Simultaneous Localization and Mapping

Steve Kim
Department of Electrical Computer Engineering
University of California, San Diego
stk002@eng.ucsd.edu

***Abstract*—This paper reviews the process of Simultaneous Localization and Mapping, aka SLAM, via Bayesian Filtering implemented with a Particle filter. We show how various sensors on a vehicle (car) can map its surroundings as well as its location within it over a period of time.**

***Index Terms*—SLAM, Particle Filter, Bayesian Filter**

I. Introduction

Consider an autonomous vehicle. It wants to navigate the environment, but has two problems - it needs to know its surroundings, and it needs to know its own location within the environment. At first glance, these two issues may seem like separate problems. However, through Simultaneous Localization and Mapping, we can show that in fact, these two problems can be solved at the same time. In this paper, we review a Bayesian filter method that is able to map the environment around a car, while at the same time, keep track of where it is on the road.

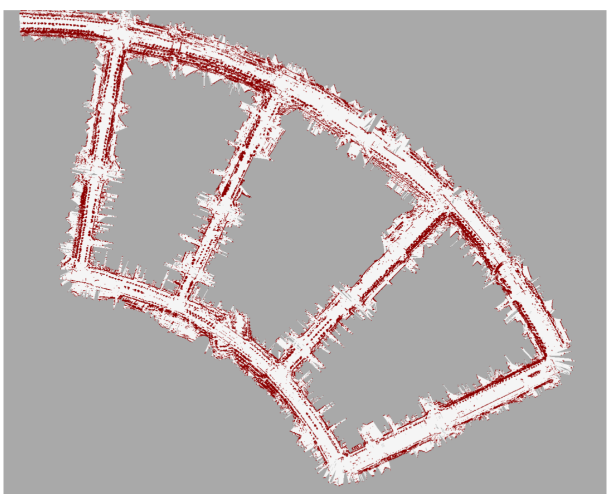

Fig. 1. Mapped and Localized Environment of Vehicle

II. Problem Formulation

The process can broadly be described in the following steps: 1. Find the motion of the vehicle and calculate its trajectory. 2. At about the same time, gather Lidar scans and consider the end points as map features 3. Correlate the results of steps 1 and 2, 4.Repeat. By combining motion and observational models, we can obtain an accurate representation of the location of our vehicle as well as its surroundings.

A. Localization Step

We will start with step 1 where we model the motion of our vehicle. We make the Markov assumption in which the current time step is only dependent on the previous time step, and is independent of all other previous steps.

$$x_{t+1} = f(x_t, u_t, w_t)\ p_f(\cdot|x_t, u_t)$$
$$x_t = \text{state of vehicle}$$
$$u_t = \text{control input}$$
$$w_t = \text{motion model}$$

We make use of the Bayes Filter, which allows us to update our understanding of the vehicles position, which in turn also gives us the prediction of our next step.

$$p_{t|t}(x_t) = p(x_t|z_{0:t}, u_{0:t-1}) \tag{1}$$

$$p_{t+1|t}(x) = \int p_f(x|s, u_t)p_{t|t}(s) \tag{2}$$

$$p_{t+1|t}(x_{t+1}) = p(x_{t+1}|z_{0:t}, u_{0:t}) \tag{3}$$

$$p_{t+1|t+1}(x) = \frac{p_h(Z_{t+1}|x)p_{t+1|t}x}{\int p_h(z_{t+1}|s)p_{t+1|t}(s)} \tag{4}$$

With equation 3, we have an updated understanding of the vehicle position, and with equation 4, we can more closely estimate our exact location.

B. Mapping Step

In order to detail a map of the vehicle's environment, Lidar scans are used to see what areas around the vehicle are occupied. This is done using a Bresenham-like line algorithm to produce unoccupied areas as well as occupied areas. Then by making the assumption that occupied cells are independent of each other given the vehicle's trajectory, we can use the following equation.

$$p(m|z_{0:t}, x_{0:t}) = \prod_{i}^{n} p(m_i|z_{0:t}, x_{0:t}) \tag{5}$$

Equation 5. m is the occupancy grid, x is the vehicle trajectory, and z are Lidar observations

By once again using Bayes rule, we can use multiple scans of points on our map to continually update whether or not areas are occupied. We can do this by calculating the log odds of occupancy per time step.

$$\lambda_{i,t} = log\frac{p(m_i = 1|z_t, x_t)}{p(m_i = -1|z_t, x_t)} - \lambda_{i,0} + \lambda_{i,t-1} \tag{6}$$

III. Technical Approach

We will now review the SLAM process with more detail.

A. Particle Filter

We use the Particle Filter method as our implementation of the Bayesian filter. The benefits of using the particle filter is that it generalizes well to multi-dimensional analysis, is widely used, and is well understood.

I. Initialize Particles:
Particles represent a Gaussian "guess" of where our vehicle will be located, as well as its orientation in the world. These "guesses" are different for each particle as Gaussian noise is added for each movement and orientation step. Particles are given x and y trajectories from our wheel encoders, and orientation from the z axis of our Fiber-Optic Gyroscope (FOG). These particles are also given a "weight" defined by the softmax of the correlation ratio between the Lidar scan and the occupancy grid.

$$p_h(z|x, m) = s(corr(z, m)) = \frac{exp(corr(z, m))}{1^T exp(corr(z, m))} \tag{7}$$

$$corr(z, m) = \sum_{y_i} \mathbf{1}\{y_i = m_i\}$$

II. Choosing the Best Particle
Now that we have particles and their correlation values in reference to Lidar data, we can compare them to each other and choose the particle with the highest weight. This 'best' particle is the particle that will most likely give us the correct location and orientation of our vehicle.

III. Rotations
One important circumstance to solve for in SLAM is the fact that our sensors only give data from one frame of reference, which is almost always changing in the world frame as the car takes turns on the road. We need to keep track of the changes in yaw (z-axis) of the vehicle and rotate any data we receive by a rotation matrix.

$${}_{world}T_{body} = \begin{bmatrix} \cos(\phi') & -\sin(\phi') & 0 & x \\ \sin(\phi') & \cos(\phi') & 0 & y \\ 0 & 0 & 1 & 0.78 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

By applying this point transformation to our data, we are able to ensure that all data would be assigned to the same world frame map coordinates. This step requires sensor readings that are **also** rotated to the same frame of reference, so we rotated the Lidar data to the FOG frame of reference, which can be seen below. Each of these rotations can be measured physically from the vehicle itself.

$${}_{Body}T_{Lidar} =$$

$$\begin{bmatrix} 0.00130201 & 0.796097 & 0.605167 & 0.8349 \\ 0.999999 & -0.000419027 & -0.00160026 & -0.0126869 \\ -0.00102038 & 0.605169 & -0.796097 & 1.76416 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}_{FOG}T_{Body} = \begin{bmatrix} 1 & 0 & 0 & -0.335 \\ 0 & 1 & 0 & -0.0350 \\ 0 & 0 & 1 & 0.78 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

III. Dead Reckoning
In developing our Particle Filter, we created a Dead Reckoning algorithm to see how a purely physical based model would perform. As we can see below, the motion model performs quite well.

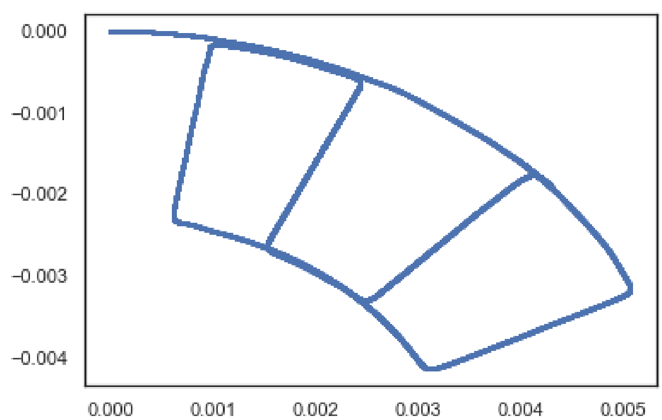

Fig. 2. Motion-Only Based Localized Map of vehicle. We see that it maps quite well. It may perform worse on sub-optimal roads

We would expect that if the roads were more varied, such as uphill or non-asphalt, that the dead reckoning model would not align as well with the roads.

III. Texture Mapping
We did not employ texture mapping in this implementation, however, if we were to, we would use stereo images in order to find the disparity measurement between the two points of view.

$$d = u_L - u_R = \frac{1}{z} f_{s_u} b \tag{8}$$

We can use the disparity between the two points to find the depth, aka distance, from observed objects. By applying the same treatment of this measurement as we did for the Lidar, we can subsequently obtain RGB values for the same coordinates as we did for Lidar scans. The final step in this process would be to take the RGB value seen by the highest weighted particle.

IV. Results

We can see how the mapping develops over time below. The algorithm we implemented is able to obtain 2d geometries of various terrain features, such as other vehicles, trees, and buildings. See an animated gif of this process at this link: https://imgflip.com/gif/4z8s8a, or in the accompanied gif ”SLAM_gif.gif”.

Upon close inspection of the mapping animation, we can see some blank areas where the first pass of the vehicle did not collect Lidar data. Once the vehicle drove the same route again, we were able to fill out the previously blank area with features. From looking through the stereo images, we saw that there was a truck blocking the Lidar’s view of the area on the first drive by, which is in accord with our expectations in regards to traffic in the real world. Our particle filter is able to improve the mapping of the environment with multiple passes, showing that as usual, more data is better.

We considered how we chose our ”best” particle, including finding an average of all particles, since the expectation of our particles should converge to the true mean. However, since we found that there was no major improvement, we decided to stick with choosing the highest weighted particle to gain higher processing speed.

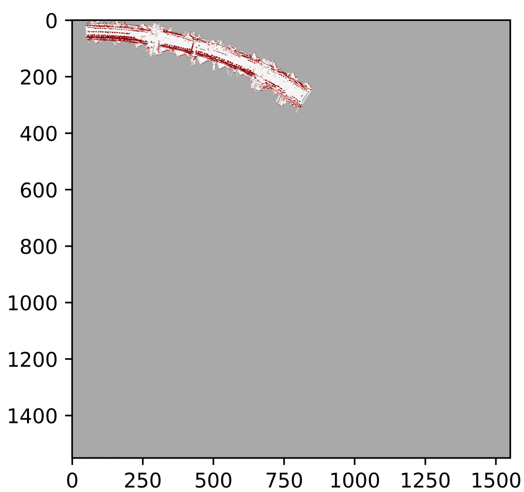

Fig. 3. Vehicle starts on its route

V. Collaboration

Collaborated with Hala Abualsaud and Luis Martin Herrera Lezama

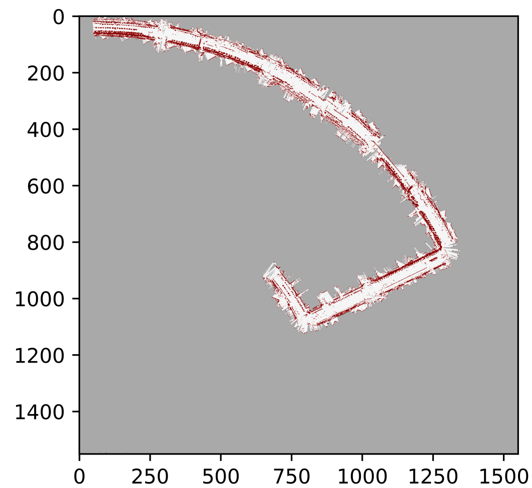

Fig. 4. Vehicle is moving forward along right leaning curve and makes two turns

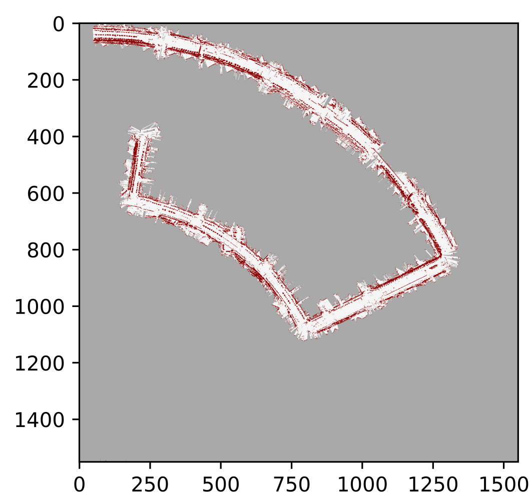

Fig. 5. Vehicle has turned right again and is about to loop back.

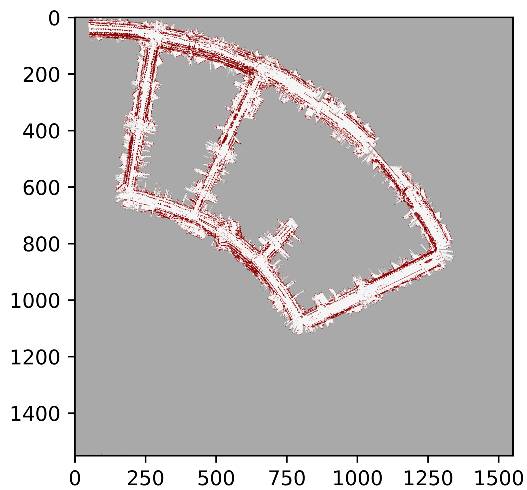

Fig. 6. Vehicle has made several more turns, repeating segments of its earlier drive

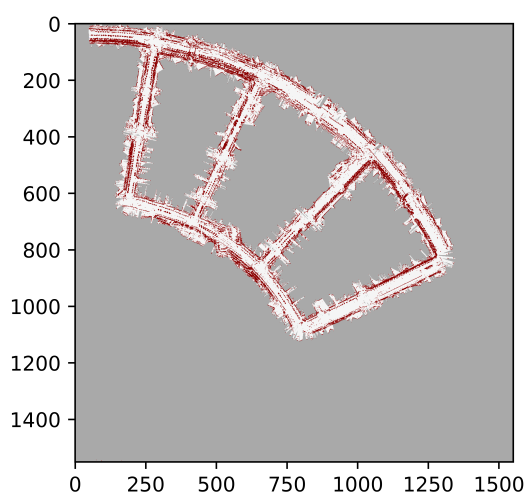

Fig. 7. Vehicle has completed its route