Jindong_Shi
2022-02-13

第 17 节课习题

1. 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

① 推导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

我们考虑相机相邻两个时刻 i 和时刻 j 有两帧图像 I_i 和 I_j

其中 $\begin{bmatrix} u_{ci} \\ v_{ci} \end{bmatrix}$，$\begin{bmatrix} u_{cj} \\ v_{cj} \end{bmatrix}$ 是同一个特征点在该两帧图像上分别经归一化后，归一化相机坐标系上的点

定义 i 时刻特征点的逆深度为 $\lambda_i = \frac{1}{z_{ci}}$

则特征点在相机坐标系 i 下有：

$$\begin{bmatrix} x_{ci} \\ y_{ci} \\ z_{ci} \end{bmatrix} = \frac{1}{\lambda_i} \begin{bmatrix} u_{ci} \\ v_{ci} \\ 1 \end{bmatrix}$$

我们根据相机 i 时刻估计的该特征点的逆深度 λ_i 以及对应的观测到的归一化相机坐标系下的坐标 $\begin{bmatrix} u_{ci} \\ v_{ci} \end{bmatrix}$ 可以估计该特征点在相机 j 时刻坐标系下的三维坐标 $\begin{bmatrix} x_{cj} \\ y_{cj} \\ z_{cj} \end{bmatrix}$：

$$\begin{bmatrix} x_{cj} \\ y_{cj} \\ z_{cj} \\ 1 \end{bmatrix} = T_{bc}^{-1} T_{wb_j}^{-1} T_{wb_i} T_{bc} \begin{bmatrix} \frac{1}{\lambda} u_{ci} \\ \frac{1}{\lambda} v_{ci} \\ \frac{1}{\lambda} \\ 1 \end{bmatrix}$$

展开成三维坐标形式：

$$f_{cj} = \begin{bmatrix} x_{cj} \\ y_{cj} \\ z_{cj} \end{bmatrix} = R_{bc}^T R_{wb_j}^T R_{wb_i} R_{bc} \frac{1}{\lambda_i} \begin{bmatrix} u_{ci} \\ v_{ci} \\ 1 \end{bmatrix} + R_{bc}^T \left(R_{wb_j}^T \left(\left(R_{wb_i} p_{bc} + p_{wb_i} \right) - p_{wb_j} \right) - p_{bc} \right)$$

$\longrightarrow$ 假如不考虑时间戳延迟，那么重投影残差为：

$$r_{cj} = \begin{bmatrix} \frac{x_{cj}}{z_{cj}} - u_{cj} \\ \frac{y_{cj}}{z_{cj}} - v_{cj} \end{bmatrix}$$

$\longrightarrow$ 如果考虑时间戳延迟的话，需要对观测到的特征点在归一化相机坐标系 i 和 j 下的坐标分别进行延时补偿：

$$z_l^i(t_d) = \begin{bmatrix} u_{ci} \\ v_{ci} \end{bmatrix} + t_d V_l^i, \quad \text{其中 } V_l^i = \left(\begin{bmatrix} u_l^{i+1} \\ v_l^{i+1} \end{bmatrix} - \begin{bmatrix} u_l^i \\ v_l^i \end{bmatrix} \right) / (t_{i+1} - t_i)$$

$$z_l^j(t_d) = \begin{bmatrix} u_{cj} \\ v_{cj} \end{bmatrix} + t_d V_l^j, \quad \text{其中 } V_l^j = \left(\begin{bmatrix} u_l^{j+1} \\ v_l^{j+1} \end{bmatrix} - \begin{bmatrix} u_l^j \\ v_l^j \end{bmatrix} \right) / (t_{j+1} - t_j)$$

此时重投影残差 $r_{cj}(t_d)$ 可计算如下：

$$f_{cj}(t_d) = \begin{bmatrix} x_{cj}(t_d) \\ y_{cj}(t_d) \\ z_{cj}(t_d) \end{bmatrix} = R_{bc}^T R_{wb_j}^T R_{wb_i} R_{bc} \frac{1}{\lambda_i} \begin{bmatrix} z_l^i(t_d) \\ 1 \end{bmatrix} + R_{bc}^T \left(R_{wb_j}^T \left(R_{wb_i} p_{bc} + p_{wb_i} - p_{wb_j} \right) - p_{bc} \right)$$

$\Longrightarrow$ 归一化后的预测值，取前两维：$\hat{z}_l^j(t_d) = \begin{bmatrix} x_{cj}(t_d) / z_{cj}(t_d) \\ y_{cj}(t_d) / z_{cj}(t_d) \end{bmatrix}$

那么：$r_{cj}(t_d) = \hat{z}_l^j(t_d) - z_l^j(t_d)$

Jindong_Shi
2022-02-13

2. 阅读论文 [1]，总结基于 B 样条的时间戳估计算法流程，梳理论文公式

对于多传感器系统而言，各个传感器的时间戳同步一直是一个棘手的问题。为了实现测量数据的时间戳同步，我们可以使用基于硬件支持的同步，也可以使用基于软件支持的同步。对于系统中硬件或软件支持的同步都不允许的情况，该论文提出了一种基于 B 样条的在线估计时间戳延迟的算法，并将该算法在 VIO 系统中进行验证和测试。

A. 使用基函数 (Basic Functions) 估计时间延迟 (Time Offsets)

任意时变状态量可以被表达成有限个基函数的加权和。比如 D 维状态量$x(t)$可以写成:

$$\boldsymbol{\Phi}(t) := \begin{bmatrix} \phi_1(t) & \dots & \phi_B(t) \end{bmatrix}, \quad \mathbf{x}(t) := \boldsymbol{\Phi}(t)\mathbf{c},$$

其中每一个$\phi_b(t)$是 Dx1 维时间函数，$\boldsymbol{\Phi}(t)$是 DxB 维累积基矩阵(stacked basis matrix)，这样一来，估计 D 维状态量$x(t)$就变成了估计 Bx1 维系数向量c。

当测量数据y_j的时间戳为t_j，而估计的状态量$x(t_j)$存在延时 d 时，误差项可写为:

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

其中 $\mathbf{h}(\cdot)$ 为测量模型，用于将状态量映射为测量值。我们可以使用一般的优化算法来优化这个非线性误差项得到延时d。误差项可近似表示成一阶泰勒展开的形式:

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

其中

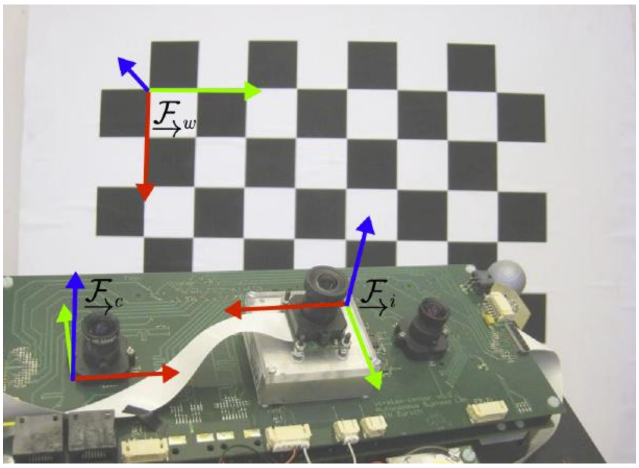
$$\mathbf{H} := \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\mathbf{x}\left(\boldsymbol{\Phi}(t_j+\bar{d})\mathbf{c}\right)}$$

时间导数$\dot{\boldsymbol{\Phi}}(t)$是可以解析求得的。

将估计问题转化为这种基函数的形式有利于我们后续使用 ML 算法，并且雅可比也可以解析求得的。

B. 应用于 Camera/IMU 的标定

Camera/IMU 的标定主要是为了标定相机和 IMU 之间的相对旋转和平移，以及两个传感器之间的延时。世界坐标系$\underrightarrow{\mathcal{F}}_w$，相机坐标系$\underrightarrow{\mathcal{F}}_c$，IMU 坐标系$\underrightarrow{\mathcal{F}}_i$的设定如下图所示:



我们主要估计的时不变参数有重力$\mathbf{g}_w$，相机坐标系和 IMU 坐标系之间的变换矩阵$\mathbf{T}_{c,i}$，相机和 IMU 之间的延时d。估计的时变参数有 IMU 的实时位姿$\mathbf{T}_{w,i}(t)$，加速度计的 bias $(\mathbf{b}_a(t))$，以及角速度计的 bias $(\mathbf{b}_\omega(t))$。

正如 A 部分中提到的，时变状态量可以表示成 B 样条函数的形式。其中 IMU 的位姿$\mathbf{T}_{w,i}(t)$可以参数化为 6x1 的样条(spline)，表示 3 自由度旋转和 3 自由度平移:

Jindong_Shi
2022-02-13

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

其中$\boldsymbol{\varphi}(t) := \boldsymbol{\Phi}_{\varphi}(t)\mathbf{c}_{\varphi}$是代表旋转的参数，$\mathbf{C}(\cdot)$将其转换为旋转矩阵。$\mathbf{t}(t) := \boldsymbol{\Phi}_t(t)\mathbf{c}_t$表示平移。

世界坐标系下的速度$\mathbf{v}(t)$和加速度$\mathbf{a}(t)$可以由下式计算得到:

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

对于给定的旋转参数$\boldsymbol{\varphi}(t)$，角速度可表达为:

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

$\mathbf{S}(\cdot)$是与角速度相关的参数化矩阵，比如使用角轴进行表示

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

为了标定相机和 IMU，这里收集了$T = [t_1, t_K]$大概 1-2 分钟的测量数据。这里使用一般的离散时间的 IMU 和相机测量公式:

$$\begin{aligned}
\boldsymbol{\alpha}_k &:= \mathbf{C}\left(\boldsymbol{\varphi}(t_k)\right)^T \left(\mathbf{a}(t_k) - \mathbf{g}_w\right) + \mathbf{b}_a(t_k) + \mathbf{n}_{a_k}\,, \\
\boldsymbol{\varpi}_k &:= \mathbf{C}\left(\boldsymbol{\varphi}(t_k)\right)^T \boldsymbol{\omega}(t_k) + \mathbf{b}_{\omega}(t_k) + \mathbf{n}_{\omega_k}\,, \\
\mathbf{y}_{mj} &:= \mathbf{h}\left(\mathbf{T}_{c,i}\mathbf{T}_{w,i}(t_j + d)^{-1}\mathbf{p}_w^m\right) + \mathbf{n}_{y_{mj}}\,,
\end{aligned}$$

其中t_k, where $k = 1 \dots K$表示 IMU 采样时间，$\boldsymbol{\alpha}_k$为加速度计测量值，$\boldsymbol{\varpi}_k$为角速度计测量值。

t_j and $j = 1 \dots J$ 为相机采样时间，d为相机和 IMU 之间的延时，可正可负。$\{\mathbf{p}_w^m | m = 1 \dots M\}$为观测到特征点的世界坐标系下的坐标，$\mathbf{y}_{mj}$为第$m$个路标点在第$j$帧图像内的观测。

$\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$ 表示为各测量值的高斯白噪声，它们相互之间独立。$\mathbf{h}(\cdot)$为相机非线性模型。

我们使用维纳过程来建模 bias 随机游走:

$$\begin{aligned}
\dot{\mathbf{b}}_a(t) &= \mathbf{w}_a(t) \qquad \mathbf{w}_a(t) \sim \mathcal{GP}\left(\mathbf{0}, \mathbf{Q}_a\delta(t - t')\right) \\
\dot{\mathbf{b}}_{\omega}(t) &= \mathbf{w}_{\omega}(t) \qquad \mathbf{w}_{\omega}(t) \sim \mathcal{GP}\left(\mathbf{0}, \mathbf{Q}_{\omega}\delta(t - t')\right)
\end{aligned}$$

至此我们可以定义出五个量(视觉测量，加速度，角速度，加速度 bias，角速度 bias)的估计误差项，以及它们的损失函数:

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

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

$$\mathbf{e}_{\alpha_k} := \boldsymbol{\alpha}_k - \mathbf{C}\left(\boldsymbol{\varphi}(t_k)\right)^T\left(\mathbf{a}(t_k) - \mathbf{g}_w\right) + \mathbf{b}_a(t_k)$$

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

$$\mathbf{e}_{\omega_k} := \boldsymbol{\varpi}_k - \mathbf{C}\left(\boldsymbol{\varphi}(t_k)\right)^T\boldsymbol{\omega}(t_k) + \mathbf{b}_\omega(t_k)$$

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

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

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

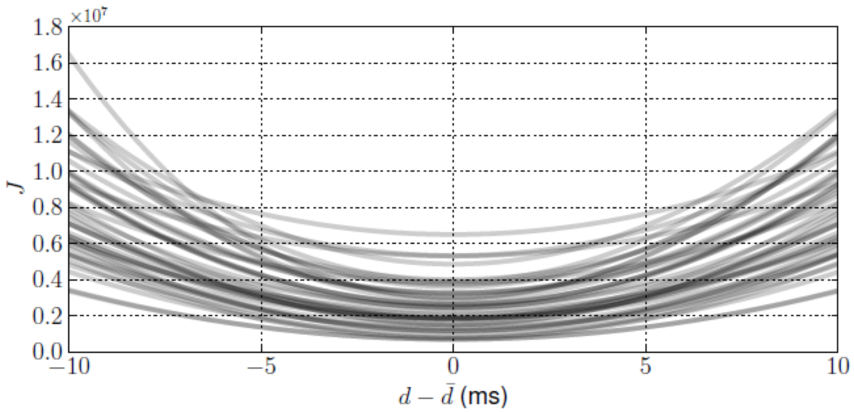
$$\mathbf{e}_{b_\omega}(t) := \dot{\mathbf{b}}_\omega(t)$$

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

LM 算法通过最小化所有损失函数和$J := J_y + J_\alpha + J_\omega + J_{b_a} + J_{b_\omega}$来一次性估计所有带待估计参数。

C. 实验结果

将算法部署到 VIO 系统中，使用 LM 算法来离线估计出采集到的 Camera 和 IMU 的数据之间的延时情况。论文测试了不同延时大小情况下，损失函数随着延时变化的情况。通过下图，论文作者发现损失函数表现为凸，最小值一直在$\bar{d}$附近，可以使用该算法有效地估计出相机和 IMU 之间的延时。



3.推导初始化时旋转误差 Eq.(17) 对时间戳延迟t_d的雅可比，参考论文 [2] 附录 D

首先根据课堂上给出的公式[17]:

初始化阶段，相机关键帧的姿态通过 VO/SFM 等可求解，有

$$\mathbf{R}_{wb_i} = \mathbf{R}_{wc_i,t_d}\mathbf{R}_{cb}, \quad \mathbf{R}_{wb_j} = \mathbf{R}_{wc_j,t_d}\mathbf{R}_{cb} \tag{15}$$

跟 IMU 预积分旋转分量构建误差，如下（可回顾 vio 初始化章节）

$$\mathbf{e}_R = \log\left(\Delta\overline{\mathbf{R}}_{i,j}\mathbf{R}_{wb_i}^\top\mathbf{R}_{wb_j}\right) \tag{16}$$

考虑陀螺仪 bias，时间延迟 t_d 有：

$$\begin{aligned}\mathbf{e}_{rot_{i,j}} = \log\Bigg(&\left(\Delta\overline{\mathbf{R}}_{i,j}\operatorname{Exp}\left(\mathbf{J}^g_{\Delta\overline{\mathbf{R}}}\delta\mathbf{b}_g\right)\right)^\top\mathbf{R}_{bc} \\ &\cdot\operatorname{Exp}\left(-\omega_{c_i}t_d\right)\mathbf{R}_{c_iw}\mathbf{R}_{wc_{i+1}}\operatorname{Exp}\left(\omega_{c_{i+1}}t_d\right)\mathbf{R}_{cb}\Bigg)\end{aligned} \tag{17}$$

论文[2]中给出了该误差项对时间戳延迟t_d的雅可比的推导过程:

Letting $\mathbf{R}_1'' = \left(\Delta\bar{\mathbf{R}}_{ij}\operatorname{Exp}\left(\mathbf{J}^g_{\Delta\bar{\mathbf{R}}}\delta\mathbf{b}_g\right)\right)^T\mathbf{R}_c^b$, $\mathbf{R}_2'' = \mathbf{R}_w^{c_i}\mathbf{R}_{c_j}^w$, and $\mathbf{R}_3'' = \mathbf{R}_b^c$, we have:

$$\begin{aligned}
&\mathbf{e}_{rot}(t_d+\delta t_d) \\
&= \operatorname{Log}(\mathbf{R}_1''\operatorname{Exp}(-\omega_{c_i}(t_d+\delta t_d))\mathbf{R}_2''\operatorname{Exp}(\omega_{c_j}(t_d+\delta t_d))\mathbf{R}_3'') \\
&\overset{(38)}{\approx} \operatorname{Log}\left(\mathbf{R}_1''\operatorname{Exp}(-\mathbf{J}_l^i\omega_{c_i}\delta t_d)\operatorname{Exp}(-\omega_{c_i}t_d)\mathbf{R}_2''\right. \\
&\qquad \left.\cdot\operatorname{Exp}(\omega_{c_j}t_d)\operatorname{Exp}(\mathbf{J}_r^j\omega_{c_j}\delta t_d)\mathbf{R}_3''\right) \\
&\overset{(36)}{=} \operatorname{Log}\left(\operatorname{Exp}(-\mathbf{R}_1''\mathbf{J}_l^i\omega_{c_i}\delta t_d)\mathbf{R}_1''\operatorname{Exp}(-\omega_{c_i}t_d)\mathbf{R}_2''\right. \\
&\qquad \left.\cdot\operatorname{Exp}(\omega_{c_j}t_d)\mathbf{R}_3''\operatorname{Exp}(\mathbf{R}_3''^T\mathbf{J}_r^j\omega_{c_j}\delta t_d)\right) \\
&= \operatorname{Log}\left(\operatorname{Exp}(-\mathbf{R}_1''\mathbf{J}_l^i\omega_{c_i}\delta t_d)\operatorname{Exp}(\mathbf{e}_{rot}(t_d))\right. \\
&\qquad \left.\cdot\operatorname{Exp}(\mathbf{R}_3''^T\mathbf{J}_r^j\omega_{c_j}\delta t_d)\right) \\
&\overset{(36)}{=} \operatorname{Log}(\operatorname{Exp}(\mathbf{e}_{rot}(t_d))\operatorname{Exp}(-\operatorname{Exp}(\mathbf{e}_{rot}(t_d))^T\mathbf{R}_1''\mathbf{J}_l^i\omega_{c_i}\delta t_d) \\
&\qquad \cdot\operatorname{Exp}(\mathbf{R}_3''^T\mathbf{J}_r^j\omega_{c_j}\delta t_d)) \\
&= \operatorname{Log}\left(\operatorname{Exp}(\mathbf{e}_{rot}(t_d))\operatorname{Exp}(D\cdot\delta t_d)\operatorname{Exp}(E\cdot\delta t_d)\right) \\
&\overset{(35)}{\approx} \operatorname{Log}\left(\operatorname{Exp}(\mathbf{e}_{rot}(t_d))(\mathbf{I}+(D+E)^\wedge\delta t_d)\right) \\
&\overset{(35)}{\approx} \operatorname{Log}\left(\operatorname{Exp}(\mathbf{e}_{rot}(t_d))\operatorname{Exp}((D+E)\delta t_d)\right) \\
&\overset{(37)}{\approx} \mathbf{e}_{rot}(t_d) + \mathbf{J}_r^{-1}(\mathbf{e}_{rot}(t_d))((D+E)\delta t_d),
\end{aligned} \tag{49}$$

with $\mathbf{J}_l^i \doteq \mathbf{J}_l(-\omega_{c_i}t_d)$, $\mathbf{J}_r^j \doteq \mathbf{J}_r(\omega_{c_j}t_d)$, $D \doteq -\operatorname{Exp}(\mathbf{e}_{rot}(t_d))^T\mathbf{R}_1''\mathbf{J}_l^i\omega_{c_i}$, and $E \doteq \mathbf{R}_3''^T\mathbf{J}_r^j\omega_{c_j}$. Summariz-

$$\Rightarrow \frac{\partial\mathbf{e}_{rot}}{\partial\delta t_d} = \mathbf{J}_r^{-1}(\mathbf{e}_{rot}(t_d))(D+E).$$

参考文献:

[1] Paul Furgale, Joern Rehder, and Roland Siegwart. “Unified temporal and spatial calibration for multi-sensor systems”. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE. 2013, pp. 1280–1286.

[2] Weibo Huang, Hong Liu, and Weiwei Wan. “Online initialization and extrinsic spatial-temporal calibration for monocular visual-inertial odometry”. In: arXiv preprint arXiv:2004.05534 (2020)