Visual-Inertial SLAM via Extended Kalman Filter

Yuxiang Kang
Department of Mechanical and Aerospace Engineering
University of California, San Diego
La Jolla, U.S.A
yuk007@ucsd.edu

Nikolay Atanasov
Department of Electrical and Computer Engineering
University of California, San Diego
La Jolla, U.S.A
natanasov@eng.ucsd.edu

***Abstract*—This paper considers a visual-Inertial simultaneous localization and mapping (SLAM) system via an extended Kalman filter. The SLAM system involves an inertial measurement unit (IMU) and a stereo camera. The sensors are attached to a moving vehicle and collected data in the process. To decrease the error caused by noise in SLAM, extended Kalman filter (EKF) was applied to correct the vehicle's trajectory based on visual features detected in camera data.**

***Index Terms*—SLAM, EKF, Extended Kalman Filter, IMU, Stereo Camera**

I. Introduction

Simultaneous localization and mapping (SLAM) is a widely used technique for estimating motion from sensor data and reconstructing the spatial configuration of an unknown environment [1]. In normal SLAM algorithms, multiple types of sensors are widely integrated, including laser range sensors (LiDAR), rotary encoders, inertial measurement units (IMU), and cameras [2-3].

In a typical SLAM problem, we are given observation $\boldsymbol{z}_{0:T}$ and control inputs $\boldsymbol{u}_{0:T}$, and are tasked to estimate the robot state trajectory $x_{0:T}$ and build a map $\boldsymbol{m}$ of the environment. A basic approach is to first predict the state trajectory $x_{0:T}$ by apply proper motion model to control inputs $u_{0:T-1}$, and then use each time stamp's measurement z_t and robot pose x_t to update map $\boldsymbol{m}$.

For Visual-Inertial SLAM, the control inputs are linear acceleration $\boldsymbol{a}_{0:T} \in \mathbb{R}^3$ and angular velocity $\boldsymbol{\omega}_{0:T} \in \mathbb{R}^3$, and observation is visual features $\boldsymbol{z}_{1:T+1}$, which are detected from camera data. The main thought is to first implement a motion model to control inputs and generate the robot's pose for next time stamp, which is the localization part of SLAM. With robot's pose, we can build a map $\boldsymbol{m}$ by implement an observation model to visual features and generates their coordinates in world frame, which the mapping part of SLAM. After localization at each step, we can use the predicted pose and visual features' coordinates to generate predicted observation, and compare it with actual observation for the landmarks observed more than once. The task is to correct the robot's pose to minimize the difference between predicted and actual observation, which achieves SLAM part.

In this project, extend Kalman filter (EKF) is implemented in both visual mapping and SLAM part at each time stamp. Kalman filter uses a series of observations over time, including statistical noise, and produces estimates of unknown variables that tend to be more accurate than those based on a single measurement alone, by estimating a joint probability distribution over the variables for each time stamp [4]. EKF enables nonlinear motion and observation models to be implemented with Kalman filter. As the models involved in this project are nonlinear, EKF is implemented.

II. Problem Formation

A. IMU localization via EKF prediction

In this part, we are tasked to implement an EKF prediction step to estimate the pose $T_t \in SE(3)$ of the IMU over time t. The input are linear velocity $\boldsymbol{v}_t \in \mathbb{R}^3$, angular velocity $\boldsymbol{\omega}_t \in \mathbb{R}^3$, prior robot pose $T_t \in SE(3)$ and pose variance $\Sigma_t \in \mathbb{R}^{6\times 6}$. The expected output are predicted robot pose $T_{t+1|t} \in SE(3)$ and pose variance $\Sigma_{t+1|t}$.

B. Landmark mapping via EKF update

In this part, we assume the IMU pose T_{t+1} is known, and are tasked to implement an EKF with the landmark positions $\boldsymbol{m}_t \in \mathbb{R}^{3\times M}$ as a state and perform EKF update steps using the visual observations $\boldsymbol{z}_{t+1}$ to keep track of the mean and covariance of $\boldsymbol{m}$.

The input of this part is IMU pose T_{t+1}, observation of visual features $\boldsymbol{z}_{t+1} \in \mathbb{R}^{4\times N_t}$, existing map of landmarks

$\boldsymbol{m}_t$, map variance $\Sigma_{m;t}$ and the data association between landmark and observation:

$$\Delta_t: \{1, \dots, M\} \rightarrow \{1, \dots, N_t\}$$

The expected output is updated map $\boldsymbol{m}_{t+1|t+1}$ and updated map variance $\Sigma_{m;t+1}$.

C. Visual-inertial SLAM

In this part, we assume the updated map $\boldsymbol{m}_{t+1|t+1}$is static and correct, and use predicted robot pose $T_{t+1|t}$ to generate predicted observation $\tilde{\boldsymbol{z}}_{t+1}$. By using the difference between predicted observation and actual observation, we can generate the robot pose update $T_{t+1|t+1}$ and robot pose variance $\Sigma_{t+1|t+1}$.

III. Technical Approach

A. IMU localization via EKF prediction

In this part, we are tasked to implement EKF predict step regarding robot pose as a state. Pose kinematics model is applied to predict the robot pose, and perturbation kinematics is applied to predict the robot variance. Given the control input $\boldsymbol{v}_t \in \mathbb{R}^3$, $\boldsymbol{\omega}_t \in \mathbb{R}^3$, prior robot pose $T_{t|t}$ and pose variance Σ_t, we can first rearrange control input as twist form:

$$\boldsymbol{\xi}_t = \begin{bmatrix} \boldsymbol{v}_t \\ \boldsymbol{\omega}_t \end{bmatrix} \in \mathbb{R}^6$$

Then, to generate the predicted pose:

$$T_{t+1|t} = T_{t|t} \exp\,(\tau_t \hat{\boldsymbol{\xi}}_t) \tag{1}$$

where τ_t is time gap between time stamps, $\hat{\boldsymbol{\xi}}_t$ is hat map of $\boldsymbol{\xi}_t$:

$$\hat{\boldsymbol{\xi}}_t = \begin{bmatrix} \hat{\boldsymbol{\omega}}_t & \boldsymbol{v}_t \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{4\times4}, \hat{\boldsymbol{\omega}} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}$$

Predicted pose variance is also generated:

$$\Sigma_{t+1|t} = \exp\left(-\tau_t \overset{\curlywedge}{\boldsymbol{\xi}}_t\right) \Sigma_{t|t}\ \exp\left(-\tau_t \overset{\curlywedge}{\boldsymbol{\xi}}_t\right)^{\mathrm{T}} + W \tag{2}$$

where $W \in \mathbb{R}^{6\times6}$ is motion noise, and can be defined as $W = \boldsymbol{w}I, \boldsymbol{w} \in \mathbb{R}^6$,

$$\overset{\curlywedge}{\boldsymbol{\xi}}_t = \begin{bmatrix} \hat{\boldsymbol{\omega}}_t & \boldsymbol{v}_t \\ 0 & \hat{\boldsymbol{\omega}}_t \end{bmatrix} \in \mathbb{R}^{6\times6}$$

B. Landmark mapping via EKF update

In this part, we are tasked to update landmark map $\boldsymbol{m}_t$ based on actual observation $\boldsymbol{z}_{0:t+1}$, predicted robot pose $T_{t+1|t}$. The prior also includes map variance $\Sigma_{m;t}$. The observation model of stereo camera is implemented to generate predicted observation.

To decrease calculation complexity, we implement EKF to each landmark observation independently. At each timestamp, we compare observation $\boldsymbol{z}_{t+1}$ with $\boldsymbol{z}_t$ to determine which landmarks in $\boldsymbol{z}_{t+1}$ are observed for the first time. For these landmarks, we can directly apply stereo camera observation model:

$$\boldsymbol{z} = \begin{bmatrix} u_L \\ v_L \\ u_L \\ v_R \end{bmatrix} = \begin{bmatrix} fs_u & 0 & c_u & 0 \\ 0 & fs_v & c_v & 0 \\ fs_u & 0 & c_u & -fs_u b \\ 0 & fs_v & c_v & 0 \end{bmatrix} \frac{1}{z_o} \begin{bmatrix} x_o \\ y_o \\ z_o \\ 1 \end{bmatrix}$$

$$= K_s \frac{1}{z_o} \begin{bmatrix} x_o \\ y_o \\ z_o \\ 1 \end{bmatrix} \tag{3}$$

$$\begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix} = T_{t+1|t} T_{io} \begin{bmatrix} x_o \\ y_o \\ z_o \\ 1 \end{bmatrix} \tag{4}$$

where $\begin{bmatrix} x_o \\ y_o \\ z_o \end{bmatrix}$ is the coordinates of the landmark in optical frame, T_{io} is the transformation matrix of camera frame in IMU frame. $\begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix}$ is the coordinates of the landmark in world frame. We can update the landmarks in $\boldsymbol{m}_t$ by:

$$\boldsymbol{m}_{t+1,k} = \begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix} \in \mathbb{R}^4 \tag{5}$$

For the landmarks that have also been observed at the former time stamp, we can first generate predicted observation by:

$$\tilde{\boldsymbol{z}}_{t+1,k} = K_s \pi({T_{io}}^{-1} {T_{t+1|t}}^{-1} \underline{\boldsymbol{m}}_{t+1,k}) \tag{6}$$

where

$$\underline{\boldsymbol{m}}_{t+1,k} = \begin{bmatrix} \boldsymbol{m}_{t+1,k} \\ 1 \end{bmatrix}$$

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

The Jacobian of predicted observation w.r.t. $\boldsymbol{m}$:

$$H_{t+1,k} = K_s \frac{\mathrm{d}\pi}{\mathrm{d}\boldsymbol{q}} ({T_{io}}^{-1} {T_{t+1|t}}^{-1} \underline{\boldsymbol{m}}_{t+1,k}) {T_{io}}^{-1} {T_{t+1|t}}^{-1} P^{\mathrm{T}} \tag{7}$$

Then we can implement EKF update step and update the map:

$$K_{t+1,k} = \Sigma_{m;t,k} {H_{t+1,k}}^{\mathrm{T}} (H_{t+1,k} \Sigma_{m;t,k} {H_{t+1,k}}^{\mathrm{T}} + I \otimes V)^{-1} \tag{8}$$

$$\boldsymbol{m}_{t+1,k} = \boldsymbol{m}_{t,k} + K_{t+1,k}(\boldsymbol{z}_{t+1,k} - \tilde{\boldsymbol{z}}_{t+1,k}) \tag{9}$$

$$\Sigma_{m;t+1,k} = (I - K_{t+1,k} H_{t+1,k}) \Sigma_{m;t,k} \tag{10}$$

where $\Sigma_{m;t}[k] \in \mathbb{R}^{3\times3}$ is the map variance of landmark k. V is observation noise.

As for the landmarks in $\boldsymbol{m}_t$ that are not observed at time stamp $t + 1$, $\boldsymbol{m}_{t+1,k}, = \boldsymbol{m}_{t,k}$.

C. Visual-inertial SLAM

With updated map $\boldsymbol{m}_{t+1}$, we can calculate the Jacobian of predicted observation w.r.t. pose T:

$$H_{t+1,k} = -K_s \frac{\mathrm{d}\pi}{\mathrm{d}\boldsymbol{q}} ({T_{io}}^{-1} {T_{t+1|t}}^{-1} \underline{\boldsymbol{m}}_{t+1,k}) {T_{io}}^{-1} ({T_{t+1|t}}^{-1} \underline{\boldsymbol{m}}_{t+1,k})^{\odot} \quad (11)$$

where

$$\begin{bmatrix} \boldsymbol{q} \\ 1 \end{bmatrix}^{\odot} = \begin{bmatrix} I & -\widehat{\boldsymbol{q}} \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{4\times 6}$$

$$H_{t+1} = \begin{bmatrix} H_{t+1,1} \\ \cdots \\ H_{t+1,N} \end{bmatrix} \in \mathbb{R}^{4N\times 6}$$

Then the robot pose is updated:

$$K_{t+1} = \Sigma_{t+1|t} {H_{t+1}}^{\mathrm{T}} (H_{t+1} \Sigma_{t+1|t} {H_{t+1}}^{\mathrm{T}} + I \otimes V)^{-1} \quad (12)$$

$$T_{t+1|t+1} = T_{t+1|t} \exp\left(K_{t+1}(\boldsymbol{z}_{t+1} - \tilde{\boldsymbol{z}}_{t+1})^{\wedge}\right) \quad (13)$$

$$\Sigma_{t+1|t+1} = (I - K_{t+1} H_{t+1}) \Sigma_{t+1|t} \quad (14)$$

IV. Results

A. IMU localization via EKF prediction

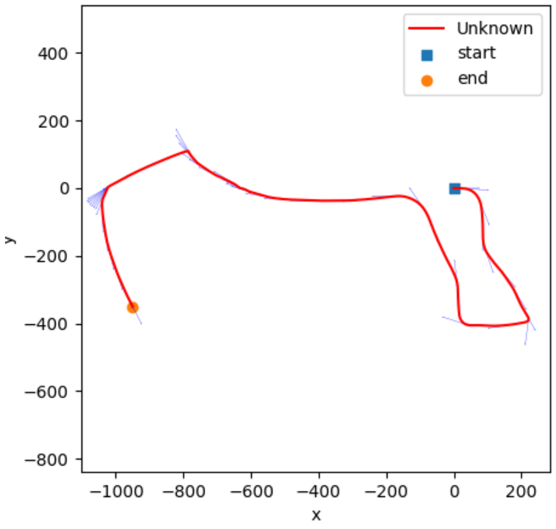

Figure 1: Robot Trajectory of Dataset 10

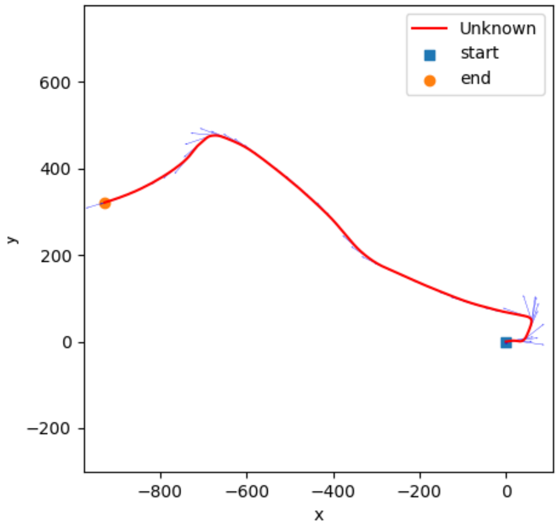

Figure 2: Robot Trajectory of Dataset 03

B. Landmark mapping via EKF update and predicted trajectory

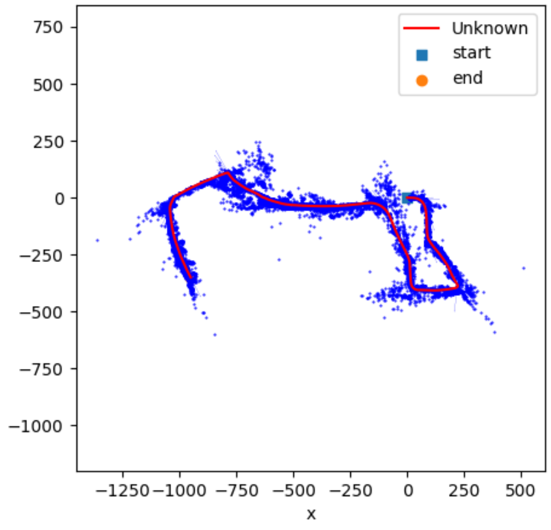

Figure 3: Landmarks in world frame of Dataset 10

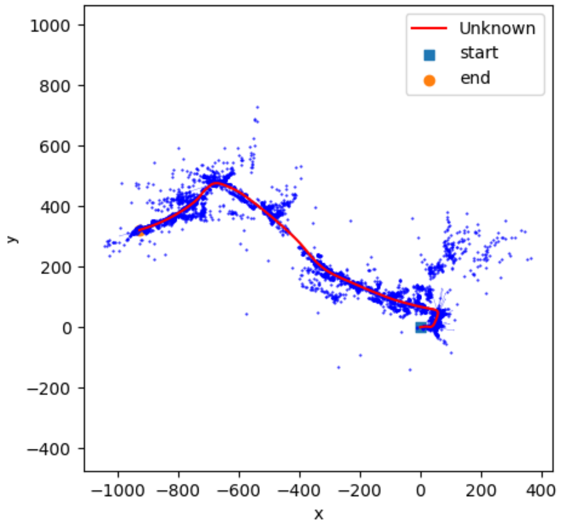

Figure 4: Landmarks in world frame of Dataset 03

C. Visual-inertial SLAM results

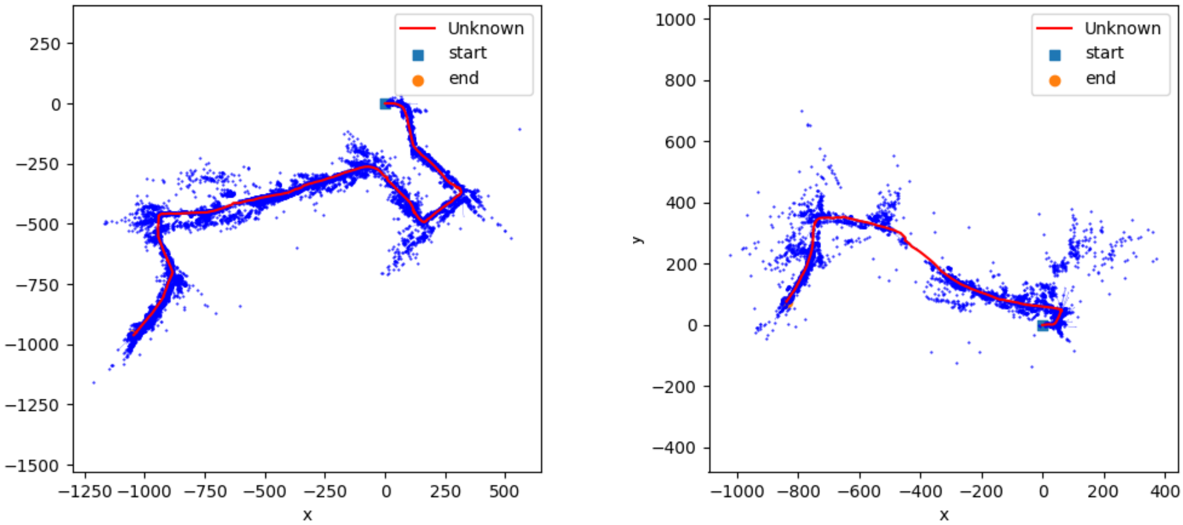

Figure 5-6: Trajectory and landmarks generated by SLAM of Dataset 10 and 03

V. Evaluation

The trajectory corrected by landmark observation via EKF is greatly different from the trajectory only based on IMU prediction. We notice that in the change between dead reckoning and SLAM in y-axis is apparently greater than x-axis.

One problem needs to be optimized is the size of matrix in SLAM update step. We can sample the landmarks instead of using all of them to decrease the matrix's size and accelerate the calculation.

References

[1] Chatila R, Laumond JP (1985) Position referencing and consistent world modeling for mobile robots In: Proceedings of International Conference on Robotics and Automation, 138–145.

[2] Taketomi, T., Uchiyama, H., & Ikeda, S. (2017). Visual SLAM algorithms: A survey from 2010 to 2016. IPSJ Transactions on Computer Vision and Applications, 9(1), 1-11.

[3] Pritsker, A. A. B. (1984). Introduction to Simulation and SLAM II. Halsted Press.

[4] Welch, G. F. (2020). Kalman filter. Computer Vision: A Reference Guide, 1-3.