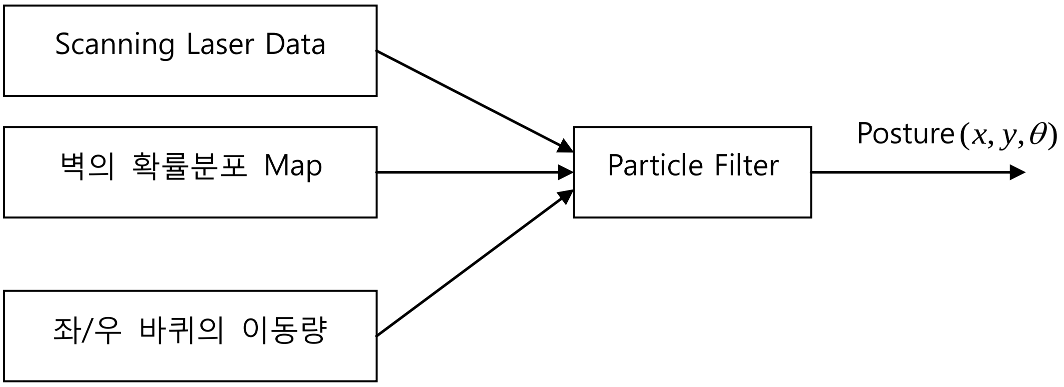
Particle Filter Localization

(Scanning Laser Range Sensor에서 측정한 데이터와 차동 바퀴형 이동체의 좌우 바퀴의 Odometry 정보를 이용한 Particle Filter Localization)

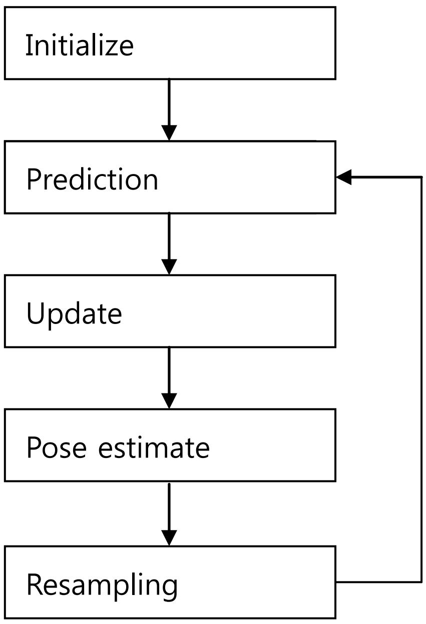
KITECH 양광웅 작성

개요

Scanning Laser로 측정한 벽의 위치와 양쪽 바퀴의 이동 량으로 Particle Filter를 수행하여 로봇의 위치와 방향을 추정하게 된다.



Particle filter의 수행은 Initialize, Prediction, Update, Normalize, Resampling, Pose Estimation 의 단계로 구성되어 있고, 수행 과정은 아래 그림과 같다.



Particle Filter 알고리즘은 Prediction과 Update 상태를 반복적으로 수행하게 된다. Prediction 상태는 로봇의 이동량에 따라 적절한 에러를 추가하여 파티클을 이동한다. Update 상태는 센서가 측정한 정보에 따라 파티클의 가중치(weight)를 업데이트 한다.

Particle Filter

Particle Filter에서 확률분포는 시간 t 에서 가중치를 가지는 N 개의 파티클에 대한 집합 S_t 로 표시된다. $\mathbf{x}_t^{(i)} = [x_t^{(i)}, y_t^{(i)}, \theta_t^{(i)}]^T$ 는 파티클의 위치를 나타내고 $w_t^{(i)}$ 는 파티클의 가중치를 의미한다.

$$S_t = \left\{ \left\langle \mathbf{x}_t^{(i)}, w_t^{(i)} \right\rangle \mid i = 1, \cdots, N \right\}$$

Initialize

파티클 필터는 전역 지도상에서 로봇의 초기 위치와 방향을 계산할 수 없다. 그래서 사용자가 알고리즘이 수행되기 전에 로봇의 초기 위치와 방향을 파티클 필터에 설정해 주어야 한다. 사용자가 초기 위치와 방향을 설정하게 되면, 설정한 위치에서 N 개의 파티클을 생성한다. 이때, 모든 파티클의 가중치는 같으며 합이 1이 된다.

$$S_{t=0} = \left\{ \left\langle \mathbf{x}_{t=0}^{(i)}, w_{t=0}^{(i)} \right\rangle \mid i = 1, \cdots, N \right\}$$

여기서 $\sum_{i=1}^{N} w_{t=0}^{(i)} = 1$ 이다.

Prediction

로봇이 u_t 만큼 이동하였을 때, 각 파티클들을 로봇의 이동량만큼 이동한다. 이때 좌우 바퀴의 회전한 양에 비례하는 노이즈를 더하여 로봇의 이동위치와 방향을 계산한다.

$$\begin{aligned}
\mathbf{x}_t^{(i)} &= \mathbf{x}_{t-1}^{(i)} + u_t \\
&= \begin{bmatrix} x_{t-1}^{(i)} \\ y_{t-1}^{(i)} \\ \theta_{t-1}^{(i)} \end{bmatrix} + \begin{bmatrix} \dfrac{\Delta s_{r,t}^{(i)} + \Delta s_{l,t}^{(i)}}{2} \cos(\theta_{t-1}^{(i)} + \dfrac{\Delta s_{r,t}^{(i)} - \Delta s_{l,t}^{(i)}}{2b}) \\ \dfrac{\Delta s_{r,t}^{(i)} + \Delta s_{l,t}^{(i)}}{2} \sin(\theta_{t-1}^{(i)} + \dfrac{\Delta s_{r,t}^{(i)} - \Delta s_{l,t}^{(i)}}{2b}) \\ \dfrac{\Delta s_{r,t}^{(i)} - \Delta s_{l,t}^{(i)}}{b} \end{bmatrix}
\end{aligned}$$

여기서 b 는 양쪽 바퀴간의 거리다.

노이즈를 고려한 좌우 바퀴의 이동량은 다음과 같다.

$$\Delta s_{r,t}^{(i)} \leftarrow \text{gauss\_rand}(\Delta s_{r,t}, k_r |\Delta s_{r,t}|)$$
$$\Delta s_{l,t}^{(i)} \leftarrow \text{gauss\_rand}(\Delta s_{l,t}, k_l |\Delta s_{l,t}|)$$

여기서 $\Delta s_{r,t}, \Delta s_{l,t}$ 은 마지막 셈플링 기간동안 오른쪽 바퀴와 왼쪽 바퀴가 각각 회전한 양이고, k_r, k_l 은 좌우 바퀴의 에러 상수다. $\text{gauss\_rand}(\mu, \Sigma)$ 는 평균 μ 에서 분산 Σ 를 가지는 가우시안 랜덤 값이다.

Update
각 파티클에 대하여 파티클의 위치 $(x_t^{(i)}, y_t^{(i)}, \theta_t^{(i)})$ 에서 Scanning Laser 센서로 측정한 벽과의 거리 값으로 벽이 존재하는 확률을 계산한다.
센서가 측정한 거리 정보가 $D_t = \{d_j \mid j = 1, \cdots, M\}$ 이고 센서의 측정 범위가 $\theta_S \sim \theta_E$ 일 때, 다음과 같이 전역공간에서 벽의 위치(x_w, y_w)를 계산할 수 있다. (센서는 로봇의 중심에 설치되어 있으며, 로봇의 방향과 같은 방향으로 설치되어 있다고 가정한다.)

$$x_w = x_t^{(i)} + d_j \cos(\theta_t^{(i)} + \theta_j),$$
$$y_w = y_t^{(i)} + d_j \sin(\theta_t^{(i)} + \theta_j).$$

여기서 $\theta_j = \theta_S + (\theta_E - \theta_S)\dfrac{j-1}{M-1}$ 이다.

전역 공간에서 벽의 위치를 계산하였다면 다음의 식으로부터 확률 $p_t^{(i)}$ 를 계산한다.

$$p_t^{(i)} = \sum_{j=1}^{M} \text{map\_prob}(x_w, y_w)$$

여기서 $\text{map\_prob}(\cdot)$ 함수는 x_w, y_w 위치에서 벽이 존재할 확률을 계산하는 함수다.
(* 전역 Map 처리.docx 파일 참조)

Normalize
각 파티클의 확률을 모두 계산하였다면, 모든 파티클들의 가중치 합이 1이 되도록 각 파티클의 가중치를 업데이트 한다. 여기서 주의할 점은 현재의 가중치 $w_t^{(i)}$ 는 과거의 가중치 $w_{t-1}^{(i)}$ 에 영향을 받지 않고 매번 새로운 값으로 계산된다는 것이다.

$$w_t^{(i)} = \frac{p_t^{(i)}}{\sum_{k=1}^{N} p_t^{(k)}}$$

이제 기존 파티클의 집합 S에 대한 Prediction과 Update 과정이 끝나 새로운 상태가 되었다.

$$S_t = \left\{ \left\langle \mathbf{x}_t^{(i)}, w_t^{(i)} \right\rangle : i = 1, \cdots, N \right\}$$

Pose Estimate

모든 파티클들의 가중치와 위치의 곱을 합하여 최종 로봇의 추정 위치 $\mathbf{p}_{est}$를 계산한다.

$$\mathbf{p}_{est} = \sum_{i=1}^{N} w_t^{(i)} \mathbf{x}_t^{(i)}$$

이때 θ의 연산에 주의하여야 한다. 예를 들자면, 각도 -180과 180을 더하여 평균을 내면 0이지만, 실제로 평균은 180이 되어야 한다. θ는 -180도 에서 +180도의 값을 가지므로 +영역에 있는 각도와 −영역에 있는 각도를 따로 계산하는데, −영역의 값을 기준으로 +영역의 값과 −영역의 값의 차를 더하여 평균을 계산한다.

Resampling

각 파티클들의 가중치에 비례하는 새로운 파티클들을 생성한다. 즉, 가중치가 작은 파티클은 소멸하게 되고 가중치가 클수록 많은 새로운 파티클들을 생성한다. 새로운 파티클은 기존 파티클의 특성 $(\mathbf{x}_t^{(i)}, w_t^{(i)})$을 그대로 물려받는다.

파티클 i가 만들어내는 새로운 파티클의 수 n은 다음과 같이 계산한다.

$$n = \left\lceil N w_t^{(i)} \right\rceil$$