

Занятие №6. EKF SLAM и GMapping

EKF SLAM

Суть EKF SLAM можно найти в [статье](#)

gMapping

Общий алгоритм представлен на рисунке:

Algorithm 1 Improved RBPf for Map Learning Require: $\mathcal{S}_{t-1}$, the sample set of the previous time step z_t, the most recent laser scan u_{t-1}, the most recent odometry measurement Ensure: $\mathcal{S}_t$, the new sample set $\mathcal{S}_t = \{\}$ for all $s_{t-1}^{(i)} \in \mathcal{S}_{t-1}$ do $\langle x_{t-1}^{(i)}, w_{t-1}^{(i)}, m_{t-1}^{(i)} \rangle = s_{t-1}^{(i)}$ <i>// scan-matching</i> $x_t'^{(i)} = x_{t-1}^{(i)} \oplus u_{t-1}$ $\hat{x}_t^{(i)} = \operatorname{argmax}_x p(x \mid m_{t-1}^{(i)}, z_t, x_t'^{(i)})$ if $\hat{x}_t^{(i)} = \text{failure}$ then $x_t^{(i)} \sim p(x_t \mid x_{t-1}^{(i)}, u_{t-1})$ $w_t^{(i)} = w_{t-1}^{(i)} \cdot p(z_t \mid m_{t-1}^{(i)}, x_t^{(i)})$ else <i>// sample around the mode</i> for $k = 1, \dots, K$ do $x_k \sim \{x_j \mid x_j - \hat{x}_t^{(i)} < \Delta\}$ end for	<i>// compute Gaussian proposal</i> $\mu_t^{(i)} = (0, 0, 0)^T$ $\eta^{(i)} = 0$ for all $x_j \in \{x_1, \dots, x_K\}$ do $\mu_t^{(i)} = \mu_t^{(i)} + x_j \cdot p(z_t \mid m_{t-1}^{(i)}, x_j) \cdot p(x_t \mid x_{t-1}^{(i)}, u_{t-1})$ $\eta^{(i)} = \eta^{(i)} + p(z_t \mid m_{t-1}^{(i)}, x_j) \cdot p(x_t \mid x_{t-1}^{(i)}, u_{t-1})$ end for $\mu_t^{(i)} = \mu_t^{(i)} / \eta^{(i)}$ $\Sigma_t^{(i)} = 0$ for all $x_j \in \{x_1, \dots, x_K\}$ do $\Sigma_t^{(i)} = \Sigma_t^{(i)} + (x_j - \mu_t^{(i)})(x_j - \mu_t^{(i)})^T \cdot p(z_t \mid m_{t-1}^{(i)}, x_j) \cdot p(x_j \mid x_{t-1}^{(i)}, u_{t-1})$ end for $\Sigma_t^{(i)} = \Sigma_t^{(i)} / \eta^{(i)}$ <i>// sample new pose</i> $x_t^{(i)} \sim \mathcal{N}(\mu_t^{(i)}, \Sigma_t^{(i)})$ <i>// update importance weights</i> $w_t^{(i)} = w_{t-1}^{(i)} \cdot \eta^{(i)}$ end if <i>// update map</i> $m_t^{(i)} = \text{integrateScan}(m_{t-1}^{(i)}, x_t^{(i)}, z_t)$ <i>// update sample set</i> $\mathcal{S}_t = \mathcal{S}_t \cup \{\langle x_t^{(i)}, w_t^{(i)}, m_t^{(i)} \rangle\}$ end for $N_{\text{eff}} = \frac{1}{\sum_{i=1}^N (\tilde{w}^{(i)})^2}$ if $N_{\text{eff}} < T$ then $\mathcal{S}_t = \text{resample}(\mathcal{S}_t)$ end if
---	---

Во-первых, зафиксируем некоторым «набором выборки», в котором будут лежать все состояния системы за всё время. Каждое состояние - это вектор из трёх элементов: x - положение робота w - вес вектора m - карта на текущий момент времени

Итак, в полученном наборе $\mathcal{S}_t$ будет столько же элементов, сколько в $\mathcal{S}_{(t-1)}$, поскольку алгоритм предполагает прохождение по всем элементам набора $\mathcal{S}_{(t-1)}$, и на каждом шаге в этот набор добавляется по одному элементу. Набор $\mathcal{S}_t$ можно считать различными гипотезами о текущем положении робота. Совершенно необязательно иметь только одну гипотезу. Наоборот, выгоднее просчитывать несколько вариантов, потому что окружающий мир может быть фрактальным и в этом случае наличие нескольких гипотез может помочь определить реальное положение.

Теперь подробнее о том, как строится каждый элемент. Сначала к значению, полученному на предыдущем шаге прибавляется значение одометрии и получается x' - априорная оценка.

Затем ищется $\arg\max$ плотности распределения величины x в зависимости от текущего знания о карте m , текущего наблюдения z и априорной оценки x' .

Если апостериорная оценка сильно отличается от априорной, то апостериорная оценка отбрасывается, в качестве предположения о новом положении робота выбирается случайная величина, распределённая как $p(x | x_{-}(t-1), u_{-}(t-1))$, то есть исключительно на основе предыдущего положения и данных одометрии. Вес наблюдения считается также тривиально.

В случае если априорная и апостериорная оценки оказались близки, в качестве предположения о положении робота выбирается случайная величина, распределённая по нормальному закону с математическим ожиданием и дисперсией, посчитанными на основании выборки положений, построенных в окрестности апостериорной оценки.

Вне зависимости от того, как считается x , на основании этого значения обновляется карта и новый посчитанный вектор добавляется в набор выборки.

По завершении работы цикла высчитывается количество элементов, которые должно содержать St , чтобы дисперсия весов была равна единице (чтобы они были распределены по стандартному нормальному закону). Если это количество оказалось меньше, чем реальное значение T , то всю операцию необходимо проделать заново, чтобы увеличить веса построенных предположений x .

From:

<https://se.moevm.pro/> - **МОЭВМ Вики** [se.moevm.pro]

Permanent link:

<https://se.moevm.pro/doku.php/courses:ros:class6>

Last update:

