Orientation Tracking for Panorama

Abhishek Peri

I. Introduction

Tracking the orientation of a robot is an integral stage of localization and further extending to the SLAM problem. Further, orientation tracking also plays an important role in panorama generation. In this project, we try to estimate the orientation of the robot/platform using IMU sensor measurements. Later, we use these measurements to stitch the RGB images gathered during the robot motion to generate a panorama.

II. Problem Formulation

The goal of this project is to determine the orientation of a body over time using measurements of IMU angular velocity and linear acceleration. The orientation will be represented by a unit quaternion q_t, and corresponds to the body frame orientation at time t. The sequence of quaternions $q_{1:T}$ represent the orientation of the body across timesteps.

The problem can be formulated as to find $q_{1:T}$ that minimizes the cost function in (3). Here, $f(q_t, \tau_t, \omega_t)$ corresponds to the motion model which is obtained by integrating over ω_t, which is the angular velocity measured by the IMU over a timeperiod of τ_t. The quaternion kinematics motion model is defined as in (1). Given the orientation q_t at timestep t, we can predict the orientation q_{t+1} at timestep $t+1$ using this model.

$$q_{t+1} = f(q_t, \tau_t, \omega_t) = exp(0, \frac{\tau_t \omega_t}{2}), \tag{1}$$

Further, since the body is undergoing a pure rotation, the acceleration of the body in world frame should be approximately $[0, 0, -g]$ across the timesteps. Here, g is the acceleration due to gravity. Hence, the measured acceleration a_t in the IMU frame should agree with gravity acceleration after it is transformed to the IMU frame using the orientation q_t, leading to the observation model in (2)

$$a_t = h(q_t) = q_t^{-1} \circ [0, 0, 0, -g] \circ q_t, \tag{2}$$

The first term in the cost function defined in (3) 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 $q_{t+1}^{-1} \circ f(q_t, \tau_t, \omega_t)$ between the predicted orientation $f(q_t, \tau_t, \omega_t)$ and the estimated orientation q_{t+1}. The error obtains the axis-angle parametrization of the relative rotation error using the quaternion log$(\cdot)$ function and measures the angle of rotation as the norm of the axis-angle vector. We also need to enforce the constraint that the quaternions q_t remain unit norm, i.e, $q_t \in H_*$, during the optimization. Hence, we have a constrained optimization problem:

$$\min_{q_{1:T}} c(q_{1:T}) : ||q_t||_2 = 1, \forall t \in \{1, 2...., T\}, \tag{4}$$

III. Technical Approach

A. Sensor Calibration

The training data provided contains the IMU readings, RGB images and also the VICON motion capture data measured at several timestamps. However, these sensor measurements have bias, scaling and noise assoicated. The bias and scaling factors can be calculated using the VICON motion data as ground-truth. Given the body is at rest at start of the motion, the state of the body $[a_t, w_t]$ ideally has to be $[0, 0, g, 0, 0, 0]$. However, due to bias the values are different and steady, we can find the average the values untill the body is at rest and then normalize the sensor readings using the bias and the associated scale factor defined in the sensor datasheets. After this we now can proceed to the orientation estimation.

B. Oreintation Estimation

In this work, the constrained optimization problem formulated in (4) is solved through the projected gradient descent approach. It is a variation of gradient descent that operates in the feasible region defined by a set of constraints, by projecting the solution back into the feasible region after each iteration. The update rule for the variables involved in the optimization is described in the (5).

$$q_{t+1}^{k+1} = \frac{q_t^k - \alpha \nabla c(q_{1:T})}{||q_t^k - \alpha \nabla c(q_{1:T})||}, \tag{5}$$

Once we have the calibrated readings of the sensor, we can estimate the motion model and the observation model using the (a_t, ω_t) at every timestep. We first initialize the sequence $q_1, q_2, ..., q_T$ using the motion model prediction $f(q_t, \tau_t, \omega_t)$. In this initial step the loss from the first term in the cost function is expected to be 0. From the next iteration, as the gradients pass to the optimization variables, the loss contributed from term 1 of (3) is no longer insignificant. The projected gradient descent is run for 10 iterations with a step size $\alpha = 0.1$ and each iteration runs over all the timestamps for a given dataset. The final optimized orientations are then used for panorama stitching detailed in the further sections.

One could face few issues during the optimization. One of the major issue being the loss goes to 'nan'. This can be attributed to the gradient calculation of the quaternion operator

$$c(q_{1:T}) = \frac{1}{2}\sum_{t=0}^{T-1} ||log(q_{t+1}^{-1}) \circ f(q_t, \tau_t, \omega_t)||_2^2 + \frac{1}{2}\sum_{t=1}^{T} ||a_t - h(q_t)||_2^2, \tag{3}$$

functions. This can be overcome by adding a small epsilon value of 1e-6 to the inputs of quaternion log function. Further, it is important to check initially if the observation model is matching with the IMU acceleration. In the case of mismatch the loss starts at a very large value and doesn't converge.

C. Panorama Generation

Once we have estimated the orientations of the robot, we can use them to stitch the RGB images captured by the camera at each timestep to generate a panorama. This can be done in the following manner:

- We assume the images lie on a spherical plane since the body is only undergoing pure rotation.
- Convert the image pixel coordinates to spherical cordinate system where each pixel is mapped to (λ, ϕ, ρ), where λ is the longitude and ϕ is the latitude. Given the horizontal and vertical FOV's for the camera, create a linear map from pixel cordintes to spherical coordinates
- These spherical coordinates are then converted to Cartesian coordinates which are then further multiplied by the rotation matrix corresponding to the orientation of the body at the given timestamp to convert them into world frame.
- The rotation matrix at a given timestamp is found using the nearby timestamps' orientation of the IMU obtained after optimization of (4)
- The obtained world frame Cartesian coordinates are then converted to the spherical coordinate system. Later, these coordinates are projected onto a cylinder that circumscribes the sphere.
- We then unwrap this cylinder and scale it so that the final image size of the panorama is (1280,960) using a linear transformation.

IV. Results

A. Orientation estimation

We optimized the orientations that were initially obtained from the motion model. We also have the orientation from the VICON data that acts as ground-truth information for verifying the opimized orientations. The estimated quaternions $q_{1:T}$ are then converted to euler angles and plotted against the VICON data's euler angles. We can also compare these two results with the orientations predicted from the motion model $f(q_t, \tau_t, \omega_t)$. These plots can be seen in I,II,III and can be inferred that the optimized orientations are indeed closer to the VICON ground-truth data than the motion model prediction.

For the plots corresponding to dataset 4, the optimization is observed to be realtively poor. This can be due to the heavy noise that causeded abrupt jumps in the IMU readings that are also reflected in the groundtruth VICON orientations. Further, in the plots corresponding to dataset 9, the naive motion model's predictions are way too off from the VICON readings. In the same plot, we can see that the optimized estimates for the orienations are closer to the VICON data. The plots for the test datasets can be seen in IV, where the comparsions are done between the motion model predictions and the optimized estimates of the orientations.

B. Panoramas

The approach presented in the section III-C is followed to generate the panoramas plotted in V and VI. The final image size of the panorama is set to (1280,960). The orientations that were used to generate these panoramas are obtained from the VICON data. However, the panoramas in VI that correspond to the test sequence are generated using the orientations obtained after optimization. Few experiments were conducted to observe the effects of initial mapping from image pixel to the spherical coordinate system. One of the experiments is presented in the plots. The left column of the plots corresponds to a map where the top left pixel is assigned to (30,-22.5) deg longitude and latitude. Whereas, in the right column this pixel corresponds to (0,0) deg longitude and latitude. In the map of the right column, the longitude runs from (0,60) and the latitude runs from (0,45). However, in the map of the left column, the longitude runs from (30,-30) from left to right and the latitude runs from (-22.5,22.5) from top to bottom. The plots in the right column seems to be visually better in comparison for a few sequences like 8,9,10.

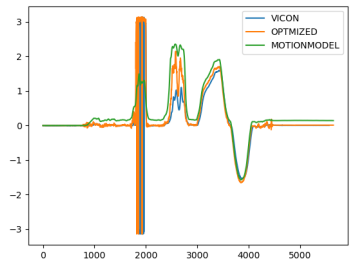
V. Conclusion

In this report, an approach to track the orientation of a body undergoing pure rotation is presented. Further, a panorama is reconstructed by stitching the RGB images taken at approximately regular intervals using the estimated orientations. This report in conclusion gives a basic idea of inferring pose of a robot by fusing information from multiple sensors inorder to reduce their individual noise.

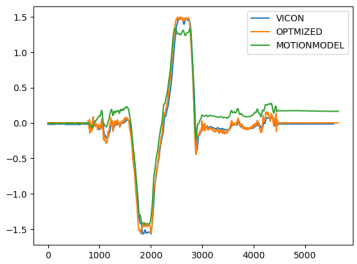
References

References

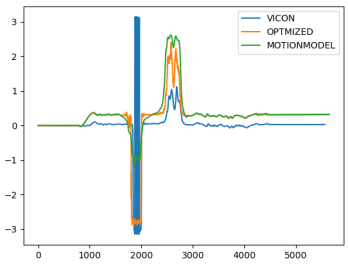
[1] Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., & Zhang, Q. (2018). JAX: composable transformations of Python+NumPy programs (0.3.13) [Computer software]. http://github.com/google/jax



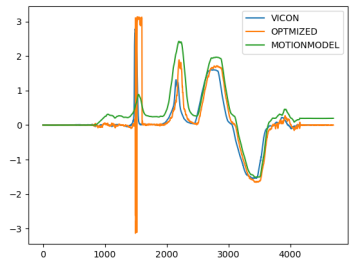
Dataset 1 Roll



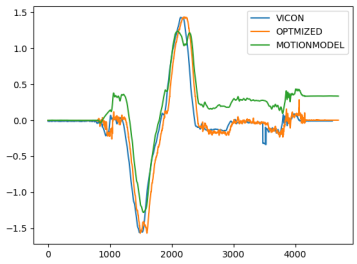
Dataset 1 Pitch



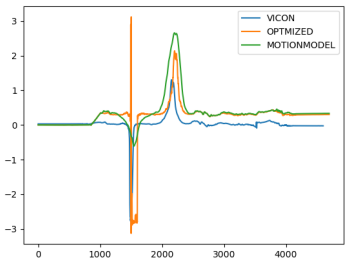
Dataset 1 Yaw



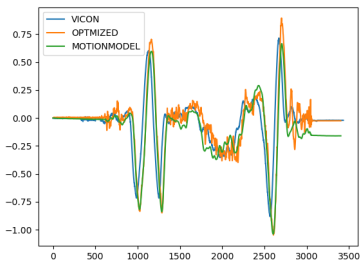
Dataset 2 Roll



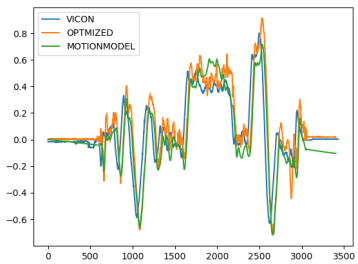
Dataset 2 Pitch



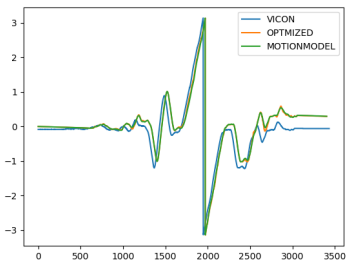
Dataset 2 Yaw



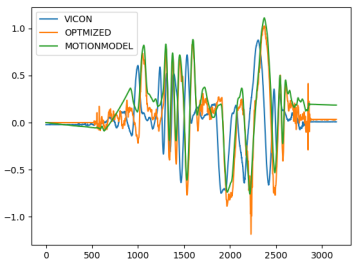
Dataset 3 Roll



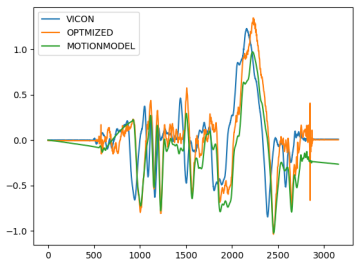
Dataset 3 Pitch



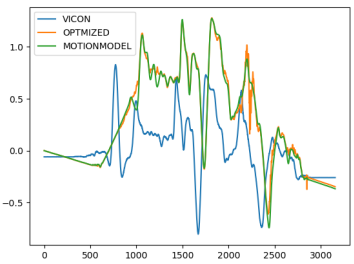
Dataset 3 Yaw



Dataset 4 Roll



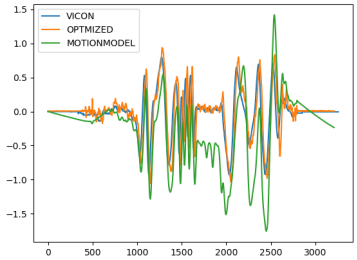
Dataset 4 Pitch



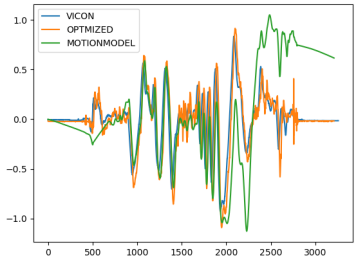
Dataset 4 Yaw

TABLE I

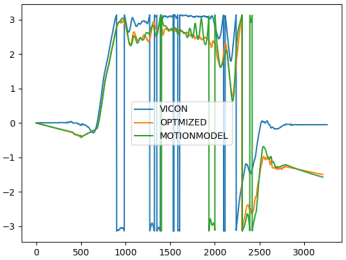
These are the comparision plots for the euler angles that correspond to the orientation of the body over timestamps. This comparison includes VICON groundtruth, motion model and the optimized estimates. It can be seen that the optimization is indeed resulting in better estimates than the naive motion model predictions



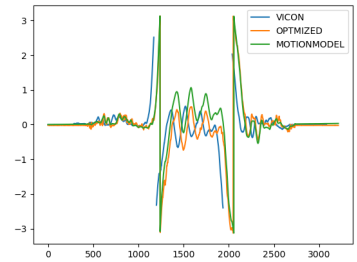
Dataset 5 Roll



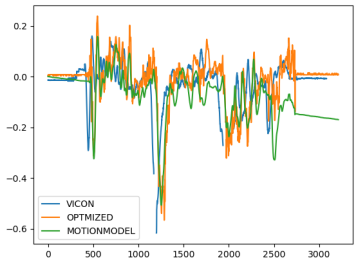
Dataset 5 Pitch



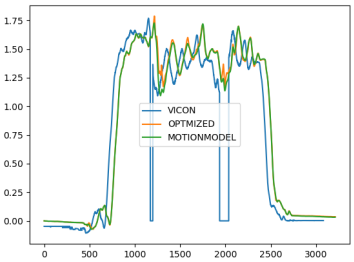
Dataset 5 Yaw



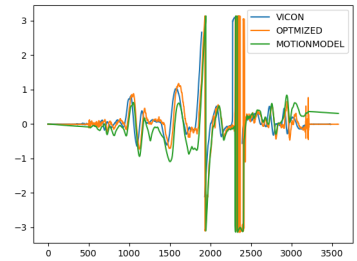
Dataset 6 Roll



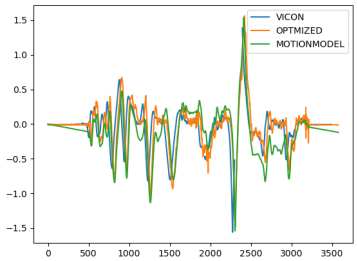
Dataset 6 Pitch



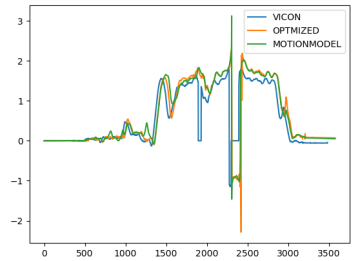
Dataset 6 Yaw



Dataset 7 Roll



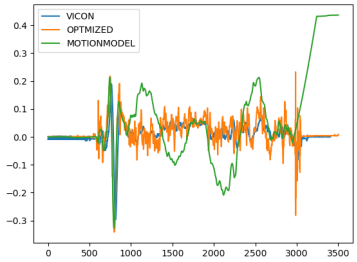
Dataset 7 Pitch



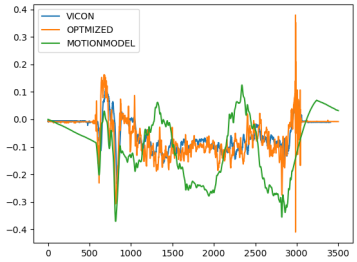
Dataset 7 Yaw

TABLE II

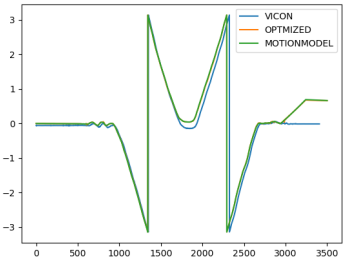
These are the comparison plots for the euler angles that correspond to datasets 5,6,7. This comparison includes VICON groundtruth, motion model and the optimized estimates. It can be seen that the optimization is indeed resulting in better estimates than the naive motion model predictions



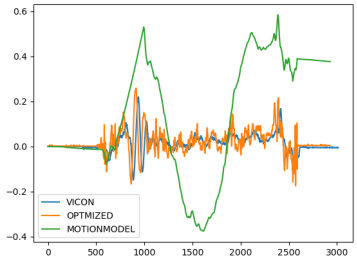
Dataset 8 Roll



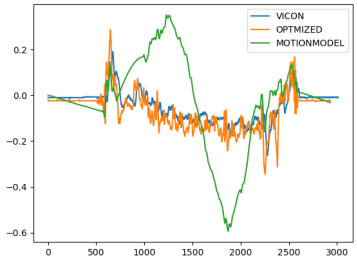
Dataset 8 Pitch



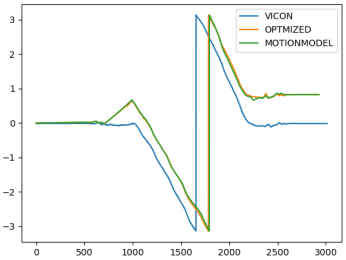
Dataset 8 Yaw



Dataset 9 Roll



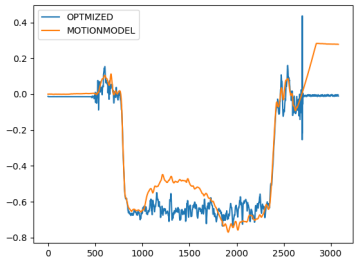
Dataset 9 Pitch



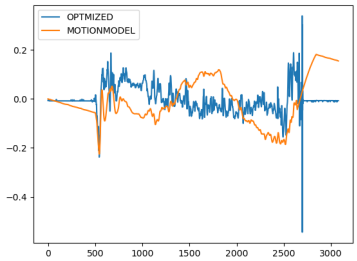
Dataset 9 Yaw

TABLE III

These are the comparision plots for the euler angles that correspond to the datasets 8,9. This comparison includes VICON groundtruth, motion model and the optimized estimates. It can be seen that the optimization is indeed resulting in better estimates than the naive motion model predictions

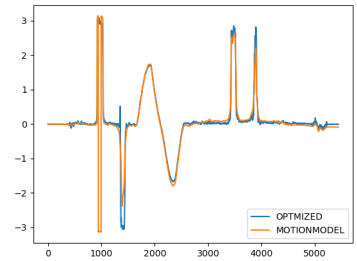


Dataset 10 Roll

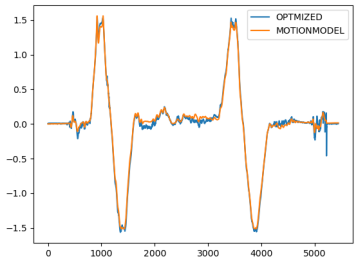


Dataset 10 Pitch

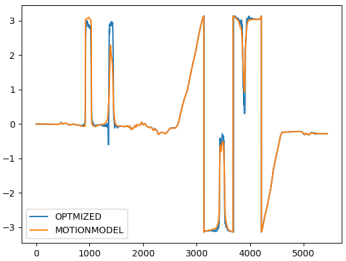
Dataset 10 Yaw



Dataset 11 Roll



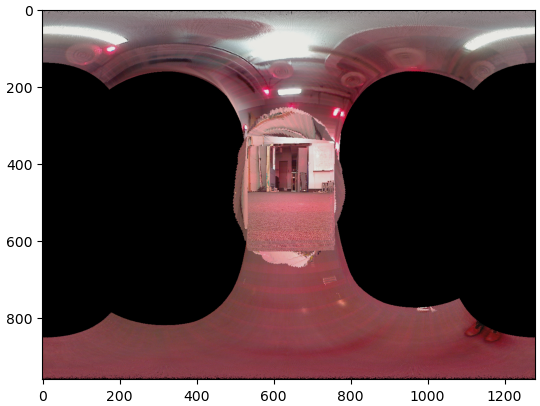
Dataset 11 Pitch



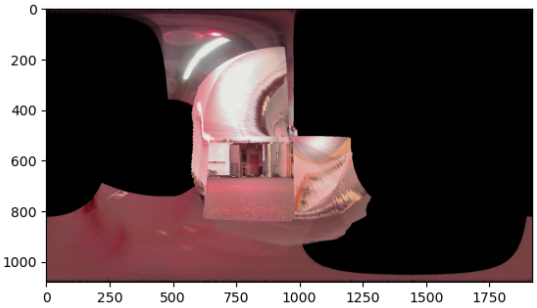
Dataset 11 Yaw

TABLE IV

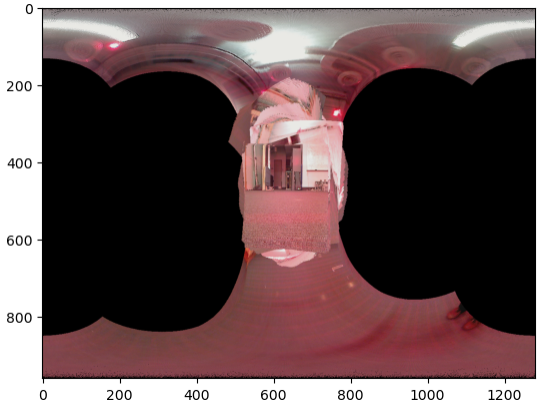
These are the comparision plots for the test dataset's euler angles that correspond to the orientation of the body over timestamps for. This comparison includes motion model and the optimized estimates.



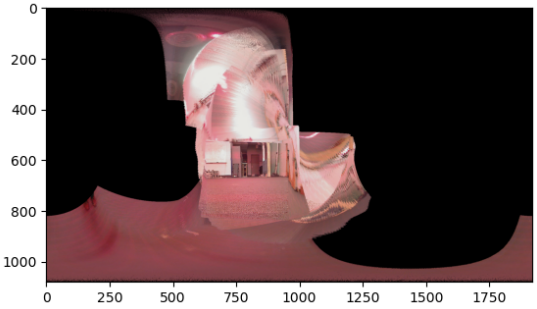
Dataset 1



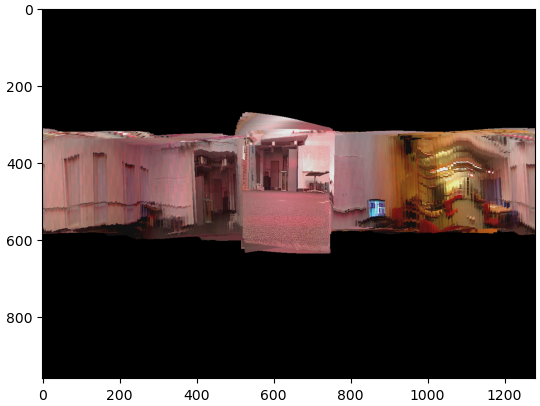
Dataset 1



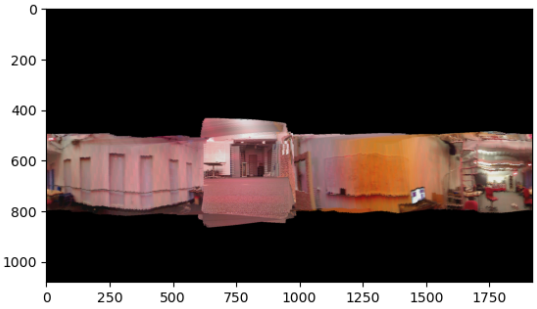
Dataset 2



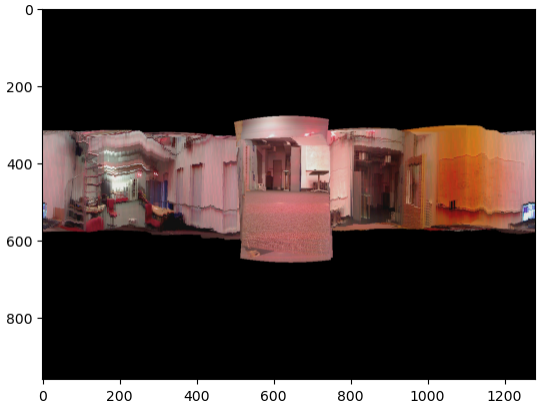
Dataset 2



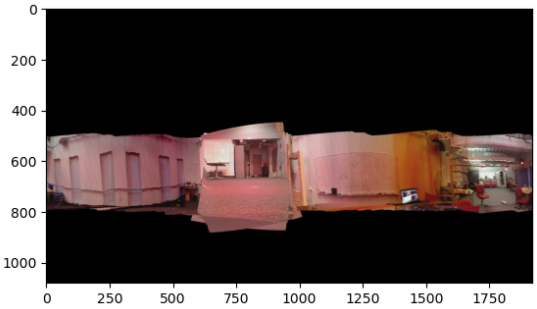
Dataset 8



Dataset 8



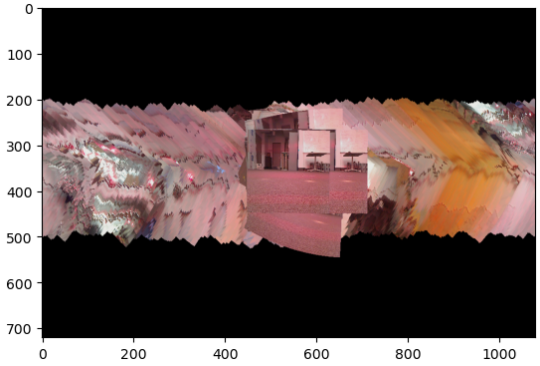
Dataset 9



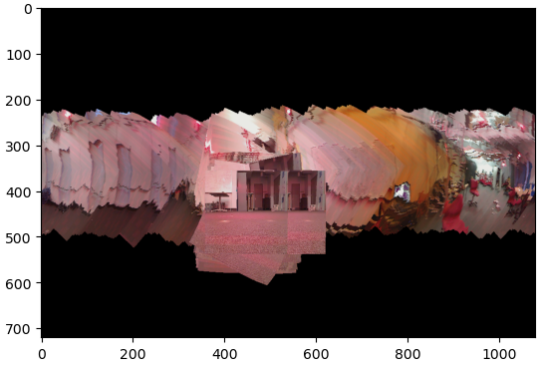
Dataset 9

TABLE V

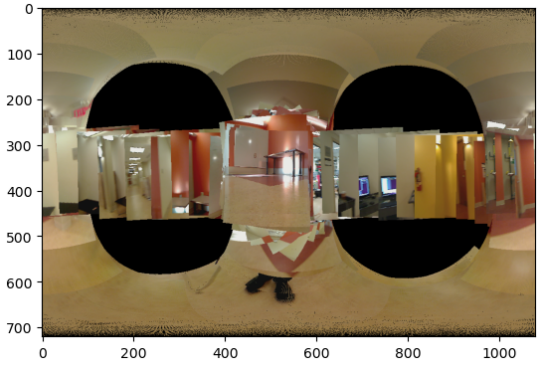
These are the panorama plots on the train sequence corresponding to datasets 1,2,8 and 9. The two columns here correspond to the results obtained using different mapping from image pixel coordinates to spherical coordinates. The left column of the table corresponds to a map where the top left corresponds to (30,-22.5) deg whereas the same image pixel in the right column corresponds to (0,0) deg in spherical coordinates (λ, ϕ, ρ)



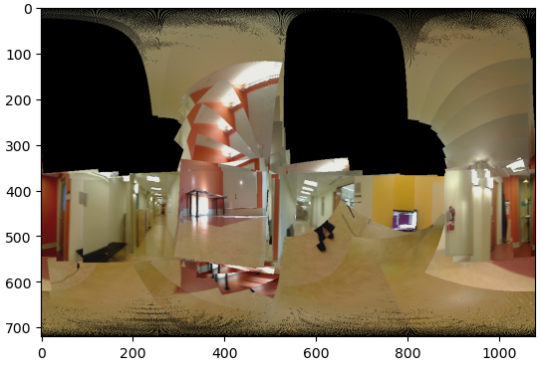
Dataset 10



Dataset 10



Dataset 11



Dataset 11

TABLE VI

These are the panorama plots on the test sequence corresponding to datasets 10 and 11. The two columns here correspond to the results obtained using different mapping from image pixel coordinates to spherical coordinates. The left column of the table corresponds to a map where the top left corresponds to (30,-22.5) deg whereas the same image pixel in the right column corresponds to (0,0) deg in spherical coordinates (λ, ϕ, ρ)