RBE 549 Computer Vision Project 4 - Deep and Un-Deep Visual Inertial Odometry Phase 1

Kaushik Kavuri Subrahmanya
Robotics Engineering
Worcester Polytechnic Institute
Worcester, Massachusetts 01609
Email: ksubrahmanya@wpi.edu

Butchi Adari Venkatesh
Robotics Engineering
Worcester Polytechnic Institute
Worcester, Massachusetts 01609
Email: badari@wpi.edu

***Abstract*—In this project, we implement the Python translation of the influential paper on Visual Inertial Odometry by Vijay Kumar et al.**

I. Introduction

We implement a select few functions from the paper and incorporate these into the starter code provided. The functions implemented are part of the Stereo Multi-State Constraint Kalman Filter (S-MCKF). The seven functions implemented are discussed below.

A. Initialize Gravity And Bias

The function first initializes and sums the IMU readings of angular velocities and linear accelerations. The mean of the angular velocities measured for the initial few readings is taken as the bias of the gyroscope, and the norm of the linear accelerations for the initial few readings is taken as gravity after being converted into a quaternion.

B. Batch IMU Processing

This function takes the IMU messages in the buffer for each specified time bound and converts the angular velocity and linear acceleration data from each message into Eigen vectors. These vectors are then passed on to the Process Model. This process is repeated till the end of the time bound. In the end, processed messages are removed from the bufer.

C. Process Model

The state of the IMU is defined by

$$\mathbf{X}_{\text{IMU}} = \begin{bmatrix} {}^{I}_{G}\mathbf{q}^{T} & \mathbf{b}_{g}{}^{T} & {}^{G}\mathbf{v}_{I}{}^{T} & \mathbf{b}_{a}{}^{T} & {}^{G}\mathbf{p}_{I}^{T} \end{bmatrix}^{T} \quad (1)$$

Here, the quaternion ${}^{I}_{G}\mathbf{q}^{T}$ represents the rotation from the inertial frame to the body frame. The third and fifth terms represent the velocity and position of the body frame in the inertial frame. The second and fourth terms represent the biases of the measured angular velocity and linear acceleration from the IMU. The seventh and the eighth term represents the relative transformation between the camera frame and the body frame. The error in IMU state is used as using the state directly could result in a few issues.
The dynamics of the estimated IMU state is given by,

$$\begin{gathered}
{}^{I}_{G}\dot{\hat{\mathbf{q}}} = \frac{1}{2}\Omega(\hat{\omega}){}^{I}_{G}\hat{\mathbf{q}}, \quad \dot{\hat{\mathbf{b}}}_{g} = \mathbf{0}_{3\times} \\
{}^{G}\dot{\hat{\mathbf{v}}} = C\left({}^{I}_{G}\hat{\mathbf{q}}\right)^{\top}\hat{\mathbf{a}} + {}^{G}\mathbf{g}, \\
\dot{\hat{b}}_{a} = \mathbf{0}_{3\times 1}, \quad {}^{G}\dot{\hat{\mathbf{p}}}_{I} = {}^{G}\hat{\mathbf{v}}, \\
{}^{I}_{C}\dot{\hat{\mathbf{q}}} = \mathbf{0}_{3\times 1}, \quad {}^{I}\dot{\hat{\mathbf{p}}}_{C} = \mathbf{0}_{3\times 1}
\end{gathered}$$

where,

$$\Omega(\hat{\omega}) = \begin{pmatrix} -\lfloor\hat{\omega}_{\times}\rfloor & \omega \\ -\omega^{\top} & 0 \end{pmatrix}$$

and ω and a are the IMU measurements for angular velocity and acceleration respectively with biases removed.

The linearized continuous dynamics for the error IMU state is given by

$$\dot{\tilde{\mathbf{x}}}_{I} = \mathbf{F}\tilde{\mathbf{x}}_{I} + \mathbf{G}\mathbf{n}_{I} \quad (2)$$

where $\mathbf{F}$ and $\mathbf{G}$ are given by,

$$\mathbf{F} = \begin{bmatrix}
-\lfloor\hat{\omega}\times\rfloor & -\mathbf{I}_{3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} \\
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} \\
-\mathbf{C}_{\hat{q}}^{T}\lfloor\hat{\mathbf{a}}\times\rfloor & \mathbf{0}_{3\times 3} & -2\lfloor\omega_{G}\times\rfloor & -\mathbf{C}_{\hat{q}}^{T} & -\lfloor\omega_{G}\times\rfloor^{2} \\
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} \\
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{I}_{3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3}
\end{bmatrix}$$

$$\mathbf{G} = \begin{bmatrix}
-\mathbf{I}_{3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} \\
\mathbf{0}_{3\times 3} & \mathbf{I}_{3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} \\
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -\mathbf{C}_{\hat{q}}^{T} & \mathbf{0}_{3\times 3} \\
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{I}_{3} \\
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3}
\end{bmatrix}$$

D. Predict New State

In order to deal with discrete time measurement from the IMU, we apply a 4th order Runge-Kutta numerical integration of the state to propagate the estimated IMU state. We first calculate the norm of the orientation, then the orientation, velocity and position are approximated using Runge-Kutta method.

E. State Augmentation

This function calculates the state covariance matrix in order to propagate the state's uncertainty. We take the IMU and camera state values, the rotation and the translation vector from the camera to the IMU. The pose of the new camera state can be computed from the latest IMU state as,

$$ {}^{C}_{G}\hat{\mathbf{q}} = {}^{C}_{I}\hat{\mathbf{q}} \otimes {}^{I}_{G}\hat{\mathbf{q}}, \quad {}^{G}\hat{\mathbf{p}}_C = {}^{G}\hat{\mathbf{p}}_C + C\left({}^{I}_{G}\hat{\mathbf{q}}\right)^{\top} {}^{I}\hat{\mathbf{p}}_C $$

The augmented covariance matrix is given by,

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

where J is given by,

$$ \mathbf{J}_I = \begin{pmatrix} C\left({}^{I}_{G}\hat{\mathbf{q}}\right) & \mathbf{0}_{3\times 9} & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} \\ -C\left({}^{I}_{G}\hat{\mathbf{q}}\right)^{\top} \lfloor {}^{I}\hat{\mathbf{p}}_{C\times} \rfloor & \mathbf{0}_{3\times 9} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{I}_3 \end{pmatrix} $$

F. Add Feature Observations

This function is used to update the features from the feature message into the map server.

G. Measurement Update

This function updates the state estimates. We calculate the Kalman gain and state error by decomposing the Jacobian matrix. Using these, IMU and camera states are updated. Finally, the state covariance is updated, and the covariance matrix is modified to be symmetric.

II. Results

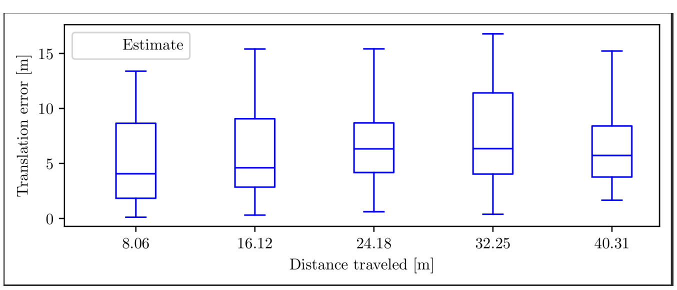


Fig. 1: Relative Translation error

Acknowledgment

The authors would like to thank the professor for the instructions and knowledge imparted via the course.

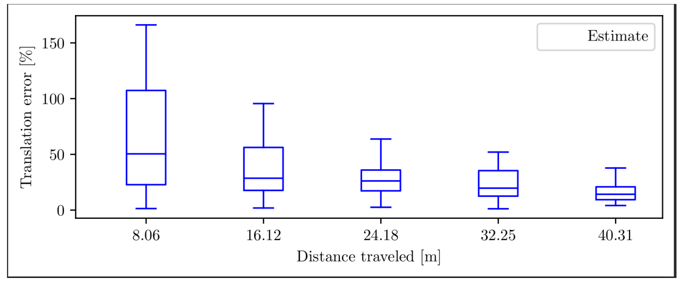


Fig. 2: Relative Translation Error Percentage

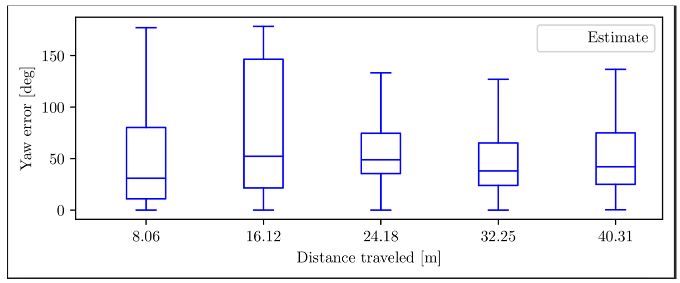


Fig. 3: Relative Yaw Error

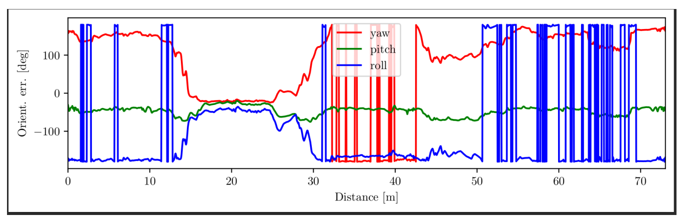


Fig. 4: Rtaton Error

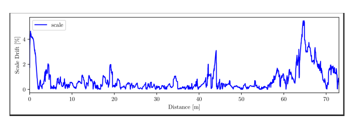
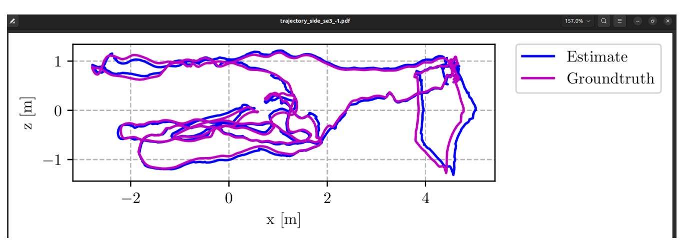
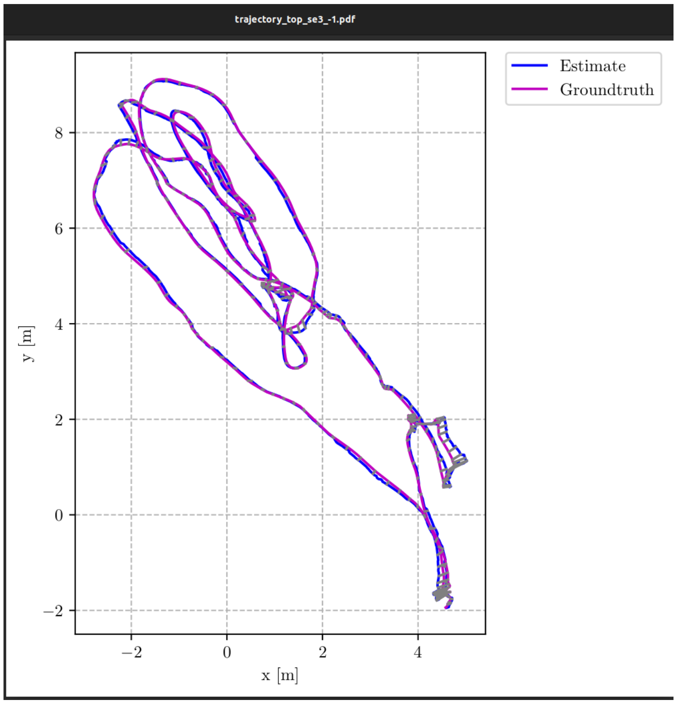
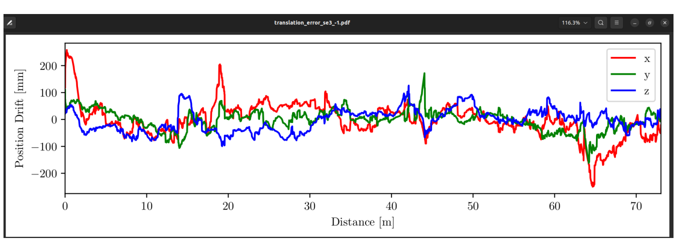


Fig. 5: Scale error




trajectory_top_se3_-1.pdf
Estimate
Groundtruth
y [m]
x [m]


translation_error_se3_-1.pdf
Position Drift [mm]
Distance [m]
x
y
z