Project 1: Orientation Tracking

Chen-Yeh Lin
Department of Electrical and Computer Engineering
University of California, San Diego
PID: A69036794

***Abstract*—This project implements an orientation tracking system using data from an Inertial Measurement Unit (IMU) to estimate the 3D orientation of a rotating body. The approach uses motion model and observation models combined with projected gradient descent to optimize the orientation trajectory by minimizing errors between predicted and measured accelerations. The optimized orientation estimates are then used to construct panoramic images by stitching together camera images based on the computed orientations, demonstrating the practical application of the tracking system in visual mapping.**

I. Introduction

Accurately tracking the three-dimensional orientation of a rotating body is a fundamental challenge in robotics and computer vision. This project aims to achieve precise orientation tracking using data from an Inertial Measurement Unit (IMU) and validating the results with a VICON motion capture system. Orientation tracking plays a crucial role in various applications, including autonomous navigation, augmented reality, and environmental mapping, where precise sensor orientation is essential for spatial awareness.

Despite the high accuracy of systems like VICON, their high cost and limited applicability to controlled environments make them impractical for many real-world scenarios. In contrast, IMU sensors provide a more affordable and widely deployable alternative. However, IMU-based orientation tracking suffers from inherent biases, noise, and drift, making it less reliable without proper calibration and optimization techniques.

To address these challenges, we propose a hybrid approach that integrates classical sensor calibration techniques with modern optimization methods. Our method first compensates for biases and scaling factors in IMU measurements through careful calibration. Then, we formulate the orientation estimation problem using a motion model and observation model, which is optimized using a projected gradient descent algorithm. This optimization minimizes the error between predicted and measured accelerations while ensuring quaternion unit norm constraints.

To demonstrate the effectiveness of our approach, we apply our optimized orientation estimates to panoramic image stitching, where camera frames are aligned based on computed orientations. This provides a practical validation of the tracking accuracy and showcases the real-world applicability of our method.

II. Problem Formulation

The orientation tracking problem can be formulated as estimating the body-frame orientation over time using measurements from an IMU sensor. At each discrete time step t, we have access to two key measurements: the angular velocity $\boldsymbol{\omega}_t \in R^3$ and the linear acceleration $\mathbf{a}_t \in R^3$. We represent the body-frame orientation at time t using a unit quaternion $\mathbf{q}_t \in H^*$, where H^* denotes the space of unit quaternions.

The evolution of orientation follows a quaternion kinematics motion model based on angular velocity measurements:

$$\mathbf{q}_{t+1} = f(\mathbf{q}_t, \tau_t \boldsymbol{\omega}_t) := \mathbf{q}_t \circ \exp([0, \tau_t \boldsymbol{\omega}_t / 2]) \quad (1)$$

where τ_t represents the time difference between consecutive measurements, $\circ$ denotes quaternion multiplication, and $\exp(\cdot)$ is the quaternion exponential function mapping axis-angle representations to quaternions.

Since the body undergoes pure rotation, it experiences only gravitational acceleration in the world frame. This leads to our observation model relating measured acceleration to orientation:

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

where g represents gravitational acceleration.

We formulate the orientation estimation as an optimization problem minimizing a cost function that combines motion and observation errors:

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

subject to the unit norm constraints:

$$\|\mathbf{q}_t\|_2 = 1, \quad \forall t \in \{1, 2, ..., T\} \quad (4)$$

This constrained optimization problem is solved using projected gradient descent. At each iteration k, we update the quaternion trajectory according to:

$$\mathbf{q}_{k+1} = \Pi_{H^*}(\mathbf{q}_k - \alpha \nabla c) \quad (5)$$

where α is the learning rate and Π_{H^*} represents projection onto the space of unit quaternions:

$$\Pi_{H^*}(\mathbf{q}) = \frac{\mathbf{q}}{\|\mathbf{q}\|_2} \quad (6)$$

III. Technical Approach

A. *Orientation Tracking Approach*

The orientation tracking problem involves estimating the body's orientation trajectory using inertial measurement unit (IMU) data through a sophisticated optimization framework. Our approach consists of several key stages: data preprocessing, initial orientation estimation, and quaternion trajectory optimization.

Data Preprocessing and IMU Calibration: We begin by addressing sensor biases and calibrating the IMU measurements. The initial static period of the dataset, typically the first few hundred samples where the sensor remains stationary, is used to estimate and remove systematic biases from both accelerometer and gyroscope readings. By computing the mean acceleration and angular velocity during this stationary period, we can subtract these bias values from subsequent measurements.

The acceleration data is scaled to gravity units, with careful attention to coordinate system alignment. Specifically, we multiply the acceleration by the gravitational constant (9.81 m/s^2) and adjust the signs to ensure correct directional representation. For the stationary period, the expected acceleration should align with the gravitational vector $[0, 0, 1]$ in the sensor's reference frame.

Initial Quaternion Integration: An initial estimate of the orientation trajectory is obtained through direct integration of angular velocity measurements. Starting from an initial quaternion $q_0 = [1, 0, 0, 0]$ representing no rotation, we propagate the orientation using the motion model:

$$q_{t+1} = q_t \circ \exp\left(\frac{\tau_t \omega_t}{2}\right) \quad (7)$$

where τ_t represents the time difference between consecutive measurements, ω_t is the angular velocity, and $\circ$ denotes quaternion multiplication. The exponential mapping $\exp(\cdot)$ converts the angular velocity to a quaternion increment, allowing smooth orientation propagation.

Quaternion Trajectory Optimization: The core of our approach involves a constrained optimization process that refines the initial quaternion trajectory. We formulate an objective function that simultaneously minimizes two key error terms: the motion model error and the observation model error.

The motion model error captures the discrepancy between predicted and estimated consecutive orientations. This is computed by measuring the relative rotation between the predicted orientation $f(q_t, \tau_t \omega_t)$ and the subsequent estimated orientation q_{t+1}. By using the quaternion logarithm, we obtain an axis-angle representation of this rotation error.

The observation model error compares the predicted acceleration with the actual IMU acceleration measurements. Since the body undergoes pure rotation, we expect the acceleration to align with the gravitational vector when transformed into the body's reference frame.

Mathematically, this is expressed through the observation model:

$$[0, a_t] = q_t^{-1} \circ [0, 0, 0, -g] \circ q_t \quad (8)$$

Gradient Descent Optimization: We employ a projected gradient descent algorithm to minimize the combined cost function. At each iteration, we compute the gradient of the cost function with respect to the quaternion trajectory and update the quaternions while enforcing the unit norm constraint:

$$q_{k+1} = \Pi_{H^*}\left(q_k - \alpha \nabla c(q_{1:T})\right) \quad (9)$$

Here, Π_{H^*} represents the projection onto the unit quaternion space, typically achieved by normalizing the quaternion vector. The learning rate α controls the step size of the gradient descent.

The optimization process iteratively refines the quaternion trajectory, balancing the motion model prediction with the acceleration measurements. This approach effectively reduces the noise inherent in raw IMU measurements and provides a more accurate estimation of the body's orientation.

Computational Efficiency: To enhance computational performance, we leverage the JAX library's automatic differentiation and just-in-time (JIT) compilation capabilities. This allows for efficient gradient computation and vectorized operations, enabling the optimization of the quaternion trajectory within a few seconds for datasets containing hundreds of measurements.

B. *Panorama Creation*

The panorama creation process involves transforming sequences of images into a unified spherical representation using rotation data from a Vicon motion capture system. Our approach consists of several key stages: spherical coordinate mapping, rotation transformation, and image projection using equirectangular projection.

Spherical Coordinate System: The spherical coordinate system forms the foundation of our panorama creation approach. As illustrated in Figure 1, this system replaces the traditional Cartesian coordinates with three parameters: the radial distance (r) measured from the origin, the polar angle (θ) measured from the z-axis, and the azimuthal angle (ϕ) measured in the x-y plane from the x-axis. Given the camera's field of view (60° horizontal, 45° vertical), each pixel (u, v) in the input image is mapped to spherical coordinates using:

$$\theta = \frac{\pi}{2} + \frac{\text{FOV}_V}{2} \cdot \left((2\frac{u}{H}) - 1\right) \tag{10}$$

$$\phi = \frac{\text{FOV}_H}{2} \cdot \left((2\frac{v}{W}) - 1\right) \tag{11}$$

where H and W are the image height and width respectively, and FOV_H, FOV_V represent the horizontal and vertical fields of view in radians.

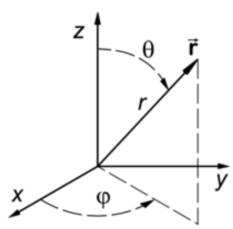


Fig. 1: Spherical coordinate system showing the relationship between Cartesian coordinates (x, y, z) and spherical coordinates (r, θ, ϕ). The radial distance (r) extends from the origin, the polar angle (θ) measures the inclination from the z-axis, and the azimuthal angle (ϕ) measures the rotation in the x-y plane.

Coordinate Frame Transformation: The transformation process involves a series of mathematical operations to properly align and project each image. First, we convert the spherical coordinates to Cartesian coordinates (x, y, z) using:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \sin\theta\cos\phi \\ \sin\theta\sin\phi \\ \cos\theta \end{bmatrix} \tag{12}$$

These coordinates are then transformed using the rotation matrix R from the Vicon system:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = R^T \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{13}$$

The transformed coordinates are converted back to spherical coordinates (θ', ϕ') using:

$$\theta' = \arccos\left(\frac{z'}{\sqrt{x'^2 + y'^2 + z'^2}}\right) \tag{14}$$

$$\phi' = \arctan 2(y', x') \tag{15}$$

Equirectangular Projection: Our panorama implementation utilizes equirectangular projection, also known as a latitude-longitude projection, to map the spherical coordinates onto a 2D plane. As shown in Figure 2, this projection method maps meridians to equally spaced vertical lines and parallels to equally spaced horizontal lines, creating a systematic mapping of the entire sphere onto a rectangular image with a 2:1 aspect ratio. This projection is particularly suitable for panoramic imagery as it preserves angular relationships and provides a natural representation of the complete spherical field of view.

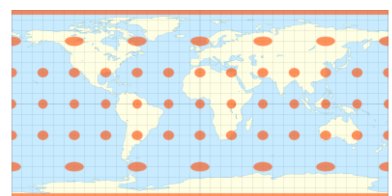


Fig. 2: Equirectangular projection of a world map, demonstrating how spherical coordinates are mapped to a rectangular image. The projection preserves angular relationships while mapping the entire sphere to a 2:1 aspect ratio rectangle.

Final Projection: The equirectangular projection maps the transformed spherical coordinates to the final panorama coordinates (u', v') through:

$$u' = \frac{\theta'}{\pi} \cdot H_{\text{panorama}} \tag{16}$$

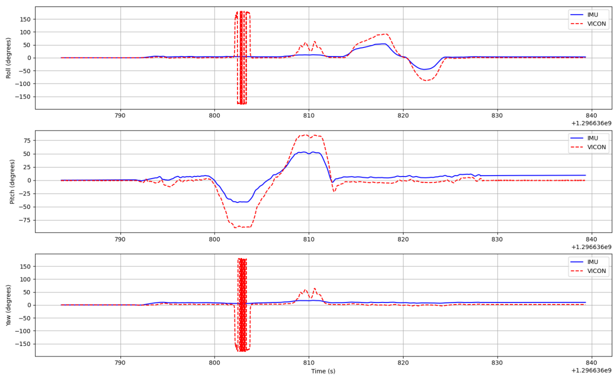
$$v' = \left(\frac{\phi' + \pi}{2\pi}\right) \cdot W_{\text{panorama}} \tag{17}$$

where H_{panorama} and W_{panorama} are set to 720×1280 pixels in our implementation, maintaining the standard 2:1 aspect ratio characteristic of equirectangular projections. The final panorama synthesis handles overlapping regions through a sequential update strategy, where newer pixel values take precedence over existing ones. This approach proves effective due to the sequential nature of the image capture process and the precise rotation data provided by the Vicon system.

IV. Results

A. Orientation Tracking Results

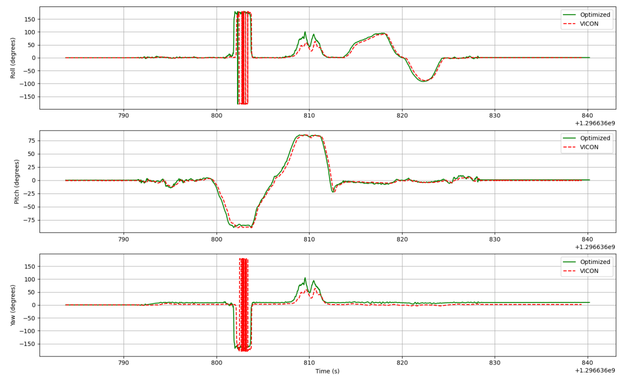
Dataset 1



(a) True Roll vs Estimated Roll

(b) True Pitch vs Estimated Pitch

(c) True Yaw vs Estimated Yaw

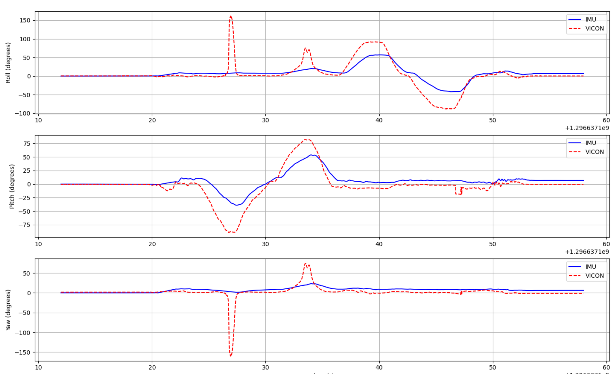


(a) True Roll vs Optimized Roll

(b) True Pitch vs Optimized Pitch

(c) True Yaw vs Optimized Yaw

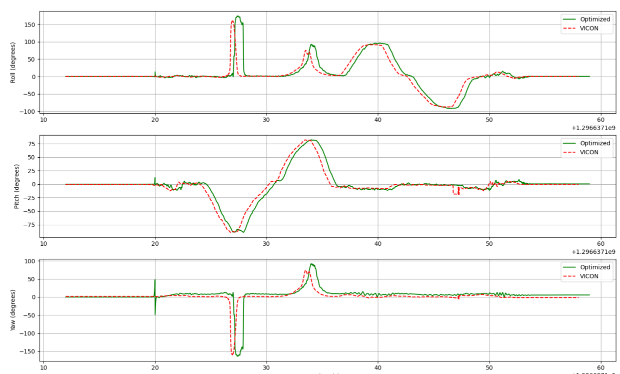
Dataset 2



(a) True Roll vs Estimated Roll

(b) True Pitch vs Estimated Pitch

(c) True Yaw vs Estimated Yaw

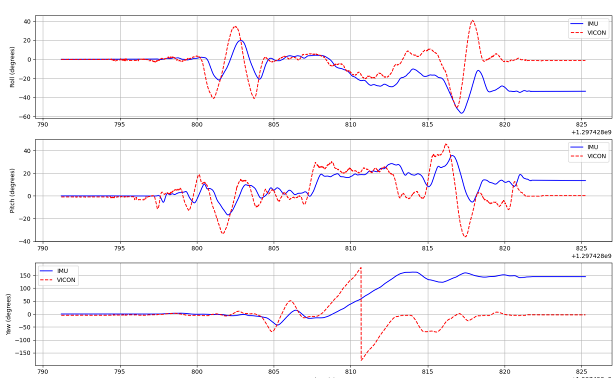


(a) True Roll vs Optimized Roll

(b) True Pitch vs Optimized Pitch

(c) True Yaw vs Optimized Yaw

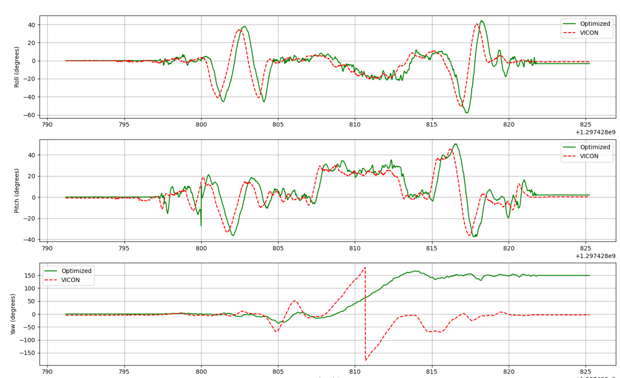
Dataset 3



(a) True Roll vs Estimated Roll

(b) True Pitch vs Estimated Pitch

(c) True Yaw vs Estimated Yaw

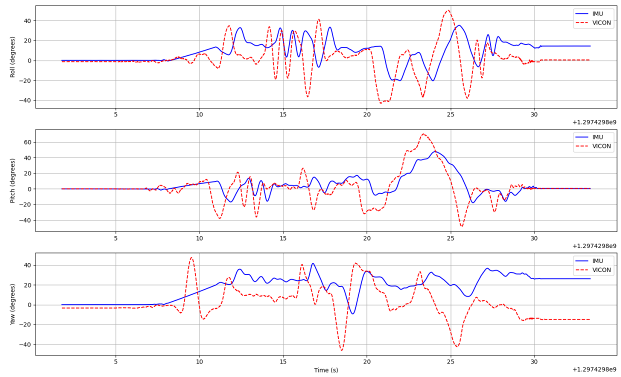


(a) True Roll vs Optimized Roll

(b) True Pitch vs Optimized Pitch

(c) True Yaw vs Optimized Yaw

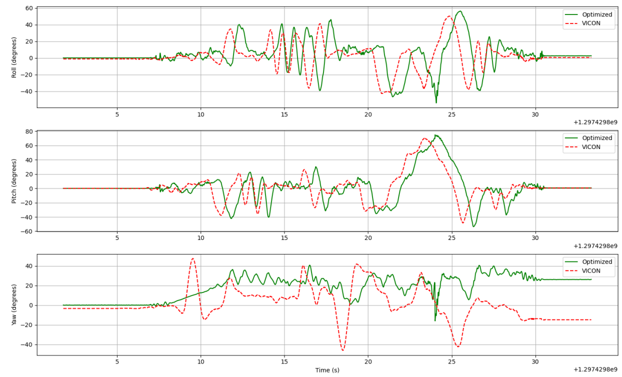
Dataset 4



(a) True Roll vs Estimated Roll

(b) True Pitch vs Estimated Pitch

(c) True Yaw vs Estimated Yaw

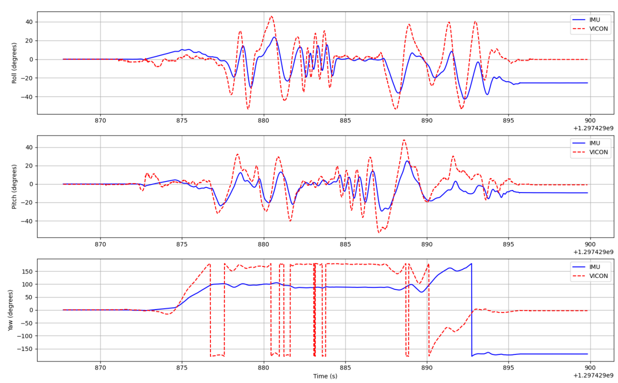


(a) True Roll vs Optimized Roll

(b) True Pitch vs Optimized Pitch

(c) True Yaw vs Optimized Yaw

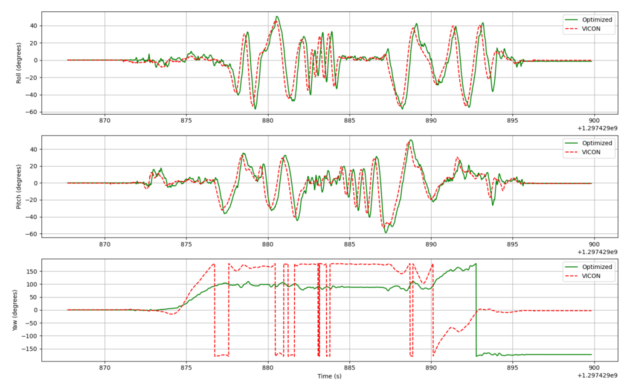
Dataset 5



(a) True Roll vs Estimated Roll

(b) True Pitch vs Estimated Pitch

(c) True Yaw vs Estimated Yaw

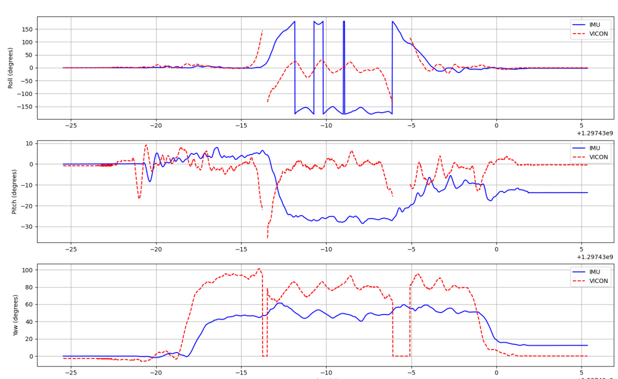


(a) True Roll vs Optimized Roll

(b) True Pitch vs Optimized Pitch

(c) True Yaw vs Optimized Yaw

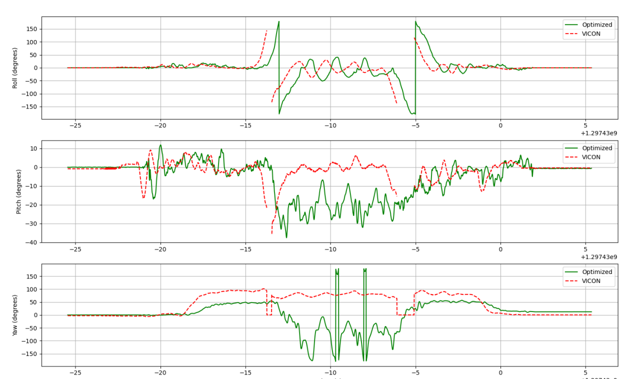
Dataset 6



(a) True Roll vs Estimated Roll

(b) True Pitch vs Estimated Pitch

(c) True Yaw vs Estimated Yaw

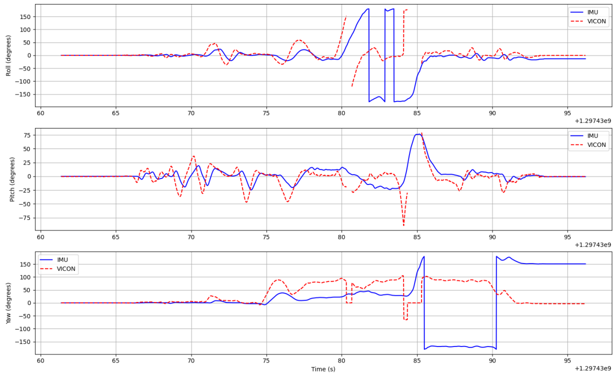


(a) True Roll vs Optimized Roll

(b) True Pitch vs Optimized Pitch

(c) True Yaw vs Optimized Yaw

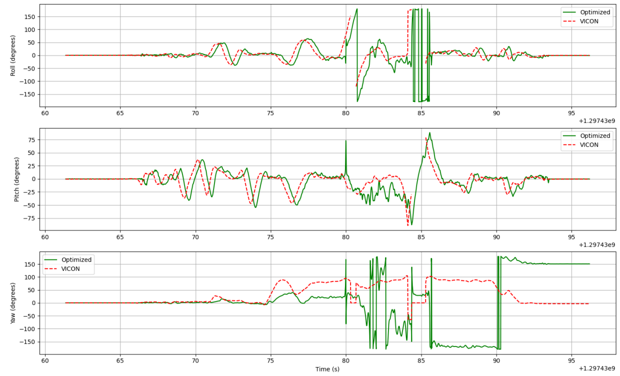
Dataset 7



(a) True Roll vs Estimated Roll

(b) True Pitch vs Estimated Pitch

(c) True Yaw vs Estimated Yaw

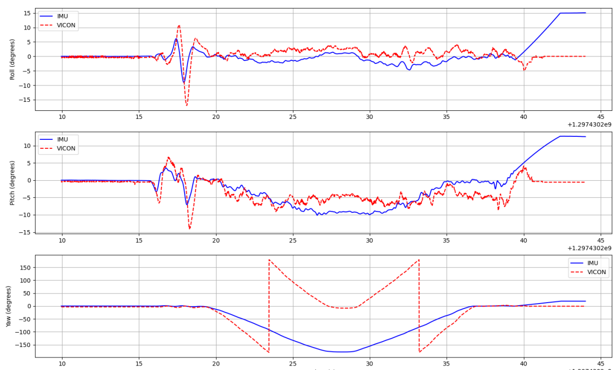


(a) True Roll vs Optimized Roll

(b) True Pitch vs Optimized Pitch

(c) True Yaw vs Optimized Yaw

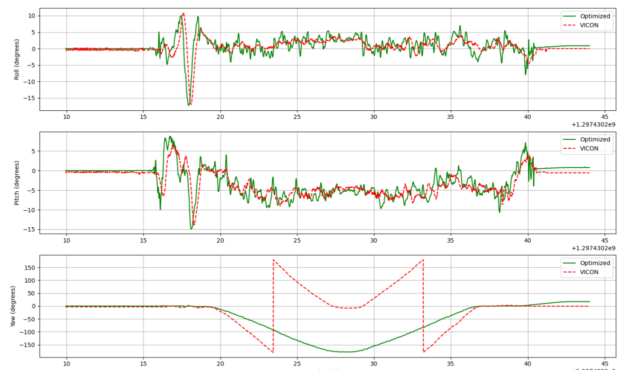
Dataset 8



(a) True Roll vs Estimated Roll

(b) True Pitch vs Estimated Pitch

(c) True Yaw vs Estimated Yaw

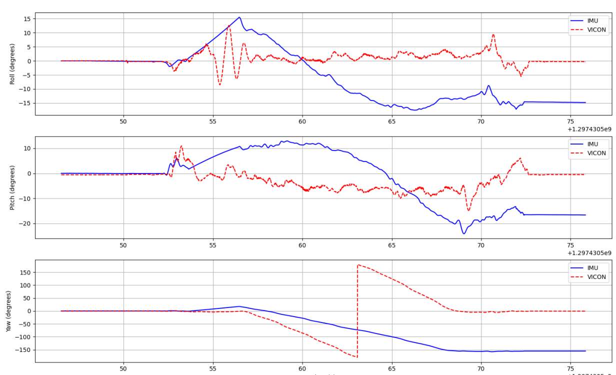


(a) True Roll vs Optimized Roll

(b) True Pitch vs Optimized Pitch

(c) True Yaw vs Optimized Yaw

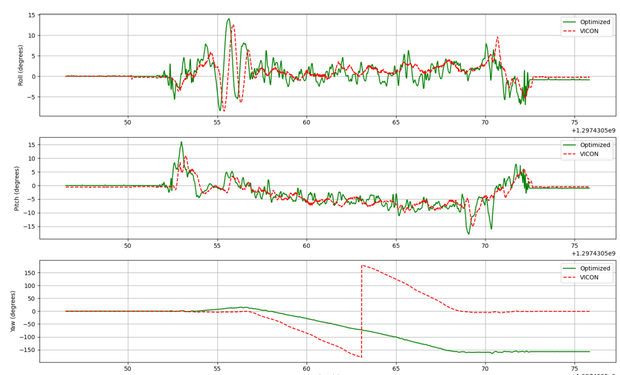
Dataset 9



(a) True Roll vs Estimated Roll

(b) True Pitch vs Estimated Pitch

(c) True Yaw vs Estimated Yaw

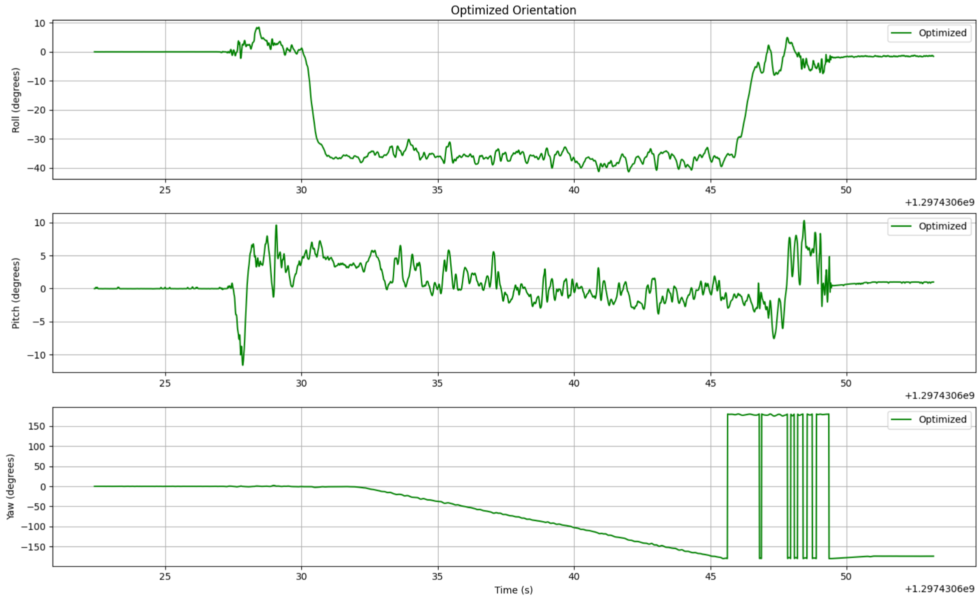


(a) True Roll vs Optimized Roll

(b) True Pitch vs Optimized Pitch

(c) True Yaw vs Optimized Yaw

Test Dataset 10

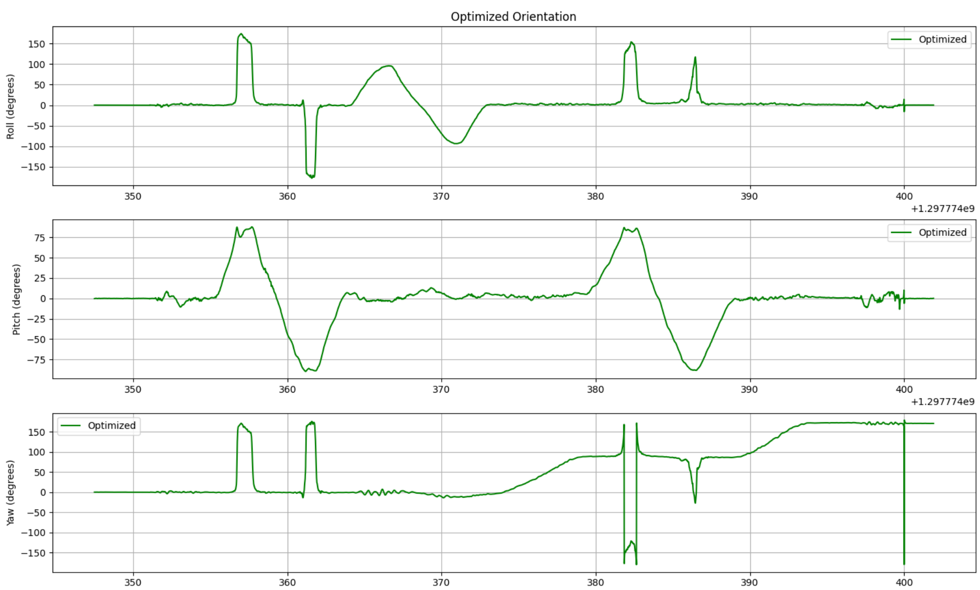


(a) True Roll vs Optimized Roll

(b) True Pitch vs Optimized Pitch

(c) True Yaw vs Optimized Yaw

Test Dataset 11



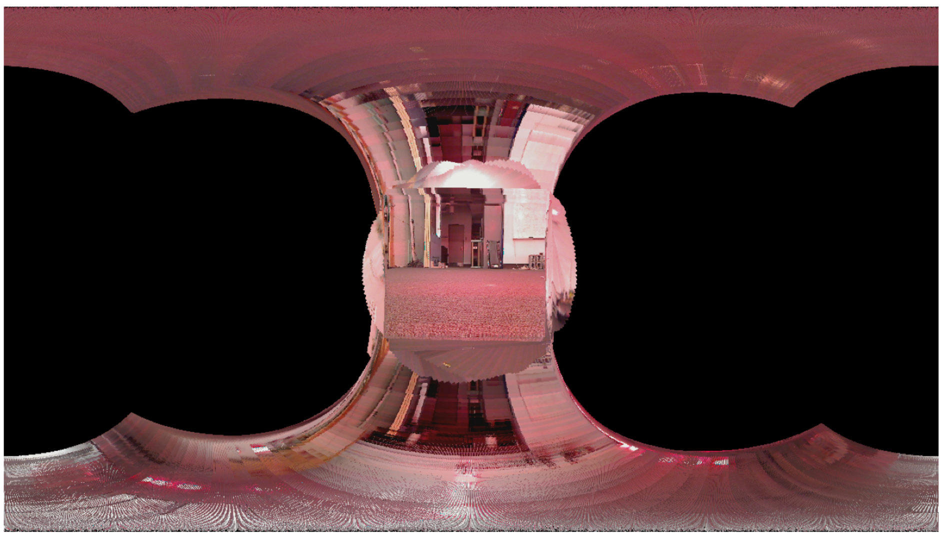
(a) True Roll vs Optimized Roll

(b) True Pitch vs Optimized Pitch

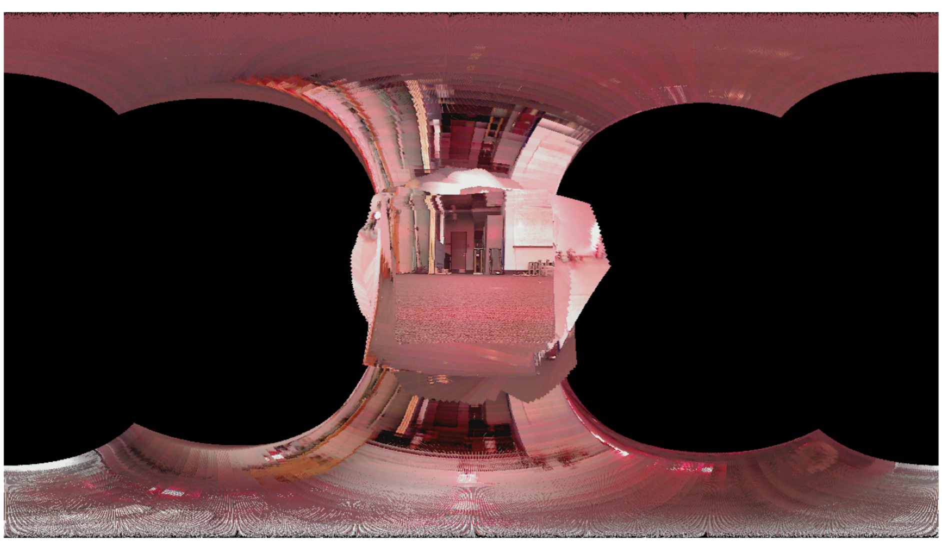
(c) True Yaw vs Optimized Yaw

B. Panorama Images

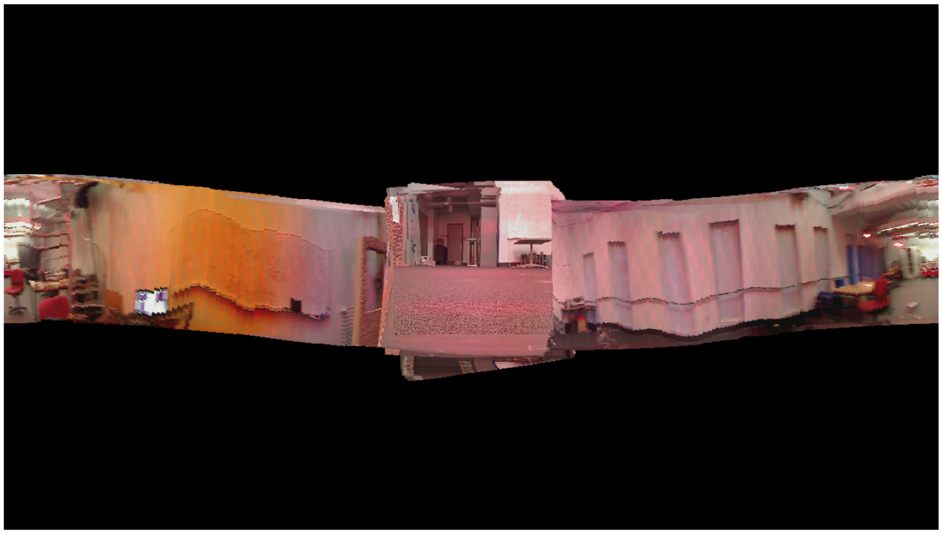
Dataset 1



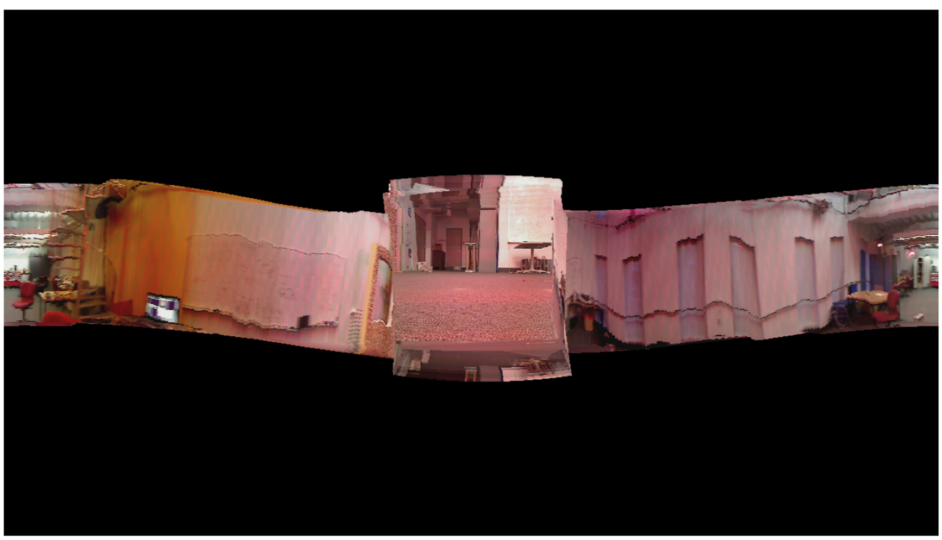
Dataset 2



Dataset 8

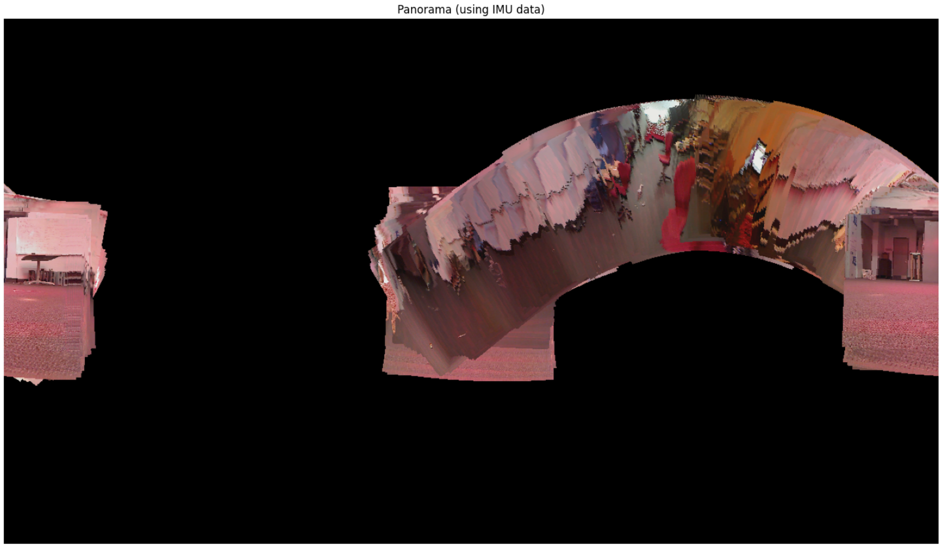


Dataset 9



Test Dataset 10

Panorama (using IMU data)



Test Dataset 11

Panorama (using IMU data)

C. Discussion

1) Orientation Tracking:

Convergence Analysis: The cost function plot for Dataset 1 exhibits rapid initial convergence, with the cost dropping sharply from approximately 1750 to below 500 within the first 20 iterations. This behavior is characteristic of gradient descent with an appropriate learning rate. The function continues to decrease monotonically and asymptotically approaches a minimum around iteration 80, suggesting our optimization successfully finds a stable solution. The smooth convergence curve indicates that our quaternion projection method effectively maintains the unit norm constraint without introducing instabilities.

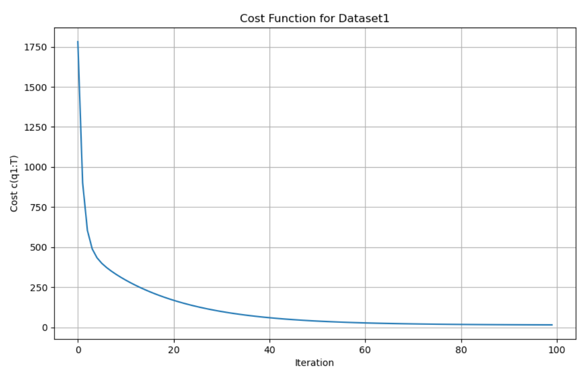


Fig. 3: Cost Function for Dataset 1

Tracking Performance: The orientation tracking results demonstrate the effectiveness of the proposed optimization approach in refining IMU-based estimates. In the training datasets, the optimized orientation trajectories closely align with the ground truth obtained from the VICON motion capture system. Initially, the estimated trajectories exhibit noticeable deviations, particularly in scenarios involving rapid rotation. However, after applying the projected gradient descent optimization, these discrepancies are significantly reduced, leading to smoother and more accurate estimates. This suggests that the optimization effectively mitigates sensor noise and drift, enhancing the reliability of the IMU-based orientation tracking.

When evaluated on test datasets, the method continues to show strong performance, although certain challenges arise. While the optimized roll and pitch angles remain largely consistent with ground truth, the yaw angle exhibits slightly larger deviations, especially in long-duration sequences. This can be attributed to the inherent drift in gyroscope measurements, which accumulates over time and introduces orientation errors that are more pronounced in yaw than in roll or pitch. Such drift effects become particularly evident in test sequences involving prolonged or fast rotational motion(dataset 6-9), where minor inaccuracies in IMU data compound over time.

The optimization process itself exhibits desirable convergence properties, as seen in the cost function trajectory. The cost rapidly decreases within the initial iterations and gradually stabilizes around a local minimum, indicating that the projected gradient descent algorithm successfully finds a stable and meaningful solution. Moreover, the smooth convergence curve suggests that enforcing quaternion normalization throughout the optimization process prevents numerical instability while maintaining computational efficiency.

Despite these promising results, the system's accuracy is inherently limited by the quality of IMU sensor data. Noise and bias in accelerometer and gyroscope readings continue to pose challenges, especially in scenarios with dynamic motion. While the current approach effectively reduces these effects, additional refinement—such as incorporating complementary filtering or nonlinear optimization techniques—could further enhance tracking accuracy.

2) Panorama::

Panorama Image Stitching:

The orientation estimates obtained through optimization are further validated through their application in panorama image stitching. In the training datasets, the refined orientation trajectories facilitate precise image alignment, enabling the seamless construction of panoramic images. The images exhibit well-aligned overlapping regions, confirming that the optimized orientation estimates are sufficiently accurate for practical applications.

When applied to test datasets, the panorama stitching process remains largely effective, though minor misalignments become noticeable in certain regions. These discrepancies appear to stem from residual orientation errors, particularly in datasets where yaw estimation deviates slightly from ground truth. Since image alignment is highly sensitive to rotational errors, even small inaccuracies in estimated orientation can lead to visible distortions or discontinuities in stitched images. This effect is more pronounced in sequences involving rapid motion, where errors accumulate more rapidly.

In addition to orientation accuracy, other factors such as lens distortion and lighting variations can influence the quality of the panorama. While the method assumes ideal image projections based solely on rotation estimates, real-world distortions may require additional compensation to ensure robust stitching across different environments. Incorporating distortion correction techniques or refining the projection model could improve the final image quality.