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**
$< x_{t-1}^{(i)}, w_{t-1}^{(i)}, m_{t-1}^{(i)} >= s_{t-1}^{(i)}$

// scan-matching
$x_t'^{(i)} = x_{t-1}^{(i)} \oplus u_{t-1}$
$\hat{x}_t^{(i)} = \mathrm{argmax}_x \, p(x \mid m_{t-1}^{(i)}, z_t, x_t'^{(i)})$

if $\hat{x}_t^{(i)} =$ **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
// sample around the mode
for $k = 1, \ldots, K$ **do**
$x_k \sim \{x_j \mid |x_j - \hat{x}^{(i)}| < \Delta\}$
end for
// compute Gaussian proposal
$\mu_t^{(i)} = (0,0,0)^T$
$\eta^{(i)} = 0$
for all $x_j \in \{x_1, \ldots, x_K\}$ **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
$\mu_t^{(i)} = \mu_t^{(i)} / \eta^{(i)}$
$\Sigma_t^{(i)} = \mathbf{0}$
for all $x_j \in \{x_1, \ldots, x_K\}$ **do**
$\Sigma_t^{(i)} = \Sigma_t^{(i)} + (x_j - \mu^{(i)})(x_j - \mu^{(i)})^T \cdot$
$p(z_t \mid m_{t-1}^{(i)}, x_j) \cdot p(x_j \mid x_{t-1}^{(i)}, u_{t-1})$
end for
$\Sigma_t^{(i)} = \Sigma_t^{(i)} / \eta^{(i)}$
// sample new pose
$x_t^{(i)} \sim \mathcal{N}(\mu_t^{(i)}, \Sigma_t^{(i)})$

// update importance weights
$w_t^{(i)} = w_{t-1}^{(i)} \cdot \eta^{(i)}$
end if
// update map
$m_t^{(i)} = \mathrm{integrateScan}(m_{t-1}^{(i)}, x_t^{(i)}, z_t)$
// update sample set
$\mathcal{S}_t = \mathcal{S}_t \cup \{< x_t^{(i)}, w_t^{(i)}, m_t^{(i)} >\}$
end for

$N_{\mathrm{eff}} = \frac{1}{\sum_{i=1}^{N} (\tilde{w}^{(i)})^2}$

if $N_{\mathrm{eff}} < T$ **then**
$\mathcal{S}_t = \mathrm{resample}(\mathcal{S}_t)$
end if

Во-первых, зафиксируемся 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 St будет столько же элементов, сколько в S_(t-1), поскольку алгоритм предполагает прохождение по всем элементам набора S_(t-1), и на каждом шаге в этот набор добавляется по одному элементу. Набор St можно считать различными гипотезами о текущем положении робота. Совершенно необязательно иметь только одну гипотезу. Наоборот, выгоднее просчитывать несколько вариантов, потмоу 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 argmax 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 качесте 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.info/ - **МОЭВМ Вики [se.moevm.info]**

Permanent link:
https://se.moevm.info/doku.php/courses:ros:class6

Last update: