Sensing & Estimation in Robotics Project 1

Rick Ho, PID: A69030974

I. Introduction

The goal of this project is to solve the orientation tracking problem using the data collected by an IMU, including accelerations and angular velocity. The raw data after processing can be used to calculate the motion model and the observation model, transforming the orientation tracking problem into an optimization problem. In this report, gradient descent will be used to solve this nonlinear optimization problem.

II. Problem Formulation

A. Calibration

Data collected by the IMU needs calibration first. The equations of conversion are as follows:

$$value = (raw - bias) * scaleFactor \tag{1}$$

$$scaleFactor = Vref/2^{bits}/sensitivity \tag{2}$$

In this project, the A/D converter has 10 bits, which could have values from 0 to 1023.

B. Orientation Tracking

After converting the raw data, we can use it to estimate the motion model and the observation model.
Using consecutive angular velocity measurements, we can estimate the orientation by computing the following quaternions.

$$\mathbf{q}_{t+1} = f(\mathbf{q}_t, \tau_t\omega_t) := \mathbf{q}_t \circ exp([0, \tau_t\omega_t]) \tag{3}$$

Since the body is undergoing pure rotation, we then use these quaternions to compute the observation model:

$$[0, \mathbf{a}_t] = h(\mathbf{q}_t) := \mathbf{q}_t^{-1} \circ [0, 0, 0, -g] \circ \mathbf{q}_t \tag{4}$$

With the motion model and observation model we can reformulate the orientation tracking problem as an optimization problem. The cost function of optimization problem is

$$c(\mathbf{q}_{1:T}) := \frac{1}{2}\sum_{t=0}^{T-1} ||2log(\mathbf{q}_{t+1}^{-1} \circ f(\mathbf{q}_t, \tau_t\omega_t)||_2^2 + \frac{1}{2}\sum_{t=1}^{T} |||[0, \mathbf{a}_t] - h(\mathbf{q}_t)||_2^2 \tag{5}$$

The first term of the cost function (equation 5) measures the error of the motion model, and the second term measures the observation model. By solving $min_{\mathbf{q}_{1:T}} c(\mathbf{q}_{1:T})$ under the constraint$||\mathbf{q}_t||_2 = 1$, we obtain the optimal orientation.

C. Panorama

Using the estimated orientation, images capture by the camera can be mapped onto a sphere and then unwrapped into a larger image.

To stitch images, we perform a pixel-wise transformation and place each pixel in its corresponding location.

III. Technical Approach

A. Calibration

Referring to the document provided in the project file, we know that reference voltage(Vref) for the accelerometer is 300 mV/g and for the gyroscope is 3.33 mV/degree/sec. Since we know that the system is static during the first few seconds, the accelerometers should only measure gravity along z-axis. The bias could be calculated by solving the equation 1.

The gyroscopes bias should be zero in the first few seconds, so I compute the mean of the first 100 data.

B. Orientation tracking

To get gradient of the cost function(equation 5), I use the functions from jax library. Since the cost function are scalar, I simply use the $grad$ to calculate the gradient.

Before applying $grad$ to the cost function, all parameters and operations must be implemented using $jax.numpy$, so I set the variables in $jax.numpy$, and build our own quaternion operations. The main quaternion operations are multiplication, exp, log and inverse.

- Multiplication

$$\mathbf{q} \circ \mathbf{p} := [q_s p_s - \mathbf{q}^T\mathbf{p}, q_s\mathbf{p}_v + p_s\mathbf{q}_v + \mathbf{q}_v \times \mathbf{p}_v] \tag{6}$$

- Exp

$$exp(\mathbf{q}) := e^{q_s}[cos||\mathbf{q}_v||, \frac{\mathbf{q}_v}{||\mathbf{q}_v||}sin||\mathbf{q}_v||] \tag{7}$$

- Log

$$log(\mathbf{q}) := [log||\mathbf{q}||, \frac{\mathbf{q}_v}{||\mathbf{q}_v||}arccos\frac{q_s}{||\mathbf{q}||}] \tag{8}$$

- Inverse

$$\mathbf{q}^{-1} := \frac{\bar{\mathbf{q}}}{||\mathbf{q}||^2} \tag{9}$$

With these operations and jax library, we can calculate gradient of cost function with respect to $\mathbf{q}_{1:T}$. To speed up the computation, I implement this process using vmap function, which enable parallel vector computation. With all elements prepared, I implemented gradient descent by setting learning rate(or step), and ran through iterations to get the optimal orientation.

Finally, I transferred the quaternions $\mathbf{q}_{1:T}$ to Euler angles, a way to better express rotation visually. Comparing with Euler angles from VICON, we could know how good is the estimation.

C. Panorama

First, each pixel needs to be converted from spherical coordinates to Cartesian coordinates. Assuming each picture covers 60 degree horizontally and 45 degree vertically, pixels could get their spherical coordinates $(\lambda, \phi, 1)$, where λ is the angle between projection on $x-y\ plane$ and $x-axis$, and ϕ is the angle between line from origin to P and its projection on $x-y\ plane$, by:

$$\lambda = \frac{60}{320}(x - 160) \tag{10}$$

$$\phi = \frac{45}{240}(y - 120) \tag{11}$$

then convert each pixel to Cartesian coordinates by:

$$x = cos(\phi)sin(\lambda) \tag{12}$$

$$x = cos(\phi)cos(\lambda) \tag{13}$$

$$z = sin(\phi) \tag{14}$$

After getting the coordinates of each pixel, I used the orientation to rotate each pixel to the corresponding place. Note that the timestamp spacing for IMU and camera may be different. Assuming spacing is constant, I calculate the spacing ratio between IMU and camera, and extract orientation start from first camera timestamp every spacing ratio.

$$spacing\ ratio = \frac{\Delta IMU\ time\ spacing}{\Delta camera\ time\ spacing} \tag{15}$$

To simplify the mapping process, I directly map the spherical coordinates back to a $180 * 360$ array. In this way, I could show the augmented image directly.

IV. Results

To determine a suitable learning rate, I tested learning rates of $10^{-3},\ 5*10^{-4},\ 10^{-4},\ 5*10^{-5}$. In the end, the learning rate 10^{-3} converged the fastest without sacrificing too much accuracy.

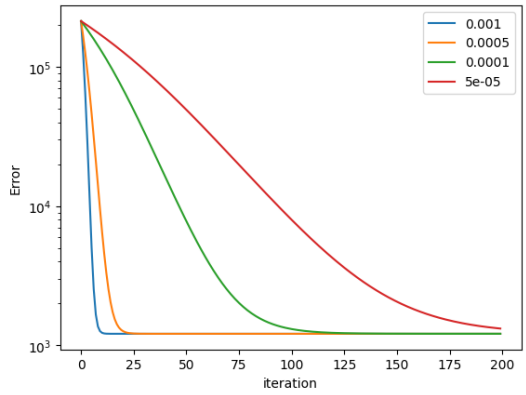

Fig. 1. Error (cost function) v.s iterations

To evaluate the performance of the orientation estimation, I plotted the estimated Euler angles against VICON ground truth. Also, with estimated orientation, we can stitch images together.

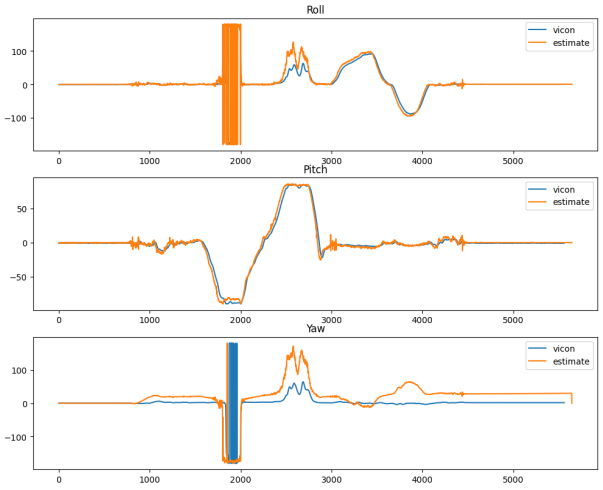

Fig. 2. Estimated orientation with VICON ground truth

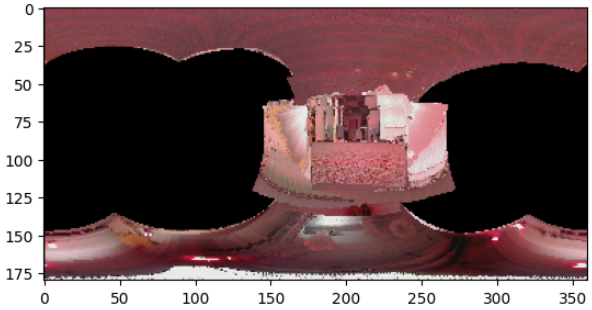

Fig. 3. Augmented image stitch based on estimated orientation

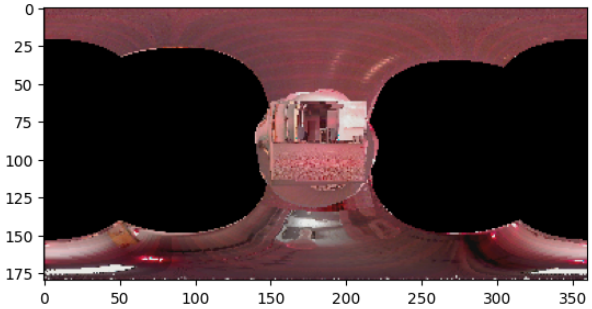

Fig. 4. Augmented image stitch based on VICON orientation

V. Test Result

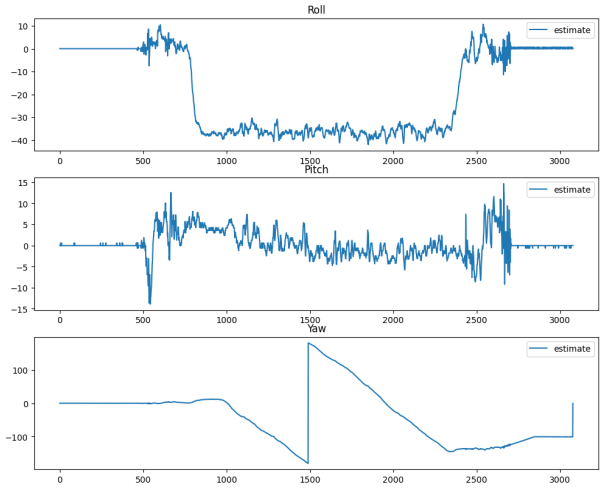

Fig. 5. test 10 estimated orientation

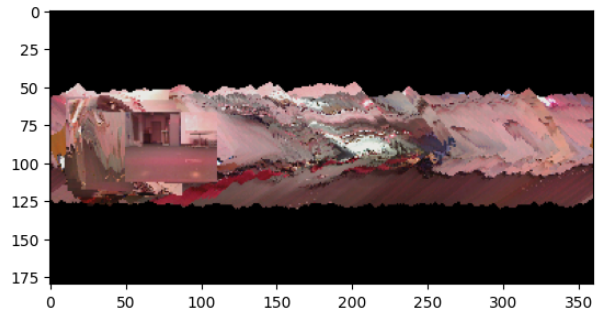

Fig. 6. test 10 augmented image

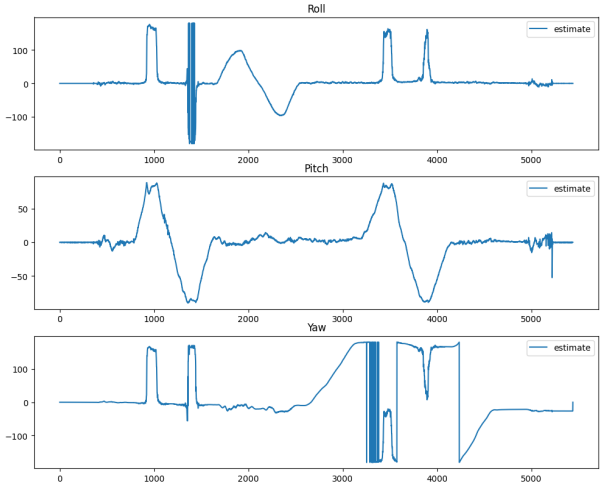

Fig. 7. test 11 estimated orientation

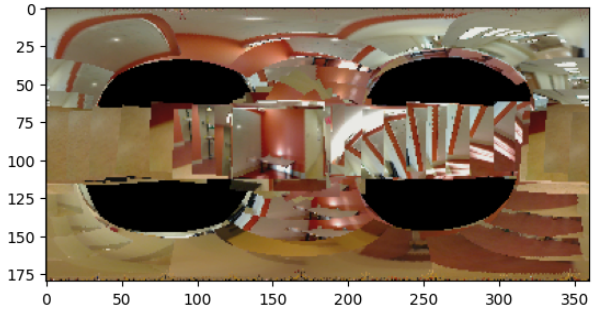

Fig. 8. test 11 augmented image

References

[1] Nikolay Atanasov, Sensing & Estimation in Robotics (Winter 2025), https://natanaso.github.io/ece276a/schedule.html.
[2] Matthew Brett, transform3d, https://matthew-brett.github.io/transforms3d/reference/transforms3d.euler.html.