ECE 276A

Project 2

**Orientation Tracking**

Due Date: Nov 15, 2017

Professor: Nikolay Atanasov

Chuqiao Song

A53239614

**Introduction**

In Robotics, it is important to know where the Robot is, and its orientation respect to world fame, as one goal for robotics is to create a human like robot to help people in their daily life. knowing where the position with respect to world is a normal ability for human being. However, it is still a bit hard in robotics, and to let robot itself know the location, people have to use inertial measurement, GPS, etc, and some algorithms to estimate best positon and its orientation. Additionally, tracking is not only significant in robotics, but also in car movement, virtual reality device, AR, and phone, any device needing movement.

In this orientation tracking project, the quaternion computation is used, and the first step is to calibrate angular velocity, and linear acceleration from accelerometers and gyroscopes within IMU. Then by using UKF method, I can combine the measurement value and predicted value into my best estimate value to represent the position and orientation of the device. Finally, by using these estimated orientation data with given graphs, a panorama image can be generated.

**Problem Formulation**

Given the training set of IMU data which contains angular velocity of yaw, pitch, roll, and acceleration in XYZ, the goal of this project is to best estimate the location of the camera. Then by using estimated data with associated picture given by training set, the panorama image can be constructed.

**1) Data modification**: By using first few second in imu training set, with some average formulas, the bias of the accelerometer and gyroscope can be calculated.

**2) Quaternion equation construction**: write python code to implement basic quaternion calculation such as addition, multiplication, conjugation, inversion, log, and exponential function. Achieve the average quaternion function by using the algorithm given in class, with weight {αi}, collection of quaternion {qi}, and threshold 0.0001. In this project, the collection of quaternion come from corresponding sigma point.

**3) unscented Kalman filter**: UKF method is a good way to achieve the good estimation of the device’s location, and orientation by combing the data from motion model and observation model, and do the predication and update step.

**4) panorama**: estimate the orientation of the camera from imu data, and use the estimation with corresponding picture with associated time period to do graph stitching in spherical coordinate then unwarp them in 2D structure.

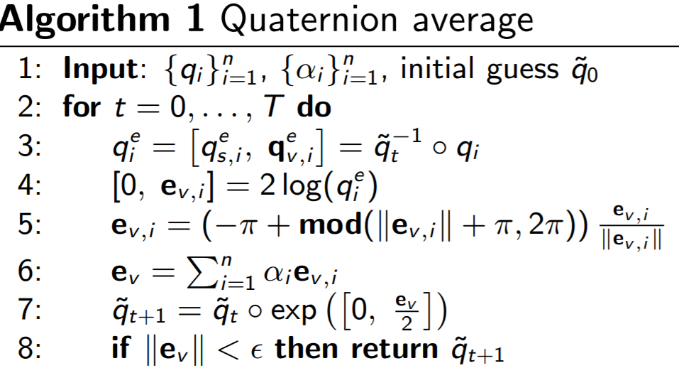
**Technical Approach**

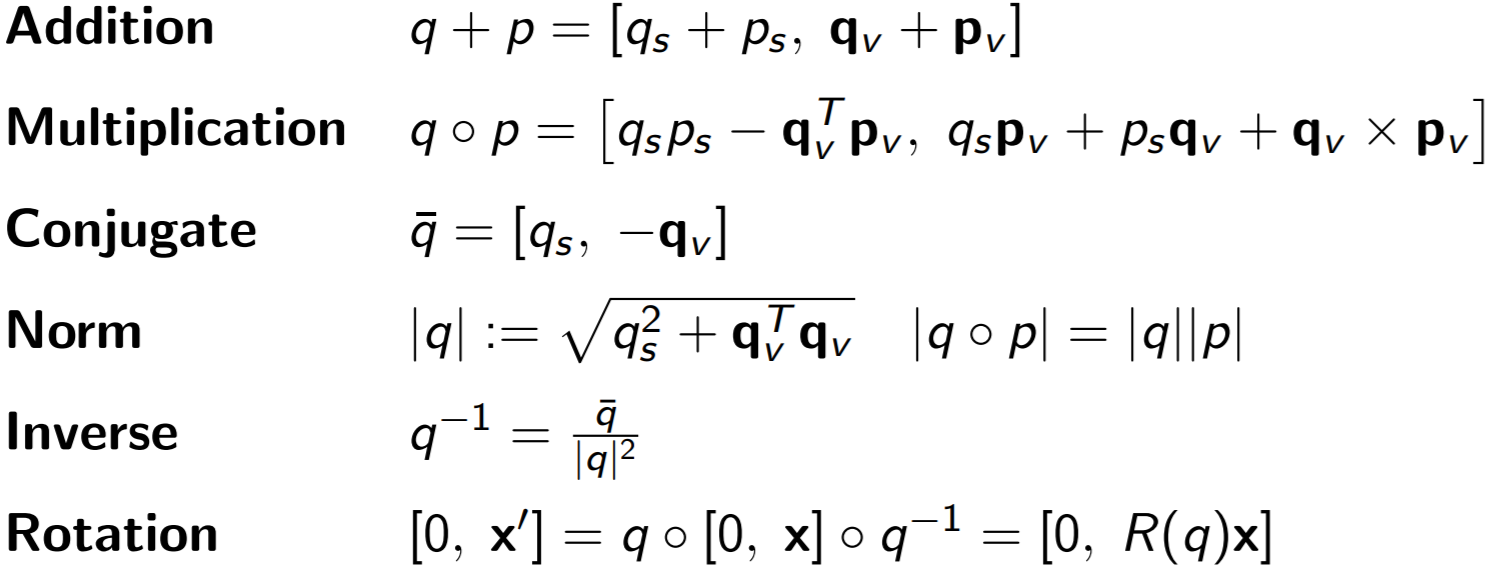
***A. Data modification***

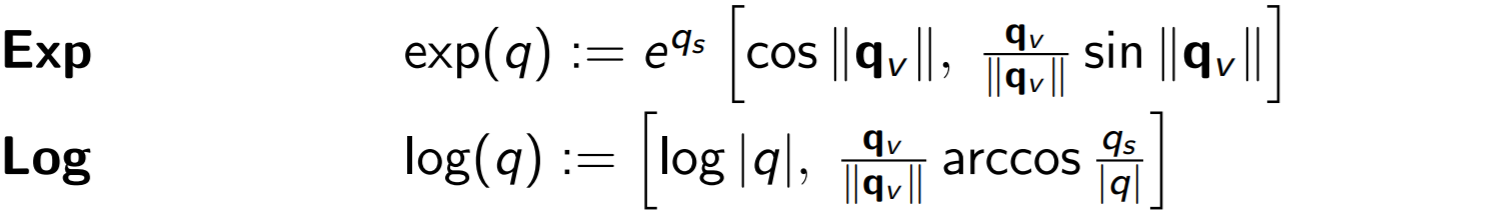
Extract the 6 dimensions’ raw data from IMU for acceleration and gyros and do the linear conversion. For acceleration conversion, the formula should be value = (raw - bias) \* scale\_factor, and scale\_factor = Vref / 1023 \* sensitivity, with Vref = 3300, sensitivity = 300. For gyros conversion, that is angular velocity, value = (raw - bias) \* scale\_factor, and scale\_factor = Vref / 1023 \* sensitivity/180\*pi, with Vref = 3300, sensitivity = 3.3. Since the first few second the camera is static, it is easy to achieve the bias for both accelerometer and gyroscope by averaging the value of first few second in IMU.

Also, the output data for estimation should be represented under Euler angle, the 3dtransformation package is used in this project to do rotation matrix to Euler angle transformation, and quaternion to Euler angle transformation.

**B. Quaternion equation construction**

 Implement basic quaternion function by using python code, where the function is given in class lecture. Especially, the under log, exponential function, the norm of qv = 0 should be precisely considered to avoid error happening.





**C. Unscented Kalman filter, orientation tracking**

In this orientation tracking part, there are two models motion model, observation model, and two steps prediction and update step. First to begin the orientation tracking loop, it is important to set the initial values with quaternion value q0 = ut [1, 0, 0, 0], and covariance Cov\_0 = 0.0001I. With the initial value set at beginning, the corresponding 7 sigma points can be generated by this formula,  where matrix square root is achieved by Cholesky Decomposition and Q is noise. Since in this project, quaternion is used to representation orientation and it has 4 dimension with 3 freedom, n = 3. In the prediction step, these 7 sigma points above corresponding to 7 quaternions are rotated error, where qt+1 are 7 quaternions predicted at time t+1.

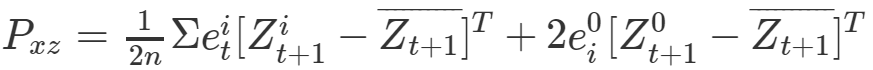
Then, use these collection of 7 quaternions with quaternion averaging function, the predicted mean is generated. Additionally, the error rotation vector evi should be calculated at meanwhile. With the calculated error rotation vector, the covariance is generated for update model, where 1/2n, and 2 are the weights for covariance in prediction step.

.

In the update step, the first thing to do is to find measurement value with corresponding qt+1, where g is the gravity of the world [0,0,1].

Then by using these 7 measurement values, the mean of measurement can be found, where i are 1, 2, 3, 4, 5, 6, and n =3. After finishing calculation of Zt+1 mean, the measurement error covariance, final covariance of measurement, and cross covariance can be generated by using

.





Then the Kalman Gain can be calculated, and the this gain the estimated covariance and mean at t+1 can also be generated.

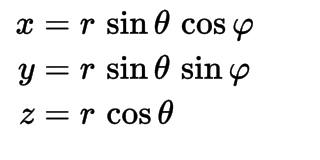
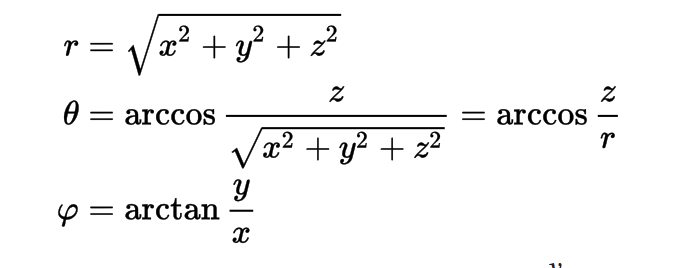
where Z is true data from accelerometer.

Now, the one iteration is finished, and to achieve the orientation and location of a camera in continuous time, it must do more iteration to estimate each position and orientation of the camera.

**D. panorama**

With the estimated orientation data from IMU and pictures at corresponding, the panorama image can be generated by following step. First, assonate each pixel from original graph to angular frame under sphere coordinate, and that is find longitude and latitude of each pixel within the horizontal 60 degrees, and vertical 45 degrees’ field of view. Second, convert this spherical coordinate (longitude, latitude, depth=1) to Cartesian coordinate, and rotated it by imu orientation data into world frame. Finally converted it back to spherical coordinate, and unwarp it to cylinder coordinate to 2d graph.

Cart2Spher Spher2Cart



To stitching all the graphs into panorama, the above steps should be iterated many times, and the final graph in my project is 4\*240 pixels height, and 6 \*320 pixels length.

**Result and Discussion**

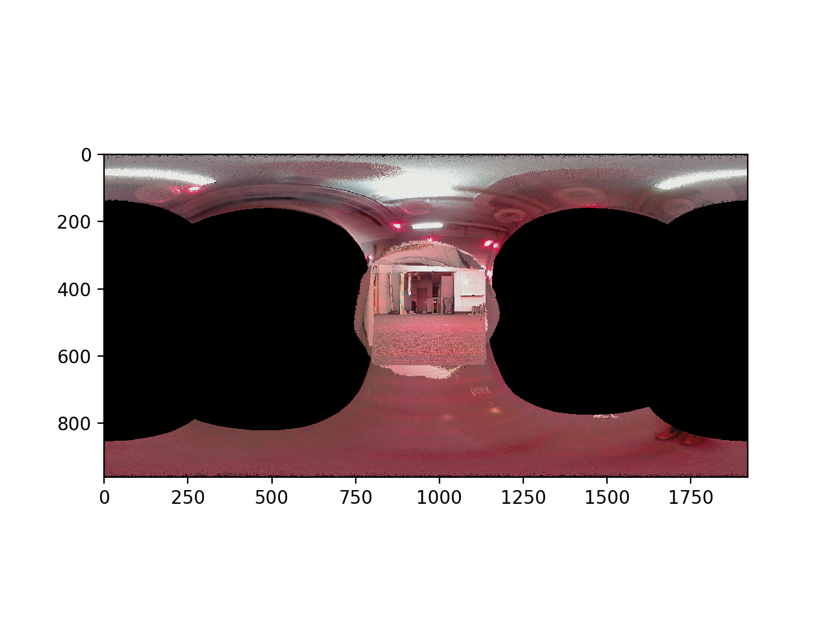
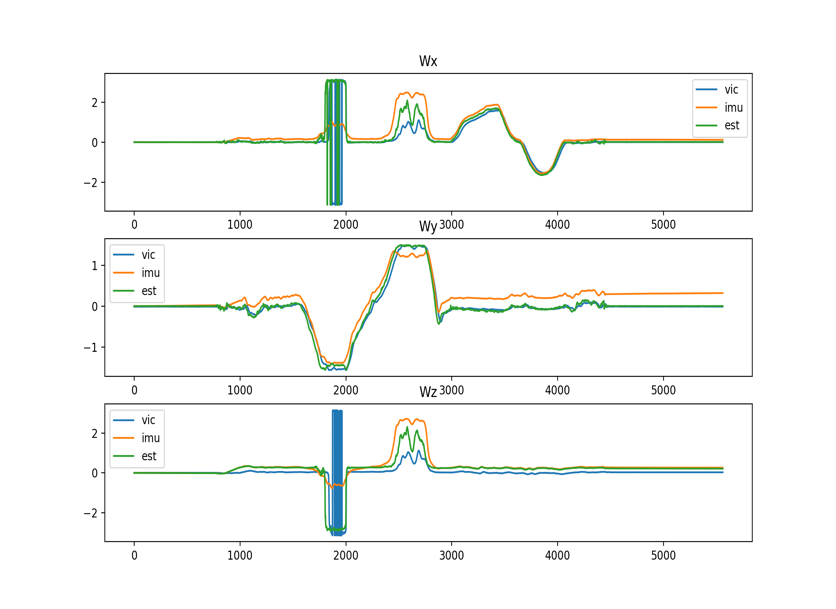
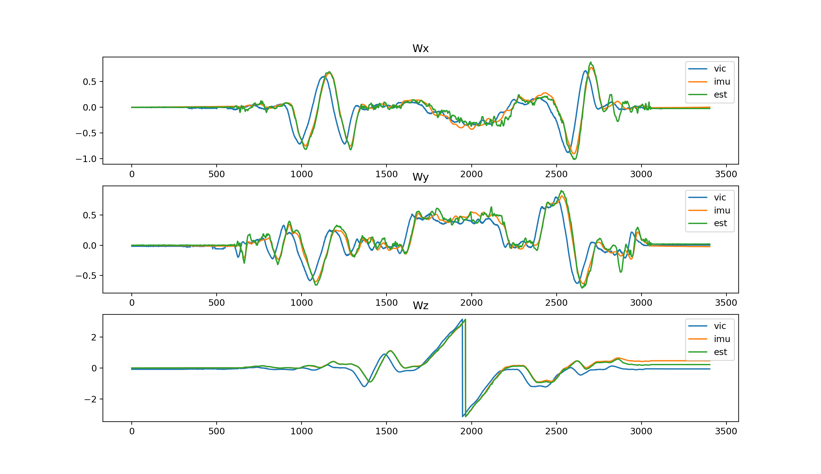
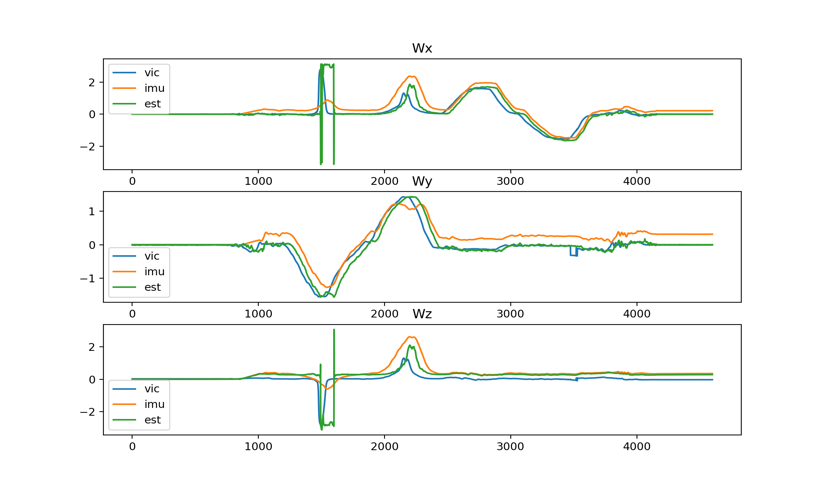
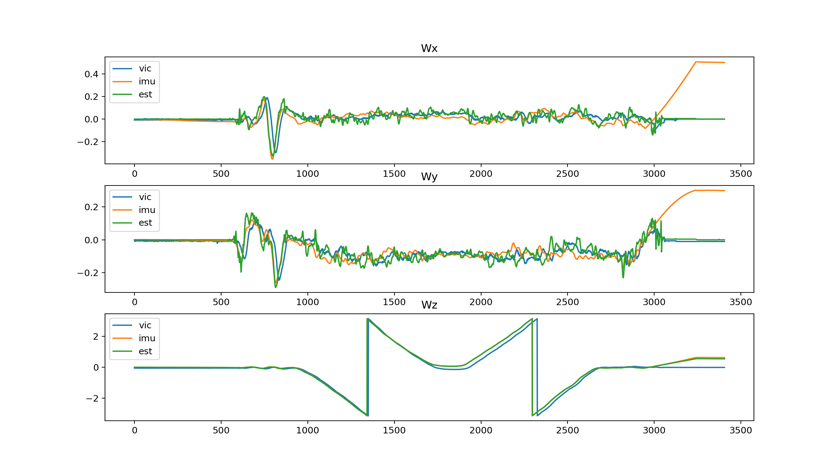
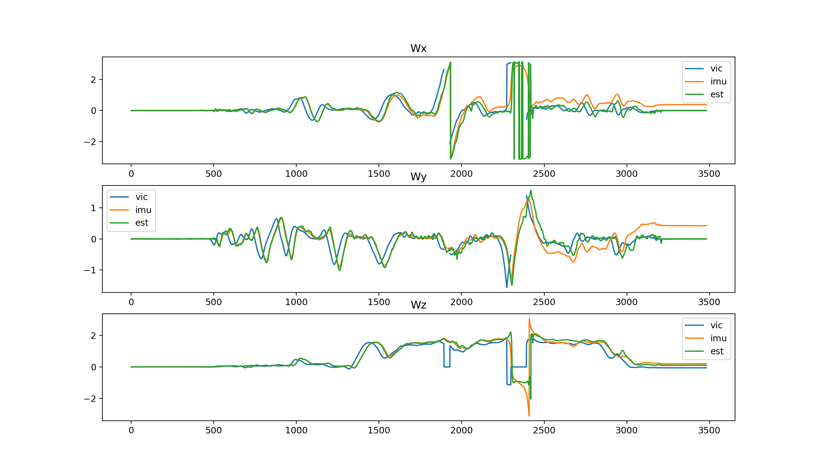


Figure1: training set 1

Both imu data, and ground truth data have some error, thus by using UKF the orientation can be best estimated. As show in left Fig1, the estimated data is in between the imu and ground truth data. The right side of Fig1 is my Panorama generated by using ground truth data, and it can clear to see the door and the roof in the picture. However, there are still a lot of method can be used to increase the clarity such as some blur technic.



Figure2: training set for 2, 3, 7, 8 with clockwise order

In most training set testing, the UKF can do best estimation, and they are kind smooth in between the ground truth data and IMU data. To some extent, it is much close to the ground truth data. In figure 2, the error between imu and ground truth data is small, the estimate line and other two lines are aligned close together. However, even if, there are much difference between imu data and ground truth data, the UKF method can do best job to align the estimate line close to the ground truth line like in figure3, especially for train-set 9.

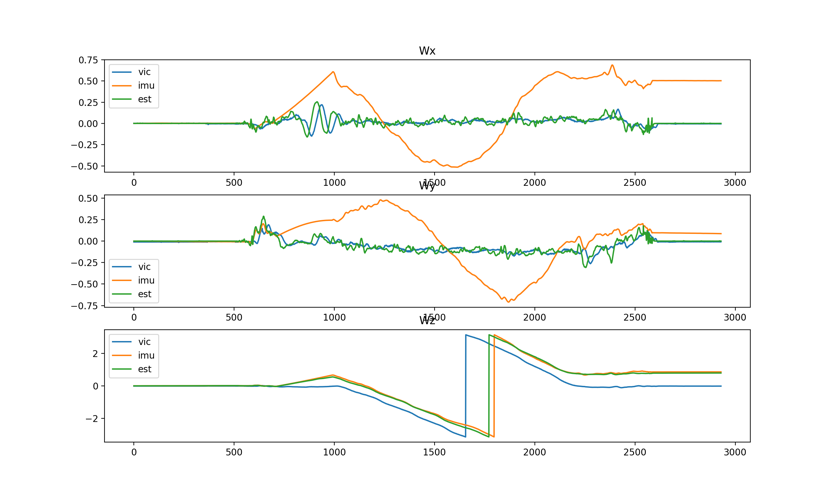
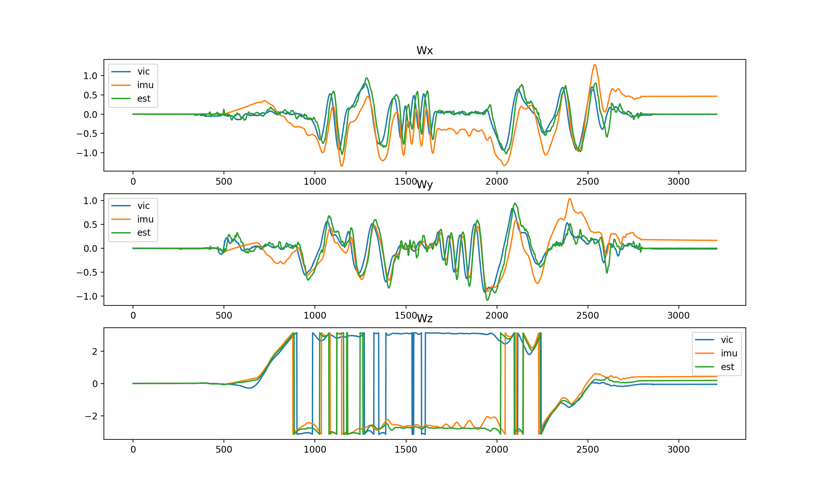
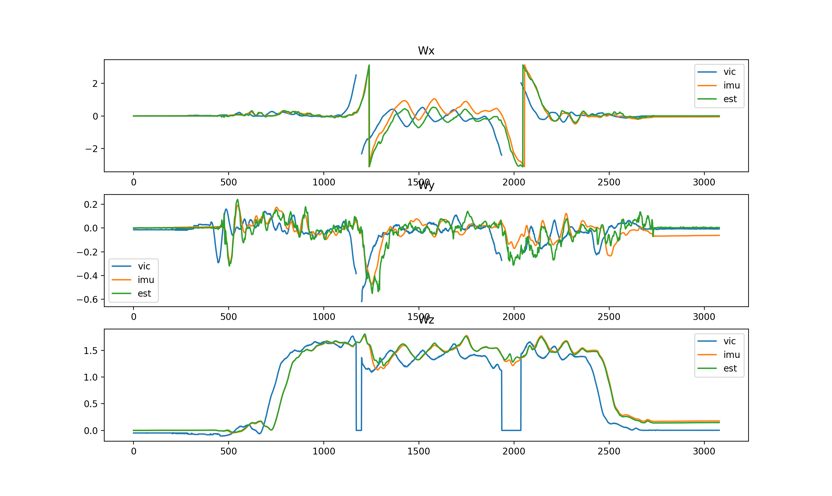
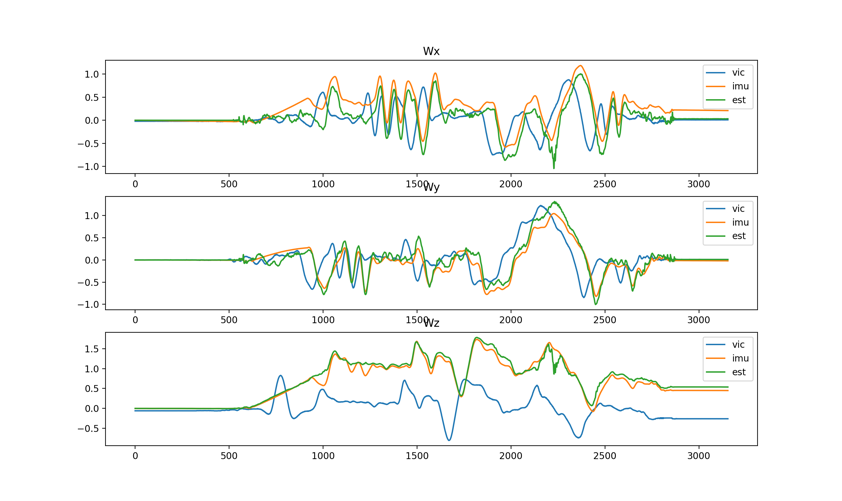


Figure3: training set for 4, 5, 6, 9 with clockwise order.

Even if UKF did good estimation, there are still some weird result appeared in training set 4 and 9. The shape looks similar for estimated line and ground truth line, but some bias exists between them, and the estimated line is much aligned to imu data.

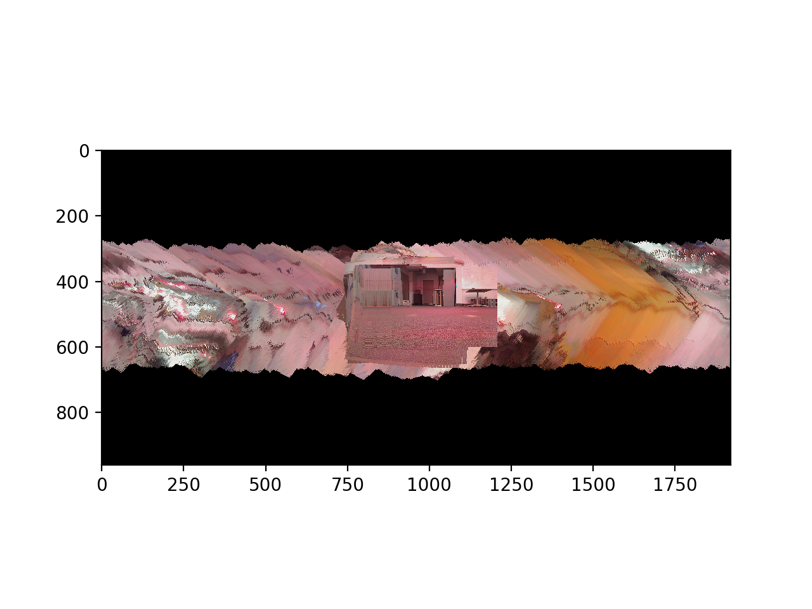
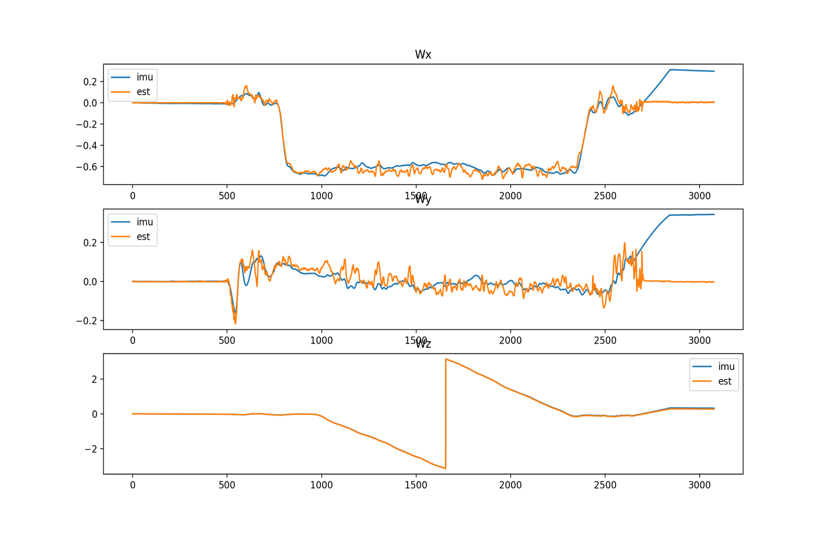
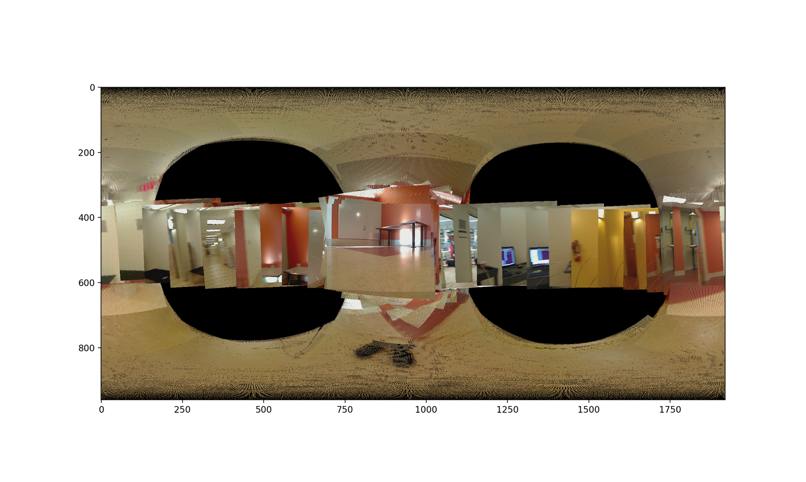
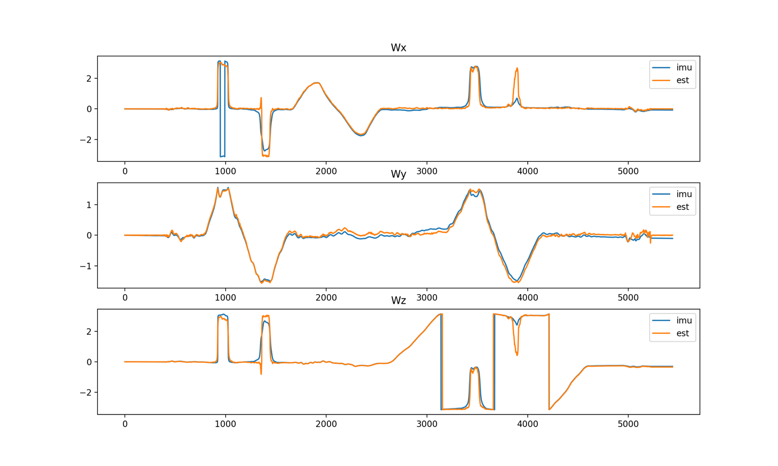
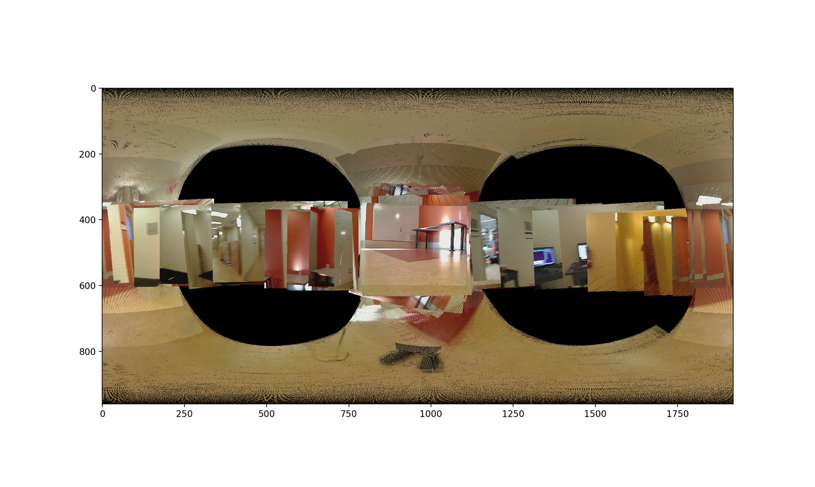
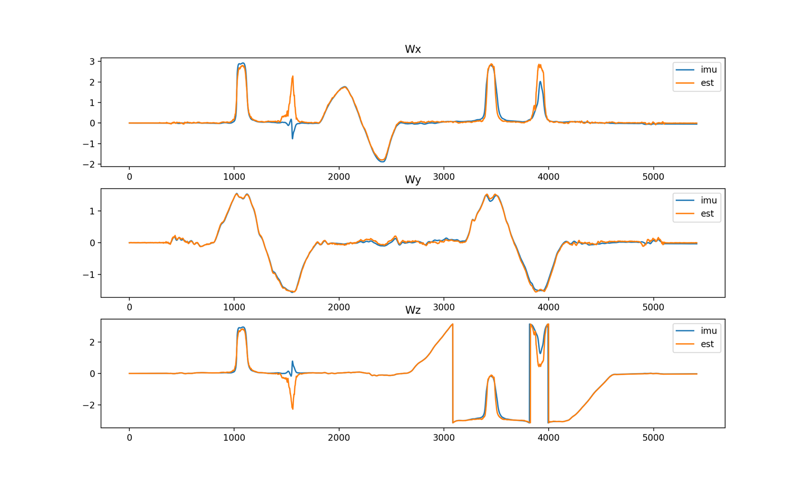


Figure4: Test set 11

It seems like the UKF did good estimation for test set 11, but Panorama image is not clear generated by using estimated data. It may be because of the estimated line is too much fuzzy or noisy, which causes the stitching step waggling. In Figure5, the estimation and Panorama result looks much better than in Figur4, and the Panorama image looks clear. However, there may be another method can be used to improve the clarity of the image, since the image is continuous enough, such technic like Gaussian blur, or we can improve our estimation step.





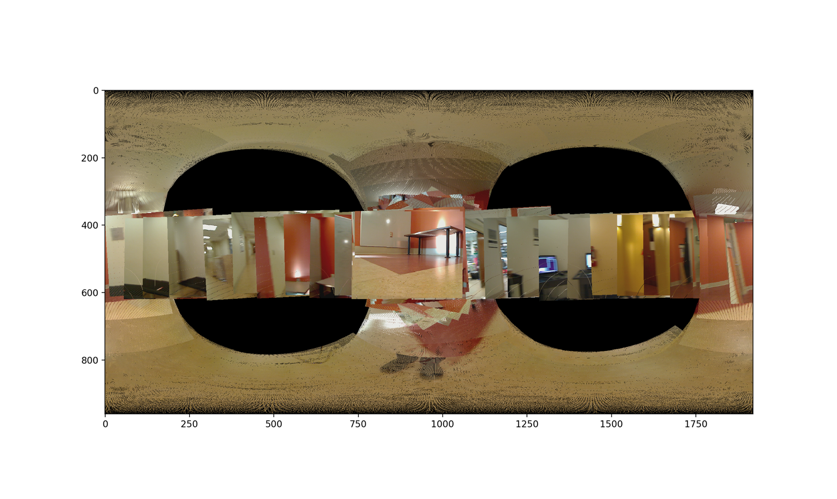
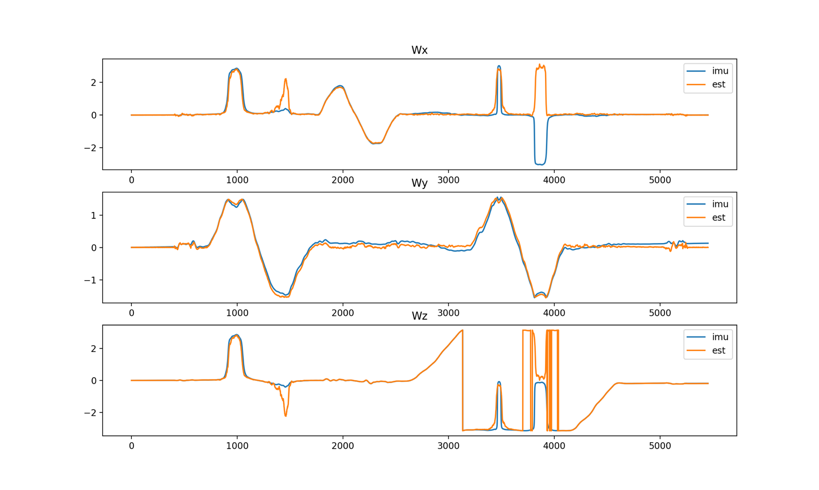


Figure5: Test set for 11, 12, 13 from top to bottom