Scalable Place Recognition Under Appearance Change for Autonomous Driving

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

[1]The University of Adelaide

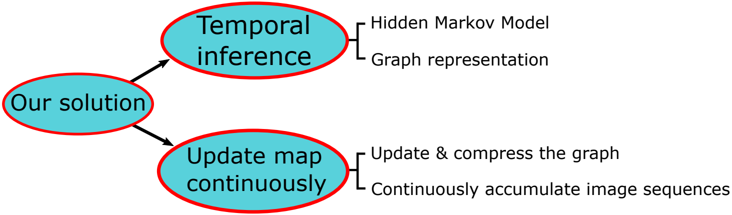
[2]Australian Centre for Robotic Vision

[3]University of Liverpool

February 15, 2024

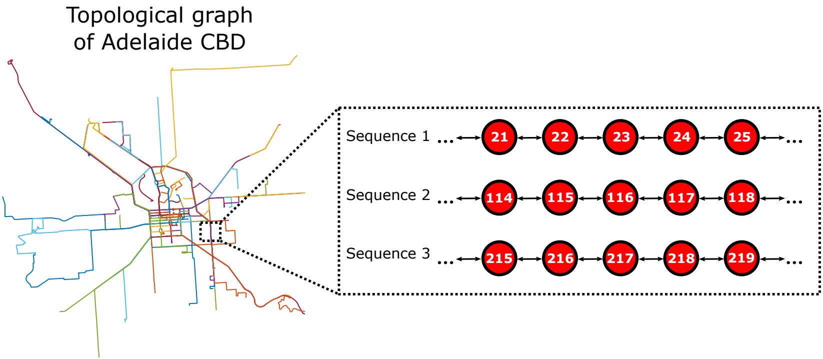
Introduction

A major challenge in place recognition for autonomous driving is to be robust against appearance changes



Graph representation

Given a dataset of M videos: $\mathcal{D} = \{\mathcal{V}_1, ..., \mathcal{V}_M\}$.
Image indices are "unroll" to represent a map as a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$:



A set of nodes $\mathcal{N} = \{1, ..., K\}$ are image indices/places.
Edge weights $w \in \mathcal{E}$ are transition probabilities between places k_1 and k_2:

$$w(\langle k_1, k_2 \rangle) = P(k_2 \mid k_1) = P(k_1 \mid k_2)$$

Query video: $\mathcal{Q} = \{Q_1, Q_2, \ldots, Q_T\}$

HMM inference

We denote transition matrix $\mathbf{E} \in \mathbb{R}^{K \times K}$, where,

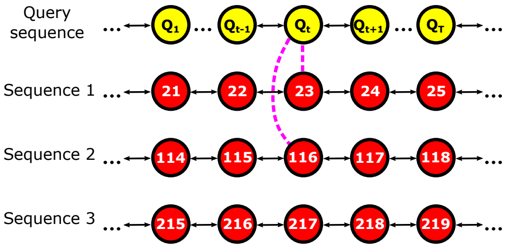
$$\mathbf{E}(k_1, k_2) = P(s_t = k_2 | s_{t-1} = k_1)$$

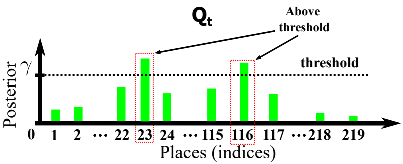
The observation model is a diagonal matrix $\mathbf{O}_t \in \mathbb{R}^{K \times K}$ obtained from image retrieval, where,

$$\mathbf{O}_t(k, k) = P(Q_t | s_t = k)$$

Belief $\mathbf{p}_t$ is calculated using matrix computation

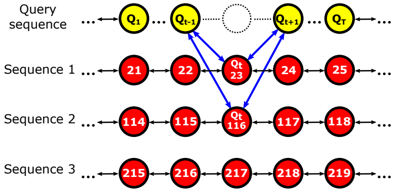
$$\mathbf{p}_t = \eta \mathbf{O}_t \mathbf{E}^T \mathbf{p}_{t-1}$$



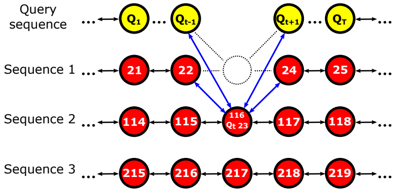


Graph update & compression

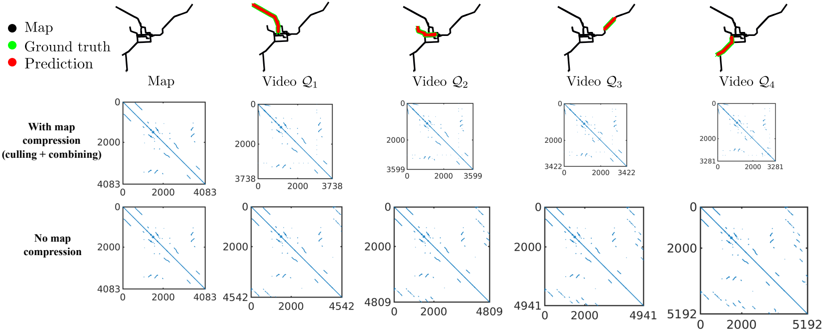
Culling



Combining



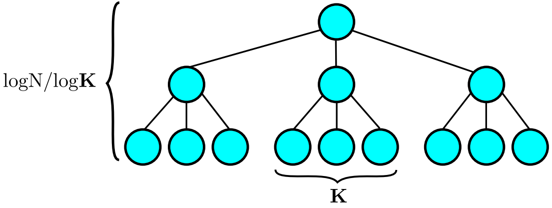
Result on Adelaide CBD dataset sourced from Mapillary



Updating observation model

$\mathcal{Q}$ is appended to the dataset, i.e., $\mathcal{D} = \mathcal{D} \cup \mathcal{Q}$, all vector $\psi(Q_t)$ is indexed to k-means tree, where,

- $\psi(.) \in \mathbb{R}^{D'}$: maps an image to a single high-dimensional vector
- N and T are $|\mathcal{D}|$ and $|\mathcal{Q}|$ respectively.



Assume the tree is balance, cost for adding $\mathcal{Q}$ is $O\left(T\mathbf{K}D'(log\ N/log\ \mathbf{K} + 1)\right)$

Training & testing time

Dataset: Oxford RobotCar (8 different sequences along a same route)

Training time

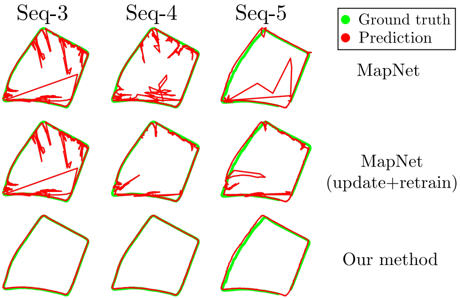
Training sequences	VidLoc	MapNet	Our method
Seq-1,2	14.1h	11.6h	**98.9s**
Seq-3	-	6.2h	**256.3s**
Seq-4	-	6.3h	**232.3s**
Seq-5	-	6.8h	**155.1s**
Seq-6	-	5.7h	**176.5s**
Seq-7	-	6.0h	**195.4s**

Sequences	Inference time (ms)
Seq-3	4.03
Seq-4	4.82
Seq-5	4.87
Seq-6	3.72
Seq-7	3.78
Seq-8	3.68

Place recognition accuracy

Mean error

Methods	Seq-3	Seq-4	Seq-5
VidLoc	38.86m, 9.34°	38.29m, 8.47°	36.05m, 6.81°
MapNet	9.31m, 4.37°	8.92m, 4.09°	17.19m, **5.72**°
MapNet (update+ retrain)		8.71m, 3.31°	18.44m, 6.94°
Our method	**6.59m**, **3.28**°	**6.01m**, **3.11**°	**15.88m**, 5.91°



THANK YOU FOR YOUR ATTENTION

POSTER ID: #17