ECE276A: Panaroma

Ashish Maknikar
PID: A59018297
UC San Diego

***Abstract*—The project is a formulation where we have raw IMU and ground truth VICON sensor data from a mobile robot. The aim of the exercise is to understand how IMU inertial data can be used to estimate the pose of the camera on the mobile robot. The pose of the camera at different time steps can then be used to generate a panorama from the captured images.**

I. Introduction

There are many situations where we might not have access to the accurate pose of the target frame in a mobile robot. Some examples could include the camera frame in aerial, on-ground or underwater autonomous vehicles.

In such situations, we have access to the initial pose of the frame and the consequent pose trajectory cum the final pose can be estimated using historical inertial data captured from an attached Inertial Measurement Unit(IMU) sensor that measure the linear and angular accelerations of the robot along all three axes. We must leverage all available information to estimate the optimal target frame poses through the trajectory of the robot. The tracked orientation can then be used to stitch together a panorama from the images capture by the camera.

II. Problem formulation

In the experiment, we have sensor gyroscope and accelerometer sensor raw data for a purely rotational motion. We must formulate a solution to stitch images from the camera undergoing the motion into a panorama.

To stitch the camera images into a panorama we require the pose information of the camera at the timestamps the image is captured. Using this pose data we can map the images onto a sphere and then flatten the sphere to obtain a flat panorama canvas.

We know that the initial pose

$$\mathbf{q}_0 = [1\ 0\ 0\ 0]$$

. Using angular acceleration obtained from the IMU gyroscope, we can formulate the poses for the later time steps using the **motion model**

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

. where:

$\mathbf{q}_t$	quaternion pose at time t
τ_t	time step
ω_t	angular acceleration
exp()	exponential function for quaternions

We know the robot is not moving linearly, thereby constraining the linear acceleration to gravity alone. Hence the measured acceleration in the IMU fram must agree with the gravity acceleration in the world frame. The transformed acceleration in the IMU frame is given by the **observation model**

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

An optimisation problem can be formulated to estimate the orientation trajectory $\mathbf{q}_{1:T}$ by combining the model in (1) and (2). We formulate a cost function as follows

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

Minimising the cost function over $\mathbf{q}$ should give us the optimal orientation trajectory subject to $||\mathbf{q}|| = 1$.

$$\min_{\mathbf{q}_{1:T}} c(\mathbf{q}_{1:T}) \tag{4}$$
$$s.t\ ||\mathbf{q}|| = 1\ \ \forall t \in [1:T]$$

III. Technical Approach

We split our tasks into three parts, namely pose calculation from the motion model (1) , optimising the orientation trajectory using gradient descent optimisation for the cost function (3) and panorama generation using the above pose information.

A. IMU Raw Data to Accelerations

We initially have raw A/D IMU data (Fig 1) that must be converted to their respective acceleration units using the sensitivity and scale factor with the following equations:

$$\text{value} = (\text{raw} - \text{bias}) * \text{scale factor} \tag{5}$$

$$\text{scale factor} = \text{Vref}/1023/\text{sensitivity} \tag{6}$$

We use the following IMU sensor data sensitivity

linear acceleration(mV/unit gravity accn(g))	300 *mV/g*
angular acceleration(mV/°/s)	3.33 *mV/ °/g*

To calculate the bias in (5) we know that the robot is immobile for the first few time-steps. Analysing the IMU sensor data, we slice the initial flat portion of the sensor data and average it out to calculate the bias.

Using (5) and (6) we can now obtain the physical values of the sensor accelerations (Fig. 2). You can observe that the z acceleration is units of gravity and that unit gravity as added to the bias.

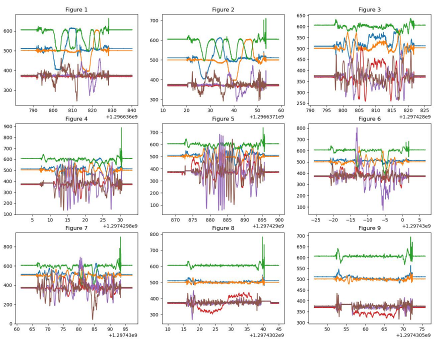

Fig. 1. All IMU Data for testset

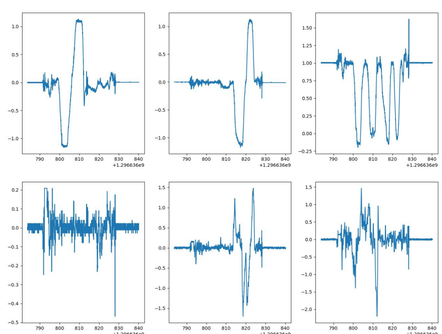

Fig. 2. Example of IMU data after removing bias and scaling

B. Pose Calculation

The motion model in (1) is used to calculate the quaternion poses $\mathbf{q}_{0:T}$ using the initial $\mathbf{q}_0$ value. The results of the above can be verified by comparing the calculated $\mathbf{q}_{0:T}$ to the VICON data rotation matrices. Although the two plots are not concurrent exactly, they follow a similar pattern, confirming out model. Their comparison is visualised in Fig 3

The linear accelerometer sensor data can be compared to the calculated $\mathbf{a}_t$ in the observation model (2). We can see the plots verifying observation model for the training (Fig. 4) and test (Fig. 5) sets respectively.

C. Gradient Descent Optimisation

We can optimise the calculated orientation trajectory $q_{1:T}$ by minimising the cost function defined in (3) subject to $||\mathbf{q}|| = 1$. We utilise projected gradient descent to perform this optimisation.

The projected gradient descent takes the gradient of the the cost funtion (3) with respect to $q_{1:T}$ and subtracts the scaled gradient from $q_{1:T}$ ensuring convergence of the cost function (3) to a minimum. This is followed by normalising

Fig. 3. Verifying Motion Model by comparing Roll,Pitch and Yaw with VICON **before optimisation**

Fig. 4. Verifying Observation Model linear acc^n with IMU accelerometer data for training set

the resulting quaternion.

$$q^{i+1}_{1:T} = q^{i}_{1:T} - \alpha * \nabla c(q^{i}_{1:T}) \tag{7}$$

$$q^{i+1}_{1:T} = \frac{q^{i+1}_{1:T}}{||q^{i+1}_{1:T}||} \tag{8}$$

where:

α The step size.

For the purpose of the project, we use a α value of 0.01.

The above steps are repeated until the cost functions keeps decreasing to a stable minimum.

We utilise jax in our code to calculate the gradient and the cost function as well as performing all quaternion operations.

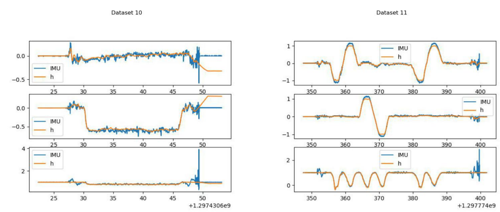

Fig. 5. Verifying Observation Model linear acc^n with IMU accelerometer data for test set

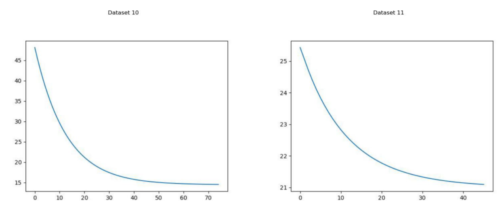

Fig. 7. Gradient Descent for test set

Care is taken to prevent a zero denominator and zero log to prevent the gradient from exploding to infinity.

We can see the reduction in cost function as the number of steps increase in Fig 6 for the training set and Fig. 7 for the test set respectively.

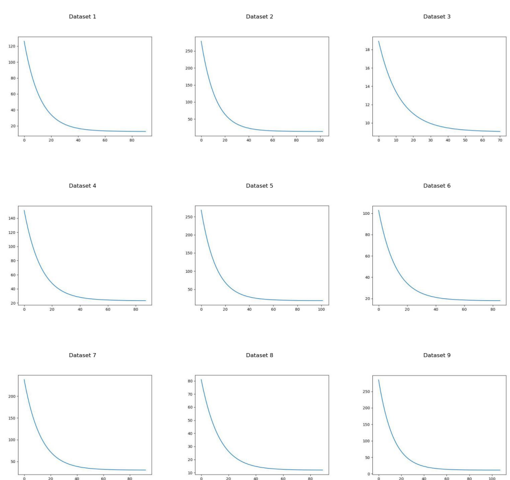

Fig. 6. Gradient Descent for training set

D. Panorama Generation

We now have the orientation data and the time steps. We observe that the images are of size 320×240. In case of the non-test data, VICON rotational matrices are used as the source for pose of the end effector. For test camera data, the generated quaternion pose data is used for orientation estimation of the camera image. The following steps are followed to generate the panorama:

- We take the vertical and horizontal fields of view to be $45°$ and $60°$ respectively.
- A mesh of size 320×240 with the (0,0) value being $[30°, -22.5°]$ is generated (credits: piazza answer, this is important as other systems are splitting the image into the four corners od the panorama). Each value in the mesh corresponds to the latitude(ϕ) and longitude (λ) of the spherical projection of the image on a sphere of radius 1 with the image center being at [1,0,0] spherical coordinates. An example is shown at Fig. 8

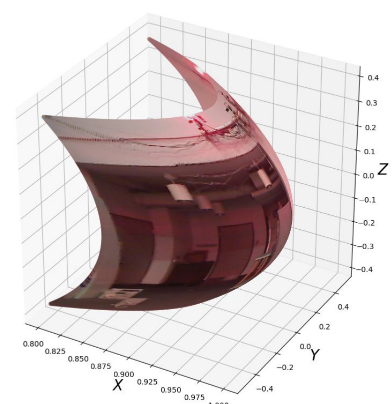

Fig. 8. Projection of a camera image onto a sphere. Image center at [1,0,0] spherrical coordinates

- For the timestamp at the image, the orientation with the closest timestamp gives the orientation of the image on the sphere. We apply the rotation to rotate the spherical projection in the previous point to get it to its actual pose(as shown in Fig 9). The coordinates are first converted from spherical to Cartesian to apply the rotation according to our defined Cartesian system and then reconverted to the ($1, \lambda, \phi$) spherical coordinates.

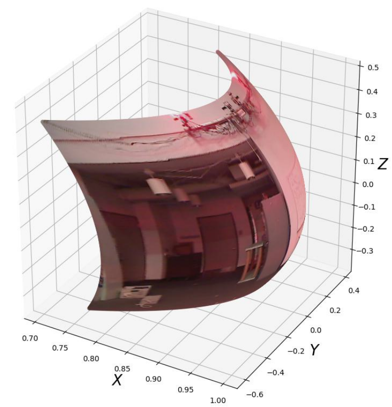

Fig. 9. Projection of Fig 8 after applying rotation

- We project the sphere onto a cylinder on a 1280×920 canvas using the following equation:

$$[x, y] = [(\phi + (\pi/2)) \times \frac{960}{\pi}, (\lambda + \pi) \times \frac{1280}{\pi}] \qquad (9)$$

The addition of $(\pi/2)$ and π to ϕ and λ respectively is to remove negative terms before scaling them onto the flat canvas.

IV. Results

The following plots are generated to verify the calculated data

- Plots comparing estimated roll, pitch, and yaw angles to the ground truth on the training sets: **Fig 10**
- plots showing your estimated roll, pitch, and yaw angles on the test sets, **Fig 11**
- Training Set panoramas, **Fig 12**
- Test Set panoramas, **Fig 13**

A. Observations

We make the following observations during the analysis of the results.

- The optimisation only helps to a certian degree. Optimisation is not effective if there is a big initial difference in the sensor values and ground truth, as shown in the Dataset 4 yaw values (reference Fig 3 and Fig 10).
- The optimisation does not work if the times are shifted as shown for the yaw component of Dataset 9.(reference Fig 3 and Fig 10)
- We can see that the roll and pitch in the test Dataset 10 is very noisy. This has resulted in a blurry panaroma for the same. (reference 11 and 13)

Fig. 10. Comparison of VICON and optimised roll pitch and yaw for training set

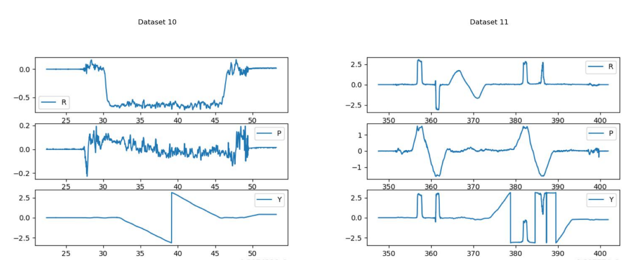

Fig. 11. Estimated roll, pitch, and yaw angles for test

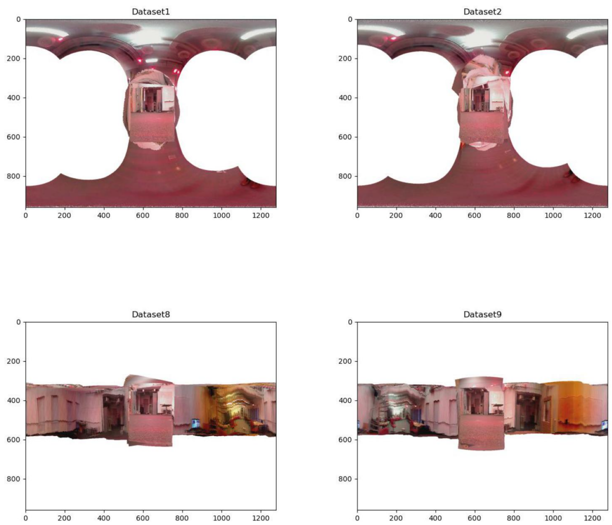

Fig. 12. Panoramas generated for training sets

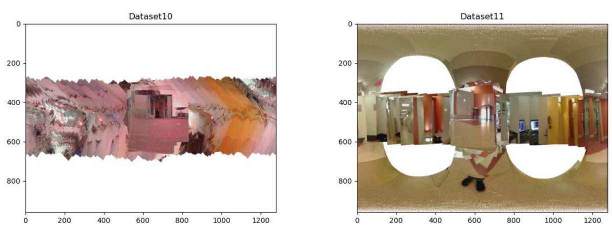

Fig. 13. Panoramas generated for test sets

- We find that the adopted cylinder unrolling strategy is only good if the axis of the cylinder is perpendicular to camera motion. It results in warped panoramas as seen in Datasets 1,2 and 11)(reference 13 and 12). No time to try different coordinate systems.

YOU CAN SEE THE ORIGINAL IMAGES AT https://drive.google.com/drive/folders/1m4ssHqBc-YiTOSUmLz44pU5Jkrr9pex3?usp=share_link