Visual-Inertial SLAM via an Extended Kalman Filter

Allen Zeng
UCSD
azeng@ucsd.edu

Abstract

This paper considers the use of an Extended Kalman Filter (EKF) to solve the Simultaneous Localization and Mapping (SLAM) problem on a subset of the KITTI-360 vehicle driving dataset. The dataset consists of sensor measurements from certain sensors mounted on the moving vehicle: two stereo cameras, and a combined global positioning system and inertial measurement unit (GPS/IMU). Image features are extracted from the stereo images to compute landmark locations. Then, GPS/IMU measurements are combined with the observed features in the EKF framework to estimate the pose of the vehicle over time and the map of landmarks in the environment.

1 Introduction

Autonomous vehicles (AVs) developed in the near future offer many benefits to society, one of which is providing a taxi or delivery service between locations in populated areas. These AVs must be able to navigate through changing environments. Before AVs can autonomously plan and navigate through an environment, an issue that must be addressed is that the AVs need to understand very the environment that they are driving through. Simultaneous Localization and Mapping (SLAM) can be performed to simultaneously build a map of the environment and localize the vehicle within the environment over time.

In this report, the 3-D SLAM problem for vehicles is addressed. From our dataset [1], we use sensor measurements from the forward-facing stereo cameras and the combined global positioning system and inertial measurement unit (GPS/IMU). Using the stereo image information, we extract features, localize them as landmarks in 3-D, and track their positions over time. From the GPS/IMU, we specifically use linear velocity and angular velocity information. Then, an Extended Kalman Filter (EKF) [2] is used to incorporate all of the information and solve the SLAM problem. While 3-D SLAM is performed, the vehicle is assumed to have minimal displacement in the vertical axis, so results are shown projected on to a 2-D plot.

2 Problem Formulation

We are seeking to solve the 3-D Simultaneous Localization and Mapping (SLAM) problem in the case of a vehicle driving through an urban environment. Using the notation of [3, 2], SLAM is a parameter estimation problem that attempts to solve for the robot state $\mathbf{x}_{0:T}$ and environment map $\mathbf{m} \in \mathbb{R}^{3M}$ given a dataset of robot inputs $\mathbf{u}_{0:T-1}$ and observations $\mathbf{z}_{0:T} \in \mathbb{R}^{4\times M\times(T+1)}$. Here, $\mathbf{x}_t := {}_W T_{I,t} \in SE(3)$. The pose ${}_W T_{I,t}$ denotes the transformation from the IMU to world coordinates at time t. This encodes the vehicle's orientation and position. And, $\mathbf{u}_t := [\mathbf{v}_t^\top, \omega_t^\top]^\top \in \mathbb{R}^6$, where v and ω is 3-D linear and 3-D angular velocity respectively. The control inputs $\mathbf{u}_{0:T-1}$ can be inferred using the sensor data [2].

For SLAM here, it is assumed that we know the data association $\Delta_t : \{1, ..., M\} \rightarrow \{1, ..., N_t\}$ for the observations, which matches landmark j to observation $\mathbf{z}_{i,t} \in \mathbb{R}^4$ with $i = \Delta_t(j)$ [2]. That is, the map has M landmarks, each with a position in $\mathbb{R}^3$, and we observe N of them at time t.

Over T timesteps, the goal is then to compute:

$$p(\mathbf{x}_{0:T}, \mathbf{m} \mid \mathbf{z}_{0:T}, \mathbf{u}_{0:T-1})$$

In the mapping step, the vehicle state trajectory $\mathbf{x}_{0:T}$, measurements $\mathbf{z}_{0:T}$, and stereo camera calibration (intrinsic parameters) are used to build map $\mathbf{m}$. In the localization step, the map $\mathbf{m}$, control inputs $\mathbf{u}_{0:T-1}$, and measurements $\mathbf{z}_{0:T}$ are used to infer the vehicle state trajectory $\mathbf{x}_{0:T}$.

As mentioned previously, the SLAM step assumes the availability of the data associations $\Delta_{0:T}$ and the observations $\mathbf{z}_{0:T}$. This correspondence can be computed prior to SLAM using the stereo images in the data. For RGB stereo images $L_{0:T}$ and $R_{0:T}$, where ($L_t, R_t \in \mathbb{R}^{h \times w \times 3}$), we must find a function f that determines $\mathbf{z}_{0:T}$:

$$\mathbf{z}_{i,t} = f(L_{0:T}, R_{0:T})$$

The data associations $\Delta_{0:T}$ are known by finding f. It is assumed that the stereo images provided are undistorted and stereo-rectified.

3 Technical Approach

We utilize the Extended Kalman Filter approach to solve the stated SLAM problem. Assuming we have observations and their data associations (tracked image feature keypoints for each timestep), we first perform IMU-based localization via EKF prediction to get a dead-reckoning estimate of the vehicle trajectory. Then we perform landmark mapping via the EKF update step. Finally, we combine the EKF steps and perform a full visual-inertial SLAM.

Lastly, we no longer assume known feature locations and associations. We discuss our approach to the feature tracking problem.

3.1 Dataset

Two subsets of the KITTI-360 dataset [1] were provided [4]. The first dataset "`0010`" contains, for each timestep, the vehicle's 3-D linear velocity, 3-D angular velocity, and feature correspondences of the stereo images. The velocity information is used as control inputs, and the feature correspondences as observations. The stereo imagery is also provided as a sanity-check to the EKF algorithm, but are not used as part of the estimation.

The second dataset "`0003`" contains the same information except for the feature correspondences. Instead, the feature information must be calculated from the stereo imagery. This process is explained in Section 3.6.

3.2 Extended Kalman Filter

We utilize the Extended Kalman Filter (EKF) approach to solve the stated SLAM problem. Here we summarize the EKF algorithm as explained in [5, 2], and follow the same notation.

The EKF is a Bayes filter that makes several assumptions: the prior PDF $p_{0|0}$ is Gaussian; the motion model is affected by Gaussian noise $\mathbf{w}_t$; the observation model is affected by Gaussian noise $\mathbf{v}_t$; the noise is independent of each other, the state $\mathbf{x}_t$, and time; and the predicted and updated PDFs are forced to be Gaussian by approximation. The predicted and updated PDFs are forced to be Gaussian by evaluating their first and second moments, then approximating them with Gaussians of the same moments. Furthermore, the EKF uses a first-order Taylor series approximation to the motion and observation models around the state and noise means [5].

The EKF steps can be summarized as the following. Given a prior

$$\mathbf{x}_t | \mathbf{z}_{0:t}, \mathbf{u}_{0:t-1} \sim \mathcal{N}(\mu_{t|t}, \Sigma_{t|t}),$$

motion model with noise

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t) \qquad \mathbf{w}_t \sim \mathcal{N}(0, W)$$
$$F_t := \frac{df}{d\mathbf{x}}(\mu_{t|t}, \mathbf{u}_t, \mathbf{0}) \qquad Q_t := \frac{df}{d\mathbf{w}}(\mu_{t|t}, \mathbf{u}_t, \mathbf{0}),$$

and observation model

$$\mathbf{z}_t = h(\mathbf{x}_t, \mathbf{v}_t) \qquad \mathbf{v}_t \sim \mathcal{N}(0, V)$$
$$H_t := \frac{dh}{d\mathbf{x}}(\mu_{t|t-1}, \mathbf{0}) \qquad R_t := \frac{dh}{d\mathbf{v}}(\mu_{t|t-1}, \mathbf{0}),$$

the prediction step is

$$\mu_{t+1|t} = f(\mu_{t|t}, \mathbf{u}_t, \mathbf{0})$$
$$\Sigma_{t+1|t} = F_t \Sigma_{t|t} F_t^\top + Q_t W Q_t^\top$$

and the update step is

$$\mu_{t+1|t+1} = \mu_{t+1|t} + K_{t+1|t}(\mathbf{z}_{t+1} - h(\mu_{t+1|t}, 0))$$
$$\Sigma_{t+1|t+1} = (I - K_{t+1|t} H_{t+1}) \Sigma_{t+1|t},$$

where the Kalman Gain is

$$K_{t+1|t} := \Sigma_{t+1|t} H_{t+1}^\top \left(H_{t+1} \Sigma_{t+1|t} H_{t+1}^\top + R_{t+1} V R_{t+1}^\top \right)^{-1}.$$

3.3 IMU-based Localization via EKF Prediction

We follow the ideas of [2] in this step, and use the same notation. We have pose $T_t := {}_W T_{I,t} \in SE(3)$, with prior $T_t | \mathbf{z}_{0:t}, \mathbf{u}_{0:t-1} \sim \mathcal{N}(\mu_{t|t}, \Sigma_{t|t})$, where $\mu_{t|t} \in SE(3)$ and $\Sigma_{t|t} \in \mathbb{R}^{6\times 6}$. For a continuous-time IMU pose $T(t)$ with noise $\mathbf{w}(t)$, we have a motion model of $\dot{T} = T(\hat{\mathbf{u}} + \hat{\mathbf{w}})$, where $[\mathbf{v}_t^\top, \omega_t^\top]^\top \in \mathbb{R}^6$. Considering Gaussian noise, we express T as a nominal pose $\mu \in SE(3)$ with small perturbation $\hat{\delta\mu} \in \mathfrak{se}(3)$:

$$T = \mu \exp(\hat{\delta\mu}) \approx \mu(I + \hat{\delta\mu})$$

We can then discretize the motion model into nominal kinematics $\mu_{t|t}$ and perturbation kinematics $\delta\mu_{t|t}$, with time discretization τ. We end up with the EKF prediction step as

$$\mu_{t+1|t} = \mu_{t|t} \exp(\tau \hat{\mathbf{u}}_t)$$
$$\Sigma_{t+1|t} = \mathbb{E}[\delta\mu_{t+1|t} \delta\mu_{t+1|t}^\top] = \exp(-\tau \overset{\curlywedge}{\mathbf{u}}_t) \Sigma_{t|t} \exp(-\tau \overset{\curlywedge}{\mathbf{u}}_t)^\top + W$$

where

$$\mathbf{u}_t := \begin{bmatrix} \mathbf{v}_t \\ \omega_t \end{bmatrix} \in \mathbb{R}^6 \qquad \hat{\mathbf{u}}_t := \begin{bmatrix} \hat{\omega}_t & \mathbf{v}_t \\ \mathbf{0}^\top & 0 \end{bmatrix} \in \mathbb{R}^{4\times 4} \qquad \overset{\curlywedge}{\mathbf{u}}_t := \begin{bmatrix} \hat{\omega}_t & \hat{\mathbf{v}}_t \\ 0 & \hat{\omega}_t \end{bmatrix} \in \mathbb{R}^{6\times 6}$$

3.4 Landmark Mapping via EKF Update

For the landmark mapping step, we again follow the ideas and notation of [2]. With known stereo camera intrinsic calibration matrix M, extrinsic calibration ${}_O T_I \in SE(3)$ (from IMU to observation/camera frame), we use the priors $\mu_t \in \mathbb{R}^{3M}, \Sigma_t \in \mathbb{R}^{3M\times 3M}$, IMU pose $T_{t+1} \in SE(3)$ (assumed known from prediction), and observation $\mathbf{z}_{t+1} \in \mathbb{R}^{4N_{t+1}}$ to perform the EKF Update step.

Our observation model with measurement noise $\mathbf{v}_{t,i} \sim \mathcal{N}(0, V)$ is

$$\mathbf{z}_{t,i} = h(T_t, \mathbf{m}_j) + \mathbf{v}_{t,i} := M\pi\left({}_O T_I T_{t+1}^{-1} \underline{\mathbf{m}}_j\right) + \mathbf{v}_{t,i}.$$

Here, the underline denotes a homogeneous coordinate.

First we calculate the predicted observations based on μ_t and the known correspondences Δ_{t+1},

$$\tilde{\mathbf{z}}_{t+1,i} := M\pi\left({}_O T_I T_{t+1}^{-1} \underline{\mu}_{t,j}\right) \in \mathbb{R}^4$$

for $i = 1, ..., N_{t+1}$. We also calculate the Jacobian of $\tilde{\mathbf{z}}_{t+1,i}$ with respect to $\mathbf{m}_j$ evaluated at μ_t, j,

$$H_{t+1,i,j} = \begin{cases} M \frac{d\pi}{d\mathbf{q}} \left({}_O T_I T_{t+1}^{-1} \underline{\mu}_{t,j} \right) \; {}_O T_I T_{t+1}^{-1} P^\top & \text{if } \Delta_t(j) = i \\ \mathbf{0} \in \mathbb{R}^{4\times 3} & \text{otherwise} \end{cases}$$

The function $\pi(\mathbf{q}) := \frac{1}{q_3}\mathbf{q} \in \mathbb{R}^4$, and

$$\frac{d\pi}{d\mathbf{q}} = \frac{1}{q_3} \begin{bmatrix} 1 & 0 & -\frac{q_1}{q_3} & 0 \\ 0 & 1 & -\frac{q_2}{q_3} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{q_4}{q_3} & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4}$$

Thus, the EKF update is

$$\begin{aligned} K_{t+1|t} &= \Sigma_{t+1|t} H_{t+1}^\top \left(H_{t+1} \Sigma_{t+1|t} H_{t+1}^\top + I \otimes V \right)^{-1} \\ \mu_{t+1|t+1} &= \mu_{t+1|t} + K_{t+1|t}(\mathbf{z}_{t+1} - \tilde{\mathbf{z}}_{t+1}) \\ \Sigma_{t+1|t+1} &= (I - K_{t+1|t} H_{t+1}) \Sigma_{t+1|t}, \end{aligned}$$

where $I \otimes V$ denotes the Kronecker product of the two matrices. The result is a block diagonal matrix with blocks of V.

3.5 Visual-Inertial SLAM

The IMU prediction step and the landmark update step is combined to form an IMU update step, obtaining a complete visual-inertial SLAM algorithm [4]. We use the same motion and observation models as before. Here, we have a prior $T_{t+1} | \mathbf{z}_{0:t}, \mathbf{u}_{0:t} \sim \mathcal{N}(\mu_{t+1|t}, \Sigma_{t+1|t})$ where $\mu_{t+1|t} \in SE(3), \Sigma_{t+1|t} \in \mathbb{R}^{6\times 6}$. We have known stereo calibration M, extrinsic ${}_O T_I \in SE(3)$, landmark positions $\mathbf{m} \in \mathbb{R}^{3M}$, and new observations $\mathbf{z}_{t+1} \in \mathbb{R}^{4N_{t+1}}$.

We calculate $\tilde{\mathbf{z}}_{t+1,i}$ similar to before, but use known (calculated) $\mathbf{m}_j$. We are instead interested in determining the IMU pose T_{t+1}, so

$$\tilde{\mathbf{z}}_{t+1,i} := M\pi \left({}_O T_I \mu_{t+1|t}^{-1} \underline{\mathbf{m}}_j \right) \in \mathbb{R}^4$$

for $i = 1, ..., N_{t+1}$. The new Jacobian is calculated with respect to T_{t+1} and evaluated at $\mu_{t+1|t}$.

$$H_{t+1,i} = -M \frac{d\pi}{d\mathbf{q}} \left({}_O T_I \mu_{t+1|t}^{-1} \underline{\mathbf{m}}_j \right) \; {}_O T_I \left(\mu_{t+1|t}^{-1} \underline{\mathbf{m}}_j \right)^{\odot} \in \mathbb{R}^{4\times 6}$$

where the operator $\begin{bmatrix} \mathbf{s} \\ 1 \end{bmatrix}^{\odot} := \begin{bmatrix} I & -\hat{\mathbf{s}} \\ 0 & 0 \end{bmatrix}^{\odot} \in \mathbb{R}^{4\times 6}$. Then vertically stacking all N_{t+1} calculated $H_{t+1,i}$ into a single matrix H_{t+1}, we end up with the following EKF update equations:

$$\begin{aligned} K_{t+1|t} &= \Sigma_{t+1|t} H_{t+1}^\top \left(H_{t+1} \Sigma_{t+1|t} H_{t+1}^\top + I \otimes V \right)^{-1} \\ \mu_{t+1|t+1} &= \mu_{t+1|t} \exp \left(\left(K_{t+1|t} (\mathbf{z}_{t+1} - \tilde{\mathbf{z}}_{t+1}) \right)^{\wedge} \right) \\ \Sigma_{t+1|t+1} &= (I - K_{t+1|t} H_{t+1}) \Sigma_{t+1|t} \end{aligned}$$

where $I \otimes V$ denotes the Kronecker product of the two matrices.

Next, in order to estimate the pose of the vehicle and the position of the landmarks simultaneously, we jointly estimate the all of the states and the covariances as one large problem. We stack the vehicle pose and landmark positions together:

$$\mu = \begin{bmatrix} \mu_V \\ \mu_L \end{bmatrix} \in \mathbb{R}^{6+3M}$$

$$\Sigma = \begin{bmatrix} \Sigma_V & 0 \\ 0 & \Sigma_L \end{bmatrix} \in \mathbb{R}^{(6+3M)\times(6+3M)}$$

where V and L denote the variables for the vehicle and landmarks respectively. Prediction then becomes

$$
\begin{aligned}
\mu_{t+1|t} &= \begin{bmatrix} \mu_{V,t+1|t} \\ \mu_{L,t+1|t} \end{bmatrix} = \begin{bmatrix} \mu_{V,t+1|t} \exp(\tau \hat{\mathbf{u}}_t) \\ \mu_{L,t+1|t} \end{bmatrix} \\
\Sigma_{t+1|t} &= F_t \Sigma_{t|t} F_t^\top + W \\
F_t &= \begin{bmatrix} \exp(-\tau \overset{\curlywedge}{\hat{\mathbf{u}}}_t) & \mathbf{0} \\ \mathbf{0} & I \end{bmatrix} \\
W &= \begin{bmatrix} W_V & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}
\end{aligned}
$$

Where W_V is the noise on the vehicle's pose.

And update then becomes

$$
\begin{aligned}
H_{t+1} &= \begin{bmatrix} H_{V,t+1|t} & H_{L,t+1|t} \end{bmatrix} \in \mathbb{R}^{4N_{t+1} \times (6+3M)} \\
K_{t+1|t} &= \Sigma_{t+1|t} H_{t+1}^\top \left(H_{t+1} \Sigma_{t+1|t} H_{t+1}^\top + I \otimes V \right)^{-1} \\
\mu_{t+1|t+1} &= \begin{bmatrix} \mu_{V,t+1|t+1} \\ \mu_{L,t+1|t+1} \end{bmatrix} = \begin{bmatrix} \mu_{V,t+1|t} \exp\left(\left(K_{t+1|t} (\mathbf{z}_{t+1} - \tilde{\mathbf{z}}_{t+1}) \right)^\wedge \right) \\ \mu_{L,t+1|t} + K_{t+1|t} (\mathbf{z}_{t+1} - \tilde{\mathbf{z}}_{t+1}) \end{bmatrix} \\
\Sigma_{t+1|t+1} &= (I - K_{t+1|t} H_{t+1}) \Sigma_{t+1|t}
\end{aligned}
$$

. Where $H_{V,t+1|t}$ is from earlier in this section and $H_{L,t+1|t}$ is from Section 3.4.

3.6 Feature Detection and Matching

As mentioned before, the "`0003`" dataset does not contain pre-computed features. So we must perform feature detection and tracking to obtain the feature correspondence information. For each pair of stereo images, we compute keypoint using our own implementation of the Harris corner detector [6]. Then, stereo matching is performed using optical flow from the left to the right image. We use OpenCV's implementation of the Pyramidal Lucas Kanade Feature Tracker [7]. Additionally, once initialized, features are tracked over time using optical flow as well. This processing recovers the data associations $\Delta_{0:T}$ and the observations $\mathbf{z}_{0:T}$ such that SLAM can be performed on this dataset.

4 Results

For the provided features in dataset `0010`, I downsampled the observations by a factor of 10 to have a reasonable computation time and memory usage. For every 10 features, I calculate the number of times that the features were observed over all timesteps, and keep only the single feature with the most observations.

In terms of project instructions [4], parts (a) and (c) were implemented first. This initial result in pictured in Figure 1, corresponding to dataset `0010`. It assumes the vehicle's initial position is the world origin, and the body frame is the IMU frame. Note in this case, the IMU prediction was performed without incorporating and map information, and the landmark positions are estimated assuming the IMU trajectory is correct. As noted in the title of the figure, this is not the SLAM step.

Upon examining the dataset, I found that the IMU angular velocity measurements provided in [4] were likely in a forward-left-upward coordinate frame and not in the forward-right-downward coordinate frame pictured in KITTI-360 [1]. Whenever the vehicle turned right (as visualized by the stereo camera image sequence), the plot in Figure 1 also showed a clockwise trajectory (and vice versa turning left), implying the body/IMU frame is forward-left-upward. If the data is actually given in forward-right-downward, then I believe that turning right should result in counter-clockwise trajectories, and vice versa.

This is thought is further shown in Figure 2. Here I show an initial result for Visual-Inertial SLAM (VI-SLAM). After accounting for the landmark observations, the EKF attempts to correct the IMU trajectory into being counter-clockwise whenever turning right in the real world. (Again, I double-checked the car rotation by visual inspection of the camera data.) Because the IMU data and the landmark observations result in conflicting pose updates, the result shown in the figure is poor.

Additionally, I checked the raw OXTS measurements from KITTI360. Measurements for the angular rate around the forward, leftward, and upward axes were available. Primarily examining the upward axes angular rates (since we are visualizing the car turning left and right), these angular rates had the same sign and roughly the same magnitude as the provided "IMU" angular velocities. So, in the end I applied a rotation from forward-left-upward to forward-right-downward on the provided IMU data prior to performing SLAM.

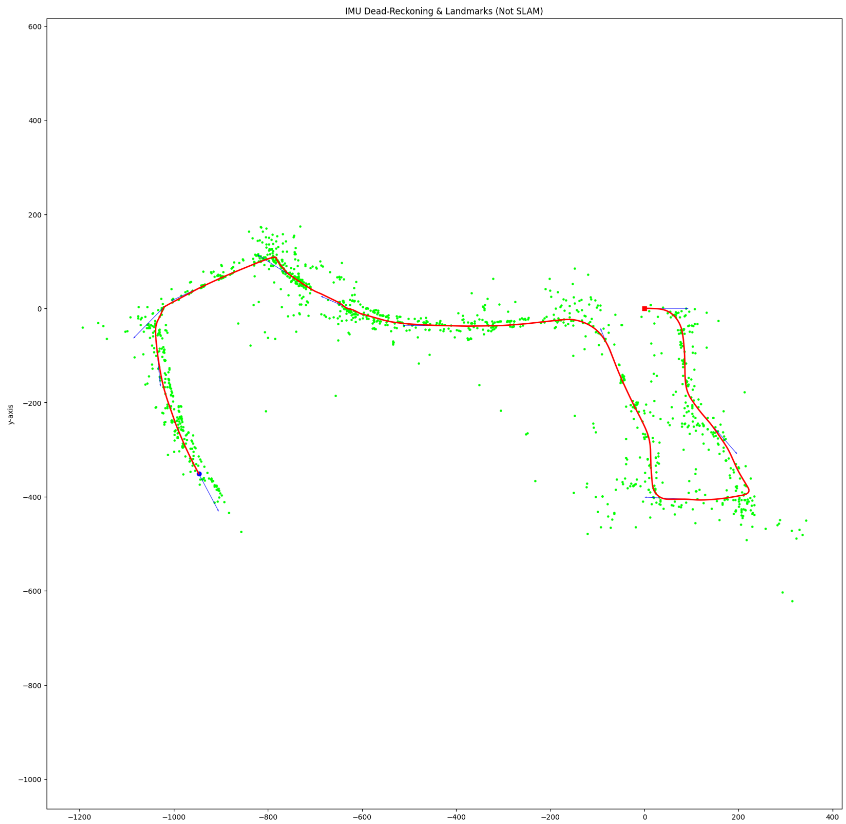

Figure 1: Initial IMU-based Localization and Landmark Mapping result.

Now, I have a revised plot in Figure 3 corresponding to parts (a) and (c). And, I have some reasonable results for part (d) shown in Figures 4 and 5. It appears that the EKF results are sensitive to the motion noise parameters, and less the other parameters. The result for Figures 4 and 5 make sense when compared to the stereo images. Again, note that the plots are in the body/IMU frame, which has a forward-right-downward frame [1]. So, real-world right turns result in a counter-clockwise trajectory in the plot, and vice versa.

For Figures 4 and 5, they were generated using a prior pose covariance of 1e-1 $\cdot I_6$, prior landmark covariance of 1e-1 $\cdot I_3$, and observation model noise of $10 \cdot I_4$. Figure 4 had motion model noise of 1e-1 $\cdot I_6$. And, figure 5 was generated with motion model noise of 1e-5 $\cdot I_6$.

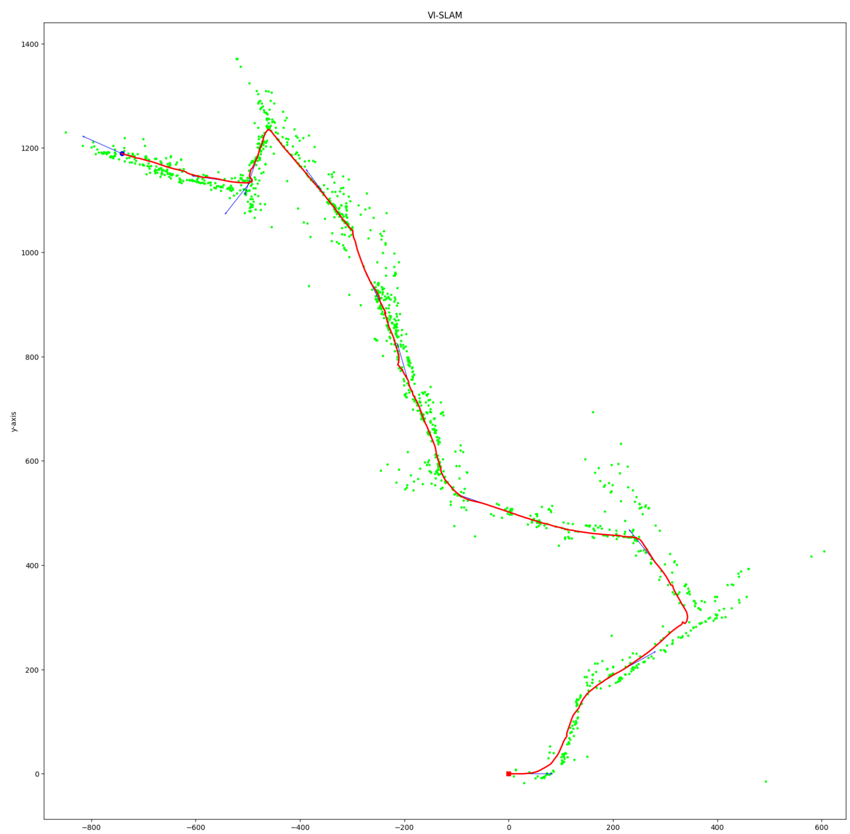

Figure 2: Initial Visual-Inertial SLAM result. The results are poor because of conflicting information between the original provided IMU measurements and feature observations

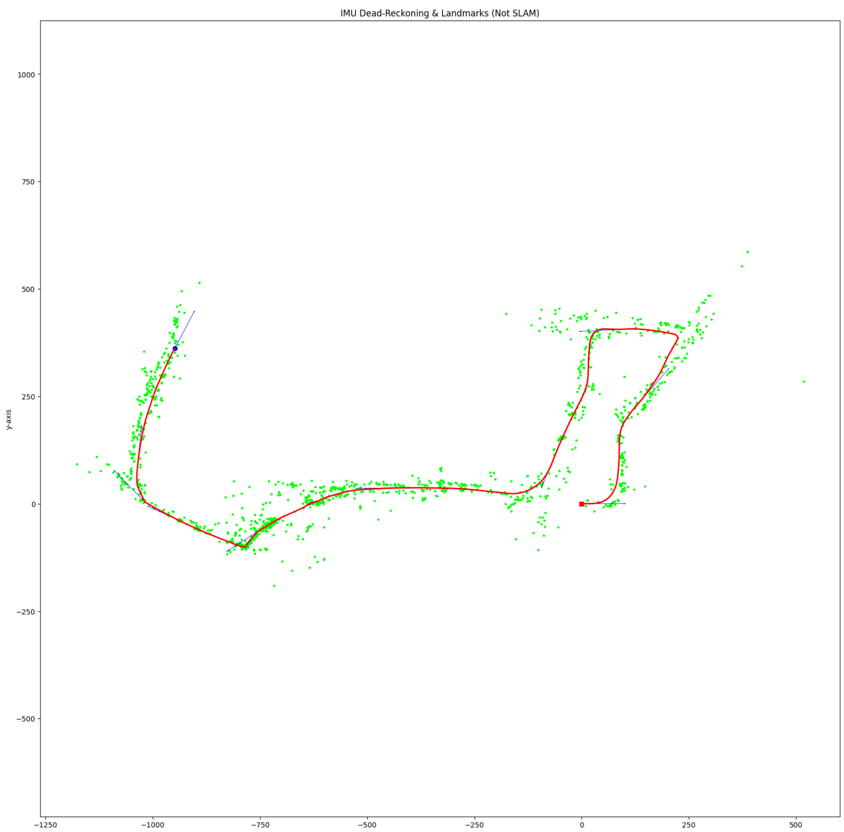

Figure 3: IMU-based Localization and Landmark Mapping result after correcting the provided IMU measurements.

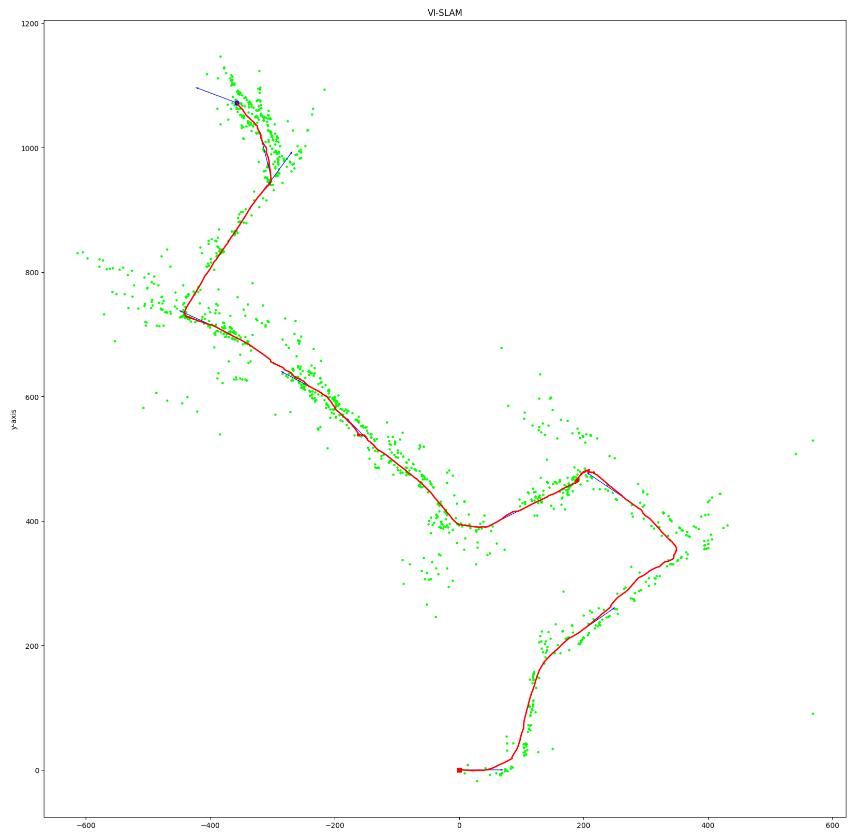

Figure 4: Visual-Inertial SLAM result after correcting the provided IMU measurements. Motion model noise of 1e-1 $\cdot I_6$.

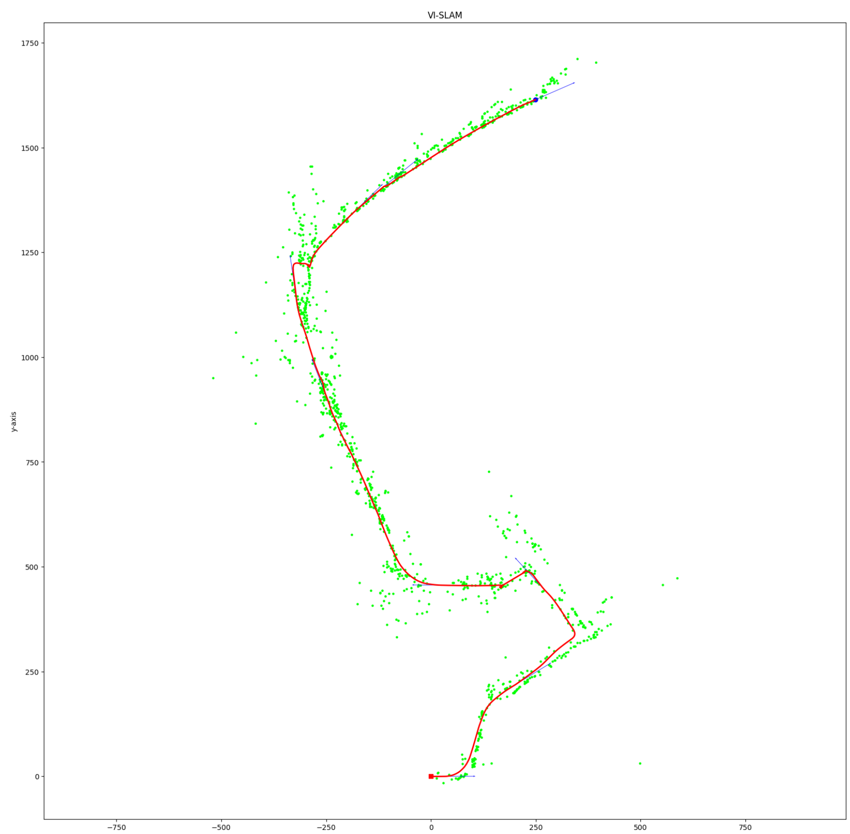

Figure 5: Visual-Inertial SLAM result after correcting the provided IMU measurements. Motion model noise of 1e-5 $\cdot I_6$.

Shown in Figure 6, are Harris corner detections of the initial left and right image of the dataset `0003`. The corners in the top image (left) were detected using my own implementation of Harris corner detection. The bottom image (right) has corresponding corners found by using OpenCV's `calcOpticalFlowPyrLK`. I have also implemented the tracking of features in the left camera from frame to frame, again using `calcOpticalFlowPyrLK`. However, due to time constraints, I was unable to run SLAM on this dataset using these new observations.

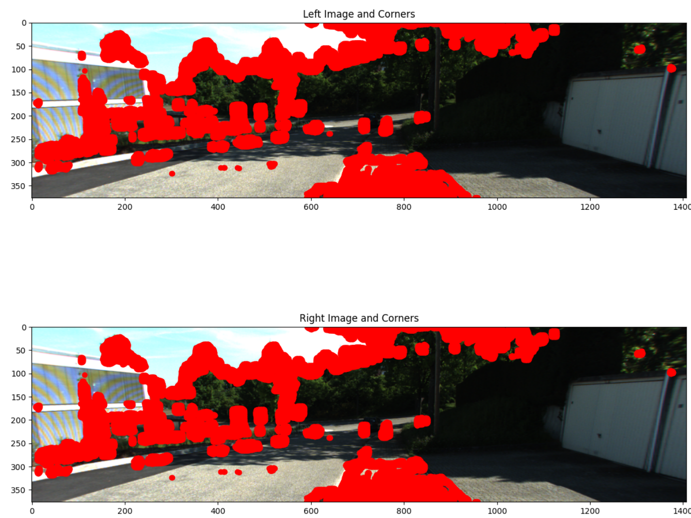

Figure 6: Harris corner detection (top) and correspondences found by optical flow (bottom).

References

[1] Jun Xie, Martin Kiefel, Ming-Ting Sun, and Andreas Geiger. Semantic instance annotation of street scenes by 3d to 2d label transfer. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[2] Nikolay Atanasov. Lecture 13: Visual-inertial slam. In *ECE276A: Sensing & Estimation in Robotics*, 2021.

[3] Nikolay Atanasov. Lecture 9: Particle filter slam. In *ECE276A: Sensing & Estimation in Robotics*, 2021.

[4] Nikolay Atanasov. Project 3: Visual-inertial slam. In *ECE276A: Sensing & Estimation in Robotics*, 2021.

[5] Nikolay Atanasov. Lecture 11: Extended and unscented kalman filtering. In *ECE276A: Sensing & Estimation in Robotics*, 2021.

[6] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, 1988.

[7] J.-Y. Bouguet. Pyramidal implementation of the lucas kanade feature tracker. 1999.