Simultaneous Localization And Mapping

Kosta Grujčić

mart 2019.

Matematički fakultet

Uvod

Šta je SLAM?

- Određivanje okruženja – mapiranje
- Određivanje pozicije agenta – lokalizacija

Šta je SLAM?

- Određivanje okruženja – mapiranje
- Određivanje pozicije agenta – lokalizacija
- **Istovremeno**

Formalna postavka

- $\mathbf{o}_t$ – opservacije u trenutku t
- $\mathbf{c}_t$ – radnja koju agent preduzima u trenutku t
- $\mathbf{x}_t$ – pozicija agenta u trenutku t

- Treba izračunati:
    - $p(\mathbf{x}_t, \mid \mathbf{x}_{0:t-1}, \mathbf{o}_{1:t-1}, \mathbf{c}_{1:t})$
    - $p(\mathbf{o_t} \mid \mathbf{x}_{0:t}, \mathbf{o}_{1:t-1}, \mathbf{c}_{1:t})$

Zašto je teško?

- *Ne znamo* da odredimo raspodele
- Neizbežan šum prlikom opažanja
- Opservacije su neprecizne na duže staze – aditivni drift

Bajesov filter

- Uvodi Markovljevo svojstvo
  - $p(\mathbf{x}_t, \mid \mathbf{x}_{0:t-1}, \mathbf{o}_{1:t-1}, \mathbf{c}_{1:t}) = p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{c}_t)$
  - $p(\mathbf{o_t} \mid \mathbf{x}_{0:t}, \mathbf{o}_{1:t-1}, \mathbf{c}_{1:t}) = p(\mathbf{o}_t \mid \mathbf{x}_t)$

- Uvodi Markovljevo svojstvo
    - $p(\mathbf{x}_t, \mid \mathbf{x}_{0:t-1}, \mathbf{o}_{1:t-1}, \mathbf{c}_{1:t}) = p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{c}_t)$
    - $p(\mathbf{o_t} \mid \mathbf{x}_{0:t}, \mathbf{o}_{1:t-1}, \mathbf{c}_{1:t}) = p(\mathbf{o}_t \mid \mathbf{x}_t)$

- Agent poseduje kvazi znanje o modelu – *verovanje*
    - $bel(\mathbf{x}_t) = p(\mathbf{x}_t \mid \mathbf{o}_{1:t}, \mathbf{c}_{1:t})$
    - $\overline{bel}(\mathbf{x}_t) = p(\mathbf{x}_t \mid \mathbf{o}_{1:t-1}, \mathbf{c}_{1:t})$

- Uvodi Markovljevo svojstvo
  - $p(\mathbf{x}_t, | \mathbf{x}_{0:t-1}, \mathbf{o}_{1:t-1}, \mathbf{c}_{1:t}) = p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{c}_t)$
  - $p(\mathbf{o_t} \mid \mathbf{x}_{0:t}, \mathbf{o}_{1:t-1}, \mathbf{c}_{1:t}) = p(\mathbf{o}_t \mid \mathbf{x}_t)$

- Agent poseduje kvazi znanje o modelu – *verovanje*
  - $bel(\mathbf{x}_t) = p(\mathbf{x}_t \mid \mathbf{o}_{1:t}, \mathbf{c}_{1:t})$
  - $\overline{bel}(\mathbf{x}_t) = p(\mathbf{x}_t \mid \mathbf{o}_{1:t-1}, \mathbf{c}_{1:t})$

- Matematička formulacija Bajesovog filtera je

$$\overline{bel}(\mathbf{x_t}) = \int p(\mathbf{x}_t \mid \mathbf{c}_t, \mathbf{x}_{t-1}) bel(\mathbf{x}_{t-1}) d\mathbf{x}_{t-1}$$

$$bel(\mathbf{x}_t) = \eta p(\mathbf{o}_t \mid \mathbf{x}_t) \overline{bel}(\mathbf{x}_{t-1})$$

Kalmanov filter

- **Pretpostavlja da je model linearan**
- **Pretpostavlja da je šum normalne raspodele**
- *Optimalan linearan filter*
- Rekurzivan postupak
  - *procena* i *ažuriranje*
  - za svaki korak je neophodan samo prethodni

Linearnost modela

- $\mathbf{x}_t = A_t \mathbf{x}_{t-1} + B_t \mathbf{c}_t + \boldsymbol{\epsilon}_t$
- $\mathbf{o}_t = C_t \mathbf{x}_t + \boldsymbol{\delta}_t$

- A_t – matrica **ekskluzivne** veze trenutne i prethodne pozicije
- B_t – matrica **ekskluzivne** veze trenutne pozicije i radnje
- C_t – matrica veze trenutne pozicije i opservacije
- $\boldsymbol{\epsilon}_t, \boldsymbol{\delta}_t$ – trenutni šum kretanja i opservacija redom
    - $\boldsymbol{\epsilon}_t \sim \mathcal{N}(0, E_t)$
    - $\boldsymbol{\delta}_t \sim \mathcal{N}(0, D_t)$

`KalmanFilter(`$\boldsymbol{\mu}_{t-1}$`, `Σ_{t-1}`, `$\mathbf{c}_t$`, `$\mathbf{o}_t$`)`

1. $\hat{\boldsymbol{\mu}}_t = A_t \boldsymbol{\mu}_{t-1} + B_t \mathbf{c}_t$
2. $\hat{\Sigma} = A_t \Sigma_{t-1} A_t^T + E_t$

3. $K_t = \hat{\Sigma}_t C_t^T (C_t \hat{\Sigma}_t C_t^T + D_t)^{-1}$ `// priraštaj`
4. $\boldsymbol{\mu}_t = \hat{\boldsymbol{\mu}}_t + K_t(\mathbf{o}_t - C_t \hat{\boldsymbol{\mu}}_t)$
5. $\Sigma_t = (I - K_t C_t)\hat{\Sigma}_t$
6. `return` $\boldsymbol{\mu}_t, \Sigma_t$

Nelinearan model – EKF

- $\mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{c}_t) + \epsilon_t$
- $\mathbf{o}_t = h(\mathbf{x}_t) + \delta_t$
- $A_t = \frac{\partial f}{\partial \mathbf{x}}\Big|_{\mathbf{x}_{t-1}, \mathbf{c}_t}$
- $C_t = \frac{\partial h}{\partial \mathbf{x}}\Big|_{\mathbf{x}_t}$

- **Zahteva više izračunavanja**
- **Nije robustan**

Neuronski pristup

State of the art

- *feature-based* (ORB-SLAM) vs. *direct methods* (LSD-SLAM)
- **upitna robusnost**
- **izražena parametrizacija**

Neural Graph Optimizer (NGO)

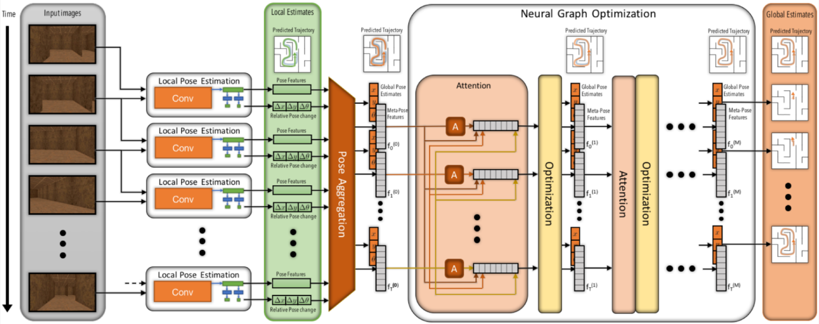
- **diferencijabilan**
- **robustan**
    - smanjuje aditivni drift

- *Front-end* je *FlowNet*
- *Back-end* je *Transformer*

Metod

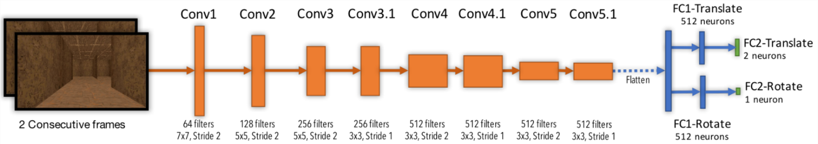
- **Mreža lokalne procene**
  - upareni frejmovi $\rightsquigarrow$ lokalna procena pozicije
  - CNN + FC
- **NGO**
  - lokalna procena pozicije $\rightsquigarrow$ globalna procena pozicije
  - *soft-attention* + *optimization* (TCN)

Model



local pose estimation i global pose estimation

Front-end



FlowNet varijanta

Attention

- Za dati *embedding* niz $\mathbf{F}^{(i-1)} = (f_1^{(i-1)}, ..., f_T^{(i-1)})$ se računa odgovarajući *query* niz $(q_1^{(i-1)}, ..., q_T^{(i-1)})$ koristeći FCL

- *Attention* vektor $a_t^{(i-1)}$ se računa:

$$C_{tu} = \langle q_t, f_u \rangle$$

$$\alpha_{tu} = \frac{C_{tu}}{\sum_{v=1}^{T}}$$

$$a_t = \sum_{v=1}^{T} \alpha_{tu} \odot f_u$$

Optimization

- Vrši se konkatenacija *attention* vektora i *embedding* vektora i ubacuje u TCN:

$$\begin{bmatrix} \mathbf{F}^{(i)} \\ \nabla\mathbf{P}^{(i)} \\ \beta^{(i)} \end{bmatrix} = \sigma_L\left(h_L\left(...h_1\left(\begin{bmatrix} f_1^{(i-1)} \\ a_1^{(i-1)} \end{bmatrix} ... \begin{bmatrix} f_T^{(i-1)} \\ a_T^{(i-1)} \end{bmatrix}\right)...\right)\right)$$

gde je $\nabla\mathbf{P}^{(i)} = (\nabla p_1^{(i)}, ..., \nabla p_T^{(i)})$

- Vrši se dodatno ufinjavanje:

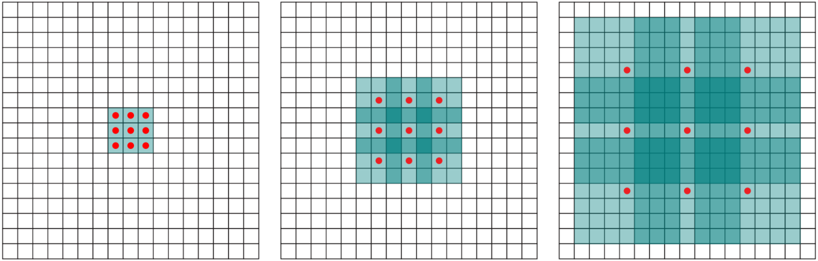
$$\Delta p_j^{(i)} = \Delta p_j^{(i-1)} + \beta_j^i \nabla p_j^{(i)}$$

Temporalne konvolutivne mreže

Dilataciona konvolucija

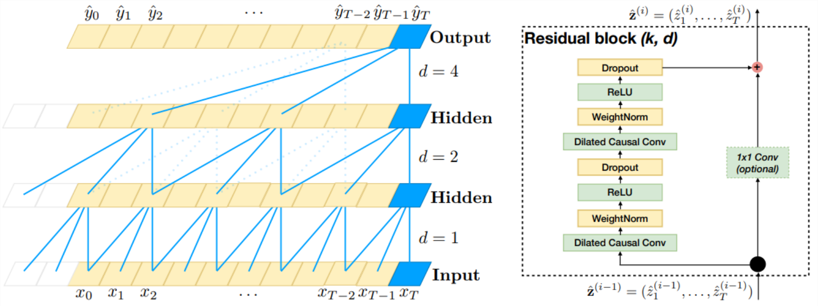
- $(F *_l k)(\mathbf{p}) = \sum_{\mathbf{s}+l\mathbf{t}=\mathbf{p}} F(\mathbf{s})k(\mathbf{t})$
    - za $l = 1$ postaje obična konvolucija

- $F_{i+1} = F_i *_{2^i} k_i$ za $i = 0, 1, ..., n-2$
    - receptivno polje od F_i je $(2^{i+1} - 1) \times (2^{i+1} - 1)$



F_1, F_2 i F_3 redom

Model

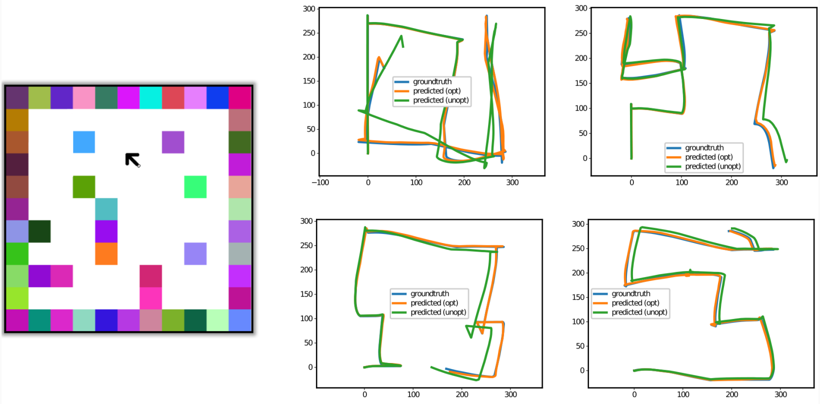


Kauzalni temporalno-konvolutivni sloj

TCN vs. RNN

- *Seq. MNSIT* (99.0 vs. 96.2)$^{\uparrow}$
- *Adding problem* (97.2 vs. 87.3)$^{\uparrow}$
- *Copy memory* (3.5^{-5} vs. 0.0197)$^{\downarrow}$
- *World-level PTB* (88.68 vs. 78.93)$^{\downarrow}$
- *Char-level PTB* (1.31 vs. 1.36)$^{\downarrow}$
- ...

Rezultati – 2D

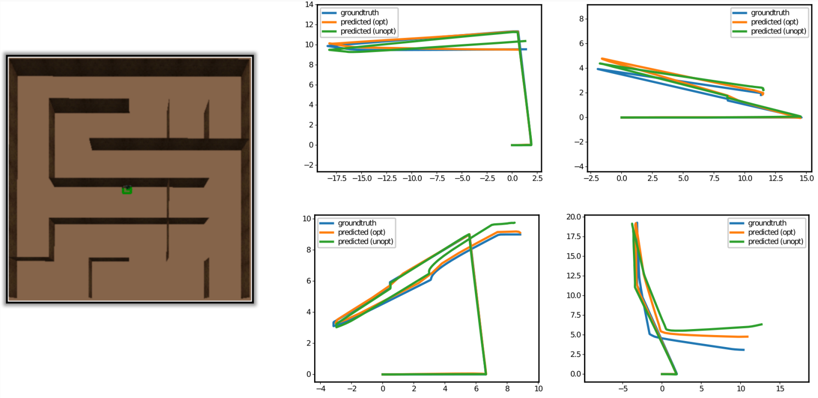


Primer 2D lavirinta (*Box2D*) i trajektorija agenta

Metrika – RMSE po poziciji

Model	**Test**
Att-Opt $\times$ 0	17.8
Att-Opt $\times$ 1	10.21
Att-Opt $\times$ 5	3.16

Rezultati – 3D



Primer 3D lavirinta (*ViZDoom*) i trajektorija agenta

Metrika – % trans. i root.

Model	Trening		Test	
	$\%Err.trans.$	$\%Err.rot.$		
Att-Opt $\times$ 0	1.65	0.117	1.62	0.122
Att-Opt $\times$ 1	1.42	0.071	1.16	0.071
Att-Opt $\times$ 5	**1.25**	**0.057**	**1.04**	**0.056**
DeepVO	1.78	0.079	2.39	0.091

Analiza

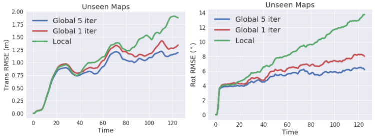


Figure 7. Translational (Left) and Rotational (Right) RMSE as a function of number of images in the trajectory in **unseen mazes**.

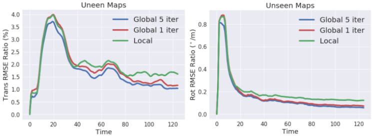


Figure 9. Ratio of the Translational (Left) and Rotational (Right) RMSE to the distance travelled as a function of number of images in the trajectory in **unseeen mazes**.

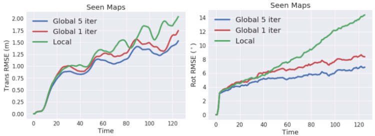


Figure 8. Translational (Left) and Rotational (Right) RMSE as a function of number of images in the trajectory in **seen mazes**.

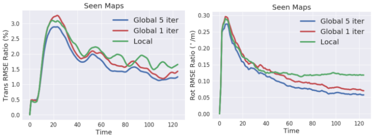


Figure 10. Ratio of the Translational (Left) and Rotational (Right) RMSE to the distance travelled as a function of number of images in the trajectory in **seen mazes**.

Dalji rad[1]

- „For many applications and environments, numerous major challenges and important questions remain open. To achieve truly robust perception and navigation for long-lived autonomous robots, more research in SLAM is needed."
- „In some applications, such as self-driving cars, precision localization is often performed by matching current sensor data to a high definition map of the environment that is created in advance."
- „One may even devise examples in which SLAM is unnecessary altogether and can be replaced by other techniques."

[1] http://rpg.ifi.uzh.ch/docs/TRO16_cadena.pdf

Reference

- **C. Cadena et al.: Past, present and future of SLAM: Toward the robust-perception age**
- **E. Parisotto et al.: Global pose estimation with an attention-based recurrent network**
- A. Dosovitskiy *et al.*: Flownet: Learning optical flow with convolutional networks
- R. Kümmerle *et al.*: g^{2}o: A general framework for graph optimization
- A. I. Mourikis and S. I. Roumeliotis: A multi-scale constraint Kalman filter for vision-aided inertial navigation
- R. Mur-Artal *et al.*: ORB-SLAM: A versatile and accurate monocular SLAM system
- A. Vaswani *et al.*: Attention is all you need