

ESE 650 Learning in Robotics, Spring 2016
Project 2: Orientation Estimation with Kalman
Filters

Due: Thursday, Feb 11st
Yiren Lu

Introduction

For ESE 650 learning in robotics second project, we are provided raw data of the IMU's accelerometer's and gyroscope's data as well as their time stamps to estimate the 3D orientation of the device. And then, use the orientation information to generate a real-time panoramic image from camera.

In this project, I firstly converted the raw data from IMU into physical units, and then inferred device orientation in terms of rotation matrices separately from accelerometer and gyroscope. In order to gain robust performance, I implemented an unscented kalman filter (UKF) to combine the accelerometer and gyroscope data. The predicted results were compared with ground truth data from Vicon. Finally, using obtained orientation information, I projected recorded images onto a 3-D cylinder to generate panoramic images.

1 Data Preprocessing

1.1 Accelerometer

To convert the raw accelerometer data into physical units, the equation is as follows:

$$scale_factor = Vref / (1023 * sensitivity) \quad (1)$$

$$value = (raw - bias) * scale_factor \quad (2)$$

From the document of ADXL335, we know that $Vref = 3.3v$, $sensitivity = 330mV/g$, for the bias, I just simply took the mean of first 100 samples, which are typically stable. And then, correct the z-axis bias of the gravity by subtract the z-axis bias of

$$1 / (Vref / 1023 / sensitivity) \quad (3)$$

1.2 Gyroscope

Converting physical values from gyroscope:

$$value = 3300/1023 * PI/180/sensitivity \quad (4)$$

2 Rotation Mathematics

Rotation group $SO(3)$ has three degrees of freedom. The orientation of the camera is usually be represented in Euler Angles as *roll*, *pitch* and *yaw*. Detailed explanation and equations that transform between them to be found at [2] and [3]. In addition, *Quaternions* extend the complex number systems with the properties:

$$q = a1 + bi + cj + dk \quad (5)$$

$$ijk = i^2 = j^2 = k^2 = -1 \quad (6)$$

$$i = jk \quad (7)$$

$$k = ij \quad (8)$$

$$j = ik \quad (9)$$

where, q is a quaternion, a, b, c, d are real numbers, i, j, k are imaginary basis elements of the quaternions.

Unit Quaternion is a 4 dimensional sphere that could be used to represent orientation and rotation. I used Quaternions to represent the orientation and rotation tranformation heavily in this project.

2.1 Roll and Pitch from accelerometer

From accelerometer, we can estimate the gravity direction, from which, we could obtain roll and pitch angles as follows:

$$Roll = -arctan(Acc_y / \sqrt{Acc_x^2 + Acc_z^2}) \quad (10)$$

$$Pitch = arctan(Acc_x / \sqrt{Acc_y^2 + Acc_z^2}) \quad (11)$$

2.2 Orientation Estimation by Gyroscope

From gyroscope, we can get the angular velocity. By accumulating the angular velocity, we can estimate the orientation of the device at a certain time. The rotation transformation could be representation in *Quaternions*. We can formulate the quaternion rotation given angular velocity $\vec{\omega}$ and time duration Δt

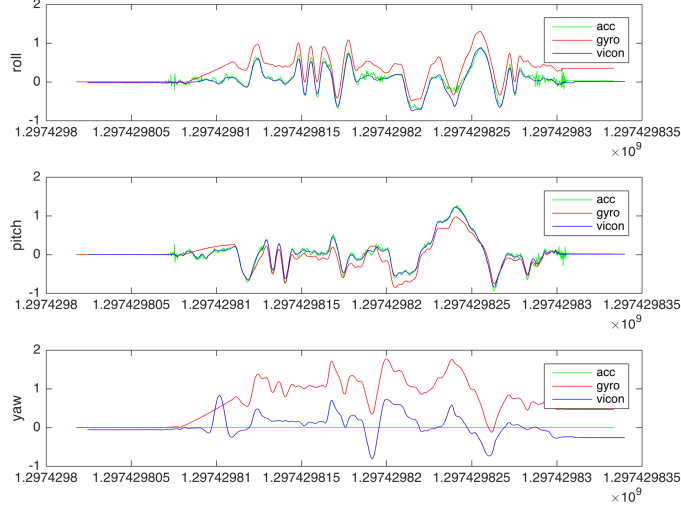


Figure 1: Naive Approach from Acc and Gyro's Estimation on dataset 4

from IMU as follows:

$$\vec{e} = \frac{\vec{\omega}}{||\vec{\omega}||} \quad (12)$$

$$\Delta\alpha = ||\vec{\omega}|| * \Delta t \quad (13)$$

$$\Delta q = [\cos(\Delta\alpha/2) \vec{e} \sin(\Delta\alpha/2)] \quad (14)$$

$$q_k + 1 = q_k \Delta q \quad (15)$$

Where, q_k is the previous rotation state, Δq is the rotation quaternion, and q_{k+1} is the transformed state. We can compute the corresponding rotation matrices from q_k .

3 Approaches

3.1 Naive Accelerometer and Gyro's Estimation

From the previous section we know that we could directly compute the orientation matrices from Accelerometer and Gyro data. In Figure 1 and 2, we can see that the accelerometers' estimation for roll and pitch is generally well. Gyro instead could catch yaw angle. However, driftings are seemed in gyro's predictions.

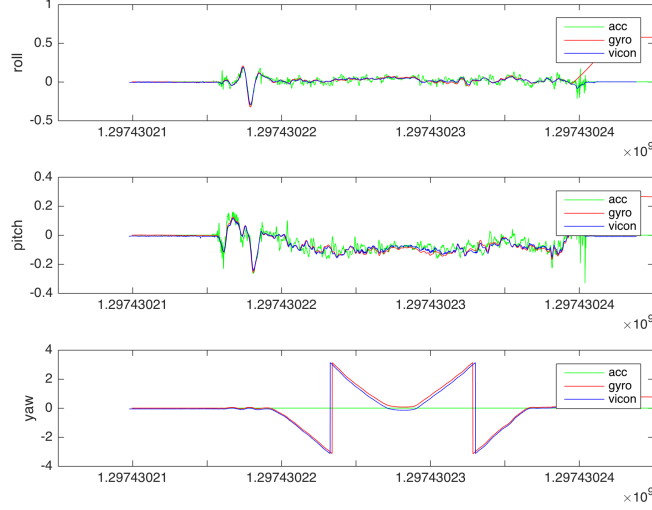


Figure 2: Naive Approach from Acc and Gyro's Estimation on dataset 8

3.2 Unscented Kalman Filter

Kalman Filter (KF) is widely applied in a variety of technologies. While, the basic KF could only serve linear system. Extend Kalman Filter (KF) resolved this issue by linearize the nonlinear transformation by Taylor series expansion. Unscented Kalman Filter (UKF) [4] skipped the *jacobian* operations in EKF by propagating the distribution through nonlinear transformation on a set of so-called *Sigma Points*. In this project, I implemented a UKF to gain more robust performance.

I implemented both a standard 7-state UKF and a 4-state UKF. The detailed algorithm for the 7-state UKF algorithm for orientation tracking refers to [1].

3.2.1 4-State UKF

The 4-state vector contains a single quaternion:

$$s_0 = [1, 0, 0, 0]^T \quad (16)$$

Process Model

The process model is simply:

$$s_{k+1} = (q_k q \Delta) \quad (17)$$

where, $q \Delta$ is generated from angular velocity and time duration as stated in 2.2.

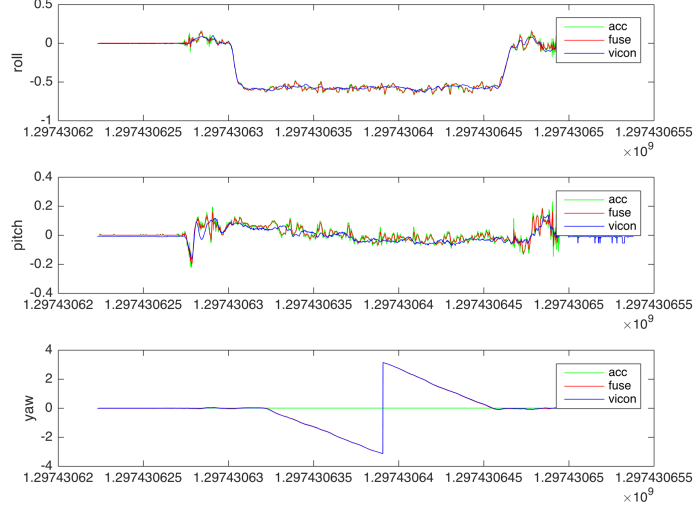


Figure 3: Fusion on dataset 8

Measurement Model

$$\vec{z}_{acc} = q_k g q_k^{-1} + \vec{v}_{acc} \quad (18)$$

where, $g = [0, 0, 0, 1]$ representing the gravity.

All the other parts of the algorithm are the same as 7-state UKF. In my implementation, I found that the 4-state UKF actually works better in most cases. Thus, the results being shown in this reports are all generated from 4-state UKF.

3.3 Fuse UKF with Gryo's

For my implementation, I find that the results my algorithm produces is mainly affected by the Accelerometer's prediction, which works poorly on *yaw* angle. Since the yaw prediction of Gyroscope works really well. I fused the *roll* and *pitch* prediction of UKF with Gyro's *yaw* angles. The results could be seen in Figure 3.

4 Image Stitching

For image stitching, I found an algorithm that project 3-D points into cylndral coordinate and then unwrap the cylinder into a panoramic image. Therefore, first assign z coordinates to the images, and then perform rotation to the image

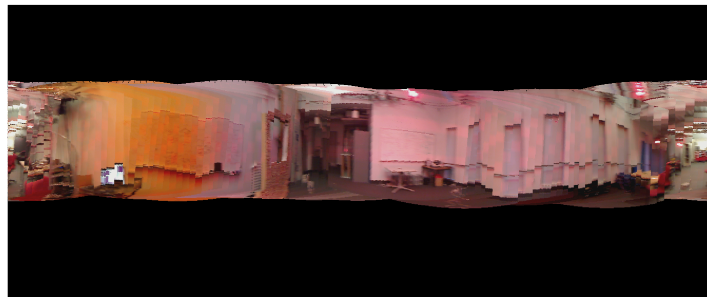


Figure 4: Panoramic Image of Dataset 2 and 8

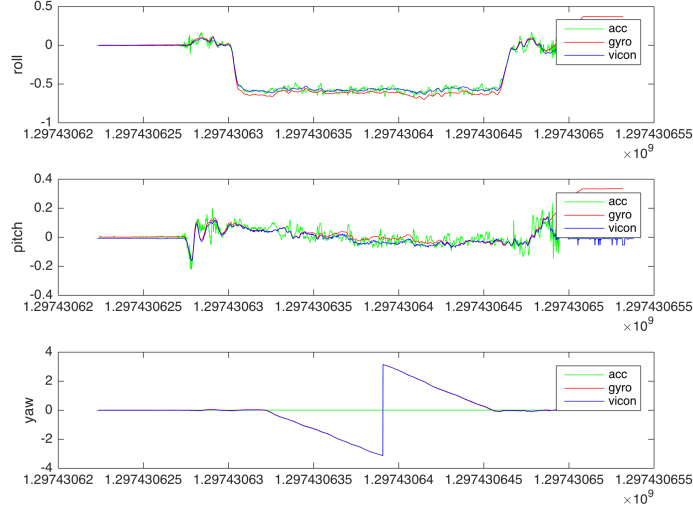


Figure 5: Naive approach on test set

using computed rotation matrices. After that perform the mapping and unwrapping. As shown in Figure 4. Due to time limitation, I only tried this single method.

5 Results and Discussion

Performing my algorithm on the test set yields the results in Figure 5 6 7 8

- *The initialization of noise covariance matrices* for process and measurement models are important. While, in this project I didnt use methods to parameterize the covariance matrices other than trial and error. However, smaller coveriances parameters means less uncertainty. We can use this to control the performance according to our belief of the robustness for process and measurement model.
- By *fusing sensors data*, we can obtain better estimation.

6 Acknowledgement

Thanks Dr. Lee for designing and preparing this project and thanks TAs for timely responses to our questions.

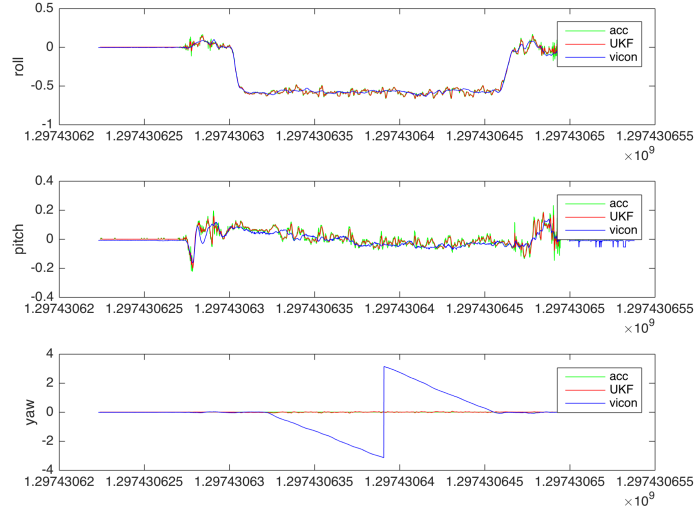


Figure 6: UKF on test set

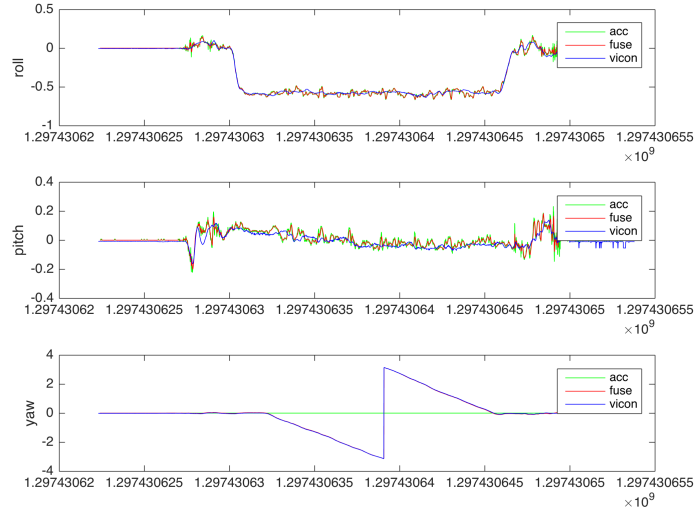


Figure 7: Fusion on test set



Figure 8: Panoramic image on test set

References

- [1] Edgar Kraft, *A quaternion-based unscented kalman filter for orientation tracking*. Information Fusion, 2003. Proceedings of the Sixth International Conference on Information Fusion, Vol. 1 (2003), pp. 47-54.
- [2] *Yaw, pitch, and roll rotations*, <http://planning.cs.uiuc.edu/node103.html>
- [3] *3D Transformations*, <http://planning.cs.uiuc.edu/node100.html>
- [4] Julier, Simon J., and Jeffrey K. Uhlmann. *New extension of the Kalman filter to nonlinear systems*. AeroSense'97. International Society for Optics and Photonics, 1997.