Visual-Inertial SLAM

Parthasarathi Kumar
Department of Electrical and Computer Engineering
University of California, San Diego
California, USA
pakumar@ucsd.edu

I. Introduction

Simultaneous Localization and Mapping or SLAM is an important problem in the field of autonomous robots where the agent has to map an unknown environment and at the same time localize itself in the map. This problem has many applications in the area of navigation, path planning, high definition mapping, etc. SLAM is generally done using a combination of sensors like camera, LiDARs, GPS, IMU, etc and involves a probabilistic approach of combining these sensor inputs.

In this paper we approach the SLAM problem on a dataset captured by an autonomous car in an urban environment equipped with a stereo camera and an inertial measurement unit. We use the detected landmarks in the stereo images along with the IMU measurements to accurately trace the path of the robot and simultaneously localize the landmarks in the map.

II. Problem Formulation

Given the IMU readings corresponding to the linear velocity v & angular velocity w and the associated landmark features z in the 2 images, the problem of visual inertial SLAM involves finding the robot pose T and the landmark positions, both in the world co-ordinate frame m. This can be stated mathematically as finding the joint distribution of the pose T and landmark positions m, conditioned on the IMU readings v & w, and detected features z -

$$p(x_{0:t}, m_{0:t} | z_{0:T}, u_{0:T-1}) \tag{1}$$

$$x_t = \begin{bmatrix} \rho \\ \theta \end{bmatrix} \in R^6$$

$$m_t \in R^{3 \times M}, z_t \in R^{4 \times N_t}$$

$$u_t = \begin{bmatrix} v \\ w \end{bmatrix} \in R^6$$

where M is the total number of landmarks across the sequence, N_t is the number of landmarks detected in the stereo image pair. Each detected feature is a 4-vector.

To solve this we leverage the motion model and observation model. The motion model describes the state of agent given the previous state and control input as in (2) where f defines the motion model and w_t is the motion noise.

$$x_{t+1} = f(x_t, u_t, w_t) \sim p_f(.|x_t, u_t) \tag{2}$$

$$z_t = h(x_t, m_t, v_t) \sim p_h(.|x_t, m_t) \tag{3}$$

The observation model helps model the surroundings based on the observation z conditioned on the observation model h and observation noise v_t as in (3)

Using (2) and (3) and applying on (1) under the Markov assumption, we can break down the problem into the following equation

$$\begin{gathered} p(x_{0:T}, m | z_{0:T}, u_{0:T-1}) = p(x_0, m_0) \\ \prod_{t=0}^{T} p_h(z_t|x_t) \prod_{t=1}^{T} p_f(x_t|x_{t-1}, u_{t-1}) \end{gathered} \tag{4}$$

III. Technical Approach

A. Bayesian Filtering

Applying Bayesian filtering, we obtain the following general equations for the predict (2) and update (3) step respectively.

$$p_{t+1|t}(x) = \int p_f(x|s, u_t) p_{t|t}(s) ds \tag{5}$$

$$p_{t+1|t+1}(x) = \frac{p_h(z_{t+1}|x) p_{t+1|t}(x)}{\int p_f(z_{t+1}|s) p_{t+1|t}(s) ds} \tag{6}$$

B. States

In our problem, the state can be divided into 2 parts, the robot pose and the landmark points, both in the world co-ordinate frame. The robot pose can be expressed as 6-vector $\xi \in R^6$ with position and orientation or alternately as T ∈ SE(3) i.e. space of pose matrices.

$$\xi = \begin{bmatrix} \rho \\ \theta \end{bmatrix}, T = exp(\hat{\xi}) \tag{7}$$

$$\hat{\xi} = \begin{bmatrix} \hat{\theta} & \rho \\ 0 & 0 \end{bmatrix}, \hat{x} = \begin{bmatrix} 0 & -x_3 & a_2 \\ x_3 & 0 & -x_2 \\ -x_2 & a_1 & 0 \end{bmatrix}$$

The landmarks are represented as -

$$m \in R^{3*M}, \tag{8}$$

where M is the number of landmarks. Each state variable is modeled as a Gaussian random variable with its prior or initial value as the mean and an associated covariance -

$$x_{t|t} \sim \mathcal{N}(\mu_{t|t}, \Sigma_{t|t}) \tag{9}$$

This leads to the Kalman filter and it's extensions to formulate the prediction and update equation.

C. Extended Kalman Filter - General Equations

The Kalman filter exploits the assumption of linearlity in state and independence of motion and observation noise, along with the properties of the Gaussian distribution to come up with the prediction and update step. However, out state is not linear and hence we use the Extended Kalman Filter to perform the prediction and update steps.

Let the motion model be defined as follows -

$$x_{t+1} = f(x_t, u_t, w_t), w_t \sim \mathcal{N}(0, W) \tag{10}$$

and the observation model as -

$$z_t = f(x_t, v_t), v_t \sim \mathcal{N}(0, V) \tag{11}$$

Using a Taylor series approximation to linearize the 2 models, we have the following prediction equations -

$$\begin{gathered} \mu_{t+1|t} = f(\mu_{t|t}, u_t, 0) \\ \Sigma_{t+1|t} = F_t \Sigma_{t|t} F_t^T + Q_t W Q_t^T \end{gathered} \tag{12}$$

where ,

$$F_t = \frac{df}{dx}(\mu_{t|t}, u_t, 0)$$

$$Q_t = \frac{df}{dw}(\mu_{t|t}, u_t, 0)$$

Similarly, we get the following update equations

$$\begin{aligned} \mu_{t+1|t+1} &= \mu_{t+1|t} + K_{t+1|t}(z_{t+1} - h(\mu_{t+1|t}, 0) \\ \Sigma_{t+1|t+1} &= (I - K_{t+1|t} H_{t+1}) \Sigma_{t+1|t} \\ K_{t+1|t} &= \Sigma_{t+1|t} H_{t+1}^T (H_{t+1} \Sigma_{t+1|t} H_{t+1}^T \\ &\qquad + R_{t+1} V R_{t+1}^T)^{-1} \end{aligned} \tag{13}$$

where ,

$$H_{t+1} = \frac{dh}{dx}(\mu_{t+1|t}, 0)$$

$$R_{t+1} = \frac{df}{dv}(\mu_{t+1|t}, 0)$$

D. Motion Model

As stated before we describe the pose of the robot by position and orientation. This is equivalent to a 4x4 pose matrix in the SE(3) space as described by (7). The motion model consists of linear and angular velocity as described in (1). The overall discrete time pose kinematics relating the robot pose with the control inputs is as follows -

$$T_{k+1} = T_k exp(\tau_k \hat{\xi_k}) \tag{14}$$

where ,

$$\hat{\xi_t} = \begin{bmatrix} \hat{\omega_t} & \mathrm{v}_t \\ 0^T & 0 \end{bmatrix}$$

Under the Kalman filter setting, we model the pose as a multivariate Gaussian with an associated mean $\mu_{t|t} \in R^{4x4}$ or SE(3) and covariance $\Sigma_{t|t} \in R^{6x6}$

$$T_t | z_{0:t}, u_{0;T} \sim \mathcal{N}(\mu_{t|t}, \Sigma_{t|t}) \tag{15}$$

To ensure all changes in T satisfy the SE(3) condition, we use nominal kinematics for mean of the pose and perturbation kinematics for the associated variance-

$$\begin{aligned} \mu_{t+1|t} &= \mu_{t|t} exp(\tau_t \hat{u}) \\ \delta\mu_{t+1|t} &= exp(-\tau_t \hat{u}_t) \delta\mu_{t|t} \\ &\qquad + w_t \end{aligned} \tag{16}$$

where,

$$w_t = \mathcal{N}(0, W)$$

is the motion noise.

E. Observation Model

The observation model consists of a fixed set of landmark points detected in a stereo image pairs. We use an external algorithm to associate the detected features in the image with the corresponding landmark points. The observation model is based on the stereo camera projection matrix as follows -

$$\begin{gathered} z_{t,i} = h(T_t, m_j) + v_{t,i}, v_{t,i} = \mathcal{N}(0, V) \\ z_{t,i} = K_s \pi(T_{I->O} T_{W->I} \underline{m}_j) + v_{t,i} \\ \pi(q) = \frac{q}{q_3}, \underline{m}_j = \begin{bmatrix} m_j \\ 1 \end{bmatrix} \end{gathered} \tag{17}$$

where v is the observation noise. Ks is the stereo camera projection matrix as described in (18).

$$K_s = \begin{bmatrix} fs_u & 0 & c_u & 0 \\ 0 & fs_v & c_v & 0 \\ 0 & 0 & c_u & -fs_u b \\ 0 & fs_v & c_v & 0 \end{bmatrix} \tag{18}$$

where f is the focal length, b is the stereo baseline , s_u and s_v are the pixel scaling and c_u and c_v are the principal point.

F. Visual Mapping

We first approach the visual mapping problem that involves locating the feature points in the images to the landmark points using the observation model described earlier. Here we use odometry to determine the intermediate car pose i.e. we apply the pose kinematics equation (14).

We further make the assumption that the landmark points are stationary and hence don't incorporate a EKF predict step. The EKF update step involves the EKF update equation as described in 13.

The state variable, i.e. the landmark position $\in R^{3M}$ and associated covariance $\in R^{3Mx3M}$. The observation model Jacobian w.r.t state variable $\in R^{4N_tx3M}$ given by -

$$H_{t+1,i,j} = \begin{cases} \frac{\partial}{\partial m_j} h(T_{t+1}, m_j) & \text{,if } \Delta_t(j) = i, \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

where,

$$\frac{\partial}{\partial m_j} h(T_{t+1}, m_j) = K_s \frac{\partial \pi}{\partial q}(T_{I->C} T_{W->I} \underline{m}_j) \\ T_{I->C} T_{W->I} P^T$$

Note that the transformation from world to IMU corresponds to the current time stamp.

G. Visual Inertial SLAM

We extend the visual mapping to incorporate the robot pose in the state variable. We will first describe the EKF update and predict step for the pose only case, i.e. state variable involving only the robot pose.

The EKF prediction step extends (16) to -

$$\begin{aligned} \mu_{t+1|t} &= \mu_{t|t} exp(\tau_t \hat{u}) \\ \delta\mu_{t+1|t} &= exp(-\tau \overset{\curlywedge}{u}) \Sigma_{t|t} exp(-\tau \overset{\curlywedge}{u})^T + W \end{aligned} \tag{20}$$

where,

$$\overset{\curlywedge}{u} = \begin{bmatrix} \hat{\omega}_t & \hat{v}_t \\ 0 & \hat{\omega}_t \end{bmatrix}$$

The EKF update step for pose only also has the same form as (13) like that of the visual mapping. The only difference is that the Jacobian w.r.t state variable $\in R^{4N_tx6}$ given by -

$$H_{t+1,i} = \begin{bmatrix} H_{t+1,1} \\ . \\ . \\ . \\ H_{t+1,N_{t+1}} \end{bmatrix}, H_{t+1,i} = \\ K_s \frac{\partial \pi}{\partial q}(T_{I->C} T_{W->I} \underline{m}_j) T_{I->C}`(T_{I->W} \underline{m}_j)^{\bullet} \tag{21}$$

where for a homogeneous vector s,

$$\begin{bmatrix} s \\ 1 \end{bmatrix}^{\bullet} = \begin{bmatrix} I & -\hat{S} \\ 0 & 0 \end{bmatrix}$$

To combine the 2 state variables, we merge the covariances into a single $(3M + 6) \times (3M + 6)$ matrix and the Kalman gain into a $(3M + 6) \times 4N_t)$ matrix. In the covariance matrix the diagoanl consists of the individual covarainces as described earlier. The cross diagonal terms correspond to the correlation between the 2 and is initialized by 0s.

H. Algortihm and Implementation details

The dataset we are working on is time synchronised across all the sensors. For each timestamp, we perform the EKF prediction on the robot pose and the joint update on the robot pose and landmark points. The state variable consisting of the pose and landmark points are maintained separately but updated simultaneously.

We initialize the landmark positions with NaN values. The initial pose is set at 0 except the roll which is set as 180 degrees to account for the flipped IMU. As we observe new points, their position and covariances are initialized. The points that are re-observed are used in the update step of the EKF. The landmark points of only these points are updated in the EKF update step.

Since we have a large number of landmark points detected in each stereo image pair we sample a subset of these points in order to ensure a decent run time for the algorithm.

IV. Results

A. Qualitative Results

Figure 1 and 2 shows the dead-reckoning , visual mapping and visual inertial SLAM for the 2 datasets. As evident, the path traced by the car in the dead reckoning is same as the visual mapping. However it is intersting to note the difference in the car trajectory and the landmark points as shown in Figure 3.

As the IMU measurements are noisy we see a significant drift in the path obtained from odometry. Incorporating the landmarks to correct the path improves the trajectory significantly.

It is however, worthwhile to note that the noise value set for the motion model and observation model play a significant role in the accuracy of the VI-SLAM trajectory as evident from the plot for dataset 2 (10.npz).

B. Parameter setting

We used observation noise of with variance of 1 and 0.3 respectively for the linear and angular velocity. For the motion model noise we use a zero mean gaussian model with varaince of 10.

References

[1] ECE 276A Slides 11,12,13

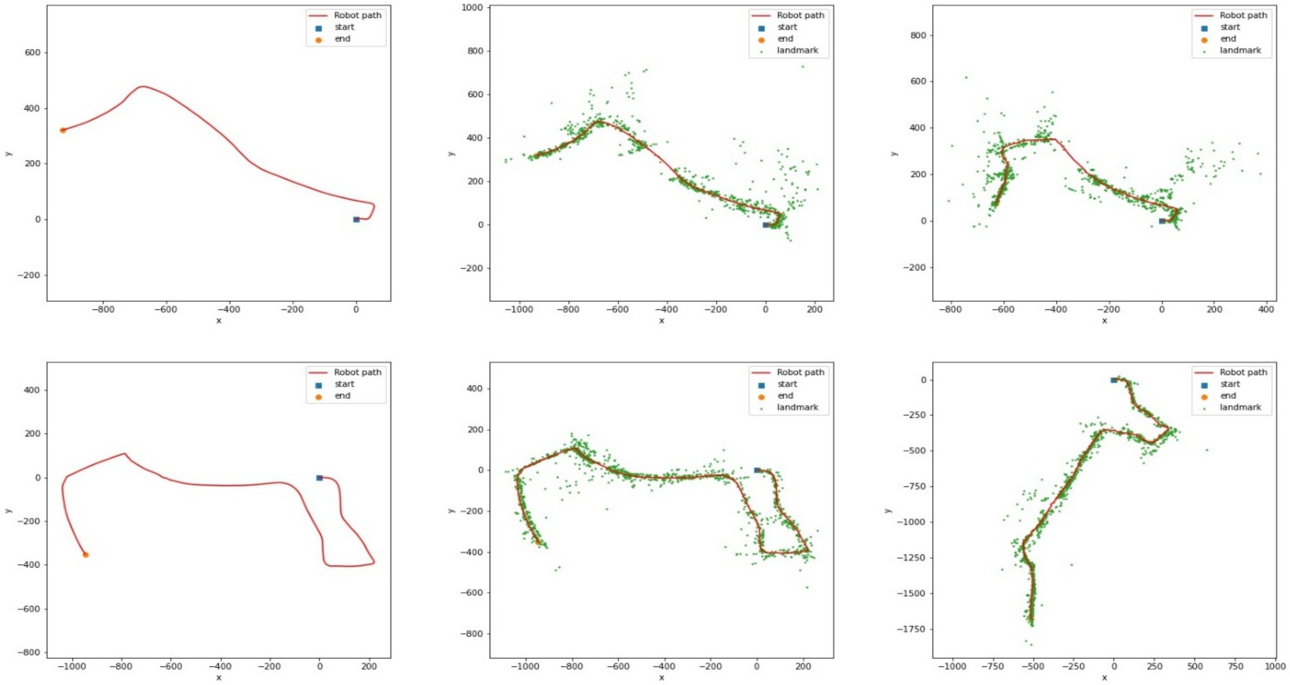

Fig. 1. Dead-reckoning, visual mapping and visual inertial SLAM on 03.npz and 10.npz

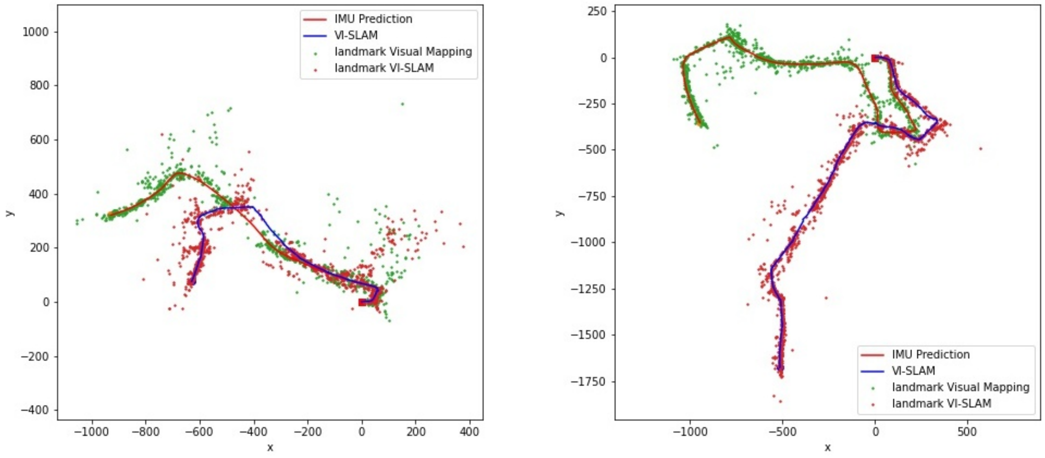

Fig. 2. Visual mapping and VI-SLAM superimposed on each other.