Visual-Inertial SLAM

Yidi Zhu
Department of Electrical Computer Engineering
University of California, San Diego
yizhu@eng.ucsd.edu

Abstract—This paper presented approaches using Extended Kalman Filter to realize the inertial SLAM.

Index Terms— Extended Kalman Filter, SE(3) Geometry and Kinematics, Visual-Inertial SLAM

I. Introduction

In robotics and mapping area, Simultaneous localization and mapping (SLAM) is an important way to construct a map of the unknown environment and keep tracking where the robot is. This chicken and egg type problem could be solved by SLAM in a easily understand form. Such computational way is implemented many kinds of intelligent devices such as cars, detecting machines and so on.

In this project, there are three steps, IMU Localization via EKF Prediction, Landmark Mapping via EKF Update and IMU Pose Updating. The updating for pose and landmark could be melted into one part since the landmark and pose have correlation to each other. Finally, we could get the odometry of the robot and modify its pose simultaneously with constructing the landmarks.

II. Problem Formulation

① Left Jacobean

The this SLAM, we would apply some perturbation to the pose∈SE(3) by computing the left Jacobean (or right Jacobean, here we use left Jacobean) to

$$\log(\exp(\hat{\xi}_1)\exp(\hat{\xi}_2))^\vee \approx \begin{cases} \mathcal{J}_L(\xi_2)^{-1}\xi_1 + \xi_2 & \text{if } \xi_1 \text{ is small} \\ \xi_1 + \mathcal{J}_R(\xi_1)^{-1}\xi_2 & \text{if } \xi_2 \text{ is small} \end{cases}$$

$$\begin{aligned} \exp\left((\xi + \delta\xi)^\wedge\right) &\approx \exp(\hat{\xi})\exp\left((\mathcal{J}_R(\xi)\delta\xi)^\wedge\right) \\ &\approx \exp\left((\mathcal{J}_L(\xi)\delta\xi)^\wedge\right)\exp(\hat{\xi}) \end{aligned}$$

We use the showing formula to do the transformation. This could be used in updating in pose.

② EKF (Extended Kalman Filter)

F is the motion model and h is the observation model. X is the state of the robot and u is the input.

$$f(x_t, u_t, w_t) \approx f(\mu_{t|t}, u_t, 0) + \left[\frac{df}{dx}(\mu_{t|t}, u_t, 0)\right](x_t - \mu_{t|t}) + \left[\frac{df}{dw}(\mu_{t|t}, u_t, 0)\right](w_t - 0)$$

$$h(x_{t+1}, v_{t+1}) \approx h(\mu_{t+1|t}, 0) + \left[\frac{dh}{dx}(\mu_{t+1|t}, 0)\right](x_{t+1} - \mu_{t+1|t}) + \left[\frac{dh}{dv}(\mu_{t+1|t}, 0)\right](v_{t+1} - 0)$$

EKF uses the first order of Taylor series approximation to compute the motion and observation model. (Wt and Vt+1 is a Gaussian noise with a covariance W and V.)

$$F_t := \frac{df}{dx}(\mu_{t|t}, u_t, 0) \text{ and } Q_t := \frac{df}{dw}(\mu_{t|t}, u_t, 0)$$

$$H_{t+1} := \frac{dh}{dx}(\mu_{t+1|t}, 0) \text{ and } R_{t+1} := \frac{dh}{dv}(\mu_{t+1|t}, 0)$$

$$\text{Prediction:} \quad \begin{aligned} \mu_{t+1|t} &= f(\mu_{t|t}, u_t, 0) \\ \Sigma_{t+1|t} &= F_t \Sigma_{t|t} F_t^T + Q_t W Q_t^T \end{aligned}$$

$$\text{Update:} \quad \begin{aligned} \mu_{t+1|t+1} &= \mu_{t+1|t} + K_{t+1|t}(z_{t+1} - h(\mu_{t+1|t}, 0)) \\ \Sigma_{t+1|t+1} &= (I - K_{t+1|t} H_{t+1}) \Sigma_{t+1|t} \end{aligned}$$

With computing the Kalman Gain first:

$$K_{t+1|t} := \Sigma_{t+1|t} H_{t+1}^T \left(H_{t+1} \Sigma_{t+1|t} H_{t+1}^T + R_{t+1} V R_{t+1}^T\right)^{-1}$$

③ Visual Mapping via EKF

$$\hat{\mathbf{z}}_{t,i} := M\pi\left({}_O T_I T_t \boldsymbol{\mu}_{t,j}\right) \in \mathbb{R}^4 \qquad \text{for } i = 1, \ldots,$$

From the connection between observation in image frame and the landmark in world frame, we can do back projection to get the estimated position of land marks in world frame.

$$H_{i,j,t} = \begin{cases} M\frac{d\pi}{d\mathbf{q}}\left({}_O T_I T_t \boldsymbol{\mu}_{t,j}\right) {}_O T_I T_t D & \text{if observation } i \text{ corresponds to landmark } j \text{ at time } t \\ \mathbf{0} \in \mathbb{R}^{4\times 3} & \text{otherwise} \end{cases}$$

Then we can compute the Jacobian of the of z with respect to m evaluated at μ, now we can do updating for mapping.

$$K_t = \Sigma_t H_t^T \left(H_t \Sigma_t H_t^T + I \otimes V\right)^{-1}$$
$$\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t + DK_t\left(\mathbf{z}_t - \hat{\mathbf{z}}_t\right)$$
$$\Sigma_{t+1} = (I - K_t H_t)\Sigma_t$$

III. Technical Approach

I separate my approach into 3 process.

1) Pre-processing

Before we begin the process of SLAM. We have several pre tasks.

I. Load the data form IMU

II. Initial the Landmark and the robot's state.

III. Compute the parameter in next computation (such as the calibration matrix M)

2) IMU Localization via EKF Prediction

Reading the data of IMU to get the angular velocity and linear velocity on x,y,z-axis. Then use the discrete time motion model to predict the next state of the robot.

After setting the covariance of the Gaussian noise, now we can compute following function to realize the EKF.

$$\boldsymbol{\mu}_{t+1|t} = \exp\left(-\tau \hat{u}_t\right)\boldsymbol{\mu}_{t|t} \qquad u_t := \begin{bmatrix} v_t \\ \omega_t \end{bmatrix}$$
$$\Sigma_{t+1|t} = \mathbb{E}[\xi_{t+1|t}\xi_{t+1|t}^T] = \exp\left(-\tau \overset{\curlywedge}{u}_t\right)\Sigma_{t|t}\exp\left(-\tau \overset{\curlywedge}{u}_t\right)^T + \tau^2 W$$

Fig1. Re-write discrete time motion model

3) Updating both landmarks and pose through EKF Updating

$$\hat{\mathbf{z}}_{t,i} := M\pi\left({}_O T_I T_t \boldsymbol{\mu}_{t,j}\right) \in \mathbb{R}^4 \qquad \text{for } i = 1, \ldots, N_t$$

I. Updating for landmarks

We need to know the process of the generation and definition for the landmarks.

Generally, landmarks are some objects which could be identified as a standard position to modify the pose of the robots. So, we always need correct mapping of landmarks to update the pose.

In this project, the landmarks all are recorded in the "features" matrix, the second index of the features is the order of the landmarks. On the other hand, when only consider the i-th term of that index, we could get the observation state (in image frame) through time of certain landmark.

When certain landmark is not detected, the observation is [-1,-1,-1,-1]. Otherwise, it should be the u v pixel in picture frame.

At the first time we observe certain landmark, we should initialize its position by its back projection in world frame,

Next time we should never initial it and just update its position by current pose of the robot and the observation of it in image frame. (We should set the covariance of the observation noise.)

Ⅱ. Updating for pose

Like the process in I, now we need to update the pose of the robot. The only difference is that now we concern Tt not μ.

Generally, by comparing the estimated projection of landmarks and the actual observation, we could update the pose.

$$H_{i,t+1|t} = M\frac{d\pi}{d\mathbf{q}}\left({}_O T_I \boldsymbol{\mu}_{t+1|t}\mathbf{m}_j\right) {}_O T_I \left(\boldsymbol{\mu}_{t+1|t}\mathbf{m}_j\right)^{\odot} \in \mathbb{R}^{4\times 6}$$

$$K_{t+1|t} = \Sigma_{t+1|t} H_{t+1|t}^T \left(H_{t+1|t}\Sigma_{t+1|t}H_{t+1|t}^T + I \otimes V\right)^{-1}$$
$$\boldsymbol{\mu}_{t+1|t+1} = \exp\left(\left(K_{t+1|t}(\mathbf{z}_{t+1} - \hat{\mathbf{z}}_{t+1})\right)^{\wedge}\right)\boldsymbol{\mu}_{t+1|t}$$
$$\Sigma_{t+1|t+1} = (I - K_{t+1|t}H_{t+1|t})\Sigma_{t+1|t}$$

We can set some limitation when updating the pose. Suppose that the error of the process is not large, we would never meet the problem that the observation and the current position of an object are quitely different. So, when this event happens, there may be something wrong with the landmark.

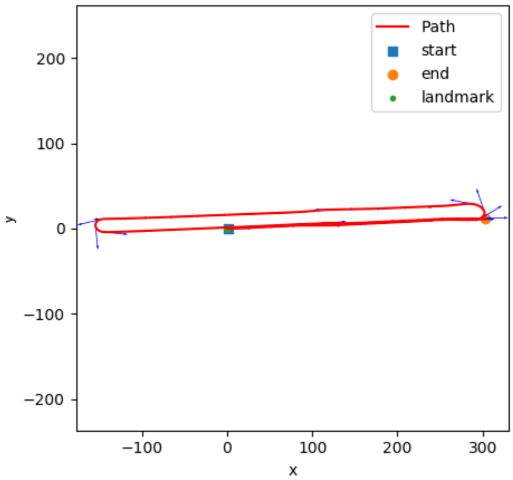
Combination:

In Visual Inertial SLAM, we need to combine the updating for landmarks and pose since there is a correlation between them. If we update them separately, we define them as independent. So, we need to combine the covariance of them to one big matrix to get the correlation term.

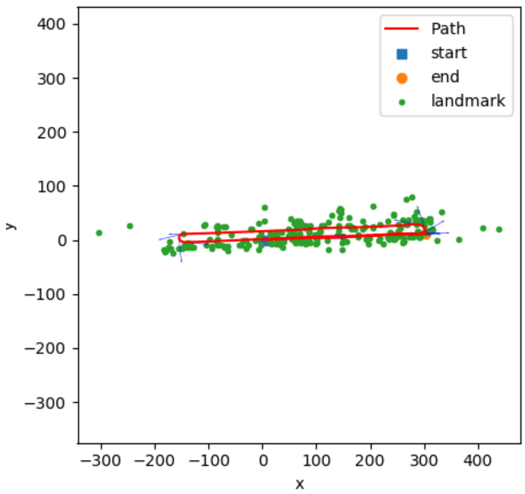
And when applying updating for the mean of the Gaussian distribution, we would separate the K again to do updating.

IV. Results:

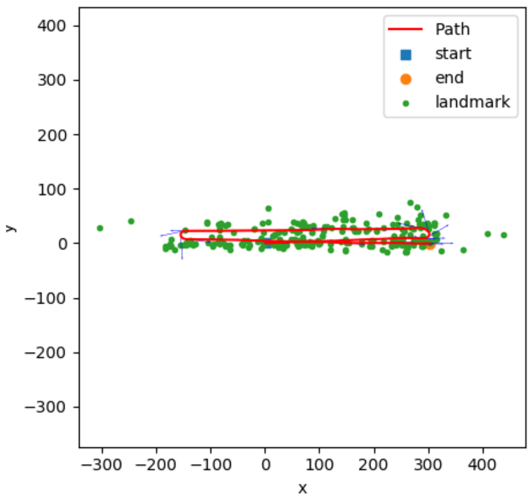
dataset:20



prediction



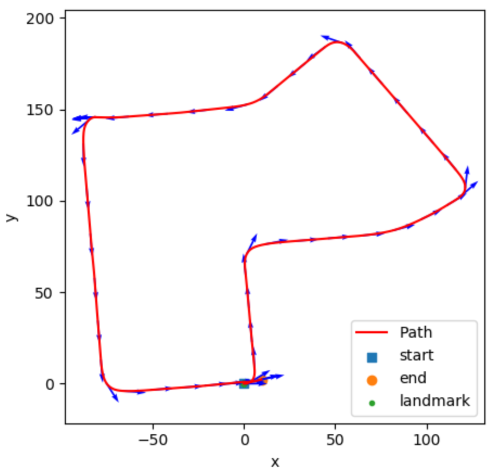
mapping



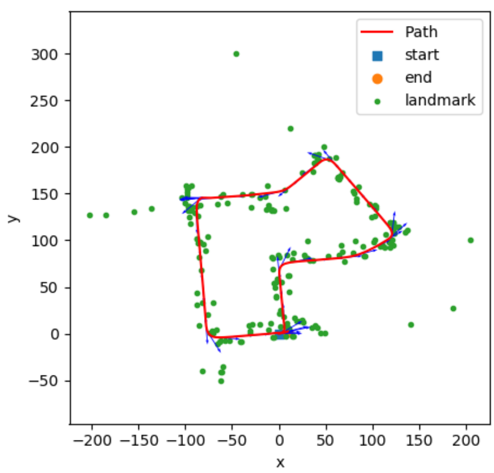
SLAM

We can see that after applying SLAM, the result becomes weird in the direction, may be the impact of noise forces the trajectory to go a little wrong when turn back. Finally, the overlap of the trajectory becomes small.

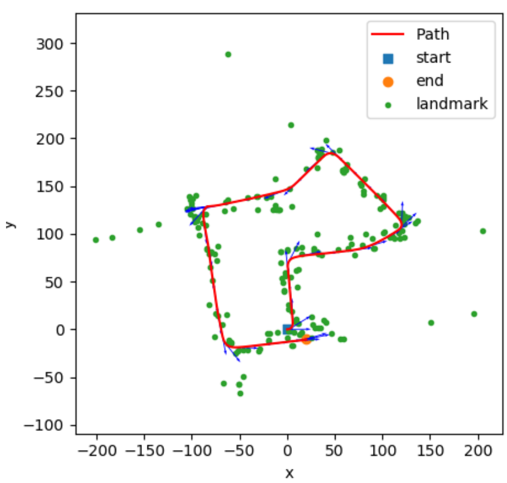
dataset:27



prediction



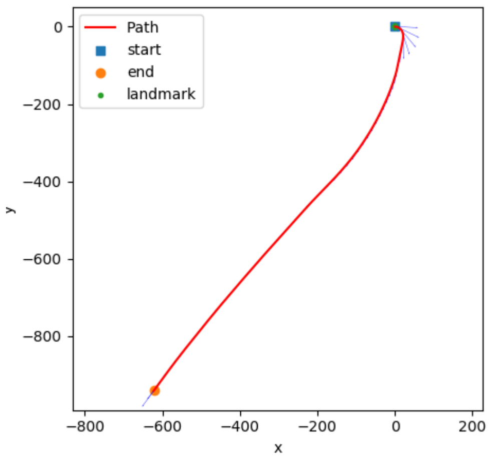
mapping



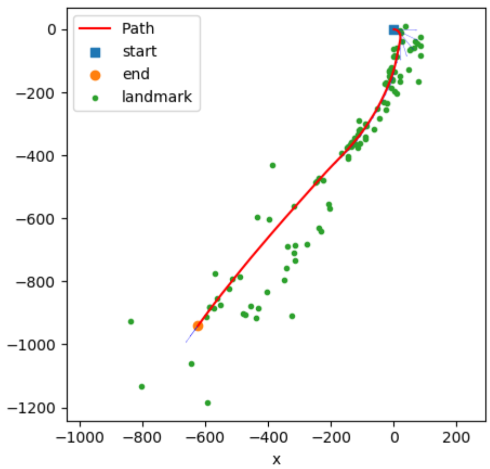
SLAM

After applying the SLAM, the trajectory misses the start point. Comparing the result to formal answer, we find little difference. We could notice that during the moving, we never get the same landmark again when we miss it. So, the offset may be caused by the landmarks which change quickly during turning at a corner.

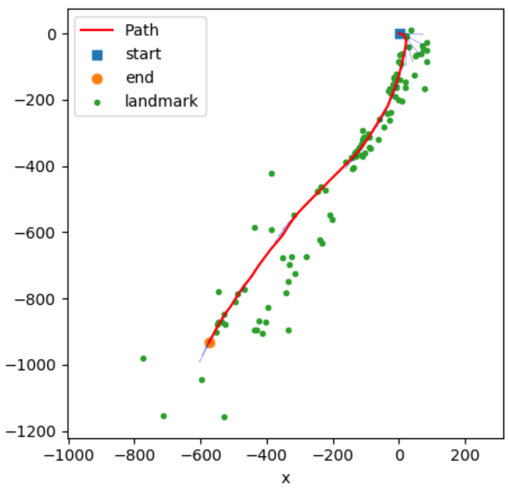
dataset:42



prediction



mapping



SLAM

We can see that after applying the SLAM, the trajectory becomes noise but we could still go to the end. An interesting thing is that on half way, a moving car is labeled as landmark and it would lead to a turn in trajectory. We could set some limitation to the updating and get ride of the impact of the moving car.

Conclusion:

We could see that the prediction and mapping could work well separately. However, when we combine them with an updating for the pose, the trajectory may some time be wrong.

It may be caused by the wrong parameter we set(covariance) or some unexpected event happen in the camera.