Project 1: Orientation Tracking

*Collaboration in the sense of discussion is allowed, however, the assignment is **individual** and the work you turn in should be entirely your own. It is absolutely forbidden to copy or even look at anyone else's code or any code generated by AI. Please acknowledge **in writing** people or AI you discuss the project with. See the collaboration and academic integrity statement here:* https://natanaso.github.io/ece276a.

Submission

Please submit the following files on **Gradescope** by the deadline shown at the top right corner.

1. **Programming assignment**: upload all code you have written for the project (do not include the training and testing datasets) and a README file with a clear concise description of the main files and how to run your code.

2. **Report**: upload your report in pdf format. You are encouraged to but not required to use an IEEE conference template[1] for your report. Please refer to Lecture 1 for the expected report structure and content[2].

Project Description

In this project, we will implement a projected gradient descent algorithm to track the 3-D orientation of a rotating body using measurements from an inertial measurement unit (IMU). Using your orientation estimates, generate a panoramic image by stitching camera images obtained by the rotating body. Instructions and tips follow.

- **Training data**: The training data is available at https://ucsdcloud-my.sharepoint.com/:f:/g/personal/natanasov_ucsd_edu/IgBek3pBUhg4Rr-TP7oMZXikARvskUMNOCnORDml2BNz878?e=e3WLBo. There are 9 different training sets, each consisting of the following:
  - **IMU data** recording the linear acceleration and angular velocity of the body in the IMU frame as well as the unix time stamps (seconds) at which the measurements were obtained,
  - **CAM data** providing 320×240 RGB images and unix time stamps at which the images were obtained,
  - **VICON data** providing ground-truth rotation matrices of the body's orientation obtained from a VICON motion capture system.

  The IMU, CAM, and VICON data are paired into 9 sequences according to the file numbers. Note that camera data is available only for 4 of the 9 sequences.

- **Test data**: The test data will be released on 01/30/2026, giving you a couple of days to report how your algorithms perform on the test set. The test data is intended for running your algorithm and discussing the results in your project report, rather than for fine tuning the performance of your algorithm to get the best possible results. This is why you will only be given a couple of days to report results on the test data. The test data will be available at the same link: https://ucsdcloud-my.sharepoint.com/:f:/g/personal/natanasov_ucsd_edu/IgBek3pBUhg4Rr-TP7oMZXikARvskUMNOCnORDml2BNz878?e=e3WLBo.

- **IMU calibration**: The biases of the accelerometers and gyroscopes are unknown and should be estimated using the VICON ground-truth data. Please refer to the IMU_reference.pdf file and the IMU datasheets in the docs folder for information. Look up the accelerometer and gyro sensitivies on the datasheet, using the 4x (instead of 1x) gyro signals to read the appropriate sensitivity. Make sure you convert sensitivity to mV/rad/sec (not degrees!). The camera axis is aligned with the IMU x axis. The relative position of the camera with respect to the IMU is roughly $(0, 0, 0.1)$, i.e., 10 cm above the

[1] https://www.ieee.org/conferences/publishing/templates

[2] https://natanaso.github.io/ece276a/ref/ECE276A_1_Introduction.pdf

IMU. See IMUandCamSensors.jpg for a visualization.

In every dataset, the first few seconds are static, i.e., there is no rotation. You can use this static data portions to calibrate the IMU biases, e.g., the acceleration measured by the accelerometer in the first few seconds should be $[0, 0, 1]^\top$ in gravity units, while the angular velocity should be $[0, 0, 0]^\top$ rad/sec.

- [40 pts] **Orientation Tracking**: In the first part of the project, our objective is to estimate the orientation of the body over time using the IMU angular velocity $\boldsymbol{\omega}_t \in \mathbb{R}^3$ and linear acceleration $\mathbf{a}_t \in \mathbb{R}^3$ measurements. We will use a unit quaternion $\mathbf{q}_t \in \mathbb{H}_*$ to represent the body-frame orientation at time t.

  Using the IMU angular velocity measurements $\boldsymbol{\omega}_t$ rad/sec and the differences between consecutive time stamps τ_t sec, we can predict the quaternion at the next step $\mathbf{q}_{t+1}$ using the quaternion kinematics **motion model**:

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

  where $\circ$ denotes quaternion multiplication and $\exp(\cdot)$ is the exponential function for quaternions defined in Lecture 3.

  You can use the motion model in (1) to verify whether your IMU calibration is correct. Starting with $\mathbf{q}_0 = [1, 0, 0, 0]$, implement a simple integration of the angular velocity $\boldsymbol{\omega}_t$ by computing $\mathbf{q}_{t+1} = f(\mathbf{q}_t, \tau_t \boldsymbol{\omega}_t)$. Plot the roll, pitch, yaw obtained from $\mathbf{q}_t$ and compare them to the values provided by the VICON ground truth. These should match relatively well. You may use the python library transforms3d[3] to convert between different orientation representations.

  Since the body is undergoing pure rotation, the acceleration of the body should be approximately $[0, 0, -g]$ m/s^2 in the world frame of reference, where g is the gravity acceleration. Note that this measurement is $[0, 0, 1]$ in gravity units. The measured acceleration $\mathbf{a}_t$ in the IMU frame should agree with gravity acceleration after it is transformed to the IMU frame using the orientation $\mathbf{q}_t$, leading to the following **observation model**:

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

  We will formulate an optimization problem to estimate the orientation trajectory $\mathbf{q}_{1:T} := \mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_T$ based on the motion model in (1) and observation model in (2). A cost function for the optimization problem can be defined as:

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

  where the first term measures the error between the estimated orientation and the motion model prediction, while the second term measures the error between the acceleration measurements and the observation model prediction. The motion model error is based on the relative rotation $\mathbf{q}_{t+1}^{-1} \circ f(\mathbf{q}_t, \tau_t \boldsymbol{\omega}_t)$ between the predicted orientation $f(\mathbf{q}_t, \tau_t \boldsymbol{\omega}_t)$ and the estimated orientation $\mathbf{q}_{t+1}$. The error obtains the axis-angle parametrization of the relative rotation error using the quaternion log function and measures the angle of rotation as the norm of the axis-angle vector. The observation model error measures the agreement between the measured acceleration $[0, \mathbf{a}_t]$ and the predicted acceleration $h(\mathbf{q}_t)$.

  During the optimization, we need to enforce the constraint that the quaternions $\mathbf{q}_t$ remain unit-norm, i.e., $\mathbf{q}_t \in \mathbb{H}_*$. Hence, we have a constrained optimization problem:

$$\begin{aligned} \min_{\mathbf{q}_{1:T}} \;& c(\mathbf{q}_{1:T}) \\ \text{s.t.} \;& \|\mathbf{q}_t\|_2 = 1, \qquad t = 1, \ldots, T. \end{aligned} \tag{4}$$

[3] https://matthew-brett.github.io/transforms3d/

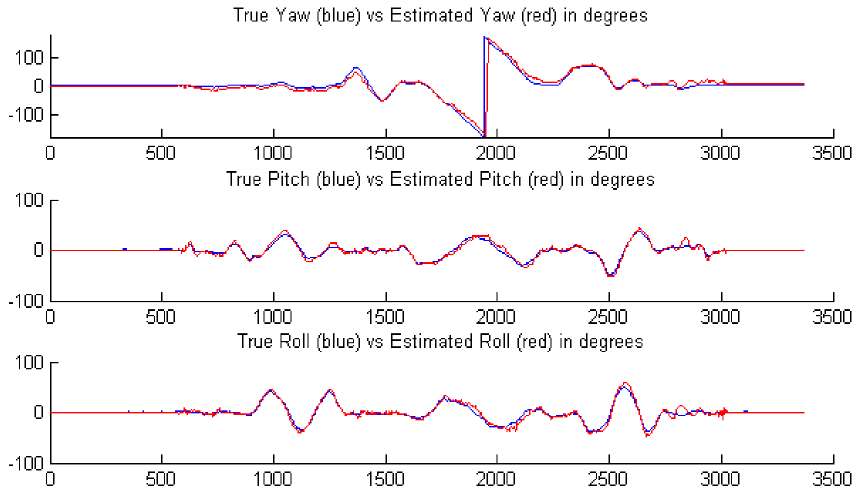

Figure 1: Ground truth (blue) versus estimated (red) roll, pitch, and yaw angles.

Initializing with $\mathbf{q}_0 = [1, 0, 0, 0]$, implement a projected gradient descent algorithm to optimize the quaternion trajectory $\mathbf{q}_t$. You may use the function torch.autograd.functional.jacobian from the pytorch library[4] to compute the gradient of the objective function. Alternatively, you can compute the gradient analytically and implement it yourself.

After you apply the gradient descent step, the simplest projection onto the space of unit quaternions $\mathbb{H}_*$ you may use is:

$$\Pi_{\mathbb{H}_*}(\mathbf{q}) = \frac{\mathbf{q}}{\|\mathbf{q}\|_2}. \tag{5}$$

For other suggestions about how to perform the projection onto $\mathbb{H}_*$, you may refer to this stackexchange post.[5] We will understand where the suggestion in this post comes from later on in the class, when we study manifolds.

Once you have estimated the orientation trajectory $\mathbf{q}_{1:T}$ and using the ground-truth VICON data, plot the estimated versus the ground-truth roll, pitch, and yaw angles of the body. See Fig. 1 for an example.

- [15 pts] **Panorama**: In the second part of the project, our objective is to construct a panoramic image by stitching the RGB camera images over time based on the body orientation $\mathbf{q}_{1:T}$. If you are not able to obtain good orientation estimates in the first part, you can still do this part using the VICON ground truth and show the results in your report. Use the closest-in-the-past timestamp of your orientation estimates to each camera image time stamp to align them. Do not worry about the panorama image looking perfectly. It is also not necessary to perform color averaging when merging different RGB values, and simply overwriting previous values is acceptable. Fig. 2 shows an example of a partial panorama constructed with noisy orientation estimates.

[4] https://pytorch.org/
[5] https://math.stackexchange.com/questions/54855/gradient-descent-with-constraints

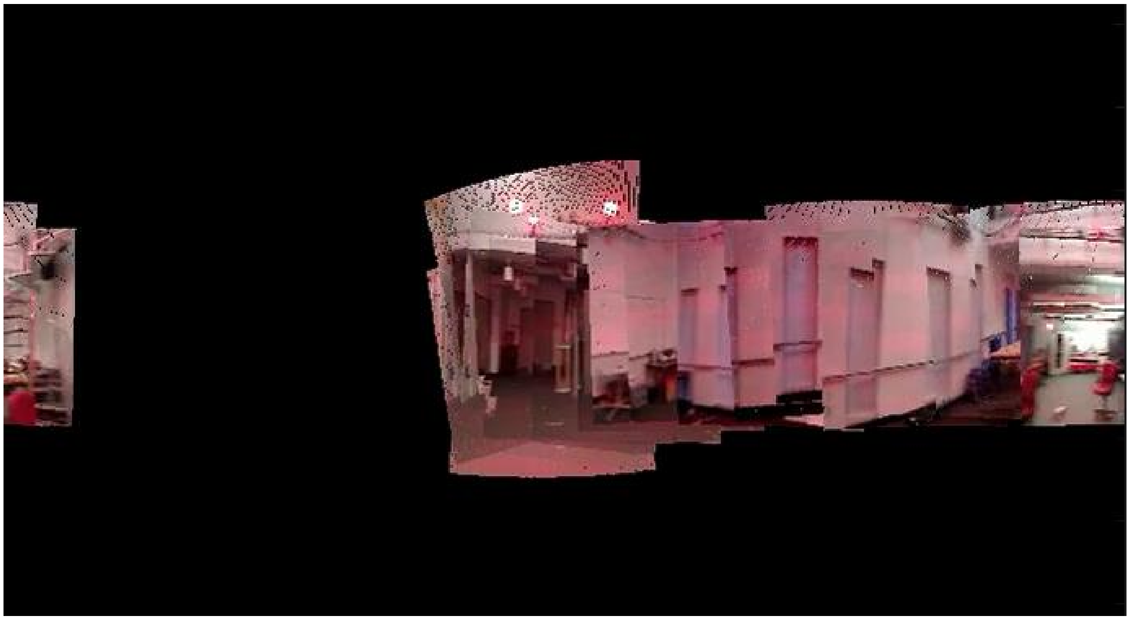

Figure 2: Panorama image stitched from the camera images using the estimated orientation trajectory of the rotating body.

Project Report

Write a project report describing your approach to orientation tracking and panorama reconstruction. Your report should include the following sections.

- [5 pts] **Introduction**: Discuss what the problem is, why it is important, and present a high-level overview of your approach.
- [10 pts] **Problem Formulation**: State the problem you are trying to solve in mathematical terms. This section should be short and clear and should define the quantities you are interested in using precise mathematical terms. You may directly use the definition in equations (1) – (4).
- [20 pts] **Technical Approach**: Describe your approach to orientation tracking and panorama reconstruction.
- [10 pts] **Results**: Present both your training results and test results, and discuss them – what worked, what did not, and why. Make sure your results include (a) plots comparing your estimated roll, pitch, and yaw angles to the ground truth on the training sets, (b) plots showing your estimated roll, pitch, and yaw angles on the test sets, (c) panorama images obtained from the training and test sets. You should include discussion of your results and any other qualitative or quantitative results that will help present your work.