Orientation Tracking and Panorama Generation.

Imoleayo Abel

I. Introduction

In several robotic systems, it is usually important to track the orientation of the system to be able to determine what actions to perform. For example, the orientation of a robotic system can be used to determine whether the system is upright or has toppled over. Another application where orientation tracking is important is in fitness/health tracking applications on smart-watches where gyroscope and accelerometer data are used to track human motion for computing health and fitness metrics. In these applications, the quality of the assertion made about the system depends on the ability to process the sensor information to correctly estimate the orientation of the system.

In this paper, we describe the implementation of an unscented Kalman filter for estimating the orientation of a camera system using angular velocity and acceleration data from an inertial measurement unit (IMU). We also describe an approach for using the estimated camera orientation over time to stitch together a sequence of images into a single continuous panorama image.

II. Problem Formulation

Given a set of gyroscope and accelerometer data containing (respectively) the angular velocity $\omega_t \in \mathbb{R}^3$ and linear acceleration $a_t \in \mathbb{R}^3$ in the body frame of an IMU over time; and a corresponding set of RGB images $C_t \in \mathbb{R}^{3\times N_t}$ over time from a camera fixed at position $p \in \mathbb{R}^3$ in the IMU frame, with N_t being the number of pixels in image t, the goal of this project is to:

1) **Unscented Kalman Filtering**: Implement an unscented Kalman filter using the gyroscope and accelerometer data to estimate the orientation of the IMU $q_t \in \mathbb{S}^3$ at each time step t.
2) **Panorama Generation**: Use the estimated orientation of the IMU (and camera) over time to generate a single panorama image of all the camera images over time.

III. Technical Approach

A. Unscented Kalman Filtering

To represent the state of the system, we use a 7-dimensional state vector $x_t = \begin{bmatrix} q_t^T & \omega_t^T \end{bmatrix}^T$ as described in [1] where $q_t \in \mathbb{R}^4$ is a unit quaternion estimating the orientation of the camera/IMU[1] system at time t and $\omega_t \in \mathbb{R}^3$ is the angular velocity of the camera/IMU system in the $x-, y-$, and $z-$ directions of the camera/IMU frame at time t.

Process Model: The following equations describe the evolution of the states of the system.

$$q_{t+1} = q_t \circ q_\Delta \tag{1}$$

$$\omega_{t+1} = \omega_{t+1} \tag{2}$$

where q_Δ is the quaternion representation of a rotation by $\omega_t \Delta t$ and $q_t \circ q_\Delta$ denotes a quaternion multiplication of q_t and q_Δ. While (2) does not represent a state "evolution" per-se, it does indicate that we use the angular velocity data at time $t+1$ as the estimate of angular-velocity portion of state at time $t+1$. While this portion of the state estimate is fixed to the data, the belief in it's accuracy i.e. the certainty of this estimate does change over time via a covariance matrix that ultimately factors into the estimate of the quaternion portion of the state of the system.

Measurement Model: For the measurement model, we estimate the accelerometer measurement in the camera/IMU frame by rotating the gravity vector in the world frame $g = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$ using the quaternion inverse of the orientation estimate at t as follows:

$$\begin{bmatrix} 0 \\ g' \end{bmatrix} = q_t^{-1} \circ \begin{bmatrix} 0 \\ g \end{bmatrix} \circ q_t \tag{3}$$

where that g' is the 3-dimensional gravity vector in the camera/IMU frame.

Prediction: In the prediction step of the unscented Kalman filter, we begin by generating a set of 13 sigma points $\{\mathcal{X}_i \in \mathbb{R}^7\}$ as follows

$$\mathcal{X}_0 = \begin{pmatrix} q_{t-1} \\ \omega_{t-1} \end{pmatrix}, \tag{4}$$

$$\mathcal{X}_{i,i+n} = \begin{pmatrix} q_{t-1} \\ \omega_{t-1} \end{pmatrix} \pm \left[\sqrt{6(P_{t-1}+Q)}\right]_i, \; i = 1, \cdots, 6 \tag{5}$$

where $[M]_i$ denotes the $i-$th column of matrix M, and $P_{t-1}, Q \in \mathbb{R}^{6\times 6}$ are the state covariance matrix at timestep $t-1$ and the noise covariance matrix of the process model respectively. To reconcile the dimension mismatch in the addition in (5), we convert the first three rows of the columns of $\sqrt{6(P_{t-1}+Q)}$ to quaternions, and then pre-multiply the outcome by q_t. This approach is used for all subsequent additions of a 4-dimensional quaternion to a 3-dimensional vector: convert the vector to a quaternion, and pre-multiply the result by the original quaternion.

[1]Since the camera is fixed at position $p \in \mathbb{R}^3$ in the IMU frame, the orientation of the camera relative to the IMU is constant, and as such we use "camera/IMU" loosely to imply the orientation/angular-velocity of one or the other since they are moving together.

After generating these 13 sigma points, we pass them through the process model to generate a new set $\{\mathcal{Y}_i \in \mathbb{R}^7\}$ of predictions of the current state of the system. We note here that the angular velocity portion of the sigma points $\mathcal{X}_i$ remains the same after the process model is applied. Only the quaternion portion undergoes a transformation. From the set $\{\mathcal{Y}_i\}$, we compute the a priori estimate of the current state x_t^- and current state covariance matrix P_t^- as follows

$$x_t^- = \text{mean}\{\mathcal{Y}_i\} = \begin{pmatrix} q_t^- \\ \omega_t^- \end{pmatrix} \tag{6}$$

$$P_t^- = \text{covariance}\{\mathcal{Y}_i\} \tag{7}$$

We implement the mean in (6) as a weighted sum of predictions $\{\mathcal{Y}_i\}$ with weight 0 for $i = 0$, and $\frac{1}{12}$ for $i = 1, 2, \cdots, 12$. We use a quaternion averaging algorithm to compute the mean of the quaternion portion, and a regular weighted sum for the angular velocity portion. To compute the covariance, we first subtract the mean x_t^- from each $\mathcal{Y}_i$. For the quaternion portion of $\mathcal{Y}_i$, this corresponds to post multiplying by the inverse of the quaternion portion of q_t^-. Once the mean is subtracted from the predictions $\{\mathcal{Y}_i\}$, we convert the quaternion portion of the outcome to 3-dimensional rotation vectors, which when combined with the angular velocity portion leads to a set $\{\mathcal{W'}_i \in \mathbb{R}^6\}$ of 13 "residual" vectors from which the a priori covariance P_t^- is computed as follows:

$$P_t^- = 2\mathcal{W'}_0 \mathcal{W'}_0^T + \frac{1}{12}\sum_{i=1}^{12} \mathcal{W'}_i \mathcal{W'}_i^T \tag{8}$$

This conversion and weighted sum approach is used for all subsequent mean and covariance computation for state vectors.

Update: In the update step, we begin by applying the measurement model to the quaternion portion of the predictions $\{\mathcal{Y}_i\}$ to generate a set of measurement estimates $\{\mathcal{Z}_i \in \mathbb{R}^3\}$. We then compute the $z_t^- \in \mathbb{R}^3-$expected measurement, and $P_{zz} \in \mathbb{R}^{3\times3}-$the uncertainty in the measurement caused by the uncertainty in the prediction as follows:

$$z_t^- = \text{mean}\{\mathcal{Z}_i\} \tag{9}$$

$$P_{zz} = \text{covariance}\{\mathcal{Z}_i\} \tag{10}$$

Note that since the set $\{\mathcal{Z}_i\}$ are regular $\mathbb{R}^3$ vectors, we compute the mean and covariances as we would for regular spatial vectors.

Next, we compute the innovation $\nu_k \in \mathbb{R}^3$ as the difference between the expected measurement z_t^- and actual measurement from the accelerometer z_t, and the innovation covariance $P_{\nu\nu} \in \mathbb{R}^{3\times3}$ as the sum of P_{zz} and $R \in \mathbb{R}^{3\times3}-$the measurement noise matrix

$$\nu_t = z_t - z_t^- \tag{11}$$

$$P_{\nu\nu} = P_{zz} + R \tag{12}$$

We also compute the cross correlation matrix $P_{xz} \in \mathbb{R}^{6\times3}$ representing the correlation between the process noise and measurement noise as follows:

$$P_{xz} = 2\mathcal{W'}_0[\mathcal{Z}_0 - z_t^-]^T + \frac{1}{12}\sum_{i=1}^{12} \mathcal{W'}_i \mathcal{Z}_i^T \tag{13}$$

The Kalman gain $K_t \in \mathbb{R}^{6\times3}$ is then computed as follows

$$K_t = P_{xz} P_{\nu\nu}^{-1} \tag{14}$$

Lastly, we use the Kalman gain to update the quaternion portion of the state estimate and state covariance as follows

$$q_t = q_t^- + K_t \nu_t \tag{15}$$

$$P_t = P_t^- - K_t P_{\nu\nu} K_t^T \tag{16}$$

Note that in (15), we first perform a regular matrix multiplication to compute $K_t\nu_t$ and then we convert the outcome to a quaternion that is then pre-multiplied by q_t^- to generate q_t.

B. Panorama Generation

To generate a panorama image, the first step was to convert the pixel locations in the camera image to spherical coordinates. This assumes that the camera images lie on a unit sphere in the camera frame of reference. Since the camera has a horizontal and vertical field of view of $\frac{\pi}{3}$ and $\frac{\pi}{4}$ respectively, it implies that the azimuth angles φ_c, and inclination θ_c (measured from the north pole) are in the range $-\frac{\pi}{6} \leq \varphi_c \leq \frac{\pi}{6}$, and $\frac{3\pi}{8} \leq \theta_c \leq \frac{5\pi}{8}$. Thus, for an `nrows` $\times$ `ncols` image, the (φ_c, θ_c) location of the pixel in the $i-$th row and $j-$th column is given as

$$\varphi_c = -\frac{\pi}{6} + \frac{j}{\texttt{ncols}}\frac{\pi}{3} \tag{17}$$

$$\theta_c = \frac{3\pi}{8} + \frac{i}{\texttt{nrows}}\frac{\pi}{4} \tag{18}$$

Next, we convert the spherical coordinates of the pixel location in the camera frame into Cartesian coordinates using the following formulae:

$$x_c = \cos\varphi_c \sin\theta_c \tag{19}$$

$$y_c = \sin\varphi_c \sin\theta_c \tag{20}$$

$$z_c = \cos\theta_c \tag{21}$$

Since we assume the images for this project all have the same dimension, the $(x_c, y_c, z_c)^T$ position in the camera frame of the $i, j-$th pixel in all the images are the same.
Now, for every camera image C_t taken at time t, we compute the Cartesian coordinates of each pixel in the world frame using the following equation

$$\begin{pmatrix} x_w \\ y_w \\ z_w \end{pmatrix} = R(q_t) \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} + p \tag{22}$$

where $R(q_t) \in \mathbb{R}^{3\times3}$ is the rotation matrix corresponding the quaternion $q_t-$the orientation of the camera/IMU system at time t, and p is the fixed position of the camera in the world frame.
Once the Cartesian coordinates of the pixels in the world

frame have been obtained, we normalize them to unit norm so they fit on a unit sphere in the world frame, and then compute the spherical coordinates in the world frame using

$$\varphi_w = \arctan \frac{y_w}{x_w} \tag{23}$$

$$\theta_w = \arccos z_w \tag{24}$$

With these world spherical coordinates we "generate" cylindrical coordinates by interpreting the spherical azimuth and inclination as the cylindrical azimuth and height respectively. Then we unwrap the cylinder into a $\texttt{prows} \times \texttt{pcols}$ panorama image, by placing the pixel at (φ_w, θ_w) in the i_p-th row, and j_p-th column of the panorama image as defined below

$$i_p = \frac{\theta_w}{\pi}\texttt{prows} \tag{25}$$

$$j_p = \frac{\pi + \varphi_w}{2\pi}\texttt{pcols} \tag{26}$$

We π in the numerator of (26) is added since the $\arctan$ function in (23) (implemented with atan2) has a range of $(-\pi, \pi]$.

IV. Result and Discussion

A. Unscented Kalman Filtering

We initialize the state and covariance matrices of the system as follows:

$$q_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{27}$$

$$\omega_0 = \omega_0 \text{ from gyroscope data.} \tag{28}$$

$$P_0 = 0.0001 I_6 \tag{29}$$

$$Q = 10^{-13} I_6 \tag{30}$$

$$R = 0.00065 I_3 \tag{31}$$

The choice of a very small process noise Q is because the process model relies directly on the angular velocity data from the gyroscope and occurs on a very small time scale. So at every time step, a portion of the state vector used for the process model comes directly from the gyroscope data and those doesn't suffer from any uncertainty propagation. This is corroborated in the quality of the panorama images which improves significantly with the choice of Q above as compared with a choice of $Q = 0.0001 I_6$. This is depicted in Fig. 1 below. Also, as seen in the plots from Fig 5-8, the roll, pitch, and yaw angles match the vicon ground truth data well.

B. Panorama Generation

While the Kalman filtering worked quite well with the orientation estimation, the panorama images only appeared decent-especially with the test set. We believe these imperfections are due to some translation of the camera during data collection since the panorama generation world on the assumption the that camera position is fixed. Interestingly, when we use only the first portion of the camera data on the train set (i.e. the part a portion of the camera data), the panorama results look significantly better suggesting that the problematic translation occurred in the latter part of data collection. Figures 2-4 below highlight this effect.

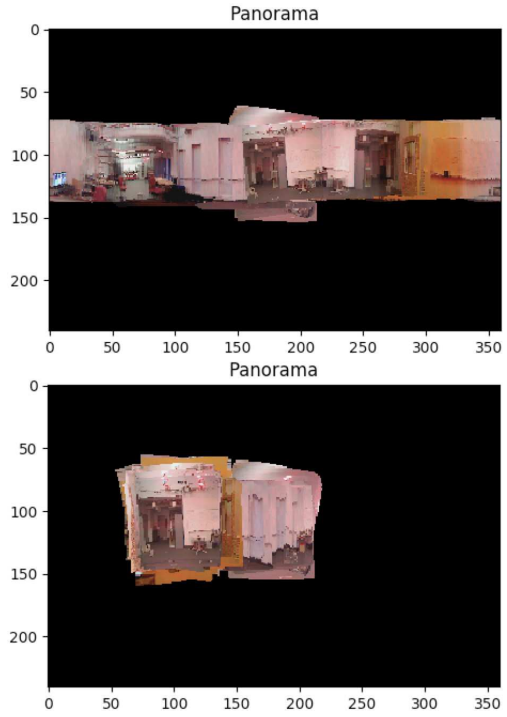

Fig. 1: Panorama for dataset $\texttt{8}$. Top: $Q = 10^{-13} I_6$, Bottom: $Q = 0.0001 I_6$.

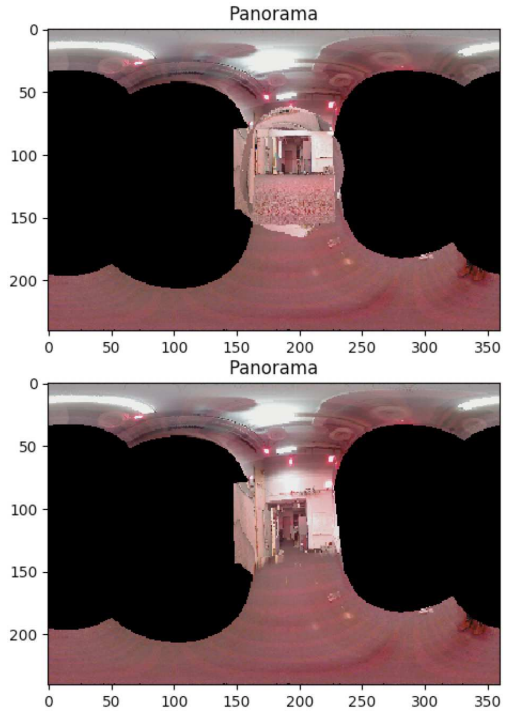

Fig. 2: Panorama for dataset $\texttt{1}$. Top: All cam data, Bottom: Portion a of cam data.

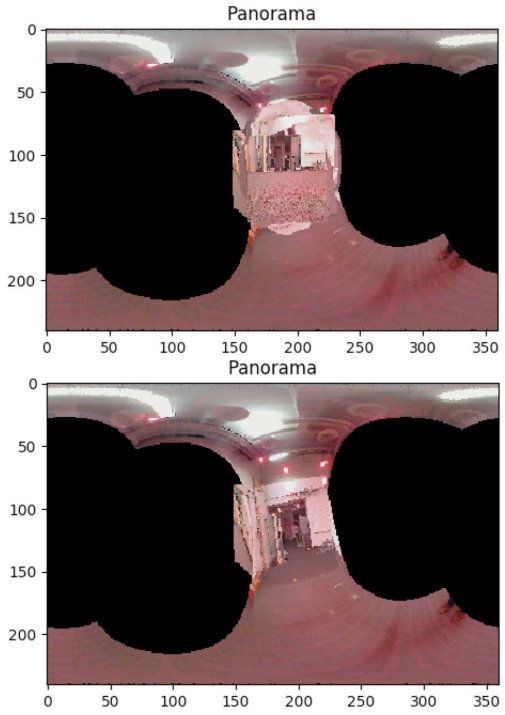

Fig. 3: Panorama for dataset 2. Top: All cam data, Bottom: Portion a of cam data.

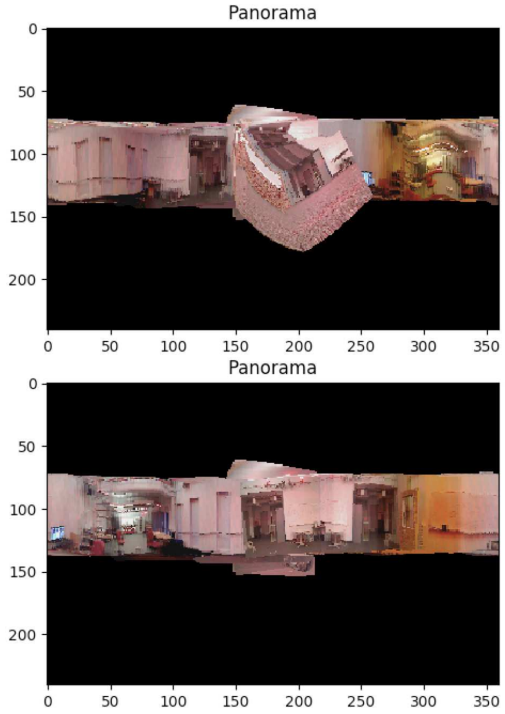

Fig. 4: Panorama for dataset 8. Top: All cam data, Bottom: Portion a of cam data.

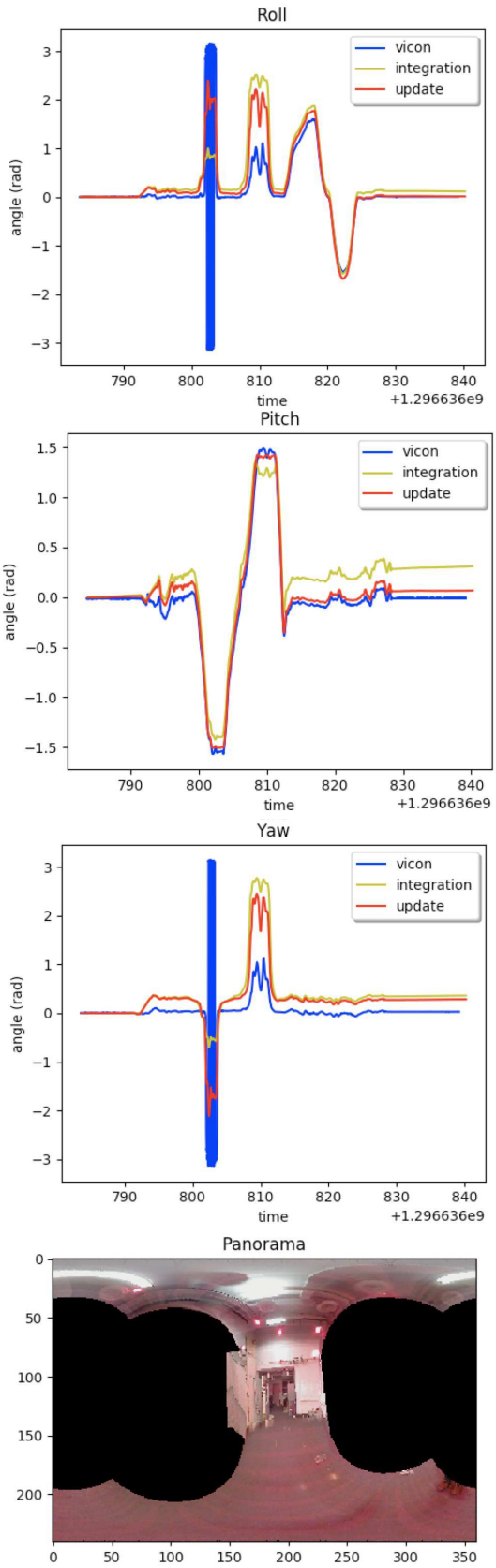

Fig. 5: Results for dataset 1. Top to bottom: Roll, Pitch, Yaw, Panorama.

REFERENCES

[1] E. Kraft, "A quaternion-based unscented kalman filter for orientation tracking," in *Proceedings of the Sixth International Conference of Information Fusion, 2003.*, vol. 1, 02 2003, pp. 47– 54.

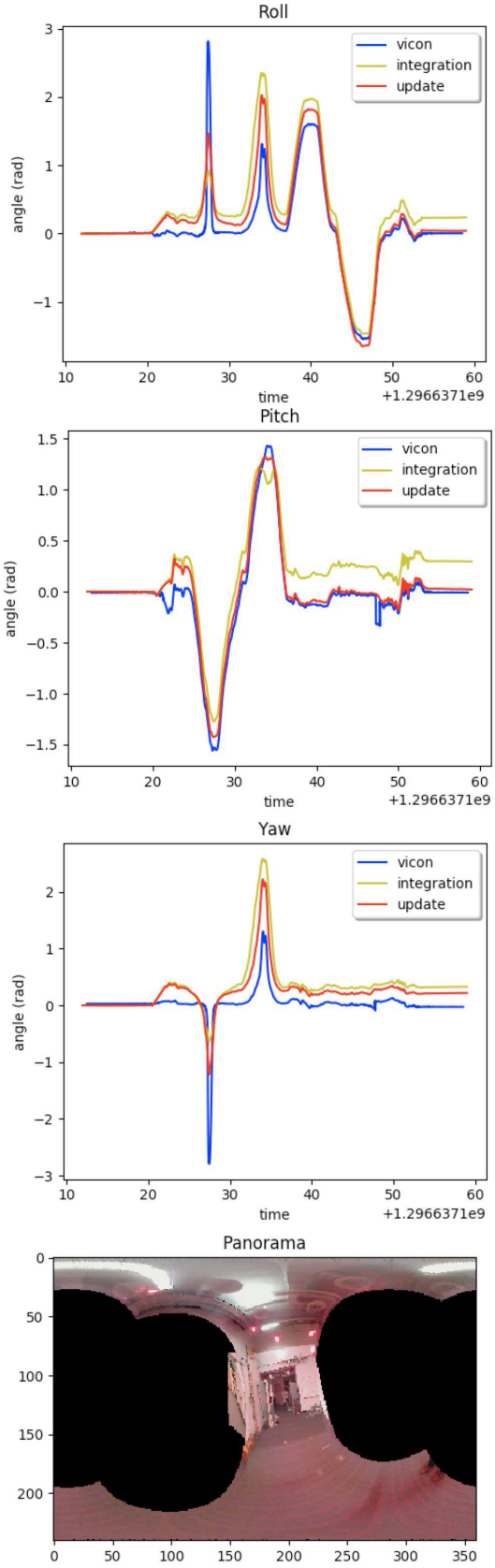

Fig. 6: Results for dataset 2. Top to bottom: Roll, Pitch, Yaw, Panorama.

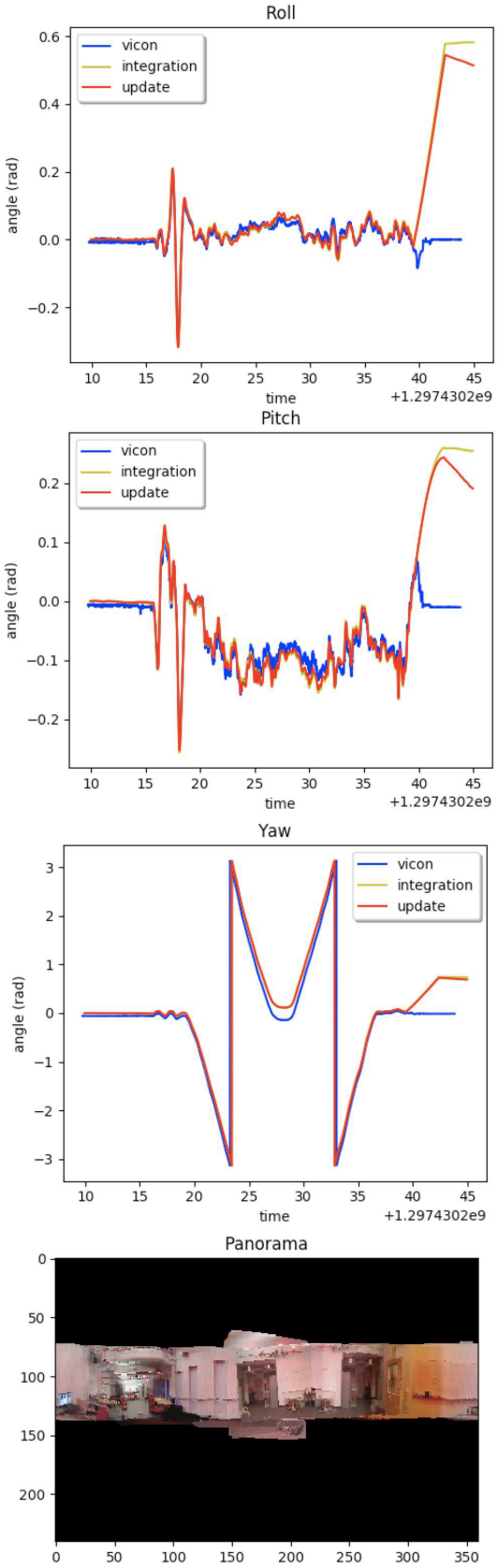

Fig. 7: Results for dataset 8. Top to bottom: Roll, Pitch, Yaw, Panorama.

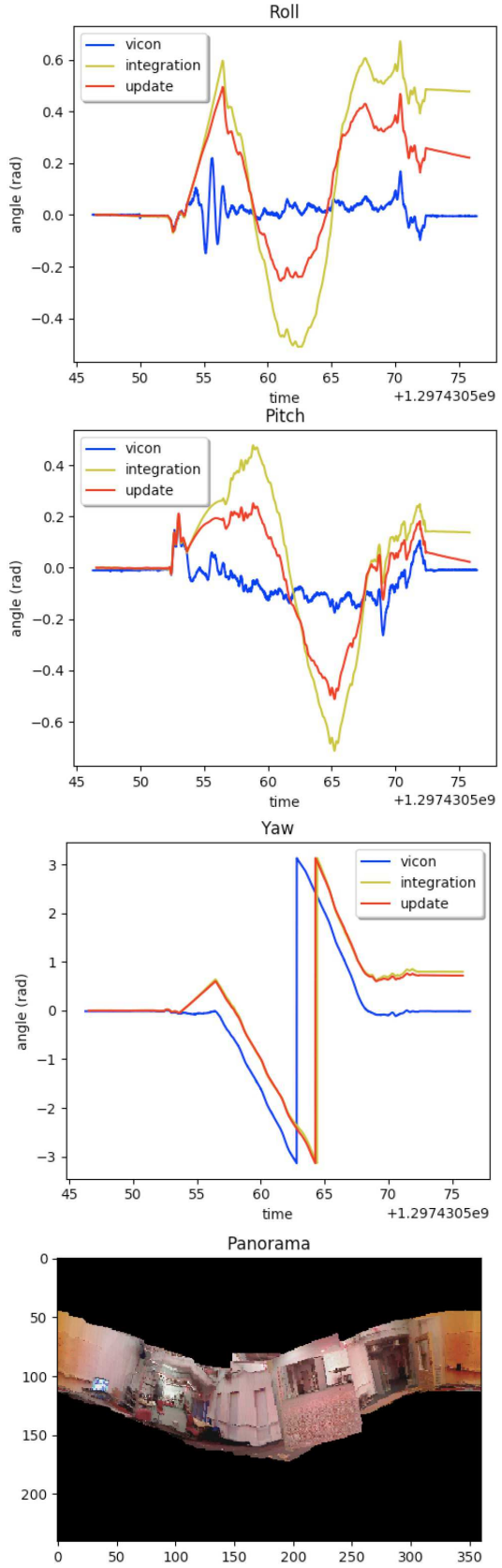

Fig. 8: Results for dataset `9`. Top to bottom: Roll, Pitch, Yaw, Panorama.

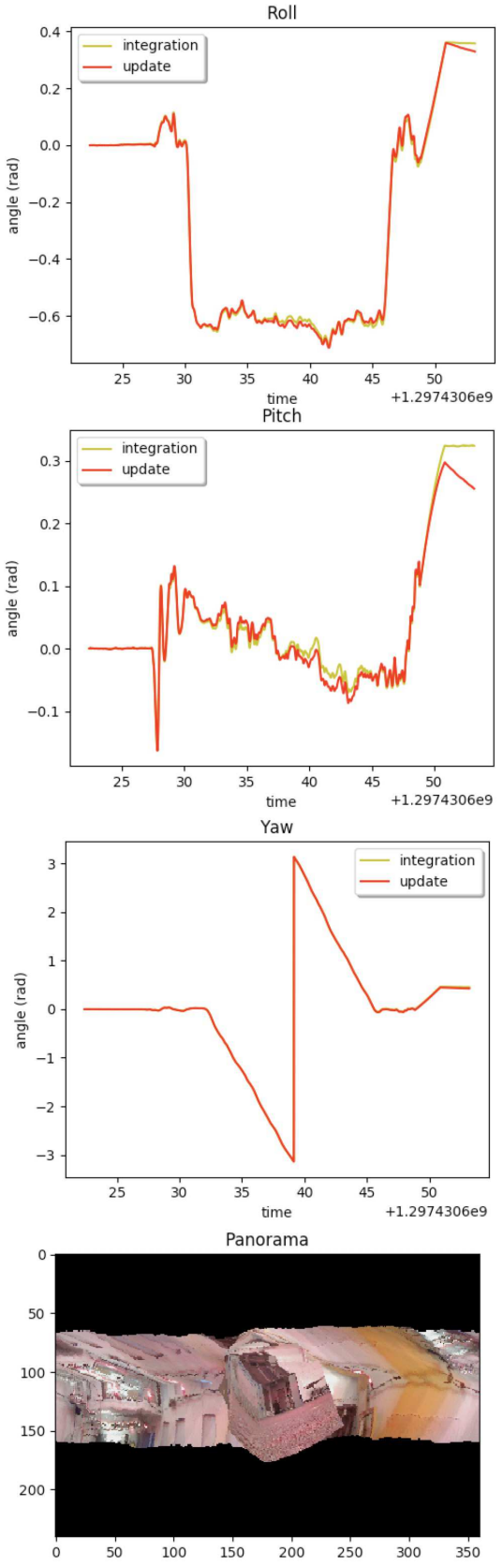

Fig. 9: Results for dataset `10`. Top to bottom: Roll, Pitch, Yaw, Panorama.

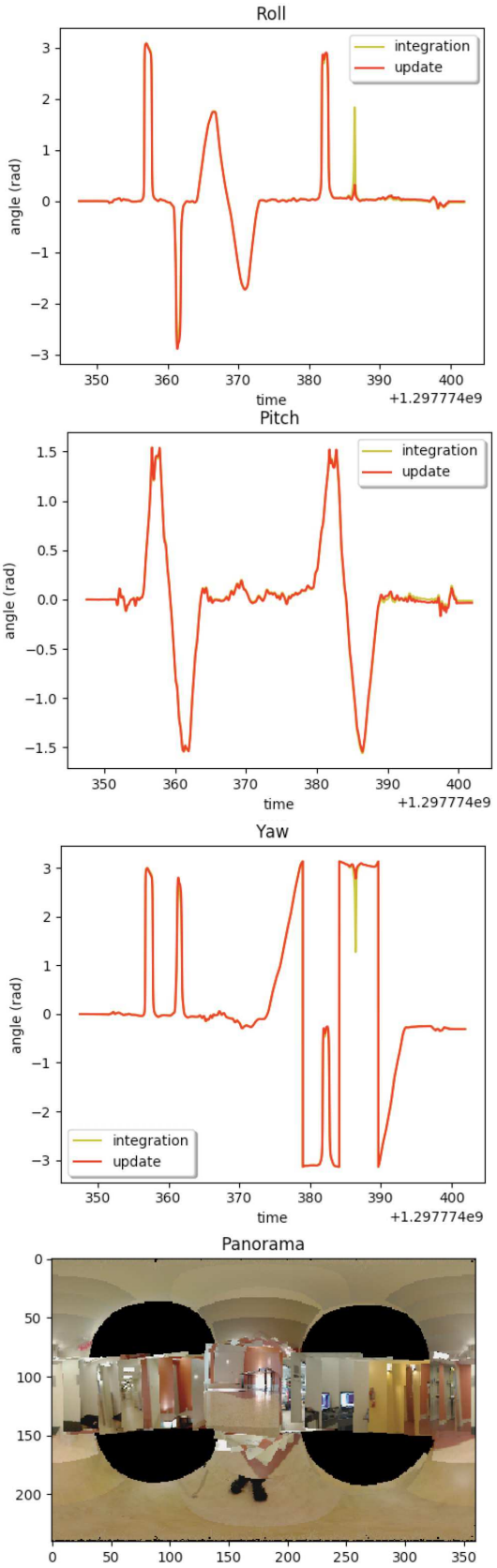

Fig. 10: Results for dataset `11`. Top to bottom: Roll, Pitch, Yaw, Panorama.

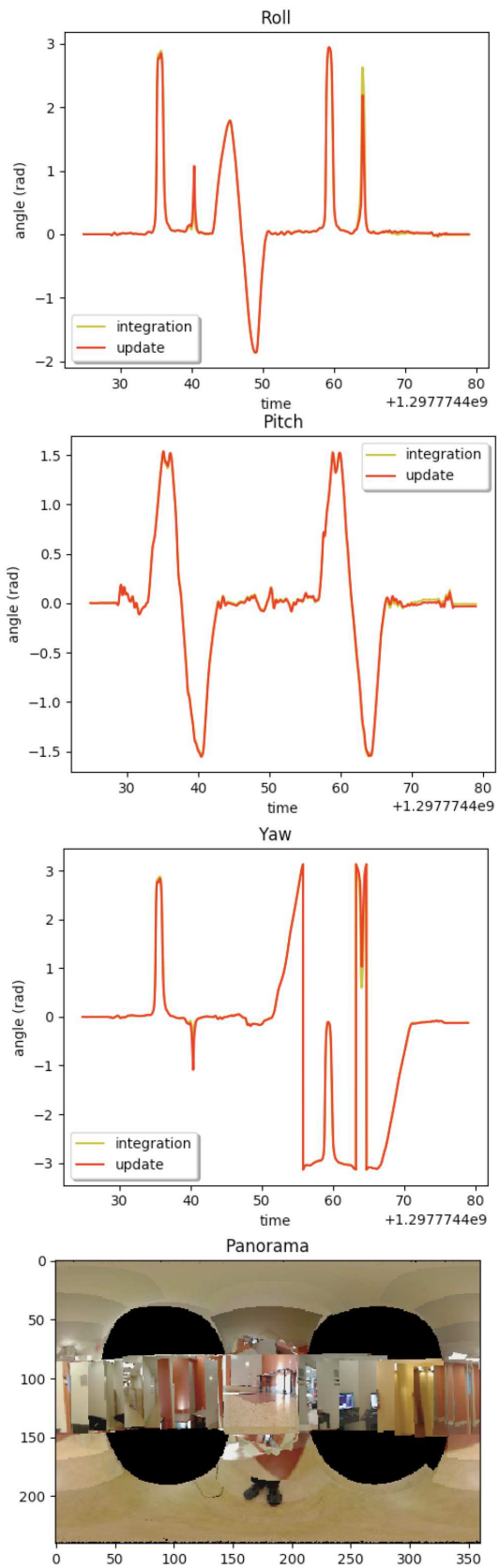

Fig. 11: Results for dataset `12`. Top to bottom: Roll, Pitch, Yaw, Panorama.

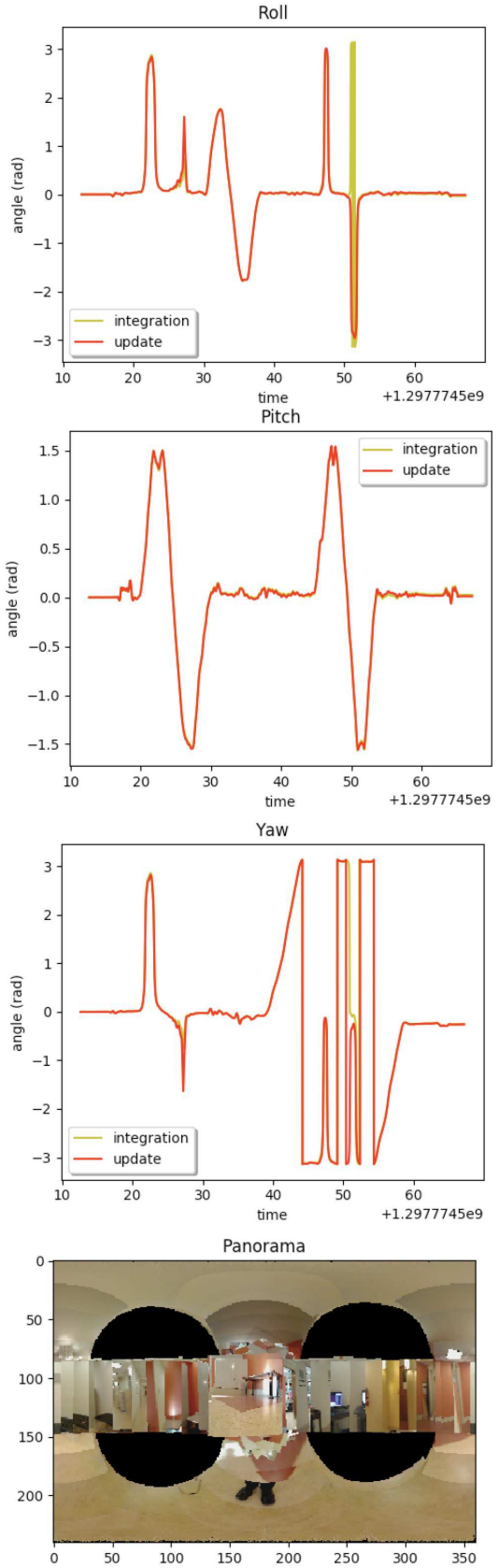

Fig. 12: Results for dataset `13`. Top to bottom: Roll, Pitch, Yaw, Panorama.