Visual-Inertial SLAM

Shakeel Ahamed Mansoor Shaikna

Mechanical and Aerospace Engineering
University of California San Diego
San Diego, CA, US
samansoo@uscd.edu

***Abstract*—This report presents a study of Simultaneous Localization and mapping (SLAM) using Extended Kalman Filter (EKF) with prediction based on Inertial Measurement Unit (IMU) data and update based on stereo camera observation model. We use the IMU data for localization using EKF predict and stereo camera model to build a 3D landmark mapping of the environment using EKF update. Combining these two steps we perform the visual- inertial SLAM.**

I. Introduction

In recent years, many research has been taking place in the fields of autonomous vehicles/systems. The main purpose of the autonomous systems is to reduce human error and provide self navigation. Usually robots navigate through the given environment with either following a trajectory or through path planning algorithms. However, when both the environment and the location is unknown we use the SLAM where it simultaneously localizes the robot position and maps the environment from various sensor readings.

The data given to us are the IMU data and the stereo camera observations from a vehicle which is exploring the environment. We model the IMU data as our motion model, and the stereo camera data serves as an observation model. It is necessary, to always model the states, observations and control inputs as a probability measure. So, we model it as a type of Bayesian Filter.

In this report, I used Extended Kalman filter (EKF) for implementing the SLAM problem. We use EKF because our states and measurements are not linear. So, we linearize the system using EKF and use the prediction and update steps of Kalman Filter. The three main steps in this project is the IMU based Localization via EKF Predict, Landmark Mapping via EKF Update and Visual-inertial SLAM.

The first step is to find the estimate of the pose based on the pose kinematics using EKF prediction step. Now, assuming that we know the positions we can now estimate the landmark positions using EKF update step. We also assume the landmarks doesn't move, so we don't perform EKF predict on the landmark state. Combining the IMU predict and the landmark update, we update the IMU using landmark positions to perform visual-inertial SLAM.

Section II discusses the problem which I am trying to solve. Section III explains in detail the approach taken to solve the problems stated in Section II i.e. approach to visual-inertial SLAM. Section IV presents the results from doing visual-inertial SLAM for each of the data given to us.

II. Problem Formulation

Our objective is to sense the environment using the stereo camera data to generate a Landmark Mapping of the environment using the pose of the robot found using the IMU data. The end result is to find the IMU pose ${}_W\mathbf{T}_I \in SE(3)$ in the world frame over time and finding world-frame coordinates of the point landmarks $\mathbf{m} \in \mathbb{R}^{3\times M}$ which generates the visual features $\mathbf{z}_t$.

A. Visual Mapping

Given a robot trajectory $\mathbf{x}_{0:T}$, we need to generate a map represented as m of the environment. For generating the map, we use landmark mapping. Given the inverse IMU pose $\mathbf{U}_t$ and the visual features observations from the IMU and stereo camera data respectively, we estimate the homogeneous coordinates of the point landmarks $\underline{\mathbf{m}}$.

We use EKF update for estimating the coordinates of the point landmarks $\underline{\mathbf{m}}$. Given a prior Gaussian distribution for $\mathbf{m} \mid \mathbf{z}_{0:t}$, we can find the observation model with observation noise $\mathbf{v}_{t,i} \sim \mathcal{N}(0, V)$.

$$\mathbf{z}_{t,i} = h(\mathbf{U}_t, \mathbf{m}_j) + \mathbf{v}_{t,i} \tag{1}$$

Before performing EKF update step, we first linearize the observation model by finding the Jacobian $H_t \in \mathbb{R}^{4N_t \times 3M}$, where N_t is the number of landmarks observed at time t and M is the total number of landmarks present. After linearizing the model, perform EKF update step, to find μ_{t+1} and Σ_{t+1} from prior $\mu_t \in \mathbb{R}^{3M}$ and $\Sigma_t \in \mathbb{R}^{3M\times 3M}$.

B. Localization

Given a map m of the environment, we need to localize the robot and estimate its trajectory $x_{0:T}$. Given the IMU measurements $\mathbf{u}_{0:T}$ and the visual feature observations $\mathbf{z}_{0:T}$, we estimate the inverse IMU pose $\mathbf{U}_t$.

We use EKF predict for estimating the inverse IMU pose $\mathbf{U}_t$. Given a prior Gaussian distribution for $\mathbf{U}_t \mid \mathbf{z}_{0:t}, \mathbf{m}_{0:t-1}$, we can find the motion model with noise $\mathbf{w}_t \sim \mathcal{N}(0, W)$.

$$\mathbf{U}_{t+1} = exp(-\tau(\mathbf{u}_t + \mathbf{w}_t)^\wedge)\mathbf{U}_t \tag{2}$$

By performing EKF predict step, we find $\mu_{t+1|t}$ and $\Sigma_{t+1|t}$ from prior $\mu_t \in SE(3)$ and $\Sigma_t \in \mathbb{R}^{6\times 6}$.

We combine these two and implement the visual-inertial SLAM. SLAM is a parameter estimation problem for $x_{0:T}$ and m. We are given the IMU data $\mathbf{u}_{0:T-1}$, and feature observations $\mathbf{z}_{0:T}$, we need to maximize the data likelihood conditioned on the parameters (MLE) or the posterior likelihood of the parameters given the data (MAP) or use Bayesian Inference to maintain the posterior likelihood of the parameters given.

The following section, gives a detailed explanation of solutions to the above stated problems.

III. Technical Approach

Visual-inertial SLAM for a vehicle using IMU and Stereo camera sensors can be implemented through the following steps:

A. Data
B. IMU Localization
C. Landmark Mapping
D. Visual-inertial SLAM

First I setup an python virtual environment and installed all the required packages such as;

- scipy
- matplotlib
- numpy

A. Data

The following data are given to us,

1) Inertial Measurement Unit: An inertial measurement unit (IMU) is device which measures and body's specific force, angular rate, orientation of the body etc., using accelerometers, gyroscopes, etc. The accelerometer and gyroscope data from an IMU attached to the vehicle is given in the form of linear velocity and angular velocity.

From this data we get the control input $\mathbf{u}_t = [\ \mathbf{v}_t, \omega_t]\ ^T$, where $\mathbf{v}_t$ is the linear velocity and ω_t is the angular velocity. We use this control input for finding the motion model and perform EKF predict step and find the inverse IMU pose $\mathbf{U}_t$.

2) Stereo Camera: A stereo camera has two or more lenses with a separate image sensor or film frame for each lens. This allows it to employ the same principle as our vision system, and therefore gives the ability to capture three-dimensional images. If we know the calibration matrix M which gives the distance per pixel and the baseline the transformation between the two stereo cameras is only a displacement along the x-axis (optical frame) of size b, we can determine the depth of a point from a single stereo observation.

From this data we are given feature observations $\mathbf{z}_t = [\ u_L, v_L, u_R, v_R]\ ^T$ corresponds to the world coordinates of a point landmark $\mathbf{m}$. We are also given K-the left camera intrinsic matrix, b-stereo camera baseline and ${}_{cam}T_{imu}$-extrinsic matrix from IMU to left camera in $SE(3)$.

$$K = \begin{bmatrix} fs_u & 0 & c_u \\ 0 & fs_v & c_v \\ 0 & 0 & 1 \end{bmatrix}$$
$${}_oT_i =_{cam} T_{imu} = \begin{bmatrix} 0 & -1 & 0 & t1 \\ 0 & 0 & -1 & t2 \\ 1 & 0 & 0 & t3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

From K we can calculate stereo calibration matrix M given by:

$$M = \begin{bmatrix} fs_u & 0 & c_u & 0 \\ 0 & fs_v & c_v & 0 \\ fs_u & 0 & c_u & -fs_ub \\ 0 & fs_v & c_v & 0 \end{bmatrix} \tag{4}$$

$$\begin{bmatrix} u_L \\ v_L \\ u_R \\ v_R \end{bmatrix} = \begin{bmatrix} fs_u & 0 & c_u & 0 \\ 0 & fs_v & c_v & 0 \\ fs_u & 0 & c_u & -fs_ub \\ 0 & fs_v & c_v & 0 \end{bmatrix} \frac{1}{z} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{5}$$

From the stereo setup, two rows of M are identical and the vertical coordinates of the two pixel observations are always the same because the epipolar lines in the stereo configuration are horizontal. So we can transform the above equation to:

$$\begin{bmatrix} u_L \\ v_L \\ d \end{bmatrix} = \begin{bmatrix} fs_u & 0 & c_u & 0 \\ 0 & fs_v & c_v & 0 \\ fs_u & 0 & c_u & fs_ub \end{bmatrix} \frac{1}{z} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{6}$$

$$d = u_L - u_R = \frac{1}{z} fs_ub$$
$$\Rightarrow z = \frac{fs_ub}{d} \tag{7}$$

From the above equations we can transform the information from our sensors into world coordinates. We now need to convert the world coordinates to map coordinates.

$$\begin{bmatrix} m_x \\ m_y \\ m_z \\ 1 \end{bmatrix} = ({}_oT_iU_t)^{-1} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{8}$$

We use this data for finding the observation model and perform EKF update step and find μ_{t+1} and Σ_{t+1} from prior $\mu_t \in \mathbb{R}^{3M}$ and $\Sigma_t \in \mathbb{R}^{3M \times 3M}$.

B. IMU Localization

We consider only the localization problem, and define the motion model using pose kinematics instead of dynamic equations. The motion model with time discretization τ and noise $\mathbf{w}_t \sim \mathcal{N}(0, W)$ is given as,

$$\mathbf{U}_{t+1} = exp(-\tau(\mathbf{u}_t + \mathbf{w}_t)^\wedge)\mathbf{U}_t \tag{9}$$

Pose Kinematics with perturbation in discrete time can be written as,

$$\mu_{t+1} = exp(-\tau\hat{\mathbf{u}}_{\mathbf{t}})\mu_t$$
$$\delta\mu_{t+1} = exp(-\tau\mathbf{u}_{\mathbf{t}}^{\curlywedge})\delta\mu_t + \mathbf{w}_t \tag{10}$$

EKF Predict step equations are,

$$\begin{aligned}\boldsymbol{\mu}_{t+1|t} &= \exp\left(-\tau\hat{\mathbf{u}}_t\right)\boldsymbol{\mu}_{t|t} \\ \Sigma_{t+1|t} &= \mathbb{E}\left[\delta\boldsymbol{\mu}_{t+1|t}\delta\boldsymbol{\mu}_{t+1|t}^T\right] \\ &= \exp\left(-\tau\mathbf{u}_{\mathbf{t}}^{\curlywedge}\right)\Sigma_{t|t}\exp\left(-\tau\mathbf{u}_{\mathbf{t}}^{\curlywedge}\right)^T + W\end{aligned} \tag{11}$$

where $\mathbf{u}_t = [\ \mathbf{v}_t, \omega_t]\ ^T$ and

$$\hat{\mathbf{u}}_t = \begin{bmatrix} \hat{\omega}_t & \mathbf{v}_t \\ \mathbf{0}^\top & 0 \end{bmatrix} \in \mathbb{R}^{4\times4} \quad \mathbf{u}_{\mathbf{t}}^{\curlywedge} = \begin{bmatrix} \hat{\omega}_t & \hat{\mathbf{v}}_t \\ 0 & \hat{\omega}_t \end{bmatrix} \in \mathbb{R}^{6\times6} \tag{12}$$

C. Landmark Mapping

There are various methods of representing the map like Landmark-based, Occupancy grid, Surfels, Polygonal mesh etc. In this project, we use Landmark Mapping. It can implemented using a collection of objects, each having a position, orientation, and an object class.

We are given the data association $\pi_t : \{1, ..., M\} \rightarrow \{1, ..., N_t\}$ stipulating which landmarks were observed at each time t is provided by an external algorithm. Also, we consider the landmarks to be static. So, we don't consider a motion model nor do a prediction step for the point landmarks.

We model the landmarks with matrix of mean $\mu_t \in \mathbb{R}^{3M}$ and covariance $\Sigma_t \in \mathbb{R}^{3M\times3M}$. We create a observed index for each timestep, i.e., the index of the pixel observed at that timestep. We do this by checking if the pixel value is not equal to -1. Now, we loop through the observed index and initialize the mean only if the prior mean is (0,0,0) and initialize the map coordinates found using Equations 6, 7 and 8. When a landmark was previously observed, we now update the corresponding mean and covariances using the EKF update equations.

Given a prior Gaussian distribution for $\mathbf{m} \mid \mathbf{z}_{0:t}$, we can find the observation model with observation noise $\mathbf{v}_{t,i} \sim \mathcal{N}(0, V)$.

$$\begin{aligned}\mathbf{z}_{t,i} &= h(\mathbf{U}_t, \mathbf{m}_j) + \mathbf{v}_{t,i} \\ h(\mathbf{U}_t, \mathbf{m}_j) &= M\pi({}_o\mathbf{T}_I\mathbf{U}_t\underline{\mathbf{m}_j})\end{aligned} \tag{13}$$

We can see that the observation model is non-linear. So as to use the EKF equations we need linearize the observation model. To find the Jacobian matrix H_t, we need to find the projection function and its derivative.

$$\begin{aligned}\pi(\mathbf{q}) &= \frac{1}{q_3}\mathbf{q} \in \mathbb{R}^4 \\ \frac{d\pi}{d\mathbf{q}}(\mathbf{q}) &= \frac{1}{q_3}\begin{bmatrix} 1 & 0 & -\frac{q_1}{q_3} & 0 \\ 0 & 1 & -\frac{q_2}{q_3} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{q_4}{q_3} & 1 \end{bmatrix} \in \mathbb{R}^{4\times4}\end{aligned} \tag{14}$$

We need the observation model Jacobian $H_t \in \mathbb{R}^{4N_t\times3M}$ evaluated at μ_t. Let the elements of $H_t \in \mathbb{R}^{4N_t\times3M}$ corresponding to different observations i and different landmarks j be $H_{t,i,j} \in \mathbb{R}^{4\times3}$

$$H_{t,i,j} = \begin{cases} M\frac{d\pi}{d\mathbf{q}}\left(oT_IU_t\underline{\boldsymbol{\mu}}_{t,j}\right)oT_IU_tP^\top & \text{if observation } i \text{ corresponds} \\ & \text{to landmark } j \text{ at time } t \\ \mathbf{0} \in \mathbb{R}^{4\times3} & \text{otherwise} \end{cases} \tag{15}$$

All the feature observations and predicted feature observations are stacked as a $4N_t$ vector at time t.

$$\tilde{\mathbf{z}}_t = M\pi\left(oT_i\mathbf{U}_t\underline{\boldsymbol{\mu}_t}\right) \tag{16}$$

Now, we can perform the EKF update step,

$$\begin{aligned}K_t &= \Sigma_tH_t^\top\left(H_t\Sigma_tH_t^\top + I\otimes V\right)^{-1} \\ \boldsymbol{\mu}_{t+1} &= \boldsymbol{\mu}_t + K_t\left(\mathbf{z}_t - \tilde{\mathbf{z}}_t\right) \\ \Sigma_{t+1} &= \left(I - K_tH_t\right)\Sigma_t\end{aligned} \tag{17}$$

D. Visual-inertial SLAM

Now, we combine the IMU Localization and Landmark Mapping to obtain a complete visual-inertial SLAM algorithm.

Given a prior Gaussian distribution for $\mathbf{U}_{t+1} \mid \mathbf{z}_{0:t}, \mathbf{u}_{0:t} \sim \mathcal{N}(\boldsymbol{\mu}_{t+1|t}, \boldsymbol{\Sigma}_{t+1|t})$, we can find the observation model with observation noise $\mathbf{v}_{t,i} \sim \mathcal{N}(0, V)$.

$$\begin{aligned}\mathbf{z}_{t+1,i} &= h(\mathbf{U}_{t+1}, \mathbf{m}_j) + \mathbf{v}_{t+1,i} \\ h(\mathbf{U}_{t+1}, \mathbf{m}_j) &= M\pi({}_o\mathbf{T}_I\mathbf{U}_{t+1}\underline{\mathbf{m}_j})\end{aligned} \tag{18}$$

We see that the observation model is the same as the mapping problem but now the variable of interest is the inverse IMU pose $\mathbf{U}_{t+1} \in SE(3)$ instead of the point landmark positions $\mathbf{m} \in \mathbb{R}^{3\times M}$.

We need the observation model Jacobian $H_{t+1|t} \in \mathbb{R}^{4N_t\times6}$ with respect to the inverse IMU pose $\mathbf{U}_t$, evaluated at $\boldsymbol{\mu}_{t+1|t}$.

$$\tilde{\mathbf{z}}_{t+1,i} = M\pi\left(oT_i\boldsymbol{\mu}_{t+1|t}\mathbf{m}_j\right) \tag{19}$$

$$H_{i,t+1|t} = M\frac{d\pi}{d\mathbf{q}}\left(oT_I\boldsymbol{\mu}_{t+1|t}\mathbf{m}_j\right)oT_I\left(\boldsymbol{\mu}_{t+1|t}\mathbf{m}_j\right)^{\odot} \in \mathbb{R}^{4\times6} \tag{20}$$

Now, perform the EKF Update,

$$\begin{aligned}K_{t+1|t} &= \Sigma_{t+1|t}H_{t+1|t}^\top\left(H_{t+1|t}\Sigma_{t+1|t}H_{t+1|t}^\top + I\otimes V\right)^{-1} \\ \boldsymbol{\mu}_{t+1|t+1} &= \exp\left(\left(K_{t+1|t}\left(\mathbf{z}_{t+1} - \tilde{\mathbf{z}}_{t+1}\right)\right)^\wedge\right)\boldsymbol{\mu}_{t+1|t} \\ \Sigma_{t+1|t+1} &= \left(I - K_{t+1|t}H_{t+1|t}\right)\Sigma_{t+1|t}\end{aligned} \tag{21}$$

IV. Results

A. Dead Reckoning

To compare the results of the SLAM, we first do dead reckoning on all the datasets and use the images to compare the results of the SLAM. The trajectories almost matched that of the vehicles path in the video given to us.

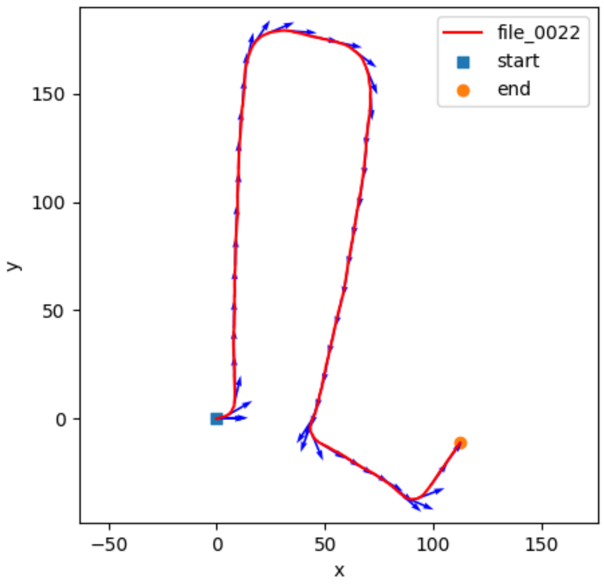

Fig. 1. Trajectory for dataset0022

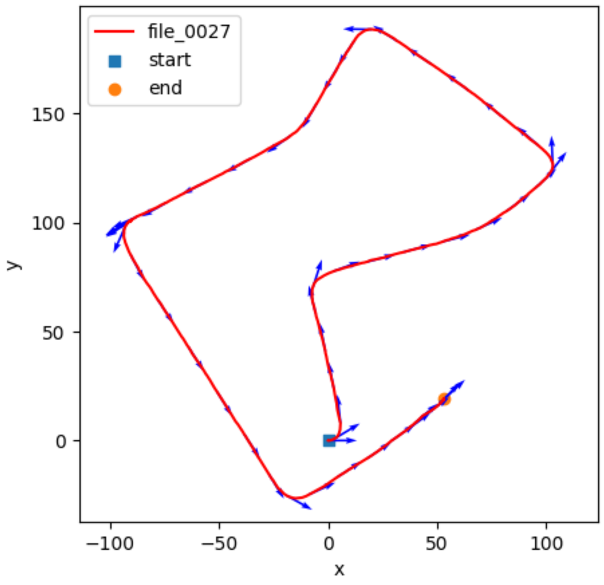

Fig. 2. Trajectory for dataset0027

B. *Visual Mapping*

For performing the mapping part alone, first I downsampled the feature observations to 10% of the observations given to us. By, doing this our computations becomes faster.

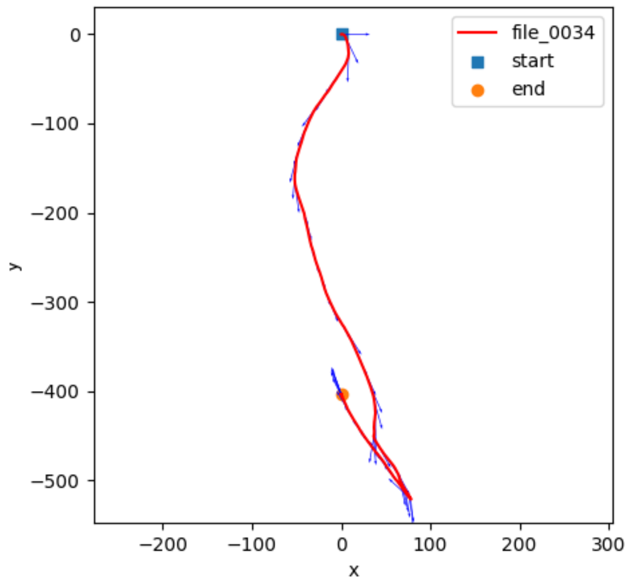

Fig. 3. Trajectory for dataset0034

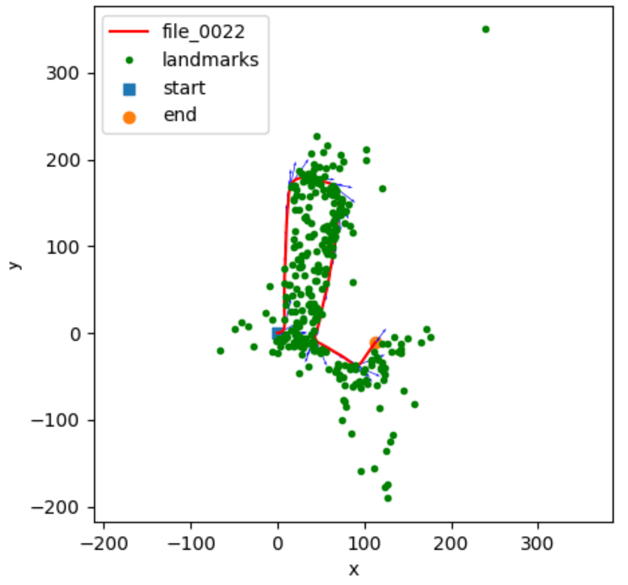

Fig. 4. Visual Map for dataset0022

C. *Visual Inertial SLAM*

I used different values of variances for noises $\mathbf{v}_t$ - V, and $\mathbf{w}_t$ - W, and variance for IMU Σ, and variance for Landmark Σ to compare the accuracy of the SLAM implementation versus the individual parts in part (a) and part (b). The results below are obtained for $W = 0.0001I^{6\times6}, \Sigma_{IMU} = 0.01I^{6\times6}, V = 5000000000, \Sigma_{Landmark} = 300I^{3M+6\times3M+6}$.

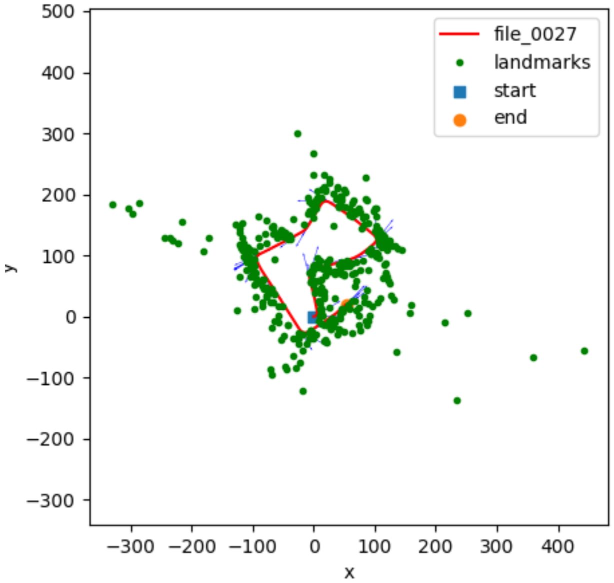

Fig. 5. Visual Map for dataset0027

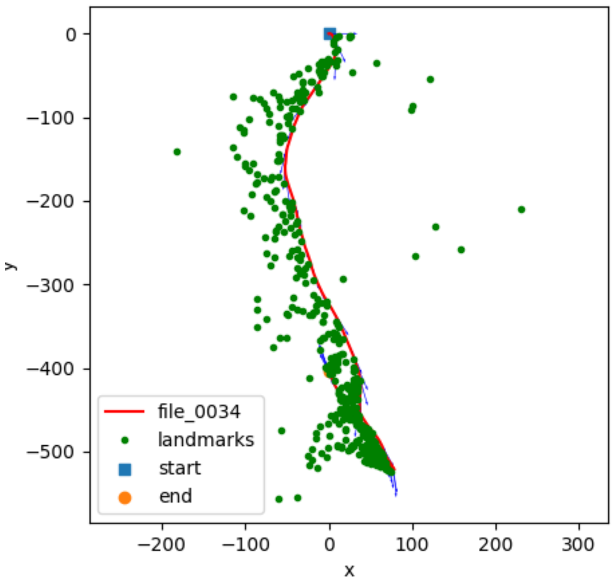

Fig. 6. Visual Map for dataset0034

We can see from the videos for dataset 0027 should do a Loop closure, for dataset 0022 it should be follow a parallel path and for dataset 0034 it should be a straight line with minor turns. I changed by variables values to achieve this and the closest I got is this result.

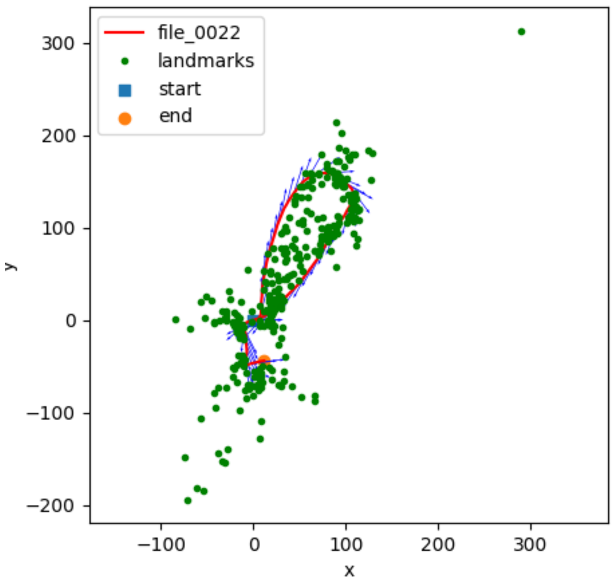

Fig. 7. SLAM for dataset0022

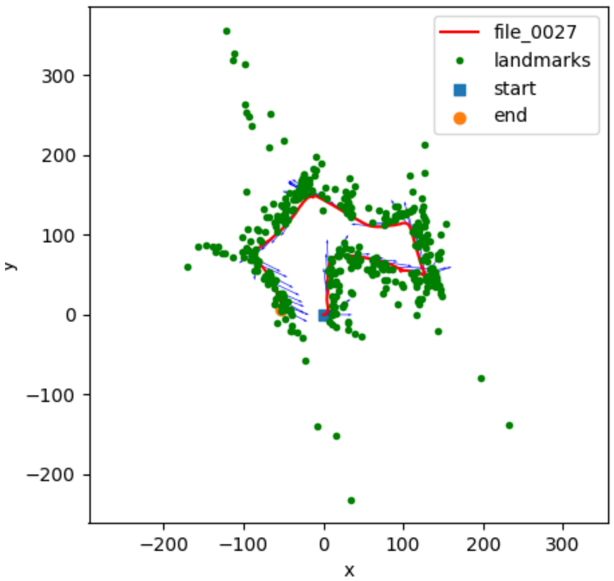

Fig. 8. SLAM for dataset0027

ACKNOWLEDGMENT

I wish to acknowledge here that I collaborated with the following students while working on the project.

- Siddarth Meenakshi Sundaram
- Venkatesh Prasad Venkataramanan

We dicussed High-Level implementation about project and also discussed about the outline of the report.

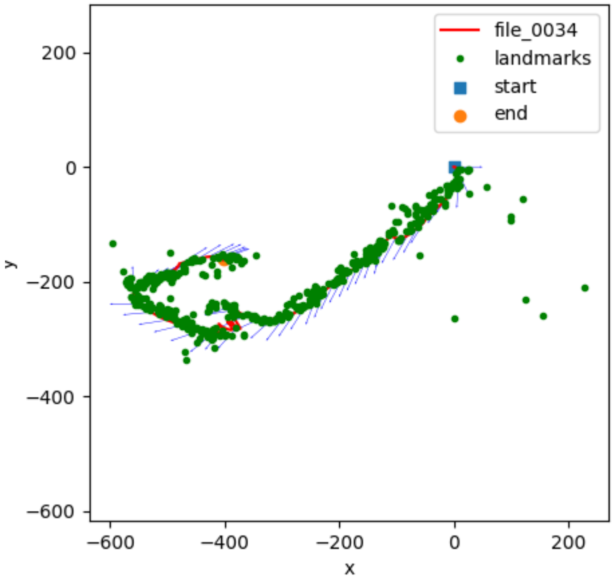

Fig. 9. SLAM for dataset0034