2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

SPRINT: Subgraph Place Recognition for INtelligent Transportation

Yasir Latif, Anh-Dzung Doan, Tat-Jun Chin, and Ian Reid

Fig. 1: Same place, different days: The dataset sourced from Mapillary contains the same area covered under various weather and lighting conditions. The data is crowd sourced and has significant variation in the cameras used and how they are placed inside the car. Images above show the same place captured during eight different runs in Adelaide, Australia.

***Abstract*— Visual place recognition is an important problem in mobile robotics which aims to localize a robot using image information alone. Recent methods have shown promising results for place recognition under varying environmental conditions by exploiting the sequential nature of the image acquision process. We show that by using k nearest neighbours based image retrieval as the backend, and exploiting the structure of the image acquisition process which introduces temporal relations between images in the database, the location of possible matches can be restricted to a subset of all the images seen so far. In effect, the original problem space can thus be restricted to a significantly smaller subspace, reducing the inference time significantly. This is particularly important for scalable place recognition over databases containing millions of images. We present large scale experiments using publicly sourced data that show the computational performance of the proposed method under varying environmental conditions.**

I. Introduction

Place recognition is the problem of localizing a mobile agent using an on-board camera as the sole sensor [2, 12]. In contrast to loop closure detection which aims to recover the robot's (mostly 6-DoF) pose with respect to a previously built map [6, 9, 17], place recognition is topological in nature, that is, it matches the current query image to one of the previously seen images (or the indices thereof) that are closest in appearance and hence in physical world. If there is no change in the viewpoint and appearance, this is a trivial task and can be addressed using a nearest-neighbour based lookup. However, we live in a perpetually changing world which often violates this assumption especially when the operational life time of the robot gets longer and appearance changes due to weather, season and time of day come into play. The problem is further compounded by changes in the sensor's viewpoint, making it difficult to match current observations with those in the past.

YL, AD, TJ, and IR are with the Australian Center for Robotic Vision (ACRV) at the University of Adelaide firstname.lastname@adelaide.edu.au.

The work in [8] proposed a scalable place recognition system combining Hidden Markov Models (HMMs) based inference and a k-means tree based image retrieval system. For each query image, the image retrieval method returns k possible matches, which are refined temporally using a HMM. The state on which this HMM operates spans all the images seen in the past, termed the database. In this work, we make the observation that the database is not a random collection of images but is structured due to temporal dependencies arising from the sequential nature of the image acquisition process. The image retrieval backend returns k nearest neighbour hypothesis for each query image. These hypothesis can come from at most k different sequences (subgraphs) in the database, allowing the inference to be limited to specific sequences in the database, instead of the whole database. Formally, if the graph $\mathcal{G}$ represents the complete database, due to the sequential image acquisition process, it is a union of at most n *unconnected* subgraphs: $\mathcal{G}(\mathcal{V}, \mathcal{E}) = \cup_{i=1}^{n} \mathcal{G}_i(\mathcal{V}_i, \mathcal{E}_i)$, where each of these graphs is either a single run in time (sequence) or a collection of sequences between which connections have been previously established. The k matches returned by the retrieval method therefore select a subset of these subgraphs spanning only the sequences involved (Fig. 2) This partitioning allows efficient inference on a much smaller than that of the whole database (Sec .III-F) More concretely, this work makes the following contributions:

- We propose a method that takes advantage of the

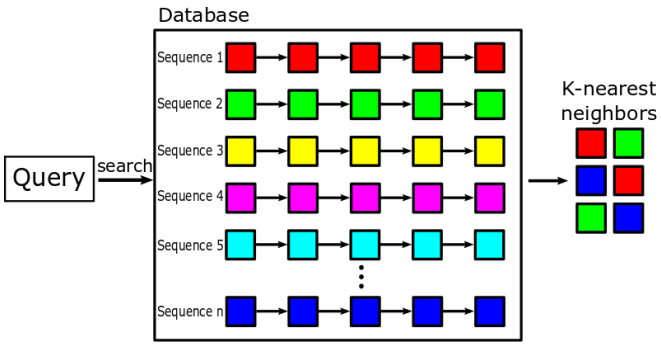

Fig. 2: The database consists of sequences collected over time (represented by different colors). For each query, the image retrieval backend returns k matches that can come from at most k different sequences. This information can be used to restrict future inference to only the selected sequences (red, green, blue here).

sequential nature of image collection, along with the limited number of hypothesis returned for each query, to limit the state space for inference.

- We show that using temporal inference with weak descriptors, a boost in accuracy comparable to deeply learned methods can be achieved.
- We show the performance of the proposed SPRINT framework on large scale uncontrolled data collected from Mapillary (Fig. 1) to show the gain in computation performance at inference time.

II. Related Work

The approach to robust place recognition can be divided into two main approaches.

Image representation learning: Representation learning approaches have received a lot of attention due to the recent advances in deep learning techniques. The first step toward scalable inference is robust feature representation, that is, each image is mapped independently to a unique representation. Methods in this approach learn a transformation that maps each image to a latent representation such that images from the same place end up closer to each other in the embedding space and images from different places are mapped far away from each other [15, 11, 1, 16, 5, 3, 18]. Some of these approaches also address the problem using Generative Adversarial Networks (GANs) [19] which are trained to generate images under a certain "neutral" condition so that traditional methods that fail due to appearance change, can still function on images by transforming them into the neutral representation.

Temporal inference: In a robotics context, instead of a single image, information always comes as a stream of images, for example from on-board cameras constantly capturing the environment. Place recognition methods can take advantage of the temporal relations between images [14, 8] to address appearance changes over the life time of a robot. This problem was initially addressed in SeqSLAM [14], who showed that taking temporal information into account allows aligning image sequences with varied appearance change. The basic premise of [14] is that even though a single image might not contain enough information for localization, considering images in a sequence allows robust localization. This is achieved by normalizing images to get a more "stable" representation.

The problem of lifelong place recognition – making decision over span of months and years – has been addressed in [4, 10]. In [4] this take the form of visual odometry segments stored as "experiences" when localization fails. Over time this gives a stable all-condition localization system. However, their methods depends on VO as the backbone, whose failure can be attributed to reasons besides localization failures. In [10] an image based place recognition method was proposed. Large scale operation was enabled by a memory model that moved plausible hypotheses (images) to main memory from a long-term memory when needed. However, inference is restricted to 1-hop neighbours in the transition matrix that can bias results. Our work is close in spirit to both of these works but focuses on computation efficiency for large scale place recognition.

III. Method

Problem Setting: We consider the case of autonomous vehicles with front facing cameras moving in an urban environment. Each vehicle is capable of transmitting the image stream to a remote server. When a query image arrives, it is processed against the database to form place recognition hypothesis which is communicated back to vehicle. Once all the images in the query sequence have been processed, the query sequence along with the place recognition decisions are merged into the database. At the start, when the database is empty, the first sequence is added directly as no place recognition decisions can be made.

The operational life time of the robot subjects the images to appearance variation as well as structural changes such as new building, temporary traffic control etc. Due to the geometry of the road, each vehicle is restricted in lateral motion by the number of lanes on the road leading to restricted changes in viewpoint. However, in the wild especially when users can put their own cameras inside the vehicle, viewpoint variants can come from how these devices are placed with respect to road. We treat the case of moving in different directions on the same road as different places because image information alone is not enough to make place recognition decision.

This work focuses on computationally efficient inference for place recognition and does not address the complexity of image retrieval system itself. The reported metrics in the experiment section include inference time for the HMM only.

We first provide a brief overview of Hidden Markov Models (HMMs) that form the basis of our Bayesian inference mechanism.

A. Hidden Markov Models

HMMs are recursive filters and update the probability of being in state i at time t ($\mathbf{x}_i^t$) given the observation z^t via the following update rule:

$$\mathbf{P}(\mathbf{x}_i^t|Z_t) = \mathbf{P}(z^t|\mathbf{x}_i^t)\sum_j \mathbf{P}(\mathbf{x}_i^t|\mathbf{x}_j^{t-1})\mathbf{P}(\mathbf{x}_j^{t-1}|Z_{t-1}) \quad (1)$$

where $Z_{t-1} = [z^0, z^1, \ldots, z^{t-1}]$ is the set of all previous observations. This update equation represents two update steps:

a) Propagation: the weighted accumulation of probabilities from all the states $\mathbf{x}_j^{t-1}$ at time $t-1$ which can lead to the state $\mathbf{x}_i^t$ at time t and

b) Update: once an observation is made, the accumulated probability is further weighted by the probability of making the observation z_t, given the state x_i at time t.

In the context of place recognition, the state consists of all the images that have been seen in the past, that is, $i = \{1 \ldots n\}$ contained in the database. The database is temporally structured and consists of K sequences $[s_1, s_2, \ldots s_k]$, each with $|s_i|$ images such that $n = \sum_{i=0}^{K} |s_i|$. The images indices are laid out sequentially, such that for the image i in sequence k, the location in the database is given by

$$\text{db}(k, i) = \sum_{p=1}^{k-1} |s_p| + i$$

In addition, the last image in each sequence connects to a special `sink` state, which prevents belief from being propagated across sequence boundaries.

The probability distribution over the images in the database is represented by a vector $\mathbf{x}^t \in \mathcal{R}^{n+1}$, where the i-th element $\mathbf{x}_i^t$ represents the probability of being at the i-th location at time t. At the start, the first sequence observed is added in its entirety to the database. From then on, for each new image observed, the follow sequence of actions take place :

B. Initialization

The initial belief, without any observations, is uniformly distribution over all the indices in the database: $\mathbf{x}_t^i = \frac{1}{N}$ for all $i \in 1 \ldots N$ such that $\sum_i^N \mathbf{x}_t^i = 1$ as is required for a valid probability distribution. This step is carried out for the first image in the query sequence only.

C. Propagation – The transition matrix

The transition matrix $\mathbf{M}$ stores the probability of moving to a state j from i at the i-th row and j-th column M_{ij}. Generally, the transition matrix is symmetric, signifying that that the move $i \rightarrow j$ is as likely as the move $j \rightarrow i$ and therefore $\mathbf{M}_{ij} = \mathbf{M}_{ji}$. Additionally, rows and columns of the transition matrix sum up to one to enforce valid incoming and outgoing transition probability distributions. The belief is propagated via the matrix-vector product

$$\hat{\mathbf{x}^t} = \mathbf{M}\mathbf{x}^{t-1} \quad (2)$$

where $\hat{\mathbf{x}^t}$ represent the belief before incorporating the current observations. The computation complexity of this matrix-vector product is $\mathcal{O}(n^2)$ for a matrix of size $n \times n$.

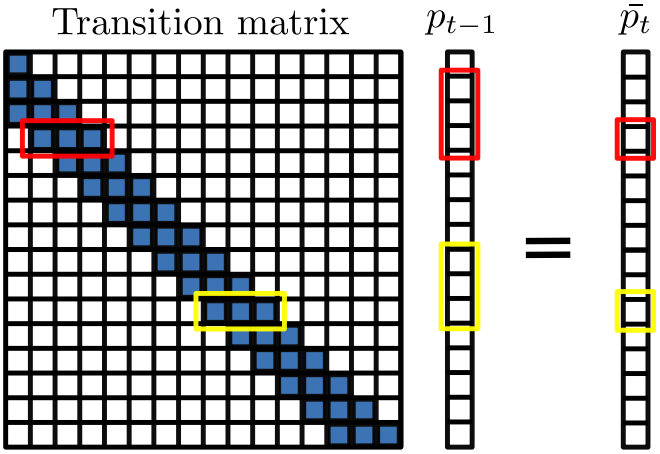

Fig. 3: Diagonal structure of the HMM problem

D. Observation based update

Once an observation is made, the observation probability $\mathbf{P}(z^t|\mathbf{x}_i^t)$, which is the probability of making the observation z^t given the the state is $\mathbf{x}_i^t$ is computed via some distance metric between the current image and the image residing at the i-th location of the database.

If we represent $\mathbf{z}^t$ as the vector containing $\mathbf{P}(z^t|\mathbf{x}^t)$ for every $\mathbf{x}_i^t$, then the belief update step can be carried out as the element wise multiplication:

$$\mathbf{x}^t = \eta\ \mathbf{z}^t \oplus \hat{\mathbf{x}^t} \quad (3)$$

where η is a normalization constant and $\oplus$ signifies element-wise multiplication. This gives us the probability distribution over all the states given all the observations. This computational complexity associated with this element-wise operation is $\mathcal{O}(n)$, for vectors of size n. Overall complexity of a naive HMM implementation is therefore $\mathcal{O}(n^2)$.

E. Sparse Transition Matrix

We first make some observations about the structure of the problem and show how this leads to an efficient implementation due to the sparsity of the problem. Place recognition is a sparse problem, in the sense that for any observed image, the set of possible matching images are very few (ideally 1) compared to all the images that have been seen in the past. This translates into a sparse state transition matrix as any place is connected to very few other places. By exploiting sparsity in the structure of the transition matrix, we can develop more efficient large scale solution at the naive solution has a quadratic complexity in the number of states.

Directionality: In a general setting, the transition matrix is symmetric meaning that a sequence of events going forward in time and going back are equivalent from the point-of-view of the Bayesian inference. However, for autonomous vehicles with front mounted cameras, the case of reversal is highly unlikely, as reversing on major road is illegal in almost all parts of the world[1]. Therefore, we model our transition

[1]Australian Road Rules - Regulation 296

matrix using directed edges. Moving in the opposite direction is treated as a different sequence since visual information alone might not be able to recognize the same place from opposing viewing directions.

Sparsity: The transition matrix contains two types of connections: the first are introduced by the nature of image collection and encode the belief that if the vehicle was at a location p at time t, the next image observed is highly likely to be in the temporal vicinity of p. To be robust to variations in vehicle speed, in addition to connecting neighbouring nodes, we connect the current node to a fixed number of nodes which we call the adjacency of p, denoted by $\mathcal{W}(p)$. This represent the set of directed connections from which the current state can be reached in a single hop. This gives the transition matrix a Toeplitz structure, as seen from Fig. 3, which is highly sparse.

The other set of connection comes from matching places over time and form the off diagonal entries (not shown). These are introduced when we are able to successfully recognize the same place at different instances of time.

A general symmetric state transition matrix can be decomposed as the sum of three matrices

$$\mathbf{M} = \mathbf{D} + \mathbf{U} + \mathbf{U^T}$$

where $\mathbf{D}$ is the Toeplitz matrix containing local relationship between nodes (see Fig. 3), $\mathbf{U}$ contains inter-sequences connections and $\mathbf{U^T}$ in the symmetric counterpart of $\mathbf{U}$. In the current system, we only maintain directed edges and therefore $\mathbf{U^T}$ is zero.

Each row of $\mathbf{D}$ encodes $\mathcal{W}(p)$ defined by the temporal adjacency and is fixed since this probability only depends on the image acquisition process and does not change over time. Instead of storing all the elements along $\mathbf{D}$, we only store the element of $\mathcal{W}$ which in conjugation with the indices of element of $\mathbf{x}$ allows the reconstruction of the complete matrix $\mathbf{D}$. The computational complexity of the product $D\mathbf{x}^t$ is $\mathcal{O}(wn)$ with multiplication done only at the corresponding non-zero locations of $\mathbf{x}^t$.

The matrix $\mathbf{U}$ contains inter-sequence connections. Due to sparsity of the place recognition problem, this matrix is highly sparse $\text{nnz}(\mathbf{U}) \ll n$ with a computational complexity of $\mathcal{O}(\text{nnz}(\mathbf{U}))$ (nnz signifies the number of non-zero elements in the matrix). The overall complexity of the update step is that of multiplying the transition matrix with the state: $\mathcal{O}(wn)$ compared to $\mathcal{O}(n^2)$ for a general transition matrix.

F. SPRINT: Efficient inference on subgraphs

Given the sparse nature of the transition matrix, we model the inference problem as a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ over the place indices with edges $\mathcal{E}$ representing the connections in the transition matrix. Each vertex $\mathbf{v}_p \in \mathcal{V}$ stores information about the sequence that image belongs to (`seq_id`) as well as the location of the image within that sequence (`image_id`). Each vertex contains an edge to every other vertex from which the current vertex can be reached, that is, the probability $\mathbf{P}(\mathbf{x}_p|\mathbf{x}_q)$ is nonzero. Temporal adjacencies are represented via a fixed vector $\mathbf{w} \in \mathcal{R}^W$ such that W is the number of vertices in the past that connect to the current vertex and the value of each element w_i for $i = 0 \dots W-1$ represent the probability of transition from node $p-i$ to the current node p in the same sequence. This leads to storing only W elements to represent the whole $\mathbf{D}$ matrix for the whole graph $\mathcal{G}$. Each vertex also stores the probability the probability $\mathbf{x}_p^t$. The last image in the sequence is connected to the `sink` which only connects to itself with a probability 1 causing the propagated belief to be "trapped" in the sink node. Only a single `sink` node exists in $\mathcal{G}$.

As mentioned earlier, the image acquisition process together with the image retrieval mechanism, limits the region of the state space that needs to be updated. Formally, the graph $\mathcal{G}$ representing the images in the database and their connections to each other, can be thought of as the union of graphs $\mathcal{G}_1, \mathcal{G}_2, \dots \mathcal{G}_n$ representing n unconnected sequences, that is, $\mathcal{G}(\mathcal{V}, \mathcal{E}) = \cup_{i=1}^{n} \mathcal{G}_i(\mathcal{V}_i, \mathcal{E}_i)$. Once connections are found between two graphs, they are merged into a single graph with nonzero off diagonal entries. This partitioning allows efficient inference on a subset of the original state space. Each sub-graph can be thought of as a smaller HMM problem with a reduced and independent state space.

As pointed out earlier, place recognition is a sparse problem – the current image will have very few potential matches over all the images seen earlier. This means that at any given moment, non-zero belief is localized in a small number of sub-graphs. We utilize this by keeping a set of active components for which the belief is non-zero. We call this the `active set`.

Graphs are added to the active set when there is a non-zero observation probability associated with them. We assume that this probability comes from an image retrieval system that takes current image as input and returns a list of nearest neighbour matches and their distances from the query image. The database is not restricted in its search as this will bias our search results.

Graphs are removed from the active set when the probability mass contained in them compared to the other graphs in the set is smaller than a predefined threshold.

In the experiment section, we show that under the assumption of a reasonable image retrieval system, the gain in computational efficiency of the sub-graph HMM over a single graph is very significant. The theoretical computational complexity of this reduced representation is $\mathcal{O}(wn_r)$ where n_r is the fraction of states in at active set compared to the full state. Below we describe how our graph representation for sparse inference compares to the traditional HMM (III-A) at each step:

Initialization: Instead of uniform initialization, which will lead to all the sequences being in the active set, the process starts with the first observation and its nearest neighbours acquired via image retrieval. Only sequences for which there is a non-zero probability of observation are added to the active set.

Propagation: Probabilities are propagated for all the vertices in the active set. This has the same mechanism as the propagation described in Section III-A but now it is carried

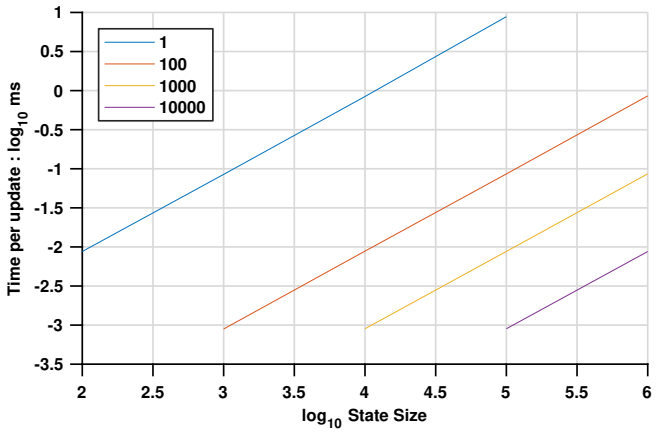

Fig. 4: (log-log) Comparison of execution times for full graph and subgraphs with different number of sequences. Each line corresponds to the number of subgraphs. (Figure best viewed in color).

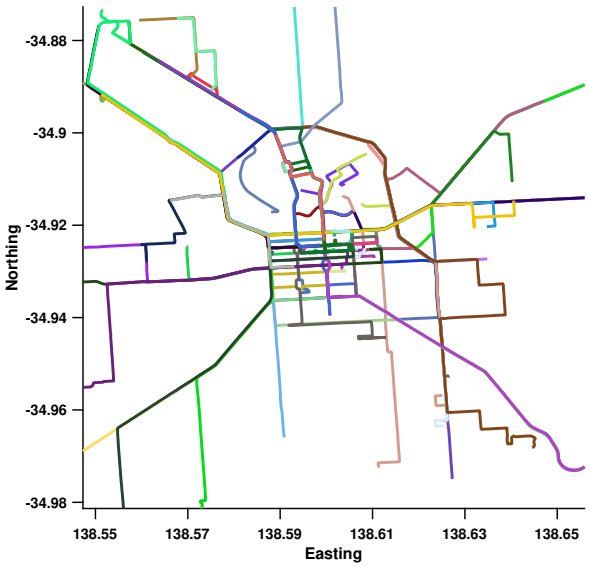

Fig. 5: GPS tracks of data captured around Adelaide CBD from Mapillary

our for each sub-graph in the active set.

Update Updates are done via image retrieval. The image retrieval mechanism uses the a whole image description of the current image and find its k-nearest neighbours. We demonstrate in the experiment section using k-nearest neighbours instead of the nearest neighbour is a valid choice, especially in the context of temporal reasoning.

After the update step, the members of the active set are reevaluated. Any sequence that has a probability mass of less than 10^{-4} contained in it is removed from the active set.[2]

IV. Experiments

We first show a set of experiment to illustrate the computational efficiency of the proposed work.

A. Computations

It is straight forward to show that in the subgraph case, if the number of sequences is greater than k (the number of nearest neighbour matches) there would at least be k subgraph active at any given moment. Under the assumption that the matches returned are not randomly selected from the whole database, it is highly likely that between two instance of time t and $t+1$, the k matches come from the same (at most) k sequences. Under these conditions, there will always be an active set of size $k < n$ graphs in the memory, which is more efficient for inference compare to carrying the complete graph.

We set up a simulation experiment with increasing number of nodes in the graph ranging from 100 to a million. In the first case, all the nodes are contained in the same graph, which represent the naive HMM algorithm. We then split the graph into multiple equally sized sequences [`100, 1000, 10000`]. We assume that the image retrieval method returns only the correct match for each query. This experiment is set up to understand the computational aspect of the problem and not accuracy of the inference. Each experiment is run 10 times and average timing results per update step are shown in Fig. 4 as a log-log plot. It should be noted that this is the best-case theoretical performance when only one sequence is in the active set at any given moment. Timing plots for experiments on real data are reported in Sec. IV-C.

[2]The corresponding code will be made available at `https://github.com/ylatif/sprint`

B. Why k-nearest neighbours as the backend?

In this experiment, we study the performance improvement gained over k- nearest neighbour image retrieval by performing HMM inference. For this experiment, we use the Nordland dataset that is collected from front facing cameras on a train in Norway over four different season: summer, winter, autumn and fall. Although there is no viewpoint variation, there is signification appearance variation due to seasonal changes, especially during winter when snow covers everything. For the experiment, we use the test split suggested in [15] consisting of 3450 images for each condition, spanning three different sections of the railroad.

Images are scaled to 64×64 and represented with a vectorized and normalized Histogram of Oriented Gradients (HoG) [7] descriptor. Since the dataset is small, image retrieval is done via exact nearest neighbour search to get the 10 nearest neighbours for each query image.

Following [15], we report the result of k-NN search in Table. I as the percentage of correctly matched image. It is interesting to see that k-NN search with 10 neighbours is able to retrieve most of the correct matches. Our task is to show that the proposed method can improve over the k-NN search by incorporating all the hypothesis and filtering them using temporal inference to get the correct match (highest probability) for each query image. The result obtained via our methods are report in Table. I. The last column corresponds to the results shown in [15] for the same data. Their method is based on deep learning specifically for the task of improving the nearest neighbour matching on the

Database	Query	1	5	10	SPRINT	[15]
summer	spring	74.26	90.03	93.80	**96.52**	93.36
	winter	33.13	52.90	62.17	78.58	**85.97**
	fall	96.14	98.38	**99.25**	98.29	97.77
winter	spring	23.86	40.12	47.97	65.51	**83.88**
	summer	21.68	37.83	45.39	62.03	**85.91**
	fall	21.16	37.42	45.16	61.88	**85.83**
spring	summer	64.55	82.78	87.80	95.30	**95.45**
	winter	38.87	59.30	66.70	81.42	**95.45**
	fall	71.68	86.46	90.99	**96.75**	95.62
fall	spring	82.12	94.72	**97.10**	96.29	94.00
	summer	96.06	98.67	**99.57**	98.12	85.48
	winter	36.58	58.93	67.77	80.03	**97.71**

TABLE I: Percentage of correctly matched images with k-NN, SPRINT and [15]

Nordland dataset.

It can be seen from Table. I that our method improves our k-NN search in most of the cases. What is interesting is that in some cases we can perform better than deeply learned method of [15]. However, it should be noted that their method is single-view, that is, for query image, a nearest neighbour returned in the learned latent space. The cases where our performance is worse involves the winter condition which is not very information for gradient computation (everything is white). This experiment shows that taking into account the sequential information with traditional hand crafted features can give a significant boost in performance even under severe appearance changes.

If no temporal reasoning is applied, then the best result from HoG is the 1-NN shown in Table.I. Even though higher values of k include a greater percentage of correct matches, from a place recognition point of view, they need to be fused to get a coherent hypothesis of the localization of the vehicle. This is what is achieved in the proposed method.

We additionally show a similar experiment on two query sequences for the Oxford RobotCar dataset [13]. We use the feature extraction method described in [8] and perform both k-NN and the proposed method. We compare against [8] with the difference being the composition of the transition matrix. The results are given in Table. II. The temporal inference combines all the hypothesis into consists place recognition decision. It also validates our assumption about forward motion encoded in the transition matrix.

Fig. 6: Sample matches found via SPRINT in the Adelaide Dataset: Top rows: Query, bottom row: Highest probability match

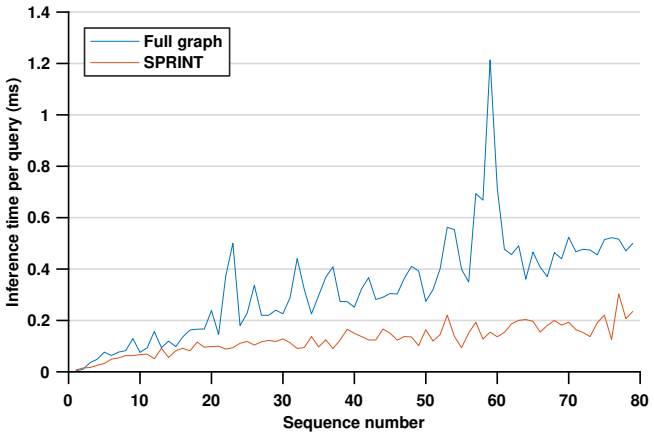

Fig. 7: Average inference time per query (ms) with increasing number of sequences in the database for the Mapillary dataset.

C. Large Scale localization in the wild

To demonstrate the performance of the proposed method of real world data, we source a dataset from Mapillary around the city of Adelaide, Australia and covers a region of about 10km^2. The collected data consists of 80 sequences with around 50 thousand images. Mapillary data is crowd sourced from vehicles and contains appearance and viewpoint changes. Some sample images from the same place are shown in Fig. 1 and GPS tracks for the collected area are shown in Fig. 5.

We run the proposed method on these 80 sequences to compare our performance against that of a single graph HMM. We use each sequence as a query sequence, which is then added to the database, therefore the size of the database grows with each query. We report the average inference time per query against the sequence number is reported in Fig. 7. As can be seen, the required time for a single inference step plateaus as the number of sequences in the database grows, suggesting that over larger databases, efficient inference can be achieved. Some sample matches are given in Fig. 6.

V. Conclusions and Future Work

This work proposes SPRINT, a scalable subgraph based HMM inference framework for large scale place recognition. We show that by exploiting the properties of the data collection method and the structure introduced on the transition matrix due to it, a more efficient inference mechanism can be achieved. This work however does not address the issue of growing number of images in the database, which is another computational bottleneck. In future, we will address storage need so that a practical large scale place recognition system can be formed.

Database	Query	1	5	10	SPRINT	[8]
26/06/14, 09:24	23/06/14, 15:36	88.42	92.26	94.71	**96.28**	92.44
	23/06/14, 15:41	87.28	91.36	93.18	**95.89**	92.19

TABLE II: Percentage of correctly matched images with k-NN, SPRINT and [8] on Oxford RobotCar

REFERENCES

[1] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5297–5307, 2016.

[2] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016.

[3] Z. Chen, A. Jacobson, N. Sünderhauf, B. Upcroft, L. Liu, C. Shen, I. Reid, and M. Milford. Deep learning features at scale for visual place recognition. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3223–3230. IEEE, 2017.

[4] W. Churchill and P. Newman. Experience-based navigation for long-term localisation. *The International Journal of Robotics Research*, 32(14):1645–1661, 2013.

[5] L. Clement and J. Kelly. How to train a cat: learning canonical appearance transformations for direct visual localization under illumination change. *IEEE Robotics and Automation Letters*, 3(3):2447–2454, 2018.

[6] M. Cummins and P. Newman. Fab-map: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 27(6): 647–665, 2008.

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. 2005.

[8] A.-D. Doan, Y. Latif, T.-J. Chin, Y. Liu, T.-T. Do, and I. Reid. Scalable place recognition under appearance change for autonomous driving. In *International Conference on Computer Vision (ICCV)*, 2019.

[9] D. Gálvez-López and J. D. Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, 2012.

[10] M. Labbe and F. Michaud. Appearance-based loop closure detection for online large-scale and long-term operation. *IEEE Transactions on Robotics*, 29(3):734–745, 2013.

[11] Y. Latif, R. Garg, M. Milford, and I. Reid. Addressing challenging place recognition tasks using generative adversarial networks. *arXiv preprint arXiv:1709.08810*, 2017.

[12] S. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford. Visual place recognition: A survey. *IEEE Transactions on Robotics*, 32(1):1–19, 2016.

[13] W. Maddern, G. Pascoe, C. Linegar, and P. Newman. 1 year, 1000 km: The oxford robotcar dataset. *The International Journal of Robotics Research*, 36(1):3–15, 2017.

[14] M. J. Milford and G. F. Wyeth. Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1643–1649. IEEE, 2012.

[15] D. Olid, J. M. Fcil, and J. Civera. Single-view place recognition under seasonal changes. In *PPNIV Workshop at IROS 2018*, 2018.

[16] H. Porav, W. Maddern, and P. Newman. Adversarial training for adverse conditions: Robust metric localisation using appearance transfer. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1011–1018. IEEE, 2018.

[17] T. Sattler, B. Leibe, and L. Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1744–1756, 2016.

[18] N. Sünderhauf, S. Shirazi, A. Jacobson, F. Dayoub, E. Pepperell, B. Upcroft, and M. Milford. Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free. *Proceedings of Robotics: Science and Systems XII*, 2015.

[19] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv:1703.10593*, 2017.