# ECE 276A Project 2: Orientation Tracking

Brian Wilcox, University of California at San Diego, MS, Electrical and Computer Engineering
bpwilcox@ucsd.edu

## I. Introduction

Orientation tracking is a prevalent problem in robotics or any 3-D orientation applications with a rotating body. Identifying the current 3-D state of a body is often a non-trivial task, especially when sensors provide noisy, biased estimates or insufficient information alone. Accurate tracking of the orientation of these bodies may sometimes combine information from multiple sensors providing information. In robotics, this is of particular importance because often a robot (or a camera) needs to know where it is oriented to the world. An interesting use of accurate orientation tracking is in panorama images, where time-dependent camera images are stitched together into one larger picture of the environment. Accurate orientation tracking enables such a feat. In this project, we use an unscented Kalman filter to track the orientation of a rotating camera. A panorama is generated using these filtered 3-D orientation states by stitching together rotated images.

## II. Problem Formulation

### A. Orientation Tracking

Given a set of IMU accelerometer $\mathcal{A} = \{a_1, a_2, ..., a_T\}$ and gyroscope data $\mathcal{G} = \{w_1, w_2, ..., w_T\}$, where $a \in \mathbb{R}^3$ represents 3-D linear acceleration and $w \in \mathbb{R}^3$ represents 3-D angular velocity, the first goal of this project is to estimate the current orientation of the rotating body, $\mathcal{O} = \{r_1, r_2, ..., r_T\}$.

### B. Panorama

Given a set of camera images in time, $\mathcal{I} = \{I_1, I_2, ..., I_T\}$ corresponding with the camera's orientation $\mathcal{O}$, the second goal of this project is to use the orientation of the camera to generate a panoramic image that stitches together the individual images.

## III. Technical Approach

### A. Modifications, Scaling and Unbiasing Data

The data from the IMU needed some modification before use. Firstly, the acceleration in the x and y direction needed to flip signs. Secondly. the angular velocity vectors were out of order, so they were re-ordered into $[\omega_x, \omega_y, \omega_z]$ order.

In order to unnbias the raw IMU data, the mean of the first 250 points (i.e. first couple seconds) was subtracted since there was no movement at the beginning of each dataset.

To scale the raw IMU data, the acceleration and angular velocity values were scaled by a scale factor composed of the sensitivity of the accelerometer and gyroscope and converting back to radians (for angular velocity).

### B. Unscented Kalman Filter - Model

In this project, the Unscented Kalman Filter is used to filter the orientation state, $q$, where $q$ is a quaternion representing the orientation of the rotating body. Because quaternions are used and manipulations of the quaternions are nonlinear, this justifies the use of the Unscented Kalman Filter and must consider special forms of the usual equations. Much of the following UKF method for quaternions is influenced by Kraft.[1]

*1) Process Model:* For the Unscented Kalman Filter (UKF), the process model is

$$q_{k+1} = A(q_k, w_k) = \tilde{q}_k \circ q_\Delta$$

where

$$w_k \sim \mathcal{N}(0, Q)$$

,

$$\tilde{q}_k \ = q_k \circ q_w$$

where $q_w$ is the process noise converted into a quaternion representation.

*2) Measurement Model:* The measurement model uses acceleration (measured in gravity units) as follows

$$z_k = H(q_k, v_k) = \overrightarrow{g}' + v_k$$

where

$$v_k \sim \mathcal{N}(0, R)$$

$$g' = q_k \circ [0, \overrightarrow{g}] \circ q_k^{-1}$$

### C. Unscented Kalman Filter - Sigma Points

In the UKF, we find $2n + 1$ sigma points using

$$\mathcal{W}_{i,i+n} = columns(\pm\sqrt{n(P_{k-1} + Q)})$$

$$\mathcal{X}_i = q_{k-1} \circ q_{\mathcal{W}}$$

where $q_{\mathcal{W}}$ is the quaternion representation of $\mathcal{W}_i$. Now, we have the $\mathcal{X}_i$ which represent the sigma points.

### D. Unscented Kalman Filter - Prediction

Next we have the prediction step. We transform the sigma points through the process model by

$$\mathcal{Y}_i = A(\mathcal{X}_i, 0)$$

Note that to compute the next predicted sigma quaternion, we use the IMU angular velocity data in the process model such that

$$q_{k+1} = q_k \circ exp([0, \frac{1}{2}\omega_t \Delta t])$$

[1] $https://natanaso.github.io/ece276a/ref/2_Kraft_UKF.pdf$

where $\omega_t$ is the angular velocity at time $t$.

These $\mathcal{Y}_i$ are the sigma point predictions, which are then averaged. To find the average of multiple quaternions, a quaternion averaging algorithm (shown in Kraft but omitted for conciseness here), $quatmean$ is used.

$$\hat{q}_k^- = quatmean(\mathcal{Y}, W^{(m)}, \hat{q}_{k-1})$$

where we input all the sigma point predictions, $\mathcal{Y}$, a vector of weights $W^{(m)} = [0, \frac{1}{2n}, ..., \frac{1}{2n}]$, and a starting point being the previous updated quaternion state estimate, $\hat{q}_{k-1}$.

The prediction step for the covariance is

$$P_k^- \sum_{i=0}^{2n} W_i^{(c)} \mathcal{W'}_i \mathcal{W'}_i^T$$

where $W^{(c)}$ is a vector of weights for the covariance, and $\mathcal{W'}_i$ is as follows

$$\mathcal{W'}_i = \overrightarrow{r}_{\mathcal{W'}}$$

where

$$\overrightarrow{r}_{\mathcal{W'}} = q_i \circ \hat{q}_k^-$$

We'll note that $\overrightarrow{r}_{\mathcal{W'}}$ can be found from the last iteration of the quaternion averaging algorithm.

### E. Unscented Kalman Filter - Update

In order to update the quaternion prediction, we must transformed our sigma points with the measurement model as

$$\mathcal{Z}_i = H(\mathcal{X}_i, 0)$$

and then average these transformations

$$z_k^- = \sum_{i=0}^{2n} W_i^{(m)} \mathcal{Z}_i$$

The innovation is calculated as the difference between the actual measured point (from accelerometer) and this mean of the transformed sigma points.

$$\nu_k = z_k - z_k^-$$

We can now compute the uncertainty (covariance) of the set $\mathcal{Z}_i$ as

$$P_{zz} = \sum_{i=0}^{2n} W_i^{(c)} [\mathcal{Z}_i - z_k^-][\mathcal{Z}_i - z_k^-]^T$$

The covariance of the innovations as

$$P_{\nu\nu} = P_{zz} + R$$

where R is the diagonal matrix covariance for the measurement noise. and the cross correlation as

$$P_{xz} = \sum_{i=0}^{2n} W_i^{(c)} [\mathcal{Z}_i - z_k^-][\mathcal{Z}_i - z_k^-]^T$$

We arrive at the Kalman gain which is the product of the cross correlation matrix and the covariance of the innovations.

$$K_k = P_{xz} P_{\nu\nu}^{-1}$$

Finally, we can write the update step for the quaternion state and its covariance matrix.

$$\hat{q_k} = \hat{q}_k^- + K_k \nu_k$$

$$P_k = P_k^- - K_k P_{\nu\nu} K_k^T$$

These steps are performed in a loop for each new data point where the last update $\hat{q_k}$ is fed into the next loop.

### F. Panorama

*1) Pixels to Spherical Coordinates:* Given T camera images with 240x320 pixels of $p_{ij} \in \mathbb{R}^3$, a vertical field of view 45°, a horizontal field of view of 60°, we start by converting each pixel location to spherical coordinates via

$$\phi_{ij} = (\frac{45}{239} * i + 67.5)\frac{\pi}{180}$$

$$\theta_{ij} = (\frac{60}{319} * j - 30)\frac{\pi}{180}$$

*2) Spherical Coordinates to Cartesian:* After we convert all pixel locations to spherical coordinates, we then want to convert these spherical coordinates to cartesian coordinates:

$$x_{ij} = r \sin\phi_{ij} \cos\theta_{ij}$$

$$y_{ij} = r \sin\phi_{ij} \sin\theta_{ij}$$

$$y_{ij} = r \cos\phi_{ij}$$

*3) Rotate to World Frame:* Now that each pixel in a camera frame is converted to cartesian coordinates, we can rotate the camera frame to the world frame using the orientation calculated from our UKF, $q_k$, for each frame at time k. To simplify the rotation, the quaternion calculated from the UKF is converted to a rotation matrix.[2] Thus, given a vector $[x_{ij}, y_{ij}, z_{ij}]^T$ for a given pixel, the world frame vector is:

$$C_{ij} = R_t [x_{ij}, y_{ij}, z_{ij}]^T$$

where $R_t$ is the rotation matrix at a time $t$ calculated as the closest rotation matrix in time to the timestamp of the camera image.

*4) Convert to Spherical Coordinates:* We may convert the world frame pixel location from cartesian coordinates back to spherical coordinates by

$$\theta_{ij} = \arctan\frac{y_{ij}}{x_{ij}}$$

$$\phi_{ij} = \arctan\frac{\sqrt{x_{ij}^2 + y_{ij}^2}}{z_{ij}}$$

*5) Inscribe to Cylinder and Unravel:* Now that the rotated pixels are back in spherical coordinates, we inscribe the pixels into a cylinder via

$$\theta_{ij} = \theta_{ij}$$

$$z_{ij} = \cos \phi_{ij}$$

We then unravel the cylinder into indices for the new panorama image of size (M,N):

$$z_{ij}^{\nu} = round((-z_{ij} + 1)\frac{M}{2})$$

$$\theta_{ij}^{\nu} = round((\theta_{ij} + \pi)\frac{N}{2\pi})$$

We can then use these indices to stitch pixels from each camera image into the respective via

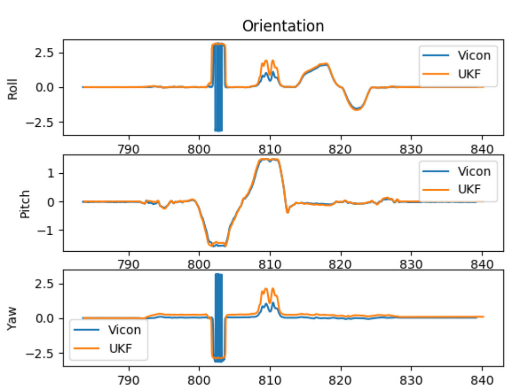
$$(Panorama)_{z_{ij}^{\nu}, \theta_{ij}^{\nu}} = (CamImage)_{ij}$$

This procedure is done for all pixels of all camera images in order to generate the full panorama that stitches together the N camera images.

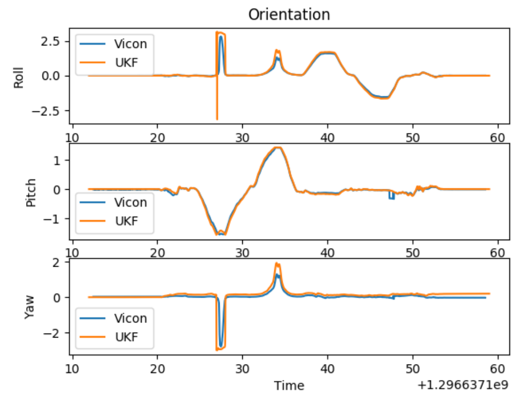
## IV. Results

### A. UKF Results - Training

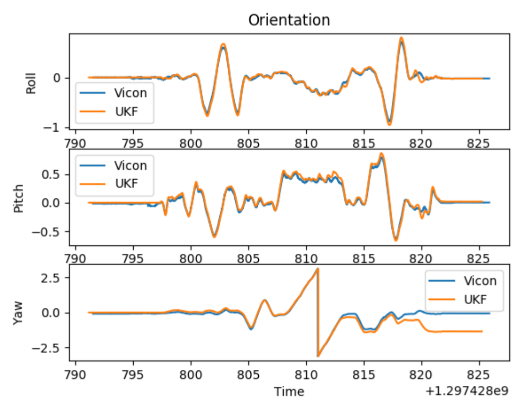
The orientation computed from the UKF was converted to euler angles and compared against the orientations as measured by the Vicon data (considered our ”ground truth”). Here, the choice of process noise and measurement noise were $Q = 0.0001I$ and $R = 0.1I$ with initial covariance as $P_0 = 0.0001I$ and initial state $q_0 = [1, 0, 0, 0]$.



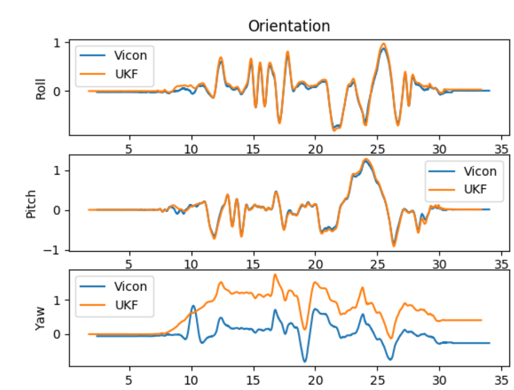

Dateset 1 Orientation Comparison UKF vs Vicon



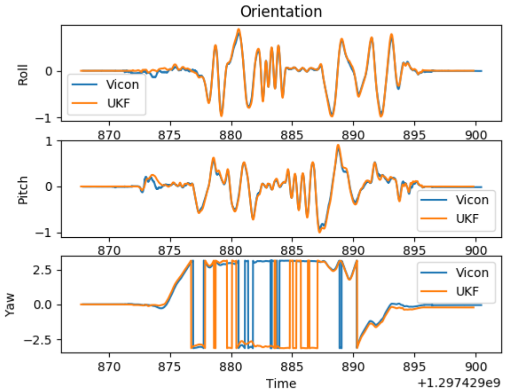

Dateset 2 Orientation Comparison UKF vs Vicon



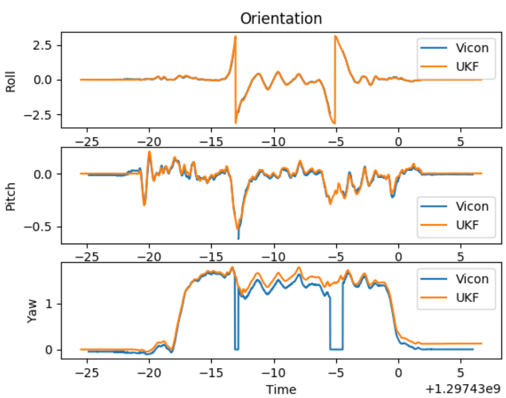

Dateset 3 Orientation Comparison UKF vs Vicon



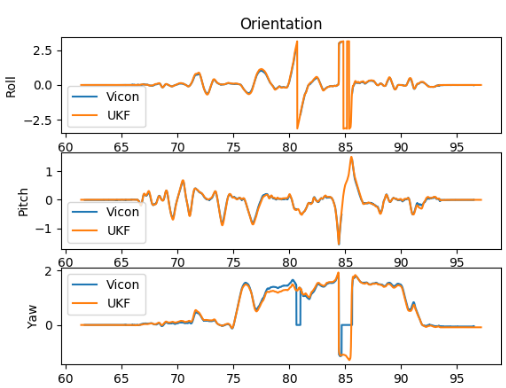

Dataset 4 Orientation Comparison UKF vs Vicon



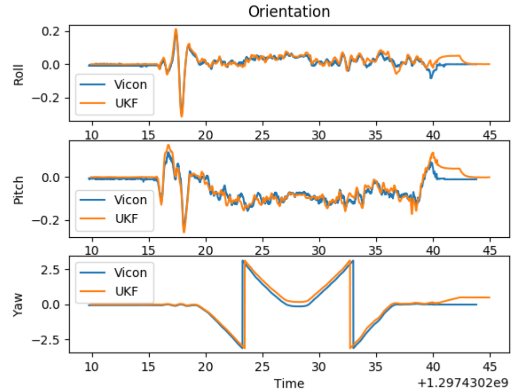

Dateset 5 Orientation Comparison UKF vs Vicon



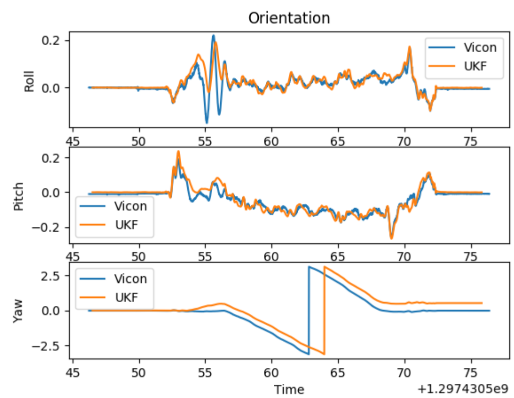

Dateset 6 Orientation Comparison UKF vs Vicon




Dateset 7 Orientation Comparison UKF vs Vicon




Dataset 8 Orientation Comparison UKF vs Vicon

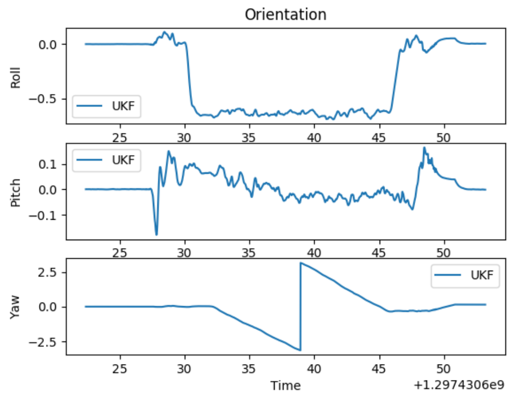



Dataset 9 Orientation Comparison UKF vs Vicon

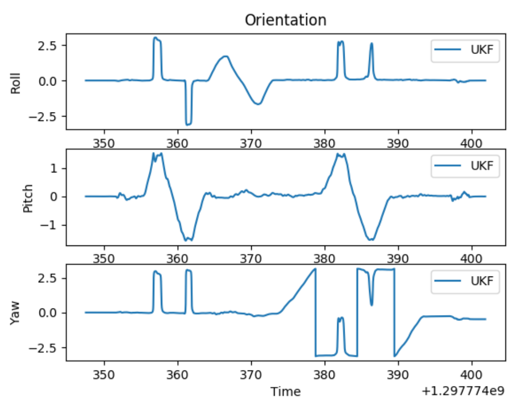
The UKF results for orientation tracking appeared to be comparable to the vicon data for most datasets.The choice of a larger measurement noise apperaed to be a good choice when comparing to vicon data. In some datasets (e.g. 4,6,9), there appears to be slight drift or discrepancy.

### *B. UKF Results - Test*

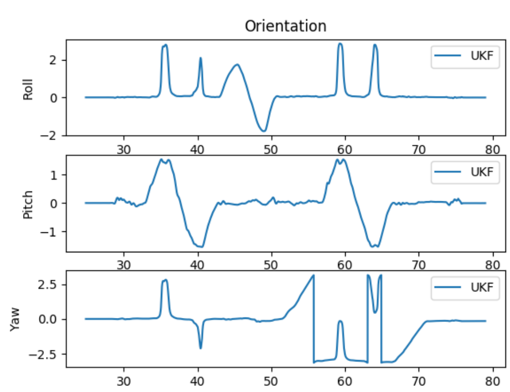
The UKF was applied to 4 test sets of data and is plotted below where the quaternions were converted to euler angles with roll, pitch, and yaw. The same initial parameters as in the training set were used.



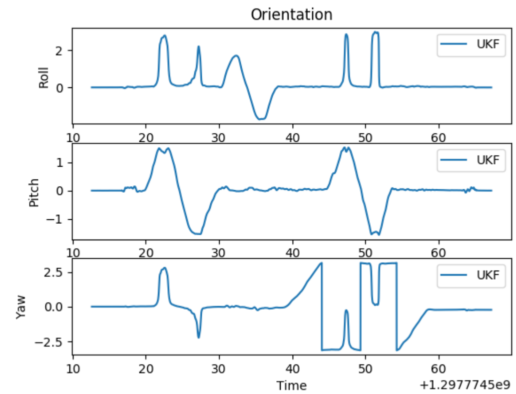

Dateset 10 Euler Angles Orientation from UKF




Dateset 11 Euler Angles Orientation from UKF



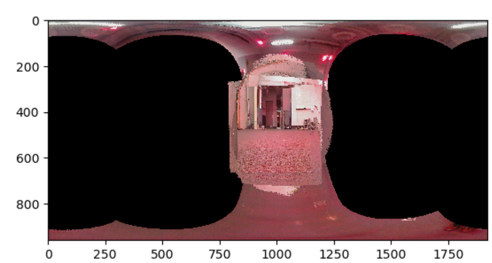

Dateset 12 Euler Angles Orientation from UKF



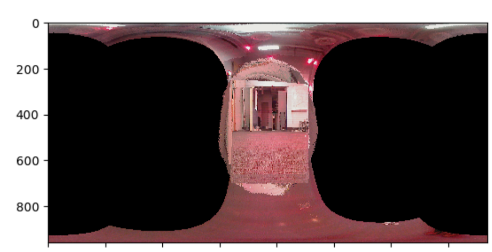

Dataset 13 Euler Angles Orientation from UKF

### C. Panorama Results - Training

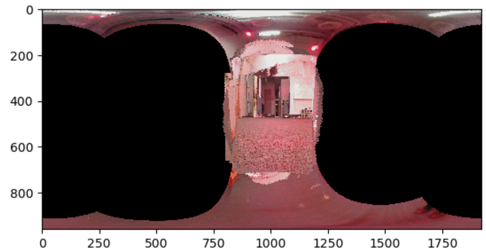
The results of the panorama generation procedure are shown below with each training set panorama given by the UKF orientation tracking followed by the panorama generated by the Vicon ”ground truth” orientations.



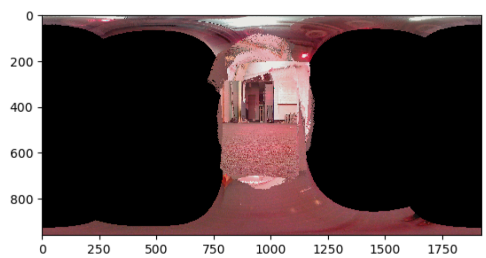
Dataset 1 Panorama from UKF orientation tracking



Dataset 1 Panorama from Vicon ground truth

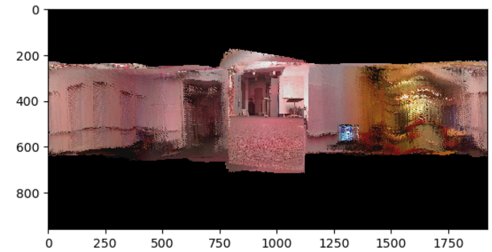



Dataset 2 Panorama from UKF orientation tracking

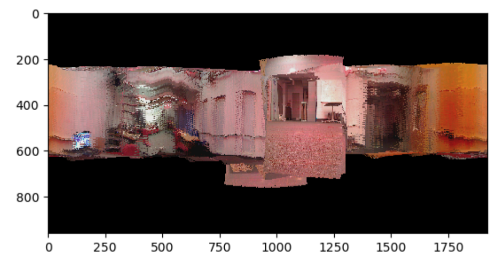



Dataset 2 Panorama from Vicon ground truth

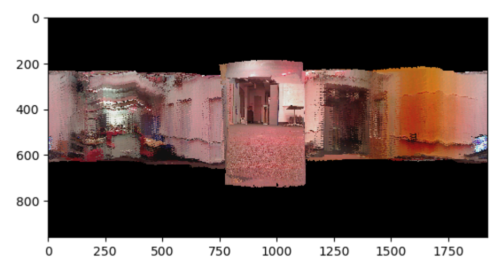

Dataset 8 Panorama from UKF orientation tracking




Dataset 8 Panorama from Vicon ground truth




Dataset 9 Panorama from UKF orientation tracking

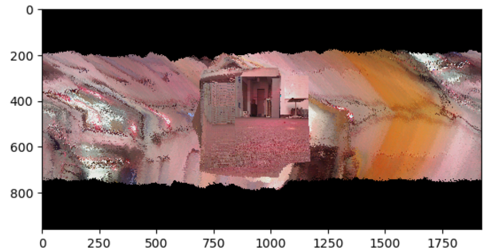



Dataset 9 Panorama from Vicon ground truth

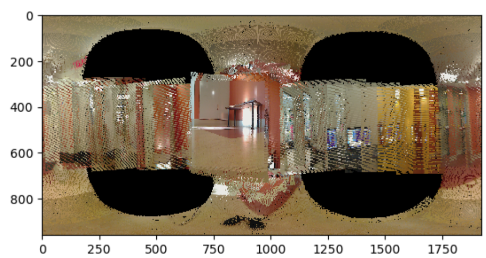
The panorama using the UKF orientation appears similar to the panorama using the Vicon data for the 4 training sets, indicating decent orientation tracking using the IMU data.

### *D. Panorama Results - Test*

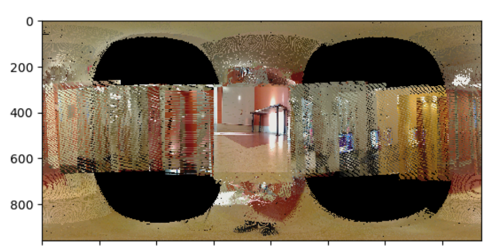
The results of the panorama generation procedure are shown below for the test set using UKF orientation tracking.



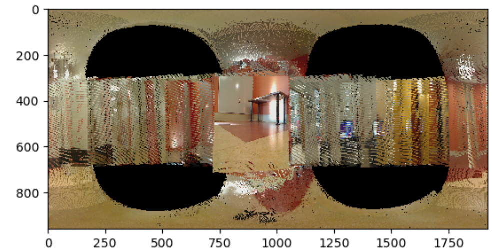

Dataset 10 Panorama from UKF orientation tracking




Dataset 11 Panorama from UKF orientation tracking




Dataset 12 Panorama from UKF orientation tracking




Dataset 13 Panorama from UKF orientation tracking

The results for the 4 test datasets appear somewhat continuous and distinguishable, which indicates that the UKF orientation tracking and panorama algorithm performed decently well. Improvements may seek more sensors for sensor fusion in tracking the orientation