Trabalho #6

Simular os algoritmos apresentados no capítulo 8 do livro [THRUN ETAL:2006] .

Tabelas com os algoritmos:

Algorithm Grid_localization($\{p_{k,t-1}\}, u_t, z_t, m$):
for all k *do*
$\bar{p}_{k,t} = \sum_i p_{i,t-1}$ **motion_model**$(\text{mean}(\mathbf{x}_k), u_t, \text{mean}(\mathbf{x}_i))$
$p_{k,t} = \eta\, \bar{p}_{k,t}$ **measurement_model**$(z_t, \text{mean}(\mathbf{x}_k), m)$
endfor
return $\{p_{k,t}\}$

Algorithm MCL($\mathcal{X}_{t-1}, u_t, z_t, m$):
$\bar{\mathcal{X}}_t = \mathcal{X}_t = \emptyset$
for $m = 1$ *to* M *do*
$x_t^{[m]} =$ **sample_motion_model**$(u_t, x_{t-1}^{[m]})$
$w_t^{[m]} =$ **measurement_model**$(z_t, x_t^{[m]}, m)$
$\bar{\mathcal{X}}_t = \bar{\mathcal{X}}_t + \langle x_t^{[m]}, w_t^{[m]} \rangle$
endfor
for $m = 1$ *to* M *do*
draw i *with probability* $\propto w_t^{[i]}$
add $x_t^{[i]}$ *to* $\mathcal{X}_t$
endfor
return $\mathcal{X}_t$

Algorithm Augmented_MCL($\mathcal{X}_{t-1}, u_t, z_t, m$):
static $w_{\text{slow}}, w_{\text{fast}}$
$\bar{\mathcal{X}}_t = \mathcal{X}_t = \emptyset$
for $m = 1$ *to* M *do*
$x_t^{[m]} =$ **sample_motion_model**$(u_t, x_{t-1}^{[m]})$
$w_t^{[m]} =$ **measurement_model**$(z_t, x_t^{[m]}, m)$
$\bar{\mathcal{X}}_t = \bar{\mathcal{X}}_t + \langle x_t^{[m]}, w_t^{[m]} \rangle$
$w_{\text{avg}} = w_{\text{avg}} + \frac{1}{M}\, w_t^{[m]}$
endfor
$w_{\text{slow}} = w_{\text{slow}} + \alpha_{\text{slow}}(w_{\text{avg}} - w_{\text{slow}})$
$w_{\text{fast}} = w_{\text{fast}} + \alpha_{\text{fast}}(w_{\text{avg}} - w_{\text{fast}})$
for $m = 1$ *to* M *do*
with probability $\max\{0.0,\ 1.0 - w_{\text{fast}}/w_{\text{slow}}\}$ *do*
add random pose to $\mathcal{X}_t$
else
draw $i \in \{1, \ldots, N\}$ *with probability* $\propto w_t^{[i]}$
add $x_t^{[i]}$ *to* $\mathcal{X}_t$
endwith
endfor
return $\mathcal{X}_t$

Algorithm KLD_Sampling_MCL($\mathcal{X}_{t-1}, u_t, z_t, m, \varepsilon, \delta$):
$\mathcal{X}_t = \emptyset$
$M = 0,\ M_\chi = 0,\ k = 0$
for all b *in* H *do*
$b =$ *empty*
endfor
do
draw i *with probability* $\propto w_{t-1}^{[i]}$
$x_t^{[M]} =$ **sample_motion_model**$(u_t, x_{t-1}^{[i]})$
$w_t^{[M]} =$ **measurement_model**$(z_t, x_t^{[M]}, m)$
$\mathcal{X}_t = \mathcal{X}_t + \langle x_t^{[M]}, w_t^{[M]} \rangle$
if $x_t^{[M]}$ *falls into empty bin* b *then*
$k = k + 1$
$b =$ *non-empty*
if $k > 1$ *then*
$M_\chi := \frac{k-1}{2\varepsilon}\left\{1 - \frac{2}{9(k-1)} + \sqrt{\frac{2}{9(k-1)}}\, z_{1-\delta}\right\}^3$
endif
$M = M + 1$
while $M < M_\chi$ *or* $M < M_{\chi_{min}}$
return $\mathcal{X}_t$

Algorithm test_range_measurement($z_t^k, \bar{\mathcal{X}}_t, m$):
$p = q = 0$
for $m = 1$ *to* M *do*
$p = p + z_{\text{short}} \cdot p_{\text{short}}(z_t^k \mid x_t^{[m]}, m)$
$q = q + z_{\text{hit}} \cdot p_{\text{hit}}(z_t^k \mid x_t^{[m]}, m) + z_{\text{short}} \cdot p_{\text{short}}(z_t^k \mid x_t^{[m]}, m)$
$+ z_{\text{max}} \cdot p_{\text{max}}(z_t^k \mid x_t^{[m]}, m) + z_{\text{rand}} \cdot p_{\text{rand}}(z_t^k \mid x_t^{[m]}, m)$
endfor
if $p/q \le \chi$ *then*
return accept
else
return reject
endif

Obs. : Pesquisar detalhes do **Algoritmo KLD Sampling MCL** no artigo

[1] DIETER FOX
Adapting the Sample Size in Particle Filters Through KLD-Sampling.
The International Journal of Robotics Research, Vol. 22, No. 12, pp. 985-1003, **2003**.
[doi: 10.1177/0278364903022012001]

Material disponível no Moodle

- Notas de aula.

Referências

[2] SEBASTIAN THRUN, WOLFRAM BURGARD & DIETER FOX
Probabilistic robotics.
MIT Press, **2006**.
Link: http://probabilistic-robotics.informatik.uni-freiburg.de/

[3] HOWIE CHOSET, KEVIN LYNCH, SETH HUTCHINSON, GEORGE KANTOR, WOLFRAM BURGARD, LYDIA KAVRAKI & SEBASTIAN THRUN
Principles of Robot Motion. Theory, Algorithms, and Implementations.
MIT Press, **2005**.
Link: http://biorobotics.ri.cmu.edu/book/

Contém uma descrição detalhada do filtro de Kalman e do EKF.

[4] Gregor Klancar, Andrej Zdešar, Sašo Blažic & Igor Škrjanc
Wheeled Mobile Robotics. From Fundamentals Towards Autonomous Systems.
Butterworth-Heinemann, **2017**.
Link: http://booksite.elsevier.com/9780128042045/manuscript.php

Contém códigos em Matlab.

Apresentações

- Os grupos terão cerca de 25 minutos para fazer as apresentações.
- As apresentações serão realizadas na seguinte data:

ND

Avaliação do trabalho

Preparar e enviar por email:

1. Relatório contendo a descrição dos algoritmos, resultados das simulações e discussão dos resultados.
2. Códigos dos programas utilizados nas simulações.
3. Slides preparados para a apresentação do trabalho.

Grupos

- Grupo #1
  - André Abido Figueiró
  - Tiago Bornia de Castro
- Grupo #2
  - Javier Pozzo
  - João Cardoso
  - Nicolas Lizarralde