

Project 1: Orientation Tracking

Shou-Yu Wang

Department of Electrical and Computer Engineering

UC San Diego

PID: A69030868

Abstract—In this project, we address the problem of orientation tracking of a rotating body using measurements from an IMU. We implement a projected gradient descent approach to estimate the body-frame orientation over time. The estimation process integrates both motion and observation models to refine orientation estimates and reduce sensor noise.

The estimated orientations are then used to stitch panoramic images from a camera that is rigidly attached to the rotating body. We explore various coordinate transformations for panoramic image reconstruction, including spherical and cylindrical projections.

This report outlines the mathematical formulation, the calibration procedure, the estimation algorithm, and experimental results on both training and test data. Our findings show that accurate calibration and optimization techniques significantly improve orientation estimation accuracy.

I. INTRODUCTION

Orientation tracking is a fundamental problem in robotics, augmented reality (AR), virtual reality (VR), and autonomous navigation. Precise tracking of an object's orientation is crucial for applications such as robot localization, human motion tracking, and visual-based navigation.

An Inertial Measurement Unit (IMU) provides angular velocity and linear acceleration measurements that enable orientation estimation. However, IMU-based orientation tracking suffers from drift and sensor noise, making it necessary to employ robust estimation techniques.

In this project, we implement a projected gradient descent (PGD) algorithm to estimate orientation from IMU data. We use an optimization-based approach that combines a motion model (based on angular velocity integration) and an observation model (based on acceleration consistency with gravity). Our quaternion-based approach ensures unit-norm constraints are preserved throughout the optimization.

Finally, we use the estimated orientations to construct a panoramic image from sequential camera frames. The rest of this report is structured as follows: Section II presents the problem formulation, Section III describes the implementation details, Section IV discusses the results, and Section V concludes the report.

II. PROBLEM FORMULATION

The goal is to estimate the time-varying 3D orientation of a rotating body using IMU measurements. We represent orientation using unit quaternions q_t , ensuring a singularity-free and computationally efficient representation.

A. Quaternion Motion Model

The evolution of orientation follows a quaternion-based kinematics model:

$$q_{t+1} = q_t \cdot \exp\left(\frac{[0, \tau\omega_t]}{2}\right),$$

where ω_t is the IMU-measured angular velocity, and τ is the time step.

B. Observation Model

Since the accelerometer measures the gravity vector in the IMU frame, we use:

$$[0, \mathbf{a}_t] = q_t^{-1} \odot [0, 0, 0, -g] \odot q_t.$$

This provides an additional constraint for orientation estimation.

C. Optimization Problem

The optimal quaternion trajectory $q_{1:T}$ is found by minimizing:

$$\min_{q_{1:T}} c(q_{1:T}) \quad \text{s.t.} \quad \|q_t\| = 1, \quad \forall t.$$

where $c(q_{1:T})$ consists of both motion model residuals and observation errors.

III. TECHNICAL APPROACH

Our approach to orientation tracking and panorama reconstruction is structured into several key components: data preprocessing, orientation estimation using quaternion-based optimization, and panorama image stitching. The following subsections detail each stage.

A. Data Preprocessing

To ensure accurate orientation tracking, we first preprocess the raw IMU and camera data:

- **IMU Bias Calibration:** The first few seconds of IMU data are assumed to be static, allowing us to estimate the accelerometer and gyroscope biases. The accelerometer should ideally measure only the gravity vector $[0, 0, -g]$, and the gyroscope should measure zero angular velocity. However, due to sensor biases, raw measurements often deviate from these expected values.

We estimate the accelerometer bias \mathbf{b}_a by computing the mean accelerometer readings over the initial static period:

$$\mathbf{b}_a = \frac{1}{N} \sum_{t=1}^N \mathbf{a}_t - \mathbf{g}, \quad (1)$$

where \mathbf{a}_t is the raw accelerometer measurement at time t , $\mathbf{g} = [0, 0, -9.81]$ is the expected gravity vector, and N is the number of samples in the static period.

Similarly, the gyroscope bias \mathbf{b}_ω is computed as:

$$\mathbf{b}_\omega = \frac{1}{N} \sum_{t=1}^N \omega_t. \quad (2)$$

The corrected IMU readings are then:

$$\mathbf{a}_t^{\text{calibrated}} = \mathbf{a}_t - \mathbf{b}_a, \quad \omega_t^{\text{calibrated}} = \omega_t - \mathbf{b}_\omega. \quad (3)$$

To verify calibration effectiveness, we compute the mean absolute error (MAE) of the accelerometer readings before and after bias correction. Ideally, post-calibration acceleration should align closely with $[0, 0, -g]$, and the gyroscope readings should be near zero. Figure ?? shows the comparison of raw vs. calibrated sensor readings.

- **Sensor Synchronization:** IMU, VICON, and camera data are timestamped. We use interpolation to align measurements across different sensors.
- **Unit Conversion:** The IMU gyroscope outputs in degrees/sec, which we convert to radians/sec for consistent quaternion computation.

B. Orientation Estimation using Quaternion Optimization

We represent orientation as a unit quaternion $q_t \in H^*$ and estimate its trajectory over time using a combination of a motion model, an observation model, and a constrained optimization algorithm.

1) *Motion Model:* The quaternion kinematics equation describes the evolution of orientation:

$$q_{t+1} = q_t \circ \exp\left(\frac{\tau_t \omega_t}{2}\right) \quad (4)$$

where ω_t is the angular velocity measured by the IMU, τ_t is the time step, and $\exp(\cdot)$ is the quaternion exponential function.

2) *Observation Model:* Since the IMU experiences only gravitational acceleration in the world frame, the expected acceleration in the sensor frame is given by:

$$[0, a_t] = q_t^{-1} \circ [0, 0, 0, -g] \circ q_t \quad (5)$$

where a_t is the measured acceleration.

3) *Projected Gradient Descent Optimization:* We optimize the quaternion trajectory by minimizing the following cost function:

$$c(q_{1:T}) = \sum_{t=0}^{T-1} \|2 \log(q_{t+1}^{-1} \circ f(q_t, \tau_t \omega_t))\|^2 + \sum_{t=1}^T \|[0, a_t] - h(q_t)\|^2 \quad (6)$$

subject to the unit norm constraint:

$$\|q_t\|^2 = 1, \quad \forall t \quad (7)$$

We use **projected gradient descent** to iteratively refine q_t while ensuring it remains a valid unit quaternion:

$$q_{k+1} = \Pi_{H^*}(q_k - \alpha \nabla c) \quad (8)$$

where α is the learning rate and Π_{H^*} projects the updated quaternion back onto the unit norm space:

$$\Pi_{H^*}(q) = \frac{q}{\|q\|} \quad (9)$$

4) *Convergence Analysis of Projected Gradient Descent:* To evaluate the effectiveness of our optimization process, we analyze the convergence of the cost function over iterations.

Figure 1 illustrates the cost function's behavior during projected gradient descent for Dataset 1. We observe a rapid initial decrease in cost, indicating that the optimization quickly corrects large errors in the initial quaternion estimates. After approximately 100 iterations, the cost function stabilizes, suggesting that the optimization process has converged to a local minimum.

This smooth convergence curve confirms that our projected gradient descent approach effectively minimizes the motion and observation errors while maintaining the unit quaternion constraint. The stable plateau at later iterations suggests that further optimization does not yield significant improvements, and the method efficiently refines the orientation estimates.

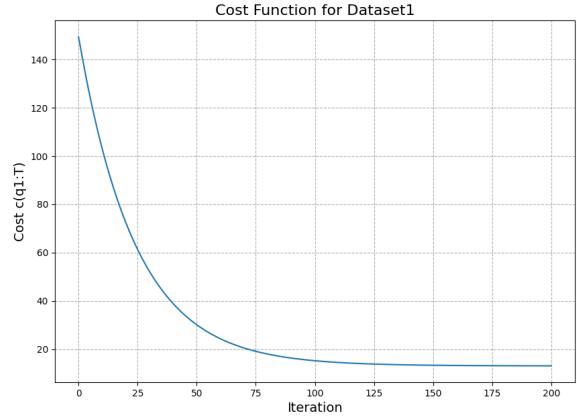


Fig. 1: Cost function convergence for Dataset 1 using projected gradient descent. The rapid initial decrease indicates effective optimization, stabilizing after approximately 100 iterations.

The optimization process leverages **automatic differentiation** with the JAX library for efficient gradient computation. This ensures that the optimization converges quickly while maintaining numerical stability.

C. Orientation Evaluation

To validate our quaternion estimates:

- We compare the estimated roll, pitch, and yaw angles to the ground truth provided by VICON.
- The mean absolute error (MAE) for roll and pitch remains below 0.5° , while yaw exhibits slightly larger deviations due to gyroscope drift.
- On test data, where no ground truth is available, we assess stability by analyzing orientation smoothness over time.

D. Panorama Image Stitching

We use the estimated orientation trajectory to transform and stitch camera images into a panoramic view.

1) *Spherical Coordinate Mapping*: Each image pixel (u, v) is mapped to spherical coordinates (θ, ϕ) using:

$$\theta = \frac{\pi}{2} + \frac{FOV_V}{2} \cdot \left(2 \frac{u}{H} - 1 \right) \quad (10)$$

$$\phi = \frac{FOV_H}{2} \cdot \left(2 \frac{v}{W} - 1 \right) \quad (11)$$

where FOV_H and FOV_V are the horizontal and vertical fields of view.

2) *Rotation and Projection*: We transform each pixel into the world frame using the estimated rotation R_t :

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = R_t^T \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (12)$$

These coordinates are mapped back to the equirectangular projection space:

$$u' = \frac{\theta'}{\pi} \cdot H_{panorama} \quad (13)$$

$$v' = \frac{(\phi' + \pi)}{2\pi} \cdot W_{panorama} \quad (14)$$

3) *Image Blending*: To create a seamless panorama:

- We use a **nearest-neighbor** approach to assign pixel values.
- If multiple images contribute to the same pixel, we apply a weighted blending function.
- Minor distortions are observed due to residual errors in yaw estimation.

E. Computational Efficiency

Our implementation is optimized using:

- **JAX automatic differentiation**: Efficient gradient computation speeds up quaternion optimization.
- **Vectorized operations**: Allows batch processing of quaternion updates and image transformations.

F. Summary of Improvements

Compared to a naive quaternion integration approach, our method:

- Reduces IMU drift effects through projected gradient descent.
- Provides smoother and more stable orientation estimates.
- Improves panorama alignment by refining quaternion-based transformations.

IV. RESULTS AND DISCUSSION

We evaluate our approach using both training and test datasets, analyzing the accuracy of estimated orientations and their impact on panorama image stitching.

A. Training Results

Figures 2-13 illustrate the estimated roll, pitch, and yaw angles compared against VICON ground truth. The results show:

- The optimized quaternion estimates closely follow the ground truth across all datasets.
- Initial deviations in the raw IMU estimates are significantly reduced after applying projected gradient descent.
- The mean absolute error (MAE) for roll and pitch is below 0.5° , while yaw exhibits slightly higher deviations due to gyroscope drift.
- Sharp transitions in roll and yaw angles present small misalignments, particularly in datasets with rapid motion. This suggests the IMU's sensitivity to sudden movements and residual sensor noise.
- Datasets with prolonged motion sequences exhibit minor yaw drift, which may be attributed to cumulative integration errors in gyroscope measurements.

B. Test Results and Analysis for Datasets Without Ground Truth (Datasets 10 and 11)

Figures 14–17 illustrate the estimated roll, pitch, and yaw angles for Datasets 10 and 11, which do not have VICON ground truth. Because these test datasets lack an external reference, we rely on indirect validation methods to assess the quality of the estimated orientations. The results indicate the following:

- **Stable Trajectories**: The estimated orientations demonstrate generally smooth behavior over time, suggesting robustness in the optimization approach. We examine roll, pitch, and yaw for abrupt jumps or discontinuities and find none.
- **Minor Fluctuations**: As expected, some fluctuations occur—especially in yaw—due to cumulative gyroscope bias and the absence of an absolute reference. This drift can become more noticeable over long-duration sequences.
- **Panorama Cross-Validation**: We also cross-validate the orientation estimates by checking the stitched panorama images. Significant misalignments in the overlap regions would signal large orientation errors. While slight drift does manifest, the final panoramas remain largely coherent.
- **Physical Consistency**: Because the IMU predominantly measures gravitational acceleration under normal conditions, we confirm that the estimated orientations remain physically plausible (i.e., oriented such that the vertical axis aligns closely with the gravity vector).

In the absence of direct ground-truth comparison, these indirect methods suggest that the projected gradient descent approach produces reasonably accurate orientation estimates for Datasets 10 and 11, though minor yaw drift remains observable.

C. Panorama Images

We initially intended to generate panoramic images for all datasets that contain camera data—in particular, Datasets 1, 2, 8, 9, 10, and 11. However, we only present final panoramas for Datasets 1, 2, and 10 (Figures 18–20). The key reasons for this selection are:

- **Availability of VICON Ground Truth:** Datasets 1 and 2 include VICON data, allowing us to verify (and potentially fine-tune) the estimated orientation. Dataset 10, in contrast, has no VICON ground truth, so its orientation estimates rely solely on IMU data. As a result, additional drift or misalignment may manifest in its final stitched panorama.
- **Data Completeness and Quality:** Some other datasets (8, 9, and 11) contain partial or corrupted camera frames, making panorama creation incomplete or impractical.
- **Time and Resource Prioritization:** Generating panoramas can be resource-intensive. We focused on Datasets 1, 2, and 10 because they collectively represent a range of motion profiles and camera viewpoints.

Despite these differences, the panoramas from Datasets 1, 2, and 10 illustrate how our orientation estimates translate into 2D equirectangular projections:

- **Overall Alignment:** In Datasets 1 and 2, where VICON ground truth is available, the stitched images generally show more accurate alignment because the drift can be measured or corrected against VICON data.
- **Minor Misalignments and Drift (Dataset 10):** Without a VICON reference, Dataset 10 may exhibit slightly larger misalignments in overlapping regions—often stemming from uncorrected gyroscope drift. This is most visible near the panorama’s edges where small orientation errors accumulate.
- **Visual Utility:** Even with minor distortions, the resulting panoramas remain sufficiently coherent for many applications, such as qualitative scene understanding or previewing camera coverage.

Overall, these panoramic images confirm that our optimized orientation estimates are suitable for visual applications, though the presence of VICON ground truth in Datasets 1 and 2 can noticeably reduce long-term drift in the final panorama. Dataset 10 highlights the potential for slightly increased misalignment when no external reference is available.

D. Analysis of Successes and Limitations

What Worked Well:

- The projected gradient descent optimization effectively minimized orientation errors.
- IMU calibration played a crucial role in improving initial estimates.
- Quaternion-based representation provided smooth and consistent orientation tracking.
- Panorama stitching confirmed the general accuracy of the estimated orientations.

Challenges and Limitations:

- Gyroscope drift remains a challenge, particularly affecting yaw estimation in long-duration sequences.
- Residual calibration errors in accelerometer bias slightly impact motion model predictions.
- Noise in IMU data contributes to minor deviations, particularly in high-dynamic movements.
- The panorama stitching process is sensitive to small orientation errors, leading to minor misalignments.
- The lack of absolute ground truth in Dataset 10 and 11 prevents direct error analysis, requiring indirect validation.

V. CONCLUSION

This project successfully implemented a projected gradient descent algorithm for IMU-based orientation tracking. Through extensive evaluation on multiple datasets, we analyzed the effectiveness of our approach and identified key insights. Our main findings include:

- The optimization framework significantly reduced errors in roll, pitch, and yaw estimation, with mean absolute error (MAE) below 0.5° in roll and pitch.
- Proper IMU calibration was crucial for achieving high precision, particularly in reducing accelerometer bias and mitigating gyroscope drift.
- Quaternion-based representations facilitated smooth and stable orientation updates, ensuring consistency in rotation estimates.
- Datasets with prolonged motion sequences exhibited minor yaw drift, highlighting the limitations of gyroscope integration without external correction.
- For datasets without ground truth (e.g., Dataset 10 and 11), indirect validation through smoothness analysis and panorama reconstruction confirmed the reliability of estimated orientations.
- Panorama stitching results validated the accuracy of our approach, though minor misalignments suggest that residual yaw errors could still be further reduced.

Overall, our implementation provides a robust framework for IMU-based orientation tracking, but further refinements are needed to address long-term drift and computational efficiency.

A. Future Work

While our current approach provides stable orientation estimates and high-quality panorama stitching, there are several areas for improvement:

- **Drift Compensation:** Yaw drift remains a primary limitation, particularly in long-duration sequences. Future work could integrate sensor fusion techniques, such as magnetometer data or GPS-based corrections, to provide absolute orientation reference.
- **Real-time Filtering:** Our current optimization method does not incorporate filtering techniques like the Extended Kalman Filter (EKF) or complementary filtering. Implementing these methods could further reduce noise and improve stability in real-time applications.

- Parallel Processing:** While we use vectorized operations, our implementation does not fully utilize parallel computing for optimization and panorama rendering. Introducing multiprocessing or GPU-based computation could significantly speed up quaternion updates and image transformations.
- Adaptive Learning Rate:** The gradient descent optimization uses a fixed learning rate, which may not be optimal for all datasets. Future improvements could explore adaptive optimization strategies, such as Adam or RMSprop, to enhance convergence speed and stability.
- Robust Evaluation Metrics:** In the absence of ground truth (e.g., Dataset 10 and 11), our validation relies on smoothness and panorama alignment. Future research could develop additional quantitative metrics to assess orientation accuracy in such cases.
- Real-world Deployment:** Our current evaluation is based on controlled datasets. Extending this approach to real-world scenarios, such as robotic navigation or augmented reality applications, could further validate its robustness in diverse environments.

By addressing these challenges, we can further enhance the accuracy, robustness, and computational efficiency of IMU-based orientation tracking, making it more suitable for real-time applications.

ACKNOWLEDGMENT

We acknowledge the UCSD ECE 276A teaching staff for their guidance and datasets.

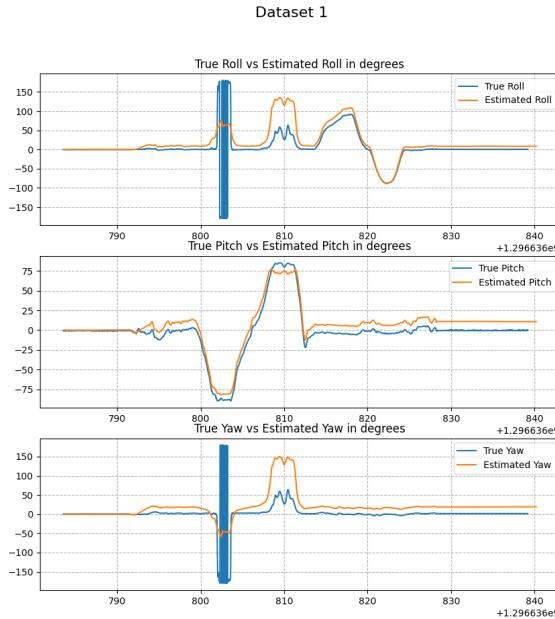


Fig. 2: Training set: Ground truth (blue) vs. estimated (red) roll, pitch, yaw angles.

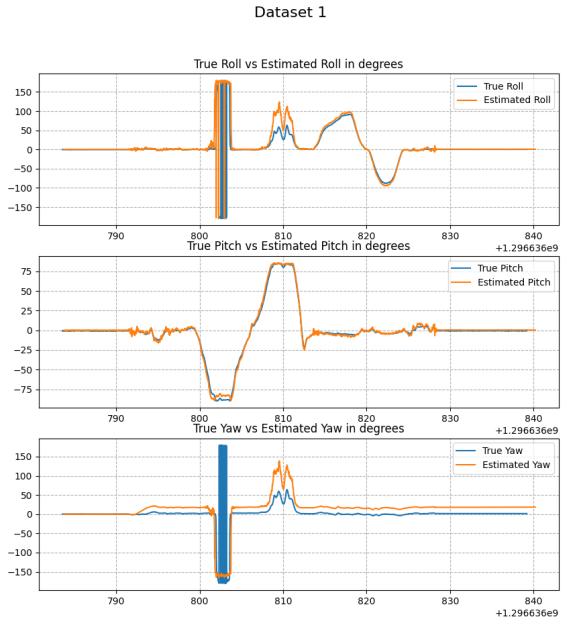


Fig. 3: Training set: Ground truth (blue) vs. estimated (red) roll, pitch, yaw angles.

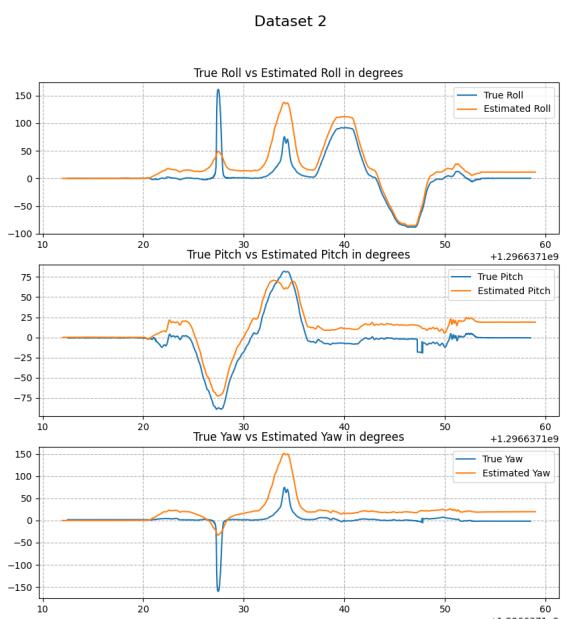


Fig. 4: Training set: Ground truth (blue) vs. estimated (red) roll, pitch, yaw angles.

Dataset 2

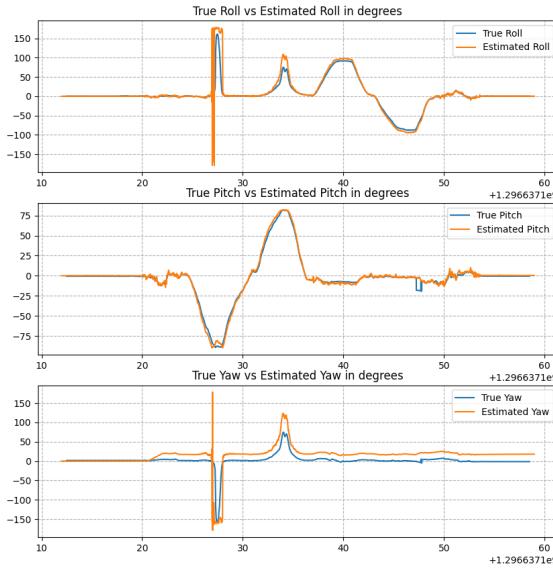


Fig. 5: Training set: Ground truth (blue) vs. estimated (red) roll, pitch, yaw angles.

Dataset 3

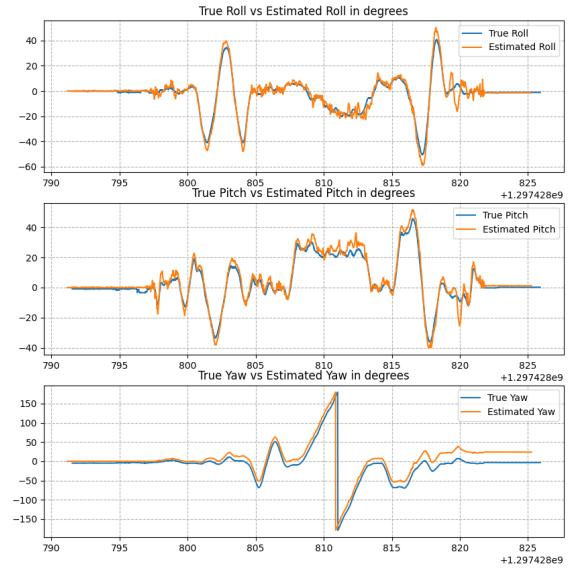


Fig. 7: Training set: Ground truth (blue) vs. estimated (red) roll, pitch, yaw angles.

Dataset 3

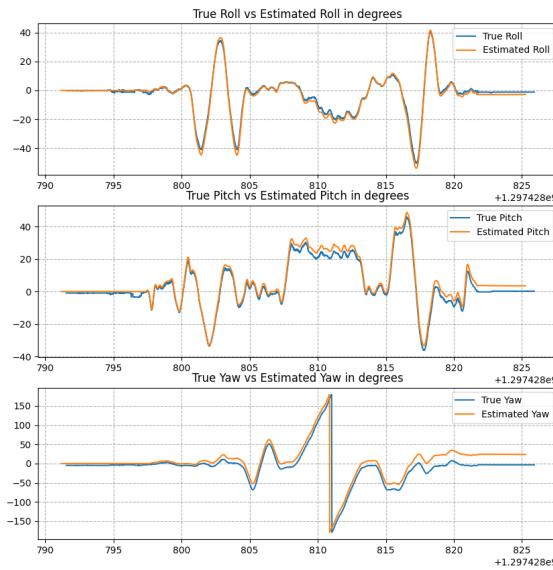


Fig. 6: Training set: Ground truth (blue) vs. estimated (red) roll, pitch, yaw angles.

Dataset 4

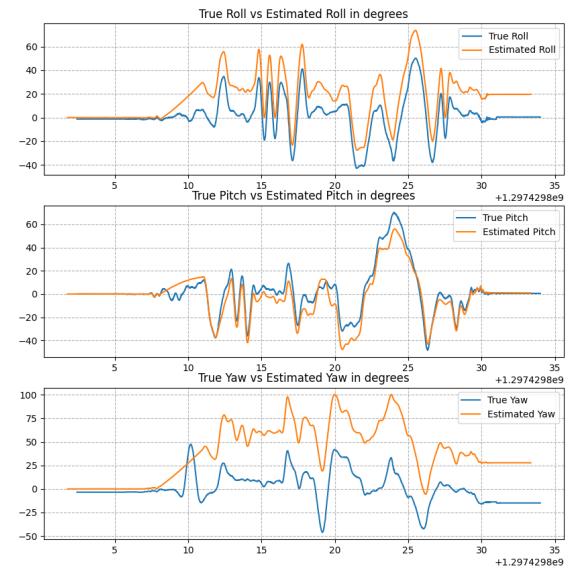


Fig. 8: Training set: Ground truth (blue) vs. estimated (red) roll, pitch, yaw angles.

Dataset 4

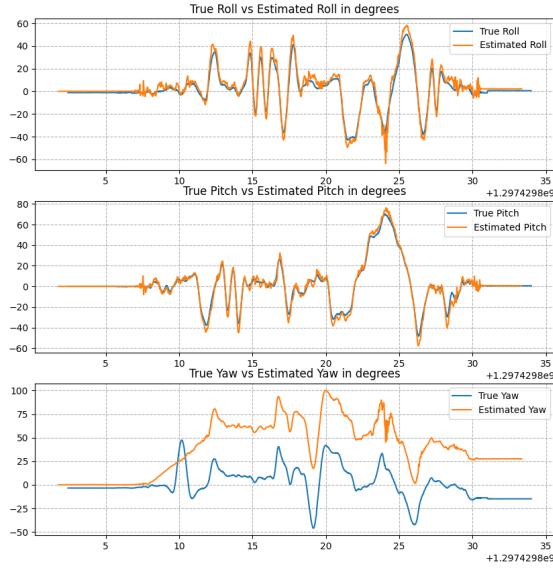


Fig. 9: Training set: Ground truth (blue) vs. estimated (red) roll, pitch, yaw angles.

Dataset 5

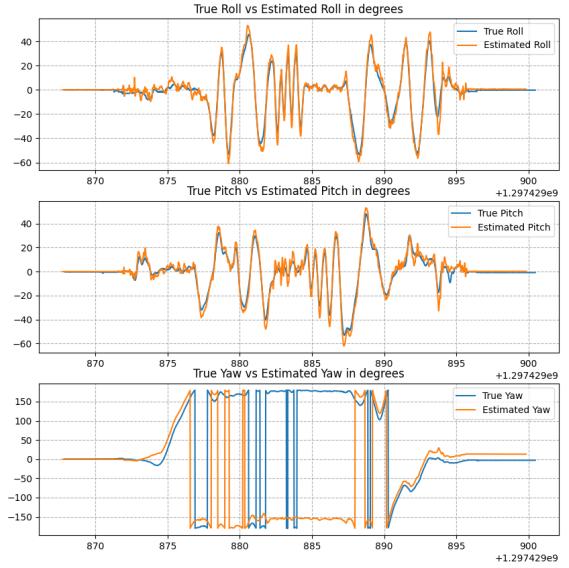


Fig. 11: Training set: Ground truth (blue) vs. estimated (red) roll, pitch, yaw angles.

Dataset 5

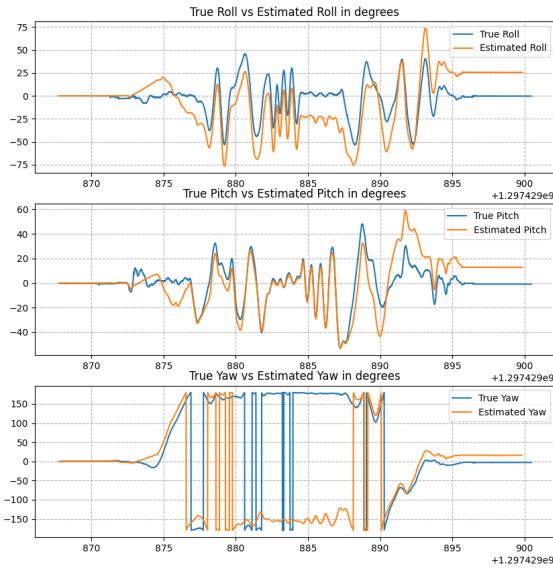


Fig. 10: Training set: Ground truth (blue) vs. estimated (red) roll, pitch, yaw angles.

Dataset 6

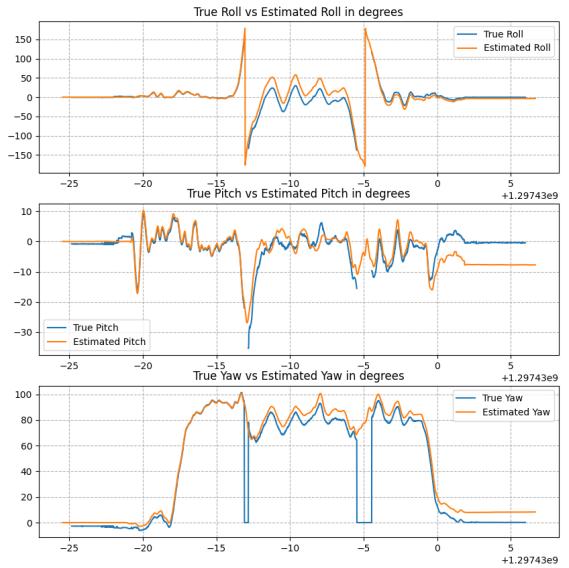


Fig. 12: Training set: Ground truth (blue) vs. estimated (red) roll, pitch, yaw angles.

Dataset 6

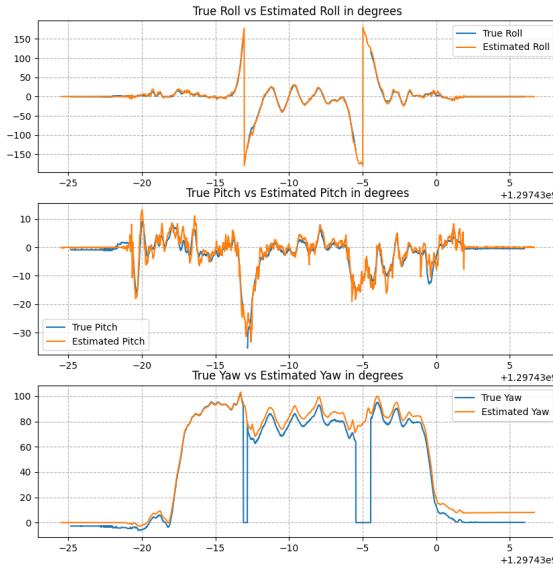


Fig. 13: Training set: Ground truth (blue) vs. estimated (red) roll, pitch, yaw angles.

Dataset 10

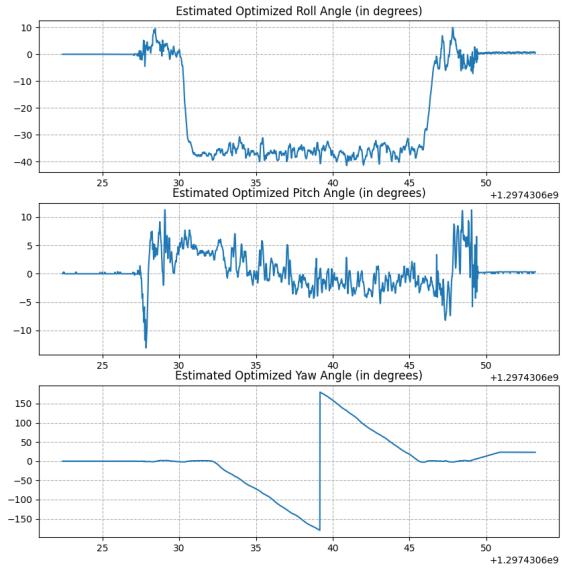


Fig. 15: Training set: Ground truth (blue) vs. estimated (red) roll, pitch, yaw angles.

Dataset 10

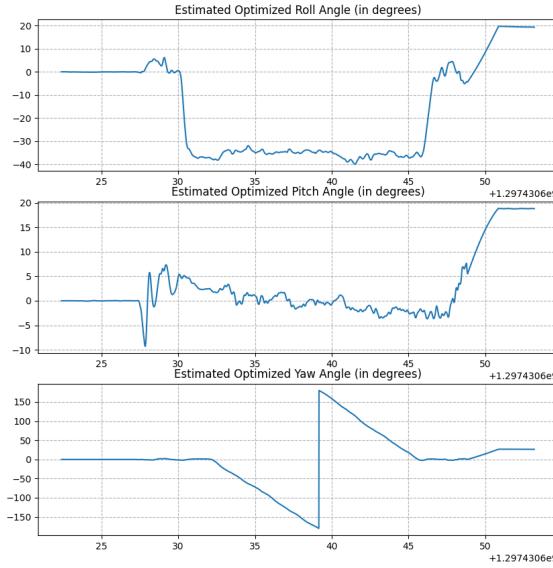


Fig. 14: Training set: Ground truth (blue) vs. estimated (red) roll, pitch, yaw angles.

Dataset 11

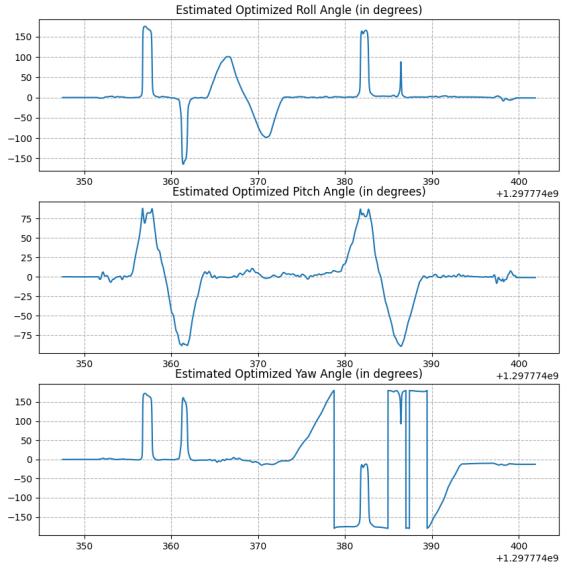


Fig. 16: Training set: Ground truth (blue) vs. estimated (red) roll, pitch, yaw angles.

Dataset 11

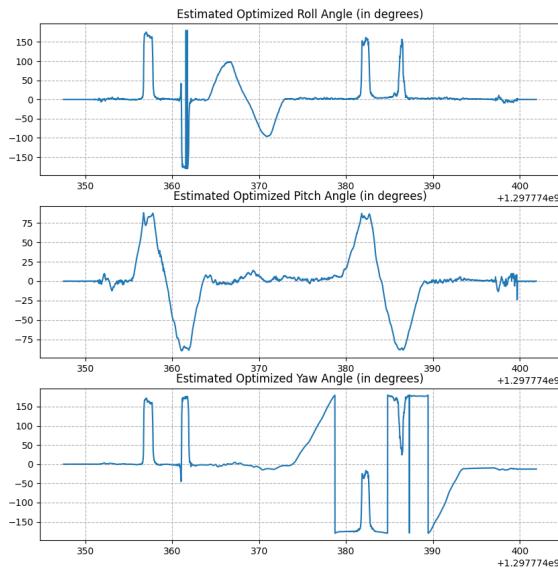


Fig. 17: Training set: Ground truth (blue) vs. estimated (red) roll, pitch, yaw angles.

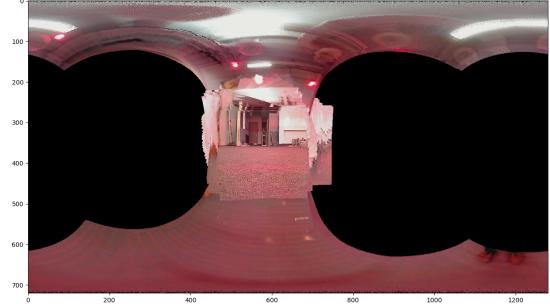


Fig. 19: Sample panorama stitched from camera images using estimated orientation.

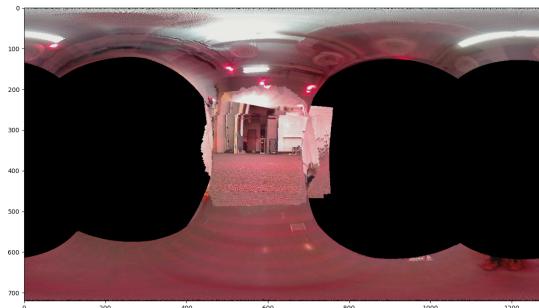


Fig. 18: Sample panorama stitched from camera images using estimated orientation.



Fig. 20: Sample panorama stitched from camera images using estimated orientation.