(t) = \dot{\Phi}_t(t)\mathbf{c}_t, \quad \mathbf{a}(t) = \ddot{\mathbf{t}}(t) = \ddot{\Phi}_t(t)\mathbf{c}_t. \quad (7)$$

角速度则有如下形式：

$$\boldsymbol{\omega}(t) = \mathbf{S}(\varphi(t))\dot{\varphi}(t) = \mathbf{S}(\Phi(t)\mathbf{c}_{\varphi})\dot{\Phi}(t)\mathbf{c}_{\varphi}, \quad (8)$$

此论文中使用的是轴角来参数化旋转，因此角度大小和旋转轴分别为：

$$\text{angle } \varphi(t) = \sqrt{\varphi(t)^T \varphi(t)}, \quad \text{axis } \varphi(t) / \varphi(t)$$

B3、下面就是测量与过程模型描述：关于角速度和加速度的测量值就不再描述，简单说一下视觉测量，在 t_j+d 时刻看到了某个路标点 P，观测为 y ， t_j 为图像时间戳，由此就能得到视觉投影模型了。

因此测量方程总结如下：

$$\boldsymbol{\alpha}_k := \mathbf{C}(\varphi(t_k))^T (\mathbf{a}(t_k) - \mathbf{g}_w) + \mathbf{b}_a(t_k) + \mathbf{n}_{a_k}, \quad (9a)$$

$$\boldsymbol{\omega}_k := \mathbf{C}(\varphi(t_k))^T \boldsymbol{\omega}(t_k) + \mathbf{b}_{\omega}(t_k) + \mathbf{n}_{\omega_k}, \quad (9b)$$

$$\mathbf{y}_{mj} := \mathbf{h}(\mathbf{T}_{c,i} \mathbf{T}_{w,i}(t_j + d)^{-1} \mathbf{p}_w^m) + \mathbf{n}_{y_{mj}}, \quad (9c)$$

作者特别提及的是，因为时间延迟 d 可正可负，所以上述模型不会出现任何的信息损失。

对于 IMU 零偏的建模，就是熟悉的其导数是个高斯白噪声：

$$\dot{\mathbf{b}}_a(t) = \mathbf{w}_a(t) \quad \mathbf{w}_a(t) \sim \mathcal{GP}(\mathbf{0}, \mathbf{Q}_a \delta(t - t')) \quad (10a)$$

$$\dot{\mathbf{b}}_{\omega}(t) = \mathbf{w}_{\omega}(t) \quad \mathbf{w}_{\omega}(t) \sim \mathcal{GP}(\mathbf{0}, \mathbf{Q}_{\omega} \delta(t - t')) \quad (10b)$$

提升题

- ② 提升：阅读论文^a，总结基于 B 样条的时间戳估计算法流程，梳理论文公式。

$$\mathbf{e}_{y_{mj}} := \mathbf{y}_{mj} - \mathbf{h}(\mathbf{T}_{c,i} \mathbf{T}_{w,i}(t_j + d)^{-1} \mathbf{p}_w^m) \quad (11a)$$

$$J_y := \frac{1}{2} \sum_{j=1}^J \sum_{m=1}^M \mathbf{e}_{y_{mj}}^T \mathbf{R}_{y_{mj}}^{-1} \mathbf{e}_{y_{mj}} \quad (11b)$$

$$\mathbf{e}_{\alpha_k} := \alpha_k - \mathbf{C}(\varphi(t_k))^T (\mathbf{a}(t_k) - \mathbf{g}_w) + \mathbf{b}_a(t_k) \quad (11c)$$

$$J_\alpha := \frac{1}{2} \sum_{k=1}^K \mathbf{e}_{\alpha_k}^T \mathbf{R}_{\alpha_k}^{-1} \mathbf{e}_{\alpha_k} \quad (11d)$$

$$\mathbf{e}_{\omega_k} := \varpi_k - \mathbf{C}(\varphi(t_k))^T \omega(t_k) + \mathbf{b}_\omega(t_k) \quad (11e)$$

$$J_\omega := \frac{1}{2} \sum_{k=1}^K \mathbf{e}_{\omega_k}^T \mathbf{R}_{\omega_k}^{-1} \mathbf{e}_{\omega_k} \quad (11f)$$

$$\mathbf{e}_{b_a}(t) := \dot{\mathbf{b}}_a(t) \quad (11g)$$

$$J_{b_a} := \frac{1}{2} \int_{t_1}^{t_K} \mathbf{e}_{b_a}(\tau)^T \mathbf{Q}_a^{-1} \mathbf{e}_{b_a}(\tau) d\tau \quad (11h)$$

$$\mathbf{e}_{b_\omega}(t) := \dot{\mathbf{b}}_\omega(t) \quad (11i)$$

$$J_{b_\omega} := \frac{1}{2} \int_{t_1}^{t_K} \mathbf{e}_{b_\omega}(\tau)^T \mathbf{Q}_\omega^{-1} \mathbf{e}_{b_\omega}(\tau) d\tau \quad (11j)$$

提升题

② 提升：阅读论文^a，总结基于 B 样条的时间戳估计算法流程，梳理论文公式。

算法的流程和公式梳理完毕之后，有一些需要补充的东西，也就是在具体实施的时候的一些说明。

首先是一些假设：

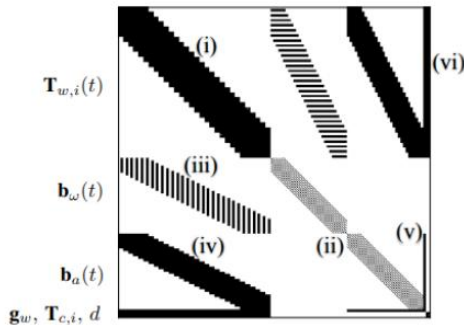
- ① 相机内参、IMU 噪声认为已知；
- ② 重力方向、相机与 IMU 的外参，都有一个初值；
- ③ 关于标定板的信息已知，从而能够表示出每个路标点在 W 系的位置；
- ④ 相机观测与标定板上的点的对应关系能够知道。

其次，对于时间延迟的初值设为 0。而 IMU 的位姿由 pnp 算法估计出粗略的相机位姿再结合外参确定。关于样条，则：

spline is initialized using the linear solution of Schoenberg and Reinsch (Chapter XIV of [17]).

IMU pose 使用 6 维 B 样条（5 次多项式），这种高阶表示使加速度为 3 次多项式，这样做能够准确捕捉标定过程中的传感器运动。

除了维数以外，节点个数同样影响着系统。受样条阶数和节点个数的影响，LM 法求解的方程矩阵可能非常大，但是它是稀疏的，在每次迭代中，使用了 CHOLMOD 来加速计算。



- ② 提升：阅读论文^a，总结基于 B 样条的时间戳估计算法流程，梳理论文公式。

最后，在优化过程中，和延时相关的图像测量误差项可能跨越样条曲线的节点边界，这会导致线性系统的稀疏形式在每次迭代中变化，如果稀疏形式不变的话，计算会特别快。

因此，图像误差项的雅克比矩阵会附加列扩展，以考虑对相邻 B 样条系数参数的可能依赖性。为此，代码接收以秒为单位的时间填充值，将时间延迟限制在其边界内。该值选太大会增加耗时，太小会导致优化失败，论文中选择的是 0.04。

routine. For the experiments in this paper, we used a time padding of 0.04 seconds.

兴趣题

- ③ 兴趣（不强求）：请推导初始化时旋转误差 Eq.(17) 对时间戳延迟 t_d 的雅克比，参考论文^b附录 D。

$$Error_{ij}(t_d) = \underbrace{\text{Log}([R_{ij} \cdot \text{Exp}(J_{ij} \cdot \delta t_d)]^T \cdot R_b^b \cdot \text{Exp}(-W_i t_d) \cdot R_i^i \cdot R_j^j \cdot \text{Exp}(W_j t_d) \cdot R_b^b)}_{R_1} \underbrace{\quad}_{R_2}$$

$$\text{设 } R_1 = [R_{ij} \cdot \text{Exp}(J_{ij} \cdot \delta t_d)]^T, R_2 = R_i^i \cdot R_j^j$$

$$\therefore Error_{ij}(t_d) = \text{Log}(R_1 \cdot \text{Exp}(-W_i t_d) \cdot R_2 \cdot \text{Exp}(W_j t_d) \cdot R_b^b)$$

$$\therefore Error_{ij}(t_d + \delta t) = \text{Log}(R_1 \cdot \text{Exp}(-W_i t_d - W_i \delta t) \cdot R_2 \cdot \text{Exp}(W_j t_d + W_j \delta t) \cdot R_b^b)$$

$$\text{BCH近似: } \approx \text{Log}(R_1 \cdot \text{Exp}(-J_i^i W_i \delta t) \cdot \text{Exp}(-W_i t_d) \cdot R_2 \cdot \text{Exp}(W_j t_d) \cdot \text{Exp}(J_j^j W_j \delta t) \cdot R_b^b)$$

$$\text{伴随性质: } = \text{Log}(\underbrace{\text{Exp}(-R_i J_i^i W_i \delta t) \cdot R_i \cdot \text{Exp}(-W_i t_d) \cdot R_2 \cdot \text{Exp}(W_j t_d) \cdot R_b^b \cdot \text{Exp}(R_j^j J_j^j W_j \delta t)}_{\text{Exp}(error(t_d))})$$

$$= \text{Log}(\text{Exp}(-R_i J_i^i W_i \delta t) \cdot \text{Exp}(error(t_d)) \cdot \text{Exp}(R_j^j J_j^j W_j \delta t))$$

$$\text{伴随性质: } = \text{Log}(\underbrace{\text{Exp}(error(t_d)) \cdot \text{Exp}(-\text{Exp}(error(t_d))^T R_i J_i^i W_i \delta t)}_D \cdot \underbrace{\text{Exp}(R_j^j J_j^j W_j \delta t)}_E)$$

$$= \text{Log}(\text{Exp}(error(t_d)) \cdot \text{Exp}(D \delta t) \cdot \text{Exp}(E \delta t))$$

$$\approx \text{Log}(\text{Exp}(error(t_d)) \cdot (1 + D \delta t) (1 + E \delta t))$$

$$\approx \text{Log}(\text{Exp}(error(t_d)) \cdot (1 + (D+E) \delta t))$$

$$\approx \text{Log}(\text{Exp}(error(t_d)) \cdot \text{Exp}((D+E) \delta t))$$

$$\text{BCH近似: } \approx error(t_d) + J_r^T(error(t_d)) (D+E) \delta t$$

$$\therefore \frac{\partial error}{\partial \delta t} = J_r^T(error(t_d)) \cdot (D+E)$$

其中:

$$J_i^i = J_i(-W_i t_d)$$

$$J_j^j = J_j(W_j t_d)$$

$$D = -\text{Exp}(error(t_d))^T R_i J_i^i W_i \delta t$$

$$E = R_j^j J_j^j W_j \delta t$$

感谢各位聆听 !
Thanks for Listening

