Classical Visual-Inertial Odometry RBE549 Project 4 Phase 1

Sumukh Porwal, Piyush Thapar, Sarah Semy
MS Robotics Engineering
Worcester Polytechnic Institute
Email: sporwal@wpi.edu, pthapar@wpi.edu, srsemy@wpi.edu

***Abstract*—In this work, we develop a stereo visual-inertial odometry system based on the Multi-State Constraint Kalman Filter (MSCKF) [1] [2], employing a tightly integrated sensor fusion framework for VIO. The system combines data from a stereo camera setup and an Inertial Measurement Unit (IMU) to estimate motion and localization. As part of the implementation, we focus on developing and integrating seven key functions central to the MSCKF algorithm.**

I. Introduction

Estimating depth using a single camera has long remained a difficult problem within computer vision. To overcome this, researchers have traditionally employed stereo camera setups, utilizing the spatial offset between two viewpoints to infer depth through feature matching across image pairs. However, this technique is predominantly effective only for static scenes and tends to fail in dynamic environments or in the presence of motion blur — challenges frequently encountered in robotics applications. Although stereo vision offers valuable three-dimensional insights, its performance degrades significantly under motion-heavy conditions, underscoring the need for more resilient alternatives.

In this project, we explore sensor fusion between a monocular camera and an Inertial Measurement Unit (IMU) to estimate the robot's trajectory and position, an approach known as Visual-Inertial Odometry (VIO). IMUs are highly responsive to rapid movements and sudden changes in acceleration, compensating for situations where the camera alone may struggle. However, IMUs are prone to drift over time without external correction. Conversely, cameras provide accurate spatial localization but are sensitive to fast motions and environmental noise. To address these complementary weaknesses, we implement a filter-based stereo VIO framework using the Multi-State Constraint Kalman Filter (MSCKF) algorithm. We plan to validate our system's performance through experiments conducted on the Machine Hall 01 easy sequence from the EuRoC dataset.

II. Dataset

For this project, we employed the Machine Hall 01 easy sequence from the EuRoC dataset [3]. Collected using a Micro Aerial Vehicle (MAV), the dataset provides stereo imagery synchronized with Inertial Measurement Unit (IMU) readings. It also includes highly accurate ground truth data for both motion and structure, captured using a Vicon Motion Capture system capable of sub-millimeter precision.

III. Initialize Gravity and Bias

To initialize gravity and bias, we utilize the first 300 messages received from the IMU while the robot is in a static state. Each message contains accelerometer and gyroscope measurements, which are used to compute the gravity vector $\mathbf{g} = [0, 0, g_{\text{norm}}]$, where g_{norm} is the norm of the accelerometer readings. The gyroscope bias $\mathbf{b}_g$ is initialized as the average of the gyroscope measurements.

IV. Batch IMU Processing

The IMU message buffer stores a sequence of measurements. For every message received before the feature timestamp, the process model is applied to propagate and update the state $\mathbf{x}$ accordingly.

V. Process Model

The continuous dynamics of the estimated IMU state is:

$$\begin{aligned}
&{}^I_G\hat{q} = \frac{1}{2}\Omega(\hat{\omega})\,{}^I_G\hat{q}, \quad \hat{b}_g = \mathbf{0}_{3\times1}, \\
&{}^G\hat{v} = C({}^I_G\hat{q})\hat{a} + {}^G g, \\
&\dot{\hat{b}}_a = \mathbf{0}_{3\times1}, \quad {}^G\hat{p}_I = {}^G\hat{v}, \\
&\dot{\hat{b}}_a = \mathbf{0}_{3\times1}, \quad {}^I\hat{p}_C = \mathbf{0}_{3\times3}
\end{aligned} \tag{1}$$

$$\Omega(\hat{\omega}) = \begin{bmatrix} -[\omega_\times] & \omega \\ -\omega^\top & 0 \end{bmatrix} \tag{2}$$

$$[\omega_\times] = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{3}$$

$$\omega_m = \omega + b_g + n_g \tag{4}$$

$$a_m = {}^I_G R \left({}^G a - {}^G g + 2[\omega_{G\times}]^G v_I + 2[\omega_{G\times}]^{2G} p_I \right) + b_a + n_a \tag{5}$$

The linearized continuous dynamics for the error IMU state is:

$$\dot{\tilde{x}} = F\tilde{x}_I + Gn_I \tag{6}$$

F and G in the above equation can be defined as:

$$F = \begin{pmatrix} -[\hat{\omega}_\times] & -I_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ -C({}^I_G\hat{q})[\hat{a}_\times] & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -C({}^I_G\hat{q})[\hat{a}_\times] & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & I_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{pmatrix}$$

$$G = \begin{pmatrix} -I_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & I_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -C({}^I_G\hat{q}) & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & I_3 \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{pmatrix}$$

VI. Predict New State

For each incoming IMU measurement, we apply fourth-order Runge-Kutta numerical integration to propagate the system state to the next time step, following the procedure outlined in the provided sample code. The integration involves computing the intermediate terms k_1, k_2, k_3, and k_4, derived from the time derivatives of orientation and acceleration. These intermediate values are then used to predict the updated system state, which includes orientation, velocity, and position.

VII. State Augmentation

State augmentation includes updating the camera pose, defined as $({}^G_{C_i}q,\ {}^Gp_{C_i})$, where ${}^G_{C_i}q$ is the orientation and ${}^Gp_{C_i}$ is the position. This pose is calculated as:

$$ {}^Gp_C = {}^Gp_I + C({}^{C_i}_G q)^\top {}^Ip_C \tag{7}$$

$$ {}^{C_i}_G q = {}^C_G q \otimes {}^C_G q \tag{8}$$

$$ J = \begin{bmatrix} J_1 & O_{6\times6N} \end{bmatrix} \tag{9}$$

$$ J = \begin{bmatrix} C({}^I_C q) & \mathbf{0}_{3\times9} & \mathbf{0}_{3\times3} & I_3 & \mathbf{0}_{3\times3} \\ \left[C({}^I_G q)^\top {}^Ip_C\right]_\times & \mathbf{0}_{3\times9} & I_3 & \mathbf{0}_{3\times3} & I_3 \end{bmatrix} \tag{10}$$

$$ P_{k|k} = \begin{bmatrix} I_{21+6N} \\ J \end{bmatrix} P_{K|K} \begin{bmatrix} I_{21+6N} \\ J \end{bmatrix}^\top \tag{11}$$

VIII. Adding Feature Observation

This step/function updates the latest feature detected to the total feature map if it does not exist. Feature map addition is done with the help of feature ID and current state ID keys.

$$ Z_i^j = \begin{bmatrix} u^j_{i,1} & v^j_{i,1} & u^j_{i,2} & v^j_{i,2} \end{bmatrix}^T \tag{12}$$

IX. Measurement Update

A single feature f_i, as observed by the stereo cameras with pose $({}^{C_i}_G q,\ {}^Gp_{C_i})$, can be represented as $({}^{C_i,1}_G q,\ {}^Gp_{C_i,1})$ and $({}^{C_i,2}_G q,\ {}^Gp_{C_i,2})$ for the left and right cameras respectively. The measurement matrix H is represented as:

$$ H = [QQ_2] \begin{bmatrix} T_H \\ O \end{bmatrix} \tag{13}$$

The residual r_n is computed as:

$$ r_n = Q^T r = T_H \tilde{X} + n_n \tag{14}$$

where r_n is the residual.

$$ R_n = \sigma^2_{im} I_{q\times q} \tag{15}$$

The Kalman gain K can be computed as:

$$ K = P T_H^T (T_H P T_H^T + R_n)^{-1} \tag{16}$$

$$ S K^T = T_H P \tag{17}$$

$$ \Delta X = K r_n \tag{18}$$

$$ P_{k+1|k+1} = (I_{k\times k} - K T_H)\, P_{k|k} \tag{19}$$

X. Results

The estimated trajectory attains an RMSE Absolute Trajectory Error of 0.10636397690854665 relative to the Vicon reference in translation, 127.46498255468062 in rotation and 1.707416155297823 in scale.

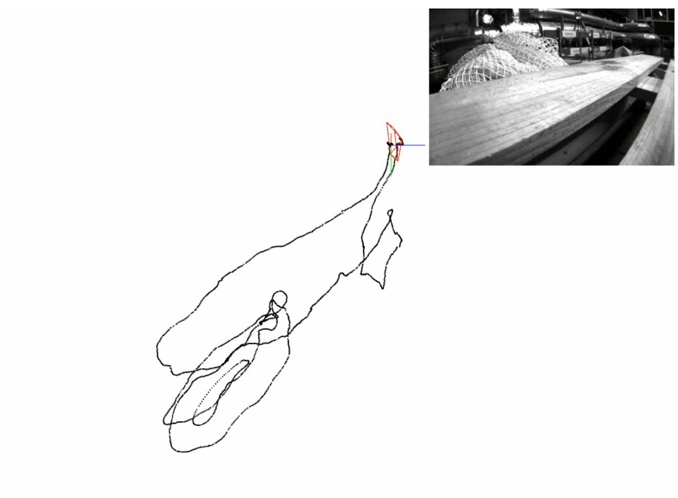

Fig. 1. Final Output from Viewer

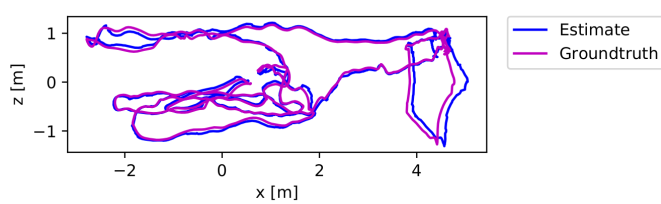

Fig. 2. Trajectory Side View

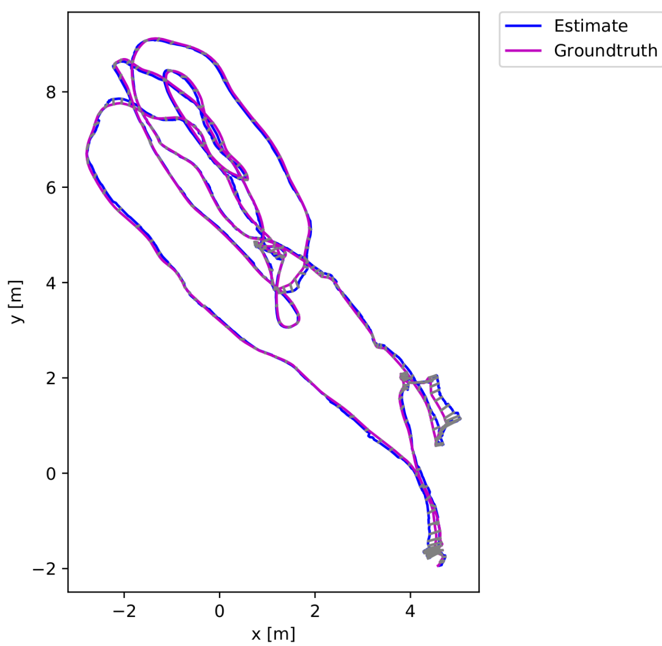

Fig. 3. Trajectory Top View

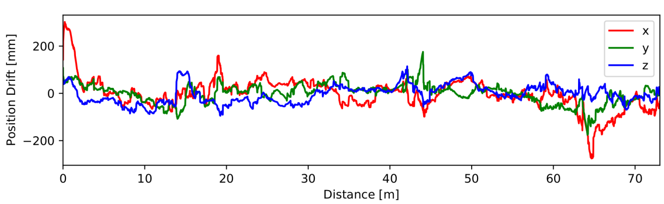

Fig. 4. Translation Drift

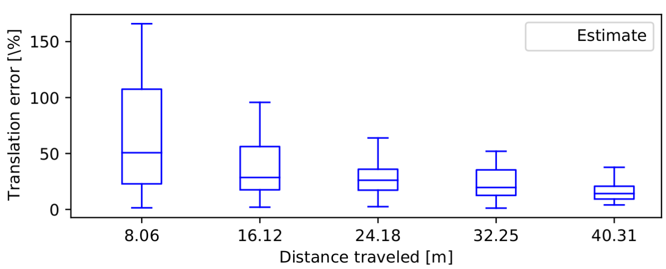

Fig. 5. Translation Error

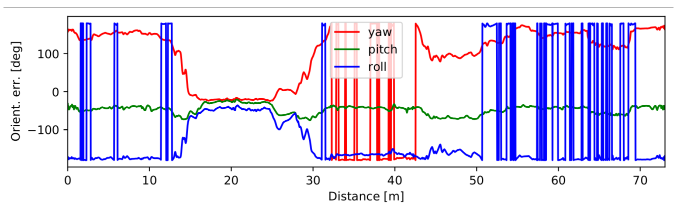

Fig. 6. Rotation Error

References

[1] Ke Sun, Kartik Mohta, Bernd Pfrommer, Michael Watterson, Sikang Liu, Yash Mulgaonkar, Camillo J. Taylor, and Vijay Kumar. Robust stereo visual inertial odometry for fast autonomous flight. *CoRR*, abs/1712.00036, 2017.

[2] Anastasios I. Mourikis and Stergios I. Roumeliotis. A multi-state constraint kalman filter for vision-aided inertial navigation. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 3565–3572, 2007.

[3] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*, 2016.