Visual-Inertial SLAM

1st Yifan Xu A53253258
The Electrical and Computer Engineering Department UCSD
La Jolla, US
yix247@eng.ucsd.edu

***Abstract*—This paper is a report for the third project in ECE276 Sensing and Estimation in Robotics, UCSD.**

***Index Terms*—SLAM, extended Kalman filter, localization, mapping**

I. Introduction

As the world continues to move deliberately toward a transportation system driven by autonomous vehicles in the purpose of reducing accidents, traffic congestion and freeing people, the technologies to achieve this goal became really important. To make vehicles drive autonomous, an accurate simultaneous localization and mapping (SLAM) method is needed.

In this project, I implemented simultaneous localization and mapping using IMU data and landmarks observation data from a driving vehicle. But the data has noise. In order to get a better result, I used the extended Kalman filter algorithm to find an accurate world position for the driving vehicle.

II. Problem Formulation

The main problem to this SLAM project was to find the most accurate location of landmarks and the driving vehicle with IMU data, landmarks observation data and extended kalman filter(EKF) algorithm.

A. IMU Localization via EKF(Prediction Step)

In this part, μ represent the pose of IMU.

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

where $u_t = \begin{bmatrix} v_t \\ \omega_t \end{bmatrix}$, $\hat{u}_t = \begin{bmatrix} \hat{\omega}_t & v_t \\ 0 & 0 \end{bmatrix}$,

$$\hat{\omega}_t = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}$$

$$\Sigma_{t+1|t} = exp(-\tau \overset{\curlywedge}{u_t})\Sigma_{t|t} exp(-\tau \overset{\curlywedge}{u_t})^T + \tau^2 W$$

where $\overset{\curlywedge}{u_t} = \begin{bmatrix} \hat{\omega}_t & \hat{v}_t \\ 0 & \hat{\omega}_t \end{bmatrix}$

B. Landmark Mapping via the EKF(Update Step)

In this part, μ represent the position of landmarks.

$$\pi(q) = \frac{1}{q_3} q$$

$$\frac{d\pi}{dq}(q) = \frac{1}{q_3} \begin{bmatrix} 1 & 0 & -\frac{q_1}{q_3} & 0 \\ 0 & 1 & -\frac{q_2}{q_3} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{q_4}{q_3} & 1 \end{bmatrix}$$

$$\hat{z}_{t,i} = M\pi({}_O T_I T_t \mu_{t,j})$$

where M is the stereo camera calibration Matrix with intrinsic parameters: $M = \begin{bmatrix} fs_u & 0 & c_u & 0 \\ 0 & fs_v & c_v & 0 \\ fs_u & 0 & c_u & -fs_u b \\ 0 & fs_v & c_v & 0 \end{bmatrix}$

if observation i corresponds to landmark j at time t

$$H_{i,j,t} = M \frac{d\pi}{dq}({}_O T_I T_t \mu_{t,j}) {}_O T_I T_t D$$

else

$$H_{i,j,t} = 0$$

where $D = \begin{bmatrix} I_3 \\ 0^T \end{bmatrix}$

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

where $I \otimes V = \begin{bmatrix} V & & \\ & \ddots & \\ & & V \end{bmatrix}$

$$\mu_{t+1} = \mu_t + DK_t(z_t - \hat{z}_t)$$

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

C. Visual-Inertial SLAM

From this part, we need to combine the IMU prediction step from part with the landmark update step and an IMU update step based on the stereo camera observation model to obtain a complete visual-inertial SLAM algorithm.

Therefore, we need to calculate the jpint distribution:

$$p(x_0 : T, z_0 : T, u_0 : T-1) =$$

$$p_{0|-1}(x_0) \prod_{t=0} p_h(z_t|x_t) \prod_{t=1} p_f(x_t|x_{t-1}, u_{t-1})$$

III. Technical Approach

For this project, we had two tasks. First, IMU localization. Second, landmarks mapping.

A. IMU Localization

- **Extended Kalman Filter Prediction**

  To implement the EKF prediction step, we need to estimate the pose $T_t \in SE(3)$ of the IMU over time t based on the SE(3) kinematics. Since we have the linear velocity and rotational velocity data through time, given a identity matrix as the initial pose, from the equations shown in the last section, we can predict the pose in the next frame. Repeat this process, we can get the robot trajectory.

B. Landmarks Mapping

- **Extended Kalman Filter Update**

  To implement the Landmarks Mapping, first we assume that the predicted IMU trajectory above is correct and focus on estimating the landmark positions. Then we need to implement the EKF Update step. To get the unknown landmark positions, we should see those landmarks positions as a state. Each time when the robot captures a visual observation (z_t in the last section), we keep the track of the mean and covariance of the landmarks positions in order to update and get the new positions. Since we know the robot trajectory, the observation through time, from the equations shown above, we can update the landmarks position.

C. Visual-Inertial SLAM

To implement the Visual-Inertial SLAM, we need to combine the IMU prediction step with the landmark update step and an IMU update step based on the stereo camera observation model to obtain a complete visual-inertial SLAM algorithm.

Predict Step

m : the total number of landmarks

$$\mu^P_{t|t} \in R^{4*4},\ \mu^L_{t|t} \in R^{4*m},$$

$$\Sigma^P_{t|t} \in R^{6*6},\ \Sigma^L_{t|t} \in R^{3m*3m},$$

$$\Sigma_{t|t} = \begin{bmatrix} \Sigma^P_{t|t} & 0 \\ 0 & \Sigma^L_{t|t} \end{bmatrix} \in R^{(6+3m)*(6+3m)}$$

$$\mu^P_{t+1|t} = exp(-\tau \hat{u}_t)\mu^R_{t|t}$$

$$\mu^L_{t+1|t} = I\mu^L_{t+1|t}$$

$$\Sigma_{t+1|t} = \begin{bmatrix} exp(-\tau \overset{\curlywedge}{u}_t) & 0 \\ 0 & I \end{bmatrix} \Sigma_{t|t} \begin{bmatrix} exp(-\tau \overset{\curlywedge}{u}_t) & 0 \\ 0 & I \end{bmatrix}^T + \begin{bmatrix} \tau^2 W & 0 \\ 0 & 0 \end{bmatrix}$$

Update Step

Nt : the number of observed landmarks at time t

$$H^P \in R^{4Nt*6},\ H^L \in R^{4Nt*3m},$$

$$H = [H^P\ H^L] \in R^{4Nt*(6+3m)}$$

$$H^P_{i,t+1|t} = M\frac{d\pi}{dq}({}_OT_I\mu^P_{t+1|t}\mu^L_{j,t+1|1})_OT_I(\mu^P_{t+1|t}\mu^L_{j,t+1|1})^{\odot}$$

where

$$\begin{bmatrix} s \\ \lambda \end{bmatrix}^{\odot} = \begin{bmatrix} \lambda I & -\hat{s} \\ 0 & 0 \end{bmatrix}$$

$$H^P_{t+1|t} = \begin{bmatrix} H^P_{1,t+1|t} \\ \vdots \\ H^P_{N_{t+1}t+1|t} \end{bmatrix}$$

if observation i corresponds to landmark j at time t

$$H^L_{i,j,t+1|t} = M\frac{d\pi}{dq}({}_OT_I\mu^P_{t+1|t}\mu^L_{j,t+1|t})_O\mu^P_{t+1|t}D$$

else

$$H_{i,j,t} = 0$$

where $D = \begin{bmatrix} I_3 \\ 0^T \end{bmatrix}$

$$K_{t+1|t} \in R^{(6+3m)*4N_t}$$

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

where $I \otimes V = \begin{bmatrix} V & & \\ & \ddots & \\ & & V \end{bmatrix}$

$$\hat{z}_{i,t+1} = M\pi({}_OT_IT_t\mu_{j,t+1})$$

$$\mu^P_{t+1|t+1} = exp((K^P_{t+1|t}(z_{t+1} - \hat{z}_{t+1})^{\wedge})\mu^P_{t+1|t}$$

where $\begin{bmatrix} \rho \\ \theta \end{bmatrix}^{\wedge} = \begin{bmatrix} \hat{\theta} & \rho \\ 0 & 0 \end{bmatrix}$

$$\mu^L_{t+1|t+1} = \mu^L_{t+1|t} + DK^L_{t+1|t}(z_{t+1} - \hat{z}_{t+1})$$

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

IV. Results

A. Training Result

1) *IMU Localization:* Results are shown below.

2) *Landmark Mapping:* Results are shown below.

B. Testing Result

Results are shown below.

C. SLAM Result

Results are shown below.

D. Discussion

For the part (a) and (b) in prblem 4, compared the figures to the video footages we can see, the Entended Kalman Filter method successfully obtained the trajectory of the robot and the positions of landmarks on all the datasets. But the accuracy is not very high. Some of the landmarks are far away from the trjectory. Also, the the magnitude of nosie will not has huge influence on the result.

For the part (c), we can get a better result, even though it looks like there is no big difference with the part (a) and (b). But I think this is because the data it's pretty good. So if the data is worse, we may get a worse result on part (a) and (b). The reason why we can get a better result on part (c) is we combine the IMU prediction step with the landmark update step and IMU update step so that we can get the covariance between robot pose and landmarks. Therefore, we get a better result.

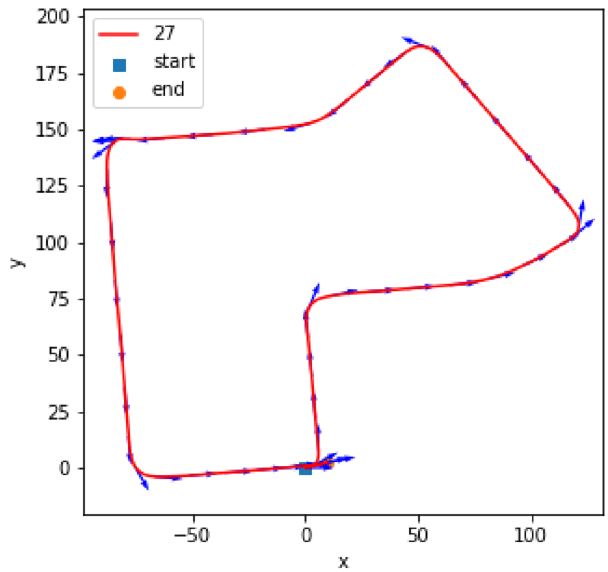

Fig. 1. DataSet 27 IMU Localization

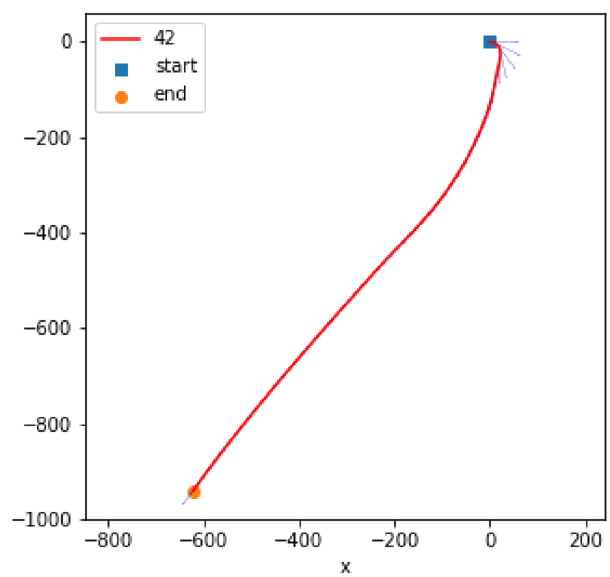

Fig. 2. DataSet 42 IMU Localization

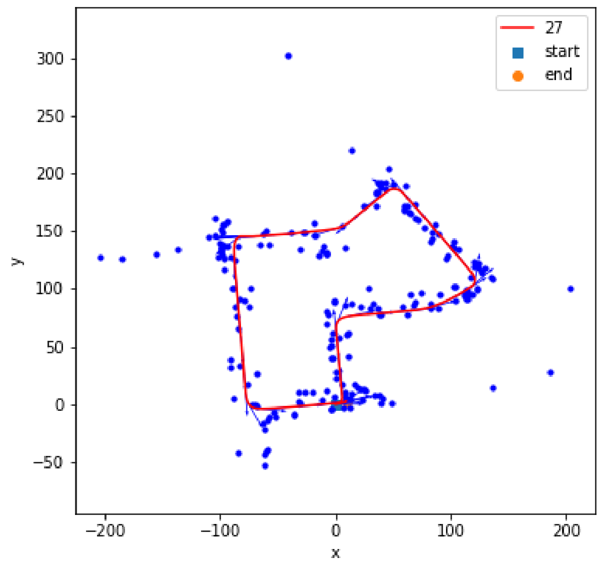

Fig. 3. DataSet 27 Landmark Mapping

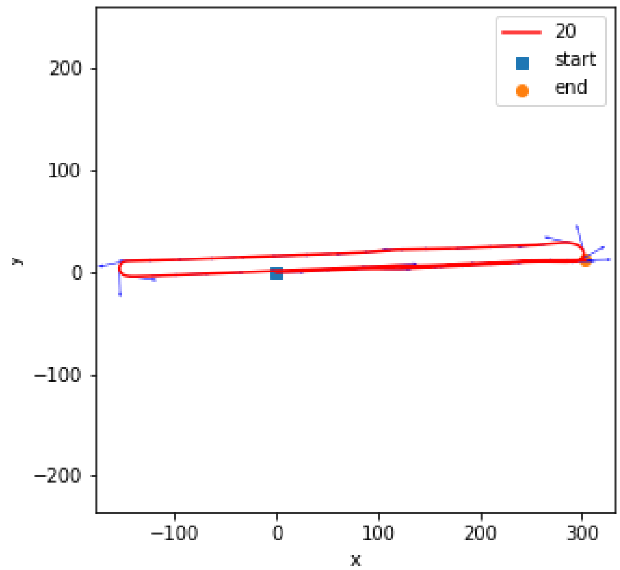

Fig. 5. Test Set 20 IMU Localization

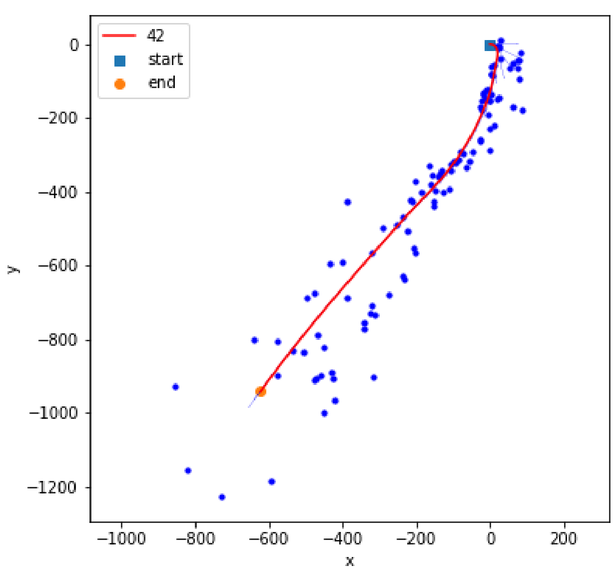

Fig. 4. DataSet 42 Landmark Mapping

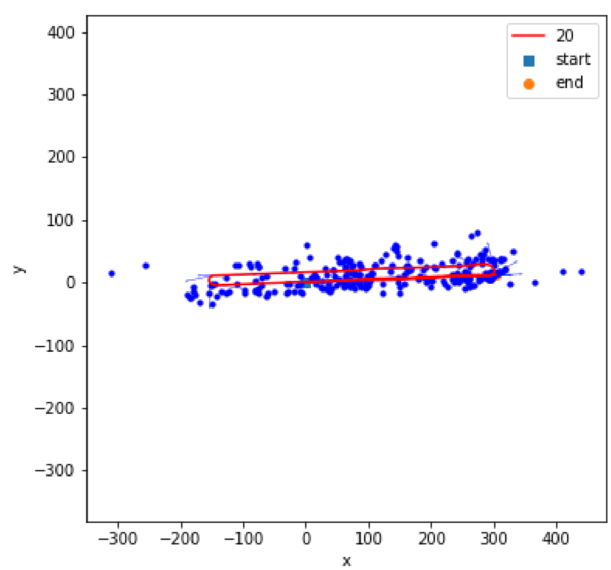

Fig. 6. Test Set 20 Landmark Mapping

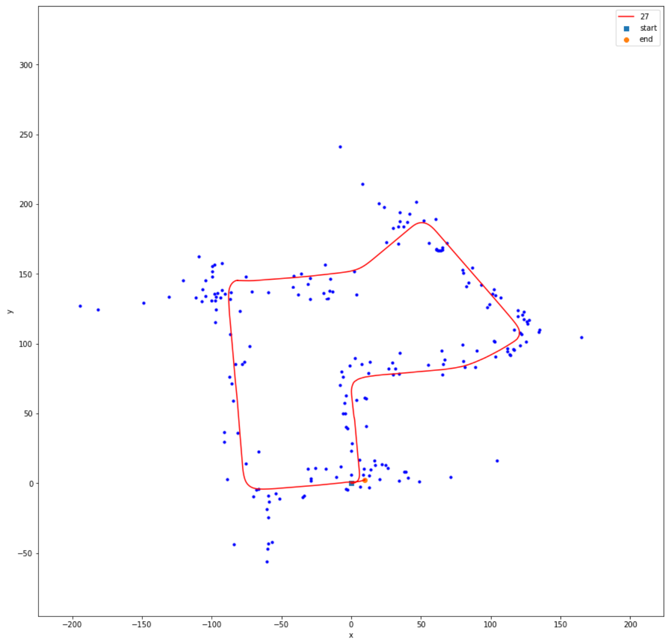

Fig. 7. DataSet 27 SLAM

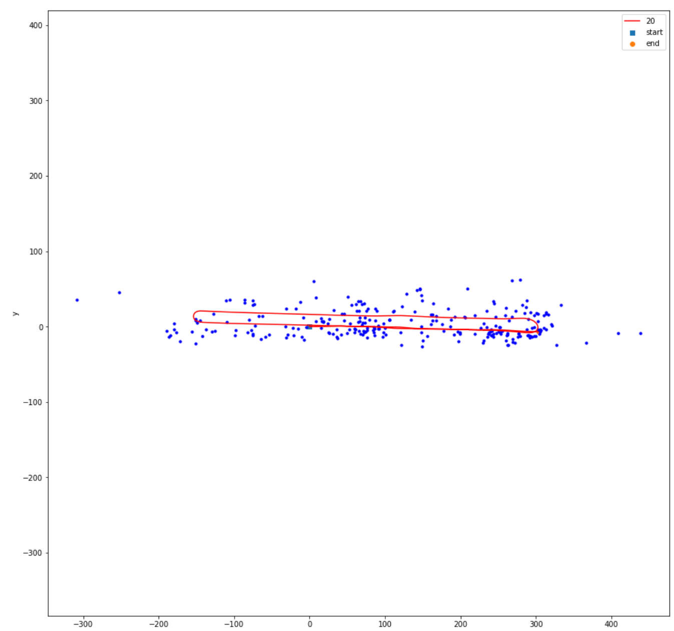

Fig. 9. Test Set 20 SLAM

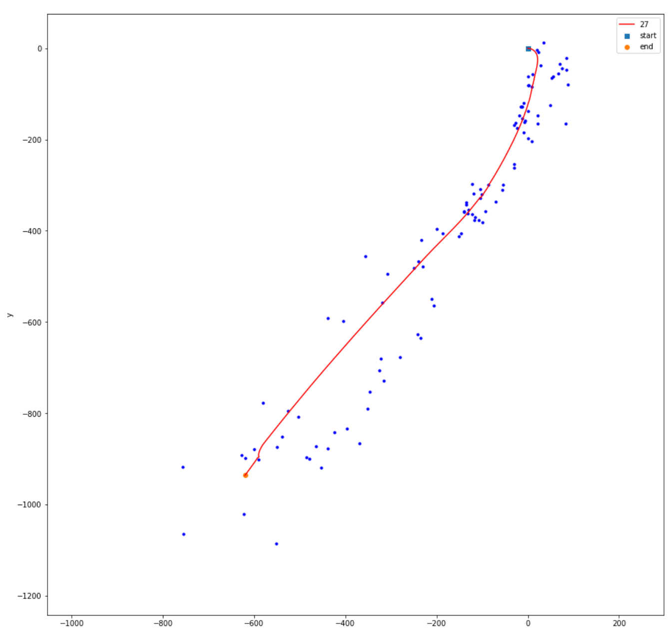

Fig. 8. DataSet 42 SLAM