Panoramic Image Generation with IMU and Camera

1st Weixiao Zhan
weixiao-zhan[at]ucsd[dot]edu

I. Introduction

This report introduces a method for generating panoramic images by integrating a camera with an Internal Measurement Units (IMU), which includes accelerometer and angular velocity sensor.

In many single camera scenarios, the limited fields of view presents a challenge: it is difficult to capture large objects within a single frame Meanwhile, feature matching based panoramic often requires lots of computation, and falls short on latency sensitive applications.

To address these issues, I propose a method that utilizes IMU data to facilitate quick and accurate panorama image generation. This method has broad applicability; for example, smartphones could utilize it to capture panoramic images, while robots could employ it for environmental mapping.

The methodology involves: 1) fitting an observation model based on accelerometer data and a motion model based the angular velocity sensor; 2) using gradient descent to estimate a series of body rotations that best fit both models; 3) using the estimated rotations and spherical projection techniques to generate a panoramic image. This process could offer a more streamlined and automated solution for panoramic images.

II. Problem Formulation

I assume the IMU and camera are closely fixed together, and the combined IMU-camera unit experiences only pure rotations. I also neglect the translation difference between the IMU and camera frames, assuming the centers of the camera frame and IMU frame are aligned with the body frame.

The IMU and camera are taking samples at different interval. Let $N^{(IMU)}$ and $N^{(cam)}$ represent the number of samples for the IMU and camera, respectively. Let $T_t^{(IMU)}$ and $T_t^{(cam)}$ denote the timestamp of t-th sample for the IMU and camera, respectively. Define the step index $t := \left[0 : N^{(IMU)}\right)$. The world frame is define to be aligned with body frame at time step 0.

The t-th IMU data includes accelerometer $a_t \in \mathbb{R}^3$ and angular velocity $w_t \in \mathbb{R}^3$, both in body frame. I use a quaternion $q_t \in H_*$ to represent the rotation from body to world fame at timestamp $T_t^{(IMU)}$. The interval between each time step can be calculated as:

$$\tau = T^{(IMU)}_{[1:]} - T^{(IMU)}_{[:-1]}$$

A. Motion and Observation Model

The motion model infers the next rotation from current rotation and angular velocity. Formally:

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

The observation model infers current rotation from current gravity acceleration. Formally:

$$a_t \sim h(q_t) = q_t^{-1} \circ [0, a_0] \circ q_t$$

In which, a_0 is the gravity acceleration at time step 0. By definition, this is also the gravity acceleration in world frame.

B. Loss Function

Our best best estimate rotations should fits in both the motion and observation model. Formally:

$$\text{mloss} = \sum_{t=0}^{N^{(IMU)}-2} \left\| 2 \log\left(q_{t+1}^{-1} \circ f\left(q_t, \tau_t, w_t\right)\right) \right\|_2^2$$

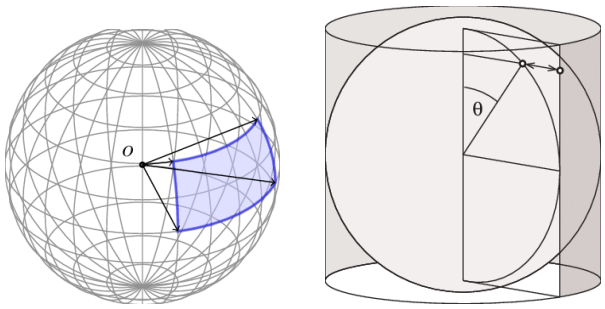
$$\text{oloss} = \sum_{t=1}^{N^{(IMU)}-1} \left\| a_t - h(q_t) \right\|_2^2$$

$$\text{best estimate} = \arg\min_{\forall q} \left(\lambda \text{mloss} + (1-\lambda)\text{oloss}\right)$$

In which, I introduced preference coefficient λ to adjust how much the estimation should favor motion model or observation model.

C. Panorama

After obtained estimated rotations, I project images to a sphere (illustrates in Figure 1a), apply rotations, and finally project to a cylinder image (illustrates in Figure 1b).



(a) FOV projection (b) cylinder projection

Fig. 1: Projections

Let the camera image has $\left(\text{imglen}_x \times \text{imglen}_y\right)$ pixels and $\left(\text{imgfov}_x \times \text{imgfov}_y\right)$ radian field of view. Thus, the radius of intermediate sphere $r := \frac{\text{imglen}}{\text{imgfov}}$ pixels, the hight of cylinder is $2r$, and the length of cylinder is the circumference of sphere $2\pi r$. Let $\left(\text{img}_x^{(t)}, \text{img}_y^{(t)}\right)$ denote one pixel's coordinate on raw image, and $\left(\text{cylinder}_x, \text{cylinder}_y\right)$ denote that pixel projection's coordinate on cylinder image.

1) Pixel's coordinate on raw image are mapped to spherical coordinates with radius r, then converted to Cartesian coordinates.

$$\begin{bmatrix}\phi\\ \theta\end{bmatrix} = \begin{bmatrix}\text{imgfov}_x(\frac{1}{2} - \frac{\text{img}_x^{(t)}}{\text{imglen}_x})\\ \text{imgfov}_y(\frac{1}{2} - \frac{\text{img}_y^{(t)}}{\text{imglen}_y})\end{bmatrix}$$

$$\begin{bmatrix}x\\ y\\ z\end{bmatrix}_c = r\begin{bmatrix}\cos\theta\cos\phi\\ \cos\theta\sin\phi\\ \sin\theta\end{bmatrix}$$

2) Each image is associated with the estimated rotation with closest-in-the-past timestamp q_t and the ground truth rotation matrix R_t. Apply the associated rotations to Cartesian coordinates.

$$\begin{bmatrix}0 & x & y & z\end{bmatrix}_r = q_t \circ \begin{bmatrix}0 & x & y & z\end{bmatrix}_c \circ q_t^{-1}$$

$$\begin{bmatrix}x\\ y\\ z\end{bmatrix}_r = R_t \begin{bmatrix}x\\ y\\ z\end{bmatrix}_c$$

3) Use an area-preserving mapping, shown in Figure 1b, from sphere onto a cylinder.

$$\begin{bmatrix}\text{cylinder}_x\\ \text{cylinder}_y\end{bmatrix} = \begin{bmatrix}(\pi - \arctan(y_r, x_r))\, r\\ r - z_r\end{bmatrix}$$

4) With coordinate mapping, set the pixel on cylinder to RGB value of the pixel on original image.

$$\text{cylinder}\left[\text{cylinder}_x, \text{cylinder}_y\right] \leftarrow \text{img}^{(t)}\left[\text{img}_x, \text{img}_y\right]$$

III. Technical Approach

A. Data preprocessing

Following the guidelines in the IMU reference, I converted the raw data to physical units. I observed that the bias in angular velocity significantly varies across the three axes. Assigning a unique bias to each axis substantially reduced the error in the motion model. However, the accelerometer did not exhibit such a bias variance across its three axes.

B. Initial values

I employed two methods of generating my initial values. The first method involves using motion model and intergrade over time. The second method relies on observation models. I utilized the cross product of a_t and a_0 to obtained axis-angle representation of rotation from body frame to world frame, which I then convert it to quaternion. The results indicate that initial values derived from the observation model sometimes resulted in better outcomes, with smaller loss compared to those obtained from the motion model and random initial values. However, this advantage diminishes if the body experiences not only pure rotation but also some acceleration, making the observation model less reliable due to inertial forces.

C. Gradient descent

I used NumPy and Pytorch(CPU) to manage in-memory data and accelerate computation; I implemented quaternion exp and log [1] , motion model f, observation model h and loss function in PyTorch to allow auto-gradient. Additionally, an early-stop mechanism is incorporated into the training loop.

The preference coefficient λ is set to $\frac{1}{2}$, learning rate to 1e-3, training epoch to 6000, and early stop patient to 10. In most trainings, the early-stop breaks the training loop before epoch running out .

D. Projections

$$\text{imglen}_x = 320, \text{imglen}_y = 240$$

$$\text{imgfov}_x = \frac{\pi}{6}, \text{imgfov}_y = \frac{\pi}{8}$$

The radius r is round to 306 and the circumference is round to 1923.

IV. Results

Data set 1,2,8,9,10,11,3,4,5,6,7 are present in Figure 2 to 12. The average loss on train set is 6.561, and average loss on test set is 7.946.

Sub-figure a(s) are the quaternion representation of the rotations. Sub-figure b(s) are the Euler angles representation of the rotation. In Sub-figure a(s) and b(s), the solid blue line shows the gradient descent results; the orange and green dot lints show the motion and observation model's initial values; the red dash line shows the ground truth. Sub-figure c(s) show the panorama image based on gradient descent rotations; sub-figure d(s) show the panorama image based on ground truth rotations. Full size images can be find at `img/` directory.

Findings

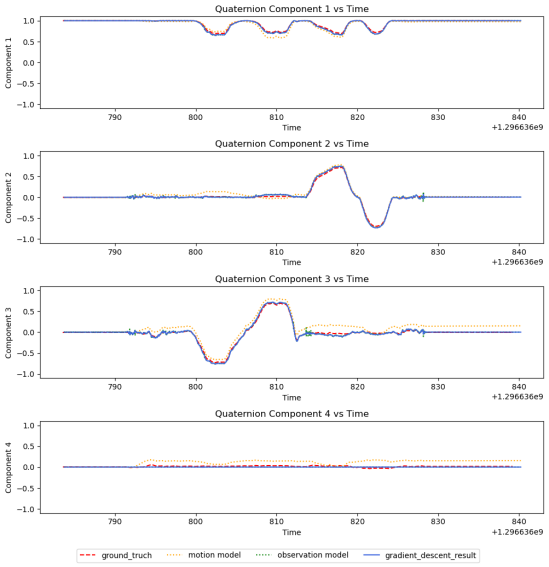
1) In must dataset, the gradient descent results followed the ground truth. The projection successfully assembles images into coherent panoramic views of the environment.
2) Quaternion q and $-q$ are representing the same rotation, Therefore, in certain sections, such as sub-figure 4a and 5a, the gradient descent results and ground truth appear flipped on the diagram, despite representing proximate rotations.
3) Euler angles exhibit singularities, causing oscillations in the plots, as observed in sub-figure 2b. This makes Euler angles difficult to interpret in those cases.
4) In some datasets, the discrepancy between motion model and observation models is minor. However, in others, such as data set 9, 10 and 11, observation model significantly deviates from both the motion model and the ground truth. This deviation could be attributed to 360-degree rotations performed by the body, during which the linear accelerations become significant and breaks the pure rotation assumption.
5) The last approximately 20% of the camera images in data set 1, 2 have translations, specifically, a vertical

[1] quaternion norms are added 1e-12 to prevent divide-by-zero

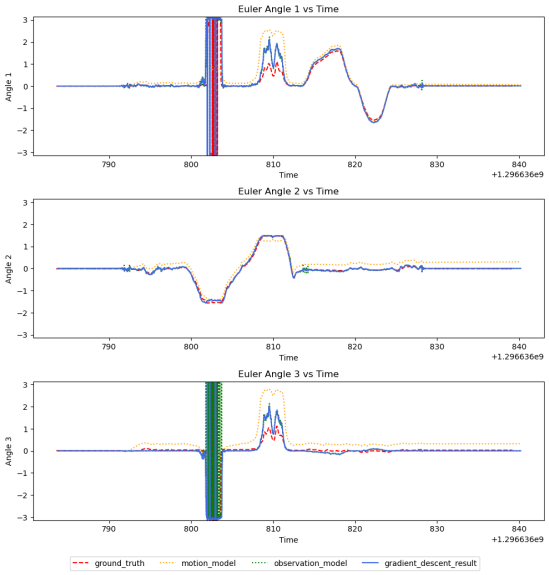
downward movement. This contradicts the foundational assumption of purely rotational body frame movement. Consequently, these images were manually excluded from panorama stitching.

6) A model incorporating the translational differences between the camera and IMU might further enhance the quality of the generated panoramas.
7) Utilizing the angular velocity sensor for rotation estimation and the accelerometer for velocity estimation could potentially allow for a comprehensive full body poses estimation, offering a more accurate model of camera movement.

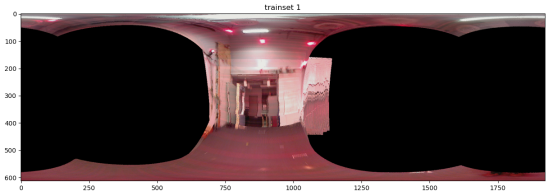
Overall, the estimated rotation allows realistic panorama generation. The in-perfection could be largely caused by non-pure rotation and numerical errors. The most challenging part of the project is to calibrate the IMU data, finding out their correct signs and bias.



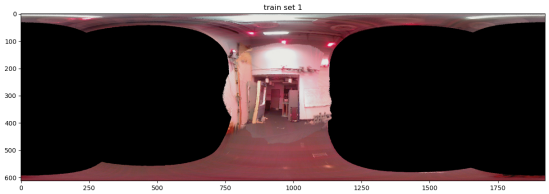
(a) quaternion



(b) Euler angles

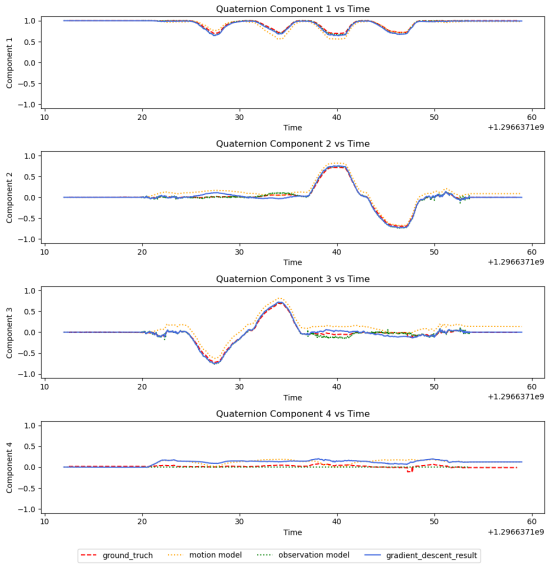


(c) gradient descent results

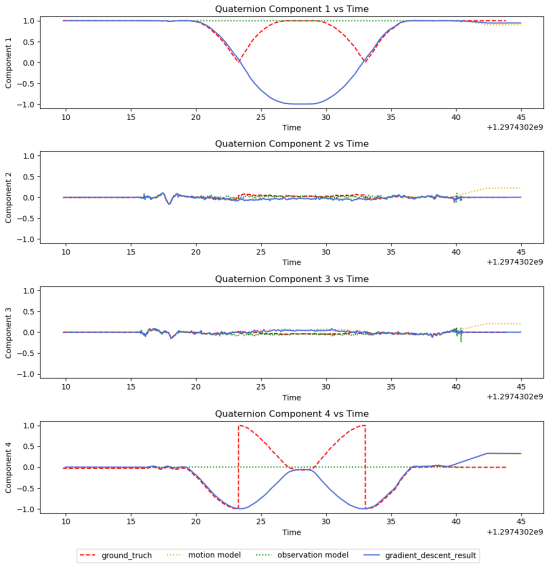


(d) ground truth results

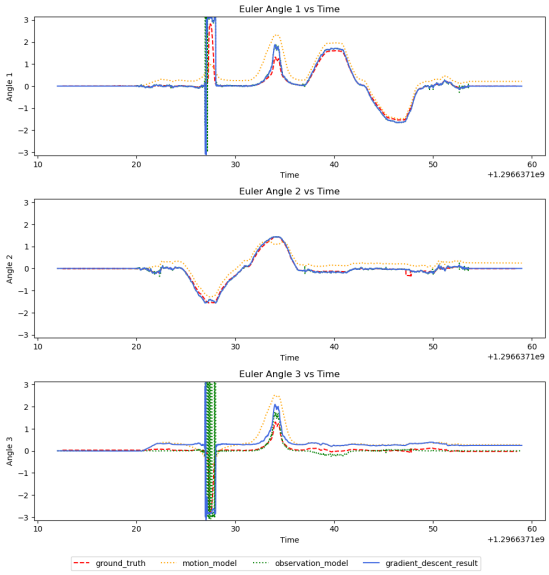
Fig. 2: Train set 1



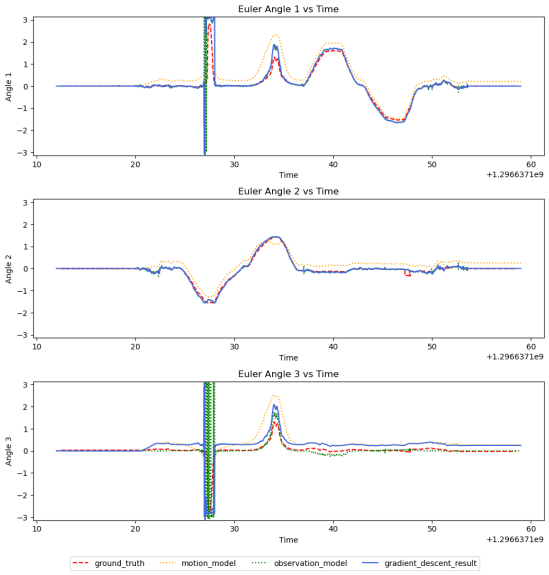
(a) quaternion



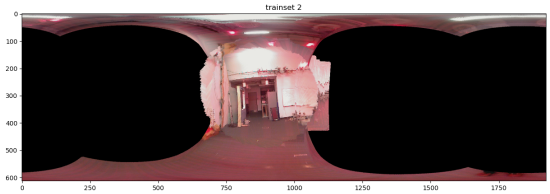
(a) quaternion



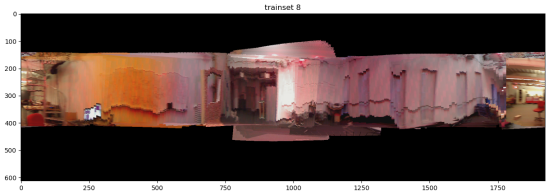
(b) Euler angles



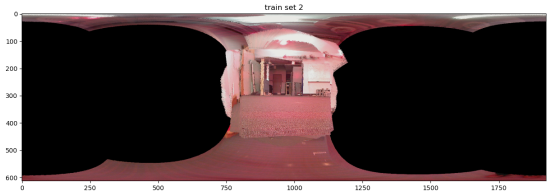
(b) Euler angles



(c) gradient descent results

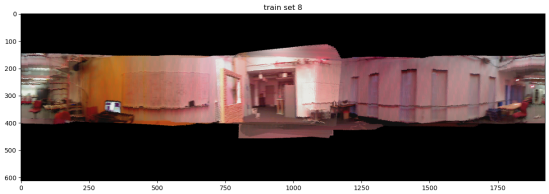


(c) gradient descent results



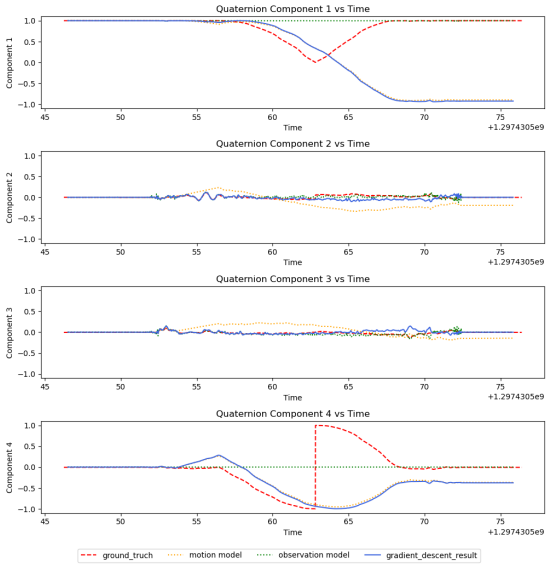
(d) ground truth results

Fig. 3: Train set 2

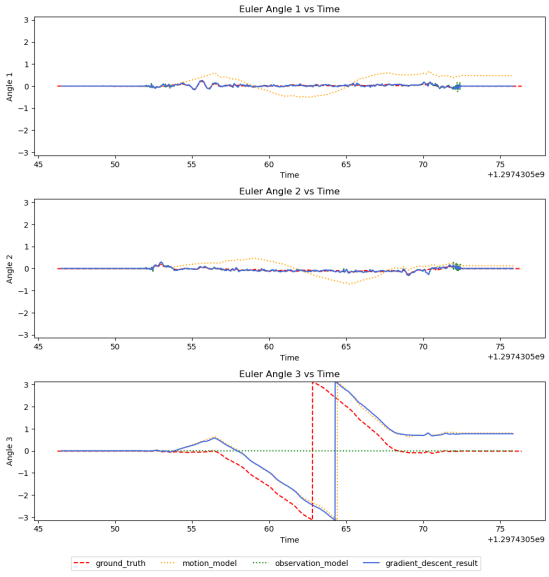


(d) ground truth results

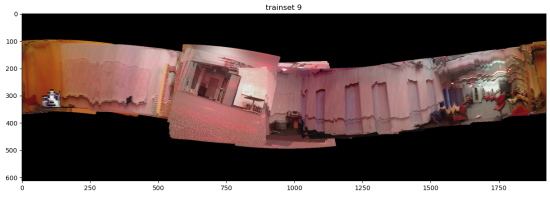
Fig. 4: Train set 8



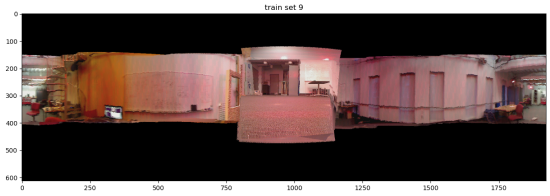
(a) quaternion



(b) Euler angles

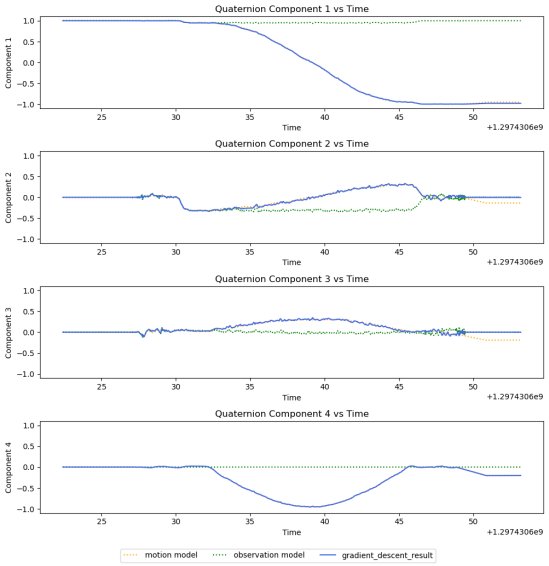


(c) gradient descent results

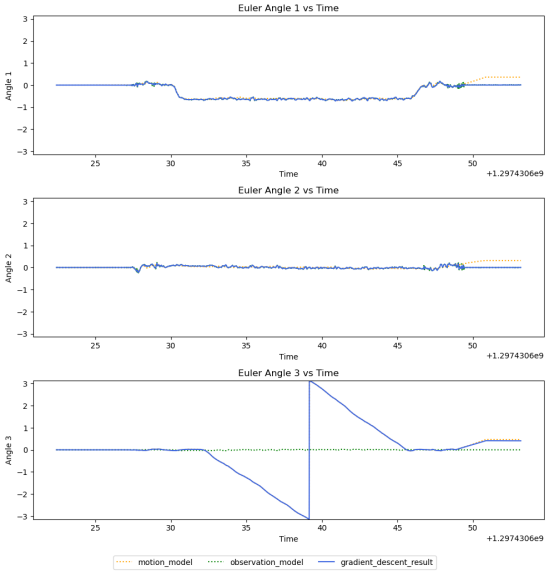


(d) ground truth results

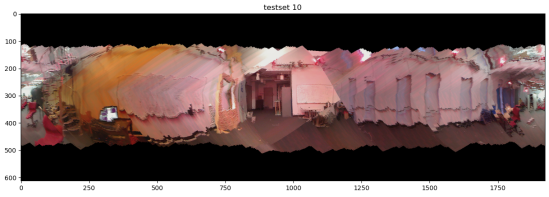
Fig. 5: Train set 9



(a) quaternion

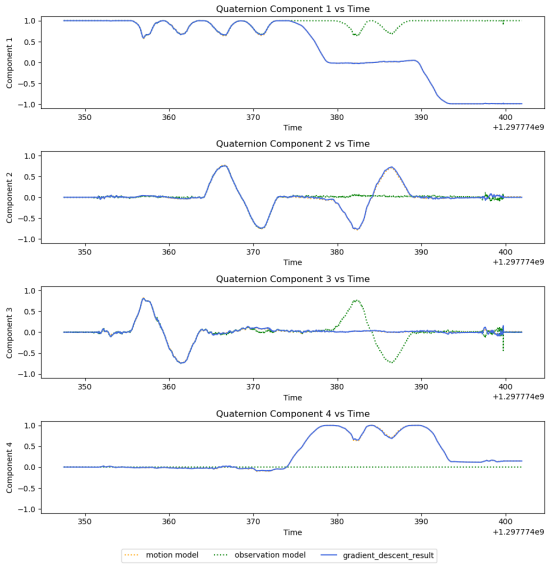


(b) Euler angles

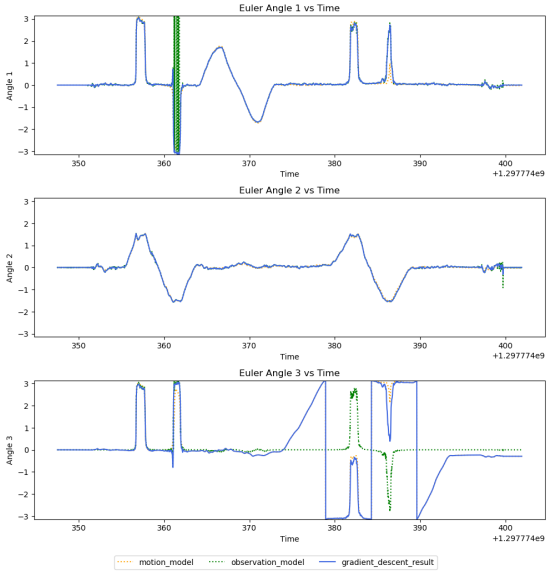


(c) gradient descent results

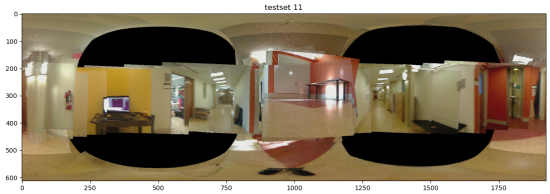
Fig. 6: Test set 10



(a) quaternion

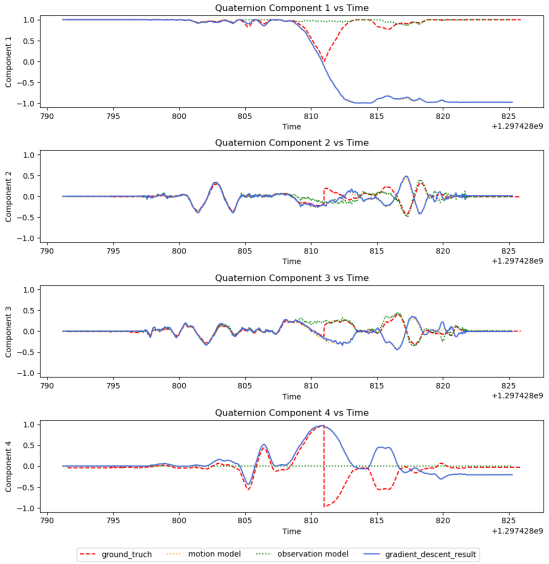


(b) Euler angles

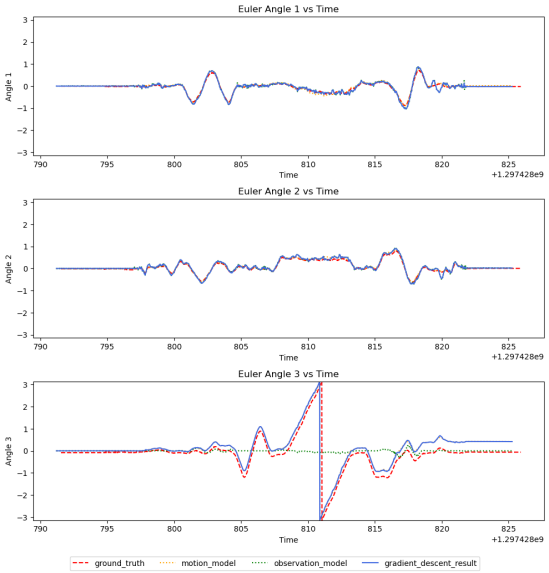


(c) gradient descent results

Fig. 7: Test set 11

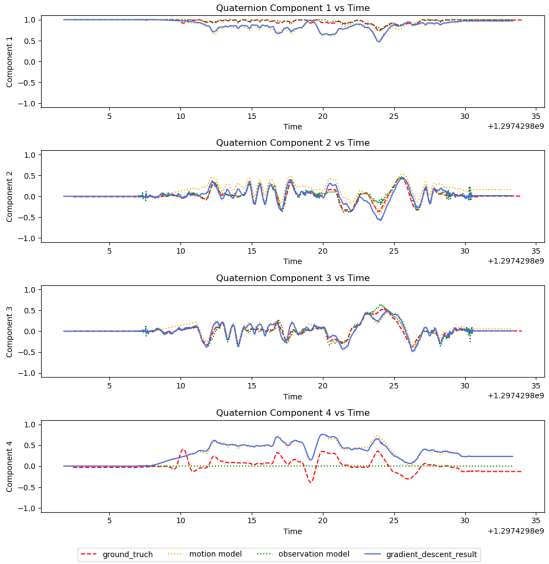


(a) quaternion

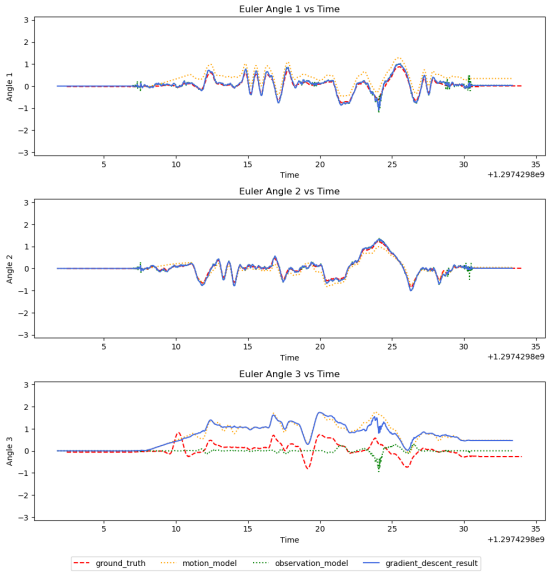


(b) Euler angles

Fig. 8: Test set 3

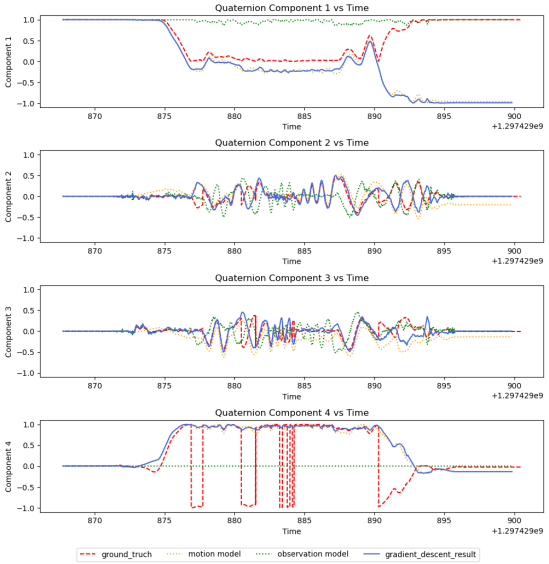


(a) quaternion

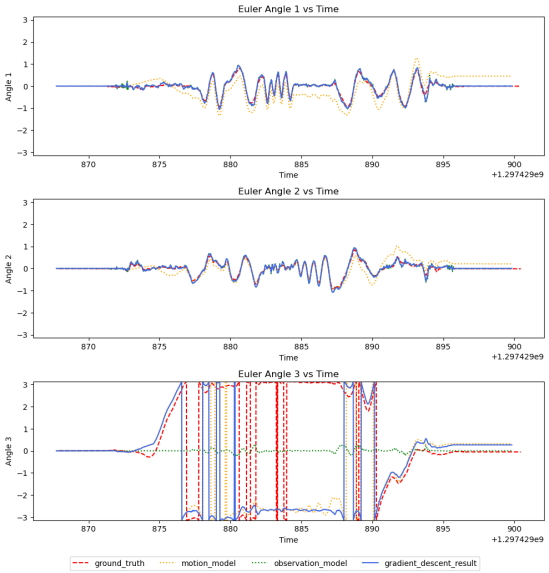


(b) Euler angles

Fig. 9: Test set 4

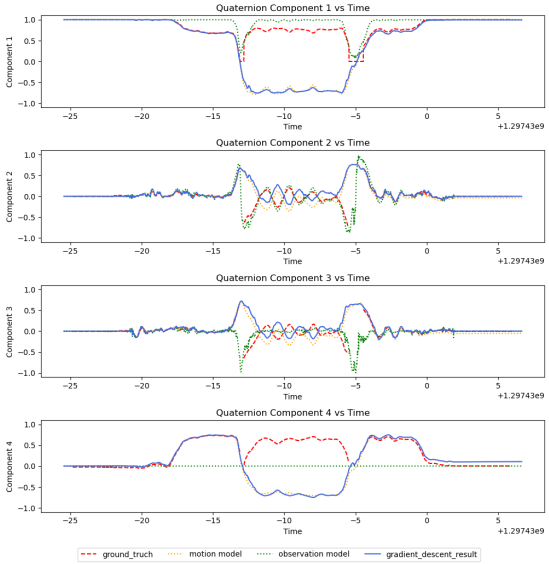


(a) quaternion

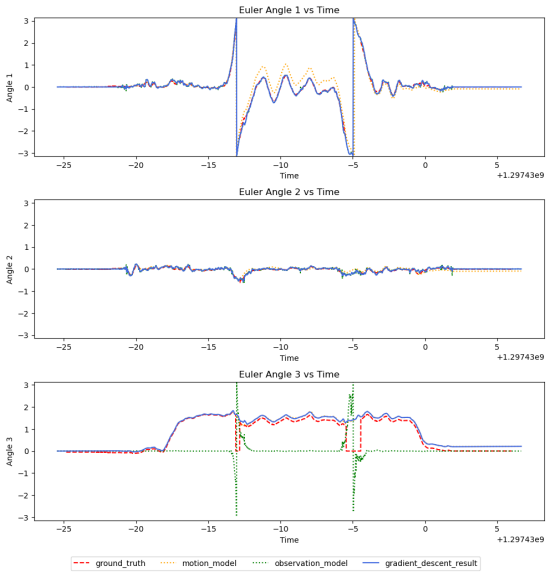


(b) Euler angles

Fig. 10: Test set 5

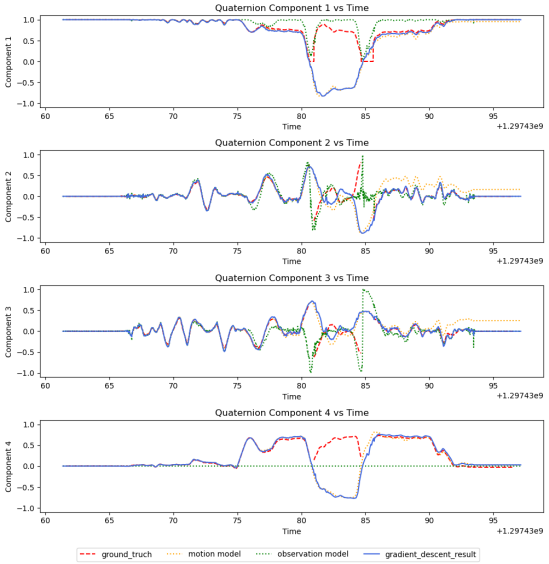


(a) quaternion

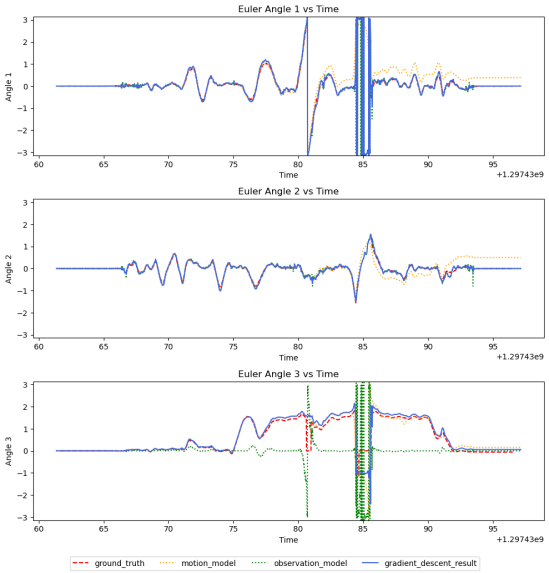


(b) Euler angles

Fig. 11: Test set 6



(a) quaternion



(b) Euler angles

Fig. 12: Test set 7