FastSLAM

SLAM with Particle Filters

As discussed in the previous course part, the particle filter is a general method for Bayesian state estimation that does not make assumptions about the form of the belief distribution. Ideally, we want to keep that generality in SLAM. In principle, the SLAM problem could be approached with a particle filter similar to localization. We can consider the robot pose and the map as a joint state variable $\langle x, m\rangle$ and estimate the belief $bel(x, m)$ using the standard PF algorithm. That would require covering the state space with a sufficient number of particles, which in practice would quickly become intractable due to the curse of dimensionality. Consider a landmark-based map representation with M landmarks, i.e., $m = \langle l_1, \ldots, l_M\rangle$. The state space would then have at least M dimensions, where in practical applications hundreds or thousands of landmarks are used.

FastSLAM Idea

The core idea of FastSLAM is to deploy the particle filter only for pose estimation and use a more efficient estimator for the map, e.g., the Kalman Filter. Recall that the full SLAM posterior is $p(x_{0:t}, m \mid z_{1:t}, u_{1:t})$. Using the rule of conditional probability it can be factored as

$$p(x_{0:t}, m \mid z_{1:t}, u_{1:t}) = p(x_{0:t} \mid z_{1:t}, u_{1:t}) \cdot p(m \mid x_{0:t}, z_{1:t}).$$

Though FastSLAM can be adapted to different map representations, here we focus on the landmark-based map $m = \langle l_1, \ldots, l_M\rangle$. Using the fact that the landmarks are independent given the poses we can rewrite the map posterior as a product of individual landmark posteriors:

$$p(x_{0:t}, l_{1:M} \mid z_{1:t}, u_{1:t}) = p(x_{0:t} \mid z_{1:t}, u_{1:t}) \prod_i^M p(l_i \mid x_{0:t}, z_{1:t}).$$

Assuming that $p(l_i \mid x_{0:t}, z_{1:t})$ is normally distributed, the product on the right can be computed with M Kalman Filters while the pose hypotheses are represented by particles. In FastSLAM, a particle thus holds a pose hypothesis and its own map estimate consisting of M Kalman Filters. In the case of a 2D world, we aim to estimate the 2D coordinates of each landmark, and therefore each Kalman Filter stores a (2×1) mean vector and a (2×2) covariance matrix.

FastSLAM Algorithm

On the top level, the FastSLAM algorithm is similar to the particle filter:

1. Prediction: Sample a new set of particles according to the current control input and the motion model.

2. Correction: Update the landmark estimates using the current measurement and calculate new importance weights.
3. Resampling: Sample a new set of particles according to the importance weights.

The prediction and resampling steps are essentially the same as in the standard particle filter. The landmark estimates are updated according to the KF correction step. Let $\mathcal{X}$ be the set of N particles, $x^{[k]}$ the pose hypothesis of particle k, and $w^{[k]}$ its importance weight. Further, M is the total number of landmarks, $z_{t,i} \in z_t$ the measurement of the ith landmark (if it was observed in that step), $\mu_i^{[k]}$ is the mean of that landmark associated with particle k, and $\Sigma_i^{[k]}$ the corresponding covariance. With that, the FastSLAM algorithm is:

```
1. Algorithm $\text{FastSLAM}(\mathcal{X}_{t-1}, z_t, u_t)$ :
2.     for $k = 1$ to $N$ do
3.         $x_t^{[k]} \sim p(x_t \mid x_{t-1}^{[k]}, u_t)$
4.         $w^{[k]} = 1$
5.         for $i = 1$ to $M$ do
6.             if $z_{t,i} \in z_t$
7.                 $\hat{z} = C_t \mu_{i,t-1}^{[k]}$
8.                 $Q = C_t \Sigma_{i,t-1}^{[k]} C_t^T + R_t$
9.                 $K = \Sigma_{i,t-1}^{[k]} C_t^T Q^{-1}$
10.                $\mu_{i,t}^{[k]} = \mu_{i,t-1}^{[k]} + K(z_{t,i} - \hat{z})$
11.                $\Sigma_{i,t}^{[k]} = (I - KC_t)\Sigma_{i,t-1}^{[k]}$
12.                $w^{[k]} = w^{[k]} \cdot normal(z_{t,i}; \hat{z}, Q)$
13.            else
14.                $\mu_{i,t}, \Sigma_{i,t} = \mu_{i,t-1}, \Sigma_{i,t-1}$
15.            endif
16.        endfor
17.    endfor
18.    normalize new weights
19.    $\mathcal{X}_t$ = resample_particles
20.    return $\mathcal{X}_t$
```

In line 8, Q is the so-called measurement covariance. Further note that above the matrix C_t maps a landmark position to a landmark measurement.

Final Remarks

In the FastSLAM version presented here, we assume that the correspondence between landmarks and landmark measurements is known. We further assume that the measurement model for landmarks follows a Gaussian distribution. However, this assumption can be dropped by using Extended Kalman Filters (EKFs) as proposed in the original **paper (https://www.aaai.org/Papers/AAAI/2002/AAAI02-089.pdf)**.