Orienation Tracking based Panorama Stitching using Unscented Kalman Filter

Nitin J. Sanket
School of Engineering and
Applied Science
University of Pennsylvania
Email: nitinsan@seas.upenn.edu

***Abstract*—This project presents an approach for orientation tracking using different filters like the Complementary Filter (CF), Kalman Filter (KF) and Unscented Kalman Filter (UKF). The tracked orientation/attitde is used to stitch a panorama in cylindrical co-ordinates assuming no translation. An error metric for euler angles is defined and the error for the results are presented. UKF outperforms CF and KF as UKF as it is the closest to Vicon results and is the least susceptible to drift.**

I. Problem Statement

The aim of the project was to estimate the bias and scale parameters of a IMU. These parameters are then used to correct the IMU measurements and run different filters to track the attitude. This attitude is then used to construct a cylindrical panorama. The algorithm was tested on 1 unseen set and 10 training sets.

II. Estimating Bias

Let the Bias of the accelerometer be B and the Scale factor be S. Let the accelerometer values for one axis be x. g' is the gravity vector in IMU frame (B). The following equations are used to calculate Bias and Scale (Finds the least squares fit). (here † represents pseudo-inverse). Here R_B^W comes from the synced Vicon data.

$$Sx + B = g'$$
$$g' = \left(R_B^W\right)^T g$$
$$g = [0,0,1]'$$
$$\begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix} \begin{bmatrix} S \\ B \end{bmatrix} = \begin{bmatrix} {g_1}' \\ {g_2}' \\ \vdots \\ {g_n}' \end{bmatrix}$$
$$\begin{bmatrix} S \\ B \end{bmatrix} = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix}^{\dagger} \begin{bmatrix} {g_1}' \\ {g_2}' \\ \vdots \\ {g_n}' \end{bmatrix}$$

This bias for the gyroscope was computed using the mean of first k values in the dataset, as the dataset was assumed to start from rest. k was chosen as 200. The bias B for one axis of the gyroscope measurements x is calculated as

$$B = \frac{1}{k}\sum_i x_i$$

The scale factor S for the gyroscope was obtained from the manufacturer's datasheet as

$$\mathrm{S} = \frac{3300}{1023}\frac{\pi}{180}\frac{1}{3.333333}$$

III. Complementary Filter

The idea behind the complementary filter [1] is that attitude is computed separately from both accelerometer and gyroscope. The IMU body axis is defined as Z up, X forward and Y left (looking from IMU's point of view). The accelerometer is symmetric with respect to Z axis hence it doesn't provide any information about Yaw (ψ) angle. The other two angles [2] are calculated as follows:

$$Roll, \phi = \tan^{-1}\left(\frac{a_y}{\sqrt{{a_x}^2 + {a_z}^2}}\right)$$

$$Pitch, \theta = \tan^{-1}\left(\frac{-a_x}{\sqrt{{a_y}^2 + {a_z}^2}}\right)$$

However, generally the IMU is never completely vertical, and hence the Yaw can be retrieved as

$$Yaw, \psi = \tan^{-1}\left(\frac{\sqrt{{a_x}^2 + {a_y}^2}}{a_z}\right)$$

The gyroscope values are integrated to obtain the angles. The integration is performed using quaternions and then the euler angles are computed from the quaternions. The update quaternion is calculated as below:

$$\alpha_\Delta = |\vec{\omega}_k|\,\Delta t$$
$$\vec{e}_\Delta = \frac{\vec{\omega}_k}{|\vec{\omega}_k|}$$
$$q_\Delta = \left(\cos\left(\frac{\alpha_\Delta}{2}\right), \vec{e}_\Delta \sin\left(\frac{\alpha_\Delta}{2}\right)\right)$$

To calculate current state quaternion the following equations are used

$$\alpha = |\vec{\omega}_k|$$
$$\vec{e} = \frac{\vec{\omega}_k}{|\vec{\omega}_k|}$$
$$q_k = \left(\cos\left(\frac{\alpha}{2}\right), \vec{e}\sin\left(\frac{\alpha}{2}\right)\right)$$

Now, the new state quaternion is described as (k is the current state and $k+1$ is the next state),

$$q_{k+1} = q_k q_\Delta$$

The euler angles are calculated from the quaternion as,

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \tan^{-1}\left(\frac{2(q_0q_1+q_2q_3)}{1-2(q_1^2+q_2^2)}\right) \\ \sin^{-1}\left(2(q_0q_2 - q_3q_1)\right) \\ \tan^{-1}\left(\frac{2(q_0q_3+q_1q_2)}{1-2(q_2^2+q_3^2)}\right) \end{bmatrix}$$

The angle measurements are blended as follows in the complementary filter,

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}_{CF} = \begin{bmatrix} \alpha & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & \gamma \end{bmatrix} \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}_{acc} + \begin{bmatrix} 1-\alpha & 0 & 0 \\ 0 & 1-\beta & 0 \\ 0 & 0 & 1-\gamma \end{bmatrix} \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}_{gyro}$$

Here $\alpha, \beta$and$, \gamma$ are the mixing parameters. α was chosen as 0.75, β was chosen as 0.75 and γ was chosen as 1.0. Effectively the gyro measurements are high pass filtered to remove drift and accelerometer measurements are low pass filtered to remove noise.

IV. Kalman Filter

The issue with a complementary filter is that the mixing coefficient are constants and hence does not perform well in the long run due to huge uncompensated drift. This is implemented in a simple linear Kalman Filter. However, due to some implementation bug, the Kalman filter has worse results than the complementary filter (this can also be because of not accounting for angle flips in euler angles and non usage of quaternions). The equations for the Kalman Filter [3] are discussed next.

A. Process Update

The prior on the state is given by,

$$p(x_0) \sim \mathcal{N}(\mu_0, \Sigma_0)$$

The transition update concept is,

$$x_k = A_k x_{k-1} + \eta_k$$

$$\eta_k \sim \mathcal{N}(0, Q_k)$$

Here, A_k is the process model. The actual transition update equations for mean and covariance are,

$$\bar{\mu}_k = A_k \mu_{k-1}$$

$$\bar{A}_k \Sigma_{k-1} A_k^T + Q_k$$

Here, the noise grows due to added uncertainty due to process model. This is the MLE estimate. Here Q_k is the process noise covariance matrix.

B. Measurement Update

The measurement update concept is,

$$z_k = C_k x_k + \nu_k$$

$$\nu_k \sim \mathcal{N}(0, R_k)$$

The actual measurement update equations for mean and covariance are,

$$\mu_k = \bar{\mu}_k + K_k(z_k - C_k \bar{\mu}_k)$$

Here, z_k is the measurement and C_k is the measurement model.

$$\Sigma_k = \bar{\Sigma}_k - K_k C_k$$

Where K_k is the kalman gain and is calculated as,

$$K_k = \bar{\Sigma}_k C_k^T (C_k \bar{\Sigma}_k C_k^T + R_k)^{-1}$$

Here, the noise drops due reduction in uncertainty from observation. This is the MAP estimate.

C. State, Process Model and Measurement Model Matrices

The State was defined as,

$$x = \begin{bmatrix} \phi \\ \theta \\ \psi \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

The process model matrix A_k is given by,

$$A_k = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

which is used as $x_k = A_k x_{k-1}$. The observation model matrix is given by,

$$C_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The covariance was initialized to $0.01 \times I_{6\times6}$, $Q_k = diag([0.01 * ones(1,3), 0.1 * ones(1,3)])$ and $R_k = diag([0.01 * ones(1,3), 0.1 * ones(1,3)])$.

V. Unscented Kalman Filter

A. Explanation of the Filter

The idea behind Unscented Kalman Filter [4] is that it respects the non-linearity in the process model. The state is defined as

$$x = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

where $[q_0, q_1, q_2, q_3]^T$ represents a unit quaternion with ${q_0}^2 + {q_1}^2 + {q_2}^2 + {q_3}^2 = 1$ hence represents only 3 degrees of freedom, not 4.

Initialization:
Initialize P_0 as the initial covariance matrix of size 6×6, state is unit quaternion and zero angular velocities, R is the measurement noise and Q is the process noise.
Generation of Sigma Points:
The square root matrix is found using,

$$S = \sqrt{(P_{k-1} + Q)}$$

Here Q is the process model covariance. The disturbance (noise) of the sigma points is calculated using [5],

$$\mathcal{W}_{i,i+n} = \text{columns}(\pm\sqrt{n}S)$$

here n is chosen instead of $2n$ as it was more numerically stable and converged faster. here n is the number of independent variables in the state (6 in this case). The Sigma Points are computed using,

$$\mathcal{X}_i = \hat{x}_{k-1} + \mathcal{W}_i$$

$$\mathcal{X}_i = \begin{bmatrix} q_{k-1}q_{\mathcal{W}} \\ \vec{\omega}_k + \vec{\omega}_{\mathcal{W}} \end{bmatrix}$$

where $\vec{\omega}_{\mathcal{W}}$ is the omega part of $\mathcal{W}$ and $q_{\mathcal{W}}$ is the quaternion part of $\mathcal{W}$.
Process Model:
The process model assumes that the angular velocity remains constant during the time interval Δt.

$$\omega_k = \omega_{k-1}$$

$$q_\Delta = \left[\cos\left(\frac{|\vec{\omega}_{k-1}|\,\Delta t}{2}\right), \frac{\vec{\omega}_{k-1}}{|\vec{\omega}_{k-1}|}\sin\left(\frac{|\vec{\omega}_{k-1}|\,\Delta t}{2}\right)\right]$$

The updated sigma points are computed using,

$$\mathcal{Y}_i = A(\mathcal{X}_i, 0) = \begin{bmatrix} q_{k-1}q_{\mathcal{W}}q_\Delta \\ \vec{\omega}_{k-1} + \vec{\omega}_{\mathcal{W}} \end{bmatrix}$$

Calculate Mean of Sigma Points:
Now, use Intrinsic Gradient Descent to find the mean quaternion.

Data: $\mathcal{Y}$
Result: $\hat{\bar{x}}_k$
Initialize $\bar{q}$ as $\mathcal{X}_1$ **while** *t¡MaxIter or* $|e|$ *≤Thld* **do**
  for $\forall i$ **do**
    $\vec{e}_i = q_i\bar{q_t}^{-1}$
  end
  Compute mean using,

$$\vec{e} = \frac{1}{2n}\sum_{i=1}^{2n}\vec{e}_i$$

$$\bar{q}_{t+1} = e\bar{q}_t$$

end

$$\bar{\omega} = \frac{1}{2n}\sum_{i=1}^{2n}\omega_i$$

Algorithm 1: Intrinsic Gradient Descent

Update model covariance:

$$\bar{P}_k = \frac{1}{2n}\sum_{i=1}^{2n}{\mathcal{W}_i}'{\mathcal{W}_i}'^T$$

Where $\mathcal{W}'$ is the mean centered $\mathcal{W}$.

$${\mathcal{W}_i}' = \begin{bmatrix} q_i\bar{q}^{-1} \\ \vec{\omega}_i - \bar{\vec{\omega}} \end{bmatrix}$$

Measurement Model Update:
Now, compute the measurement updated transformed sigma points,

$$\mathcal{Z}_i = \begin{bmatrix} {q_i}^{-1}gq_i \\ \vec{\omega}_k \end{bmatrix}$$

where g is the gravity vector. Now, compute $\bar{z}_k$ = mean$\mathcal{Z}_i$.
Compute measurement model covariances:
The covariance corresponding to the measurement model update is computed as,

$$P_{zz} = \frac{1}{2n}\sum_{i=1}^{2n}\phi_i{\phi_i}^T$$

here,

$$\phi_i = \begin{bmatrix} q_i\bar{q}^{-1} \\ \vec{\omega}_i - \bar{\vec{\omega}} \end{bmatrix}$$

The quaternions and omegas above correspond to the ones in $\mathcal{Z}$.

The innovation term is given by:

$$\nu_k = z_k - \bar{z}_k$$

Here, z_k is the observation, i.e., stacked accelerometer and gyroscope readings.

The innovation covariance is calculated using,

$$P_{\nu\nu} = P_{zz} + R$$

Here R is the measurement model covariance.

The cross-covariance is calculated as,

$$P_{xz} = \frac{1}{2n}\sum_{i=1}^{2n}\mathcal{W}_i{\phi_i}^T$$

Update Kalman Gain, State Covariance and State: The kalman gain is calculated as,

$$K_k = P_{xz}P_{\nu\nu}^{-1}$$

Update state as,

$$\hat{x}_k = \hat{\hat{x}}_k + K_k\nu_k$$

Update State Covariance as,

$$P_k = \bar{P}_k - K_k P_{\nu\nu} K_k^T$$

B. Covariance Changes throughout UKF

A visualization of covariance changes though the UKF is depicted in this subsection. Refer to Fig. 1, where we see that the initial covariance of the state is large (resembling a uniform distribution). The 3-axis plot shows the orientation vector as a point in 3D space and the ellipsoid shows the covariance of the quaternion part of the state vector. In Fig. 2, we see that the initial sigma points have almost same covariance and they look similar to the initial state covariance. The process update on sigma increases the covariance and hence uncertainty as shown in Fig. 3. The measurement covariance is shown in Fig. 4 and Final updated state covariance is shown in Fig. 5.

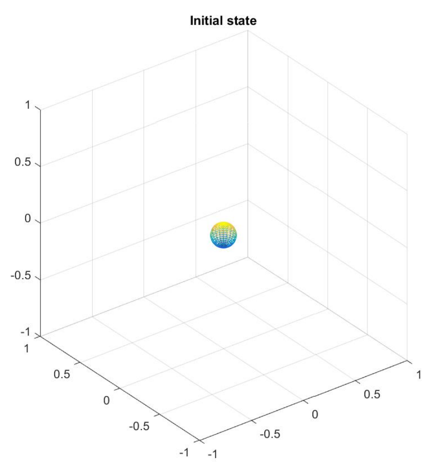

Fig. 1. Initial Covariance.

C. Observations during tuning the UKF Parameters

There are mainly 4 parameters which influence UKF output. R_a corresponds to noise in observation model corresponding to accelerometer values, R_g corresponds to noise in observation model corresponding to gyroscope values. Also, Q_a and Q_g correspond to noise in process model. UKF was very sensitive to Q_a, if Q_a was below 79 the UKF was very unstable. By varying Q_g, there was a lag introduced mainly in Yaw when the UKF was badly tuned. R_a had little influence

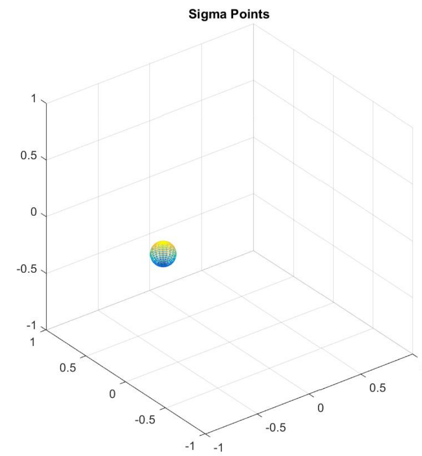

Fig. 2. Covariance of Sigma Points.

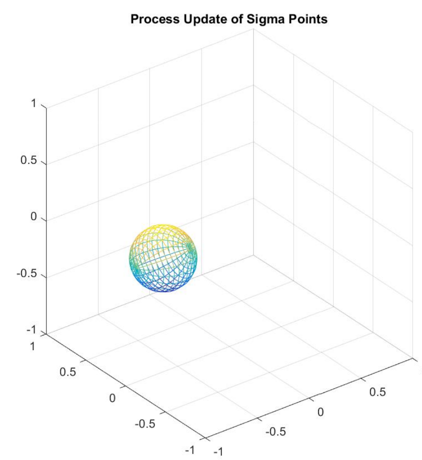

Fig. 3. Covariance after Process Update.

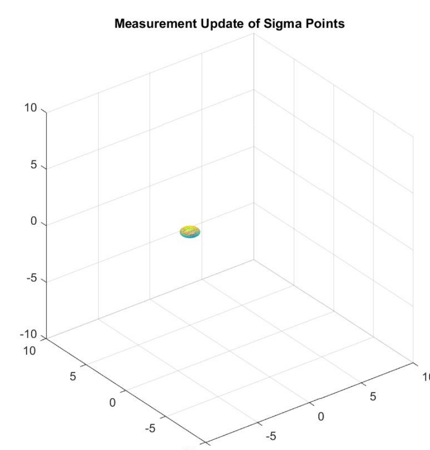

Fig. 4. Covariance of Measurement Update.

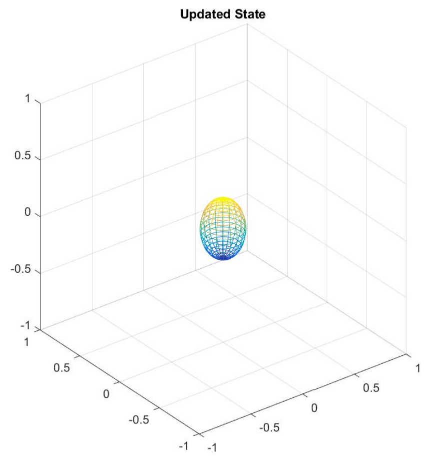

Fig. 5. Updated State Covariance.

on yaw. R_g influenced how much the values drift and jump around. Keeping this value to 0.01 made the trick. The final UKF gains used were a tradeoff between lag and error.

VI. Image Stitching

A. Homography Based Image Stitching

In this case, the canvas as assumed as multiple plans i.e., a homographic transformation was sufficient to map the image. The camera matrix is given by,

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

where f_x and f_y are the focal lengths in x and y respectively. c_x and c_y are the image center co-ordinates. A value of 240 for f_y and 200 for f_x was used which was found by eyeballing the output to make it look "good". A sample output for train set 1 is shown in Fig. 6. The projection equation given below is used for projecting the pixels into world canvas.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K\left[R|0\right] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

The translation is assumed to be zero.

Calculate tilt as ϕ, rotate the image by $-\phi$. The equations used are as follows:

$$\Delta y = f_y \text{tan}(\theta)$$

$$\Delta x = f_x \text{tan}(\psi)$$

Now, transform the co-ordinates,

$$\hat{y} = x + \Delta x - CanvasCenter_x$$

$$\hat{x} = y + \Delta y - CanvasCenter_y$$

As we can see from Fig. 6, the projection gets worse and drifts as the angle increases. To make this projection better cylindrical projection is used.

B. Cylindrical Projection Based Image Stitching

The world co-ordinates of the point in 3D space from pixel co-ordinates is calculated using [6] [7],

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = R \begin{bmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ f \end{bmatrix}$$

Here, $f = \sqrt{{f_x}^2 + {f_y}^2}$, f_x and f_y are same as the ones in K matrix. Also, here u, v are already mean centered.

For cylindrical projection, the following equations are used,

$$\text{Angle}, \theta = \text{tan}(\frac{Y}{X})$$

$$\text{Height}, h = \frac{Z}{\sqrt{X^2 + Y^2}}$$

The transformed pixel co-ordinates are,

$$\hat{x} = -f\theta + CanvasCenter_x$$

$$\hat{y} = fh + CanvasCenter_y$$

A sample output for train set 1 is shown in Fig. 7.

To interpolate the pixels better, a simple weighted average of the non-zero pixels was performed according to the equation given below,

$$p' = \alpha p_c + (1 - \alpha)\, p_i$$

where p' is the new pixel value and p_c is the current canvas pixel value and p_i is the value from the image in current frame. A sample output with $\alpha = 0.75$ is shown in Fig. 8. Clearly, the image is smoother with lesser holes, but it it also more blurred.

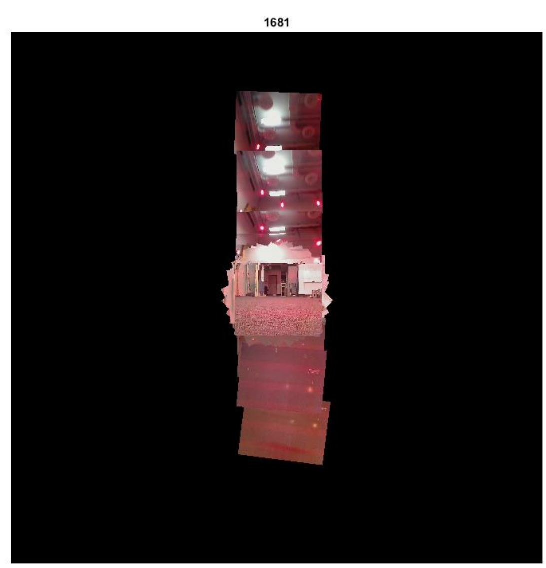

Fig. 6. Image Stitching output for train set 1 using homography.

VII. Results

A. Euler Angle Plots for all Train Datasets

The plots for train datasets 1 to 9 are shown in Figs. 9, 10, 11, 12, 13, 14, 15, 16 and 17.

B. Euler Angle Plots for Test Dataset

The plots for test dataset is shown in Fig. 18.

C. Panorama Images For Train Dataset

The panorama images using cylindrical projection and no blending for train datasets 1, 2, 8 and 9 are shown in Figs. 19, 20, 21 and 22 respectively.

Fig. 7. Image Stitching output for train set 1 using cylindrical projection without blending.

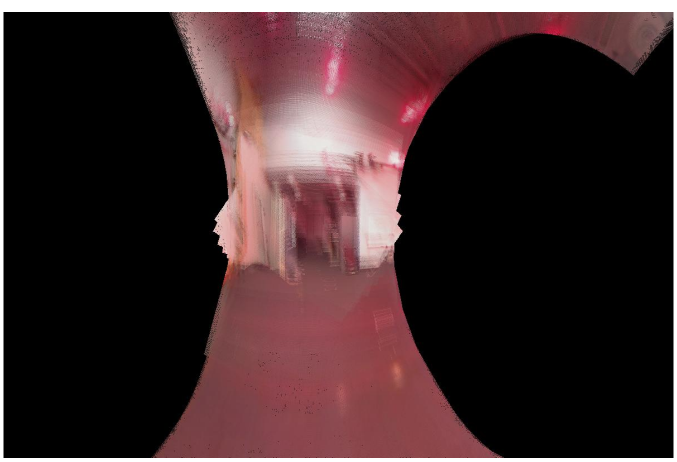

Fig. 8. Image Stitching output for train set 1 using cylindrical projection with blending ($\alpha = 0.75s$).

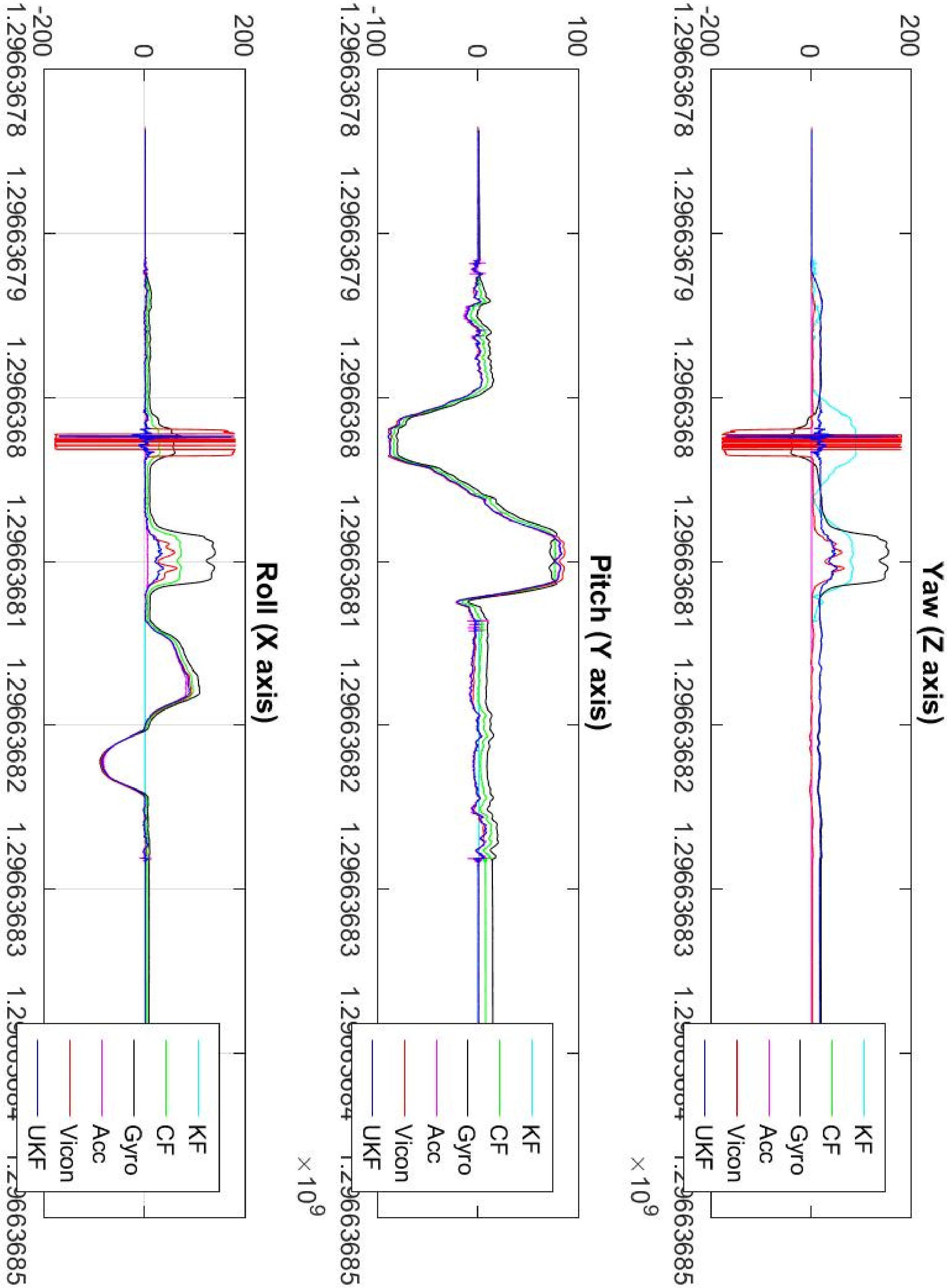

Fig. 9. Euler angle plots for training set 1 (all the implemented filters).

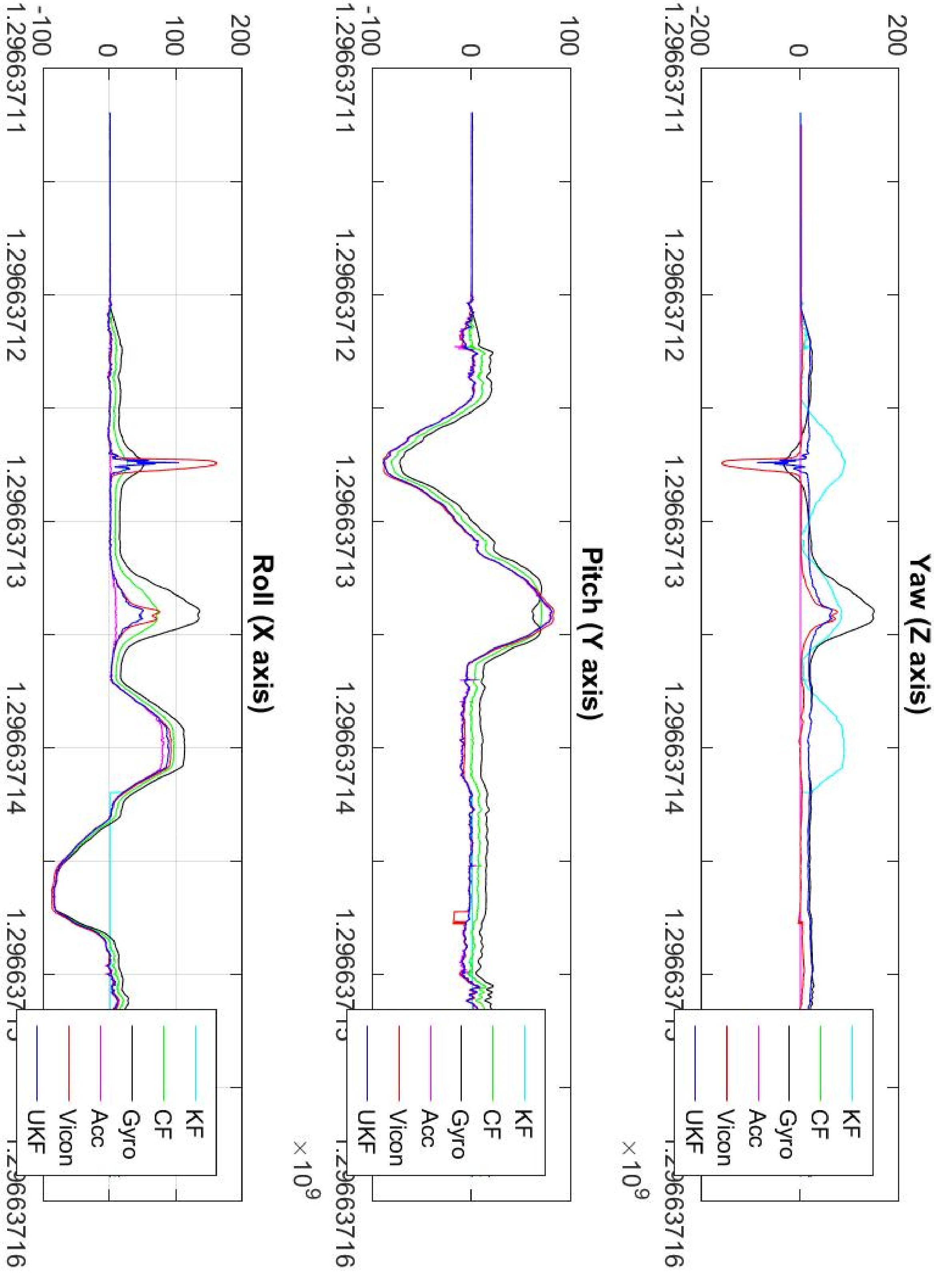

Fig. 10. Euler angle plots for training set 2 (all the implemented filters).

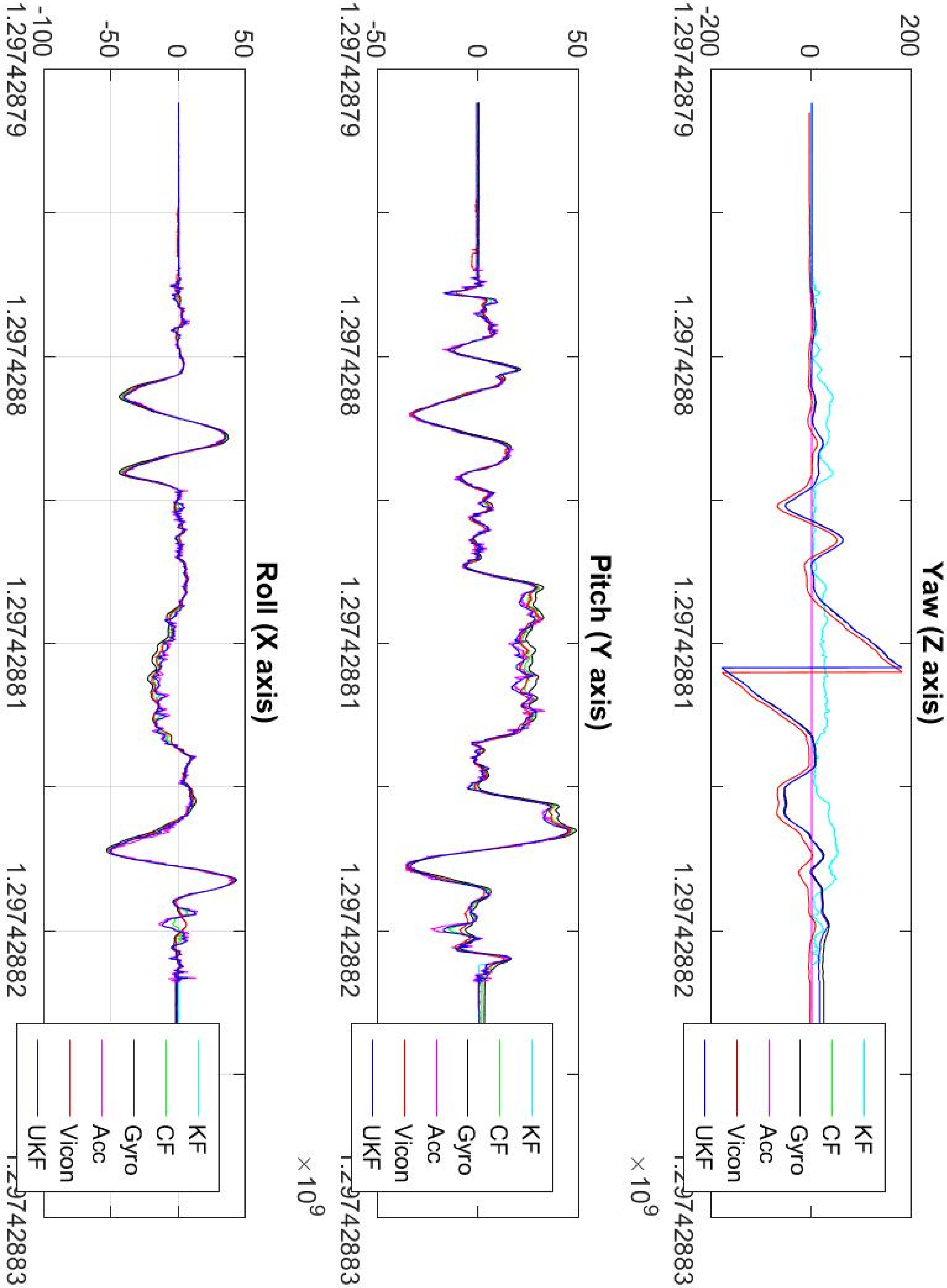

Fig. 11. Euler angle plots for training set 3 (all the implemented filters).

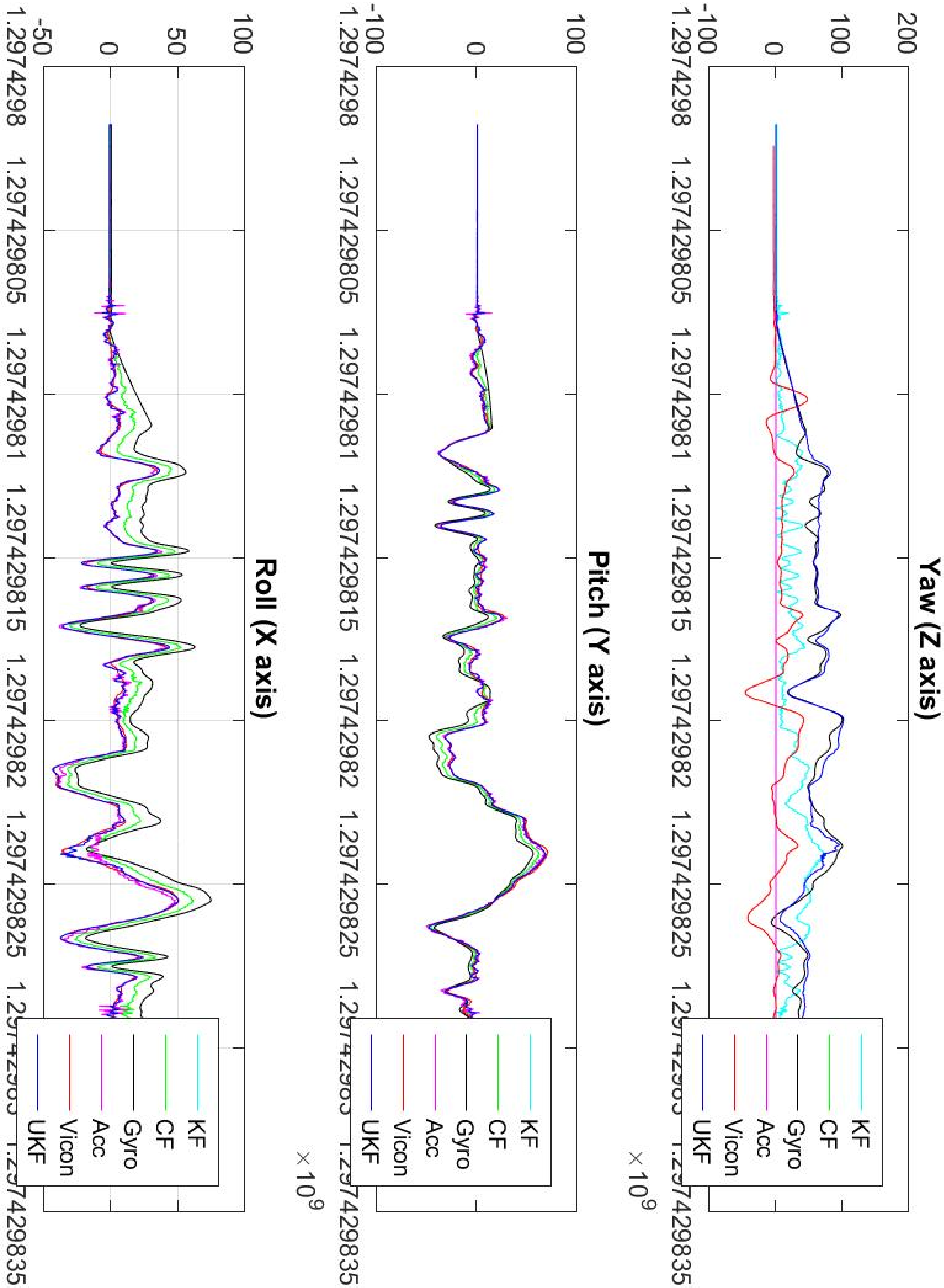

Fig. 12. Euler angle plots for training set 4 (all the implemented filters).

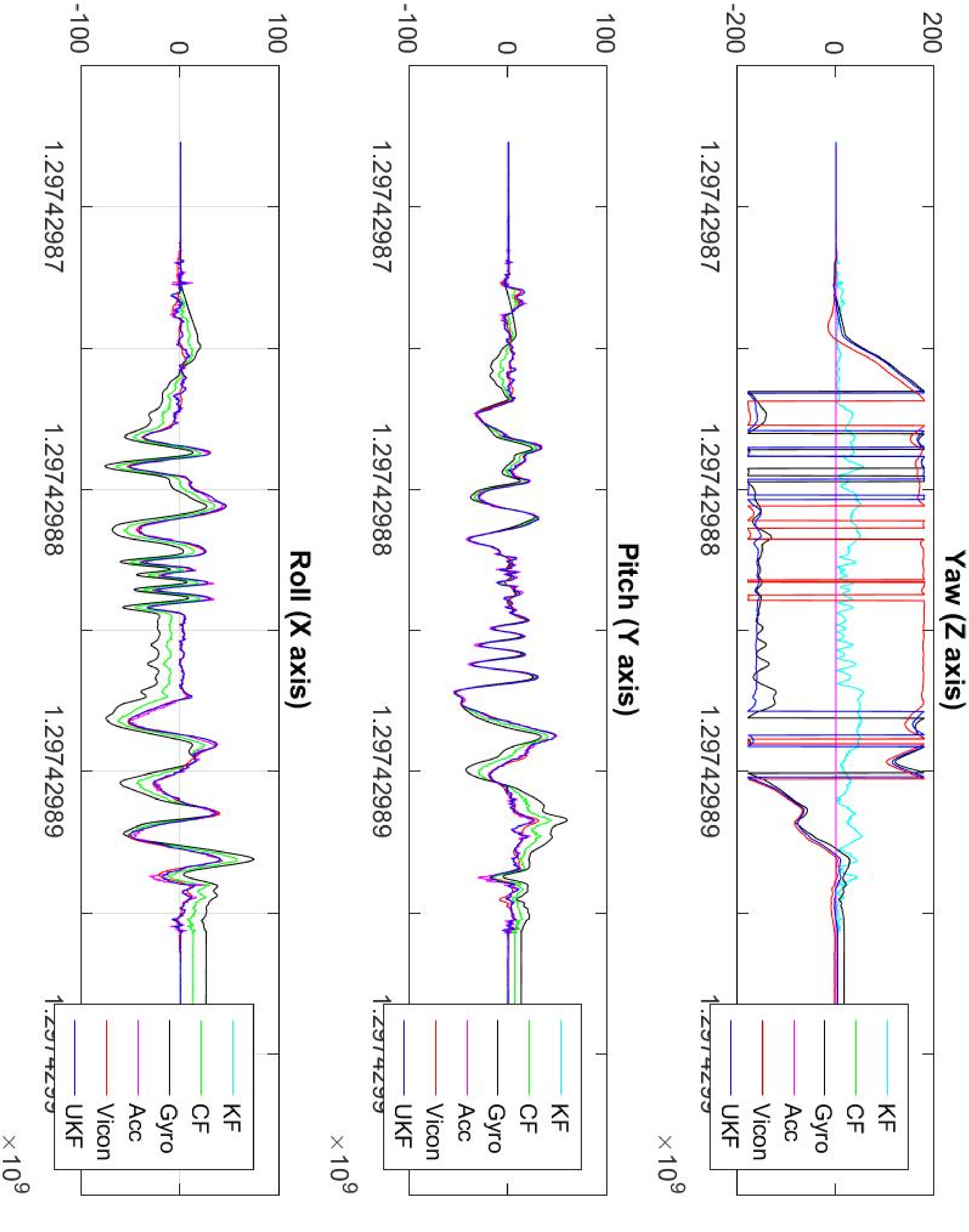

Fig. 13. Euler angle plots for training set 5 (all the implemented filters).

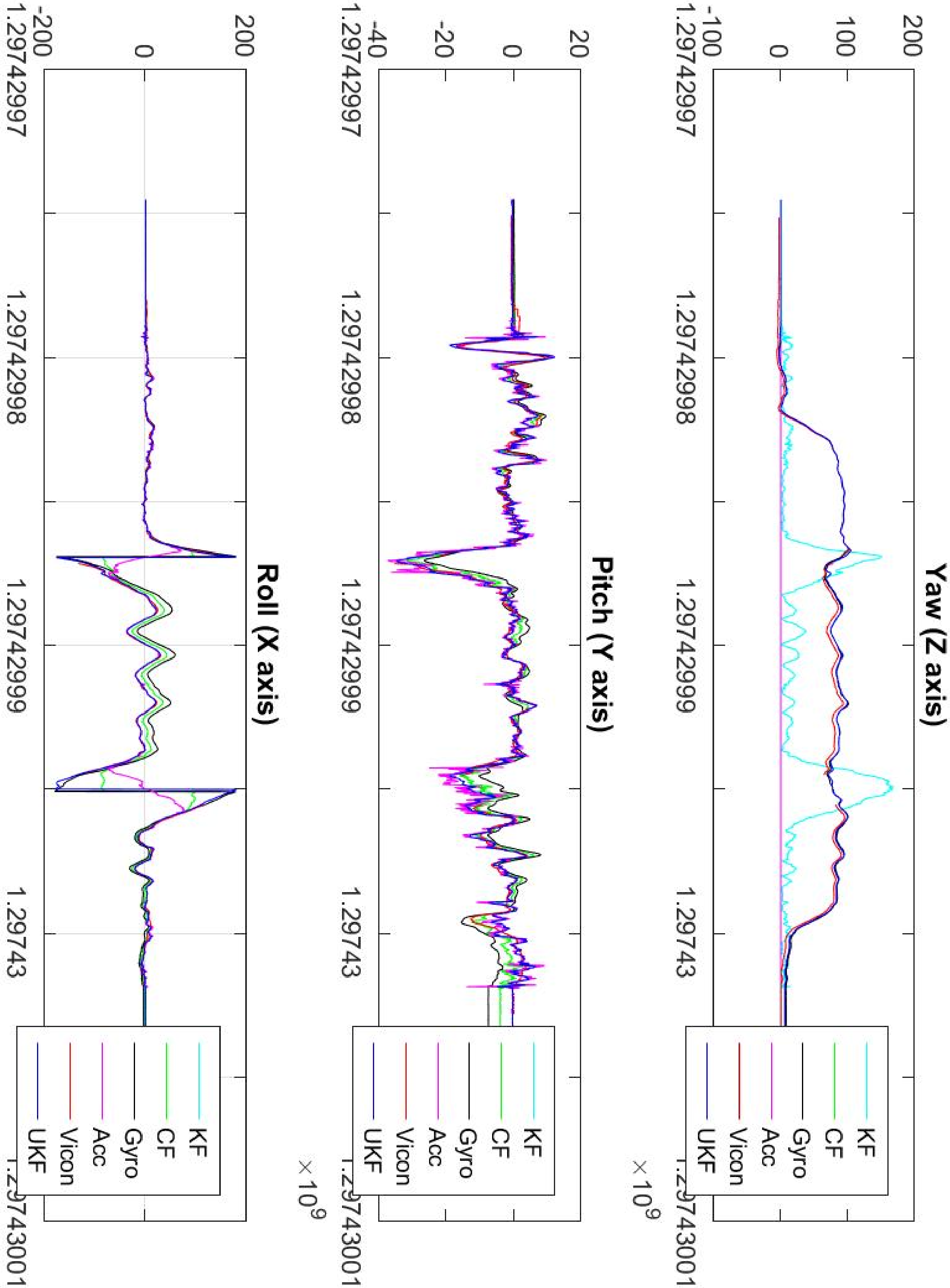

Fig. 14. Euler angle plots for training set 6 (all the implemented filters).

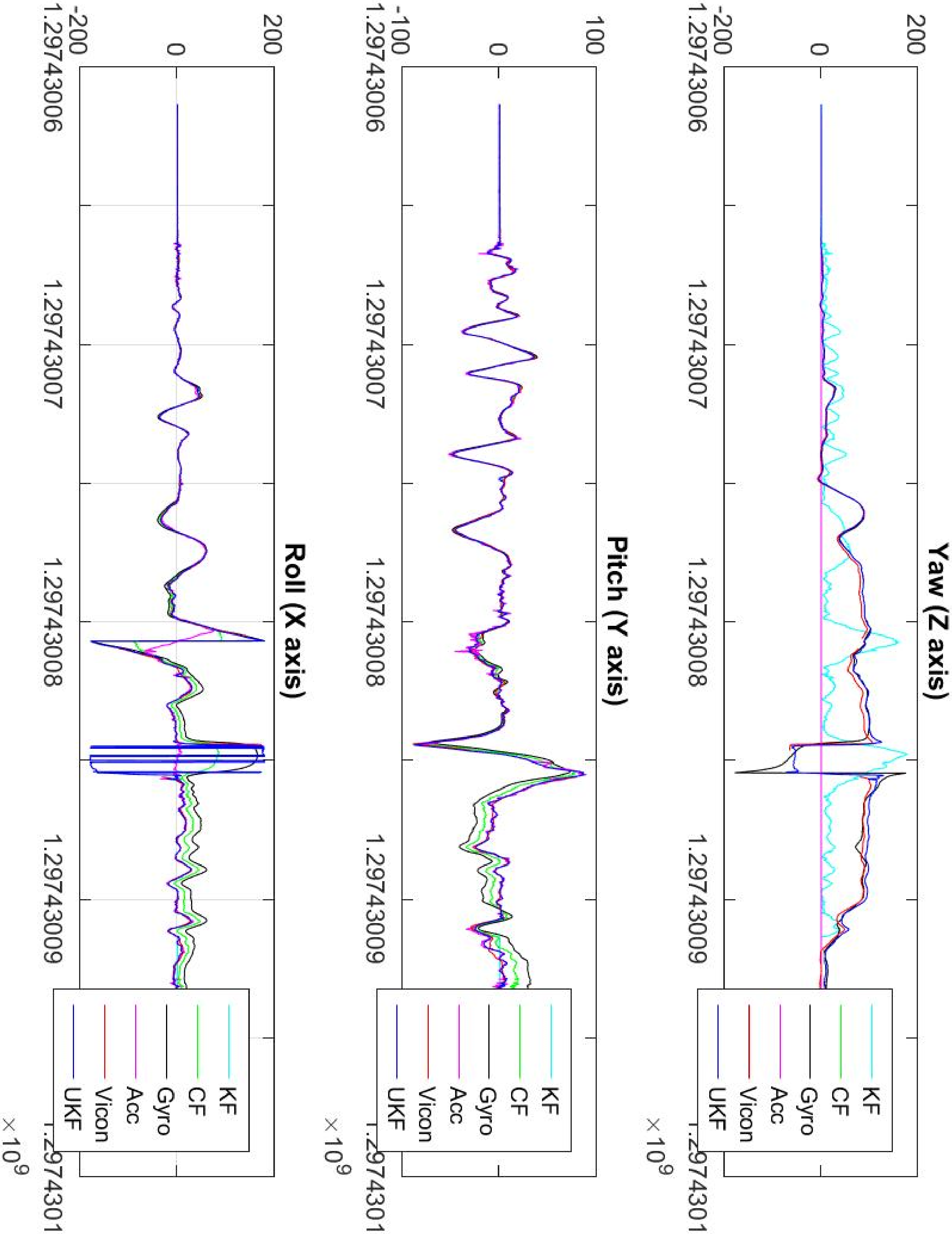

Fig. 15. Euler angle plots for training set 7 (all the implemented filters).

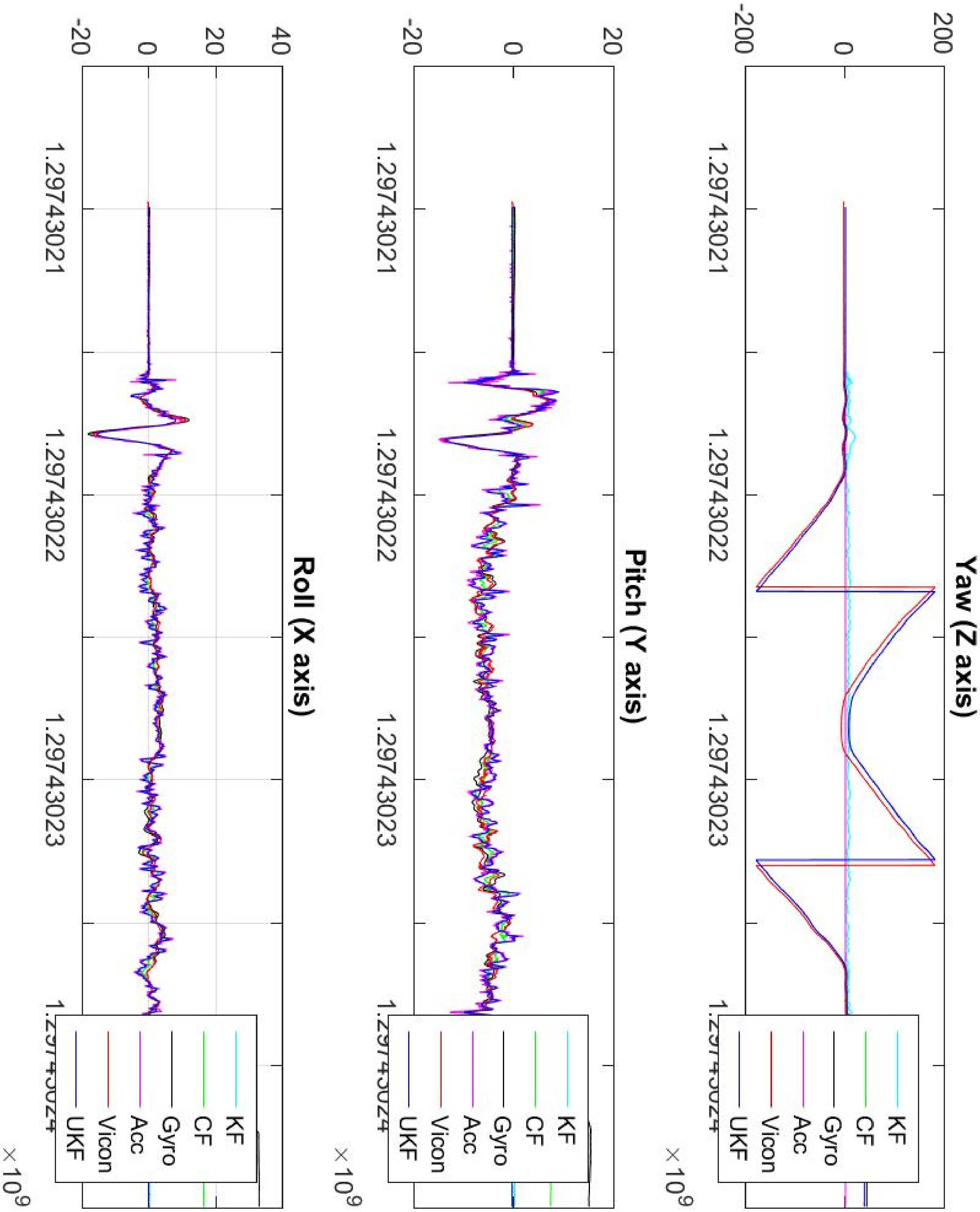

Fig. 16. Euler angle plots for training set 8 (all the implemented filters).

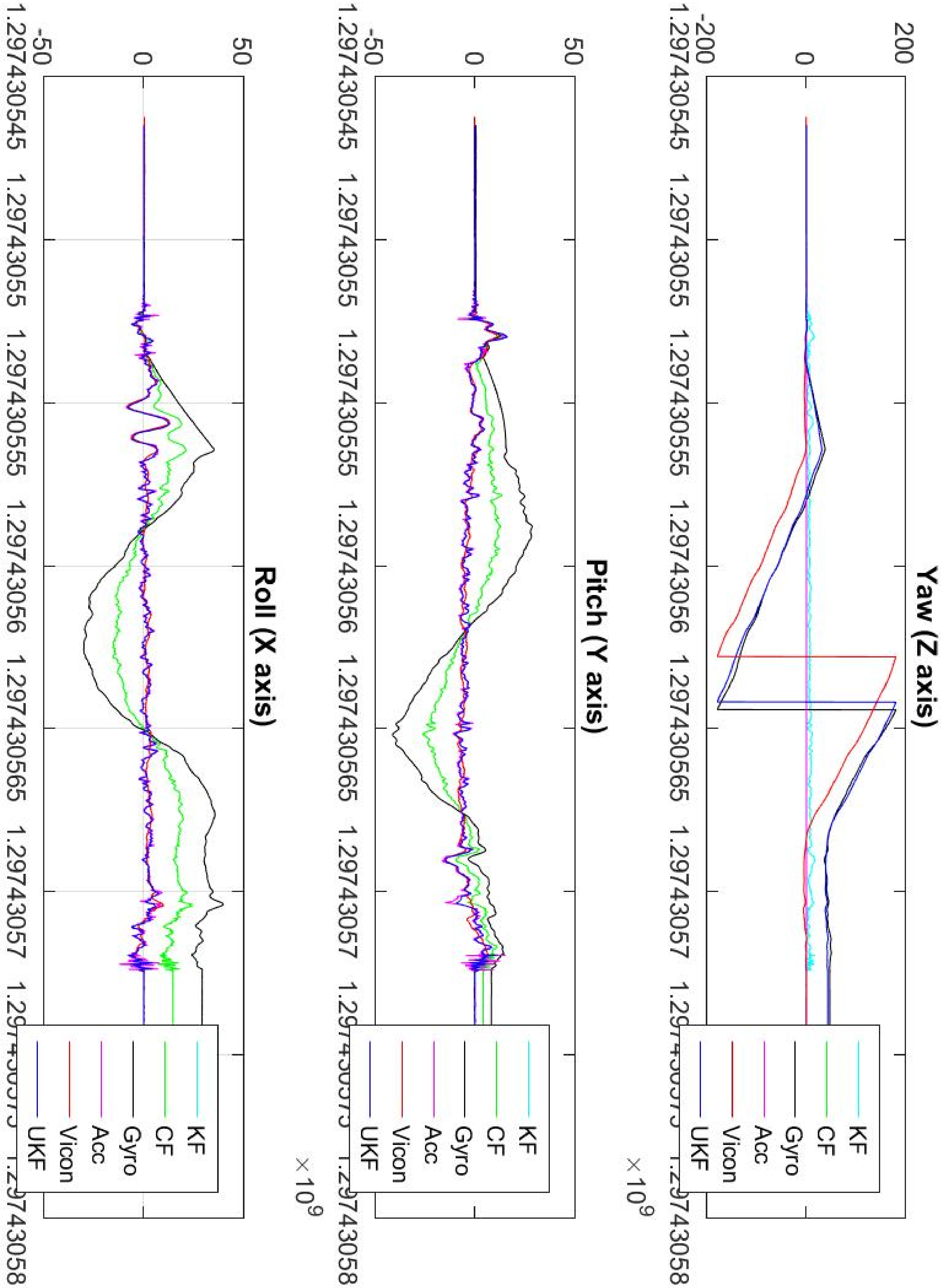

Fig. 17. Euler angle plots for training set 9 (all the implemented filters).

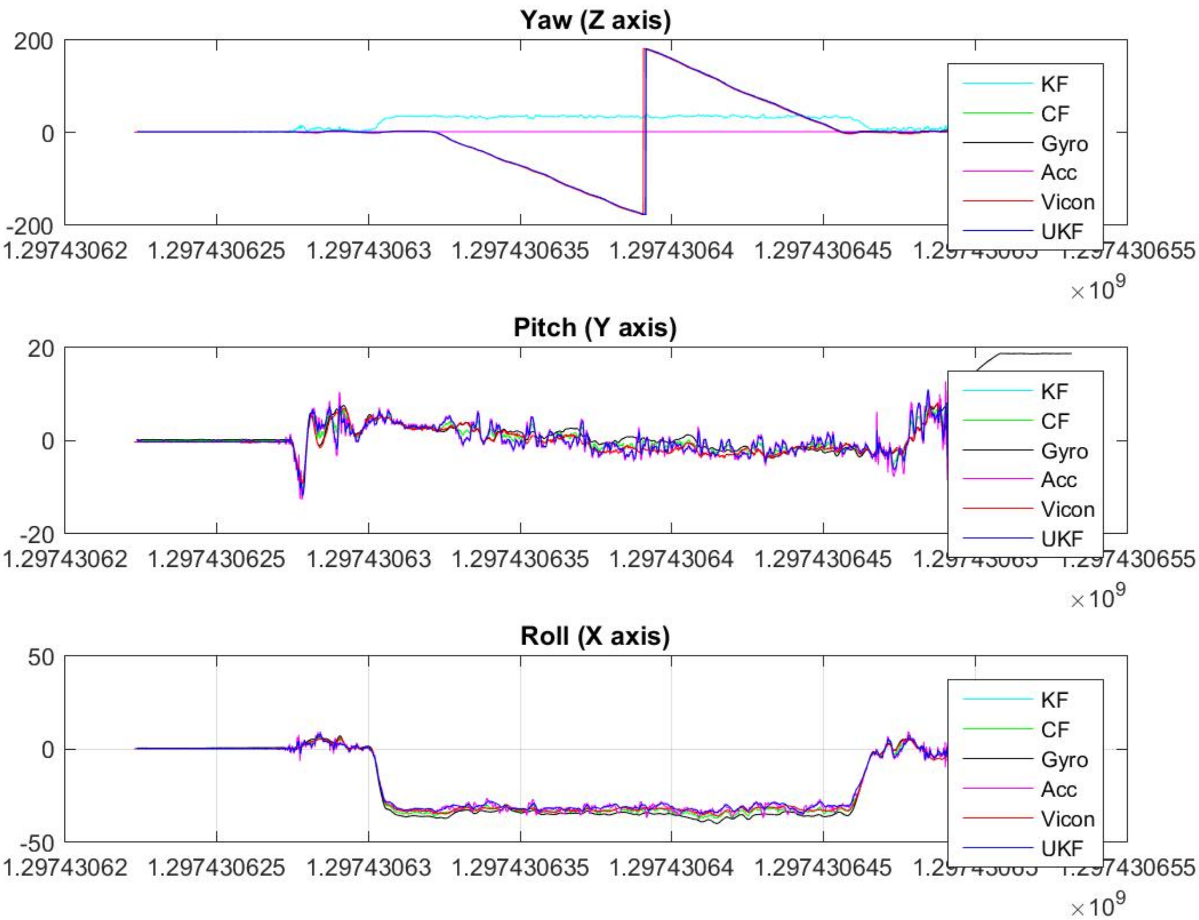

Fig. 18. Euler angle plots for test set (all the implemented filters).

Fig. 19. Panorama Image using Cylindrical projection for training set 1 without blending.

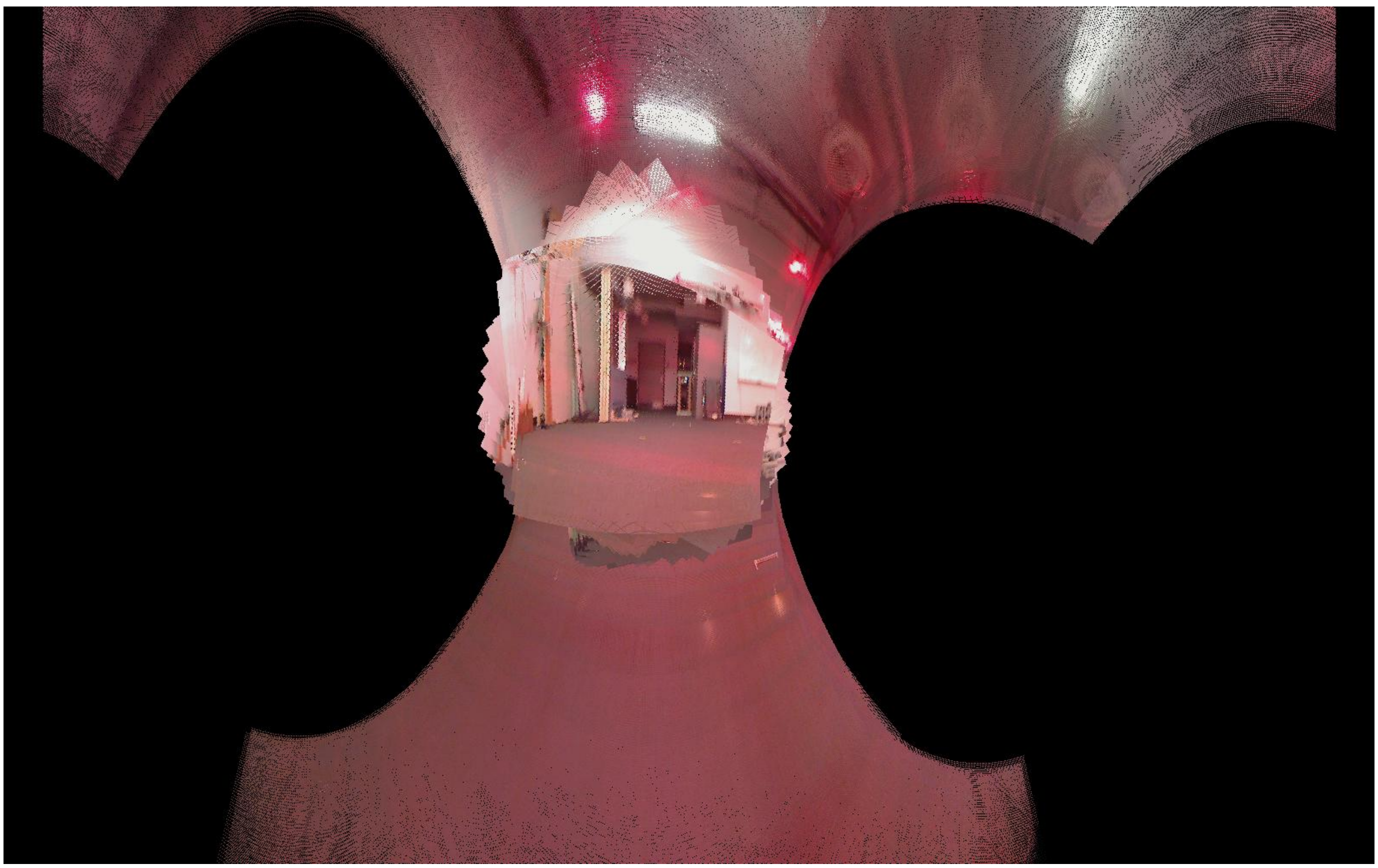

Fig. 20. Panorama Image using Cylindrical projection for training set 2 without blending.

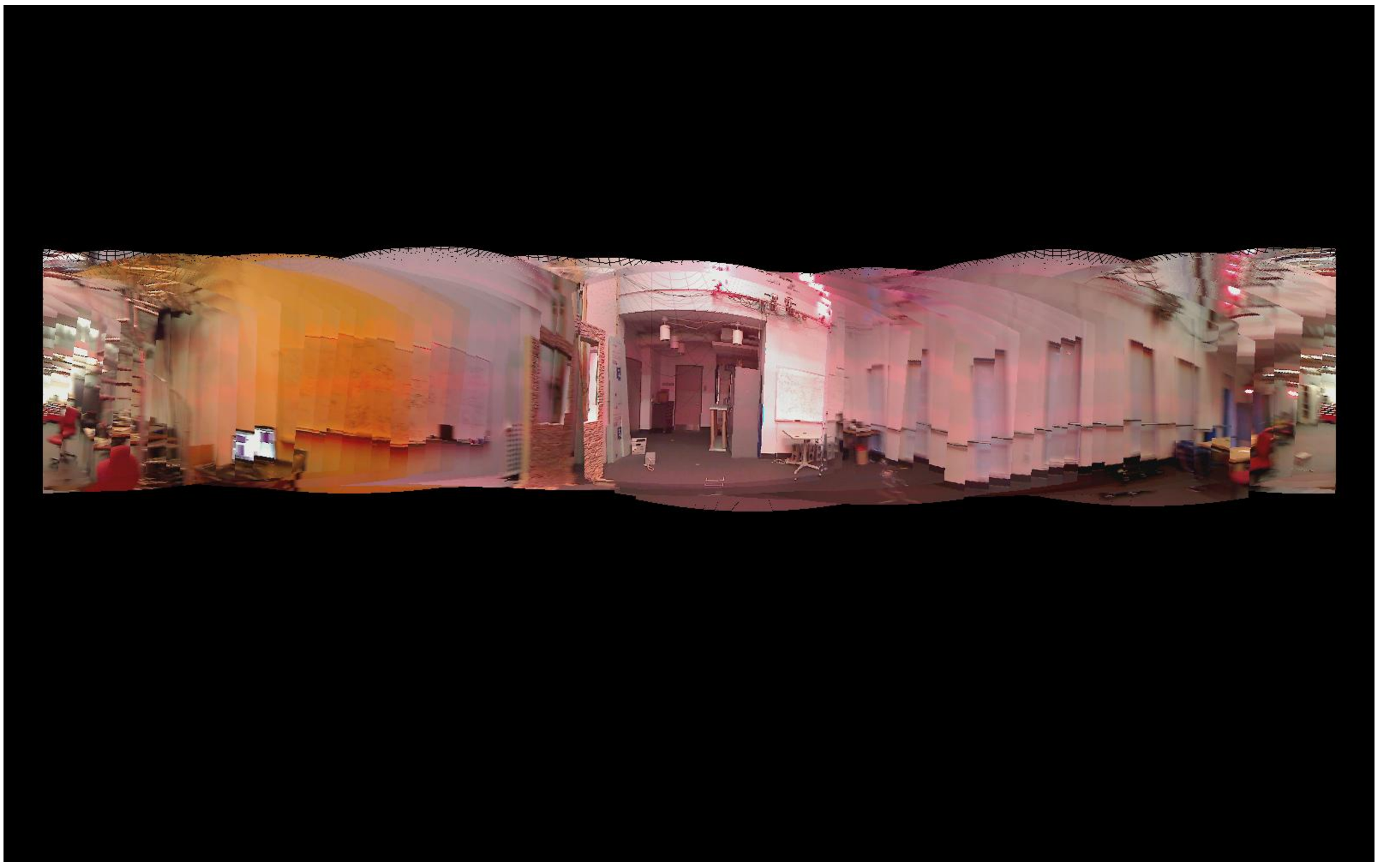

Fig. 21. Panorama Image using Cylindrical projection for training set 8 without blending.

Fig. 22. Panorama Image using Cylindrical projection for training set 9 without blending.

The panorama images using cylindrical projection and with blending weight of 0.25 for train datasets 1, 2, 8 and 9 are shown in Figs. 23, 24, 25 and 26 respectively.

D. Panorama Images For Test Dataset

The panorama images using cylindrical projection and no blending for test dataset is shown in Fig. 27.

The panorama images using cylindrical projection with blending weight of 0.25 for test dataset is shown in Fig. 27.

E. Analysis of results

Clearly, UKF outperforms all other alogithms as expected, it has the least drift and is the closest to vicon. Surprisingly, for most datasets with smooth movements CF perfroms almost on par with UKF. KF doesn't work as expected possibly due to wrong tuning or a bug in the code. The gyro integration also follows vicon but drifts a lot. Also, gyroscope is follows vicon with a lot of spikes as expected. However, if UKF is badly tuned it performs the worst of them all. Mainly, when the accelerometer process noise was not tuned to the right value, yaw output of UKF was the worst.

VIII. Estimating Camera Parameters

The webcam used is Logitech C905 which uses a 1/4.5" sensor. The sensor is of size $4.5mm \times 3.6mm$. Horizontal Focal length is calculated as,

$$f_x = \frac{sensorsize \times WorkingDist}{HorizontalFOV} = \frac{4.5mm \times 6ft}{35^\circ} = 235.13mm$$

$$f_y = \frac{sensorsize \times WorkingDist}{VerticalFOV} = \frac{3.6mm \times 6ft}{30^\circ} = 219.456mm$$

$$f = \sqrt{f_x^2 + f_y^2} = 321.631mm$$

IX. Time Synchronization and Error Calculation

Time Synchronization was done by finding minimum absolute closest timestamp between 2 sequences. The error metric used for euler angles which respects flips was called Mean Absolutte Difference Of Square (MADS). MADS Error between 2 sequences a and b with n samples is defined as,

$$MADS = \frac{1}{n}\sum_{i=1}^{n} \left|{a_i}^2 - b_i^2\right|$$

MADS Error won't work if the absolute difference matters, in our case we are sure that angles will be close enough to not have this problem. This combats the euler angle flipping problem.

X. Videos

The Videos pertaining to panorama formation can be found at https://goo.gl/BJitRl.

XI. Important Lessons Learnt

UKF is highly sensitive to the scale of covariances.

XII. Conclusions

Overall, UKF outperformed all the other methods by being the most robust and having the least drift. MADS error was the lowest. Surprisingly, complementary filter worked very well though it had some drift towards the end of every sequence. Also, Kalman filter didn't work as expected maybe due to wrong tuning or a bug in the code. The panorama stitching with blending on cylindrical projection gave the best/most visually appealing results. The preformance of the pipeline on the test set was good.

Acknowledgment

The author would like to thank Dr. Daniel D. Lee and all the Teaching Assistants of ESE 650 course for all the help in accomplishing this project.

References

[1] Complementary Filter, Link
[2] Attitude from accelerometer, Link
[3] Kalman Filter, Link
[4] Edgar Kraft, *A quaternion-based unscented Kalman filter for orientation tracking*, Sixth International Conference of Information Fusion, Vol. 1, 2003.
[5] S. J. Julier and J. K. Uhlmann, *A new extension of the Kalman filter to nonlinear systems*, International Symposium on Aerospace/Defense Sensing, Simulation and Controls, 1997.
[6] Cylindrical Projection for Panorama Stitching, Link
[7] Another Cylindrical Projection for Panorama Stitching, Link

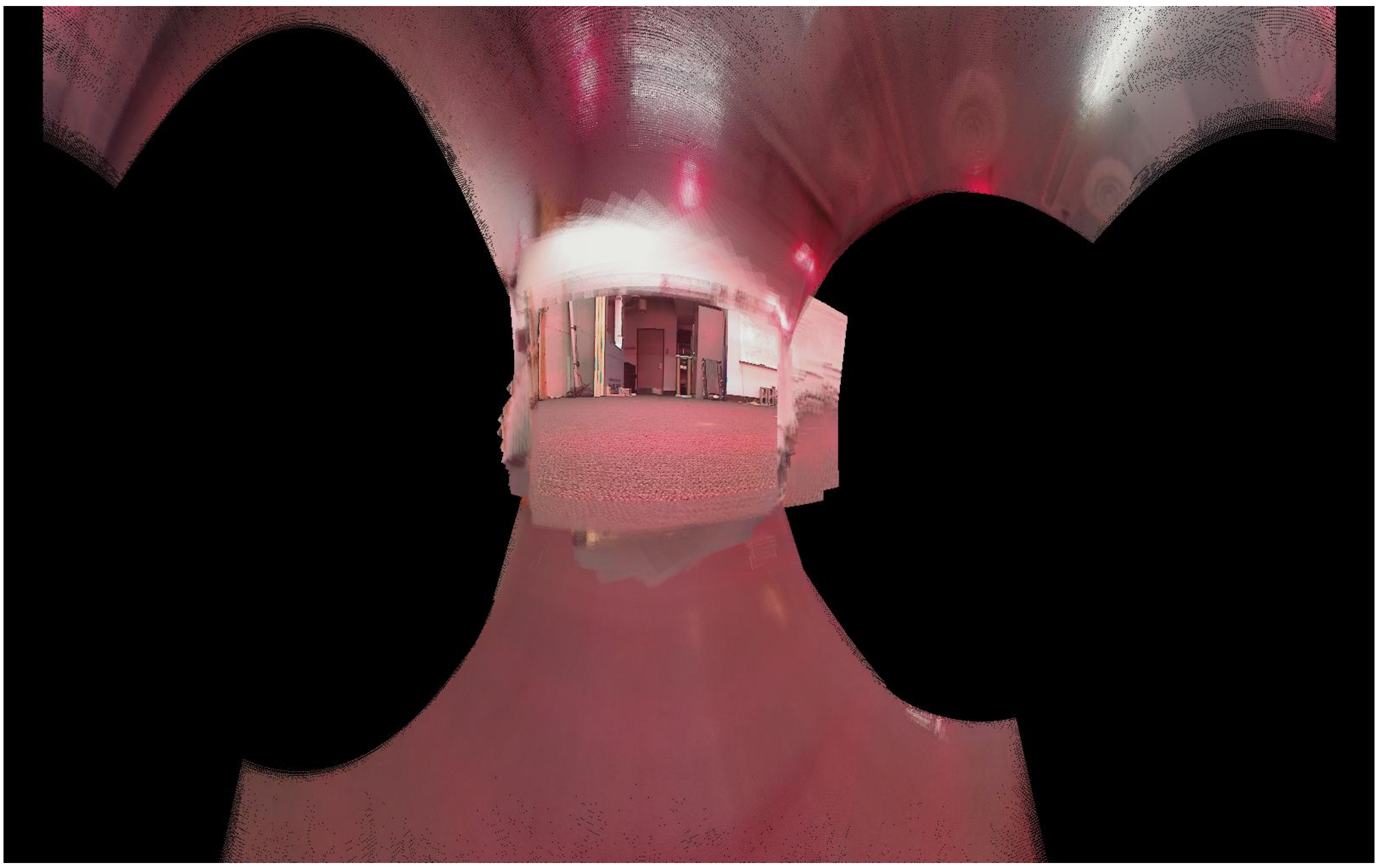

Fig. 23. Panorama Image using Cylindrical projection for training set 1 with blending weight 0.25.

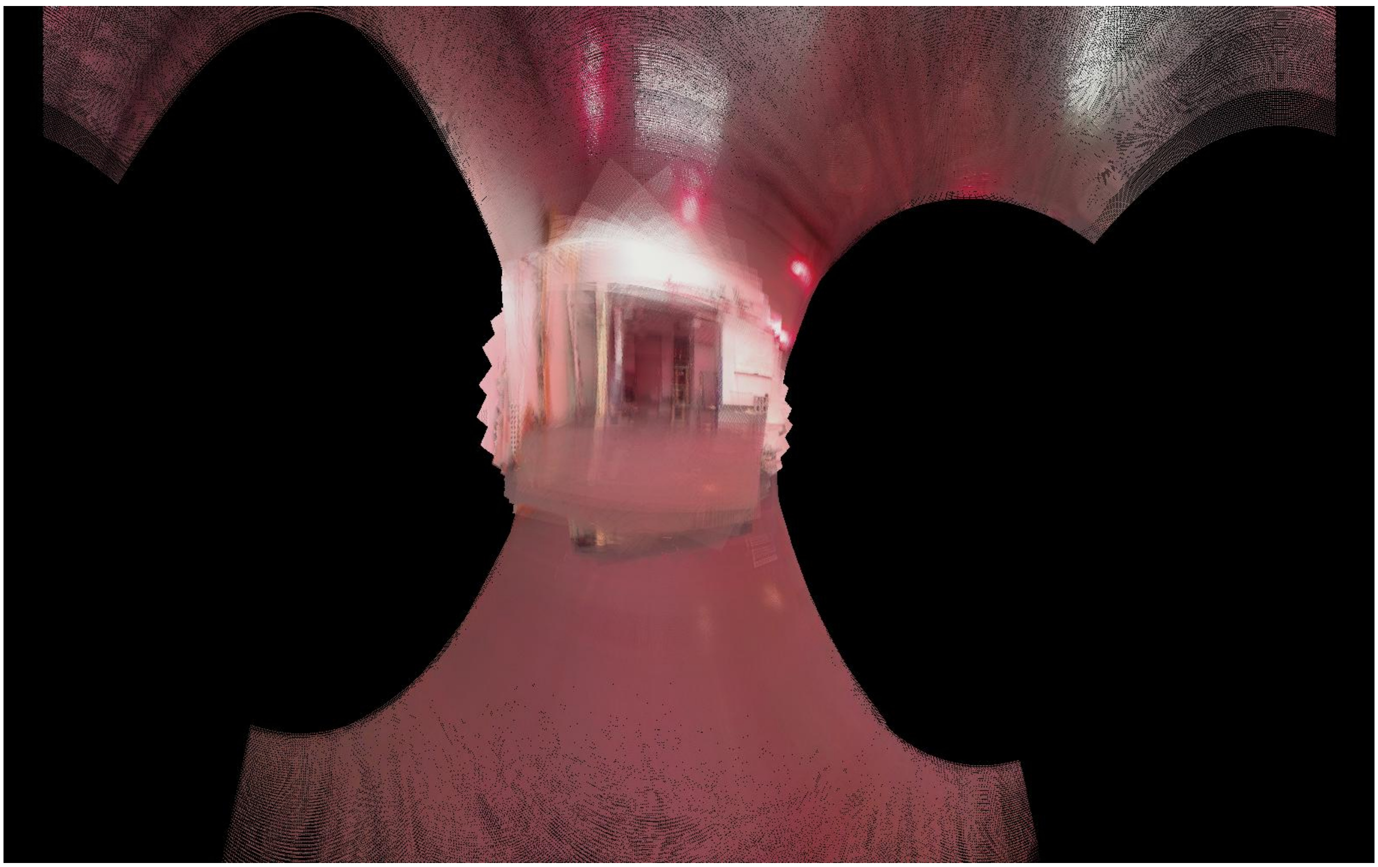

Fig. 24. Panorama Image using Cylindrical projection for training set 2 with blending weight 0.25.

Fig. 25. Panorama Image using Cylindrical projection for training set 8 with blending weight 0.25.

Fig. 26. Panorama Image using Cylindrical projection for training set 9 with blending weight 0.25.

Fig. 27. Panorama Image using Cylindrical projection for test set without blending.

Fig. 28. Panorama Image using Cylindrical projection for test set with blending weight 0.25.