这是标题

陈烁龙

2022 年 9 月 14 日

目录

1 概述 **1**

2 数学推导 **1**

2.1 系统状态向量和状态误差向量 . . . 1

2.2 系统的过程模型 1

2.3 测量模型 2

2.4 滤波更新的技巧 3

插图

表格

1 概述

论文名字：《Robust Stereo Visual Inertial Odometry for Fast Autonomous Flight》，提出了一种基于滤波器的双目视觉加惯导的里程计算法。由于其主要应用于小型的飞行机器上，所以算法的实时性和精度是比较需要考量的地方。小型飞行器的挂载能力通常有限，不能使用比较大的传感器，所以视觉相机和 IMU 就成为了 SLAM 的首选传感器。

另外，滤波的方式虽然略逊色于基于优化的方式，但是其计算量小，是比较合适的。至于选择使用双目相机，是因为其相比与单目相机，适应场景更强，当然也会带来额外的计算量。

该论文使用了紧耦合的方式组织惯导系统和视觉系统。

2 数学推导

2.1 系统状态向量和状态误差向量

首先，定义状态向量：

$$\boldsymbol{x}_I = \begin{pmatrix} {}^I_G\boldsymbol{q}^T & \boldsymbol{b}_g^T & {}^G\boldsymbol{v}_I^T & \boldsymbol{b}_a^T & {}^G\boldsymbol{p}_I^T & {}^I_C\boldsymbol{q}^T & {}^I\boldsymbol{p}_C^T \end{pmatrix}^T$$

其中：${}^I_G\boldsymbol{q}^T$ 表示从 $\boldsymbol{G}$ 系 (全局坐标系，论文里是惯性坐标系) 到 $\boldsymbol{I}$ 系 (IMU 的坐标系，由于载体坐标系和 IMU 的坐标系重合，所以就是载体坐标系) 的姿态变换四元素；$\boldsymbol{b}_g^T$ 表示陀螺的零偏；${}^G\boldsymbol{v}_I^T$ 表示载体坐标系想对于惯性坐标系的速度向量；$\boldsymbol{b}_a^T$ 表示加速度计的零偏；${}^G\boldsymbol{p}_I^T$ 表示载体坐标系想对于惯性坐标系的位置向量；${}^I_C\boldsymbol{q}^T$ 表示相机坐标系想对于载体坐标系的姿态四元素；${}^I\boldsymbol{p}_C^T$ 表示相机坐标系想对于载体坐标系的位置向量。

而后，定义状态的误差向量：

$$\tilde{\boldsymbol{x}}_I = \begin{pmatrix} {}^I_G\tilde{\boldsymbol{\theta}}^T & \tilde{\boldsymbol{b}}_g^T & {}^G\tilde{\boldsymbol{v}}_I^T & \tilde{\boldsymbol{b}}_a^T & {}^G\tilde{\boldsymbol{p}}_I^T & {}^I_C\tilde{\boldsymbol{\theta}}^T & {}^I\tilde{\boldsymbol{p}}_C^T \end{pmatrix}^T$$

注意，这里使用轴角来替换四元素表示姿态，由一个四维向量变成了三维向量。这里的误差是这么定义的：

1. 姿态：

$$\tilde{\boldsymbol{q}} = \delta\boldsymbol{q} = \boldsymbol{q} \otimes \hat{\boldsymbol{q}}^{-1}$$

$$\tilde{\boldsymbol{q}} = \delta\boldsymbol{q} \approx \begin{pmatrix} \frac{1}{2}{}^G_I\tilde{\boldsymbol{\theta}}^T & 1 \end{pmatrix}^T$$

这是因为对于轴角 $\boldsymbol{\theta}$ 而言，其构成的四元素为：

$$\boldsymbol{q} = \begin{pmatrix} \frac{\sin\|\frac{1}{2}\boldsymbol{\theta}\|}{\|\frac{1}{2}\boldsymbol{\theta}\|}\frac{1}{2}\boldsymbol{\theta} \\ \cos\|\frac{1}{2}\boldsymbol{\theta}\| \end{pmatrix}$$

当向量 $\boldsymbol{\theta}$ 是小角度时，可以得到上文中的近似 (类似于取极限)。

2. 其他状态向量

对于其他状态向量，其误差向量可以表示为 (以 ${}^G\boldsymbol{p}_I^T$ 为例)：

$${}^G\tilde{\boldsymbol{p}}_I^T = {}^G\boldsymbol{p}_I^T - {}^G\hat{\boldsymbol{p}}_I^T$$

上文描述的就是 IMU 的状态向量，而相机的状态误差向量则为：

$$\tilde{\boldsymbol{x}}_{C_i} = \begin{pmatrix} {}^{C_i}_G\tilde{\boldsymbol{\theta}}^T & {}^G\tilde{\boldsymbol{p}}_{C_i}^T \end{pmatrix}^T$$

注意，相机的位姿在同一个状态解算中有多个。综上，整个系统的状态误差向量为：

$$\tilde{\boldsymbol{x}} = \begin{pmatrix} \tilde{\boldsymbol{x}}_I^T & \tilde{\boldsymbol{x}}_{C_1}^T & \tilde{\boldsymbol{x}}_{C_2}^T & \dots & \tilde{\boldsymbol{x}}_{C_N}^T \end{pmatrix}^T$$

2.2 系统的过程模型

该部分主要的目的是得到在 EKF 中预测部分的状态转移方程和误差模型，因此，在预测步骤中，我们主要基于的是 IMU 的状态。首先是 IMU 状态的微分方程，其描述了 IMU 状态随时间变化的规律：

$$\begin{cases} {}^I_G\dot{\hat{\boldsymbol{q}}} = \frac{1}{2}\Omega(\hat{\boldsymbol{\omega}}){}^I_G\hat{\boldsymbol{q}} \\ \dot{\hat{\boldsymbol{b}}}_g = \boldsymbol{0}_{3\times1} \\ {}^G\dot{\hat{\boldsymbol{v}}} = C({}^I_G\hat{\boldsymbol{q}})^T\hat{\boldsymbol{a}} + {}^G\boldsymbol{g} \\ \dot{\hat{\boldsymbol{b}}}_a = \boldsymbol{0}_{3\times1} \\ {}^G\dot{\hat{\boldsymbol{p}}}_I = {}^G\hat{\boldsymbol{v}} \\ {}^I_C\dot{\hat{\boldsymbol{q}}} = \boldsymbol{0}_{3\times1} \\ {}^I\dot{\hat{\boldsymbol{p}}}_C = \boldsymbol{0}_{3\times1} \end{cases}$$

其中：$C(\star)$ 表示将四元素表示的姿态变换成旋转矩阵。另外：

$$\begin{cases} \hat{\boldsymbol{\omega}} = \boldsymbol{\omega}_m - \hat{\boldsymbol{b}}_g \\ \hat{\boldsymbol{a}} = \boldsymbol{a}_m - \hat{\boldsymbol{b}}_a \\ \Omega(\hat{\boldsymbol{\omega}}) = \begin{pmatrix} -\lfloor \hat{\boldsymbol{\omega}} \rfloor_\times & \hat{\boldsymbol{\omega}} \\ -\hat{\boldsymbol{\omega}}^T & 0 \end{pmatrix} \end{cases}$$

上两个式子表示将 IMU 器件的偏差从原始测量值中移除。于是，我们可以得到如下的微分方程：

$$\begin{cases} \dot{\tilde{\boldsymbol{x}}}_I = \boldsymbol{F}\tilde{\boldsymbol{x}}_I + \boldsymbol{G}\boldsymbol{n}_I \\ \boldsymbol{n}_I = \begin{pmatrix} \boldsymbol{n}_g^T & \boldsymbol{n}_{\omega g}^T & \boldsymbol{n}_a^T & \boldsymbol{n}_{\omega a}^T \end{pmatrix}^T \end{cases}$$

其中：$\boldsymbol{n}_g$ 表示陀螺的高斯噪声，$\boldsymbol{n}_a$ 表示加速度计的高斯噪声，$\boldsymbol{n}_{wg}$ 表示陀螺的随机游走速率，$\boldsymbol{n}_{wa}$ 表示加速度计的随机游走速率。具体系数矩阵 $\boldsymbol{F}$ 的和 $\boldsymbol{G}$ 形式请查看论文附录 A。比如，$\boldsymbol{F}$ 的第三行第一列的矩阵块为：$-C({}^I_G\hat{\boldsymbol{q}})^T \lfloor \hat{\boldsymbol{a}} \rfloor_\times$，推导：

$$\begin{aligned} {}^G\dot{\boldsymbol{v}} - {}^G\dot{\hat{\boldsymbol{v}}} &= C({}^I_G\hat{\boldsymbol{q}})^T(\boldsymbol{a}_m - (\boldsymbol{b}_a - \tilde{\boldsymbol{b}}_a)) + {}^G\boldsymbol{g} \\ \rightarrow {}^G\dot{\tilde{\boldsymbol{v}}} &= -C({}^I_G\hat{\boldsymbol{q}})^T(\boldsymbol{a}_m + \tilde{\boldsymbol{b}}_a - \boldsymbol{b}_a) - {}^G\boldsymbol{g} \\ \rightarrow \frac{\partial {}^G\dot{\tilde{\boldsymbol{v}}}}{\partial {}^I_G\tilde{\boldsymbol{\theta}}} &= -\boldsymbol{I} \cdot C({}^I_G\hat{\boldsymbol{q}})^T \left\lfloor \boldsymbol{a}_m + \tilde{\boldsymbol{b}}_a - \boldsymbol{b}_a \right\rfloor_\times \\ &= -C({}^I_G\hat{\boldsymbol{q}})^T \lfloor \hat{\boldsymbol{a}} \rfloor_\times \\ \rightarrow \frac{\partial {}^G\dot{\tilde{\boldsymbol{v}}}}{\partial \tilde{\boldsymbol{b}}_a} &= -C({}^I_G\hat{\boldsymbol{q}})^T \end{aligned}$$

这样，也得到了 $\boldsymbol{F}$ 的第三行第四列的矩阵块 (求导的地方使用了李代数左乘扰动模型，即如何处理包含姿态的矩阵对姿态的求导)。

得到连续的状态微分方程之后，我们需要对其进行离散化，才能在实际工程上应用。也就是我们要求解上面的微分方程。利用四阶 Runge-Kutta 方法，可以得到[1]：

$$\begin{cases} \boldsymbol{\Phi}_k = \boldsymbol{\Phi}(t_{k+1}, t_k) = e^{\int_{t_k}^{t_{k+1}} F(\tau)d\tau} \\ \boldsymbol{Q}_k = \int_{t_k}^{t_{k+2}} \boldsymbol{\Phi}(t_{k+1}, \tau)\boldsymbol{G}\boldsymbol{Q}\boldsymbol{G}^T\boldsymbol{\Phi}(t_{k+1}, \tau)^T d\tau \end{cases}$$

[1]推导过程可以参考吴云老师的《最优估计》教材第三章。

其中 $\boldsymbol{\Phi}(t_{k+1}, t_k)$ 即为从时刻 t_k 到时刻 t_{k+1} 的状态转移矩阵。而 $\boldsymbol{Q}_k$ 即为系统的噪声协方差矩阵。根据卡尔曼滤波器的状态预测步骤，我们对上一时刻的状态进行预测，得到当前时刻的状态和其协方差矩阵：

$$\begin{cases} \tilde{\boldsymbol{x}}_{I_{k+1|k}} = \boldsymbol{\Phi}(t_{k+1}, t_k)\tilde{\boldsymbol{x}}_{I_k} \\ \boldsymbol{P}_{II_{k+1|k}} = \boldsymbol{\Phi}(t_{k+1}, t_k)\boldsymbol{P}_{II_{k|k}}\boldsymbol{\Phi}(t_{k+1}, t_k)^T + \boldsymbol{Q}_k \end{cases}$$

最后，在预测部分，我们的协方差矩阵为 (注意，相机状态与 IMU 状态之间的协方差矩阵块在预测时，需要作用一个系数矩阵)：

$$\boldsymbol{P}_{k+1|k} = \begin{pmatrix} \boldsymbol{P}_{II_{k+1|k}} & \boldsymbol{\Phi}_k\boldsymbol{P}_{IC_{k+1|k}} \\ \boldsymbol{P}_{IC_{k+1|k}}^T\boldsymbol{\Phi}_k^T & \boldsymbol{P}_{CC_{k+1|k}} \end{pmatrix}$$

当新的相机位姿到来时，我们需要纳入进来。状态向量部分的增加如前面所述，而协方差矩阵则为：

$$\boldsymbol{P}_{k|k} = \begin{pmatrix} \boldsymbol{I}_{21+6N} \\ \boldsymbol{J} \end{pmatrix} \boldsymbol{P}_{k|k} \begin{pmatrix} \boldsymbol{I}_{21+6N} \\ \boldsymbol{J} \end{pmatrix}^T$$

其中，J 矩阵可以通过误差传播得到 (关心的是新相机位姿和 IMU 位姿之间的关系)：

$$\begin{cases} {}^C_G\hat{\boldsymbol{q}} = {}^C_I\hat{\boldsymbol{q}} \otimes {}^I_G\hat{\boldsymbol{q}} \\ {}^G\hat{\boldsymbol{p}}_C = {}^G\hat{\boldsymbol{p}}_I + C({}^I_G\hat{\boldsymbol{q}}^T){}^I\hat{\boldsymbol{p}}_C \\ \dfrac{\partial {}^C_G\hat{\boldsymbol{q}}}{\partial \delta {}^I_G\boldsymbol{\theta}} = C^T({}^I_G\hat{\boldsymbol{q}}) \\ \dfrac{\partial {}^G\hat{\boldsymbol{p}}_C}{\partial \delta {}^I_G\boldsymbol{\theta}} = C({}^I_G\hat{\boldsymbol{q}}^T) \lfloor {}^I\hat{\boldsymbol{p}}_C \rfloor_\times = C^T({}^I_G\hat{\boldsymbol{q}}) \lfloor {}^I\hat{\boldsymbol{p}}_C \rfloor_\times \\ \dfrac{\partial {}^G\hat{\boldsymbol{p}}_C}{\partial {}^G\hat{\boldsymbol{p}}_I} = \boldsymbol{I}_{3\times 3} \end{cases}$$

2.3 测量模型

测量模型参考李圣雨师兄的论文对应的目录文件 (*lishengyu*)。注意里面用到了一个技巧，在构建测量模型是，通过在方程上左乘特征点的左零空间，将特征点边缘化，以加速求解。

2.4 滤波更新的技巧

在 EKF 中，每次得到新 IMU 数据，都会进行状态预测，但是测量更新并不是一直在做。有两种条件触发测量更新：

1. 当跟踪的特征丢失的时候，进行测量更新。这时该特征点对应的相机帧不再增加 (因为已经跟踪丢失了，不像 ORB-SLAM 一样进行重定位操作)，即使的进行测量更新可以修正窗口内的系统状态。

2. 当窗口的大小到达限制上限。设定一个窗口大小，当实际的窗口大小到达限制大小时，意味着要丢弃部分相机帧，所以即使进行测量更新。

至于如何丢弃相机帧，论文的做法是每隔一个更新步骤就移除两个相机帧。具体而言，基于窗口内倒数第二帧和倒数第二帧之间的运动，要么最老的一个相机帧被移除，要么倒数第二帧被移除。重复这样做两次，就可以移除掉两个帧。由于倒数第一帧 (最新的一帧) 保持着提取的特征点，所以不考虑移除。

这么做是有现实背景的，当载体高速运动是，倒数第二帧和倒数第二帧之间的运动会很大，那么移除掉最老的一个相机帧；如果倒数第二帧和倒数第二帧之间的运动很小 (比如飞行器载体悬停)，那么移除掉倒数第二帧。