Optimized Orientation Tracking using Riemann Optimization

Saksham Jindal
Department of Electrical and Computer Engineering
University of California, San Diego

I. Introduction

Accurate tracking of orientation is essential for many applications, such as robotics and virtual reality. In robotics, it is important for accurately determining the end effector orientation in manipulation tasks and object interaction in navigation tasks. However, orientation estimated using inertial sensors called Inertial Measurement Unit (IMU) data in vulnerable to various errors, including sensor noise, biases, and other environmental factors that lead to inaccurate estimations. We combine the accelerometer and gyroscope readings from the IMU to estimate the 3D orientation of a rotating body using discrete-time quaternion kinematics. Further, the 4D orientations are optimized using stochastic gradient descent constrained on a unit radius hypersphere Riemann manifold. This algorithm adjusts the quaternions based on a constrained optimization function involving angular velocity and acceleration from the inertial measurements resulting in a more precise estimate of the orientation. The approach has potential to improve the performance of robotic and virtual reality systems allowing them to perform the respective tasks at higher precision. One such application is demonstrated in the paper where we use optimized orientation estimates and images captured from a camera attached to the rotating body and stich a 360º panorama image.

Keywords—orientation tracking, projected gradient descent, Riemann optimization, accelerometer, gyroscope, robotics

II. Promblem Formulation

We formulate the orientation tracking problem using multi-objective constraint optimization problem. We have the angular velocity $\omega_t \in \mathbb{R}^3$, angular acceleration $a_t \in \mathbb{R}^3$ measurements in the IMU frame, quaternions $q_t \in \mathbb{H}_*$ to denote body-frame orientations at time t. The tracking problem can be broken down into:

a) **Quaternion kinematics motion model:** Assuming a unit quaternion at initial time step, the goal is to predict next time-step quaternion q_{t+1} on the current time step given the current time stamp q_t, angular velocity ω_t from the IMU and difference in time stamps τ_t. We denote the motion model as $f(q_t, \tau_t\omega_t)$

b) **Acceleration observation model:** Since the body is rotating with an angular velocity, acceleration measured in the world frame of reference ($[0, 0, -g]$) differs from the acceleration a_t. The goal is to calculate the oriented or transformed acceleration in the body frame. We call it observation model and denote it by $h(q_t)$

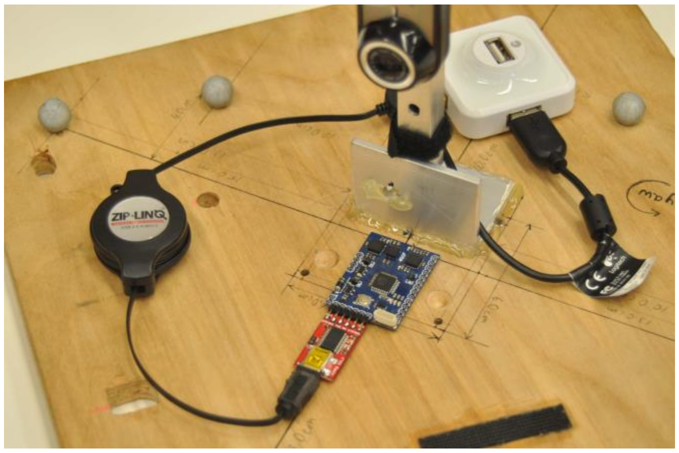


Figure 1 Data collection setup involving inertial and camera sensor

c) **Constrained optimization of orientation:** Since we have measurements from the accelerometer (a_t) and gyroscope (ω_t), we want to simultaneous optimize the motion model $f(q_t, \tau_t\omega_t)$ and observation model $h(q_t)$. We formulate the optimization function $c(q_{1:T})$ to estimate and optimize the orientation trajectory $q_{1:T}$ using projected gradient descent on a unit radius hypersphere manifold for Riemann optimalization.

Finally, we demonstrate the applications of an accurate orientation tracking by constructing a **panoramic** image from RGB camera image over time based on the body orientations $q_{1:T}$

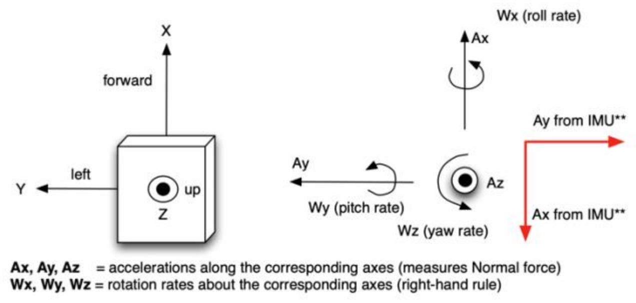
III. Technical Details

In the following section, we dive deep into the models discussed in the previous section and discuss the approach to orientation tracking and panorama generation.

A. Calibration of IMU data

The IMU gives raw A/D values and cannot be physically interpreted before scaling. Further, we need to remove the bias from the raw measurements and scale the corresponding value to convert to physical units.

value = (raw-bias) scale_factor


X
forward
Y
left
up
Z
Wx (roll rate)
Ax
Ay
Wy (pitch rate)
Az
Wz (yaw rate)
Ay from IMU**
Ax from IMU**
Ax, Ay, Az = accelerations along the corresponding axes (measures Normal force)
Wx, Wy, Wz = rotation rates about the corresponding axes (right-hand rule)

Accelerometer : We calculate the bias on first 500 values for the accelometer. For the scale factor we use the following equation

$$\text{scale\_factor} = \frac{Vref}{1023\ sensitivity}$$

We use a reference voltage of 3300 mV and accelerometer sensitivity of 300 mV/g. Further we do corrections to the reported accelerometer reading since IMU $A_x$and A_y direction is flipped (due to device design), so respective positive acceleration in body frame will result in negative acceleration reported by the IMU

Gryoscope : We calculate the bias on first 500 values for the gryscope. For the scale factor we use the following equation

$$\text{scale\_factor} = \frac{\pi Vref}{1023\ sensitivity\ 180}$$

We use a reference voltage of 3300 mV and gryoscope sensitivity of 3.33 mV. Further we do corrections to the reported accelerometer reading since IMU $A_x$and A_y direction is flipped (due to device design), so respective positive acceleration in body frame will result in negative acceleration reported by the IMU.

B. *Quartenion Kinematics Motion Model*

Quaternion motion model is used to estimate the orientation of body over time (expressed in unit quaternion $q_t \in \mathbb{H}_*$) using the IMU angular velocity $\omega_t \in \mathbb{R}^3$ (in body-frame coordinates) We use quaternions, specifically, for building the motion model and expressing orientations owing to gimbal lock leading to singularities in 3D rotation matrix. In continuous time, the orientation of the body can be updated using the following equation:

$$\dot{q}(t) = q(t) \circ \left[0, \frac{\omega(t)}{2}\right] \qquad (1)$$

where $\circ$ is the quaternion product and $\left[0, \frac{\omega(t)}{2}\right]$ is the quaternion representation of angular velocity. In discrete time, the orientation of the body at the next step can be predicted as:

$$q_{t+1} = f(q_t, \tau_t\omega_t) \coloneqq q_t \circ exp\left(\left[0, \frac{\tau_t\omega_t}{2}\right]\right) \qquad (2)$$

where τ_t is the time difference between consecutive time steps and $exp(.)$ is the exponential function. Here, we assume that ω_t is constant angular velocity in $[t, t+1)$.

To track the orientations of the rigid body (without optimization), we initialize the initial time step quaternion q_0 with an identity quaternion. We run discrete time quaternion kinematic to estimate quaternion sequence over time $q_{1:T}$ using equation angular velocities and time stamps from the IMU in equation 2. We further convert this quaternion sequence into a sequence of roll, pitch and yaw angles over time and plot them against VICON data used as the ground truth.

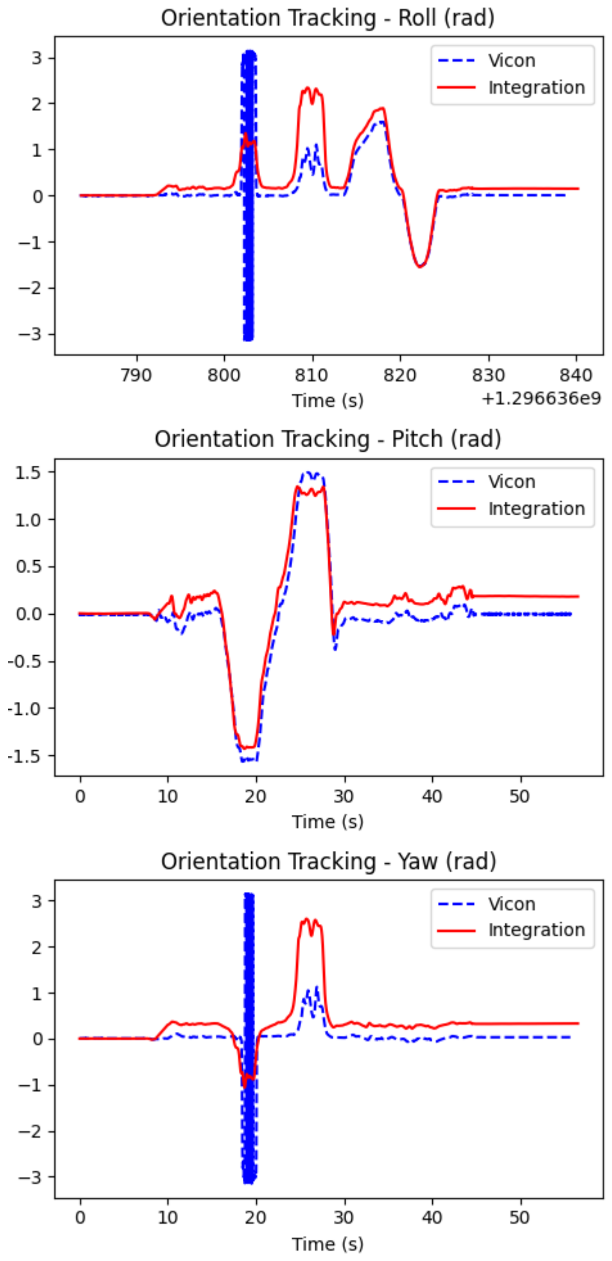


Figure 3: Tracking results with Euler integration and comparison with Vicon data on dataset "1"

C. *Observation Model*

Acceleration observation model is used to estimate the body frame acceleration $z_t \in \mathbb{R}^3$ of the rotating body. Since the body is undergoing rotation, the acceleration of the body in the world frame $[0, 0, -g]$ would differ from the acceleration measured in the body-frame coordinate system. The hypothesis in building the observation model is that acceleration measured in the body frame $a_t \in \mathbb{R}^3$ should agree with the gravity acceleration (acceleration in world coordinate frame) after it is transformed to the IMU frame using orientation q_t .

$$z_t = h(q_t) = {q_t}^{-1} \circ [0, -ge_3] \circ q_t \tag{3}$$

D. *Constrained optimisation of orientation*

We formulate the optimization problem to estimate the quaternion trajectory $q_{1:T}$ based on the motion model (2) and the observation model (3). The cost function of the problem is defined as

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

The cost function used in the optimization problem consists of two terms:

a) Error from Motion Model: It measures the error between the estimated orientation and the motion model prediction and is based on the relative rotation between the predicted orientation and the estimated orientation. The term involving the error from motion model is involves using logarithmic map from quartenion difference in H_* to recover a Hthis rotation vector.

b) Error from Observation Model: It measures the sum of the norm of the residual error between the measured acceleration from accelerometer a_t and observation model $h(q_t)$ in $\mathbb{R}^3$ across the time sequence $1{:}T$

Since it involves simultaneous minimization of two error terms, there is a trade-off in the cost function that arises from the balance between the two terms. While initializing the parameters of the cost function, the error from the motion model is zero. However, using quaternions initialized from the motion model may reflect poorly on the quality of the estimates of the observation model leading to a large error in the observation model prediction. Minimizing the error from just model might lead to a large error in the other model. The optimization problem tries to find a balance between the two terms in the cost function so that both the motion model and observation model predictions are accurate.

To resolve the trade-off between the motion model and observation model predictions can be adjusted based on how much we “trust” the measurements from the accelerometer or gyroscope. Although not experimented here, we could assign a different weight to the cost terms interpreted as a measure of how much we trust the gyroscope vs accelerometer measurements (e.g., which ones are noisier).

We have the constrained optimization problem

$$\min_{q_{1:T}} c(q_{1:T})$$
$$s.t. \left|\left|q_t\right|\right|_2 = 1$$

Since the rotation quaternions that represent rotation in 3D space have a unit norm, the optimization problem is constrained as the quaternions q_t must remain unit norm throughout the optimization. The unit norm constraint corresponds to the rotation quaternion q_t lying on a unit sphere in $\mathbb{H}_*$

Since the rotation quaternions lie on a unit radius hypersphere, it can be a considered as a type of Riemann manifold. We use *Riemannian Stochastic Gradient Descent (RSGD)* [1] for the constrained optimization. Considering an SGD update of the form in the Euclidean space

$$q_t^{i+1} \leftarrow q_t^i - \alpha \nabla_{q_{1:T}} c$$

where α is the leaning rate and $\nabla_{q_{1:T}} c$ is the gradient of the objective function $c(q_{1:T})$. On the Riemann manifold $(\mathcal{M}, \rho)$, the RSGD re-defines the update

$$q_t^{i+1} \leftarrow exp(-\alpha h(q_{1:T}))$$

where $h(q_t) \in \mathcal{T}_{q_t}\mathcal{M}$ denotes the orthogonal projection of gradient of $c(q_{1:T})$ at $q_{1:T}$ on the tangent space $\mathcal{T}_{q_t}\mathcal{M}$. Further, we use a first order approximation of exponential map $exp(.)$ using a retraction function $\mathcal{R}: \mathcal{T}_{q_t}\mathcal{M} \rightarrow \mathcal{M}$ that maps $h(q_t)$ on the tangent space to manifold $\mathcal{M}$.

Let $n \in \mathcal{T}_{q_t}\mathcal{M}$ denote the unit vector along $h(q_t)$ on the tangent space. We update the quaternions q_t^{i+1} using the following formulation

$$q_t^{i+1} = q_t^i \; cos\,(||h(q_t)||) + n \; sin\,(||h(q_t)||)$$

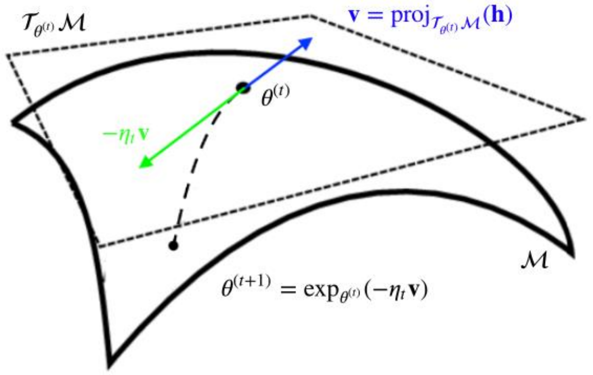

$\mathcal{T}_{\theta^{(t)}}\mathcal{M}$
$\mathbf{v} = \mathrm{proj}_{\mathcal{T}_{\theta^{(t)}}\mathcal{M}}(\mathbf{h})$
$\theta^{(t)}$
$-\eta_t \mathbf{v}$
$\mathcal{M}$
$\theta^{(t+1)} = \exp_{\theta^{(t)}}(-\eta_t \mathbf{v})$

Figure 4: Gradient step via Exponential Map [2]

E. *Stitching panaroma*

We generate the panorama by projecting the cartesian coordinates of the camera images to a unit radius sphere and reproject the spherical coordinates to a cylinder of unit radius. A summary of the steps involved in generating the panorama is presented here:

a) For the spherical coordinates, we use the geographical coordinate system (GCS) [3] using which where we extract the latitude (θ) and longitudes (ϕ) of the images points centered at a circle of a unit radius (r) in camera coordinate frame. Using a horizontal and vertical field of view of 60 degree and 45 respectively, we adopt a convention where θ and ϕ lies in the range $\left[-\frac{\pi}{6}, \frac{\pi}{6}\right]$ and $\left[-\frac{\pi}{8}, \frac{\pi}{8}\right]$.

b) We convert the spherical coordinate to cartesian coordinates the camera frame using the following conversion approach [4]

$$x = r\cos\theta\cos\phi$$
$$y = -r\cos\theta\sin\phi$$
$$x = -r\sin\theta$$

c) For each of the camera image at time t, we find the orientation value (q_t) at nearest time stamp and extract rotation matrix ($R(q_t)$) for transformation from the camera coordiante frame (P_C) to world coordinate frame P_W

$$P_W = R_C P_C + C_W$$

where C_W is the translation [0, 0, 0.1] in the world coordinate frame.

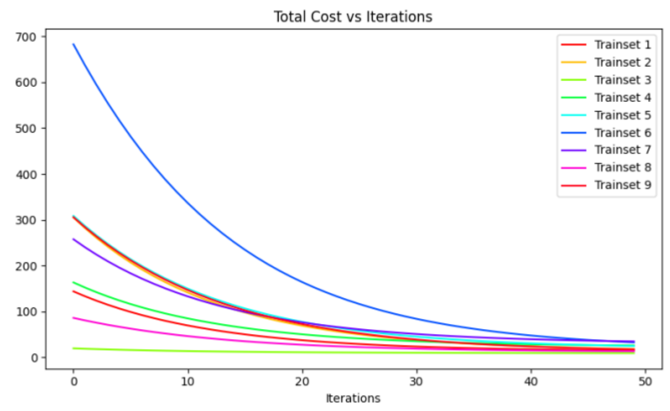
d) Further, we convert the cartesian coordinates (x, y, z) in the world frame to spherical coordinates (θ, ϕ, r) and re-project them to a sphere of unit radius. We inscribe the sphere in a cylinder so that a point $(\theta, \phi, 1)$ on the sphere has a height θ on the cylinder and longitude ϕ on the sphere.

$$\theta = \arcsin\left(\frac{-z}{r}\right)$$
$$\phi = \arctan\left(\frac{-y}{x}\right)$$
$$r = \sqrt{x^2 + y^2}$$

e) Finally, we unwrap the cylinder to surface to a rectangular image with width 2π radians and height π radians.

IV. Results

We present the orientation tracking and panorama generation results in the appendix. The results on the training set are available in Appendix A. to Appendix I. and results from the test set are available in the Appendix J. to Appendix K. For running orientation tracking, we initialize quaternion at time step t_1 with [1,0,0,0] and introduce slight perturbation of 0.0001 while calculating exponential and logarithmic maps of the quaternions. We experimented with perturbation of different order values and found that we count introduce perturbations only as less as 0.000001 as introducing perturbation of lesser order, would blow up our cost and gradient values. Further, we observed that it would be a better strategy to perform computations with double-precision (*float-64*) against the default single-precision (*float-32*) used in modern day solvers.



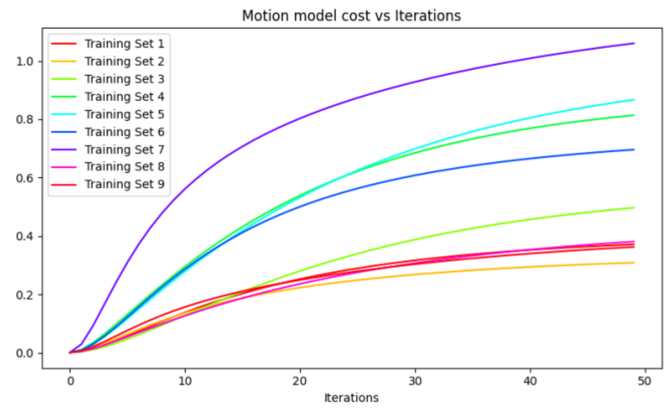


Figure 5: Total cost (above) and motion model (cost) against optimization iteration steps

During the calibration step, we used first 500 values of each of the dataset. While performing Euler integration in the motion model, we found that motion model is sensitive to abrupt changes in roll, pitch or yaw values (Appendix C and D). We used a step size or learning rate of 0.01 while performing gradient steps. For each of the datasets we observed convergence in total cost values with increase in motion model cost (as expected) as we increase the gradient steps. We observe high initial cost and high motion model cost (at near convergence of total cost) for dataset “5”, “dataset “6” and dataset “9” for which we infer that there is high noise both in accelerometer and gyroscope. This can be confirmed from the plots in Appendix E, F and I. Optimization of orientations does not help since observation cost remains high for such datasets at near convergence. For datasets “1”, “2”, “8” and “9”, we observed that the motion model cost diverges relatively less while showing convergence in optimization model. Finally, we observe changes in the panorama plots before and after convergence. For instance, in dataset “8”, we can approximate the rigid body undergoing rotation around “z” axis due to relatively less values for roll and pitch values. As expected, it should a horizontal panorama accounting for images captured while rotating in the x-y plane.

V. Future Work

As a part of future work, one could experiment with relative weight values assigned to motion model and observation model optimization. Further, it remains to be seen whether calibrating with higher number of sampled in noisier dataset influences results of Euler integration. One could also experiment with adaptive optimization strategies [5] and observe if there is a significant effect on final convergence

References

[1] S. Bonnabel, “Stochastic gradient descent on Riemannian manifolds” *IEEE Transactions on Automatic Control*, vol. 58, no. 9, pp. 2217–2229, 2013

[2] A. Bloch, “Stochastic Gradient Descent on Riemannian Manifolds”, 2019 https://andbloch.github.io/Stochastic-Gradient-Descent-on-Riemannian-Manifolds/

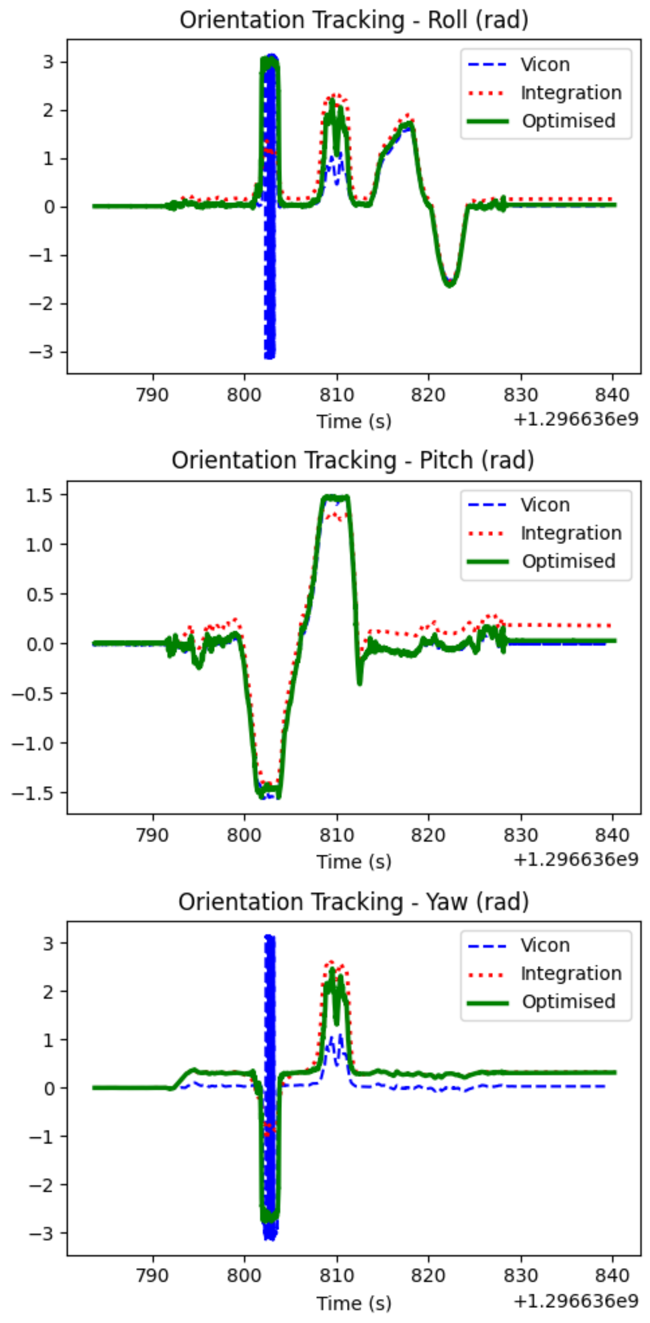
[3] Wikipedia. “Geographic Coordinate System”. 2023. https://en.wikipedia.org/wiki/Geographic_coordinate_system

[4] Wikipedia. “Geographic coordinate conversion”. 2023. https://en.wikipedia.org/wiki/Geographic_coordinate_conversion

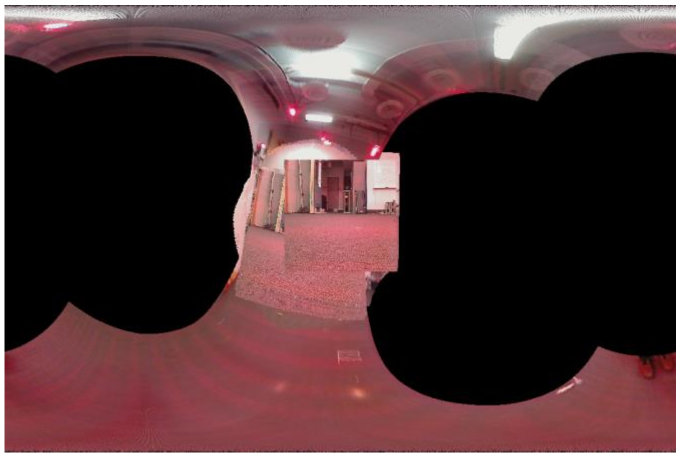
[5] G. Bécigneul, O. Ganea, “Riemannian Adaptive Optimization Methods” International Conference on Learning Representations (ICLR), 2019

Appendix

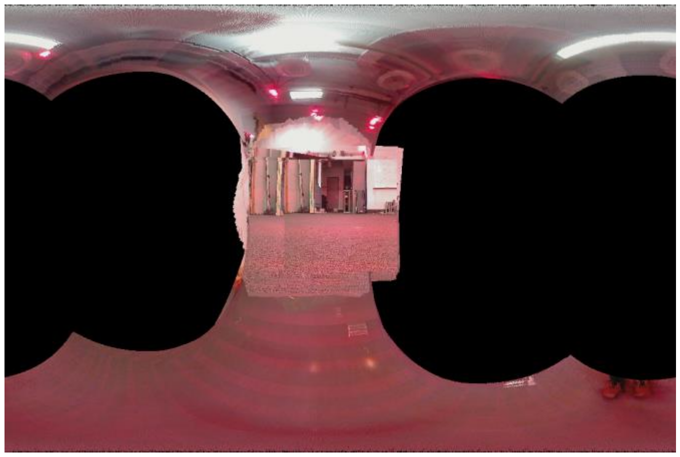
A. *Results on Dataset: Trainset "1"*



Orientation Euler angles (Roll, Pitch and Yaw) from vicon, motion model (integration) and optimization of orientaion

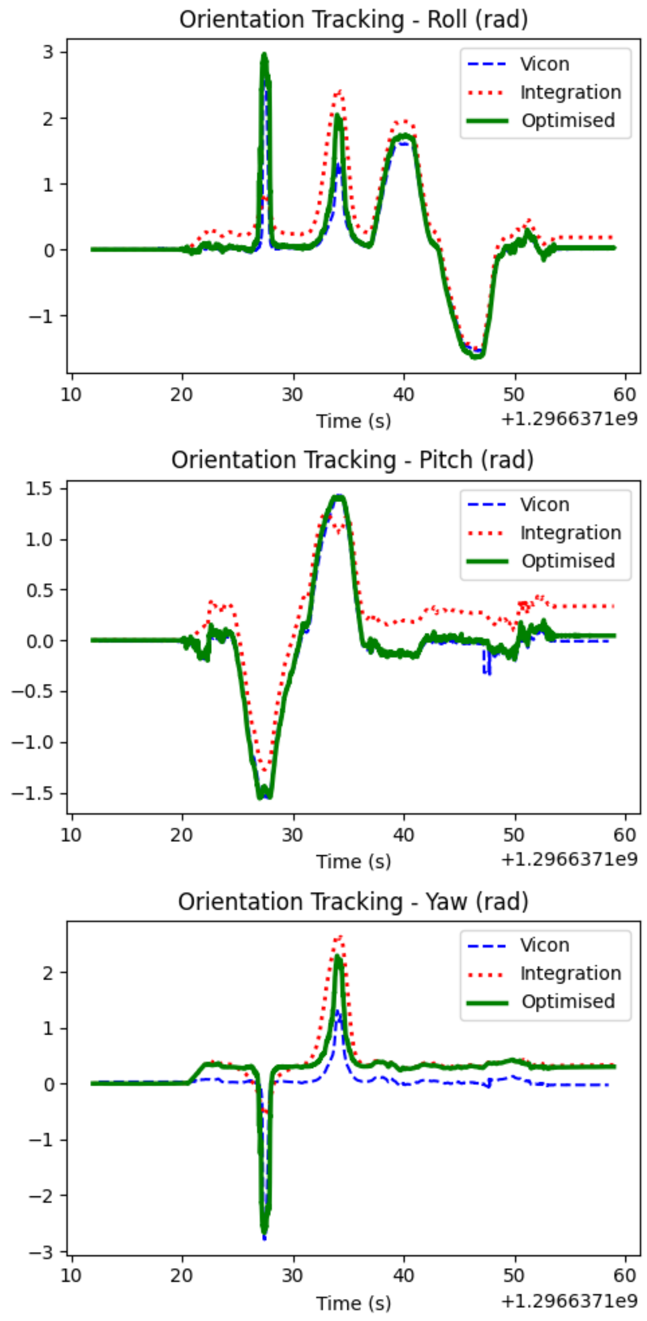


Panorama before optimization of orientation

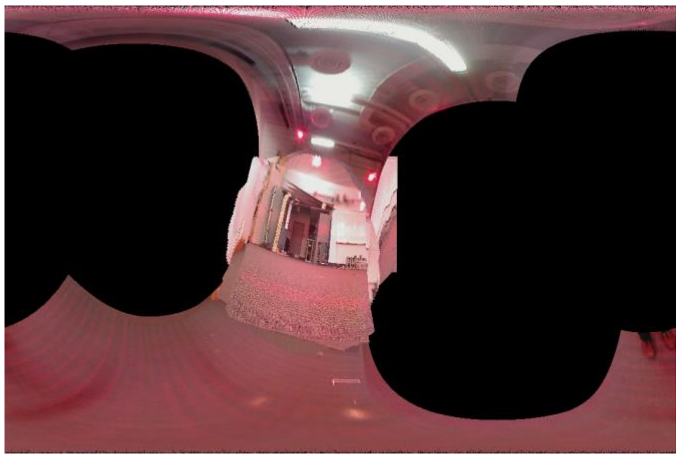


Panorama after optimization of orientation

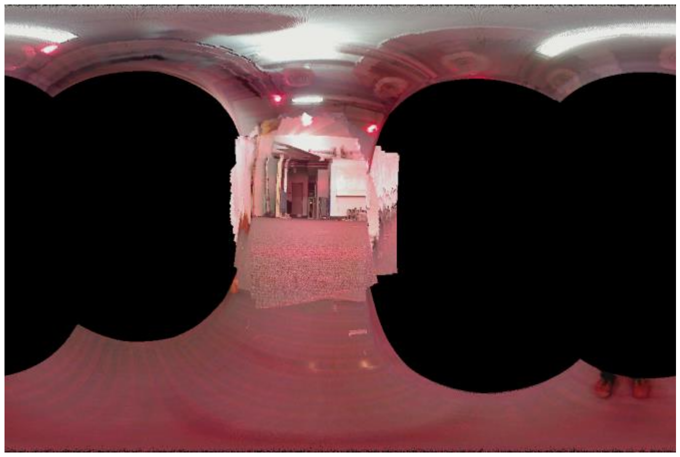
B. Results on Dataset: Trainset "2"



Orientation Euler angles (Roll, Pitch and Yaw) from vicon, motion model (integration) and optimization of orientaion

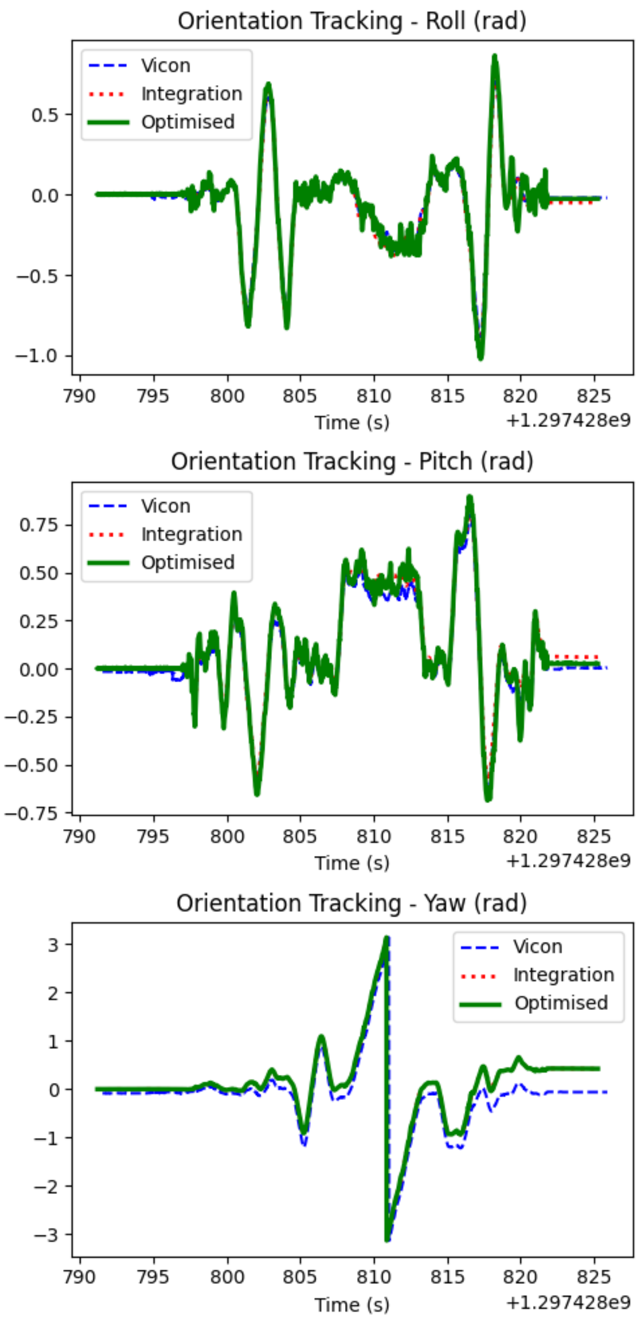


Panorama before optimization of orientation



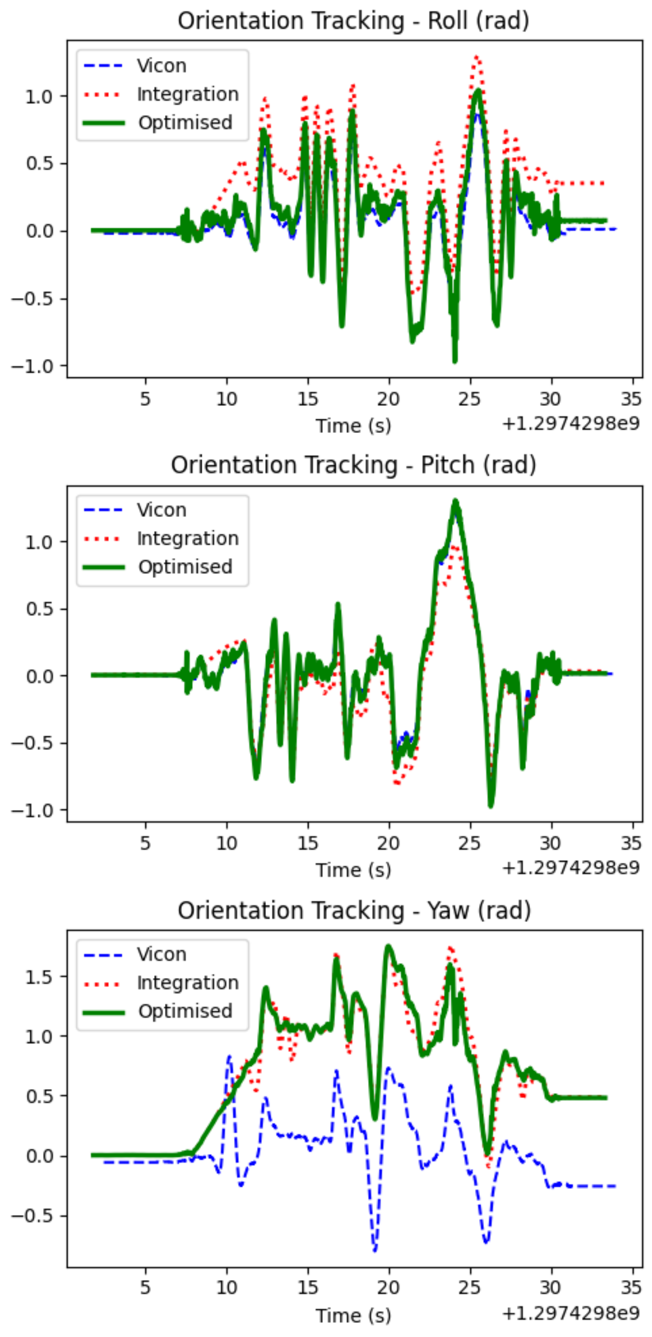
Panorama after optimization of orientation

C. Results on Dataset: Trainset “3”



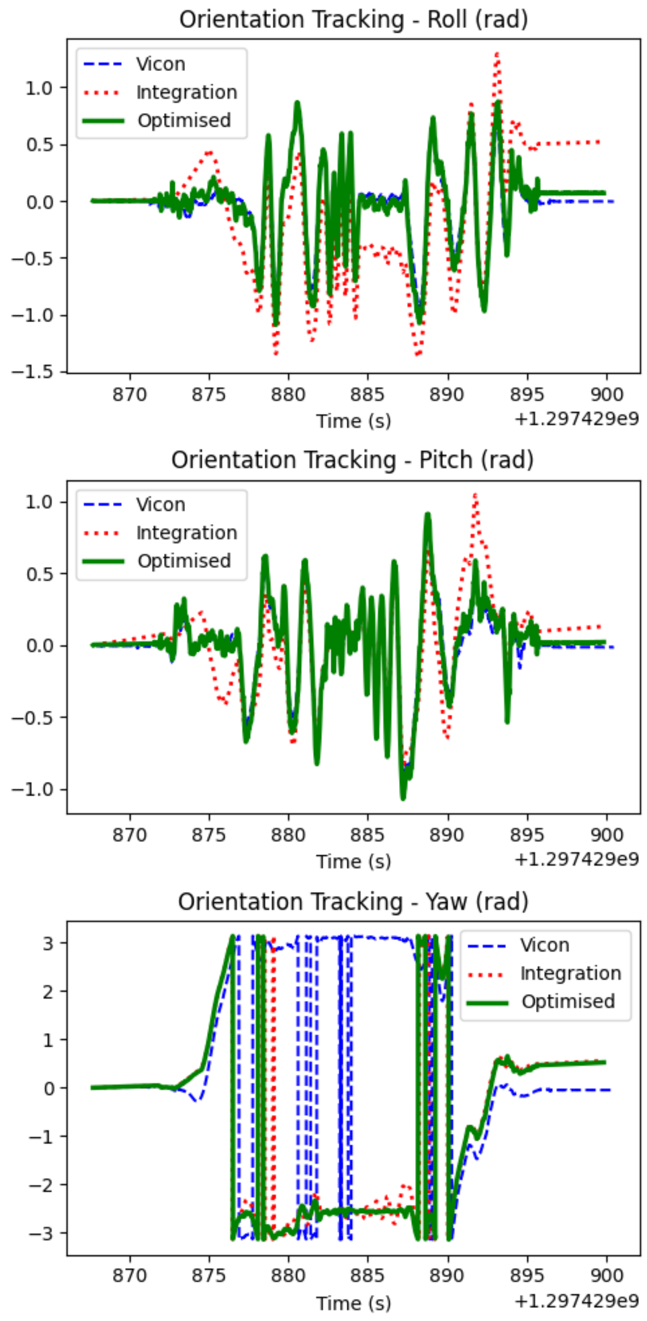
Orientation Euler angles (Roll, Pitch and Yaw) from vicon, motion model (integration) and optimization of orientation

D. Results on Dataset: Trainset “4”



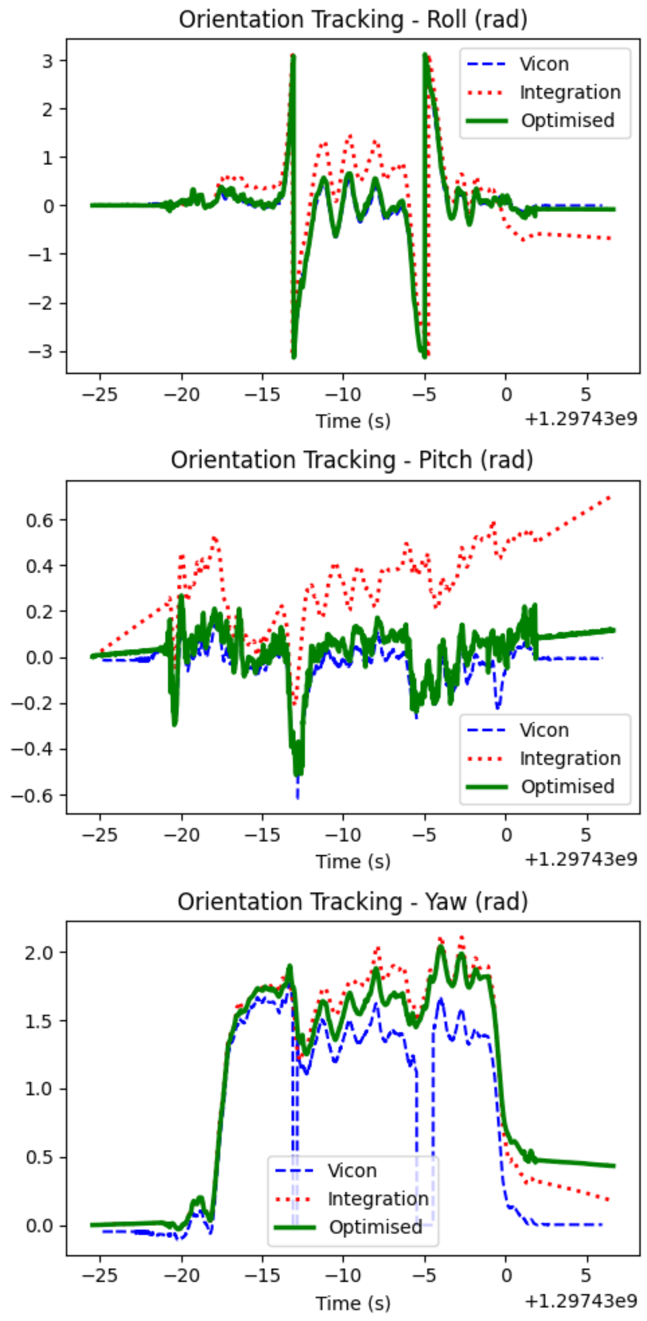
Orientation Euler angles (Roll, Pitch and Yaw) from vicon, motion model (integration) and optimization of orientation

E. Results on Dataset: Trainset "5"



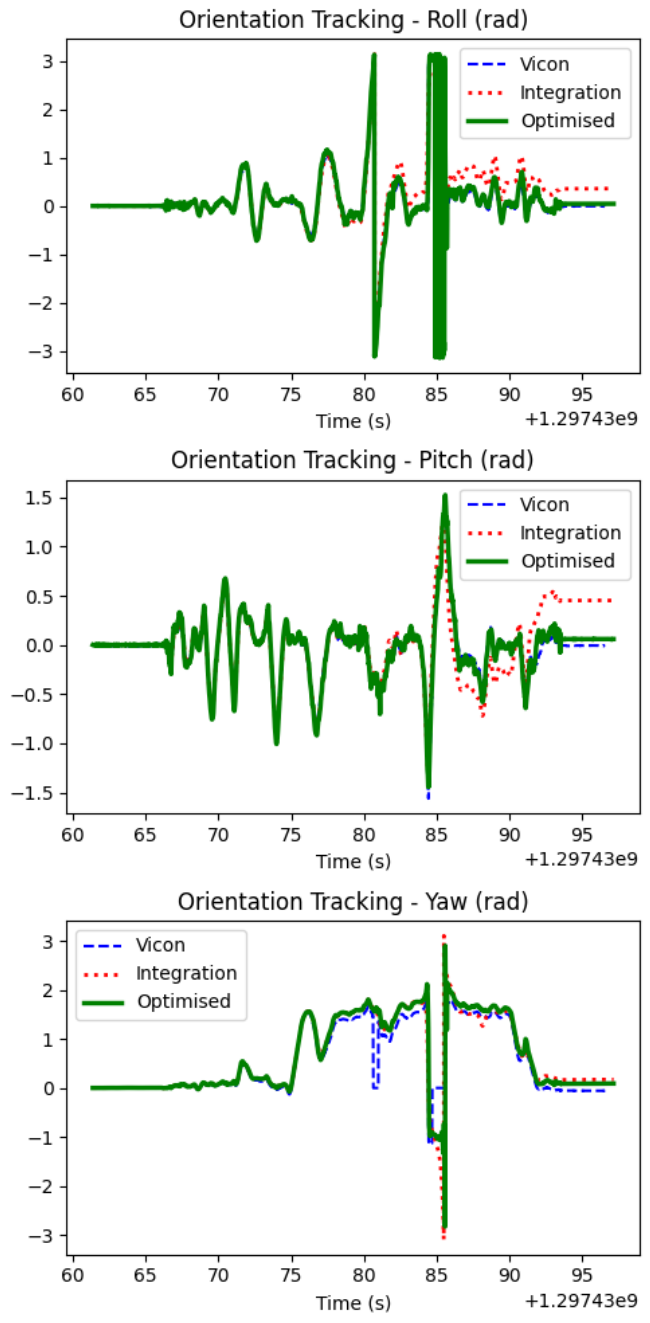
Orientation Euler angles (Roll, Pitch and Yaw) from vicon, motion model (integration) and optimization of orientation

F. Results on Dataset: Trainset "6"



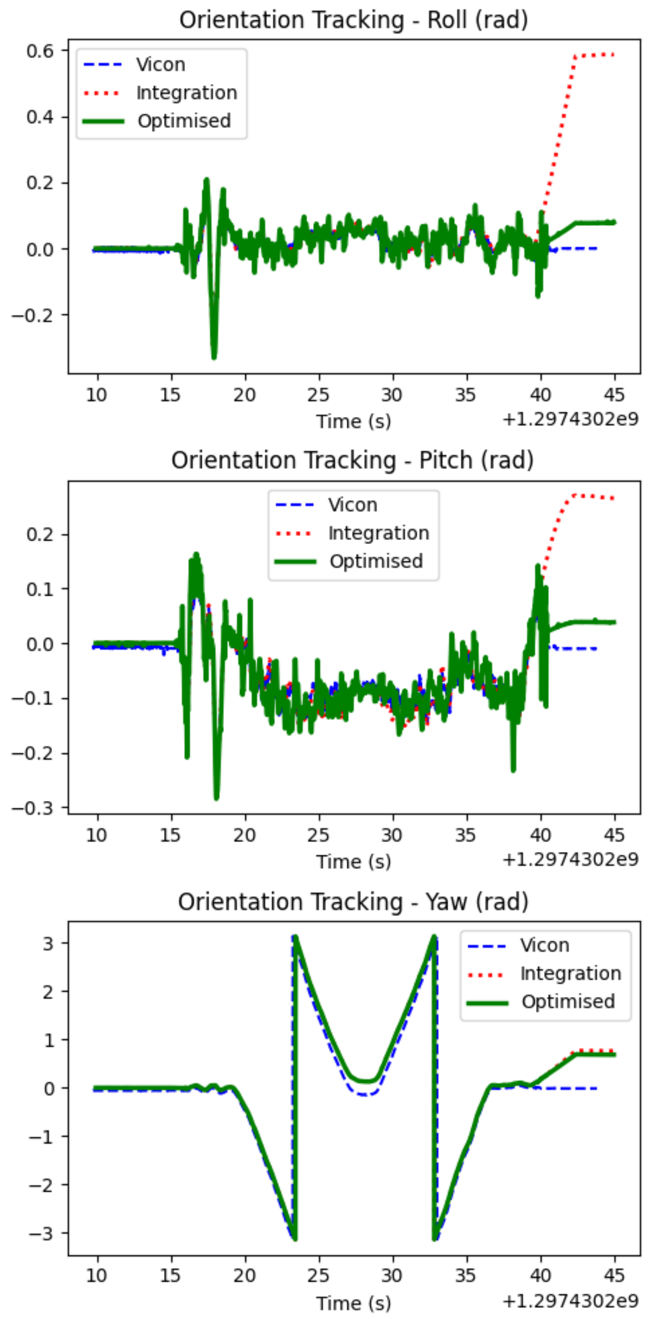
Orientation Euler angles (Roll, Pitch and Yaw) from vicon, motion model (integration) and optimization of orientation

G. Results on Dataset: Trainset "7"

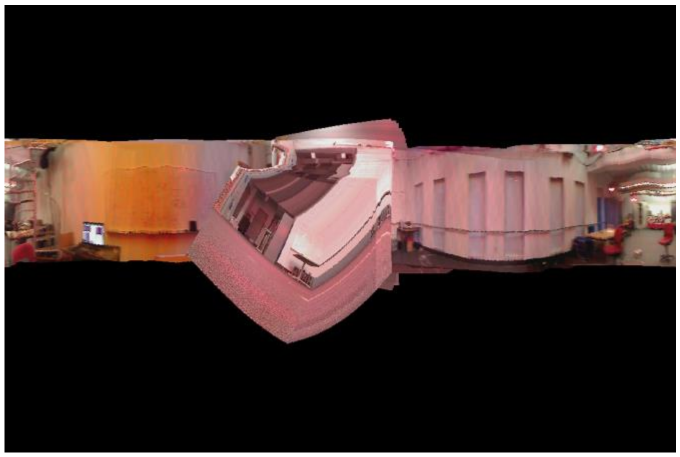


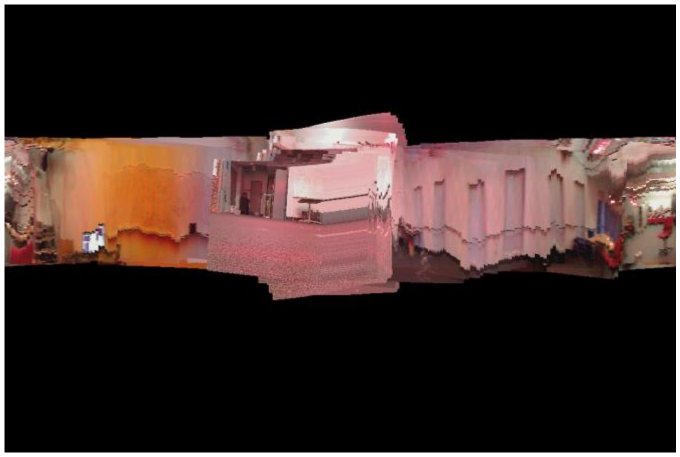
Orientation Euler angles (Roll, Pitch and Yaw) from vicon, motion model (integration) and optimization of orientation

H. Results on Dataset: Trainset "8"

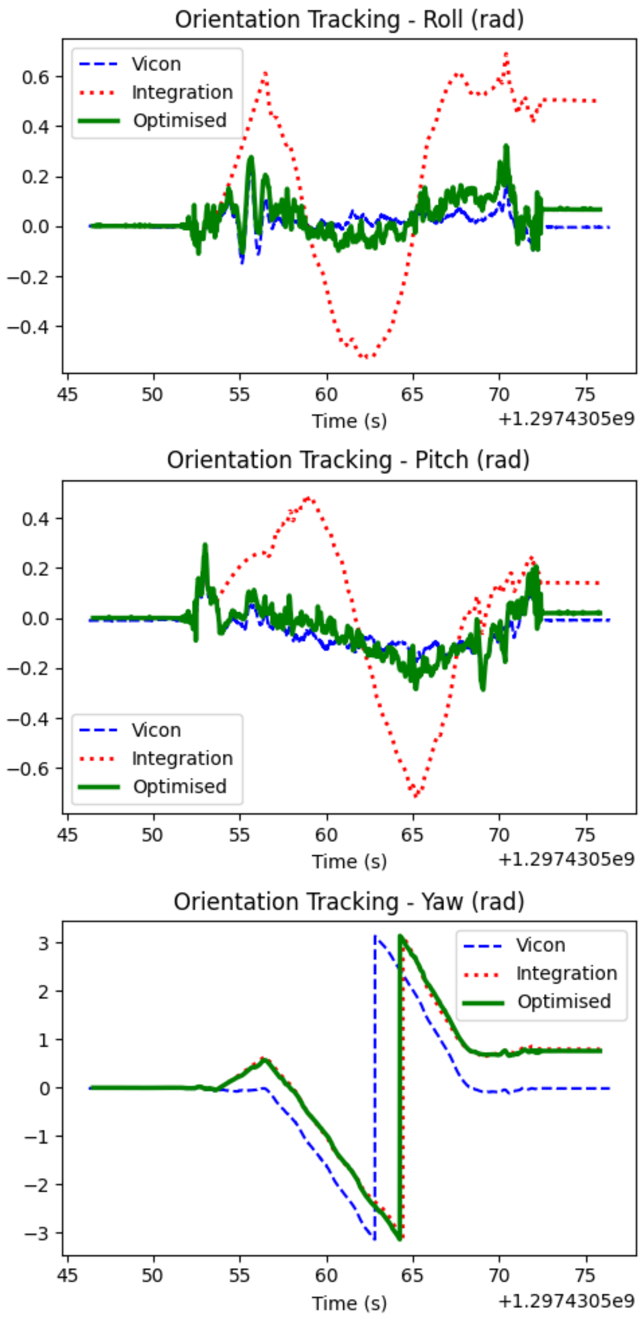


Orientation Euler angles (Roll, Pitch and Yaw) from vicon, motion model (integration) and optimization of orientation

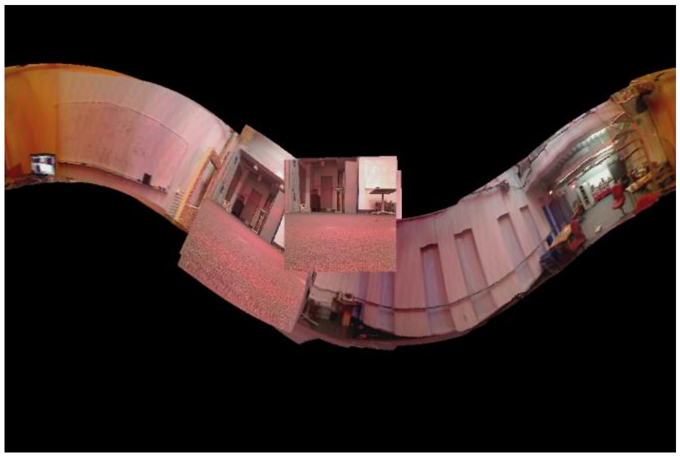

Panorama before optimization of orientation


Panorama after optimization of orientation

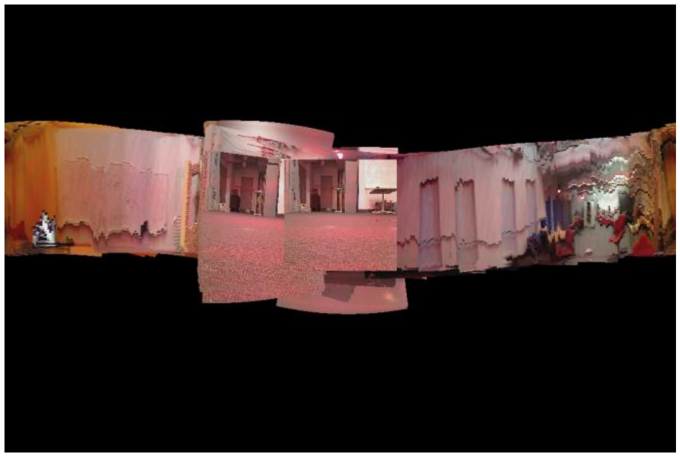
I. Results on Dataset: Trainset "9"



Orientation Euler angles (Roll, Pitch and Yaw) from vicon, motion model (integration) and optimization of orientation

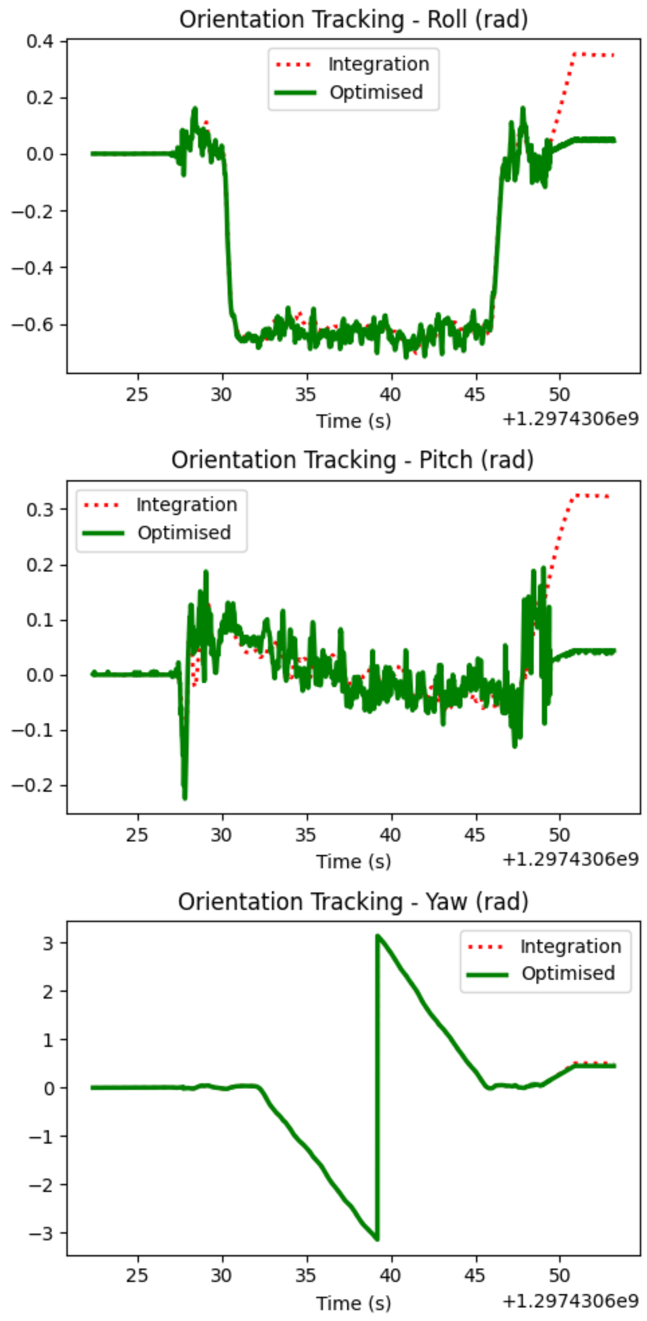


Panorama before optimization of orientation

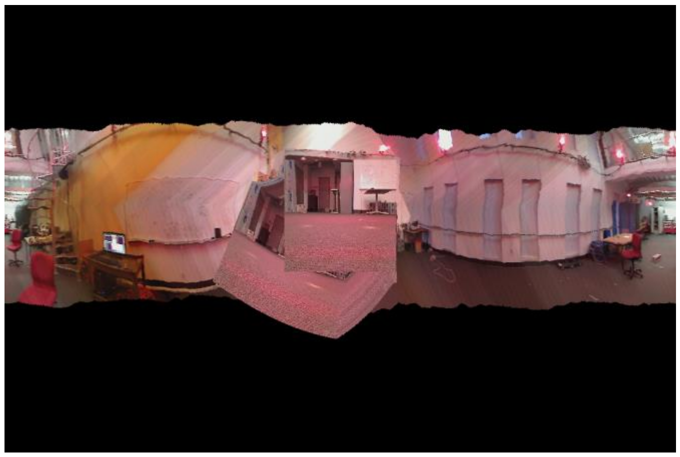


Panorama after optimization of orientation

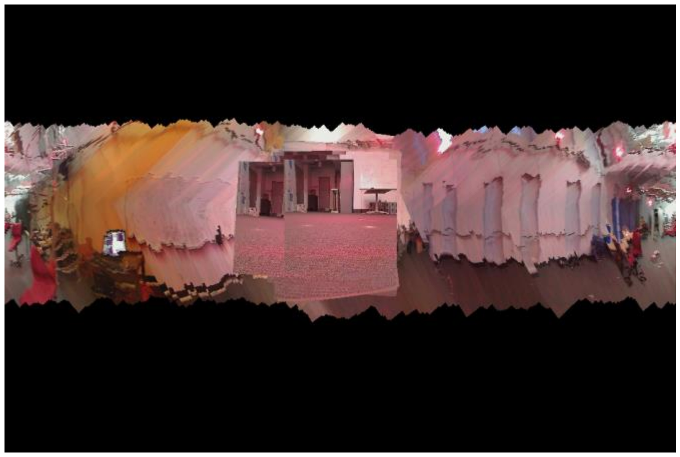
J. Results on Dataset: Testset “10”



Orientation Euler angles from the motion model (integration) and optimization of orientation

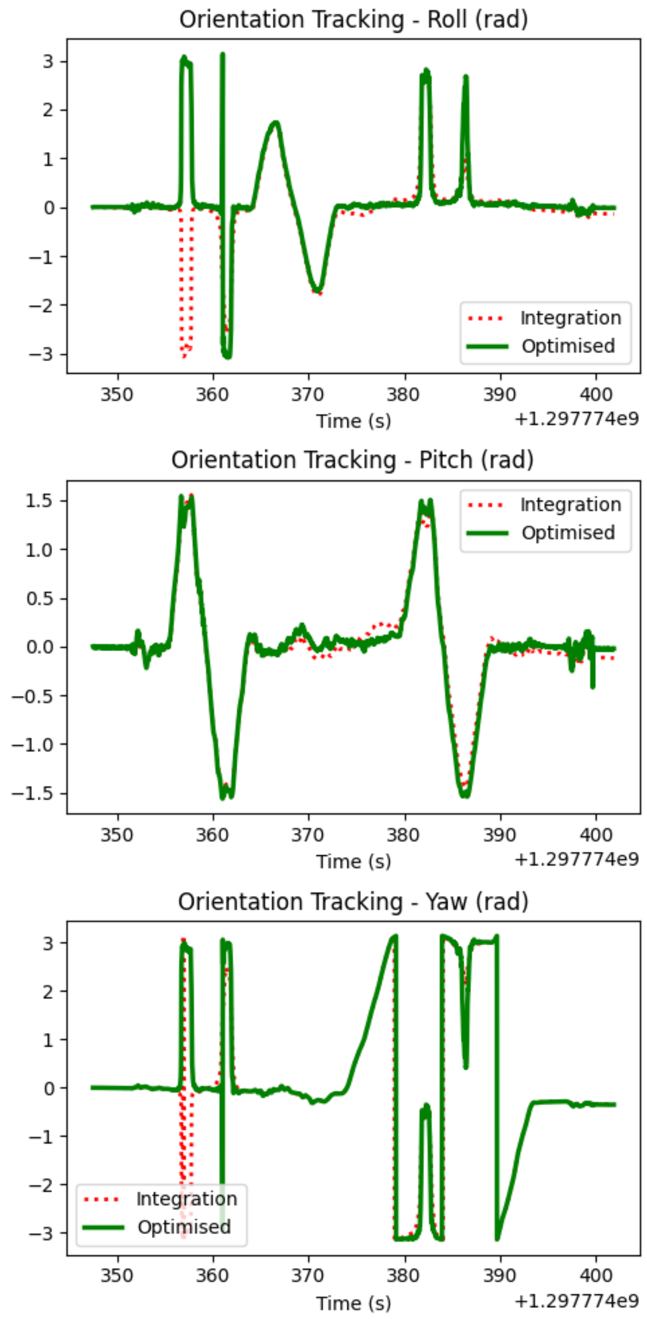


Panorama before optimization of orientation

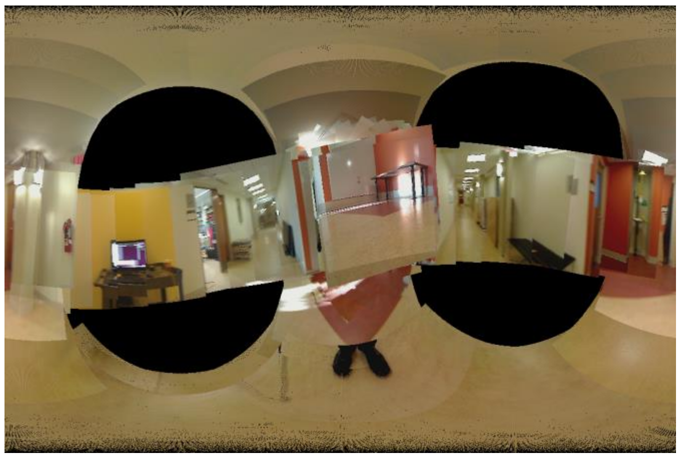


Panorama after optimization of orientation

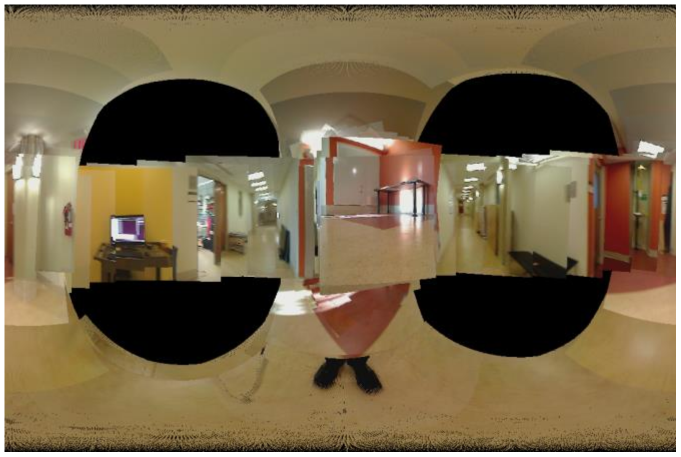
K. *Results on Dataset: Testset "11"*



Orientation Euler angles (Roll, Pitch and Yaw) from the motion model (integration) and optimization of orientation



Panorama before optimization of orientation



Panorama after optimization of orientation