# Visual Inertial Odometry

Shounak Naik
*Robotics Engineering Department,*
*Worcester Polytechnic Institute,*
Worcester, MA, USA.
ssnaik@wpi.edu

Venkatesh Mullur
*Robotics Engineering Department,*
*Worcester Polytechnic Institute,*
Worcester, MA, USA.
vmullur@wpi.edu

## I. Introduction

This project aims to implement Vision-aided Odometry using Multi-state Constraint Kalman Filter (MSCKF). The need to implement MSCKF in the application of Visual Intertial Odometry is because of the challenges faced by the autonomous flight vehicle to use high-quality sensors and efficient processors. With this approach, we can compensate for costly and heavy sensors and processors. In this project, we implement a filter-based approach with help of the fusion of two sensors: A stereo camera and an IMU. In this implementation, we use MSCKF to determine the state and localization of the robot is done by sensor fusion of IMU and a stereo camera.

## II. Initialize Gravity and Bias

The 6-DOF IMU sensor used goes through a bias and has to be fixed in every reading. The 6-DOF defines the 3 degrees for rotation (gyroscope) and 3 degrees for acceleration (accelerometer). This can also be called as calibration of the IMU sensor, where we calculate the bias in both rotation and acceleration and then subtract this bias from every other reading of the IMU.

This is done by keeping the rotor stationary for around 100-200 readings and then taking the mean of these readings. Ideally, the gyroscope reading should be $[0, 0, 0]$ but due to the noise and bias, there is a small fluctuating value present in the gyroscope reading.
To calibrate this, as mentioned above we take a mean of a few readings while keeping the rotor stationary, and then subtract them from the next IMU readings.

Ideally, the accelerometer reading should be $[0, 0, -g]$ in the world frame but due to the noise and bias in the cheap IMU sensor, we observe some fluctuations in these readings as well. Since the accelerometer measures the linear acceleration in 3 axes, so gravity is mentioned in the 3rd axis. The bias in the accelerometer is also removed using the same process used in the gyroscope. These activities are performed before the start of the flight.

## III. Batch IMU processing

IMU batch processing is done to read the IMU message till the next set of images is available from the stereo camera.

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

Fig. 1. State Vector

$$\widetilde{\mathbf{X}}_k = \begin{bmatrix} \widetilde{\mathbf{X}}_{\mathrm{IMU}_k}^T & \boldsymbol{\delta\theta}_{C_1}^T & {}^G\widetilde{\mathbf{p}}_{C_1}^T & \dots & \boldsymbol{\delta\theta}_{C_N}^T & {}^G\widetilde{\mathbf{p}}_{C_N}^T \end{bmatrix}^T$$

Fig. 2. Error State Vector

Before that, it is necessary to define what the state vector looks like to estimate the next states. The state vector consists of the states in the camera as well as IMU. The state vector looks like the following.

Figure 1 shows the state vector, where $q$ is the quaternion that describes the rotation from the global to the IMU frame. $bg$ represents the bias in gyro and $ba$ is the bias in accelerometer. $pi$ and $vi$ are the positions and velocities of the body frame in the inertial(world) frame. $q_c^I$ and $p_c^I$ are the transformations between the IMU and camera frames.
For $N$ camera poses, the state vector adds a new state in the buffer with the first element being the states with the IMU sensor.

The aim of the $batch\_imu\_processing$ function is to predict the next state and update the state information using the process model for a given time step given the IMU messages. The state information is updated after processing the IMU data and terminates after time-bound.

## IV. Process Model

The process model predicts the IMU state using the motion model where the motion model is derived from the error states which are given below in figure 2. The error in quaternion is a quaternion operation which is shown below.

$$\delta q = q \bigotimes q^{-1}$$

and the other errors are additive errors where they just add to the previous quantity. These error states are used to determine the process model of the robot. The angular velocity and linear acceleration are derived which is gven in figure 3 where the

$$
\begin{gathered}
{}^{I}_{G}\dot{\hat{\mathbf{q}}} = \frac{1}{2}\Omega(\hat{\boldsymbol{\omega}}){}^{I}_{G}\hat{\mathbf{q}}, \quad \dot{\hat{\mathbf{b}}}_g = \mathbf{0}_{3\times1}, \\
{}^{G}\dot{\hat{\mathbf{v}}} = C\left({}^{I}_{G}\hat{\mathbf{q}}\right)^{\top}\hat{\mathbf{a}} + {}^{G}\mathbf{g}, \\
\dot{\hat{b}}_a = \mathbf{0}_{3\times1}, \quad {}^{G}\dot{\hat{\mathbf{p}}}_I = {}^{G}\hat{\mathbf{v}}, \\
{}^{I}_{C}\dot{\hat{\mathbf{q}}} = \mathbf{0}_{3\times1}, \quad {}^{I}\dot{\hat{\mathbf{p}}}_C = \mathbf{0}_{3\times1}
\end{gathered}
$$

Fig. 3. IMU dynamics

$$
\boldsymbol{\Phi}_k = \boldsymbol{\Phi}(t_{k+1}, t_k) = \exp\left(\int_{t_k}^{t_{k+1}} \mathbf{F}(\tau)d\tau\right)
$$

$$
\mathbf{Q}_k = \int_{t_k}^{t_{k+1}} \boldsymbol{\Phi}(t_{k+1}, \tau)\mathbf{G}\mathbf{Q}\mathbf{G}\boldsymbol{\Phi}(t_{k+1}, \tau)^{\top}d\tau
$$

Fig. 4. Covariance Matrix

$\hat{\omega}$ and $\hat{a}$ are given as follows.

$$\hat{\omega} = \omega_m - b_g$$

$$\hat{a} = m - b_g$$

where the $\Omega$ is the quaternion derivative and is given by the following.

$$\Omega(\omega) = \begin{bmatrix} \hat{\omega} & \omega \\ \omega_T & 0 \end{bmatrix}$$

here, $\hat{\omega}$ is a skew-symmetric matrix of the $\hat{\omega}$ vector and the linearized continuous error dynamics of IMU error state are defined as follows,

$$\dot{\tilde{X}}_I = F\tilde{X}_I + Gn_I$$

The $n_I$ denotes the Gaussian noise of the accelerometer and gyro reading. To propagate the IMU measurement in discrete time, we apply the Runge Kutta method of order 4.

F matrix in the above-given equation (discrete time equation) is used to derive the discrete time state transition matrix and the G matrix is used to obtain the discrete-time noise covariance matrix. $\phi_K$ is approximated by Taylor expansion till the 3rd order of F, while $Q_k$ is a discrete-time state covariance matrix obtained by continuous time methods of state covariance Q and G matrix. The observability constraint is applied by modifying the transition matrix. Here the state transition matrix is corrected by making it symmetric.

## V. Next Step Prediction

Here, we predict the Next Step of the robot using the Runge Kutta Method. This method is used to mathematically approximate the next timestep using a linear approximation of non-linear functions. Here we approximate the next IMU State given the conditions of the current IMU state. The RK 4 method can be summarized by the following flowchart. Basically, the calculated $k1$, $k2$, $k3$ and $k4$ can be used to obtain the next state of the IMU.

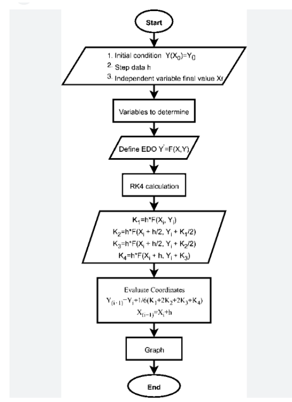


Fig. 5. Runge Kutta of Order 4

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

## VI. State Augmentation

When new images are received, the state should be augmented with the new camera state. The pose of the new camera state can be computed from the latest IMU state as follows The augmented covariance matrix is given by the following equation given in VI and the J matrix is given by the equation given in VI.

## VII. Adding feature Observation

Here, we check if the feature observed is in the map server dictionary or not. If it is not there, we add a new key to the map server dictionary.

## VIII. Measurement model and update

A single feature $fj$ is observed by the stereo cameras with the pose. The stereo cameras have different poses, for left and right cameras respectively, at the same time instance. Although the state vector only contains the pose of the left camera, the pose of the right camera can be easily obtained using the extrinsic parameters from the calibration.

The dimension is then reduced to $R^3$ assuming the stereo images are properly rectified. But by representing the same in $R^4$ we can skip the rectification and the camera poses are given by The position of the feature in the world frame is calculated using Gaussian-Newton least square minimization. The residual of measurement can be approximated by the following equation

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

$$
\mathbf{J}_I = \begin{pmatrix} C\left({}^{I}_{G}\hat{\mathbf{q}}\right) & \mathbf{0}_{3\times9} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} \\ -C\left({}^{I}_{G}\hat{\mathbf{q}}\right)^{\top}\lfloor {}^{I}\hat{\mathbf{p}}_{C\times}\rfloor & \mathbf{0}_{3\times9} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{I}_3 \end{pmatrix}
$$

$$
\mathbf{z}_i^j = \begin{pmatrix} u_{i,1}^j \\ v_{i,1}^j \\ u_{i,2}^j \\ v_{i,2}^j \end{pmatrix} = \begin{pmatrix} \frac{1}{^{C_{i,1}}Z_j} & \mathbf{0}_{2\times 2} \\ \mathbf{0}_{2\times 2} & \frac{1}{^{C_{i,2}}Z_j} \end{pmatrix} \begin{pmatrix} ^{C_{i,1}}X_j \\ ^{C_{i,1}}Y_j \\ ^{C_{i,2}}X_j \\ ^{C_{i,2}}Y_j \end{pmatrix}
$$

Fig. 6. Stereo camera measurement

$$
\begin{aligned}
^{C_{i,1}}\mathbf{p}_j &= \begin{pmatrix} ^{C_{i,1}}X_j \\ ^{C_{i,1}}Y_j \\ ^{C_{i,1}}Z_j \end{pmatrix} = C\left(^{C_{i,1}}_{G}\mathbf{q}\right)\left(^{G}\mathbf{p}_j - {}^{G}\mathbf{p}_{C_{i,1}}\right) \\
^{C_{i,2}}\mathbf{p}_j &= \begin{pmatrix} ^{C_{i,2}}X_j \\ ^{C_{i,2}}Y_j \\ ^{C_{i,2}}Z_j \end{pmatrix} = C\left(^{C_{i,2}}_{G}\mathbf{q}\right)\left(^{G}\mathbf{p}_j - {}^{G}\mathbf{p}_{C_{i,2}}\right) \\
&= C\left(^{C_{i,2}}_{C_{i,1}}\mathbf{q}\right)\left(^{C_{i,1}}\mathbf{p}_j - {}^{C_{i,1}}\mathbf{p}_{C_{i,2}}\right)
\end{aligned}
$$

Fig. 7. Camera poses of left and right

The pose of the feature in the global frame is calculated using the camera pose and thus the uncertainty of $p_j$ in a global frame is correlated to camera states. To remove this correlation, the residual in equation 4 is projected onto the null space V of $H_J$

- Find Cam0 pose.
- Find Cam1 pose.
- Find 3d feature position in the world frame and its observation with the stereo cameras
- Convert the feature position from the world frame to the cam0 and cam1 frame.
- Modifies the measurement Jacobian to ensure observability constrain.
- Compute the residual.

## IX. Updating

The update process is done in the following way,

- Check if H and r are empty
- Decompose the final Jacobian matrix to reduce computational complexity
- Compute the Kalman gain.
- Compute the error of the state.
- Update the IMU state.
- Update the camera states.
- Update state covariance.
- Fix the covariance to be symmetric

## X. Results

The results of our implementation can be seen as follows. We have also plotted errors with respect to ground truth of the EuROC dataset.

$$
r_o^j = V^T r^j = H_{x,o}^j \tilde{x} + n_o^j
$$

Fig. 8. Removing the correlation /Residual equation

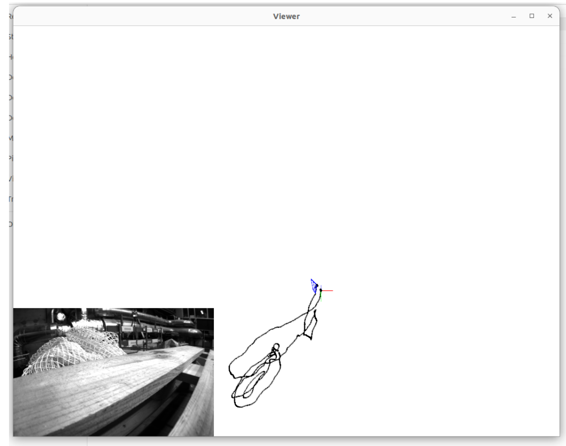


Fig. 9. The resulting trajectory as seen via pangolin.

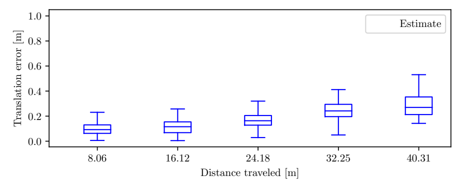


Fig. 10. Relative Translation Error

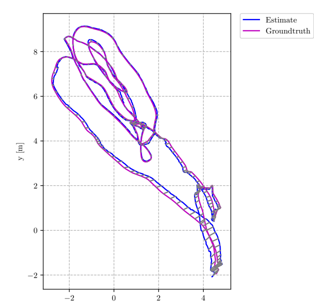


Fig. 11. Ground truth vs Estimated Trajectory.

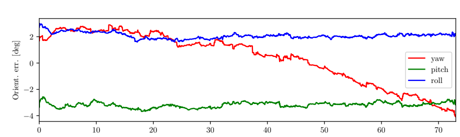


Fig. 12. Rotatation Translation Error