Visual Inertial SLAM

1st Marcus Schaller
Department of Electrical and Computer Engineering (ECE 276A)
University of California San Diego
La Jolla, U.S.A
mschalle@ucsd.edu

***Abstract*—Simultaneous Localization and Mapping or SLAM plays a crucial role in navigating for autonomous vehicles. This process allows cars to quickly create a map of their surroundings and correct for errors created by their on-board sensors by localizing themselves on the map. There are a number of techniques used in SLAM to localize a robot, and one common technique discussed in this paper is Visual Inertial SLAM. This paper will discuss how the data extracted from a stereo camera and an IMU unit is used to estimate a vehicle's position.**

I. Introduction

Autonomous vehicles use a variety of techniques to determine their position. One common technique used is known as Simultaneous Localization and Mapping or SLAM. This technique involves using sensors to gather information about the vehicles surroundings and using SLAM algorithms to generate a map of the vehicles surroundings and localize its position on the map. As a result, this creates a sort of chicken and egg problem as both localization and mapping is important when used in this problem. [1] This paper will discuss how this problem can be solved using a technique known as Visual Inertial SLAM. This techniques involves using a stereo camera to track landmarks around a vehicle and an Inertial Measurement Unit or IMU to track the vehicle's motion.

II. Problem Formulation

As stated previously SLAM is a chicken-and-egg problem. This means that often a vehicle that is utilizing SLAM has to generate a map while at the same time localizing itself within that map. In order to address this problem a dataset must be identified and used to map and localize the vehicle through the use of SLAM.

A. Problem Data-set

The problem proposed in this project involves a data-set which provides the motion data from an IMU sensor. The IMU data given is the linear velocity which can be expressed as $v_t \in \mathbb{R}^3$ and $\omega_t \in \mathbb{R}^3$ respectively. In addition landmark pixel positions given in the optical frame stereo camera which can be expressed as (u_L, v_L) and (u_R, u_L). For the purpose of SLAM these features can be expressed as $\mathbf{z}_t \in \mathbb{R}^{4\times N_t}$. In addition a the baseline of the stereo camera *b* is given as well as the camera's calibration matrix *K* which provides fs_u, fs_v, c_u, and c_v. Finally in order to track the vehicles motion at the IMU the transformation from the left camera to the IMU frame is given as ${}_IT_c \in SE(3)$ An example of the visual features that are provided in each left and right camera frame can be seen in figure 1 and 2 respectively.

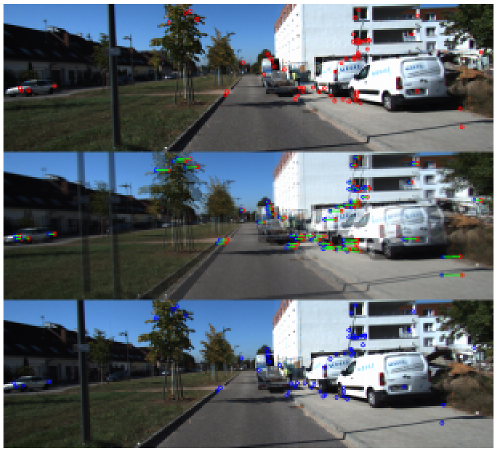

Fig. 1. Left Camera Features

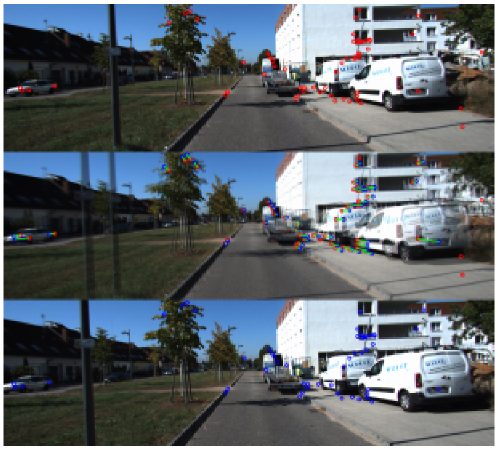

Fig. 2. Right Camera Features

B. Localization and Mapping

Localization and Mapping involves creating a motion and observation model that can be used to estimate the vehicles position. The motion model which is used to localize and map the vehicle can be represented as the following:

$$\mathbf{x}_{t+1}=f\left(\mathbf{x}_t,\mathbf{u}_t,\mathbf{w}_t\right),\quad \mathbf{w}_t\sim\mathcal{N}(\mathbf{0},W) \tag{1}$$

With x_t signifying the vehicles state, u_t signifying the vehicles control input, and w_t representing vehicle noise.

The observation model problem (which helps to make adjustments to the vehicles estimated position) can be stated as: [3]:

$$\mathbf{z}_{\mathrm{t}}=h\left(\mathbf{x}_t,\mathbf{v}_t\right),\quad \mathbf{v}_{\mathrm{t}}\sim\mathcal{N}(\mathbf{0},V) \tag{2}$$

With z_t representing landmarks observed, x_t once again vehicle state, and v_t representing observation noise.

These technique that will be used in the project is known as the Extended Kalman Filter or EKF. It will use the motion and observation models to predict the location of the vehicle based on the positions of the landmarks.

III. Technical Approach

To develop a Visual Inertial SLAM model it was necessary to first localize the IMU, followed by landmark mapping, and finally combining these elements to create a SLAM model. These steps were completed as follows.

A. IMU- based Localization via EKF Prediction

At this point, the IMU data is used to predict the mean of the pose of the vehicle at any time *t*. Given the prior $T_t \mid \mathbf{z}_{0:t},\mathbf{u}_{0:t-1}\sim\mathcal{N}\left(\boldsymbol{\mu}_{t|t},\Sigma_{t|t}\right)$ with $\boldsymbol{\mu}_{t|t}\in SE(3)$ and $\Sigma_{t|t}\in\mathbb{R}^{6\times6}$. It is safe to assume that T_0 starts at the origin meaning it is equal to I. From there the prediction step is solved using the following equations:

$$\mu_{t+1|t}=\exp\left(-\tau\hat{u}_t\right)\mu_{t|t} \tag{3}$$

$$\Sigma_{t+1|t}=\exp\left(-\tau\hat{\hat{\mathbf{u}}}_t\right)\Sigma_{t|t}\exp\left(-\tau\hat{\hat{\mathbf{u}}}_t\right)^{\top}+W \tag{4}$$

where:

$$\mathbf{u}_t:=\begin{bmatrix}\mathbf{v}_t\\ \omega_t\end{bmatrix} \tag{5}$$

Knowing $\hat{\omega}_t$ is the skew symmetric matrix of ω_t as well as $\hat{v}_t$ is the skew symmetric matrix of v_t one can calculate the following matrices which are used in the previous equation.

$$\hat{\mathbf{u}}_t:=\begin{bmatrix}\hat{\omega}_t & \mathbf{v}_t\\ \mathbf{0}^{\top} & 0\end{bmatrix}\in\mathbb{R}^{4\times4} \tag{6}$$

$$\hat{\hat{\mathbf{u}}}_t:=\begin{bmatrix}\hat{\omega}_t & \hat{\mathbf{v}}_t\\ 0 & \hat{\omega}_t\end{bmatrix}\in\mathbb{R}^{6\times6} \tag{7}$$

B. Landmark Mapping via EKF Update

Once the IMU pose is known it is used to estimate the positions of the landmarks around the vehicle. The first step is to initialize mean $\mu\in\mathbb{R}^{3M}$ and covariance $\Sigma\in\mathbb{R}^{3Mx3M}$. This is done when each landmark *N* is first seen by the vehicle. The mean is initialized to be the position of the landmark calculated when it is first seen and the covariance is the identity matrix multiplied by a random scalar noise. In order to calculate the position of the landmark the x,y, and z positions are calculated using the calibration matrix K.

$$z=\frac{K_{11}b}{u_L-u_R} \tag{8}$$

$$y=z\left(\frac{v_L-K_{23}}{K_{22}}\right) \tag{9}$$

$$x=z\left(\frac{u_L-K_{13}}{K_{11}}\right) \tag{10}$$

The next step is to update the mu and covariance every time the same landmark is seen in future timestamps. However, the following matrix elements must first be setup. The first of these is the calibration matrix *M*

$$M:=\begin{bmatrix}f_{S_u} & 0 & c_u & 0\\ 0 & f_v & c_v & 0\\ f_u & 0 & c_u & -f_{s_u}b\\ 0 & f_{s_v} & c_v & 0\end{bmatrix} \tag{11}$$

The predicted observations based on μ_t is calculated as:

$$\tilde{\mathbf{z}}_{t+1,i}:=M_{\pi}\left({}_OT_IT_{t+1}^{-1}\underline{\mu}_{t,j}\right)\in\mathbb{R}^4\quad\text{for } i=1\ldots..N_{t+1} \tag{12}$$

Next the observation model Jacobian must be calculated as the sparse matrix:

$$H_{t+1.i,j}=\begin{cases}M\frac{d\pi}{dq}\left({}_OT_IT_{t+1}^{-1}\underline{\mu}_{t.j}\right)*_O T_IT_{t+1}^{-1}P^T & \text{if } \Delta_t(j)=i\\ 0.\in\mathbb{R}^{4\times3} & \text{otherwise}\end{cases} \tag{13}$$

In order to calculate the previous values it is important to know how to calculate the projection function and its derivative.

$$\pi(\mathbf{q}):=\frac{1}{q_3}\mathbf{q}\in\mathbb{R}^4 \tag{14}$$

$$\frac{d\pi}{d\mathbf{q}}(\mathbf{q})=\frac{1}{q_3}\begin{bmatrix}1 & 0 & -\frac{q_1}{q_3} & 0\\ 0 & 1 & -\frac{q_2}{q_3} & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & -\frac{q_4}{q_3} & 1\end{bmatrix}\in\mathbb{R}^{4\times4} \tag{15}$$

Once both of these are calculated the updated Kalman gain, mean, and covariance can be calculated using the following equations.

$$K_{t+1}=\sum_t H_{t+1}^T\left(H_{t+1}\Sigma_tH_{t+1}^T+I\otimes V\right)^{-1} \tag{16}$$

$$\mu_{t+1} = \mu_t + K_{t+1}\left(z_{t+1} - \tilde{z}_{t+1}\right) \tag{17}$$

$$\Sigma_{t+1} = \left(I - K_{t+1}H_{t+1}\right)\Sigma_t \tag{18}$$

The mean and covariance is calculated for each time step and the final positions of the landmarks can be estimated as the mean that was calculated for each landmark. In order to reduce computational time the landmark dataset can be reduced to include a smaller set of landmarks. Due to time constraints this was done in this project.

C. Visual-Inertial SLAM

The final step in the SLAM process involves combining the previous two steps. One key difference is that now the landmark positions m are known. If completed correctly this should provide a more accurate map of the robots path. The first step is to once again initialize the vehicle mean μ_0 and calculate the predicted observation based on the $\mu t + 1|t$ that can be calculated from equation 5.

$$\tilde{\mathbf{z}}_{t+1,i} := M\pi\left(oT_I\mu^{-1}_{t+1|t}\underline{\mathbf{m}}_j\right) \tag{19}$$

A key difference is that the Jacobian is now $H \in \mathbb{R}^{4N_t\times(3M+6)}$ as the observation matrix calculated as following is now appended to the end of the observation matrix found in equation (15).

$$H_{t+1,i} = -M\frac{d\pi}{d\mathbf{q}}\left(oT_I\mu^{-1}_{t+1|t}\underline{\mathbf{m}}_j\right)oT_I\left(\mu^{-1}_{t+1|t}\underline{\mathbf{m}}_j\right)^{\odot} \in \mathbb{R}^{4\times6} \tag{20}$$

In order to calculate the previous equation one must know the following:

$$\hat{\boldsymbol{\xi}}\underline{\mathbf{s}} = \underline{\mathbf{s}}^{\odot}\boldsymbol{\xi} \quad \begin{bmatrix} \mathbf{s} \\ 1 \end{bmatrix}^{\odot} := \begin{bmatrix} 1 & -\hat{\mathbf{s}} \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{4\times6} \tag{21}$$

From there the observation matrix can be formed by stacking each landmark's $H_{t+1,i}$ and performing the EKF update step.

$$\begin{aligned} K_{t+1} &= \Sigma_{t+1|t}H^{\top}_{t+1}\left(H_{t+1}\Sigma_{t+1|t}H^{\top}_{t+1} + I\otimes V\right)^{-1} \\ \mu_{t+1|t+1} &= \mu_{t+1|t}\exp\left(\left(K_{t+1}\left(z_{t+1} - \tilde{z}_{t+1}\right)\right)^{\wedge}\right) \\ \Sigma_{t+1|t+1} &= \left(I - K_{t+1}H_{t+1}\right)\Sigma_{t+1|t} \end{aligned} \tag{22}$$

Once this is completed for every time step the mean can be plotted for all of the landmarks and the IMU poses to generate a map of the vehicles motion and observations, therefore completing visual inertial SLAM.

IV. Results

A. IMU-based Localization results

The vehicle pose path was created using only the IMU-based Localization EKF predict step and can be seen in Fig. 3.

B. Landmark Mapping via EKF Update

Using the EKF update step the landmarks on the map in Fig. 4 was created.

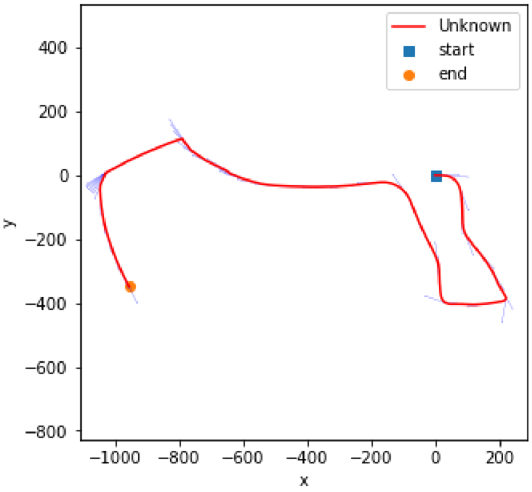

Fig. 3. Vehicle Path (meters)

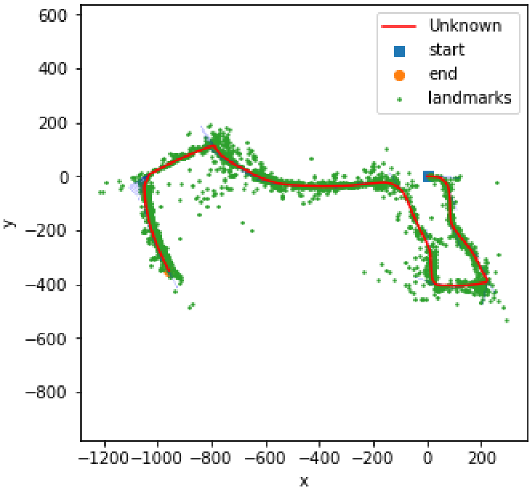

Fig. 4. Vehicle Path with landmarks (meters)

C. Visual Inertial SLAM

Combining steps 1 and 2 the figures 5 and 6 were created showing the path of the vehicle and comparing it to the plot in figure 4. A video of the path was included with the project submission

D. Possible improvements

One major improvement that was made in order to run the SLAM algorithm in a timely manner was to reduce the number of landmarks that were used in the algorithm. Only every 10th landmark was used resulting in a much quicker but less accurate SLAM map. If a method was implemented to only update landmarks that were newly initialized this would result in a much faster algorithm. Additionally, it was found that using very high V values was the only way to successfully generate the pose. Given more time further research should be done into this.

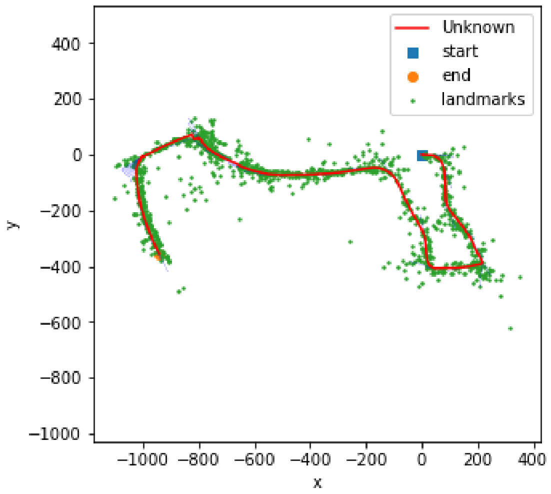

Fig. 5. Vehicle Path with landmarks via SLAM (meters)

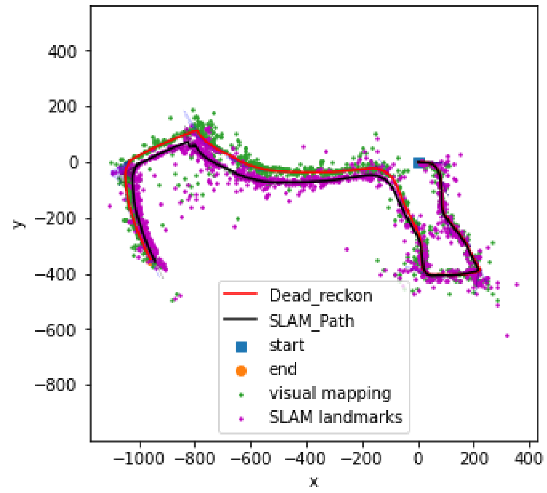

Fig. 6. Vehicle Path with landmarks compared Dead-reckon with SLAM (meters)

Acknowledgment

Thank you Professor Atanasov as well as Mo and Arash. I appreciate all the time and effort you spent on putting this project together as well as all the time you put into answering questions on Piazza.

References

[1] D. Agarwal, “How SLAM Works For Self-Driving Cars: A Brief But Detailed Overview,” AutoVision News, 05-Jun-2020. [Online]. Available: https://www.autovision-news.com/sensing/how-slam-works/. [Accessed: 21-Feb-2021].

[2] N. Atanasov, “Lecture 13: Visual-Inertial SLAM,” in ECE276A: Sensing amp; Estimation in Robotics, 01-Mar-2021.

[3] N. Atanasov, “Lecture 10: EKF,UKF,” in ECE276A: Sensing amp; Estimation in Robotics, 17-Feb-2021.