Project4: VIO

30 November 2022

Thabsheer Jafer Machingal
M.S. Robotics Engineering
Worcester Polytechnic Institute
Worcester, MA
tmachingal@wpi.edu

Krishna Madhurkar
M.S. Robotics Engineering
Worcester Polytechnic Institute
Worcester, MA
ksmadhurkar@wpi.edu

I. Introduction

Odometry is the use of data from the movement of actuators to estimate change in position over time through devices such as rotary encoders, camera, IMU e.t.c.. Visual odometry is the process of determining the position and orientation of a robot by analyzing the associated camera images. Visual and Inertial Odometry (VIO)is the practice of coupling camera with and IMU to determine the position and orientation of a robot. VIO is the only viable alternative to GPS and lidar-based odometry to achieve accurate state estimatation[1]. VIO can be loosely coupled or tightly coupled[2] and loosely couple VIO is preferred in odometry, because it is cheaper to compute.

II. Data

We are using the Machine Hall 01 easy subset of the EuRoC dataset[4] to test our implementation. The data is collected using a VI sensor, which includes synchronized 20Hz stereo images and 200Hz IMU messages, carried by a quadrotor flying a trajectory. The ground truth is provided by a sub-mm accurate Vicon Motion capture system.

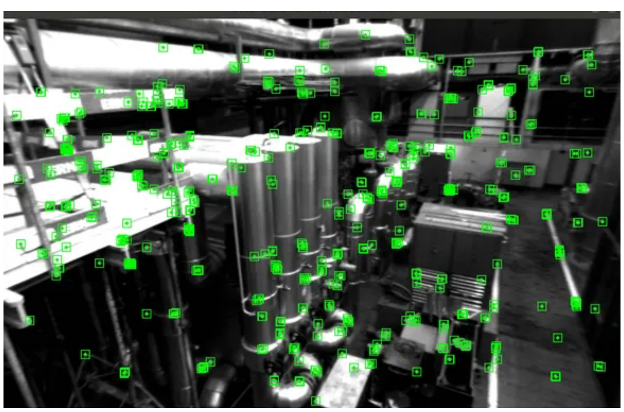

Fig. 1. Sample image with features in the dataset

III. Pipeline

We are given the starter code which contains the code for simulation and the environment is configured. We are required to write 8 functions and ther formula is given in the paper[3] and the appendix has the necessary equations for the implementation. We wrote the following functions.

- Initializes the gravity and IMU bias
- Batch IMU processing
- Process model
- Predict new state
- State augmentation
- Add feature observations
- measurement update

The architecture is called Stereo Multi-State Constraint Kalman Filter (S-MSCKF), utilizes a kalman filter for the problem.

A. Filter description

The IMU state is defined as:
$x_I = ({}^I_G q^T, b_g^T, {}^G v_I^T, b_a^T, {}^G P_I^T, {}^I_C q^T, {}^I p_c^T)$ where ${}^I_G q$ is a querternion that represents the rotation from the inertial frame to the body frame. Please note that body and IMU frame are the same. The vectors ${}^G v_I$, ${}^G p_I$ represents the velocity and position of the object in the inertial frame.

1) Initializes gravity and IMU biases: Note that ibn the above description of IMU, b_g and b_a are the biases and we initialilize the value of the biases for the first few steps. and Gravity is estimated from the linear acceleration on the body and its normalizes so that the estimation is consistent with the inertial frame. The quartenions ${}^I_C q$ and ${}^I p_C$ represents the relative transformation between camera and body frame(IMU frame). Using the true IMU state would force singularity problems in the covariance matrix, hence an error state is defined as follows:
And the error quartenion $\delta q = q@q_{est}^{-1}$ is related as equation1.

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

$$\delta q = (\frac{1}{2} R_s^T 1) \qquad (1)$$

where R_s is a small angle rotation in 3D (${}^G_I\theta$). Please note that for N camera states the state vector would be $x_{est} = (x_{estI}^T, x_{estC1}^T....x_{estCN}^T)^T$ where each camera error state is defined as follows:

$$x_{estCi}^{T} = ({}_{G}^{C_i}\theta_{est}^{T}, Gp_{est}{}_{C_i}^{T}) \tag{2}$$

B. Process model

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

where $\hat{\omega}$ and $\hat{a}$ are the IMU measurements for angular velocity and acceleration respectively with biases removed. meaning, $\hat{\omega} = \omega_m - \hat{b}_g$ and $\hat{a} = a_m - \hat{b}_a$.

C. Observability constraint

Observability is a the property of a system to estimate a variable from a different known variable, this is essential in a coupled system, like VIO. When it comes to EKF-based vio, there are multiple methods to achieve this. Note that we are implementing observability constraint EKF (OC-EKF), this method does not heavily depend on an accurate initial estimate.

D. Measurement update

The measurement update step is executed when either the algorithm loses a feature or the number of camera poses in the state reaches the limit.One third of the camera states are marginalized once the buffer is full, which can cause sudden jumps in computational load in realtime implementations. As described in the paper[3], two camera states are removed every other update step. All feature observations obtained at the two camera states are used for measurement update. Although we are producing two images from stereo camera, some frames have to be ngelected due to not enough correlation of features, this might limit some observation, but in practice this barely affects the performance.

E. Image Processing

1) Feature descriptor: For this project as explained in the paper[3], we are using FAST feature detector and the features are tracked using KLT optical flow algorithm. This method significantly reduces the CPU usage.

2) Outlier rejection: A 2-point RANSAC is applied to remove outliers in temporal tracking. In addition, a circular matching similar to is performed between the previous and current stereo image pairs to further remove outliers generated in the feature tracking and stereo matching steps.

IV. Result

As expected, we obtained the trajectory using camera and IMU measurements as the robot track the trajectory and return to the initial position. A video of the implementation is given in the directory as 'output.mp4'.

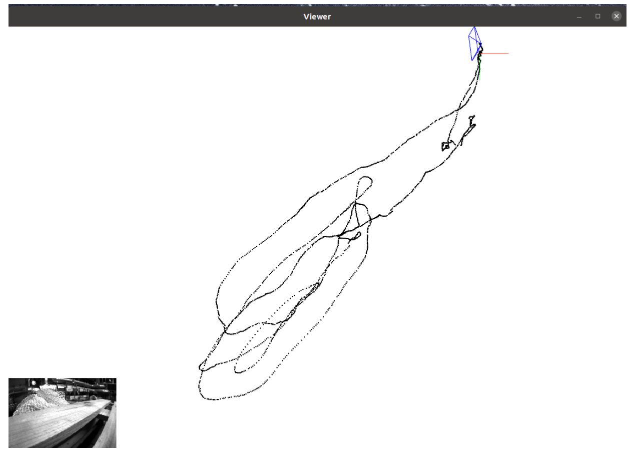

Fig. 2. Result from VIO

References

[1] "Visual and Inertial Odometry." http://www.ifi.uzh.ch/en/rpg/ research/researchv o.html(accessedN ov.30, 2022)

[2] C.Y uan, J.Lai, P.Lyu, P.Shi, W.Zhao, andK.Huang, ovelF ault T olerantN avigationandP ositioningM ethodwithStereo Camera/M icro-ElectroM echanicalSystemsInertialM easurementU nit(M EM S IM U)inHostileEnvironment, "M icromachines, vol.9, no.12, p.626, Dec.2018, doi : 10.3390/mi9120626.

[3] K. Sun et al., "Robust Stereo Visual Inertial Odometry for Fast Autonomous Flight," IEEE Robot. Autom. Lett., vol. 3, no. 2, pp. 965–972, Apr. 2018, doi: 10.1109/LRA.2018.2793349

[4] [1] M. Burri et al., "The EuRoC micro aerial vehicle datasets," The International Journal of Robotics Research, vol. 35, no. 10, pp. 1157–1163, Sep. 2016, doi: 10.1177/0278364915620033.