Visual Inertial Simultaneous Localization and Mapping

Hongbo Qian
Department of Electrical and Computer Engineering
University of California, San Diego
hoqian@ucsd.edu

***Abstract*—Enabling the autonomous vehicle to perceive the environment is the core technology in autonomous driving. In this project, In this project, I used data from stereo camera with feature extractions from images and IMU measuring linear velocity and angular velocity to implement the Visual Inertial Simultaneous Localization an Mapping(VI-SLAM). Following the instruction, I did IMU Localization via EKF Prediction first to get the trajectory. And then I did Landmark Mapping via EKF Update to update the landmarks on the map. Finally, I combined these two steps to implement an IMU update step based on the stereo-camera observation model to obtain a complete visual-inertial SLAM algorithm.**

***Index Terms*—Visual-inertial SLAM, Kalman filter, EKF**

I. Introduction

These years, autonomous driving related technologies have become the focus of discussion, and the technology is also advancing at a rapid pace. Although people can walk freely, it is not easy to give vehicles the ability to move on the road. Not only do we need to provide the vehicle with the ability to perceive the external environment, such as providing various sensors, we also need to use the data returned by the sensor to construct a map, that is, let the vehicle know where it is. In this project, we focused on using stereo camera and IMU to collect images an velocity.

II. Problem Formulation

In this project, we have three inputs, including linear acceleration $a_t \in \mathbb{R}^3$ and rotational velocity $\omega_t \in \mathbb{R}^3$, features $z_{t,i} \in \mathbb{R}^4$ from stereo camera(left and right image pixels) for $i = 1, ..., N_t$.

$$z_t = \begin{bmatrix} z_{t,1}^T & ... & z_{t,N_t}^T \end{bmatrix} \in \mathbb{R}^{4N_t} \tag{1}$$

The transformation ${}_oT_i \in SE(3)$ from the IMU to the camera optical frame and the stereo camera calibration matrix K_s are assumed to be known.

$$K_s = \begin{bmatrix} f_{s_u} & 0 & c_u & 0 \\ 0 & f_{s_v} & c_v & 0 \\ f_{s_u} & 0 & c_u & -f_{s_u}b \\ 0 & f_{s_v} & c_v & 0 \end{bmatrix} \tag{2}$$

In the above equation, f is focal length, s_u, s_v is pixel scaling, c_u, c_v is principal point and b is the stereo baseline.

Another assumption is that the data association $\Delta_t : \{1, ..., M\} \rightarrow \{1, ..., N_t\}$ stipulating that landmark j corresponds to observation $z_{t,i} \in \mathbb{R}^4$ with $i = \Delta_y(j)$ at time t is known or provided by an external algorithm. The landmarks that are not observable at time t have an associated measurement as following

$$z_{t,i} = \begin{bmatrix} -1 & -1 & -1 & -1 \end{bmatrix} \tag{3}$$

The project have been divided into three steps. In the first step, we need to predict the localization of the vehicle after a short time. Based on this, I can use linear velocity and linear acceleration with the IMU motion model to estimate the pose of the pose $T =_W T_I \in SE(3)$ of IMU over time.

As for the landmark mapping update step, I don't need to do the prediction step. Only compute the update step to get world frame coordiantes $m_j \in \mathbb{R}^3$ of the $j = 1, ..., M$ point landmarks that generated the visual features $z_{t,i} \in \mathbb{R}^4$.

Finally, I need to combine the IMU prediction step from part (a) with the landmark update step from part (b) and implement an IMU update step based on the stereo-camera observation model to obtain a complete visual-inertial SLAM algorithm. I need to Output all of the results into image, which will put in the fourth part.

III. Technical Approach

In this part, we mainly talk about how to solve the problem in detail. I first load all of the data from the data/10.npz including linear velocity, angular velocity, timestamps, features, K_s, baseline, and IMU to camera transformation. I need to use all of these parameters in the project. As mentioned above, the features have been matched between left and right camera using an external algorithm.

A. IMU Localization via EKF Prediction

For localization only part, I want to implement EKF prediction step. I use τ as the time length between time step $t+1$ and t. The mean and covariance of pose is initialized as the identity matrix.

Let the prior of the pose be a normal distribution with mean $\mu_{t|t} \in SE(3)$ and covariance $\Sigma_{t|t} \in \mathbb{R}^{6\times 6}$. For every τ, the prediction step can be written as following:

$$\mu_{t+1|t} = \mu_{t|t} exp(\tau_t \hat{u}_t) \tag{4}$$

$$\Sigma_{t+1|t} = E[\delta\mu_{t+1|t} \delta\mu_{t+1|t}^T] \tag{5}$$

$$\Sigma_{t+1|t} = exp(-\tau_t \overset{\curlywedge}{\mathrm{u}}_t) \Sigma_{t|t} exp(-\tau_t \overset{\curlywedge}{\mathrm{u}}_t)^T + W \tag{6}$$

where $u_t = \begin{bmatrix} v_t \\ \omega_t \end{bmatrix} \in \mathbb{R}^6$, $\hat{u}_t = \begin{bmatrix} \hat{\omega}_t & v_t \\ 0^T & 0 \end{bmatrix} \in \mathbb{R}^{4\times4}$, $\overset{\curlywedge}{u}_t = \begin{bmatrix} \hat{\omega}_t & \hat{v}_t \\ 0^T & \hat{\omega}_t \end{bmatrix} \in \mathbb{R}^{6\times6}$. The hat operator means a skew-symmetric matrix for a 3x1 vector as follows.

$$\hat{\theta} = \begin{bmatrix} 0 & -\theta_3 & \theta_2 \\ \theta_3 & 0 & -\theta_1 \\ -\theta_2 & \theta_1 & 0 \end{bmatrix} \tag{7}$$

I did the above prediction step for dead-reckoning to see whether the trajectory is reasonable before doing the update step.

B. Landmark Mapping via EKF Update

To update landmark positions, we use the EKF update step to calculate new landmark locations $\mu_{t+1|t+1}$ in the world frame given the observations $z_{t+1} \in \mathbb{R}^{4Nt+1}$ and the localization prediction $\mu_{t+1|t}, \Sigma_{t+1|t}$ in the previous part. The observation model for the stereo camera is the following equation:

$$z_{t,i} = K_s\pi({}_OT_IT_{t+1}^{-1}\underline{m}_j) + v_{t,j} \tag{8}$$

where $\underline{m}_j$ represents the homogeneous matrix. $\pi(q) = \frac{1}{q_3}q \in \mathbb{R}^4$. $v_t \sim \mathcal{N}(0, I \otimes V)$. The EKF update step can be written like this:

$$K_{t+1} = \Sigma_t H_{t+1}^T (H_{t+1}\Sigma_t H_{t+1}^T + I \otimes V)^{-1} \tag{9}$$

$$\mu_{t+1} = \mu_t + K_{t+1}(z_{t+1} - K_s\pi({}_OT_IT_{t+1}^{-1}\underline{\mu}_j)) \tag{10}$$

$$\Sigma_{t+1} = (I - K_{t+1}H_{t+1})\Sigma_t \tag{11}$$

The Jacobian matrix of $\tilde{z}_{t+1,i}$ can be evaluated at $\mu_{t,j}$:

$$H_{t+1,i,j} = \begin{cases} K_s \frac{d\pi}{dq}({}_OT_IT_{t+1}^{-1}\underline{\mu}_{t,j})_OT_IT_{t+1}^{-1}P^T, & \text{if} \Delta_t(j) = i \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

$$\frac{d\pi}{dq} = \frac{1}{q_3}\begin{bmatrix} 1 & 0 & -\frac{q_1}{q_3} & 0 \\ 1 & 0 & -\frac{q_2}{q_3} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{q_4}{q_3} & 0 \end{bmatrix} \in \mathbb{R}^{4\times4} \tag{13}$$

$$I \otimes V = \begin{bmatrix} V & & \\ & \ddots & \\ & & V \end{bmatrix} \tag{14}$$

From the above equations, we can compute the EKF update step.

C. Visual-Inertial SLAM

In part c, we want to implement a full Visual-Inertial SLAM. It means that we need to combine the IMU prediction step from part (a) with the landmark update step from part (b) and implement an IMU update step based on the stereo-camera observation model to obtain a complete visual-inertial SLAM algorithm.

The equations should be similar as before, the only difference is that I need to combine the Kalman gain together. First I concatenate the mean and covariance of features and IMU together.

$$\mu = \begin{bmatrix} \mu_f \\ \mu_{IMU} \end{bmatrix} \in \mathbb{R}^{3M+6} \tag{15}$$

$$\Sigma = \begin{bmatrix} \Sigma_f & C \\ C^T & \Sigma_{IMU} \end{bmatrix} \in \mathbb{R}^{3M+6\times3M+6} \tag{16}$$

After we define this, we can do the prediction step. Actually, the prediction step is the same as before. But in the update step, we also need to combine the Jacobian matrix together.

$$H = \begin{bmatrix} H_f & H_{IMU} \end{bmatrix} \in \mathbb{R}^{3M+6} \tag{17}$$

And then every step is the same as part b to update the pose of the vehicle and also the features on the map. We need to compute the Kalman gain, the combined mean and covariance.

$$K_{t+1} = \Sigma_t H_{t+1}^T (H_{t+1}\Sigma_t H_{t+1}^T + I \otimes V)^{-1} \tag{18}$$

$$\mu_{t+1} = \begin{bmatrix} \mu_t + K_{t+1}(z_{t+1} - K_s\pi({}_OT_IT_{t+1}^{-1}\underline{\mu}_j)) \\ \mu_{t|t}exp(\tau_t\hat{u}_t) \end{bmatrix} \tag{19}$$

$$\Sigma_{t+1} = \begin{bmatrix} (I - K_{t+1}H_{t+1})\Sigma_t \\ exp(-\tau_t\overset{\curlywedge}{u}_t)\Sigma_{t|t}exp(-\tau_t\overset{\curlywedge}{u}_t)^T \end{bmatrix} \tag{20}$$

IV. Result

A. IMU Localization via EKF Prediction

Just implement the prediction step. Didn't update the trajectory. The dead reckoning result is shown below. The result looks reasonable with the vehicle trajectory being smooth. I can do the next step.

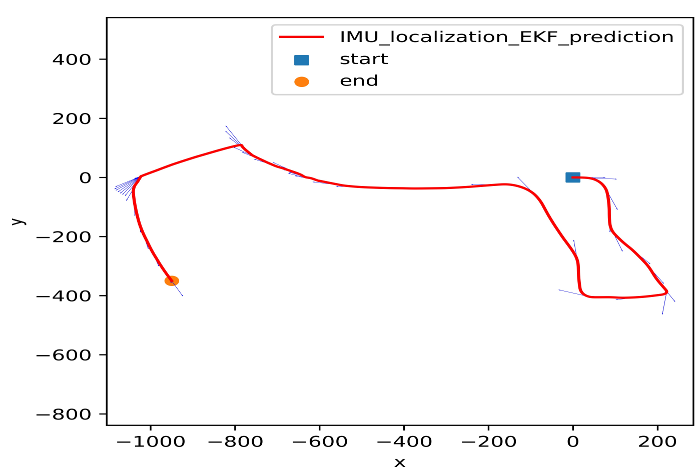

Fig. 1. IMU localization EKF prediction

B. Landmark Mapping via EKF Update

The result in this part is shown in figure 2 and figure 3. In this part, I didn't use all of the features which are supplied in order to accelerate the speed. The observation noise covariance I used is 30.

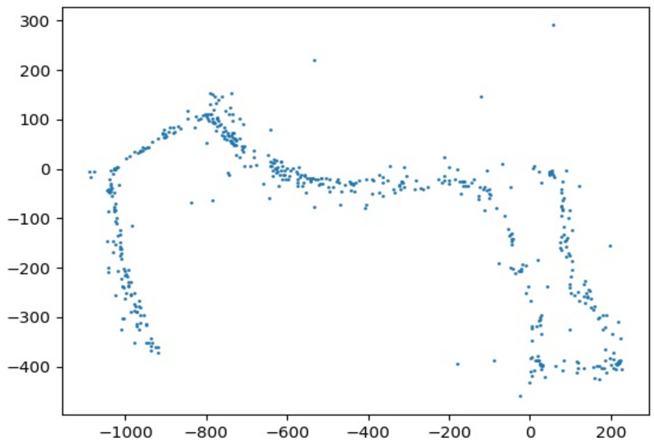

Fig. 2. landmarks

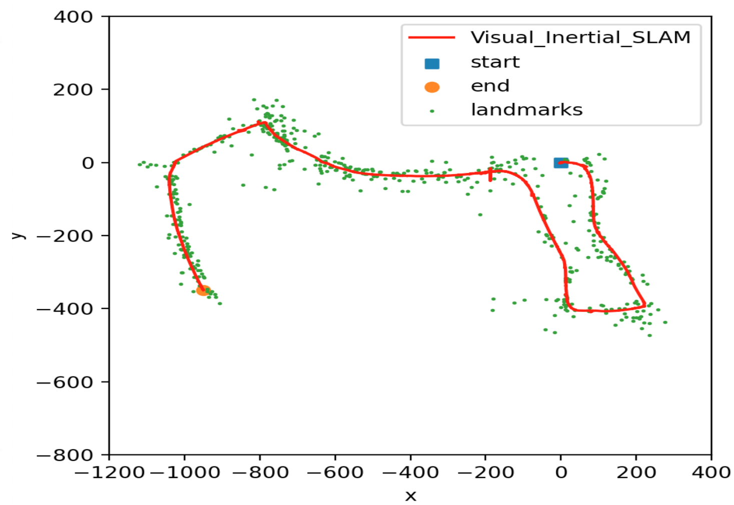

Fig. 4. Visual Inertial SLAM Result

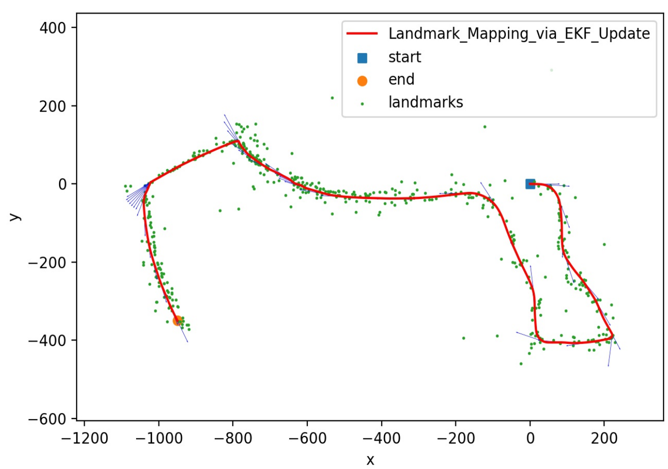

Fig. 3. Landmark Mapping via EKF Update

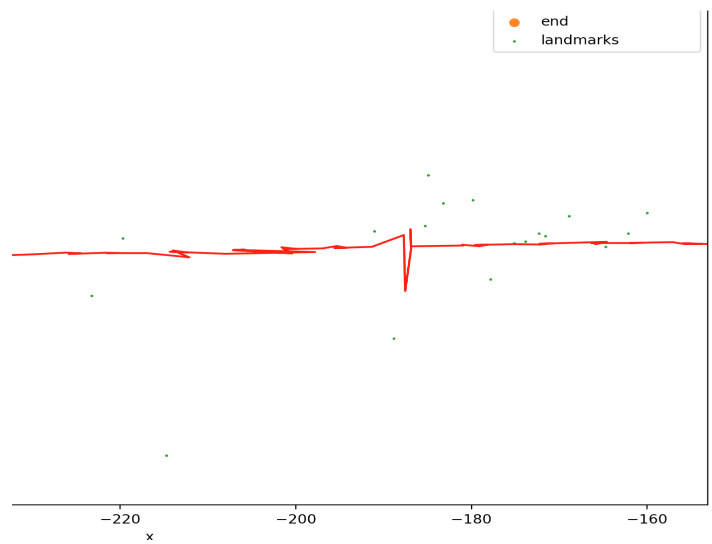

Fig. 5. The curve of the result

C. *Visual-Inertial SLAM*

The result is shown if figure 4 and figure 5.

In this part, the prior landmark covariance is $I_{3\times3}$, the prior IMU covariance is $I_{6\times6}$, the observation noise covariance is $30 * I$ and the process noise covariance I used is 0.0001. I only used 4 percent points to draw the result. However, the line is not always smooth, it will have some curves. This is because of the noise. If the noise is bigger, the result will be worse. But overall, the map points are reasonable enough.

V. Discussion

From the above results, the Visual-Inertial SLAM algorithm seems resonable to get the vehicle trajectory and the environment. But I think the speed of it is still not enough to run at real time. And it can be affected a lot by the noise. However, in practice, the error may be larger, which will make the results very inaccurate, so improving the calculation speed and reducing the impact of errors on the results is the future direction of efforts.