Visual Localization Under Appearance Change: A Filtering Approach

Anh-Dzung Doan [1] Yasir Latif [1 2] Tat-Jun Chin [1 2] Yu Liu [1 2]
Shin Fang Ch'ng [1 2] Thanh-Toan Do [3] Ian Reid [1 2]

[1]The University of Adelaide

[2]Australian Centre for Robotic Vision

[3]University of Liverpool

February 15, 2024

Overview

Visual localization under appearance change

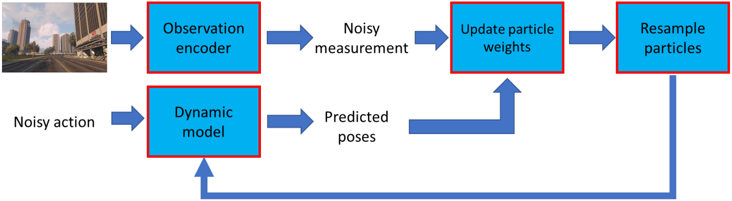
Self-driving cars → Planning & navigation → Visual localization

A major challenge in visual localization for autonomous driving is to be robust against appearance changes

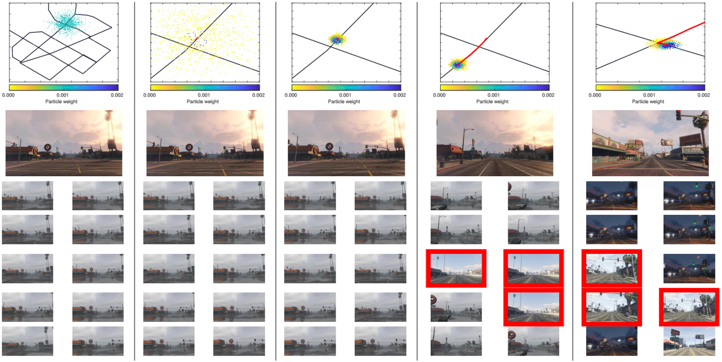
Our solution:

Temporal inference → Particle filter
- Monte Carlo principle
- Observation encoder

Monte Carlo-based visual localization



Example:



Dynamic model

Let the 6 DoF camera pose be given by:

$$s_t = [r_t, \, \Omega_t]^T$$

where,

- r_t: 3D position at time t
- Ω_t: Euler orientation at time t
- u_t: noisy action at time t
- z_t: noisy measurement at time t
    - $u_{1:t}$, $z_{1:t}$: noisy action and measurement up to time t

We represent $p(s_t | u_{1:t}, z_{1:t})$ with a set of N particles:

$$\mathcal{S}_t = \{s_t^{[1]}, s_t^{[2]}, ..., s_t^{[N]}\}$$

$$\mathcal{W}_t = \{w_t^{[1]}, w_t^{[2]}, ..., w_t^{[N]}\}$$

Dynamic model

Randomly sample noisy action $u_t^{[i]} = [v_t^{[i]}, \psi_t^{[i]}]^T$ according to Gaussian distribution:

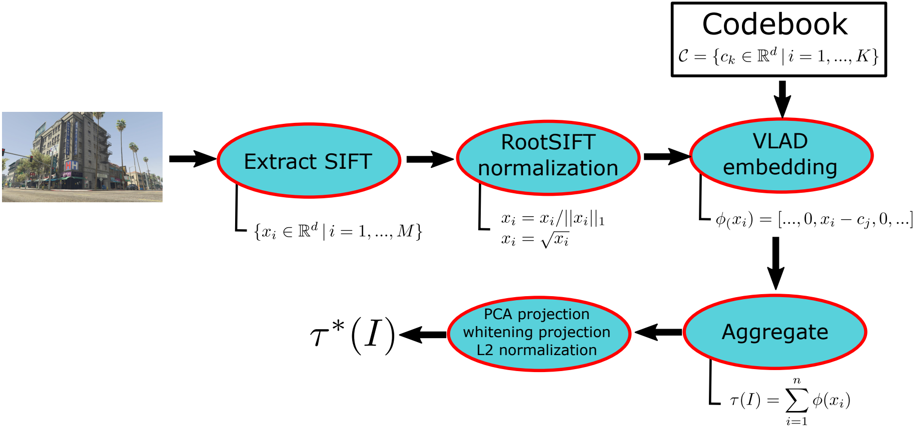
$$v_t^{[i]} \sim \mathcal{N}(\mu_v, \Sigma_v)$$
$$\psi_t^{[i]} \sim \mathcal{N}(\mu_\psi, \Sigma_\psi)$$

Our proposed motion model:

$$s_t^{[i]} = \begin{bmatrix} r_{t-1}^{[i]} + v_t^{[i]} \\ \varphi^{-1}\left(\varphi(\psi_t^{[i]}).\varphi(\Omega_{t-1}^{[i]})\right) \end{bmatrix}$$

where, $\varphi(.)$ be a function that maps an Euler representation to Direction Cosine Matrix (DCM) and $\varphi^{-1}(.)$ is its inverse mapping.

Observation encoder



1. Retrieve nearest images in database given a query
2. Use meanshift to select largest cluster
3. Calculate mean of translation and rotation $\rightarrow$ noisy measurement z_t

Updating particle weights & resampling

For each particle, its weight is computed:

$$w_t^{[i]} = p\left(z_t | s_t^{[i]}\right) \propto e^{-\frac{1}{2}(z_t - s_t^{[i]})^T \Sigma_o^{-1} (z_t - s_t^{[i]})}$$

All particle weights are normalized:

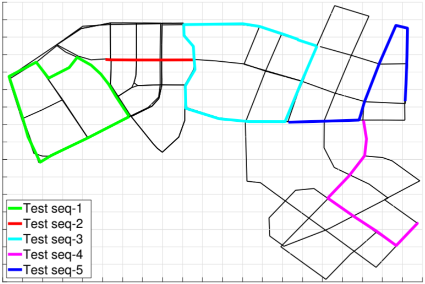
$$\forall i, w_t^{[i]} = \frac{w_t^{[i]}}{\sum_{j=1}^{n} w_t^{[j]}}$$

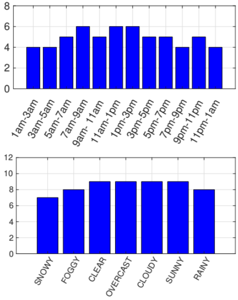
Particles are resampled using Stochastic universal sampling [1]

[1] D. Whitley, "A genetic algorithm tutorial," Statistics and computing, 1994.

Experiments on synthetic dataset

Data collected from computer game Grand Theft Auto V (GTA V) using G2D [2] [3]





We simulate there are 59 vehicles running in different routes, weathers and times of day. Coverage area is 3.36 km^2

[2] A.-D. Doan, A. M. Jawaid, T.-T. Do, and T.-J. Chin, "G2D: from GTA to Data," arXiv preprint arXiv:1806.07381, pp. 1–9, 2018

[3] https://github.com/dadung/G2D

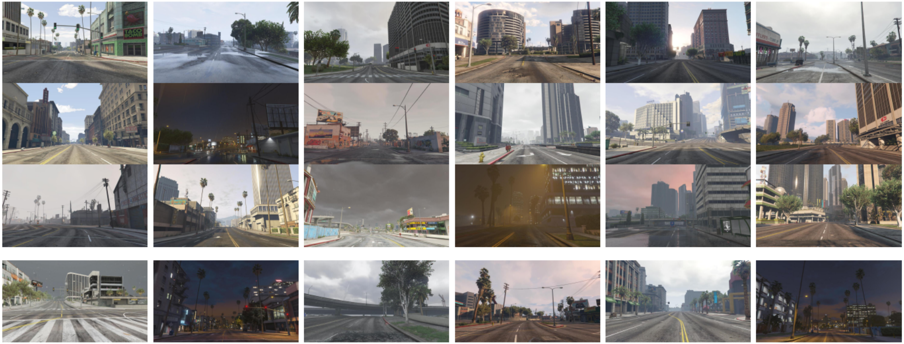
Experiments on synthetic dataset

Testing sequences information:

Sequences	# images	Time & Weather	Traversal distance
Test seq-1	1451	9:36 am, snowy	1393.03m
Test seq-2	360	10:41 pm, clear	359.74m
Test seq-3	1564	11:11 am, rainy	1566.93m
Test seq-4	805	6:26 pm, cloudy	806.56m
Test seq-5	1013	3:05 pm, overcast	1014.91m

Table: Statistics of the testing sequences in the synthetic dataset

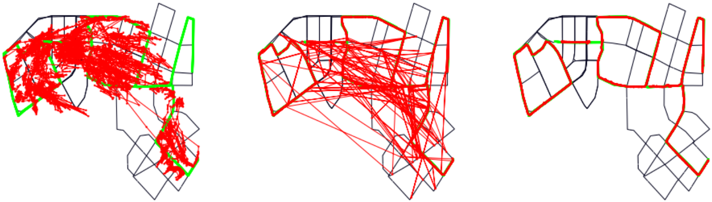
Experiments on synthetic dataset



Dataset is published in: `http://tiny.cc/jd73bz`

Experiments on synthetic dataset

We compare against MapNet [4] and image retrieval [5]



(a) MapNet (b) Image retrieval (c) Our method

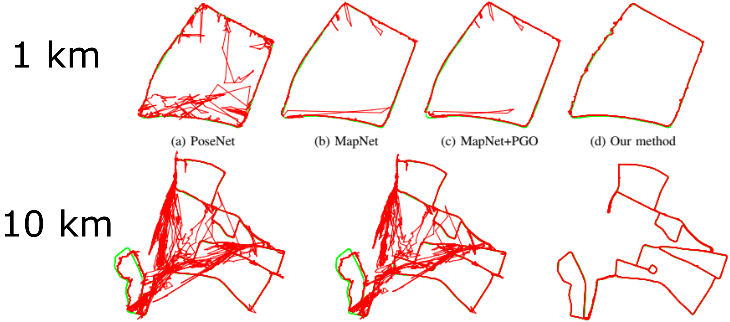
	MapNet	Image retrieval	Our method
Test seq-01	37.45m, 4.61°	**2.57m**, **3.31**°	2.63m, 3.46°
Test seq-02	31.06m, 0.96°	**4.31m**, **1.38**°	6.12m, 3.32°
Test seq-03	98.34m, 4.28°	3.29m, **3.47**°	**3.21m**, 4.03°
Test seq-04	38.50m, 1.53°	2.73m, **1.17**°	**2.58m**, 1.82°
Test seq-05	807.93m, 9.71°	**1.78m**, **7.06**°	1.83m, 7.29°

[4] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, and J. Kautz, "Geometry-aware learning of maps for camera localization," in CVPR, 2018

[5] Our proposed observation encoder without motion model

Experiments on real dataset (Oxford RobotCar)

We compare against MapNet [6], MapNet with pose graph optimization (PGO) and PoseNet [7]



(a) PoseNet (b) MapNet (c) MapNet+PGO (d) Our method

(e) MapNet (f) MapNet+PGO (g) Our method

[6] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, and J. Kautz, "Geometry-aware learning of maps for camera localization," in CVPR, 2018

[7] A. Kendall, M. Grimes, and R. Cipolla, "PoseNet: A convolutional network for real-time 6-DoF camera relocalization," in CVPR, 2015

Conclusion

- A practical filtering approach to exploit the temporal smoothness of an image sequence
- An observation encoder robust against appearance change
- A synthetic dataset from GTA V (`http://tiny.cc/jd73bz`)
- Code will be available soon

THANK YOU FOR YOUR ATTENTION