ECE276A: Visual Intertial SLAM

Ashish Maknikar
PID: A59018297
UC San Diego

***Abstract*—A car is made to move in an outdoor environment and data from sensors mounted on the robot is captured. Inertial and visual data is captured from the onboard IMU and camera respectively. This project aims to use the Extended Kalman filter to perform visual inertial SLAM on the car motion.**

I. Introduction

The current experiment is an apparent and common manifestation of the SLAM problem in modern cars. Cars fitted with autonomous driving capabilities are required to perform SLAM by relying on visual data in conjunction with other sensors. The image features are provided in the project, reducing the task of extracting and tracking features of the car in motion.

Our experiment involves the motion of a car to collect visual-inertial data. The following sensors are used to collect data:

- The IMU collects the linear and angular velocity data of the robot.
- An onboard camera captures images with the same frequency as the IMU. The features are extracted and provided to us.

This data gives us sufficient information about the car motion and its environment such that SLAM can be performed. We can surmise the pose and location of the robot from the IMU data. We are able to perform basic dead reckoning of the car trajectory based on the IMU data alone. The stereo camera features allow us to map the environment and simultaneously improve the precision of trajectory prediction of the car by incorporating the visual features and their subsequent changes captured during the motion of the car. The features captured in the stereo camera(as opposed to a single frame camera) can be extrapolated to 3D cartesian coordinates.

II. Problem formulation

ALSO CONTAINS PARTS OF TECHNICAL APPROACH since the problem cannot be formulated succinctly without part of the technical approach

After analysing the sensor data (explained in the technical approach) at time t, we obtain the following:

- time interval τ_t between time step t and $t+1$
- robot linear velocity at time t v_t
- the robot yaw angular velocity at time t ω_t
- coordinates of all the image features/landmark points through the trajectory $m \in R^{3\times M}$.

We know the following from the robot configuration:

- Intrinsic Calibration Matrix of the camera ${}_{IMU}T_{camera}$
- The horizontal (x) displacement between the two frames of the stereo camera b

A. Motion Model Prediction Step Dead Reckoning

We use nominal kinematics and perturbation kinematics with time discretization as the motion model,

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

$$\Sigma_{t+1|t} = exp(-\tau_t \hat{\hat{u}}_t)\, \Sigma_{t+1|t}\, exp(-\tau_t \hat{\hat{u}}_t)^T + W \tag{2}$$

where:

$$u_t = \begin{bmatrix} v_t \\ \omega_t \end{bmatrix}$$

$$\hat{u}_t = \begin{bmatrix} \hat{\omega}_t & v_t \\ 0^T & 0 \end{bmatrix}$$

$$\hat{\hat{u}}_t = \begin{bmatrix} \hat{\omega}_t & v_t \\ 0^T & 0 \end{bmatrix}$$

where x_t represents the robot position on map at tiem t and W is the covariance matrix associated with the motion model noise w_t.
For dead reckoning without any landmarks:

$$x_{t+1|t+1} = x_{t+1|t} \tag{3}$$

$$\Sigma_{t+1|t+1} = \Sigma_{t+1|t} \tag{4}$$

B. Visual Mapping Only

For Part 2, we consider only the mapping problem. We have the observation model

$$\tilde{z}_{t+1,i} = K_s \pi({}_O T_I\, T_{t+1}^{-1} \underline{\mu_{t,j}}) \tag{5}$$

where μ_t is the prior known positions of the landmark features. We can then calculate the Jacobian as

$$H_{t+1,i,j} = \begin{cases} K_s \frac{d\pi}{dq} \pi({}_O T_I\, T_{t+1}^{-1} \underline{\mu_{t,j}})\, {}_O T_I\, T_{t+1}^{-1} & , if \Delta_t(j) = i \\ 0, \;\; otherwise \end{cases} \tag{6}$$

Where $\Delta_t(j)$ maps points from t to $t+1$ and $i \in [1, N_{t+1}]$, N_{t+1} is the number of landmark features in the current frame.

The following equations are used to estimate the mapping of the landmark features using EKF:

$$K_{t+1} = \Sigma_t H_{t+1}^T (H_{t+1} \Sigma_t H_{t+1}^T \;+\; I)^{-1} \tag{7}$$

$$\mu_{t+1} = \mu_t + K_{t+1}(z_{t+1} - \tilde{z}_{t+1} \tag{8}$$

$$\Sigma_{t+1} = (I - K_{t+1} H_{t+1}) \Sigma_t \tag{9}$$

C. EKF Update Step with Visual Features

We use the 1 to determine $x_{t+1|t+1}$ and $\Sigma_{t+1|t+1}$.

We use the observation model as referred to in (5)

$$\tilde{z}_{t+1,i} = K_s \pi({}_OT_I\ x_{t+1}^{-1}\underline{\mu_{t,j}}) \tag{10}$$

Jacobian of $\tilde{z}_{t+1,i}$ gives us:

$$H_{t+1,i} = -K_s \frac{d\pi}{dq}\pi({}_OT_I\ x_{t+1}^{-1}\underline{\mu_{t,j}})\,{}_OT_I(\mu_{t+1|t}^{-1}\underline{\mu_{t,j}})^{\odot} \tag{11}$$

where:

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

The update steps are:

$$K_{t+1} = \Sigma_t H_{t+1}^T (H_{t+1}\Sigma_t H_{t+1}^T \ + \ I)^{-1} \tag{12}$$
$$x_{t+1} = x_t\ exp((K_{t+1}(z_{t+1} - \tilde{z}_{t+1})^{\Lambda}) \tag{13}$$
$$\Sigma_{t+1} = (I - K_{t+1}H_{t+1})\Sigma_t \tag{14}$$

III. Technical Approach

A. Data processing

The data from the inputs is processed as follows:

- **IMU Features and Visual Landmark features**: are provided to us and no processing is required. We can ignore the y and z components of the linear velocity since it is not substantial since the car only moves forward in its own frame. 1 and 2 show the recoreded linear and angular velocities respectively.

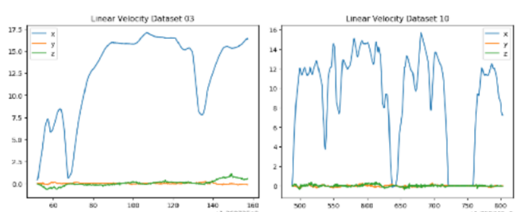

Fig. 1. Linear Velocity from IMU

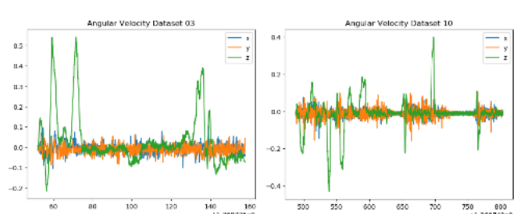

Fig. 2. Angular Velocity from IMU

- K_s: We can calculate the K_s matrix from the given values of K and b using the following equations

$$K = \begin{bmatrix} f\ s_u & 0 & c_u \\ 0 & f\ s_v & c_v \\ 0 & 0 & 1 \end{bmatrix} \tag{15}$$
$$K_s = \begin{bmatrix} f\ s_u & 0 & c_u & 0 \\ 0 & f\ s_v & c_v & 0 \\ f\ s_u & 0 & c_u & -f\ s_u b \\ 0 & f\ s_v & c_v & 0 \end{bmatrix} \tag{16}$$

B. Dead Reckoning

To begin, we will first move the robot as a single entity and use the angular and linear velocity measurements without any noise. This ensures that our code will work properly. The robot orientation and position is updated at each time step according to 1 and 3. We then plot the same for the two data sets.

C. Visual Mapping

We follow the equations in 5 to get μ which represents the final predicted locations of the points in 3D space. We discard the z coordinates and plot the points.

IV. Results

A. Dead Reckoning

The car trajectory generated for dead reckoning is shown in **??** and **??** on Dataset 10 and 03 respectively.

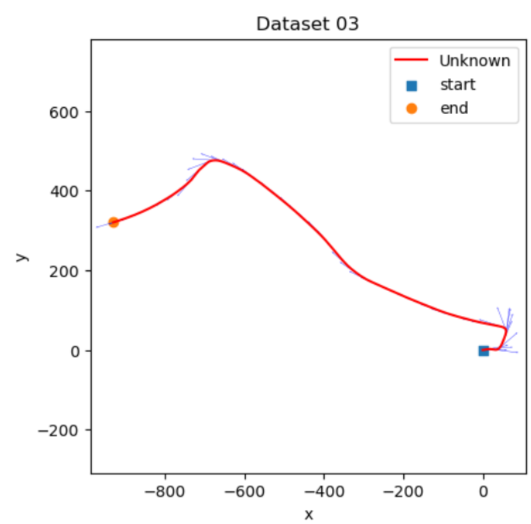

Fig. 3. Dead Reckoning for Dataset 03

B. Visual Mapping

We perform visual Mapping as shown in 5 and 6.

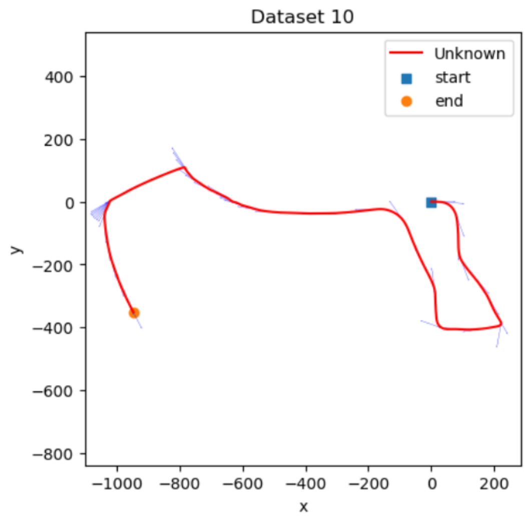

Fig. 4. Dead Reckoning for Dataset 10

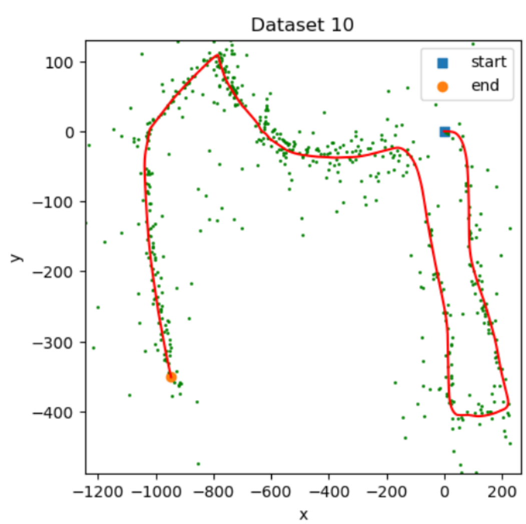

Fig. 5. EKF for Dataset 3

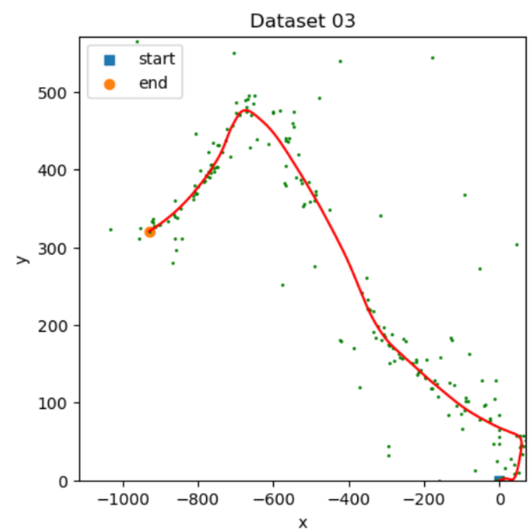

Fig. 6. EKF for Dataset 10