Particle SLAM Filter

Bolin He
Department of Electrical Computer Engineering
University of California, San Diego
b2he@ucsd.edu

I. INTRODUCTION

Simultaneous localization and mapping (SLAM) is the process by which a mobile robot can build a map of an environment and at the same time use this map to compute its own location. Particle Filter, a tool for tracking the state of a dynamic system modeled by a Bayesian network, is similar to Kalman Filter but computationally tractable for high-dimensional problems. In this project, I will utilize Particle Filter for mapping by occupancy grid with a humanoid robot. I first tried mapping from the first scan and then implemented a prediction only particle filter. Once the prediction only filter worked, I included an update step by laser scan to correct the robot pose. Finally, I can obtain the mapping and robot trajectory.

II. PROBLEM FORMULATION

The purpose in this project is to do simultaneous localization and mapping at indoor environment. I need to transform the robot data into the same frame, and then use particle filter to estimate the robot state with mapping simultaneously. Some basic knowledge are followed:

II-A. Rigid Body Motion

Assume the rigid body position is S_B in body frame and S_w in world frame. There exists a rotation matrix $R \in SO(3)$, where

$$R = R_z(\psi)R_y(\theta)R_x(\phi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}$$

such that

$$S_W = R \cdot S_B$$

by including the translation matrix p, the rigid body transformation can be written as

$$\begin{bmatrix} S_w \\ 1 \end{bmatrix} = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} \begin{bmatrix} S_B \\ 1 \end{bmatrix}$$

II-B. Particle Filter

Initially, N particles are

$$\mu_{0|0}^{(k)} = (0,0,0), \ \ \alpha_{0|0}^{(k)} = \frac{1}{N}, \ \ i = 1,2,...,N$$

with $\mu^{(k)}$ as robot state and $\alpha^{(k)}$ as weight (namely confidence).

- Prediction Step

Besides the robot state $\mu_{t|t}^{(k)}$, Gaussian motion noise $\epsilon \sim \mathcal{N}(0, \epsilon)$ will be added to $\mu^{(k)}$, which will results in

$$\mu_{t+1|t}^{(k)} = f(\mu_{t|t}^{(k)}, u_t + \epsilon_t)$$

- Update Step

Utilize the observation z_{t+1}, I can update the particle weight using the laser correlation model

$$p_h(z_{t+1} \,|\, \mu_{t+1}^{(k)}, m) \propto exp(corr(y_{t+1}^{(k)}, m))$$
$$\alpha_{t+1|t+1}^{(k)} = \frac{\alpha_{t+1|t}^{(k)} \, p_h(z_t \,|\, x, m)}{\Sigma_j \alpha_{t+1|t}^{(j)} \, p_h(z_t \,|\, x^j, m)}$$

- Resample Step

Set the criteria

$$N_{eff} := \frac{1}{\Sigma_{k=1}^{N} (\alpha_{t|t}^{(k)})^2} \leq N_{threshold}$$

to resample the particles and normalize their weights as $\{\bar{\mu}_{t|t}^{(k)}, \bar{\alpha}_{t|t}^{(k)}\}$ to eliminate the collective error.

III-C. Occupancy Grid Map

Occupancy grid map is a vector $m \in R^n$, where m_i cell is free ($m_i = 0$) or occupied ($m_i = 1$). By iteration, for each individual observed cell i, I can assign the log-odds $\lambda_{i,t}$ such that

$$\lambda_{i,t+1} = \lambda_{i,t} + log\, g_h(z_{t+1} \,|\, m_i, x_{t+1})$$

And the map pmf $\gamma_{i,t}$ can be recovered from $\lambda_{i,t}$ as

$$\gamma_{i,t} = p(m_i = 1 \,|\, z_{0:t}, x_{0:t}) = 1 - \frac{1}{1 + exp(\lambda_{i,t})}$$

Finally I can create a grid map with fully 0 or 1.

III. TECHNICAL APPROACH

III-A. Rigid Body Motion

In this project, I need to transform the laser data from laser frame into world frame, which includes

$${}_{World}T_{Laser} = {}_{World}T_{Body} \cdot {}_{Body}T_{Head} \cdot {}_{Head}T_{Laser}$$

where ${}_{Head}T_{Laser}$ is translation only, and ${}_{Body}T_{Head}$ is associated with the head joint and neck joint at specific timestamp. After Synchronize the timestamp, at each timestamp, I can obtain

$${}_{Body}T_{Laser} = {}_{Body}T_{Head} \cdot {}_{Head}T_{Laser}$$

${}_{World}T_{Body}$ depends on the particle in the world frame. When I do particle filter update step, I need to calculate ${}_{World}T_{Body}$ for each particles, and thus find the transformation ${}_{World}T_{Laser}$.

III-B. Particle Filter

- Prediction Step

I try to initialize particles with different number, which is 10, 50 and 100, with equally distributed weight.

At the very first beginning, I try dead-reckoning, a prediction with no noise and only one single particle, to see whether the robot trajectory make sense or not.

Once it works, I add some gaussian noise with $\mu = 0$, $\Sigma = 0.05$ to each particle individually. At this step, the absolute laser pose motion will be also added to $\mu_{t|t}^{(k)}$.

- Update Step

At this step, for each particle, I need to calculate its ${}_{World}T_{Body}$, and then calculate the map correlation $corr$ for this particle, combined with softmax function, finally obtain p_h. I can update the particles with normalized weight once collect all the p_h information as

$$\alpha_{t+1|t+1}^{(k)} = \frac{\alpha_{t+1|t}^{(k)}\, p_h(z_t\,|\,x, m)}{\Sigma_j \alpha_{t+1|t}^{(j)}\, p_h(z_t\,|\,x^j, m)}$$

- Resample Step

I set $N_{threshold} = 5$, once $N_{eff} > N_{threshold}$, I start resample step to obtain new particles set and equally normalized weight.

III-C. Occupancy Grid Map

At the former procedures, after each iteration, I can obtain a new particles set with different weights. I choose the particle with most weight as the best particle. I assume the robot locates exactly at this particle at that single timestamp. I calculate the ${}_{World}T_{Body}$ of this single particle, and thus find ${}_{World}T_{Laser}$. At this timestamp, laser scan ranges from -135° to 135° with 1081 data. I set the criteria as $0.1 < range < 30$ to eliminate the invalid data.

Then I use the tool bresenham2D to draw the contour between start points and end points. The start point is the best particle's location, and the end points are calculated from the valid data in laser scan. In real world, the end points are where the wall or namely boundary is. I remove the data which $z\ axis$ is smaller than 0.01, which mean these end points are hitting the ground. I set the end points with $logodd(4)$, and other occupied points with $logodd(1/4)$. The free points remain zero. Which results in

$$\lambda_{i,t+1} = \lambda_{i,t} + log\, g_h(z_{t+1}\,|\,m_i, x_{t+1})$$

After all the iteration, I can obtain a log-odd map. I convert it back to probability

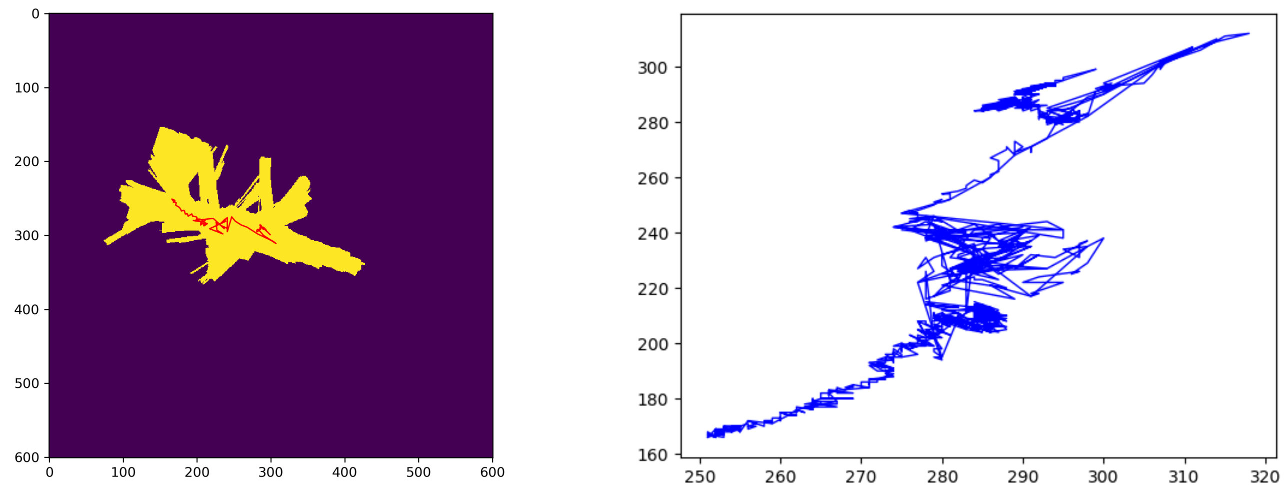
$$\gamma_{i,t} = p(m_i = 1 \mid z_{0:t}, x_{0:t}) = 1 - \frac{1}{1 + exp(\lambda_{i,t})}$$

and set a clip to avoid over confidence.

Finally I can show my map in IDE.

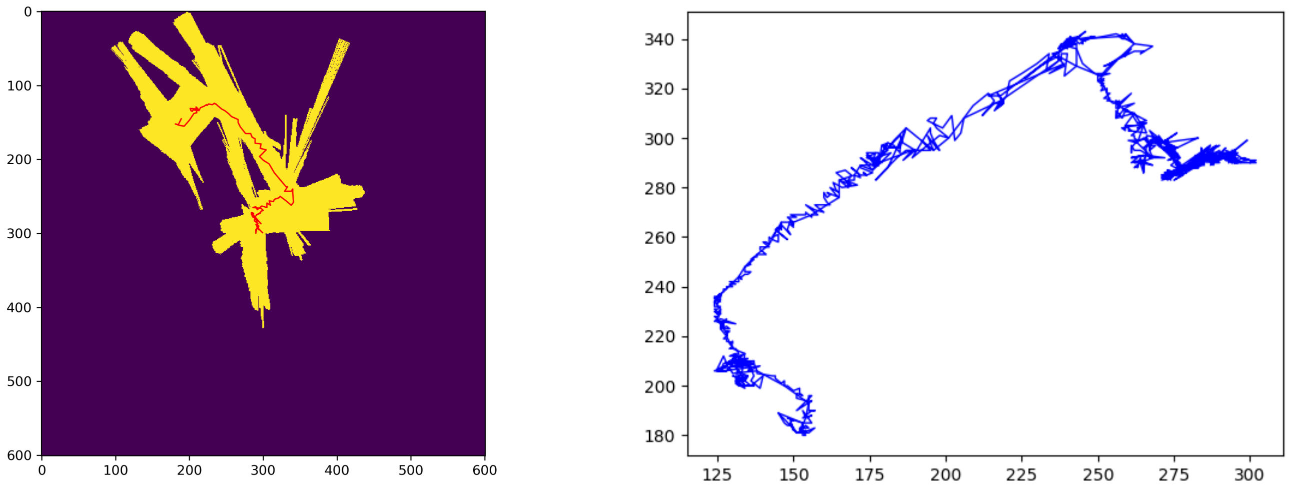
IV. RESULT

IV-A. Trainset 0



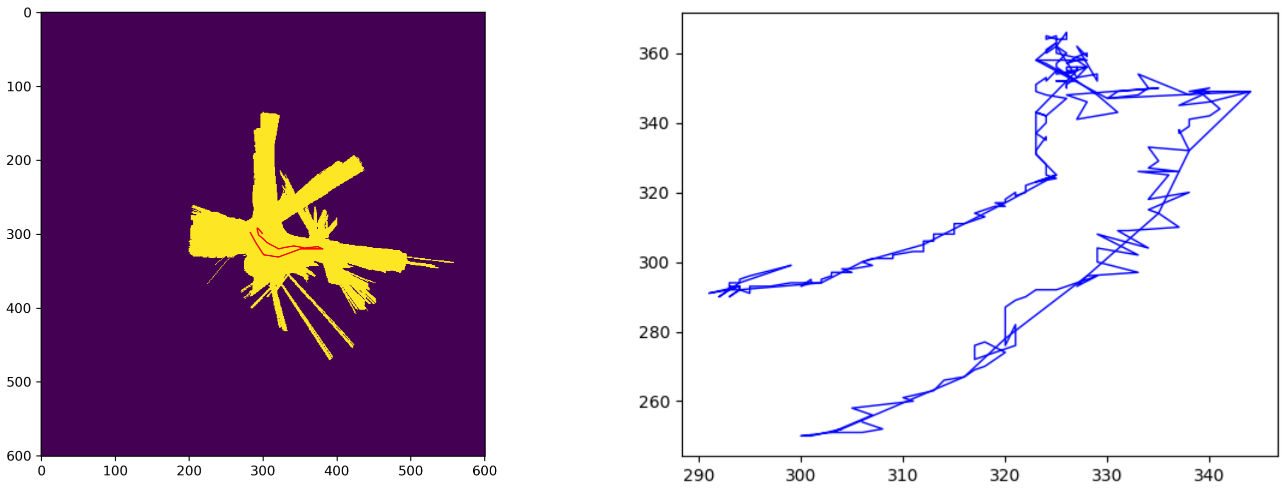
figure(a). Map and trajectory of transit 0

IV-B. Trainset 1



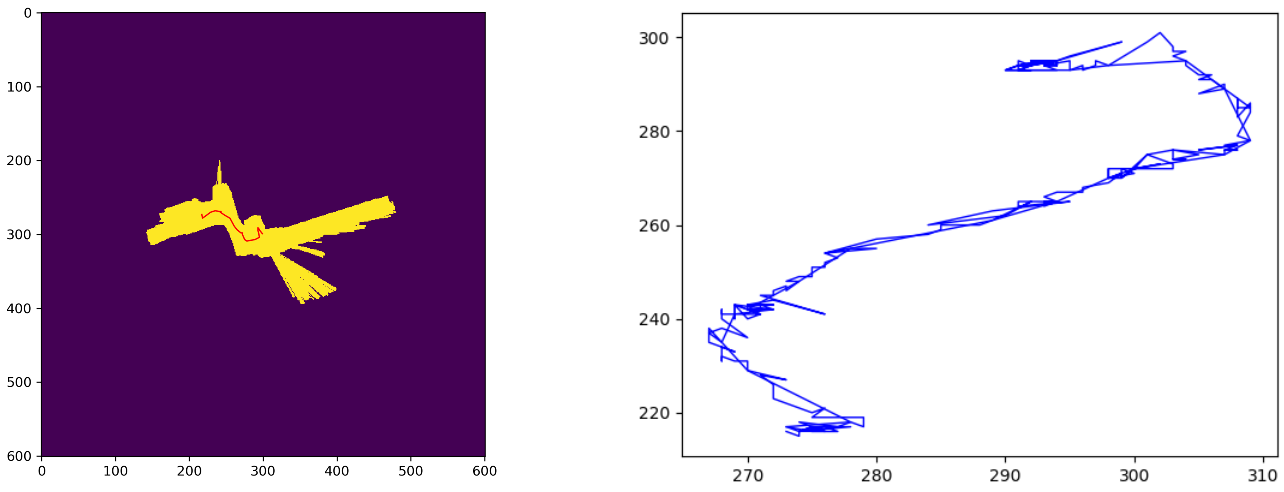
figure(b). Map and trajectory of transit 1

IV-C. Trainset



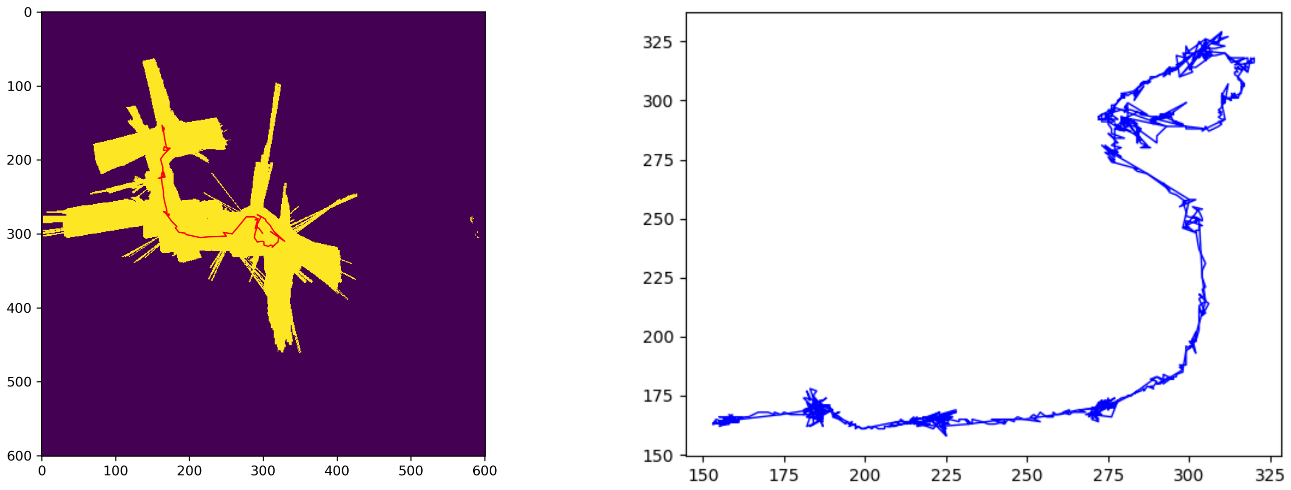
figure(c). Map and trajectory of transit 2

IV-D. Trainset 3



figure(d). Map and trajectory of transit 3

IV-E. Trainset 4

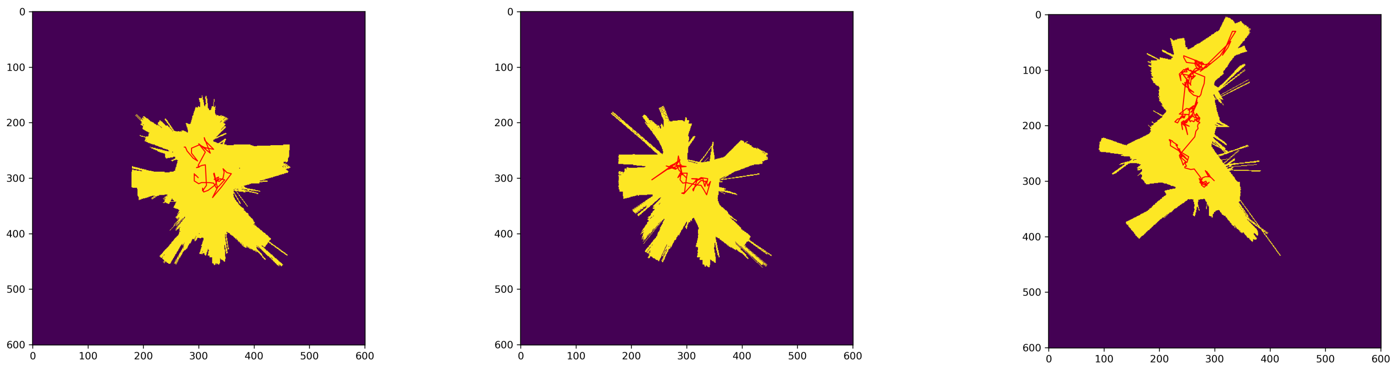


figure(e). Map and trajectory of transit 4

IV-F. Discussion

Different numbers of particles and time steps will have different results. Intuitively, the more particles and the smaller time step we iterate, we will have better results.

Take trainset 4 as example, some failure cases are below:



figure(f). Some failure cases of trainset 4

At these cases, some of them have too few particles, some of them have too large time step. The best result of transit 4 is 100 particles and 30 time step. Because if I have only a few particles, they are not representative for all the possible locations in the space. And a large time step will probably miss some information and thus not accurate at all. However, the balance of the accuracy and the configuration varies from algorithms and hardwares. It requires us to have more tests to keep a balance.

Moreover, trainset 3 can usually have good results even with a less particles or larger time step. From my point of view, trainset 3 have relative smaller space and a straight-form hallway, which means it should have better data. Oppostiely, those trajectories with more corners and hallways, or move relatively faster, would be more difficult to obtain idealized result.