Visual Localization in the Presence of Appearance Changes Using the Partial Order Kernel

Maryam Abdollahyan*, Silvia Cascianelli†, Enrico Bellocchio†, Gabriele Costante†, Thomas A. Ciarfuglia†,
Francesco Bianconi†, Fabrizio Smeraldi* and Mario L. Fravolini†
*School of Electronic Engineering and Computer Science
Queen Mary University of London, Mile End Road, London E1 4NS, United Kingdom
Email: m.abdollahyan@qmul.ac.uk
†Department of Engineering
Università degli Studi di Perugia, Via G. Duranti 93, 06125 Perugia, Italy
Email: silvia.cascianelli@studenti.unipg.it

***Abstract*—Visual localization across seasons and under varying weather and lighting conditions is a challenging task in robotics. In this paper, we present a new sequence-based approach to visual localization using the Partial Order Kernel (POKer), a convolution kernel for string comparison, that is able to handle appearance changes and is robust to speed variations. We use multiple sequence alignment to construct directed acyclic graph representations of the database image sequences, where sequences of images of the same place acquired at different times are represented as alternative paths in a graph. We then use the POKer to compute the pairwise similarities between these graphs and the query image sequences obtained in a subsequent traversal of the environment, and match the corresponding locations. We evaluated our approach on a dataset which features extreme appearance variations due to seasonal changes. The results demonstrate the effectiveness of our approach, where it achieves higher precision and recall than two state-of-the-art baseline methods.**

***Index Terms*—visual localization, partial order graphs, kernel methods**

I. Introduction

Accurate and efficient localization is a critical problem for autonomous navigation systems; however, real-life scenarios present many challenges for visual localization. One such challenge arises from changes in the appearance of the environment. Appearance changes result from a number of factors, including illumination variations, different weather conditions and seasonal changes. A localization system must be able to deal with the mismatches between the images that result from appearance changes. This is particularly important in long-term navigation tasks, where the robot revisits an environment multiple times.

In this work, we propose a novel sequence-based visual localization approach that addresses the problem of appearance changes. Our approach consists of two phases: the first phase is constructing the graph representations of database image sequences; the second phase is comparing the graphs to newly acquired query image sequences. In the first phase, we use a multiple sequence alignment algorithm [1] to align the alternative image sequences from the database, i.e, sequences of images of the same place that were previously collected at different times and, hence, differ in appearance. The output of each alignment is a directed acyclic graph (DAG). Using such a representation not only allows us to exploit the temporal sequentiality of images, but also efficiently models the alternative image sequences in the form of alternative paths in the graph. In addition, it does not require the alternative paths in the graph to be of equal length and therefore, is robust to differences in the traversal speed. In the second phase, we consider the query image sequences collected during the localization phase and convert them to DAGs without alternative paths. We measure the pairwise similarities between these graphs and the graphs constructed during the exploration phase using the Partial Order Kernel (POKer) [2]. The POKer is a convolution kernel developed for the comparison of strings that contain alternative substrings. It provides a measure of similarity that is equal to a weighted sum of the scores of local alignments between all possible pairs of paths in the two graphs. In other words, it sums up the contributions of all images. Based on these similarities, the corresponding locations are matched. We extract the image descriptors using a convolutional neural network (ConvNet) [3], trained for place recognition tasks, and use them in order to compare the images and compute the alignment scores. It has been shown that descriptors obtained by ConvNets specifically trained for place recognition have invariant properties with respect to appearance changes and increase the robustness of place recognition algorithms [4].

We evaluated our approach on the standard Nordland dataset which was collected across four different seasons. The image sequences from three seasons constitute the training data, and the sequence from the remaining season was used for testing, in a cross-validation fashion. The experimental results show that our approach is an effective method for localization in the presence of appearance changes.

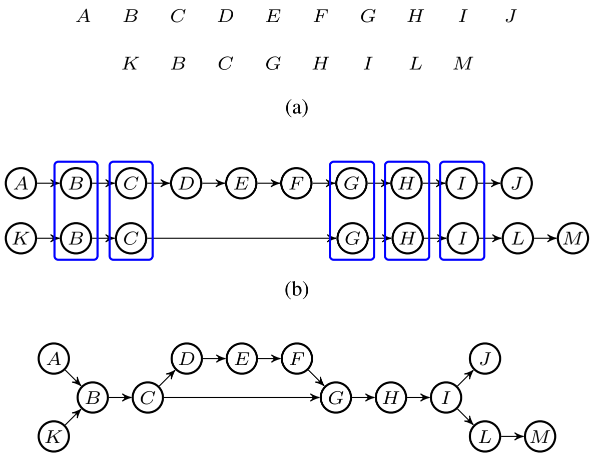
II. Related Work

Various approaches have been proposed to address the problem of appearance changes. In [5], a probability distribution was learnt to model the illumination variation in images. In [6], to reduce illumination variations, images were transformed into an illumination-invariant colour space. A number of approaches exploit image descriptors such as SIFT and SURF to handle appearance changes (e.g., [7]). More recently, the use of ConvNets to extract descriptors that are robust to appearance changes has gained a lot of attention. In [8], a neural network was trained to learn illumination-invariant descriptors that map the image patches into a new lower-dimensional space where non-matching images are easily separable. Incorporating features learnt using ConvNets has been shown to improve the performance of place recognition systems, as these features are more robust to appearance changes [9], [10], [11]. In this work, we employ the recently released ConvNet VGG-Places365 [3] to extract the descriptors. This ConvNet was trained on a dataset of images from diverse types of environments. ConvNets specifically trained for place recognition have been shown to outperform networks trained using generic data [12], [4].

A number of approaches, relying on the fact that some appearance changes such as seasonal changes are cyclic and therefore predictable, learn a transformation between the images [13], [14]. In [14], a superpixel vocabulary for each season and a dictionary to translate the words from one season to their matches in another were generated. This, however, requires the pairs of training images to be perfectly aligned. Our approach does not make any assumptions on the nature of appearance changes or pixel alignment of images.

Another category of approaches leverage the sequentiality of images to handle appearance changes. The state-of-the-art method SeqSLAM [15] considers sequences of images instead of single images. Given an image, the method finds the local best match within every short image sequence. Localization is then done by searching the image similarity matrix for sequences of local best matches. SeqSLAM assumes constant speed during the traversals. A modified version of SeqSLAM that is invariant to speed variations was introduced in [16]. In our approach, we represent the multiple sequences of images of the same place collected at different times and possibly at different speeds as a partial order graph. The image sequences can diverge from one another to form alternative paths in the graph. These paths may be of different lengths. This allows us to deal with mismatches between the images that are due to speed variations. In [17], a Hidden Markov Model was used to compute the most likely path through the image similarity matrix. While the method, similar to ours, uses dynamic programming (the Viterbi algorithm) to align the sequences, transitions between states are probabilistic. By contrast, our proposed graph representation specifies exactly which transitions are possible at each point. In [18], a modified version of the Smith-Waterman algorithm [19] was used to find matching subsequences within the image sequences in order to detect intersections between maps. The multiple sequence alignment algorithm used in our approach is also an extended version of the Smith-Waterman algorithm, however, it works with partial order graphs instead of standard sequences.

The methods described in [20] and [21] built a directed acyclic data association graph to model the matching between an image sequence and a database. The localization task then becomes a minimum-cost flow problem, i.e., computing a shortest path in this graph. Our approach differs in that the graphs are constructed from the database image sequences only and are later compared to query image sequences using the POKer.



(a)

(b)

(c)

Fig. 1: (a) String representation of two image sequences. Each letter denotes a single image. (b) DAG representation of two image sequences. (c) DAG representation of the MSA obtained by aligning the image sequences shown in (b) using the POA algorithm.

III. Localization Using the Partial Order Kernel

A. Building the Graph Representations of Image Sequences

We represent an image sequence (Fig. 1a) as a directed acyclic graph (DAG). In this graph, nodes represent images and there exist directed edges between nodes whose corresponding images are consecutive in the image sequence (Fig. 1b).

We use the Partial Order Alignment (POA) algorithm [1] to build the graph representations of database image sequences. The POA algorithm is an approach to multiple sequence alignment (MSA). It extends the classic sequence alignment algorithms of NeedlemanWunsch [22] and Smith-Waterman [19] to work with partial order graphs. Given a set of alternative image sequences from the database, i.e., image sequences which are of the same location but different due to appearance changes, we align each sequence to a growing MSA in an iterative manner. To find the optimal alignment, a similarity score is assigned to each aligned pair of images while gaps are penalized. We use the cosine similarity between the descriptors as scores. As for gaps, we use a linear gap penalty. Nodes that are aligned and identical (based on the high cosine similarity of their descriptors) are merged into a single node and redundant edges are removed so that there exists at most one edge between any given pair of nodes. The output is an MSA in the form of a DAG (Fig. 1c).

In the DAG representation of database image sequences, a node may have several predecessors and successors. Each path from a source to a sink node represents a complete traversal; merged nodes allow switching between alternative subsequences of images from different database sequences (e.g., J and LM in Fig. 1c). The paths may be of different lengths. Such a representation is advantageous as it does not assume that all regions within the sequences are homologous over their entire length, an assumption that indeed does not hold here due to appearance changes and speed variations. The POA algorithm runs in polynomial time, while obtaining the pairwise alignments between all possible pairs of image sequences would require exponential time.

The query image sequences are represented simply as DAGs without any branches (Fig. 1b).

B. Comparing the DAGs Using the Partial Order Kernel

We use the Partial Order Kernel (POKer) [2] to compute the similarities between the graph representations of database and query image sequences. The POKer is a convolution kernel [23] developed for the comparison of strings containing alternative substrings that, as we have shown, can be efficiently represented by DAGs. The POKer takes as input a pair of DAGs G_x and G_y. Let $\Pi_n(G_x)$ and $\Pi_n(G_y)$ be the sets of paths of length n in G_x and G_y, respectively. The POKer is then defined as

$$K(G_x, G_y) = \sum_{n\geq 0} K_n(G_x, G_y) \tag{1}$$

$$= \sum_{n\geq 0} \sum_{\substack{\pi_x\in\Pi_n(G_x)\\ \pi_y\in\Pi_n(G_y)}} \exp(\beta S(\pi_x, \pi_y)) \tag{2}$$

where $S(\pi_x, \pi_y)$ is the score of the local alignment of the n nodes along a path π_x in G_x with the same number of nodes along a path π_y in G_y, and $\beta \geq 0$ is a parameter. For the alignments, we use the same scores and gap penalty as in Section III-A. Valid values for β are those for which the kernel remains positive semi-definite.

The POKer produces a measure of similarity that is equal to an exponentially weighted sum of the scores of all the possible local alignments between any path in G_x and any path in G_y, that is, any choice of image subsequences. This accounts for the contributions of all the alignments of subsequences from the query image sequence against subsequences from the paths in the database graph. The importance of the contribution of non-optimal alignments to the kernel value is controlled by parameter β. For $\beta \rightarrow \infty$, only the best alignments are taken into account.

The POKer is implemented using dynamic programming. Despite considering the contributions of a number of paths that is exponential in the number of branching points in the database graph, the POKer has a time complexity that is linear in the number of nodes in the strong product of the two DAGs.

IV. Experiments

A. Dataset and Parameters

We chose the Nordland dataset[1] for the experimental evaluation of our approach. The dataset consists of video footage of a 728km-long train journey between two cities in Norway, recorded from the perspective of the train driver. The journey was recorded once in every season. We subsampled each video at 0.5fps, yielding a total of four image sequences. We refer to these image sequences as the Spring, Summer, Autumn and Winter sequences. Note that all sequences are of equal length and that images with the same numerical index are from the same place (which serves as the ground truth). The dataset features severe appearance changes due to different weather conditions and seasonal changes. The train occasionally goes through tunnels and stops at stations. As customary for this dataset [4], [9], we removed all the images captured inside the tunnels and at the stops. The descriptors were extracted from the fifth layer of the VGG-Places365 ConvNet [3]. We applied Locality-Sensitive Hashing (LSH) [24] to reduce the dimensionality of the descriptors from 100352 to 4096.

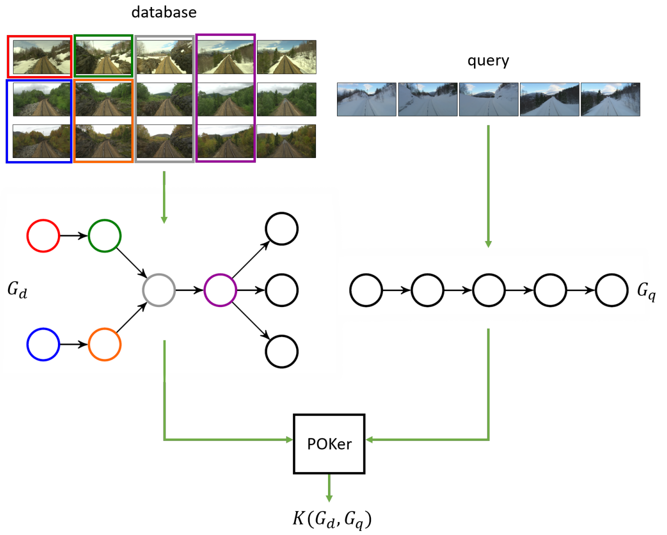
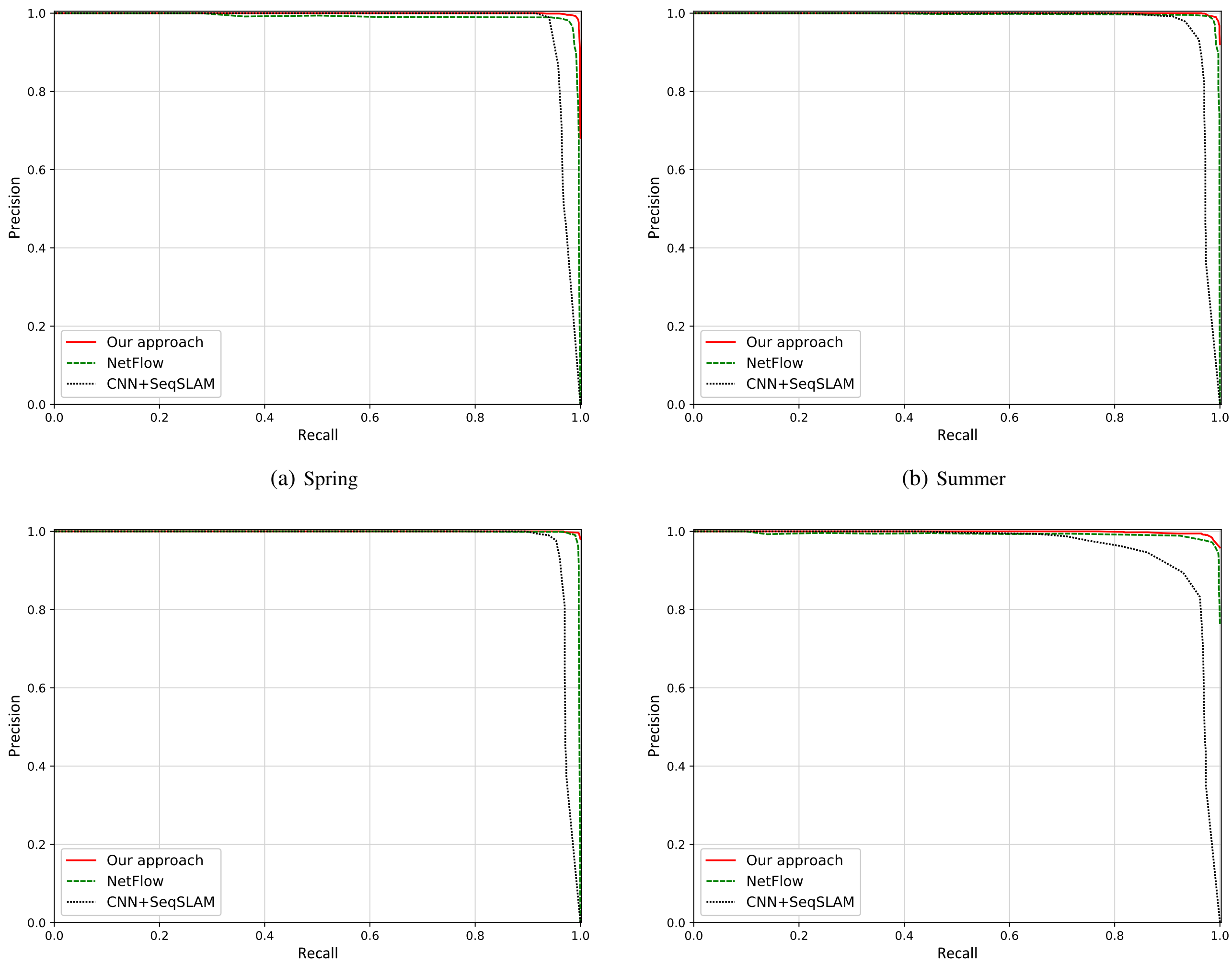


Fig. 2: Overview of our method applied to the Nordland dataset. On the left: three database image sequences of a place in spring, summer and autumn, respectively. A DAG representation of these alternative sequences is built using the POA algorithm. On the right: a query image sequence of the same place in winter, represented as a DAG. Query and database graphs are compared using the POKer which produces a measure of similarity between the graphs.

We performed four sets of experiments, each time using the image sequence belonging to a different season for generating the query image sequences. The data for each set of experiments was generated as follows: during the exploration phase, we consider three of the image sequences in the dataset, i.e., three seasons. We cut each sequence into subsequences of length 15. As a result, for each location, there exist three alternative image sequences in the database (one per season). We generate triplet image sequences by selecting the three image sequences of the same place in different seasons, according to the ground truth. For each triplet, we align the image sequences in that triplet and build its graph representation, as explained in Section III-A (e.g., left column in Fig. 2).

During the localization phase, we consider the remaining

[1] https://nrkbeta.no/2013/01/15/nordlandsbanen-minute-by-minute-season-by-season/



(a) Spring

(b) Summer

(c) Autumn

(d) Winter

Fig. 3: Precision-recall curves obtained for the four sets of experiments using the (a) Spring, (b) Summer, (c) Autumn and (d) Winter sequences as the query image sequence, respectively.

image sequence in the dataset, i.e., the fourth season. We generate the query image sequences by cutting this sequence into subsequences of length 15. We convert each of these to a DAG without alternative paths, as explained in Section III-A (e.g., right column in Fig. 2). We then compare them to the database triplets using the POKer.

For the alignment parameters, we used the Hamming distance between the descriptors as scores, since after applying LSH, the cosine similarity between the original high-dimensional data is approximated by the Hamming distance between the low-dimensional data. The gap penalty was set to -1. For the POKer, we used $\beta = 1$.

B. Baseline Methods

We used two state-of-the-art baselines to evaluate the performance of our method: the algorithm presented in [20] using network flows, and SeqSLAM combined with ConvNet features. We refer to these methods as NetFlow and CNN+SeqSLAM, respectively. For both baselines, we used the same features as those used in our method, i.e., descriptors extracted by the pre-trained VGG-Places365 ConvNet.

Note that these methods match an image to another, not an image sequence to multiple image sequences (here, a triplet). Therefore, to obtain a measure of similarity between a query image sequence and a triplet of database image sequences we proceeded as follows: we compare the query image sequence to each database image sequence in the triplet separately. The results are three matrices, where each matrix stores the similarity scores between all pairs of images from the query sequence and one of the database sequences. We then fuse the three matrices by choosing the maximum score for each pair of images as their final similarity score (we considered both average and maximum of scores and chose the maximum as it yielded the better performance). The similarity between the query sequence and the database triplet is the average of entries in this matrix. Parameters for both baselines were set to those that performed best for the most challenging image sequences in the dataset (Summer vs. Winter).

TABLE I: Comparison of the average recall values obtained by our method and the baselines.

Precision (%)	**Recall (%)**		
	Our approach	***NetFlow***	***CNN+SeqSLAM***
100	90.7	37.0	75.5
95	99.9	99.2	92.6
90	99.9	99.6	95.1

C. Results

The precision-recall curves for the four sets of experiments are shown in Fig. 3. As can be seen, in each case, our method either matches or outperforms the baselines in all parts of the curve. Table I reports the recall values obtained by our method and the baselines for three precision values of practical interest, averaged over the four experiments. Our approach achieves a high level of recall ($> 90\%$) with 100% precision, and by sacrificing 5% of precision, almost 100% recall is achieved; in comparison, both NetFlow and CNN+SeqSLAM show (at times significantly) lower performance.

V. Conclusion and Future Work

In this work, we addressed the problem of appearance changes in visual localization by using a directed acyclic graph representation together with the Partial Order Kernel (POKer). The graph representation we introduced efficiently models appearance variations and correlations among consecutive images in the form of alternative paths in a graph.

We showed how the POKer effectively computes the similarities between these graphs to match the corresponding image sequences. Experiments on the standard Nordland dataset suggest that our approach is robust to severe appearance changes and significantly outperforms two state-of-the-art methods in such a setting. The results encourage us to investigate the application of our approach to further challenging scenarios, including localization in indoor environments and in the presence of extreme speed variations.

Acknowledgment

We gratefully acknowledge the support of the NVIDIA Corporation for the donation of a Titan X GPU used in this work to the Department of Engineering, Università degli Studi di Perugia.

References

[1] C. Lee, C. Grasso, and M. F. Sharlow, “Multiple sequence alignment using partial order graphs,” *Bioinformatics*, vol. 18, no. 3, pp. 452–464, 2002.

[2] M. Abdollahyan and F. Smeraldi, “POKer: a partial order kernel for comparing strings with alternative substrings,” in *25th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, 2017, pp. 263–268.

[3] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, “Places: a 10 million image database for scene recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[4] N. Sünderhauf, S. Shirazi, F. Dayoub, B. Upcroft, and M. Milford, “On the performance of convnet features for place recognition,” in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 4297–4304.

[5] A. Ranganathan, S. Matsumoto, and D. Ilstrup, “Towards illumination invariance for visual localization,” in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 3791–3798.

[6] W. Maddern, A. Stewart, C. McManus, B. Upcroft, W. Churchill, and P. Newman, “Illumination invariant imaging: applications in robust vision-based localisation, mapping and classification for autonomous vehicles,” in *Proceedings of the Visual Place Recognition in Changing Environments Workshop, IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China*, vol. 2, 2014, p. 3.

[7] C. Valgren and A. J. Lilienthal, “SIFT, SURF & seasons: appearance-based long-term localization in outdoor environments,” *Robotics and Autonomous Systems*, vol. 58, no. 2, pp. 149–156, 2010.

[8] N. Carlevaris-Bianco and R. M. Eustice, “Learning visual feature descriptors for dynamic lighting conditions,” in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE, 2014, pp. 2769–2776.

[9] R. Gomez-Ojeda, M. Lopez-Antequera, N. Petkov, and J. Gonzalez-Jimenez, “Training a convolutional neural network for appearance-invariant place recognition,” *arXiv preprint arXiv:1505.07428*, 2015.

[10] 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.

[11] S. Cascianelli, G. Costante, E. Bellocchio, P. Valigi, M. L. Fravolini, and T. A. Ciarfuglia, “Robust visual semi-semantic loop closure detection by a covisibility graph and CNN features,” *Robotics and Autonomous Systems*, vol. 92, pp. 53–65, 2017.

[12] Z. Chen, O. Lam, A. Jacobson, and M. Milford, “Convolutional neural network-based place recognition,” *arXiv preprint arXiv:1411.1509*, 2014.

[13] S. M. Lowry, M. J. Milford, and G. F. Wyeth, “Transforming morning to afternoon using linear regression techniques,” in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 3950–3955.

[14] P. Neubert, N. Sünderhauf, and P. Protzel, “Superpixel-based appearance change prediction for long-term navigation across seasons,” *Robotics and Autonomous Systems*, vol. 69, pp. 15–27, 2015.

[15] 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*. IEEE, 2012, pp. 1643–1649.

[16] E. Pepperell, P. I. Corke, and M. J. Milford, “All-environment visual place recognition with SMART,” in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1612–1618.

[17] P. Hansen and B. Browning, “Visual place recognition using HMM sequence matching,” in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE, 2014, pp. 4549–4555.

[18] K. Ho and P. Newman, “Multiple map intersection detection using visual appearance,” in *International Conference on Computational Intelligence, Robotics and Autonomous Systems*, 2005.

[19] T. F. Smith and M. Waterman, “Identification of common molecular subsequences,” *Journal of molecular biology*, vol. 147, pp. 195–197, 1981.

[20] T. Naseer, L. Spinello, W. Burgard, and C. Stachniss, “Robust visual robot localization across seasons using network flows,” in *AAAI Conference on Artificial Intelligence*, 2014, pp. 2564–2570.

[21] O. Vysotska and C. Stachniss, “Lazy data association for image sequences matching under substantial appearance changes,” *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 213–220, 2016.

[22] S. B. Needleman and C. D. Wunsch, “A general method applicable to the search for similarities in the amino acid sequence of two proteins,” *Journal of molecular biology*, vol. 48, no. 3, pp. 443–453, 1970.

[23] D. Haussler, “Convolution kernels on discrete structures,” Technical report, Department of Computer Science, University of California at Santa Cruz, Tech. Rep., 1999.

[24] A. Gionis, P. Indyk, R. Motwani *et al.*, “Similarity search in high dimensions via hashing,” in *Vldb*, vol. 99, no. 6, 1999, pp. 518–529.