SLAM using Kalman Filter

Khushboo Agrawal
PID: A53271205 (March'19)
University of California San Diego
khagrawa@ucsd.edu

***Abstract*— This report is a project overview of a course project in which we aim to implement SLAM algorithm using an extended Kalman filter to create a 2D map of the environment with the help of the IMU sensor data and stereo camera data. The algorithm is implemented on 3 datasets and the results are presented in the report.**

I. INTRODUCTION

Using robots in the present scenario not only to make human lives simpler but also to automate things have become an essential part of the technological progress. To explore this genre of robotics where robots are used to track and plot the environment have been a pioneer problem statement. To name a few of its interesting applications includes surveillance, crop detection, indoor fire-saving etc. Algorithms such as Visual SLAM based algorithm are able to simultaneously build 3D maps of the world while tracking the location and its orientation. SLAM algorithms can also be clubbed with deep learning for perception recognition for gradient based geometry detection problems and build Convolution Neural Network architecture to learn features such as in crop detection based on plant color. One such attempt to understand this and be able to solve some of the complex yet important applications in future, this project is the start of these problems where we implement simultaneous localization and mapping using extended kalman filter and plot 2D map of the environment by showing the landmarks and the robot trajectory. For this problem, a car is mounted by a camera and IMU and the stereo camera as well as the IMU data are collected. The report is formulated in three different sections namely Problem Formulation, Technical Approach and Results which provide an overview of the project.

II. PROBLEM FORMULATION

The problem of SLAM (Simultaneous Localization and Mapping) is framed as mapping given a robot trajectory and given the observation of the environment, localize the robot. The implementation is carried out by implementing the Extended Kalman Filter (EKF). To implement the EKF, we assume that the noise and probability are independent and Gaussian. The problem is formulated in two parts namely sensor and data initialization, Extended Kalman Filter Algorithm.

A. Sensor and Data Initialization

- In this part of the problem, we load the data and return the timestamps, features, linear velocity, rotational velocity, intrinsic matrix, b and extrinsic matrix.
- Given a world coordinate of point m [x,y,z] as $[u_L, v_L, u_R, v_R]$, we write the stereo camera matrix for the transformation as:

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

From the given matrix, we can calculate the depth as:

$$d = u_L - u_R = fs_u b/z \tag{2}$$

$$z = fs_u b/u_L - u_R \tag{3}$$

With the help of the following equations, we can calculate actual depth from the disparity leading us to a rich texture combined depth sensor that can simply be implemented by two cameras.

B. Extended Kalman Filter Algorithm

We define the implementation of Extended Kalman Filter Algorithm to plot the 2D map as follows:

$$x_{t+1} = f(x_t, u_t, w_t) \tag{4}$$

where $x_t \sim \mathcal{N}(\mu_{t|t}, \sum_{t|t})$ and $(w_t \sim \mathcal{N}(0, W))$.

$$f(x_t, u_t, w_t) \approx f(x_t, u_t, 0) + F(x_t - \mu_{t|t}) + Q(w_t) \tag{5}$$

where F= $\frac{df(x_t,u_t,0)}{dx}$ and Q= $\frac{df(x_t,u_t,0)}{dw}$. As the noise and state probability distribution is independent and gaussian, we write as:

$$\mu_{t+1|t} = f(\mu_{t|t}, \mu_t, 0) \tag{6}$$

$$\Sigma_{t+1|t} = F_t \Sigma_{t|t} F_t^T + Q_t W Q_t^T \tag{7}$$

From the observation model we write,

$$z_{t+1} = h(x_{t+1}, v_{t+1}) \tag{8}$$

where $v_t \sim \mathcal{N}(0, V)$. From taylor approximation we get:

$$h(x_{t+1}, v_{t+1}) \approx h(\mu_{t+1|t}, 0) + H(x_{t+1} - \mu_{t+1|t}) + R(v_{t+1}) \tag{9}$$

where $H = \frac{dh(\mu_{t+1},0)}{dx}$ and $R = \frac{dh(\mu_{t+1},0)}{dv}$.

$$\mu_{t+1|t+1} = \mu_{t+1|t} + K_{t+1|t}(z_{t+1} - m_{t+1|t}) \tag{10}$$

$$\Sigma_{t+1|t=1} = \Sigma_{t+1|t} K_{t+1|t} S_{t+1|t} K_{t+1|t}^T \tag{11}$$

$$K_{t+1|t} = C_{t+1|t} S_{t+1|t}^{-1} \tag{12}$$

$$m_{t+1|t} \approx h(\mu_{t+1|t}, 0) \tag{13}$$

$$S_{t+1|t} \approx H_{t+1} \Sigma_{t+1|t} H_{t+1}^T + R_{t+1} V_{t+1} R_{t+1}^T \tag{14}$$

$$C_{t+1|t} \approx \Sigma_{t+1|t} H_{t+1}^T \tag{15}$$

III. Technical Approach

This section of the report discusses the technical aspects of each part of the problem and the way it was implemented.

A. Dataset

The data is loaded and it consists of two sensor data and parameters of the robot with respect to the environment:

- IMU sensor: Accelerator and gyroscope data in form of angular and linear velocities.
- Camera data giving unique set of features.
- Time stamps of the IMU readings.
- Camera Intrinsic matrix
- Stereo camera baseline
- Extrinsic matrix from IMU to left camera

B. Prediction/Localization

In this part of the problem we predict the co variance and mean as follows:

- We consider the discrete version of the motion model. To predict we use the exponential map where $\hat{u}$ is the skew symmetric velocity vector. By adding small noise w, we get the following predicted mean and co-variance matrices:

$$\mu_{t+1|t} = exp(-\tau\hat{u}\mu_{t|t}) \tag{16}$$

$$\Sigma_{t+1|t} = exp(-\tau\hat{U})\Sigma_{t|t}exp(-\tau\hat{U})^T + W\tau^2 \tag{17}$$

$$\hat{U} = \begin{bmatrix} \hat{w} & \hat{v} \\ 0 & \hat{w} \end{bmatrix} \tag{18}$$

C. Update/Mapping

The mapping is carried out in the following steps:

- In this step, we model landmark as being part of the gaussian distribution of mean and co-variance. We initialize the mean as 0, and the visited landmark as -1. We then search for new landmarks by checking the above criterion and then covert the fearture pixel $[u_L, v_L, u_R, v_R]$ to the world frame and find the disparity by taking the difference in the pixel location.
- We get the projection in the world frame by implementing the following transformation:

$$[x_w, y_w, z_w] = H_{cw}^{-1} R_{oc}^{-} 1 Z_o K^{-1}(u_l, v_l) \tag{19}$$

where K is the intrinsic matrix of camera, z_o is the optical axis, R_{oc} is the transformation from camera to the optical frame and H_{cw} is the transformation from world to camera frame.
- For the landmarks which are seen before, we update by following steps:

$$H_t = M\frac{d\pi}{dq}(T_{oi}\mu_{tj})T_{oi}D \tag{20}$$

$$K_t = \Sigma_t H_t^T (H_t \Sigma_t H_t^T + V)^{-1} \tag{21}$$

$$\mu_{t+1} = \mu_t + DK_t(z_t - \hat{z_T}) \tag{22}$$

$$\Sigma_{t+1} = (I - K_t H_t)\Sigma_t \tag{23}$$

In the above equations, M is the stereo calibratio matrix, K_t is the kalman gain, D is the dialation matrix, V is the process noise (3000) and z are the new observed features. The values correspond to the wrold coordinate $[x_w, y_w, z_w]$ and the trajectory is plotted with the landmarks discussed in the results section.

IV. Results

1) Dataset1: Figure 1 shows the result of the first dataset. As the video indicates, it was driven on the busy street where the landmarks are close by, therefore considering varied value of noise makes a difference in the updation step. Here the noise is considered as 3000 for the updation step.

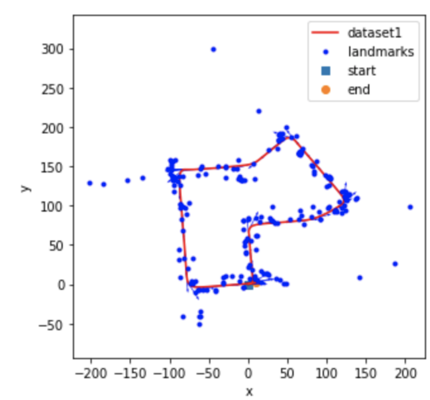

Fig. 1. EKF SLAM for dataset1, red-trajectory, blue- landmarks

2) Dataset2: Figure 2 shows the result of the second dataset. As the video indicates, the car goes straight on highway with relatively less obstacles, therefore varying the noise in the updating step will not make much difference as the obstacles are relatively far than the previous case.

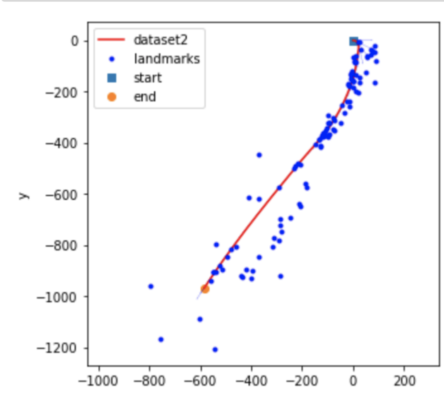

Fig. 2. EKF SLAM for dataset2, red-trajectory, blue- landmarks

A. Testset

Figure 3 shows the result of the testset. As the video indicates, the car take a loop with two u turns and returning to the starting point. Here we observe that the landmarks are close by, therefore changing the value of noise in the sensor data is considered. Noise v in the updation step is 3000.

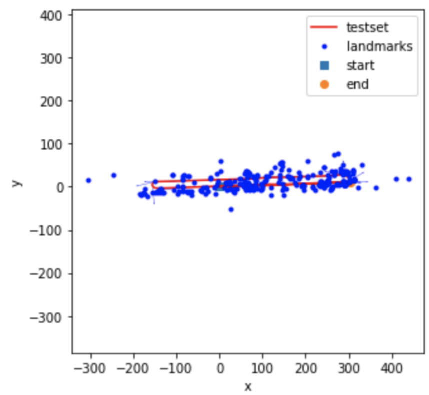

Fig. 3. EKF SLAM for testset, red-trajectory, blue- landmarks

V. Conclusion

From the following implementation, we conclude the following: Fig. 4. landmarks

- The IMU data during the prediction step allows us to predict without much drift as the car come back to its position.
- Updating the trajectory to be synchronous to the landmark observations shows the benefits of the kalman filter working correctly
- We see that the stereo cameras gives a good projection of the depth of landmarks
- We observe that the points spread out as the time increases.

Acknowledgement

The author would like to thank Prof. Nikolay Atanasov for giving the opportunity to learn extended kalman filter through this project and the TA's for their guidance.