

Project 1: Orientation Tracking

Mazeyu Ji, PID:A59023027

Abstract—This report details a project aimed at tracking the three-dimensional orientation of a rotating body using Inertial Measurement Unit (IMU) data and constructing panoramic images from camera images captured by the rotating body. By implementing a projected gradient descent algorithm, this project successfully estimates the orientation of the body over time and uses these orientation estimates to stitch together panoramic images. In the orientation tracking phase, the project first calibrates the IMU to accurately estimate the biases and scale factors of accelerometers and gyroscopes. Then, it achieves precise orientation estimation by optimizing a cost function designed to minimize the prediction errors of the motion model and the observation model. In the panorama construction phase, the project demonstrates how to stitch images from a time series into a panorama based on estimated or ground truth orientations. The results section of the report presents the orientation estimation results on the training dataset, including comparisons with VICON ground truth values, and performance on the test dataset. Additionally, panoramic images obtained from both the training and test datasets are showcased. The report not only elucidates the challenges encountered and solutions implemented during the project but also discusses the project outcomes, providing directions for future work.

Index Terms—IMU calibration, orientation tracking, gradient descent, panorama stitching, quaternion estimation

I. INTRODUCTION

THE problem addressed in this project involves tracking the three-dimensional orientation of a rotating body using data from an Inertial Measurement Unit (IMU) and constructing panoramic images from camera images captured by this body. This task is critical in various fields, including robotics, augmented reality, and autonomous navigation systems, where understanding an object's orientation in space and creating comprehensive visual representations of environments are foundational elements. The ability to accurately track orientation and stitch together panoramic images enables enhanced interaction with and navigation through physical spaces, making it a vital area of research and application.

Orientation tracking and panorama construction pose significant challenges due to the need for precise calibration and interpretation of sensor data. Inaccuracies in sensor measurements, such as biases and scale factors in IMU components, can lead to errors in orientation estimation, which, in turn, affect the quality of the constructed panoramas. Additionally, the process requires the integration of disparate data types—namely, IMU readings and camera images—into a coherent framework that accounts for temporal alignment and spatial configuration.

Our approach to addressing this problem begins with the calibration of the IMU to correct for any biases and inaccuracies in the sensor data. This involves estimating the biases and scale factors of the accelerometers and gyroscopes using ground truth data obtained from a VICON motion capture

system. With calibrated IMU data, we implement a projected gradient descent algorithm to estimate the orientation of the rotating body over time. This algorithm optimizes a cost function that minimizes the difference between predicted and observed orientations, ensuring that the estimated orientations are as accurate as possible.

For the construction of panoramic images, our method aligns camera images over time based on the estimated orientations. This process involves selecting the most appropriate orientation estimate for each camera image, accounting for temporal discrepancies and ensuring that the images are stitched together in a manner that reflects the true spatial arrangement of the environment.

II. PROBLEM FORMULATION

The objective of this project is to solve the problem of estimating the 3-D orientation of a rotating body over time using measurements from an Inertial Measurement Unit (IMU) and to utilize these orientation estimates for constructing a panoramic image by stitching together camera images obtained from the rotating body. The mathematical formulation of this problem is presented through a series of precise equations as follows:

Orientation Estimation: The orientation of the body at time t is represented by a unit quaternion $q_t \in H^*$, which encapsulates the rotation from a reference frame to the body frame at time t . The IMU provides measurements of angular velocity ω_t and linear acceleration a_t in the body frame, which are critical for estimating the orientation.

Quaternion Kinematics Motion Model: The motion model for predicting the quaternion at the next timestep $t+1$ is given by:

$$q_{t+1} = f(q_t, \tau_t \omega_t) := q_t \circ \exp([0, \tau_t \omega_t / 2]).$$

where τ_t is the time difference between consecutive measurements, \exp denotes the quaternion exponential function, and \circ represents quaternion multiplication.

Observation Model: Given that the body undergoes pure rotation, the observation model for the measured acceleration a_t in the IMU frame is:

$$a_t = h(q_t) := q_t^{-1} \circ [0, 0, 0, -g] \circ q_t.$$

indicating that the measured acceleration should align with the gravitational acceleration when correctly transformed by the estimated orientation q_t .

Optimization Problem: The optimization problem is formulated to estimate the sequence of orientations $\{q_1, q_2, \dots, q_T\}$ that minimizes the cost function:

$$c(q_1 : T) := \frac{1}{2} \sum_{t=0}^{T-1} \|2 \log(q_{t+1}^{-1} \circ f(q_t, \tau_t \omega_t))\|^2 +$$

$$\frac{1}{2} \sum_{t=1}^T \|a_t - h(q_t)\|_2^2,$$

which is subject to the constraint that each q_t remains a unit quaternion ($\|q_t\|_2 = 1$ for all t).

Panorama Construction: The construction of the panoramic image involves aligning and stitching camera images captured at different times, based on their corresponding orientation estimates. For images C_1, C_2, \dots, C_N and orientations q_1, q_2, \dots, q_N , each image C_i is transformed to the panoramic frame using its orientation q_i . The panoramic image P is then created by overlaying these transformed images:

$$P = \bigoplus_{i=1}^N T(q_i, C_i),$$

where \bigoplus indicates combining and blending the images. This process hinges on the accuracy of the orientation estimates, with the quality of P reflecting the precision of these estimates.

III. TECHNICAL APPROACH

A. Orientation Tracking

Our technical approach to orientation tracking involves a two-step process: calibration of the Inertial Measurement Unit (IMU) and implementation of the projected gradient descent algorithm for estimating orientation.

1) IMU Calibration: The technical approach for converting raw IMU sensor data to physical units involves precise mathematical formulas that ensure the accuracy of subsequent applications such as orientation tracking and panorama reconstruction:

Bias Correction: The initial step involves adjusting the raw sensor outputs to account for any static offset, known as the bias. This is necessary to ensure that the rest state of the sensor corresponds to zero.

Scale Factor Determination: A scale factor is calculated to transform the bias-corrected data into physical units. For a 10-bit A/D converter with a reference voltage (Vref), the scale factor is given by:

$$\text{scale_factor} = \frac{V_{\text{ref}}}{1023} / \text{sensitivity}$$

This factor is applied to the bias-corrected data to convert it into units of g-force for accelerometers or degrees per second for gyroscopes.

Physical Unit Conversion: For accelerometer data, after bias correction, the physical acceleration in g's is computed as:

$$\text{acceleration} = (\text{raw} - \text{bias}) \times \text{scale_factor}$$

For gyroscope data, the raw values, once bias-corrected and scaled, are further adjusted to account for the unit conversion from degrees to radians per second, which is necessary for most rotational motion calculations:

$$\text{angular_rate} = (\text{raw} - \text{bias}) \times \text{scale_factor} \times \frac{\pi}{180}$$

These conversions from raw data to physical units are essential for accurately interpreting the IMU data. The accelerometer's output provides the orientation relative to gravity, while the gyroscope measures the rate of rotation. When combined, they allow for a comprehensive understanding of the device's motion and orientation in space, enabling the accurate stitching of images in panorama reconstruction tasks.

2) Projected Gradient Descent: The Projected Gradient Descent (PGD) method is specifically tailored for optimization problems with constraints, such as ensuring a quaternion q , representing orientation, remains on the unit sphere ($\|q\| = 1$). The gradient g is the core component that guides the optimization process. Here is a consolidated view incorporating the provided content:

Gradient Calculation: The gradient of the cost function $\nabla p(q)$ is computed, which points in the direction of the steepest ascent of $p(q)$.

Projection onto the Tangent Plane: To maintain the unit norm constraint, the gradient $\nabla p(q)$ is projected onto the tangent plane of the unit sphere at q , resulting in the descent direction g_k [1]. This projection is mathematically expressed as:

$$g_k = \nabla p(q) - (\nabla p(q) \cdot q)q$$

Gradient Descent Update: The quaternion is updated along the direction of g_k , scaled by a step size α_k :

$$q_{\text{temp}} = q_k - \alpha_k g_k$$

Re-projection onto Sphere: The updated quaternion q_{temp} is then re-normalized to ensure it remains on the unit sphere:

$$q_{k+1} = \frac{q_{\text{temp}}}{\|q_{\text{temp}}\|}$$

This process iteratively moves the quaternion q towards the minimum of the cost function while strictly adhering to the constraint that q must represent a valid orientation by staying on the unit sphere. Through this methodology, PGD effectively finds the orientation that best fits the IMU data.

B. Panorama Reconstruction

Panorama Reconstruction is divided into two parts, spherical projection and cylindrical projection. We first project the 2D pixel coordinates to latitude and longitude coordinates, and then project them to the world coordinate system using an estimated rotation matrix. Finally, we project the world coordinate system onto a cylinder, and the 2D unfolding results in a panorama.

1) Spherical Projection: To compute the world coordinates of each pixel in an image for panorama reconstruction, the following technical approach is used:

Latitude and Longitude Calculation: In an image, each pixel's location is initially represented in pixel coordinates, which need to be mapped to latitude (ϕ) and longitude (λ) on a unit sphere. The pixel coordinates are $(x_{\text{pixel}}, y_{\text{pixel}})$, with the origin typically at the top left of the image. To convert

these coordinates to latitude and longitude, the pixel coordinates are first normalized based on the image dimensions, then scaled by the camera's field of view. The normalized coordinates, (x_{norm}, y_{norm}) , are calculated by translating the pixel coordinates to have the center of the image as the origin and then dividing by half the total number of pixels in width and height, respectively:

$$x_{norm} = \frac{2x_{pixel}}{\text{width}} - 1$$

$$y_{norm} = \frac{2y_{pixel}}{\text{height}} - 1$$

The field of view (FoV) for the camera, given as 60° horizontally and 45° vertically, is converted to radians:

$$FoV_h = 60 \times \frac{\pi}{180}$$

$$FoV_v = 45 \times \frac{\pi}{180}$$

Using the normalized coordinates and the field of view, the longitude and latitude for each pixel are calculated as follows:

$$\lambda = x_{norm} \times \frac{FoV_h}{2}$$

$$\phi = y_{norm} \times \frac{FoV_v}{2}$$

These formulas map the 2D pixel coordinates to their corresponding positions in spherical coordinates, which can be used to transform and project the image onto a spherical panorama.

Spherical to Cartesian Conversion: Each pixel's spherical coordinates (λ, ϕ) are then converted to Cartesian coordinates. The conversion assumes that each pixel represents a point on a unit sphere centered at the origin of the coordinate system. The formulas for conversion from spherical to Cartesian coordinates (assuming a radius of 1) are:

$$x = \cos(\phi) \cdot \cos(\lambda)$$

$$y = \cos(\phi) \cdot \sin(\lambda)$$

$$z = \sin(\phi)$$

Applying Camera-to-World Rotation: The Cartesian coordinates are rotated to the world frame using the camera-to-world orientation matrix R . This rotation aligns the image with the world's coordinate system. For a point p in Cartesian coordinates, the rotated point p' is given by:

$$p' = Rp$$

These steps transform the image pixels into world coordinates that accurately reflect their position as if they were points on the surface of a sphere centered at the camera's position in the world space. This forms the basis for the subsequent projection of these coordinates onto a cylinder for panorama unwrapping.

2) Cylindrical Projection: For cylindrical projection in the context of panorama creation, the world coordinates of each pixel are projected from a spherical to a cylindrical surface and then unwrapped into a flat image. This process entails several steps:

Calculate Cylindrical Coordinates: From the world coordinates in Cartesian form, cylindrical coordinates are computed. The angular position, or width on the cylinder, is determined by $\theta = \arctan 2(Y, X)$, ensuring it wraps around the cylinder. This angle is normalized to a $[0, 1]$ range by dividing by 2π . The height on the cylinder is calculated from the Z coordinate, scaled to a $[0, 1]$ range as height = $\frac{Z}{2} + 0.5$, where the division by 2 centers the height and the addition of 0.5 adjusts for the negative values of Z.

Project onto 2D Plane: The cylindrical coordinates are then mapped to a 2D plane, which will be the flat panoramic image. The width and height values are scaled to match the desired resolution of the output image. The width corresponds to the X-axis of the 2D image and is scaled by the image width, and similarly, the height is scaled by the image height.

Create Cylindrical Image: With the 2D plane coordinates, a new image array is populated. Each pixel's color in the new image is determined by the corresponding point on the 3D cylinder. If the original 3D data includes color information, this is transferred to the appropriate pixels in the 2D image.

This method of projecting to a cylinder and then unwrapping to a 2D plane is particularly well-suited to panoramic imaging, as it preserves the vertical dimension of the scene while allowing for horizontal wrapping, creating a continuous panoramic view when the image is viewed as a loop.

IV. RESULTS

A. IMU Calibration

Through calibration, we have converted the scale-less raw data into real data with units, as shown in Fig. 1. The gyroscope has been transformed to radians per second (rad/sec), and the scale of the accelerometer has been converted to gravitational acceleration (g).

Subsequently, using the fundamental Quaternion Kinematics Motion Model and observation model, we integrated the unit quaternions and $[0, 0, 0, -g]$ as the initial rotation and initial acceleration. This allowed us to calculate the roll, pitch, and yaw (rpy), as well as the three-axis acceleration, at every moment. By comparing these values with the ground truth obtained from the VICON system, we confirmed the accuracy of our calibration. As shown in Fig. 5, the calibration results are relatively accurate, reflecting true motion. Although the results for training set 1 are presented, our calibration was verified to be precise across all datasets, except for training set 4. The inaccuracy in training set 4 stems from a glitch encountered during data collection, where a reset pin activation caused all three gyroscope values to jump to a fixed value before returning to normal several seconds later. This locked the gyros at a nominal zero value, deviating from the true zero bias level. Fig. 3 illustrates this phenomenon, where a uniform drift is observed between 6-10 seconds, indicative of an abnormal gyro bias caused by the glitch, despite the absence

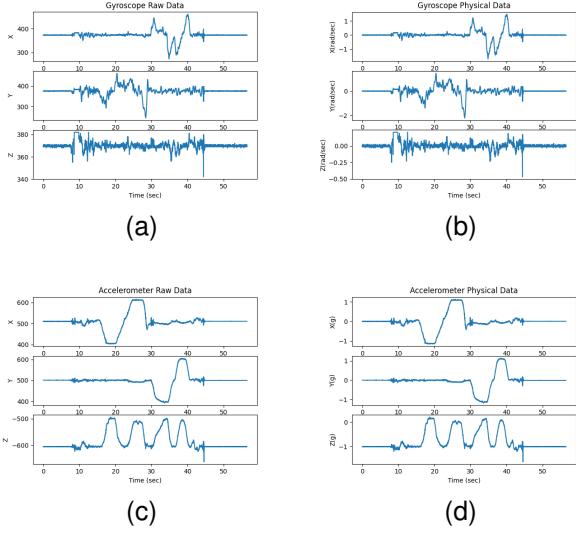


Fig. 1. The calibration outcome for the gyroscope and the accelerator is presented here. (a) represents the raw data and (b) depicts the physical data post-calibration of the gyroscope. (c) represents the raw data and (d) depicts the physical data post-calibration of the accelerator.

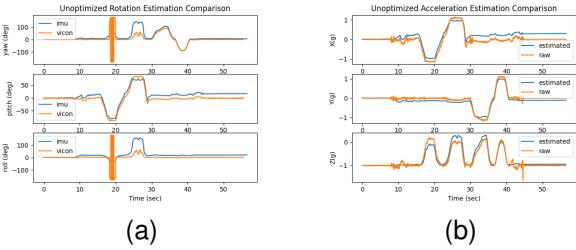


Fig. 2. We compared the estimated values and ground truth through a straightforward forward integration, with (a) showing the comparison of roll, pitch, and yaw (rpy), and (b) illustrating the comparison of acceleration. The fluctuations in rpy observed in figure (a) are reasonable, as they correspond to angle jumps around 2π .

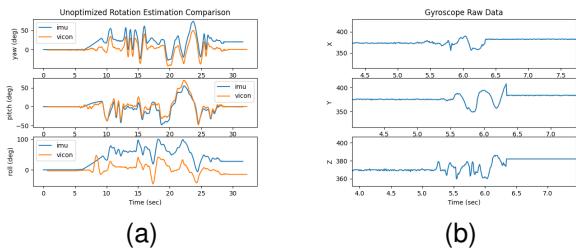


Fig. 3. In training set 4, a glitch occurred in the gyroscope, resulting in an incorrect bias, which caused an abnormal drift during the rotation process. The drift can be observed in figure (a), while figure (b) shows the bias that occurred during the corresponding time period.

of angular velocity during this period according to the ground truth.

B. Comparison of Estimated and Ground Truth Angles

Subsequently, we optimized the cost function $c(q_1 : T)$ using Projected Gradient Descent (PGD), which is defined as:

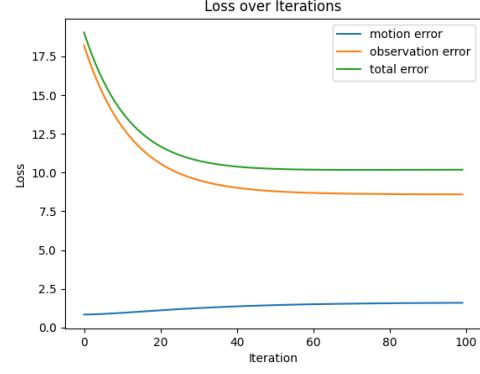


Fig. 4. Example picture of loss over iterations. The blue line refers to the motion loss, the orange line refers to the observation loss and the green line refers to the total loss.

$$c(q_1 : T) := \frac{1}{2} \sum_{t=0}^{T-1} \|2 \log(q_{t+1}^{-1} \circ f(q_t, \tau_t \omega_t))\|^2 + \frac{1}{2} \sum_{t=1}^T \|a_t - h(q_t)\|_2^2,$$

This optimization is divided into two parts: the motion model error and the observation model error. We used the orientation at each moment obtained from a simple integration using the motion model as initial values. From the Fig. 4, it can be observed that at the beginning of the optimization, the error from the motion model is zero, indicating perfect alignment with the initial motion-based estimates. However, there was a significant error from the observation model, which is derived from the accelerometer measurements.

As the optimization iterations progressed, the motion error increased slightly, suggesting a departure from the simple integration predictions. Simultaneously, the observation error decreased, indicating an improving alignment with the accelerometer data. Both errors gradually approached a state of equilibrium and began to converge, with the overall loss tending towards a steady value. The plot shows a smooth optimization process without any oscillations or instabilities, indicating a stable convergence to the solution.

Following this, we applied the optimization algorithm to each dataset in our training set. The optimized orientations were then converted into roll, pitch, and yaw (rpy) angles and compared with the ground truth (GT) provided by the VICON system. We observed that the optimized images were generally well-aligned with the GT. Some images, such as in the Fig. 5a and the Fig. 5e, exhibited intense fluctuations, which were attributed to the angles oscillating around 2π ; nevertheless, the overall performance of the optimization can be described as exceptional.

Despite the high accuracy in yaw and pitch, the optimization was unable to correct the drift in roll observed between 6-8 seconds in training set 4, due to a glitch in the gyroscope's IMU data. This underscores the significant challenge that sensor malfunctions pose in optimization problems. Sensor

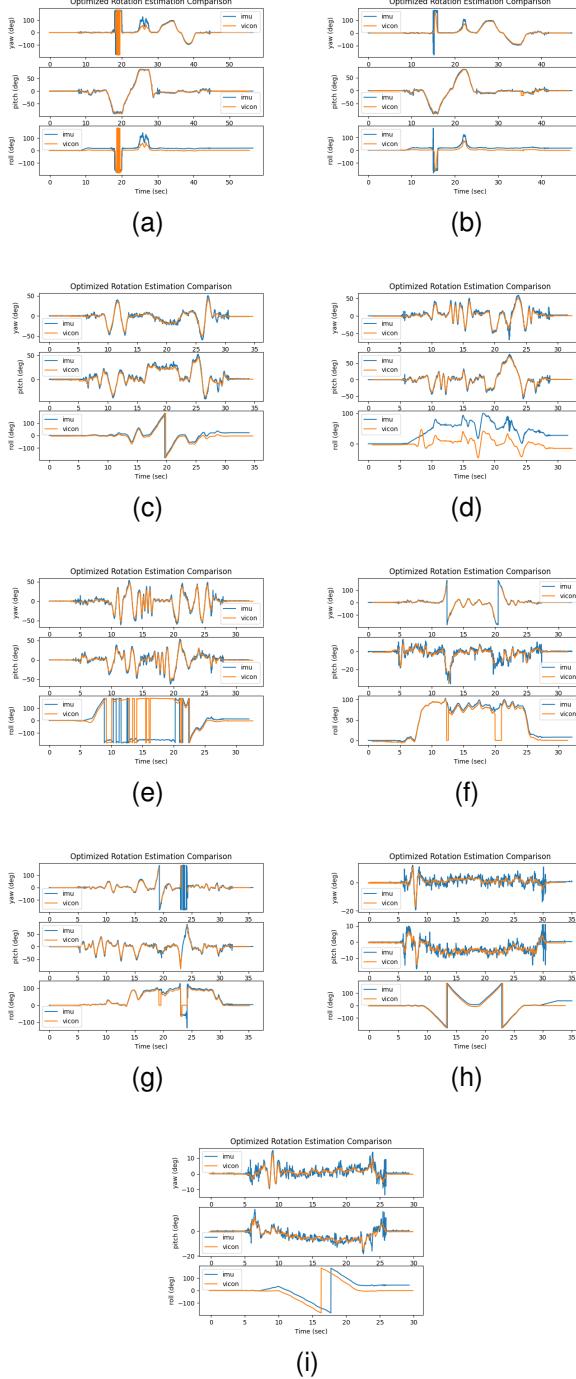


Fig. 5. Comparison between estimated rotations and ground truth on all training sets.

failures are, in fact, a common occurrence in practical applications, which is precisely why the integration of multiple sensors is crucial to ensure the stable operation of the system.

C. Estimated Angles on the test sets

We have applied the optimization techniques used on the training datasets to the test datasets, achieving optimized rotation results. These outcomes are displayed in Fig. 6.

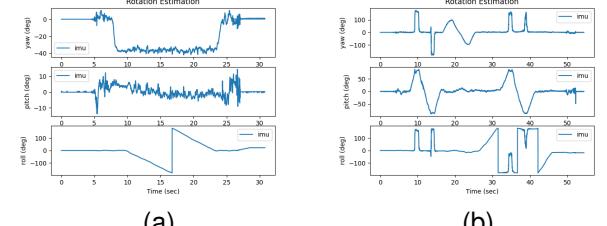


Fig. 6. The optimized rotation on the test set, (a) refers to the test set 10, while (b) refers to the test set 11.

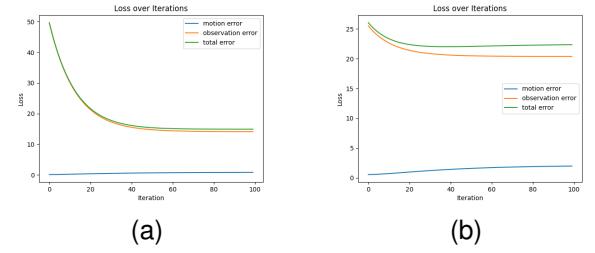


Fig. 7. The training loss on the test set, (a) refers to the test set 10, while (b) refers to the test set 11.

Since there is no ground truth available for the test datasets, we cannot directly determine the accuracy of the estimates from the images alone. Instead, we must infer the precision of our estimates by examining the panoramic images produced later. However, as shown in Fig. 7, the optimization process converged smoothly, which fundamentally indicates that the optimization was conducted correctly.

D. Panorama Image Assessment

In the panorama generation phase on the training sets, we utilized both the estimated rotations and the precise ground truth from the VICON system. The comparative analysis began with the creation of spherical images, as seen in Fig. 8, where the point cloud representation in 3D space was used to evaluate the spatial orientation of our images.

We then transitioned to a cylindrical mapping of these spherical images, as presented in Fig. 9. This step allowed us to analyze the panorama in a 2D unfolded representation, highlighting differences in stacking directions due to the varying rotation axes across different training sets, particularly noticeable in sets 1, 2 versus 8, 9.

The panoramas crafted from VICON data demonstrated smoother image transitions, showcasing the accuracy of the ground truth. In contrast, the panoramas based on our estimated rotations, despite largely aligning the images correctly, revealed some jagged edges in sections where the estimation accuracy was not optimal.

Following the methodology established with the training sets, we proceeded to perform the same panoramic stitching process on the test sets. As depicted in Fig. 10, we first constructed a three-dimensional spherical point cloud to visualize the orientation data in a comprehensive 3D space.

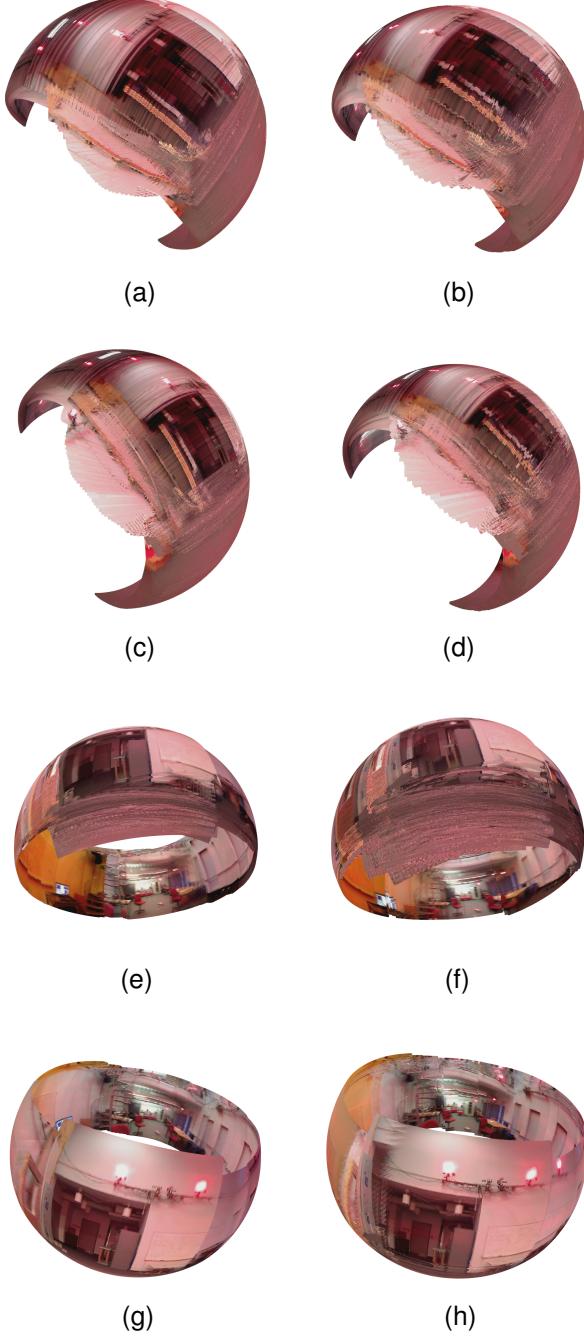


Fig. 8. Spherical point clouds comparison: From top to bottom, the sequences are train set 1, 2, 8, 9. The left side shows point clouds stitched using the ground truth, while the right side shows point clouds stitched using estimated values.

Subsequently, the point cloud was transformed into a cylindrical projection, showcased in Fig. 11, where we unfolded the images into a two-dimensional plane for a more accessible visual analysis. It can be clearly seen that the stitching effect of the panoramic images is very good.

Notably, test set 10 exhibited a horizontal rotation spanning a full circle, as evidenced by the resulting panorama with images stacked neatly in parallel. In contrast, test set 11 underwent rotations both horizontally and vertically, completing a full rotation in each plane. This unique movement

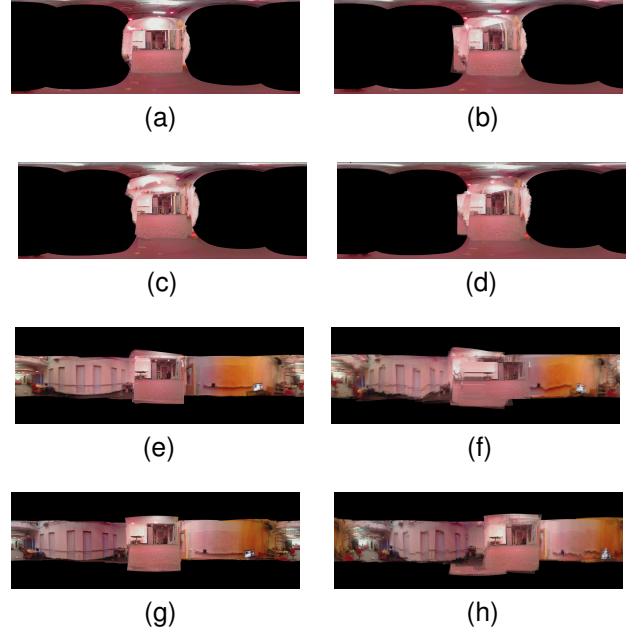


Fig. 9. Panoramic comparison: From top to bottom, the sequences are train set 1, 2, 8, 9. The left side shows panoramas stitched using the ground truth, while the right side shows panoramas stitched using estimated values.

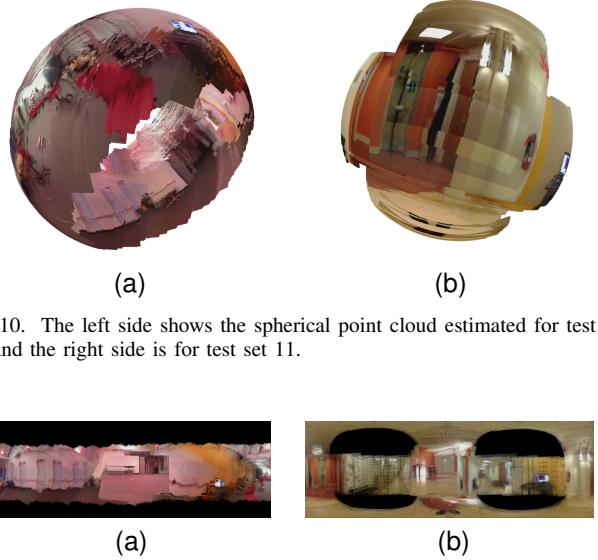


Fig. 10. The left side shows the spherical point cloud estimated for test set 10, and the right side is for test set 11.



Fig. 11. The left side shows the panorama estimated for test set 10, and the right side is for test set 11.

pattern led to a panorama that features not only horizontal image continuity but also a distinctive vertical stacking at the center and sides. This vertical element adds a layer of complexity to the panorama, illustrating the algorithm's ability to handle compound rotational movements and produce a coherent panoramic image that accurately reflects the dynamic range of rotations encountered in the dataset.

V. CONCLUSION

In conclusion, this project demonstrated a successful approach to estimating the 3D orientation of a rotating body

using IMU data and generating panoramic images aligned with these estimates. Key achievements include accurate IMU calibration, effective orientation estimation through optimization, and the creation of panoramic images from both training and test datasets. Challenges such as sensor glitches were identified, highlighting areas for future improvement. The project validates the feasibility of using estimated orientations for practical applications like panorama stitching, setting a foundation for further research and development in sensor-based orientation tracking and image processing.

REFERENCES

- [1] Joriki. “Gradient Descent with Constraints.” Mathematics Stack Exchange. [Online]. Available: <https://math.stackexchange.com/questions/54855/gradient-descent-with-constraints>. Accessed on February 6, 2024.