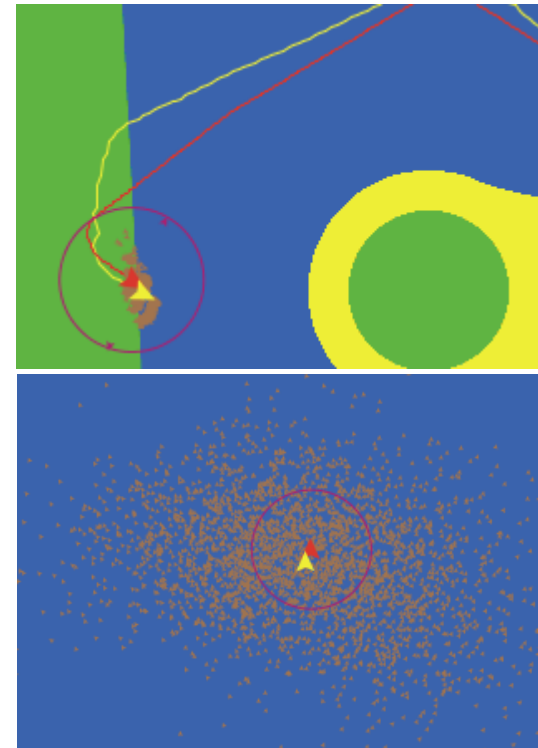
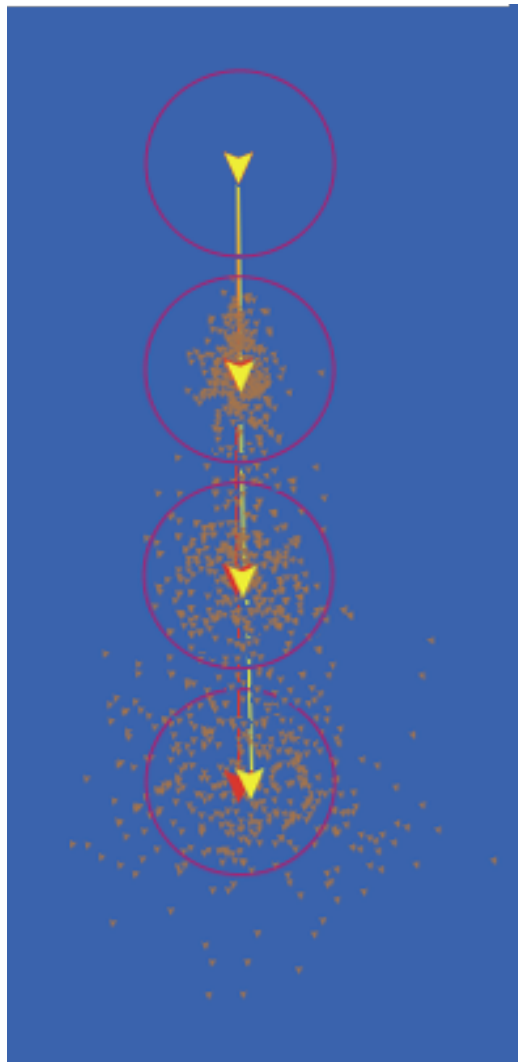


Monte Carlo Lokalizace

Martin Skalský



Proč Lokalizace?

Problém určení pozice robota
a věcí kolem něj. (filtrování dat, state
estimation)

- Je důležitá

*„Knowledge about where things are is at the
core of any succesfull physical interaction with
the environment“*

Sebastian Thrun

- Je obtížná

*„Other vehicles were very good at sensing
obstacles, but had difficulty following waypoints
or were scared of their own shadow,
hallucinating obstacles where they weren't“*

Thomas Strat

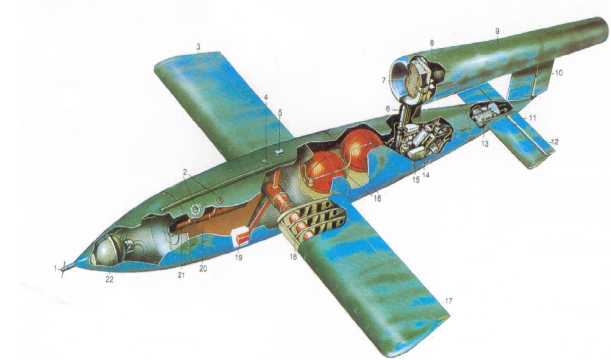


Lokalizace je obtížná

- Prostředí je částečně pozorovatelné
 - Senzory jsou drahé
 - Nepřesné
 - Poskytují pouze nepřímé informace
- Nedeterministické
- Dynamické
 - ještě navíc bych to chtěl často real time (pokud nejsem v cernu)

Tři druhy problémů

- Sledování Pozice
 - výrazně jednodušší než ostatní problémy
- Určení Pozice
 - složitost roste s rozsahem světa
- Problém Uneseného robota
 - Výrazně nejobtížnější problém



Jak na to?

Poskytneme robotovi pravděpodobnostní model situace:

- specifikujeme množinu \mathbf{X}_t skrytých a množinu \mathbf{E}_t pozorovatelných stavových proměných.

- Predikční model světa:

$$\mathbf{P}(\mathbf{X}_{t+1} | \mathbf{X}_{0:t}) = \mathbf{P}(\mathbf{X}_{t+1} | \mathbf{X}_t)$$

(z Markovova předpokladu)

- Model Senzorů:

$$\mathbf{P}(\mathbf{E}_t | \mathbf{X}_{0:t}) = \mathbf{P}(\mathbf{E}_t | \mathbf{X}_t)$$

(z Markovova předpokladu)

- Počáteční stav:

$$\mathbf{P}(\mathbf{X}_0)$$

(všimněte si vztahu s problémem lokalizace)

Pane, pojd'tě si hrát

- Je Markovův předpoklad smysluplný?
- Stačí nám takovýto model?
- Chce to ještě něco?



Pane, pojd'tě si hrát

- Je markovův předpoklad smysluplný?
-závisí na doméně



- Stačínám takový to model?

-ANO... sestojíme sdruženou distribuční funkci pro všechny proměnné:

$$\mathbf{P}(\mathbf{X}_t, \mathbf{E}_{1:t}, \mathbf{A}_{1:t}) = \mathbf{P}(\mathbf{X}_0) \prod_{i=1}^t \mathbf{P}(\mathbf{X}_i | \mathbf{X}_{i-1}, \mathbf{A}_{i-1}) \mathbf{P}(\mathbf{E}_i | \mathbf{X}_i).$$

-marginalizace nám odpoví na všechny otázky o doméně

- Chce to ještě něco? -stacionární proces, rekurzivní rovnice pro filtrování

Rekurzivní rovnice

- Umožňuje nám postupně přidávat pozorování do aktuální představy o světě(sekvenčně)

$$\mathbf{P}(\mathbf{X}_{t+1} | \mathbf{e}_{1:t+1}, \mathbf{X}_0) = \alpha \mathbf{P}(\mathbf{e}_{t+1} | \mathbf{X}_{t+1}) \sum_{x_t} \mathbf{P}(\mathbf{X}_{t+1} | \mathbf{x}_t) \mathbf{P}(\mathbf{x}_t | \mathbf{e}_{1:t}, \mathbf{X}_0).$$

-identifikujeme 2 části výpočtu

Predikci a upravení

pravděpodobností podle pozorování

Spojitá doména!



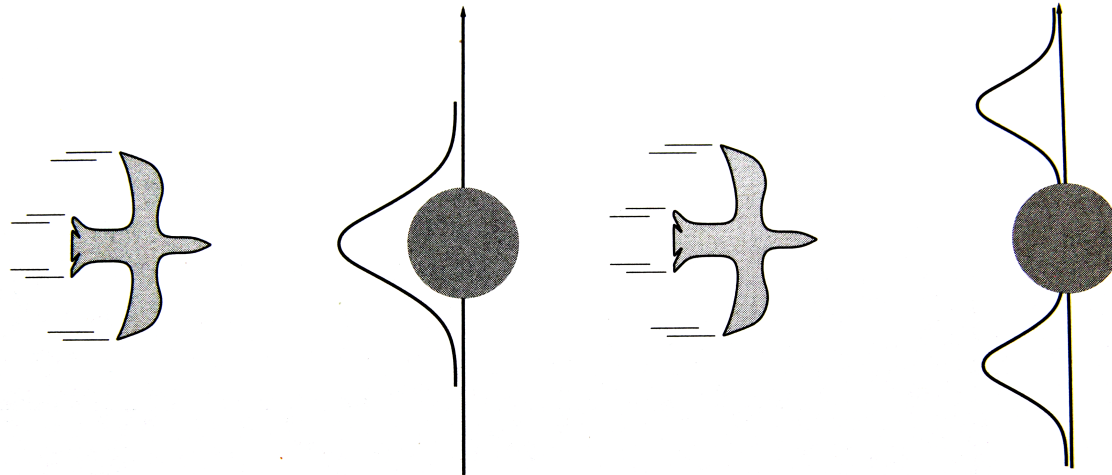
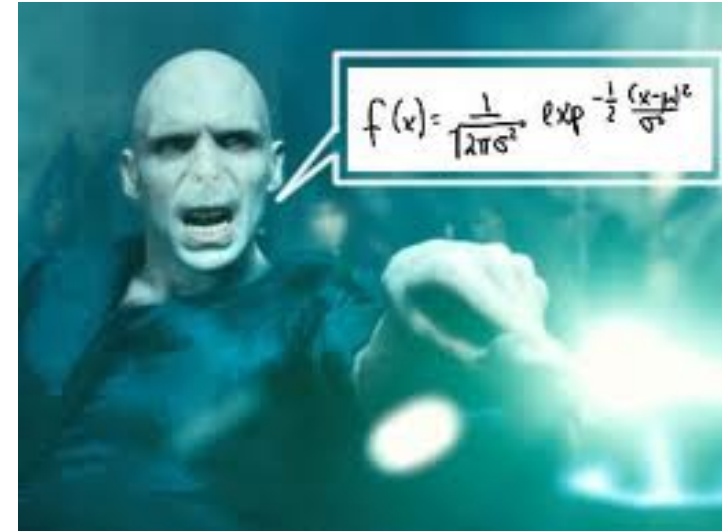
- Integrál místo násobení
- Diskretizace
- Parametrické modely- potřebujeme vyšší vyjadřovací sílu.

Diskretizace

Hidden Markov Models/Dynamic bayesian networks
-diskretizují na “mřížku“

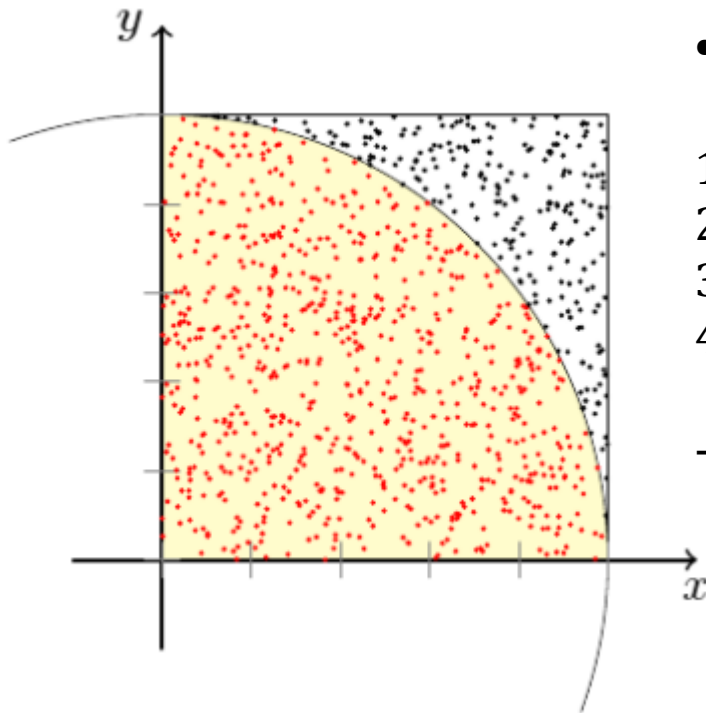
Kalmanův filter

- aproximace pomocí normálního rozdělení
- nenáročný na výpočetní sílu
- Není schopen využívat univerzálního predikčního modelu ani modelu senzorů



Monte Carlo Metody

- Monte Carlo je Rodina pravděpodobnostních algoritmů



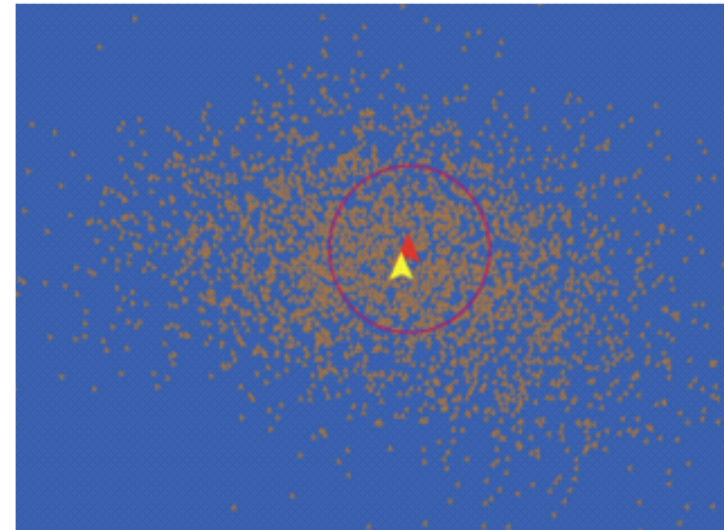
- Příklad: Výpočet π

1. Sestrojte čtverec a mu vepište kružnici
2. Vemte hrst čočky a vyhod'te do vzduchu
3. Spočtete poměr k zrnek čočky ve čtverci a v kružnici
4. $\pi = k4$

-Čím více čočky, tím větší přesnost

Monte Carlo Lokalizace

- Sekvenční Monte Carlo
- Rozložení pravděpodobnosti pozic robota je reprezentováno množinou vážených částic $P = \{p_i = (v_i, \mathbf{x}_i)\}$
 - v_i – váha
 - \mathbf{x}_i – vektor stavových proměných
- Interference probíhá nad touto množinou částic



3 kroky algoritmu

Predikční krok

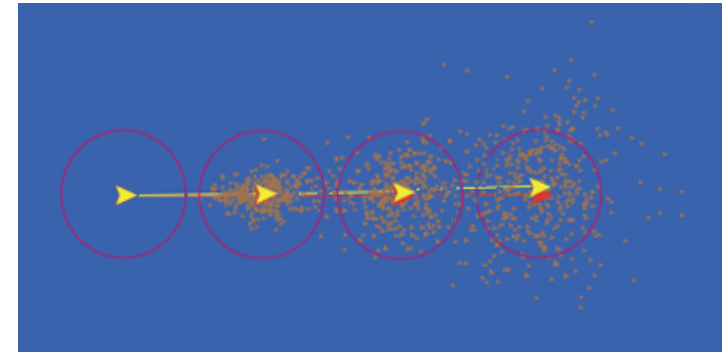
- Využijeme přechodový model :

$$\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{X}_t)$$

všem částicím vygenerujeme:

$$\mathbf{x}_{i(t+1)} = (\mathbf{X}_{t+1}|\mathbf{x}_{it})_i(\omega)$$

- Typicky se zvyšuje nejistota ohledně pozice robota
- Dead reckoning (navigace výpočtem)



3 kroky algoritmu

Korekční krok (update)

- Využívá model senzorů:

$$\mathbf{P}(\mathbf{E}_t | \mathbf{X}_t)$$

- Pomocí Bayesovy věty upravuje váhy částic:

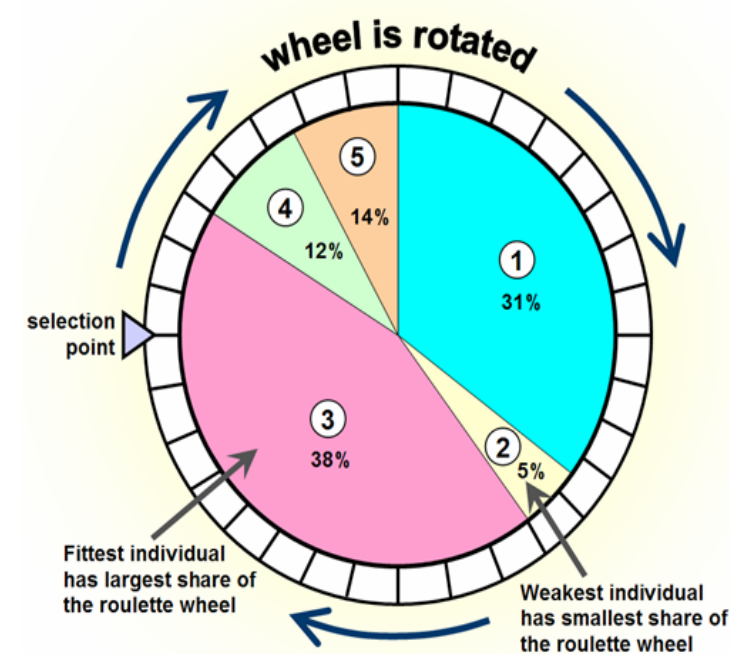
$$v'_{it} = \alpha v_{it} P(\mathbf{e}_t | \mathbf{x}_i)$$

α – normalizační faktor

3 kroky algoritmu

Převzorkování (resampling)

- Cílem je získat množinu prvků se stejnými vahami
 - prvky s malými vahami zanikají
 - s velkými se duplikují
- Snaha je omezovat šum při převzorkování
- Polya process → bojujeme proti strátě variability
- Při velkém počtu prvků hraje volba převzorkovacího algoritmu spíše okrajovou úlohu.



Vlastnosti

- Schopnost reprezentovat libovolné rozložení pravděpodobnosti.
- Akceptovat libovolný model světa a senzorů
- Přirozeně koncentruje výpočetní sílu na místa která nás zajímají
- Schopno řešit problém sledování pozice i problém globální lokalizace
- Snadná paralizovatelnost
- Možnost Hard Real Time



Vlastnosti

- Může mít problém s příliš přesnými senzory
 - I zcela selhat při použití příliš přesných senzorů



Rozšíření

- Mixture MCL
 - Snaha řešit i problém uneseseného robota
 - Část vzorků je generována pouze podle aktuálního pozorování
 - variabilní přimíchávání vzorků

Kontakt: skalsky@email.cz

Literatura:

S. J. Russell and P. Norvig, *Artificial intelligence : a modern approach*, 3rd ed., Upper Saddle River: Prentice Hall, 2009.

U. Wilensky, „NetLogo,“ Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL., 1999. [Online]. Available: <http://ccl.northwestern.edu/netlogo/index.shtml>. [Přístup získán 2. 1. 2012.].

S. Thrun, D. Fox, W. Burgard a F. Dellaert, „Robust Monte Carlo localization for mobile robots,“ *Artificial Intelligence*, sv. 128, pp. 99-141, May 2001

D. A. Levin, *Markov chains and mixing times*, Providence: American Mathematical Society, 2009