

Orientation Tracking and Panorama Construction based on IMU Data

Zejia Wu

ECE Department

UC San Diego

zew024@ucsd.edu

Abstract—This report presents a method for tracking the 3D orientation of a rotating body using data from an Inertial Measurement Unit (IMU) and constructing panoramic images from camera frames aligned using estimated orientations. The approach consists of IMU calibration to correct biases, quaternion-based orientation estimation using projected gradient descent, and panorama reconstruction through cylindrical projection. The orientation estimation process is validated against VICON ground truth data, and the reconstructed panoramic images demonstrate the effectiveness of the proposed method. Results indicate that the optimization approach successfully minimizes estimation errors, with some limitations due to sensor noise and local optimum.

Index Terms—IMU Calibration, Orientation Tracking, Projected Gradient Descent, and Panorama Stitching.

I. INTRODUCTION

This project focuses on estimating the 3D orientation of a rotating body using IMU data and generating panoramic images from captured camera frames. Orientation tracking is crucial in robotics, augmented reality, and autonomous navigation, where accurate spatial awareness is essential for interaction and control.

Achieving precise orientation estimation is challenging due to sensor biases, noise, and integration drift in IMU measurements. Errors in orientation tracking directly impact panorama quality, requiring careful calibration and data fusion. Additionally, the optimization of the orientation trajectory presents challenges, as the problem involves non-linear constraints due to the unit-norm quaternion requirement. Gradient-based methods must ensure stability while preventing convergence to incorrect local minima. The process also demands efficient temporal alignment of sensor data and accurate handling of discontinuities in rotation representation.

Our approach begins with IMU calibration to correct sensor biases using ground truth from a VICON motion capture system. We then estimate the orientation trajectory using a projected gradient descent algorithm, minimizing errors in motion and observation models while enforcing unit quaternion constraints. Finally, we construct panoramas by aligning images based on estimated orientations, ensuring spatial coherence across frames.

II. PROBLEM FORMULATION

A. Orientation Estimation

Given IMU data consisting of angular velocity $\omega_t = [\omega_t^x, \omega_t^y, \omega_t^z]^T$ and linear acceleration $\mathbf{a}_t = [a_t^x, a_t^y, a_t^z]^T$, we want to estimate the orientation of the body represented as a unit quaternion \mathbf{q}_t .

Motion model. The orientation evolution is governed by the quaternion kinematics model:

$$\mathbf{q}_{t+1} = \mathbf{q}_t \circ \exp([0, \tau_t \omega_t^T / 2]^T), \quad (1)$$

where $\exp(\cdot)$ is the quaternion exponential map, and τ_t is the time step difference. Both τ_t and ω_t are available in the IMU data.

Observation model. The observed acceleration in the IMU frame should align with gravity when transformed using the body orientation quaternion:

$$[0, \mathbf{a}_t^T]^T = \mathbf{q}_t^{-1} \circ [0, 0, 0, \mathbf{g}] \circ \mathbf{q}_t, \quad (2)$$

where $\mathbf{g} = -9.81$ is the gravity constant and \mathbf{a}_t are the IMU accelerometer readings.

Optimization. In the above models, the only set of variables that we need to optimize is the orientation quaternions $\{\mathbf{q}_t\}_{t=1}^T$, while other information can be obtained from IMU data. Hence we can construct the following cost function:

$$c(\mathbf{q}_{1:T}) = \frac{1}{2} \sum_{t=0}^{T-1} \|2 \log(\mathbf{q}_{t+1}^{-1} \circ (\mathbf{q}_t \circ \exp([0, \tau_t \omega_t^T / 2]^T)))\|_2^2 + \frac{1}{2} \sum_{t=1}^T \| [0, \mathbf{a}_t^T]^T - \mathbf{q}_t^{-1} \circ [0, 0, 0, \mathbf{g}] \circ \mathbf{q}_t \|_2^2 \quad (3)$$

The optimization problem can be formulated as

$$\min_{\mathbf{q}_{1:T}} c(\mathbf{q}_{1:T}), \text{ s.t. } \|\mathbf{q}_t\|_2^2 = 1, \forall t \in \{1, 2, \dots, T\}. \quad (4)$$

1) Panorama Construction: For each pixel point (i, j) on the captured image, we can get its 3D position in the camera frame (x^c, y^c, z^c) by assuming it is on a sphere with radius 1. Given the estimated body rotation R and translation p with respect to the world frame, we can convert (x^c, y^c, z^c) to the world frame:

$$[x^w, y^w, z^w]^T = R[x^c, y^c, z^c]^T + p. \quad (5)$$

By projecting all the points to a cylinder in the world frame, combining and blending all the points from all the captured images, and unwrapping the cylinder, we can get a panorama of the environment corresponding to the movement of the IMU-Camera system.

III. TECHNICAL APPROACH

Our approach consists of three main steps: IMU calibration, quaternion-based orientation estimation using projected gradient descent, and panorama synthesis.

A. IMU Calibration

To ensure accurate sensor readings, we correct biases and determine scale factors using a static calibration phase. For the 10-bit A/D converter with the reference voltage (V_{ref}), the scale factor is given by:

$$s = \frac{V_{\text{ref}}}{1023 \times \text{sensitivity}}, \quad (6)$$

where the unit of the sensitivity is mV/g. The IMU raw data (r_a corresponding to acceleration and r_ω corresponding to angular velocity) and the calibrated data (acceleration a and angular velocity ω) have the following relationships:

$$\begin{aligned} a &= (r_a - b_a) \times s_a \\ \omega &= (r_\omega - b_\omega) \times s_\omega \times \frac{\pi}{180}, \end{aligned} \quad (7)$$

where s_a and s_ω are scale factors, and b_a and b_ω are biases.

For calculating the bias, we utilize the fact that the angular velocity at the start should be $[0, 0, 0]^T$ and the linear acceleration should be $[0, 0, 1]^T$. Hence, for a few time steps at the beginning, we have

$$\begin{aligned} b_a &= r_a - [0, 0, \frac{1}{s_a}]^T \\ b_\omega &= r_\omega. \end{aligned} \quad (8)$$

Hence we can calculate the bias by taking the average estimation over the starting ~ 100 points using the above equations.

Finally, the gyroscope data is converted to radians per second, while accelerometer data is adjusted to reflect gravitational acceleration. This calibration ensures precise motion tracking and enhances the reliability of subsequent orientation estimates.

B. Orientation Estimation

Orientation tracking is formulated as a constrained optimization problem as shown above, ensuring unit quaternion constraints are maintained. The cost function integrates motion and observation model errors, with optimization performed iteratively using projected gradient descent. The steps include:

1. Gradient Computation: The cost function's gradient is calculated to guide the optimization process.

$$g^k = \nabla_{q_{1:T}^k} c(q_{1:T}^k).$$

2. Projection onto Tangent Space: The gradient is projected onto the tangent plane of the unit quaternion manifold to maintain constraints.

$$g_t^k = g^k - \langle g^k, q_{1:T}^k \rangle q_{1:T}^k.$$

3. Gradient Descent Update: The quaternion is updated along the projected gradient direction.

$$q_{1:T}^{k,\text{tmp}} = q_{1:T}^k - \alpha^k g_t^k.$$

4. Re-projection onto the Unit Sphere: The updated quaternion is normalized to enforce unit norm constraints.

$$q_{1:T}^{k+1} = \frac{q_{1:T}^{k,\text{tmp}}}{\|q_{1:T}^{k,\text{tmp}}\|}.$$

This method iteratively refines the orientation estimates while avoiding drift and numerical instability.

C. Panorama Synthesis

The estimated orientations are used to align camera images in a panoramic frame. The process involves:

Spherical projection. Image pixels are mapped to latitude-longitude coordinates based on camera parameters. Suppose that the pixel coordinates of a point on the image are (i, j) , we can normalize it so that the origin of the frame lies at the center of the image:

$$\begin{aligned} x_{\text{norm}} &= \frac{i}{w/2} - 1 \\ y_{\text{norm}} &= \frac{j}{h/2} - 1 \end{aligned}$$

where w and h are the width and height of the image. Given the field of view of the camera, denoted as (f_x, f_y) , by geometry and perspective imaging principle, we have

$$\begin{aligned} \lambda &= \arctan(x_{\text{norm}} \tan(\frac{f_x}{2})) \\ \phi &= \arctan(y_{\text{norm}} \tan(\frac{f_y}{2})) \end{aligned}$$

where λ and ϕ are longitude and latitude in the spherical coordinate system. By doing this, we can map every pixel on the image onto a sphere centered at the camera in its own frame.

Transformation to world coordinates. Assuming that the radius of the sphere is 1, pixels can be converted to a Cartesian coordinate system in the body frame and then transformed to the world frame using the estimated camera-to-world transformation.

$$\begin{bmatrix} x^w \\ y^w \\ z^w \end{bmatrix} = R \begin{bmatrix} 0 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \sin \phi \cos \lambda \\ \sin \phi \sin \lambda \\ \cos \phi \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0.1 \end{bmatrix}, \quad (9)$$

where R is the rotation matrix corresponding to the quaternion we estimated, and the matrix on the right side of R is to change the coordinate frame from the camera convention to the robotics convention.

Cylindrical projection. For seamless panorama stitching, we need to project all the pixel points from the world system onto a cylindrical surface. Assuming that the radius of the cylinder is 1, we have the cylindrical coordinates

$$(\rho, \phi, z) = (1, \text{np.arctan } 2(\frac{y}{x}), z)$$

in which the range of $\text{np.arctan } 2(\cdot)$ is $(-\pi, \pi)$ since it can distinguish quadrants. By unwrapping the cylinder, the position of the point on the panorama should be $(\frac{\phi+\pi}{2\pi}, \frac{z+1}{2})$.

Image merging. Given the position of all the pixel points in all the captured images on the panorama, we can blend them to generate the final panorama, ensuring spatial consistency.

IV. RESULTS

In the sections below, we will use results from training set 1 as illustrative examples. The results of the test set will be shown in the last subsection. Full results on all datasets are in the “./figures” subfolder of the submitted code. Please see “README.md” for a more detailed explanation.

A. IMU Calibration and Forward Integration

IMU calibration significantly reduced sensor biases, improving raw data accuracy. Figure 1 shows the raw IMU readings and the processed data. We can see that the processed data all start from zero angular velocities and accelerations except for a_z , which is g in the rest state (+1 in our data since the unit is g).

To test the correctness of our data calibration, we did a simple forward integration of the orientation using the motion model in Equation 1 and compared them with the VICON ground truth data. Moreover, by applying integrated quaternions to the world frame acceleration $[0, 0, g]^T$ as in the observation model Equation 2, we can get the estimation of linear accelerations in the body frame and compare them with the IMU readings, which shows the observation error of this integration method. The results are shown in Figure 2. The comparison between estimated and ground-truth roll, pitch, and yaw angles showed close alignment, with minor discrepancies caused by sensor noise and integration drift, meaning that our calibration is correct and the gyroscope data and the accelerometer data are consistent.

B. Orientation Estimation

By applying projected gradient descent optimization on the cost function Equation 3, we can get a better estimation of the orientation quaternions. The optimization process demonstrated smooth convergence, as indicated by the loss function trends in Figure 3. The motion model error was initially low (since we used the integration results as the start point of estimation) but increased slightly due to the optimization adjusting estimates to better match the observation model. The observation error decreased over iterations, confirming that the estimates became more consistent with accelerometer measurements.

Figure 4 shows the same comparison with those in Figure 2 but with the optimized estimation of orientations. We can

see that the gap between the estimation and the ground truth as well as the gap between the estimated accelerations and the IMU accelerometer readings both become much smaller, which shows that the projected gradient descent algorithm optimized the orientation trajectory effectively, yielding stable and consistent estimates.

C. Panorama Reconstruction

The ground truth panoramic image for the dataset 1, which is constructed using the VICON rotation matrices, is shown in the first subfigure of Figure 5, while the panorama created by using our estimated orientation trajectory is shown in the second subfigure.

The constructed panoramic images closely resembled those using ground-truth orientations, validating the accuracy of estimated orientations. Some misalignment was observed in datasets with higher sensor noise, affecting the clarity of stitched images. The cylindrical projection method effectively preserved spatial coherence, ensuring seamless image merging.

D. Results of Test Sets

In this section, we will show the results of the test set. Figure 6 (test set 10) and Figure 7 (test set 11) show similar results as in Figure 2 (training set 1), suggesting that the IMU data is calibrated consistently and accurately.

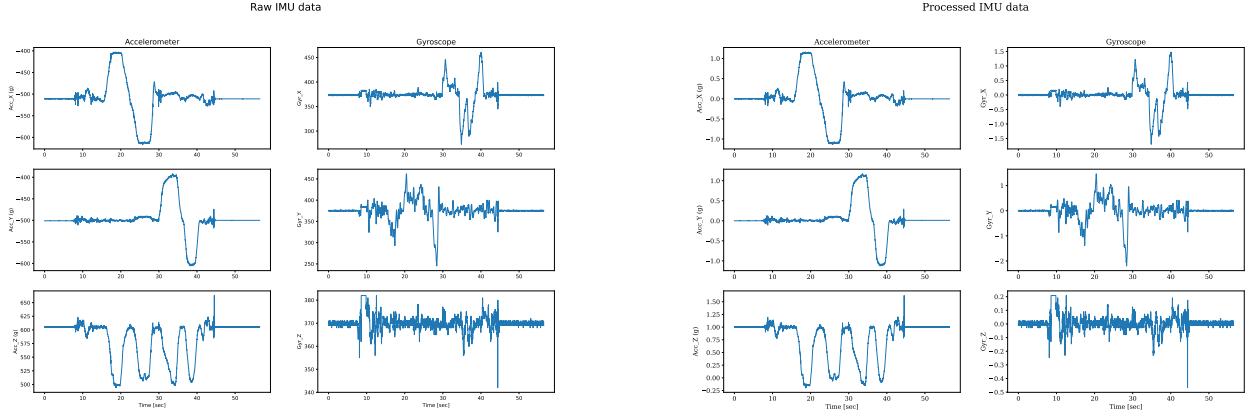
By applying optimization, the losses decrease smoothly, as shown in Figure 8 (test set 10) and Figure 9 (test set 11).

After optimization, the raw, pitch, and yaw, as well as the observation error are shown in Figure 10 (test set 10) and Figure 11 (test set 11), demonstrating smaller observation errors (we don’t have ground truth for the comparison of roll, pitch, and yaw).

The test datasets lacked ground-truth comparisons, so the accuracy of estimated orientations was inferred from the panorama quality. In the test set 10, as shown in Figure 12, the horizontal rotation resulted in a well-aligned panorama with minimal distortion. In the test set 11, as shown in Figure 13, which included vertical and horizontal rotations, the resulting panorama exhibits smooth transitions but shows slight vertical misalignment in some sections, likely due to minor errors in orientation estimates.

V. CONCLUSION

This project demonstrated the effectiveness of a constrained optimization approach for orientation tracking using IMU data. The estimated orientations were successfully utilized for panoramic image reconstruction. Future improvements could include filtering techniques such as an Extended Kalman Filter (EKF) or incorporating additional sensor fusion methods to enhance robustness.



(a) Raw IMU Data

(b) Processed IMU Data

Fig. 1: The comparison between the raw IMU readings and the processed IMU data, which was obtained by subtracting the bias and multiplying by the scale factor. The left column in each subplot shows the linear acceleration while the right column shows the angular velocities.

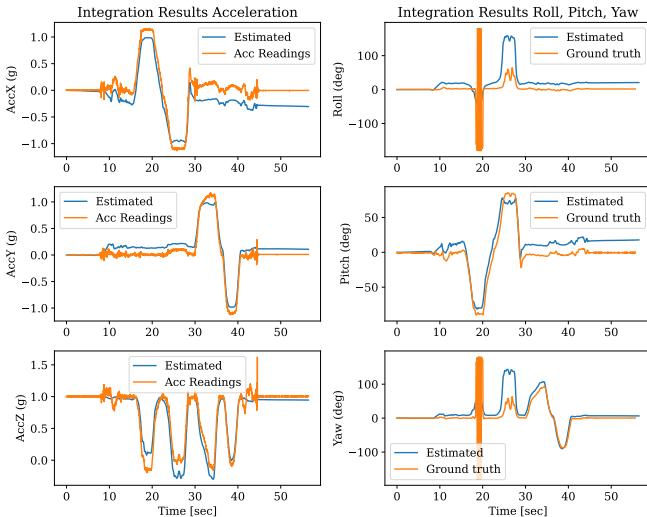


Fig. 2: Comparison with Ground Truth using **integration estimations**. The right column is the comparison between roll, pitch, and yaw obtained from the forward integration and those from ground truth, while the left column shows the **estimation** of body frame accelerations vs. accelerometer readings of IMU, i.e. observation errors for the integration method.

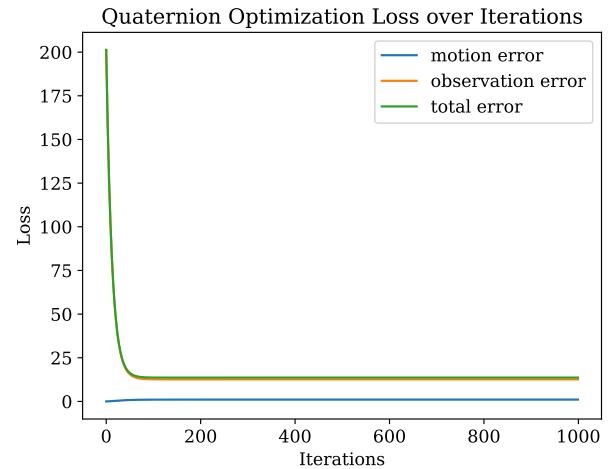


Fig. 3: The training loss trends over iterations.

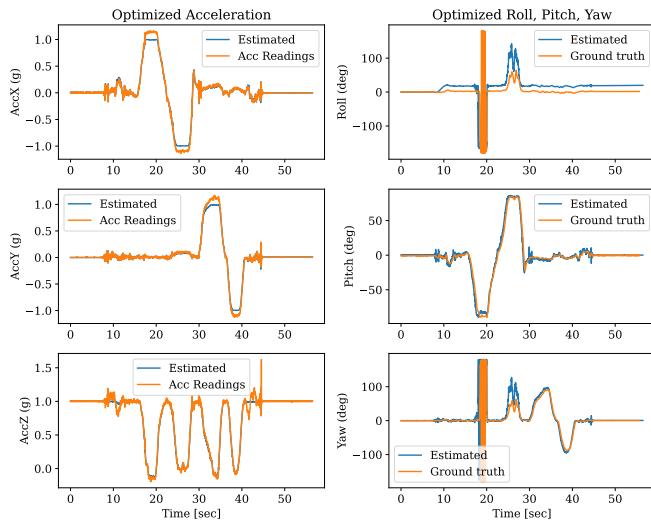
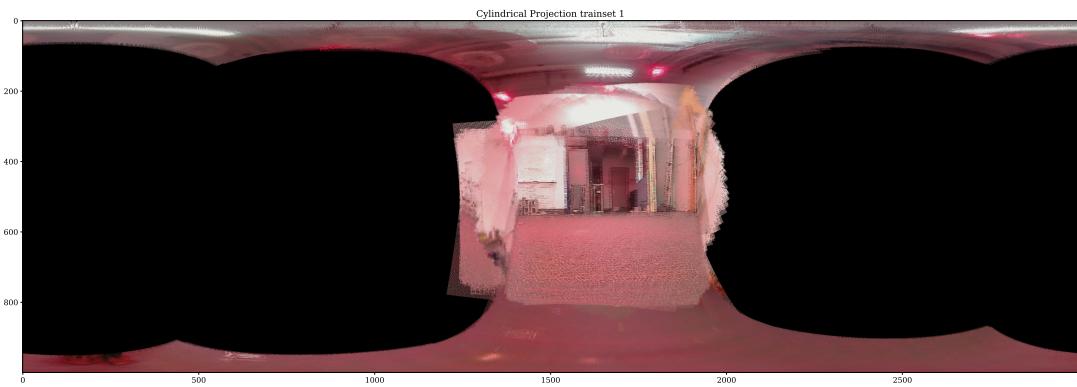


Fig. 4: Comparison with Ground Truth using **optimized estimations**. The right column is the comparison between roll, pitch, and yaw obtained from the optimization and those from ground truth, while the left column shows the **estimation** of body frame accelerations vs. accelerometer readings of IMU, i.e. observation errors for the optimization method.



(a) Ground Truth Panorama



(b) Estimated Panorama

Fig. 5: The comparison between the ground truth panorama for the dataset 1 and the panorama constructed using the estimated orientation trajectory.

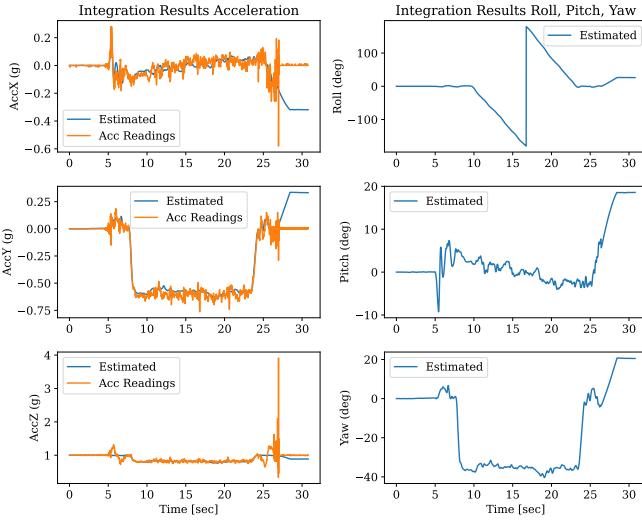


Fig. 6: Estimations obtained by **forward integration** for **test set 10**. The right column shows the roll, pitch, and yaw obtained from the forward integration, while the left column shows the **estimation** of body frame accelerations vs. accelerometer readings of IMU, i.e. observation errors for the integration method.

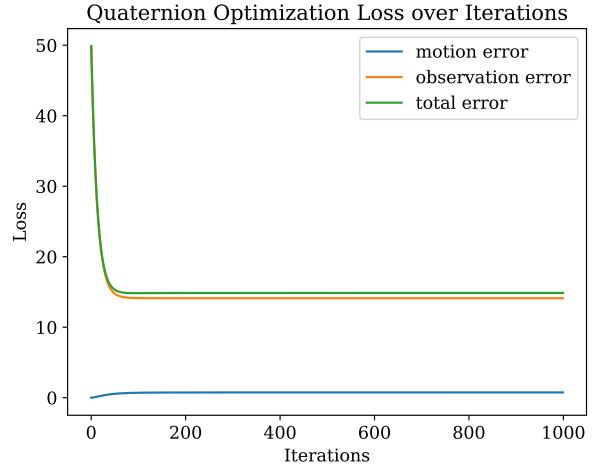


Fig. 8: The training loss trends over iterations for **test set 10**.

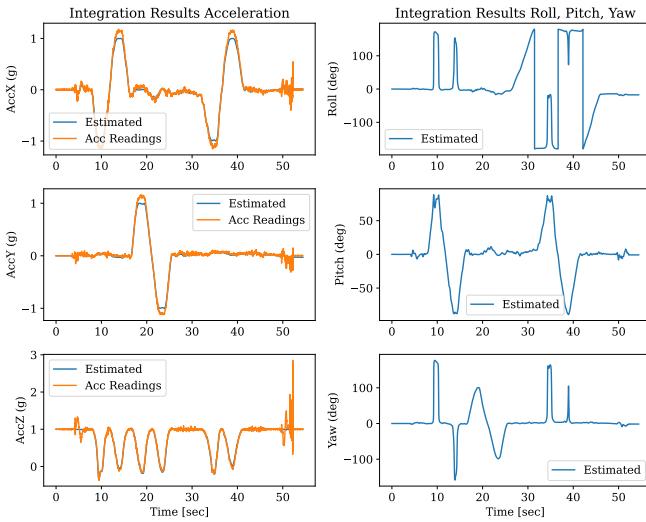


Fig. 7: Estimations obtained by **forward integration** for **test set 11**. The right column shows the roll, pitch, and yaw obtained from the forward integration, while the left column shows the **estimation** of body frame accelerations vs. accelerometer readings of IMU, i.e. observation errors for the integration method.

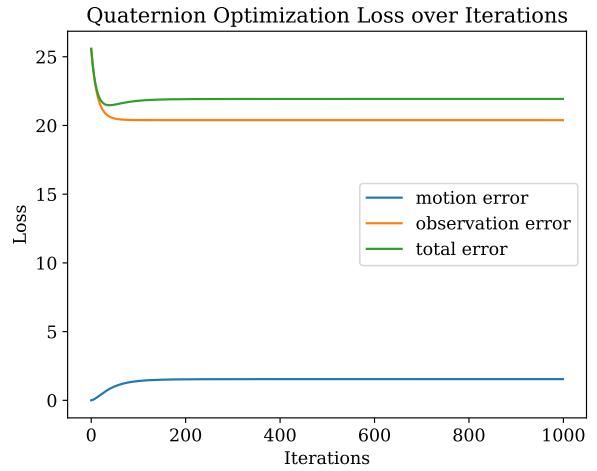


Fig. 9: The training loss trends over iterations for **test set 11**.

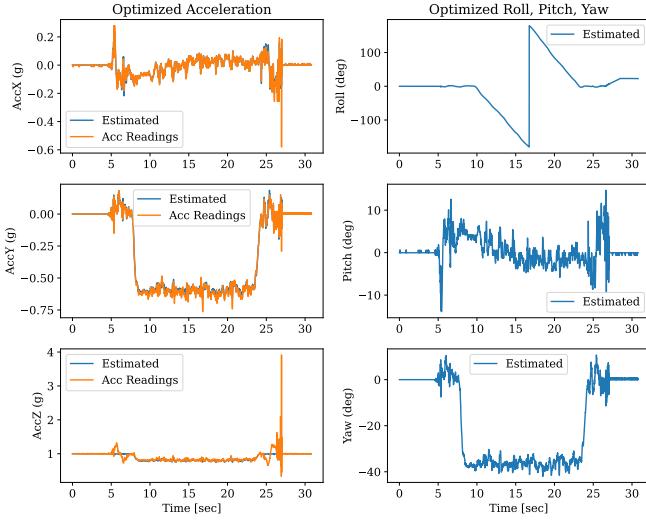


Fig. 10: Estimations obtained by **optimization** for **test set 10**. The right column shows the roll, pitch, and yaw obtained from the optimization, while the left column shows the **estimation** of body frame accelerations vs. accelerometer readings of IMU, i.e. observation errors for the optimization method.

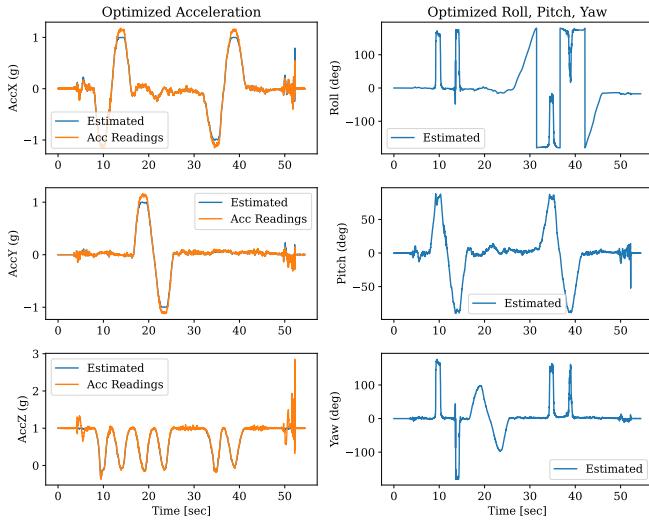


Fig. 11: Estimations obtained by **optimization** for **test set 11**. The right column shows the roll, pitch, and yaw obtained from the optimization, while the left column shows the **estimation** of body frame accelerations vs. accelerometer readings of IMU, i.e. observation errors for the optimization method.



Fig. 12: The panorama constructed using the estimated orientation trajectory for **test set 10**.

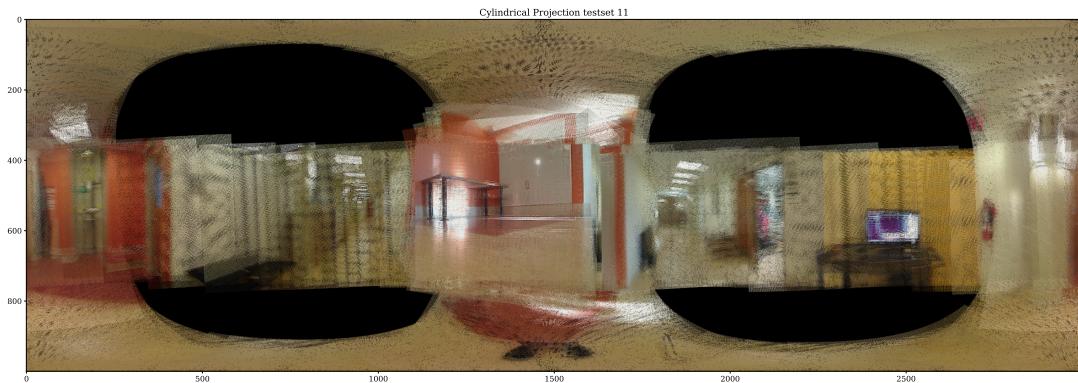


Fig. 13: The panorama constructed using the estimated orientation trajectory for **test set 11**.