Visual Inertial Simultaneous Localization and Mapping using Extended Kalman Filter

Chhabra Rachit (rachhabra@ucsd.edu)
Department of Mechanical and Aerospace Engineering, University of California San Diego

I. Introduction

The goal of this project is to create a particle filter simultaneous localization and mapping model using data from an autonomous vehicle. SLAM, as commonly known in the industry, is a mathematical problem of building a map or an unknown environment, and at the same time track the position of an agent (a vehicle in this problem) in this space. SLAM is used in a variety of applications such as autonomous vehicles, virtual and augmented reality applications, etc. SLAM is a very computation heavy algorithm and scales exponentially with the increase in landmark features around the agent. In this project, using the data provided first IMU localization via EKF is done. Next, using stereo camera data provided and the stereo camera model, Landmark Mapping via EKF Update is done. And finally, the two parts are combined, and the Visual Inertial SLAM is carried out. The final output is the updated trajectory and features map of the vehicle and its environment, and it is finally plotted.

II. Problem Formulation

We are provided with the following data from the vehicle:

(i) Angular Velocity, v^t and Linear Velocity of the vehicle in the body frame.
(ii) Timestamps τ_t UNIX standard seconds-since-the-epoch January 1, 1970.
(iii) Observation data from the Stereo Camera images from the two cameras.
(iv) The intrinsic calibration parameters for the stereo camera.
(v) Transformation matrices ${}_IT_c \in SE(3)$ to compute a pose from left camera frame to the world frame.

Using the given Initial Data, linear velocities v_t, and the angular velocity ω_t provided by the IMU sensor, and the landmark locations from the stereo camera, we need to estimate:

(i) Trajectory of vehicle at each time step τ (EKF Prediction Step) using the $SE(3)$ kinematics of the IMU over time t - Motion Model:

$$\mu_{t+1|t} = \mu_{t|t} \exp\left(\tau_t \widehat{u_t}\right) + \text{Perturbations}$$

where $\mu_{t+1|t}$ is the position of the vehicle in world frame coordinates time t , u_t is the control being applied on the particle (velocity and the angular velocity from the IMU sensor).

(ii) The position of the landmarks m_i around the vehicle in the surrounding environment using the pixel coordinates $z_{t+1|i}$ of the detected visual features i from the stereo camera data assuming that the predicted IMU trajectory from part (a) above is correct.
Observation Model:

$$\begin{aligned} z_{t+1|i} &= h(T_{t+1}\boldsymbol{m}_j) + \boldsymbol{v}_{t+1,i} \\ &= K_s \pi\left({}_OT_I T_{t+1}^{-1} \underline{m_j}\right) + \boldsymbol{v}_{t+1,i} \end{aligned}$$

where K_s is the intrinsic parameter matrix, ${}_OT_I$ is t camera to IMU transformation matrix, $\underline{m_j}$ is the homogeneous coordinates in the world frame of the visible features, and $\boldsymbol{v}_{t+1}$ is the measurement noise at time t.

Finally, the above two things, localization and mapping need to be done simultaneously assuming neither the map not the trajectory is given. Given the IMU prediction step from part (a) with the landmark update step from part (b), we implement an IMU update step based on the stereo-camera observation model to obtain a complete visual-inertial SLAM algorithm.

III. TECHNICAL APPROACH

In this section. we discuss the methods and algorithms used in constructing different parts of the project, IMU localization via EKF, Landmark Mapping via EKF, and Visual Inertial SLAM.

In this project, we did not need to synchronize the data before using as all the data was already synchronous. We were given with two data sets, '03.npz' which had 1023 timestamps and 5105 features, and '10.npz', which had 3342 timestamps and 13289 features. First of all, we transform the stereo to IMU transformation matrix, to reverse the direction of z - axis of the IMU. This is done using the following transformation.

$$_{I}T_{C,new} = Rot_x(\pi)\ _{I}T_C$$

where $Rot_x(\pi) \in SE(3)$ is the rotation of 180° about positive z axis.

Extended Kalman Filter is a type of Bayes filter which has the following assumptions:

- The motion model is nonlinear with noise $\boldsymbol{v}_t \sim \mathcal{N}(0, V)$
- The observation model is nonlinear with noise $\boldsymbol{w}_t \sim \mathcal{N}(0, W)$
- $\boldsymbol{v}_t$ and $\boldsymbol{w}_t$ are independent of each other.
- The prior pdf $p_{t|t}$ is Gaussian.

IMU localization via EKF Prediction

The first step in the project is to carry out this part. We need to find the position and control for the vehicle at every timestep. According to the motion model mentioned in the previous section, we find the mean μ_{imu} as each time step, and update the pose, and covariance of the pose of the vehicle using the equations:

$$\mu_{imu,t+1|t} = \mu_{imu,t|t} \exp\left(\tau_t \widehat{u_t}\right)$$

$$\Sigma_{t+1|t} = \exp(-\tau \breve{\boldsymbol{u}}_t)\, \Sigma_{t|t} \exp(-\tau \breve{\boldsymbol{u}}_t)^T + W$$

where we take the prior to be for the mean to be 4x4 $SE(3)$ matrix and initializing it to be a zero matrix (Assuming the place from where it starts to be zero), and a 6x6 matrix diagonal matrix for the covariance, $u_t = [v_t \quad \omega_t]^T$ and the hat maps $\hat{\boldsymbol{u}}_t$, $\breve{\boldsymbol{u}}_t$ are:

$$\hat{u}_t := \begin{bmatrix} \widehat{\omega}_t & v_t \\ 0^T & 0 \end{bmatrix} \in \mathbb{R}^{4\times 4}$$

$$\breve{u}_t := \begin{bmatrix} \widehat{\omega}_t & \hat{v}_t \\ 0 & \widehat{\omega}_t \end{bmatrix} \in \mathbb{R}^{6\times 6}$$

The trajectory is calculated with time, using the above equations. Perturbations could also be added to the motion model used, but the effect of a small perturbation was very large and thus it was neglected for this project. Finally, the state poses at each time were sent to the function, visualize_trajectory to visualize the trajectory formed.

Landmark Mapping via EKF Update

For this step, we assume that the pose of the vehicle with time and the trajectory estimated in the first part is correct and using that, we estimate the landmark positions using the features of the image coordinates provided. Firstly, as there is a large amount of features, we first, filter the data and keep only 1 in **skip (**values mentioned at the last**)** features, so that it reduces the time for computation.

a) First step in this is to filter the given data (only take those features which are visible (let N_T) at the current timestep, $z_{L/R} \neq -1$) and convert them to the world coordinates using the inverse camera model:

$$z_t = M\pi\left({\mu_{imu}}^{-1}\ {}_{O}T_{I}\underline{m}\right) + v_t$$

$$z = \frac{f s_u b}{u_L - u_R}$$

$$\begin{bmatrix} u_L \\ v_L \\ u_R \\ v_R \end{bmatrix} = \begin{bmatrix} f s_u & 0 & c_u & 0 \\ 0 & f s_v & c_v & 0 \\ f s_u & 0 & c_u & -f s_u b \\ 0 & f s_v & c_v & 0 \end{bmatrix} \frac{1}{z} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Gaussian noise v_t is also added such that $v_t \sim \mathcal{N}(0, V)$.

b) Whenever we see a feature for the first time, we update its covariance in the covariance matrix by 0.001 x I_{3x3}.

c) Now, according to the EKF Update algorithm, we find $\tilde{\boldsymbol{z}}$ with respect to the world coordinates and then calculate the Jacobian of $\tilde{z}$, H_{stereo}.

$$\tilde{\boldsymbol{z}}_t = M\pi\left({}_{O}T_{I}\ {\mu_{imu}}^{-1}\underline{\mu_{m,t}}\right)$$

$$H_m = \begin{cases} M \dfrac{d\pi}{dq}\left({}_O T_I \ {\mu_{imu}}^{-1} \underline{\mu_{m,t}} \right)\left({}_O T_I \ {\mu_{imu}}^{-1} P^T \right) ; \ \Delta_t = i \\ 0 \ ; otherwise \end{cases}$$

where $\mu_{m,t}$ is the mean of the visible features at time t.

d) Finally, we calculate the Kalman Gain at each time step using $\tilde{z}$ and H. Finally, the mean and covariance of each visible feature is updated.

$$K_{t+1} = \Sigma_t H_{t+1}^T (H_{t+1} \Sigma_t H_{t+1}^T + I \otimes V)^{-1}$$

$$\mu_{m,t+1|t} + K_{t+1|t}(z_t - \tilde{z}_t)$$

$$\Sigma_{t+1} = (I - K_{t+1} H_{t+1}) \Sigma_t$$

Visual Inertial SLAM

As discussed earlier, in this step, we simultaneously update the pose of the vehicle as well as the landmark positions of the immediate environment. Here we combine the mean of both the IMU state and the stereo features into one matrix. Similarly, the covariance matrix is also stacked into a $(3M + 6)$ x $(3M + 6)$ matrix.

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

$$\Sigma \in \mathbb{R}^{(3M+6) \times (3M+6)}$$

a) Similar to the update step in previous part, we calculate the N_T, z_t, and then along with calculating the H_{stereo}, we also calculate the H_{imu} using the equation:

$$H_{imu} = -M \frac{d\pi}{dq}\left({}_O T_I \, {\mu_{imu}}^{-1} \underline{\mu_{m,t}} \right) {}_O T_I \, ({\mu_{imu}}^{-1} \underline{\mu_{m,t}})^{\odot}$$

and then the two H matrices are stacked together such that,

$$H_{slam} = [H_{stereo} \quad H_{imu}] \in \mathbb{R}^{4N_t \times (3M+6)}$$

b) Now, using this combined Jacobian H_{slam}, we calculate the Kalman Gain:

$$K_{t+1|t} = \Sigma_{t+1|t} H_{t+1|t}^T (H_{t+1|t} \Sigma_{t+1|t} H_{t+1|t}^T + I \otimes V)^{-1}$$

c) Now, finally, we update the covariance matrix and the both the means μ_{imu} and μ_{stereo}

$$\begin{bmatrix} \mu_{imu} \\ \mu_{stereo} \end{bmatrix} = \begin{bmatrix} \mu_{imu,t+1|t} \\ \mu_{stereo,t|t} \exp(-\tau \hat{u}_t) \end{bmatrix}$$

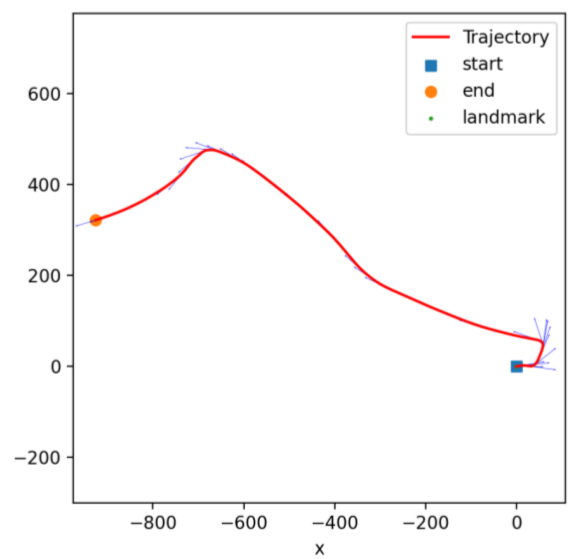
$$\Sigma_{t+1|t+1} = (I - K_{t+1|t} H_{t+1|t}) \Sigma_{t+1|t} (I - K_{t+1|t} H_{t+1|t})^T + K_{t+1|t} V K_{t+1|t}^T$$

This is calculated over all time steps, and noise for both the motion model and the observation model is changed to find the best value for the hyperparameters.

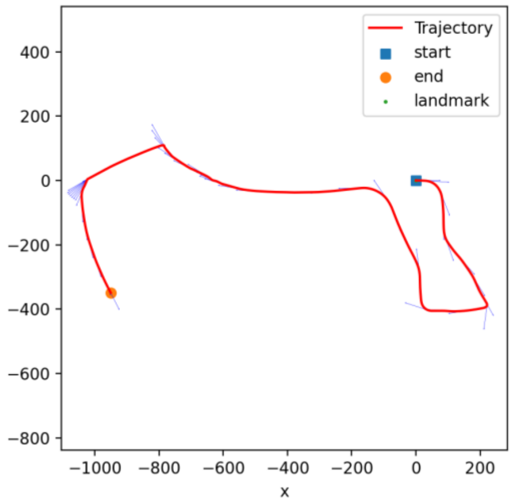
IV. RESULTS

After we have used the Extended Kalman Filter to implement the SLAM algorithm here we discuss the results of the all the three parts mentioned in the Introduction, and how the algorithm performed on two sets of real-life vehicle data.

The path of the vehicle in world frame can be visualized by the following graph.



IMU based Localization - '03.npz'

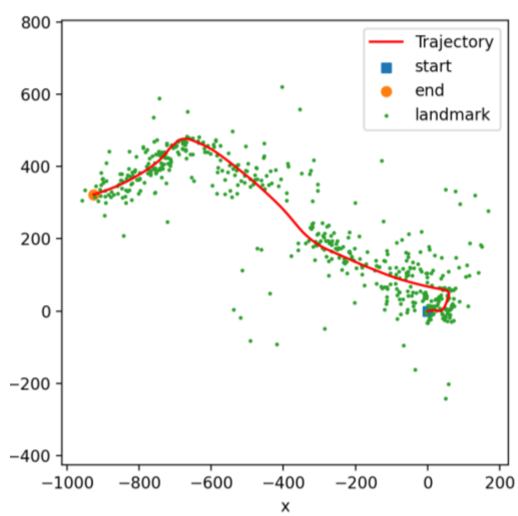


IMU based Localization - ’10.npz’

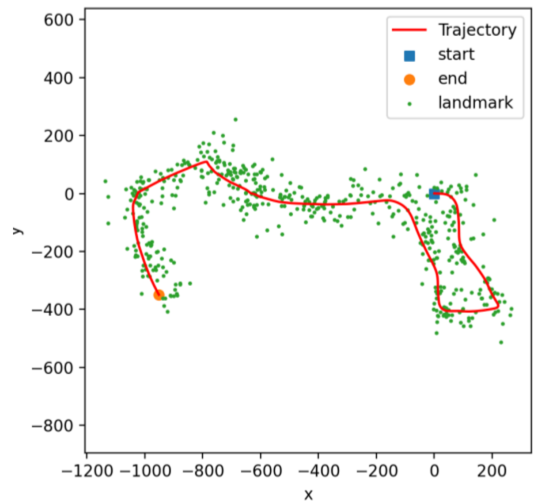
We can see using the IMU data, the trajectory of the vehicle in both data ’03.npz’ and ’10.npz’ is pretty similar to the stereo images from the camera provided to us. Thus, comparing the two, it can be said that the algorithm for IMU Localization works correctly.

Landmark Mapping using EKF Update

Now considering the trajectory from part (a), the landmark mapping of the features is estimated and shown as follows:



Landmark Mapping - ’03.npz’

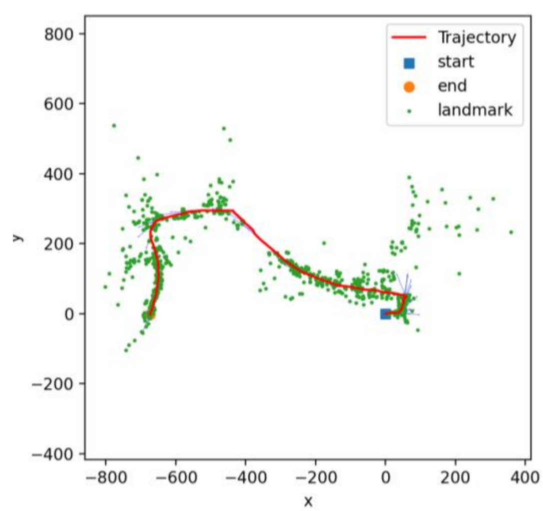


Landmark Mapping - ’10.npz’

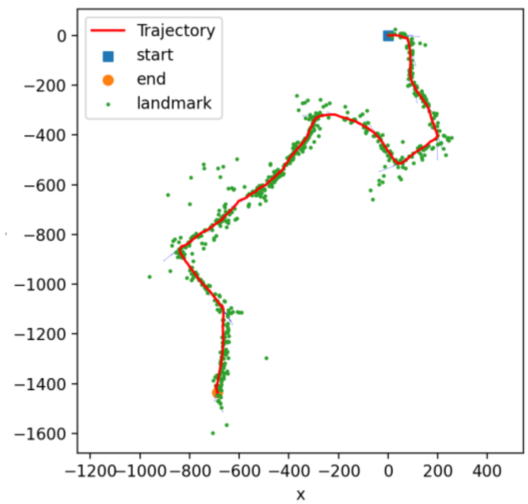
The shape of the trajectory is similar to the trajectory as part (a) as there was no update the position of the vehicle. Also, it is observed that as the noise in the features measurements increases, the landmarks become more and more spread out, and away from the trajectory, which sounds logical. Around one in every 4 features were used (around 1250 features), to reduce the computation time.

Visual Inertial SLAM

Finally, both the prediction and landmark mapping step were combined, and below are the resulting simultaneous localization of the vehicle and mapping of the environment.



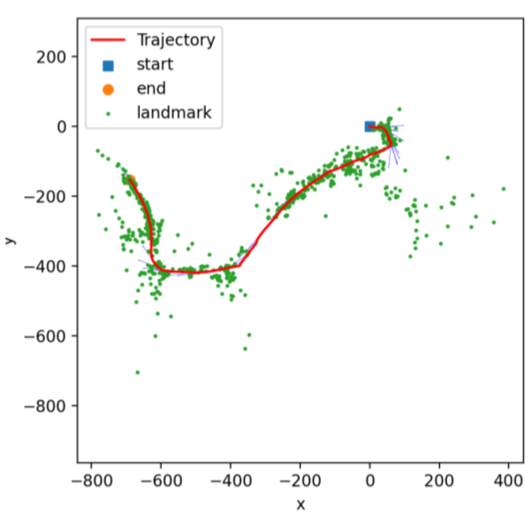
Visual Inertial SLAM - ’03.npz’
No. of features = 1 in every 6



Visual Inertial SLAM - '10.npz'
No. of features = 1 in every 10

Here we can see that the trajectory this time is different from the one in first part. This is because as we move along in time, the location is simultaneously corrected by the algorithm. It should also be noted that, as we increase the number of features, the computation time also increases exponentially. This is because while calculating, the Kalman Gain, we need to take an inverse of the 3m x 3m matrix which takes a lot of computation.

Also, the camera to IMU transformation matrix provided initially had the z direction of the IMU in the opposite direction and thus, the output was also inverted as shown below.



Visual Inertial SLAM - '03.npz' - Uncorrected

To correct this, the IMU pose was pre multiplied with an SE(3) matrix with rotation of 180 degrees about positive x axis and keeping the displacement to be zero.

In this project, the following areas were explored but could not make into the final report. More work can be done in reducing the computation time, by using different methods to calculate the inverse of the high dimension matrix.

Description	**Parameter**	**Initialized value**
No. of values skipped per one feature	**skip**	6 - '03.npz' 10 - '10.npz '
Measurement Noise Covariance	**V**	0.01 x I_{4x4}
State Noise Covariance	**W**	$diag$ (0.2,0.2,0.2,0.01,0.01,0.01)
Prior Landmark Covariance	Σ_{stereo}	0.001 for any feature seen for the first time
Prior State Covariance	Σ_{imu}	0.001 x I_{6x6}

Hyper-parameters for Visual Inertial SLAM