Project 4 - Phase 1: Classical Visual-Inertial Odometry RBE/CS549 Computer Vision

Shambhuraj Mane
MS Robotics Engineering
samane@wpi.edu

Swati Shirke
MS Robotics Engineering
svshirke@wpi.edu

Jesulona Akinyele
MS Robotics Engineering
jfakinyele@wpi.edu

***Abstract*—This paper subsumes of the end-to-end pipeline of implementation of classical Visual-Inertial Odometry. The technique consists of a fusion of Camera and IMU data for state estimation using a Multi-State Constraint Kalman Filter (MSCKF).**

Index Terms — Visual Intertial Odometry

I. INITIALIZE IMU GRAVITY AND BIAS

For every sensor, there is some bias value which refers to a systematic error or offset in the measurements provided by a sensor. Gravity and Gyroscope bias are initialized. We are using the reading from the first 200 messages of IMU to calculate bias and gravity values. The initial value of the gravity vector g is set to $[0, 0, gnorm]$, where gnorm represents the magnitude of the first 200 accelerometer readings.

II. BATCH IMU PROCESSING

IMU data received is stored into a buffer, as we want to use it in a First-Come-First-Served basis. We utilize the process model to update our state estimation based on each IMU message within the IMU message buffer received before the feature timestamp.

III. PROCESS MODEL

System Dynamics of IMU are given by the following equations. These derivatives are integrated by using the 4th-order Range-Kutta equation to find the values of these state variables.

$$\begin{aligned} {}^I_G\dot{\bar{q}} &= \frac{1}{2}\Omega(\omega(t))\, {}^I_G\bar{q}(t), \\ \dot{b}_g(t) &= n_{wg}(t), \\ {}^G\dot{v}_I(t) &= {}^Ga(t), \\ \dot{b}_a(t) &= n_{wa}(t), \\ {}^G(\dot{p})_I(t) &= {}^Gv_I(t), \end{aligned} \tag{1}$$

where $\hat{\omega} \in \mathbb{R}^3$ and $\hat{a} \in \mathbb{R}^3$ are the IMU measurements for angular velocity and acceleration respectively with biases removed.

Where ${}^I_G\bar{q}(t)$ is the unit quaternion describing the rotation from global frame G to IMU frame I. $\omega(t) = [\omega_x, \omega_y, \omega_z]^T$ is the rotational velocity in IMU frame, and

$$\lfloor \omega_\times \rfloor = \begin{bmatrix} -\lfloor \omega_\times \rfloor & \omega \\ \omega^T & 0 \end{bmatrix} \tag{2}$$

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

Gyroscope measurement is given by the following equation

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

Accelerometer measurement is given by the following equation

$$a_m = {}^I_GR({}^Ga - {}^Gg + 2\lfloor \omega_{G\times} \rfloor\, {}^Gv_I + 2\lfloor \omega_{G\times} \rfloor\, {}^{2G}p_I) + b_a + n_a \tag{5}$$

where ${}^I_GR$ is the rotation matrix calculated from quaternion ${}^I_G\bar{q}$. By applying the expectation operator on equation 1, the following equations are obtained for propagating IMU state estimates:

$$\begin{aligned} {}^I_G\dot{\hat{\bar{q}}} &= \frac{1}{2}\Omega(\hat{\omega}^I_G)\hat{\bar{q}}, \\ \dot{\hat{b}}_g &= 0_{3\times1}, \\ {}^G\dot{\hat{v}}_I &= C^T_{\hat{q}}\hat{a} - 2\lfloor \omega_{G\times} \rfloor\, {}^G\hat{v}_I + \lfloor \omega_{G\times} \rfloor\, {}^{2G}\hat{p}_I + {}^Gg, \\ \dot{\hat{b}}_a &= 0_{3\times1}, \\ {}^G\dot{\hat{p}}_I &= {}^G\hat{v}_I \end{aligned} \tag{6}$$

The continuous time model of an IMU is governed by the following equation.

$$\dot{\tilde{X}}_{IMU} = F\tilde{X} + Gn_{IMU} \tag{7}$$

Where F and G are:

$$F = \begin{bmatrix} \lfloor \omega_\times \rfloor & -I_3 & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times1} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ -C({}^I_G\hat{q})^T\lfloor \hat{a}_\times \rfloor & 0_{3\times3} & 0_{3\times3} & -C({}^I_G\hat{q})^T & 0_{3\times1} \\ 0_{3\times3} & 0_{3\times3} & I_3 & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \end{bmatrix} \tag{8}$$

$$G = \begin{bmatrix} -I_3 & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & I_3 & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & -C({}^I_G\hat{q})^T & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & I_3 \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \end{bmatrix} \tag{9}$$

IV. PREDICT NEW STATE

State prediction in the Visual-Inertial Odometry system is performed using the Extended Kalman Filter (EKF). Given the non-linear nature of the state transition model, the EKF provides a recursive means to predict the state by linearizing the non-linear functions around the current estimate:

$$\hat{x}_{k+1|k} = f(\hat{x}_{k|k}, u_k) \tag{10}$$

where f represents the state transition function, $\hat{x}_{k|k}$ the previous state estimate, and u_k the control input from the IMU. The process noise covariance matrix Q is also updated to account for the uncertainty in prediction.

V. STATE AUGMENTATION

In the state augmentation function, the camera position ${}^G p_c$ and orientation ${}^C_G q$ are updated using the last IMU state. The state covariance matrix P is also augmented to include these new states.

A. Camera Pose Computation

The pose of the camera is computed based on the latest IMU state using the following equations:

$${}^G p_c = {}^G p_I + C({}^C_G q)^T {}^I p_c \tag{11}$$

$${}^C_G \mathbf{q} = {}^C_I \mathbf{q} \otimes {}^I_G \mathbf{q} \tag{12}$$

where ${}^G p_I$ is the position of the IMU, ${}^C_I \mathbf{q}$ and ${}^I_G \mathbf{q}$ are the quaternions representing the rotations from the camera to the IMU and from the IMU to the global frame, respectively, and ${}^I p_c$ is the position of the camera relative to the IMU.

B. Jacobian Matrix Computation

The Jacobian J of the transformation is computed to facilitate the update of the covariance matrix P. The Jacobian matrix is given by:

$$J = [J_1 \quad 0_{6\times6N}] \tag{13}$$

$$J_1 = \begin{bmatrix} C({}^I_C q) & 0_{3\times9} & 0_{3\times3} & I_3 & 0_{3\times3} \\ [C({}^I_G q)^T {}^I p_C \times] & 0_{3\times9} & I_3 & 0_{3\times3} & I_3 \end{bmatrix} \tag{14}$$

where $0_{3\times9}$ and $0_{3\times3}$ are zero matrices of appropriate sizes, I_3 is the 3×3 identity matrix, and $\times$ denotes the cross product matrix of ${}^I p_C$.

C. Covariance Matrix Augmentation

Using the Jacobian matrix J, the state covariance matrix P is augmented to reflect the increased uncertainty due to the addition of the camera state:

$$P_{k|k} = \begin{bmatrix} I_{21+6N} \\ J \end{bmatrix} P_{k|k} \begin{bmatrix} I_{21+6N} \\ J \end{bmatrix}^{\mathsf{T}} \tag{15}$$

where I_{21+6N} is the identity matrix sized to match the augmented state dimensions, incorporating all previous states and the newly added camera state.

This structured approach ensures that the visual information, represented by the camera pose, is correctly integrated into the overall state estimate of the system. This integration helps in maintaining an accurate and robust estimation of the system state in a dynamic environment.

VI. ADD FEATURE OBSERVATION

Newly detected features are added to the feature map with their observed measurements and associated uncertainties. For each feature observed in the image frame at time t, the feature's location in pixel coordinates is added to the state vector along with a unique identifier. The state update is given by:

$$Z^i_j = \begin{bmatrix} u^{i,1}_j \\ v^{i,1}_j \\ u^{i,2}_j \\ v^{i,2}_j \end{bmatrix} \tag{16}$$

where $u^{i,1}_j$ and $v^{i,1}_j$ are the coordinates in the left camera, and $u^{i,2}_j$ and $v^{i,2}_j$ in the right camera for the i-th observation of feature j.

VII. MEASUREMENT UPDATE

The measurement update function processes the measurement matrix H and the residual vector r to compute the Kalman gain K, and subsequently updates the IMU state X_{IMU}, camera state, and the state covariance matrix P.

Constructing the Measurement Matrix H: The measurement matrix H consists of block rows $H(j)$, where $j = 1, \ldots, L$ corresponds to all detected features. When the number of measurements exceeds the number of state components, which is often the case, QR decomposition is applied to matrix H to manage its dimensionality and improve numerical stability:

$$H = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix}$$

Here, Q_1 and Q_2 are orthogonal matrices from the QR decomposition, and R is an upper triangular matrix representing the reduced form of H.

Computing the Residual r_n: The residual r_n is calculated by projecting the original residual r onto the column space of Q_1, effectively reducing the influence of measurement noise:

$$r_n = Q_1^T r = R\tilde{X} + n_n$$

where $\tilde{X}$ denotes the state error and n_n represents the noise in the measurement process.

Covariance Matrix of the Noise Vector n_n: The covariance matrix R_n of the noise vector n_n is derived from the noise characteristics of the measurements, typically expressed as:

$$R_n = \sigma_{\text{im}}^2 I_q \times q$$

where σ_{im}^2 is the variance of the measurement noise and q is the dimensionality of the subspace spanned by Q_1.

Kalman Gain K: The Kalman gain K is typically computed using:

$$K = PH^T(RR^T + R_n)^{-1}$$

To circumvent numerical instabilities inherent in inverting large matrices directly, the computation is reformulated as solving the linear system $Ax = b$ where:

$$A = RPR^T + R_n \quad \text{and} \quad b = RP^T$$

Here, A is symmetric, ensuring a more stable solution when computing K as:

$$K^T = A^{-1}b$$

State Correction ΔX: Once K is obtained, it is used to compute the correction for the state:

$$\Delta X = Kr_n$$

Updating the Covariance Matrix P: Finally, the state covariance matrix P is updated to reflect the reduced uncertainty after the measurement update:

$$P_{k+1|k+1} = (I - KH)P_{k|k}(I - KH)^T + KR_nK^T$$

This rigorous approach to the measurement update ensures that visual information is optimally integrated into the state estimate, improving the precision and reliability of the visual-inertial odometry system.

VIII. TRAJECTORY ERROR EVALUATION

We plotted the error between Ground Truth and Estimate Trajectory with the rpg trajectory evaluation toolbox. The absolute median trajectory error (ATE) is 0.08970106068178003 m and the root mean square translation error (RMSE) is 0.11358074282391588 m.

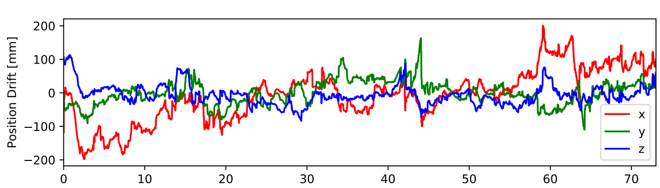

Fig. 1. Translation Error

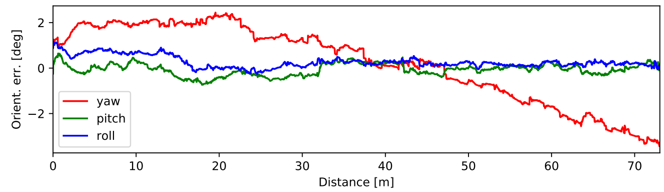

Fig. 2. Rotation Error

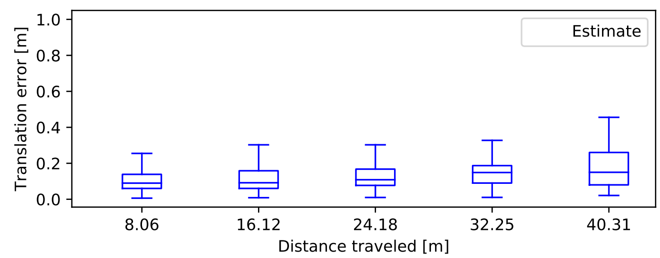

Fig. 3. Relative Translation Error (Absolute)

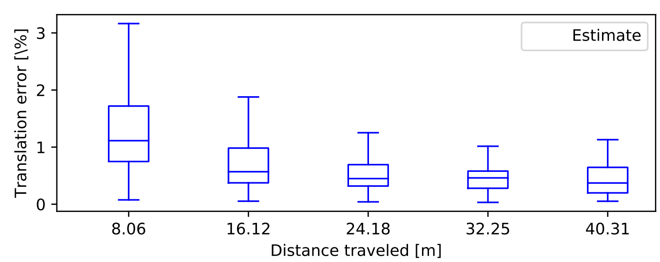

Fig. 4. Relative Translation Error (Percentage)

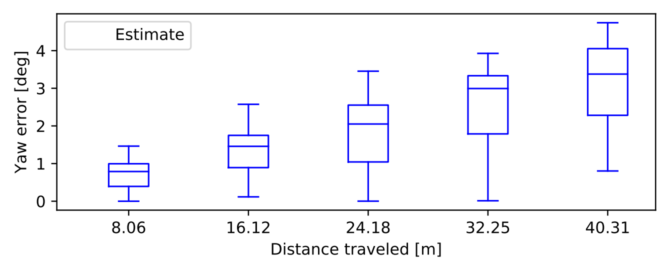

Fig. 5. Relative Yaw Error

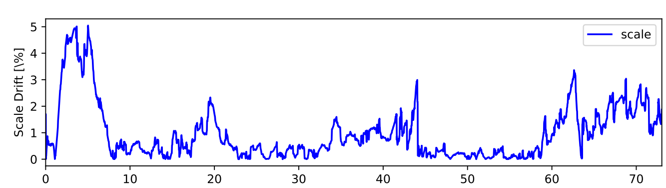

Fig. 6. Scale Error

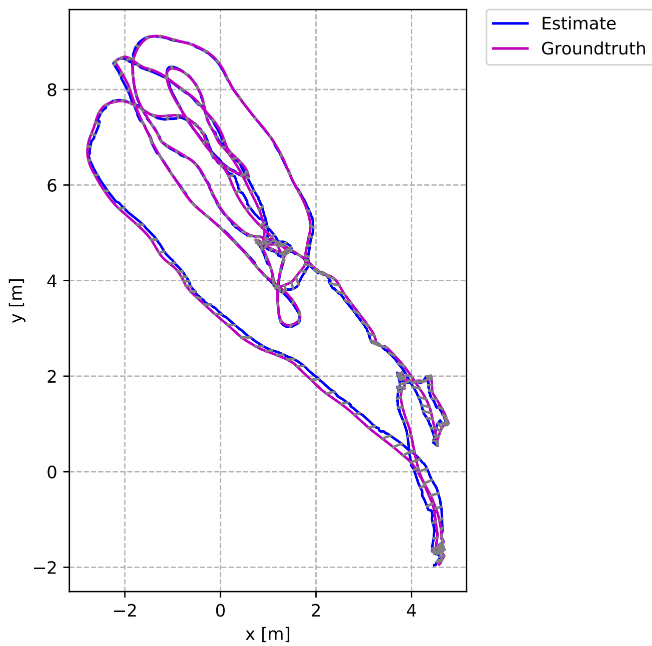

Fig. 7. Trajectory(Top View)

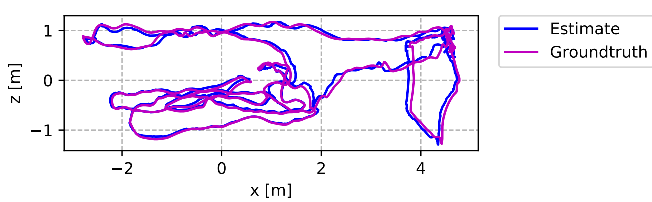

Fig. 8. Trajectory(Side View)

VISUAL-INERTIAL ODOMETRY RESEARCH PROBLEMS

Advanced Feature Detection and Matching: Researching algorithms that remain robust under varying lighting conditions and dynamic environments, focusing on feature detection and matching to improve the accuracy and reliability of VIO systems.

Outlier Rejection Techniques: Developing sophisticated techniques for outlier rejection to handle erroneous feature associations, which are critical for maintaining the integrity of VIO in complex scenes.

Alternative Filtering Techniques: Exploring filtering techniques beyond the Extended Kalman Filter (EKF), such as the Unscented Kalman Filter (UKF) and Multi-State Constraint Kalman Filters (MSCKF), to enhance the state estimation process in VIO systems.

Trajectory Planning for Uncertainty Management: Investigate methods for planning trajectories that actively mitigate the growth of uncertainty in VIO estimation. This could involve developing algorithms that intelligently steer the robot to minimize changes in uncertain directions or adjust its motion to maintain better observability of critical state variables.

Adaptive Sensor Fusion Strategies: Explore adaptive sensor fusion strategies that dynamically adjust the weighting of visual and inertial measurements based on the estimated uncertainty. By incorporating feedback from the VIO estimator's uncertainty, the fusion process can be optimized to maintain stability and accuracy over longer periods of operation.

Long-Term Estimation Stability: Develop techniques to improve the long-term stability of VIO estimation by addressing the inherent limitations in observability. This could involve incorporating additional sensor modalities, leveraging external information sources (such as GPS or landmarks), or implementing advanced filtering and smoothing techniques to mitigate the effects of accumulated uncertainty over time.

REFERENCES

[1] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks." arXiv, Jan. 28, 2018. doi: 10.48550/arXiv.1608.06993.

[2] A. I. Mourikis and S. I. Roumeliotis, "A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation," Proceedings 2007 IEEE International Conference on Robotics and Automation. IEEE, Apr. 2007. doi: 10.1109/robot.2007.364024.