Lazy Data Association For Image Sequences Matching Under Substantial Appearance Changes

Olga Vysotska and Cyrill Stachniss

***Abstract*—Localization is an essential capability for mobile robots and the ability to localize in changing environments is key to robust outdoor navigation. Robots operating over extended periods of time should be able to handle substantial appearance changes such as those occurring over seasons or under different weather conditions. In this paper, we investigate the problem of efficiently coping with seasonal appearance changes in online localization. We propose a lazy data association approach for matching streams of incoming images to a reference image sequence in an online fashion. We present a search heuristic to quickly find matches between the current image sequence and a database using a data association graph. Our experiments conducted under substantial seasonal changes suggest that our approach can efficiently match image sequences while requiring a comparably small number of image to image comparisons.**

***Index Terms*—Localization, Place Recognition, Visual-Based Navigation**

I. Introduction

LOCALIZATION is essential for goal-directed navigation. The ability to identify that a robot is at a previously visited place is an important element of localization. Handling large appearance changes such as those depicted in Figure 1 is a challenging problem and cannot be neglected in the context of persistent autonomous navigation. Localization through image matching or through appearance-based approaches for handling seasonal changes has been addressed by different researchers in the past, for example [6], [7], [11], [22], [27].

Several visual place recognition systems exploit features such as SURF [2] or SIFT [16]. These features encode local gradients computed at keypoints. Matching approaches relying on such features can deal with viewpoint changes and they show a great performance if the appearance of the environment does not change dramatically. They, however, perform rather poor under extreme perceptual changes. Recently, a series of robust visual localization approaches has been proposed including FAB-MAP2 [7], SeqSLAM [19], SP-ACP [23], as well as [22], [30]. Some of these methods have been shown to robustly recognize previously seen locations even under a wide spectrum of visual changes including dynamic objects, different illumination, and varying weather conditions.

Manuscript received: August, 28, 2015; Revised October, 26, 2015; Accepted December, 7, 2015.

This paper was recommended for publication by Editor Jingshan Li upon evaluation of the Associate Editor and Reviewers' comments.

This work has partly been supported by the European Commission under the grant numbers FP7-610603-EUROPA2.

The authors are with Institute for Geodesy and Geoinformation, University of Bonn, Germany olga.vysotska@uni-bonn.de

Digital Object Identifier (DOI): see top of this page.

Fig. 1: Examples of typical image pairs taken at the same places within multiple datasets. The image pairs are successively found by our approach. First row: Freiburg dataset; second row: Alderley dataset; third row: Nordland dataset. Fourth row: day/night scene from the VPRiCE'15 Challenge dataset.

It is essential for robot navigation that the location information is available in a timely manner, i.e., that the algorithm provides an online solution to the localization problem. Several existing techniques operate either in an online fashion but have issues to deal with strong seasonal changes or they can handle such changes but may not work online. In our work, we address the problem of online image sequence matching tailored to situations with large appearance changes.

The contribution of this paper is an online approach to image sequence matching that uses a data association heuristic while searching for matching image sequences in a data association graph. The heuristic estimates the expected cost of matching images considering the best matches found so far. Our directed acyclic data association graph is built incremen-

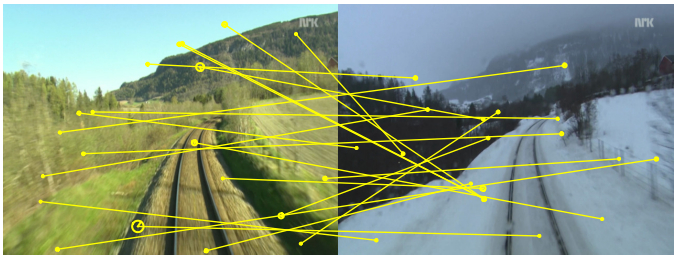

Fig. 2: Illustration of the fact that SIFT features do not perform well under strong seasonal variations. As a result of the seasonal changes, most SIFT matches illustrated by the lines between the images are outliers as the lines do not connect corresponding points.

tally whenever new sensor data arrives and its leaves model the data association hypotheses currently under consideration. For matching images, we rely on learned features from deep convolutional neural networks and the image similarity defines the cost in the data association graph. We furthermore show how additional location information can be exploited in the process and evaluate our method on several real world datasets.

II. Related Work

Pose estimation is a frequently studied problem in robotics and different approaches have been proposed for visual localization [1], [3], [7], [8], [9]. The ability to localize is an essential prerequisite for most autonomous navigation systems. Dealing with substantial variations in the visual input has been recognized as an obstacle for persistent autonomous navigation and this problem has been investigated by different researchers [7], [11], [15].

The majority of visual place recognition systems exploit features such as SURF [2] or SIFT [16] and several approaches apply bag-of-words techniques, i.e. they perform matching based on an appearance statistics of such features. To improve the robustness of appearance-based place recognition, Stumm *et al.* [27] consider the constellations of visual words and keeping track of their covisibility. Often, approaches using SIFT or SURF show a great performance if the appearance of the environment does not change radically. As also reported in our previous work [22], the matching performance of SIFT or SURF degrades under strong perceptual changes. An example, which illustrates this fact, is shown in Figure 2. The two images are taken from the same place and similar view points but during different seasons. As can be seen from the matches illustrated through the yellow lines, most of the correspondences are outliers. This examples illustrates our experience, that SIFT and SURF features are not well-suited for image matching under strong seasonal variations. Across season matching using SIFT and SURF has been investigated by Valgren and Lilienthal [29] by combining features and geometric constraints, which can improve the matching. In previous approaches, we proposed the use of tessellated HOG features [22], [30]. In contrast to that, in this paper we apply deeply learned features proposed by Sermanet *et al.* [25] and suggested for place recognition by Chen *et al.* [5]. We use them as an alternative to tessellated HOG features as they provide a better matching performance in our settings. Another recent work [28] suggests a technique for place recognition, where features stem from convolutional neural networks. The authors extract features from the landmark proposals, construct the similarity matrix by comparing the landmark features using the cosine distance and also take into account the size of the bounding boxes for the matched landmarks. The recognition task is then performed by selecting individual matches based on the highest similarity score.

In terms of aligning image sequences, several approaches have been proposed. For example, Matsumoto *et al.* [17] use the image sequences and directional relations between images to perform visual navigation in a corridor environment. SeqSLAM [19], which also aims at matching image sequences under strong seasonal changes, computes an image-by-image matching matrix that stores dissimilarity scores between all images in a query and database sequence. SeqSLAM computes a straight-line path through the matching matrix and select the path with the smallest sum of dissimilarity scores across image pairs to determine the matching route. Milford *et al.* [18] also present a comprehensive study about the SeqSLAM performance on low resolution images. Related to that, Naseer *et al.* [22] focus on offline sequence matching using a network flow approach. If odometry is available, this approach can also be combined with a least squares SLAM system to build metric maps [21].

The approach by Neubert *et al.* [23] aims at predicting the change in appearance on top of a vocabulary. For the vocabulary, the method predicts the change of the visual word over different seasons but the learning phase requires an accurate image alignment over seasons. A recent approach by Johns and Young [14] builds a statistic on the co-occurrence of features under different conditions. It relies on the ability to detect stable and discriminative features over different seasons. Finding such discriminative and stable features under strong changes is however a challenge. To avoid addressing the problems of finding features that are robust under extreme perceptual differences, Churchill and Newman [6] store different appearances for each place. These so-called experiences enable a robot to localize in previously learned experiences and associate a new data to places. A recent extension of that paradigm targets vast-scale localization by exploiting a prioritized recollection of relevant experiences so that the number of matches can be reduced [15].

Biber and Duckett [4] address the problem of dealing with changes in the environments by representing maps at different time scales. Each map is maintained and updated using the sensor data modeling short-term and long-term memories. This enables handling variations. In contrast to that, Stachniss and Burgard [26] model different instances of typical world states using clustering.

There are furthermore approaches combining laser and visual information for large-scale localization at city scale. The approach of Pascoe *et al.* [24] exploits laser data and vision information during the mapping phase with a survey vehicle but can then localize a car only using a camera.

To achieve a visual localization in a long term autonomy setup, Furgale and Barfoot [10] propose a teach and repeat system that is based on a stereo setup. The approach exploits

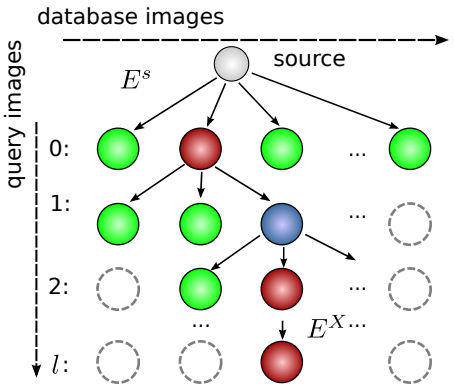

Fig. 3: Schematic illustration of the graph structure for the search. To perform an online localization our algorithm compares only image pairs that correspond to the green nodes and expands the green area on the fly. Red nodes correspond to matches of similar images along the path through the data association graph and blue indicates matches along the path with a low similarity.

local submaps and enables a robot to navigate on long trajectories but this method does not address large perceptual changes with respect to the taught path.

In this paper, we introduce a lazy data association scheme inspired by the work of Hähnel *et al.* [13] to come up with an online solution that can be executed on a robot while navigating. A key goal is to reduce the number of image-to-image comparisons with respect to existing methods such as [19], [22], [30]. In contrast to the work by Hähnel *et al.*, we use a heuristic that considers the cost of the path taken so far in order to speed-up the search. Our work is an extension of a short paper [31] presented at the ICRA 2015 Workshop on Visual Place Recognition in Changing Environments.

III. Lazy Matching for Online Operation

The main goal of this paper is to propose an *online* algorithm that exploits the sequence information to perform a global localization under strong appearance changes. We perform localization in the sense that we match a sequence $\mathcal{Q}$ of the images that we receive from sensor with a reference or database sequence of images called $\mathcal{D}$. For every incoming image, we want to know if there is a corresponding match in the database and if so, to which image of the database it corresponds to. The database itself is organized as regular files. In memory, we only keep an index to the images and feature descriptions and load individual images on demand from disk.

A. Data Association Graph

We use a directed acyclic graph $G = (X, E)$ as our main data structure for modeling the data association problem. We model the sequential image matching task as finding a shortest path in the data association graph G. We build up the data association graph on the fly and therefore only need to compare images if our search algorithm expands the corresponding node in G.

The key idea of this data association graph can be explained as follows. A node in the graph represents a potential match between two images. We aim at finding the best combination of matching images by searching a path through this graph, see Figure 3 for an illustration, where the cost of visiting a node depends on the similarity of both images.

In more detail, we propose the following graph structure.

a) Nodes: We have two types of nodes in X: the root or start node x^s and matching nodes. A matching node x_{ij} models a match of the image $i \in \mathcal{Q}$ with the image $j \in \mathcal{D}$. The more similar two images are, the more likely is the fact that they represent the same place. The similarity of an image pair is defined as $z_{ij} \in [0, 1]$, where $z_{ij} = 1$ means that both images appear identical. The similarity z_{ij} is computed by comparing the images $i \in \mathcal{Q}$ and $j \in \mathcal{D}$ only through their global image descriptor using the cosine distance.

As we are building the graph online, new nodes x_{ij} need to be created as soon as a new image i is recorded. Adding a node x_{ij} to the graph, however, comes at a *computational cost* as we need to compare images to compute z_{ij}. Thus, for building up the graph, we seek to avoid instantiating unnecessary nodes x_{ij}, i.e. nodes, which are not part of the matching sequence.

b) Edges: Similar to the nodes, we use two types of edges $E = \{E^s, E^X\}$ according to the types of nodes the edges connect. The set of edges E^s connects the source node x^s with the matching nodes x_{0j} corresponding to matching the first query image with any database image $j \in \mathcal{D}$, i.e.,

$$E^s = \{(x^s, x_{0j})\}_{j \in \mathcal{D}}. \tag{1}$$

The second set of edges E^X connects the matching nodes. In our approach, we define the set E^X of edges in a slightly different way to [30], [22] as

$$E^X = \{(x_{ij}, x_{(i+1)k})\}_{k=j-K,\ldots,j+K}, \tag{2}$$

where K is a fanout parameter that influences the nodes that are connected between the query images i and $i+1$. The fanout basically models that the robot can move at different speeds through the environment or that the cameras can operate at different framerates. The larger K is, the larger the branching factor of the graph and, thus, the value of K impacts the speed of the search described in the following sections. In our current implementation, we use a constant fanout parameter of $K = 5$. In the remainder of this paper, the nodes $x_{(i+1)k}$ are referred to as the children $\mathrm{ch}(x_{ij})$ of the node x_{ij}.

c) Weights/Costs: Each edge $e \in E$ has a weight or cost associated to it. This weight $w(e)$ is related to the similarity score z_{ij}. The weight of an edge $e = (x_{ij}, x_{i'j'}) \in E^X$ is inverse proportional to the similarity of the node to which this edges leads to, i.e.

$$w(e) = \frac{1}{z_{i'j'}}, \tag{3}$$

where $z_{i'j'}$ is a similarity score computed when comparing image i' and j' using the cosine distance.

B. Computing Image Similarity based on Features from Deep Convolutional Neural Networks

The computation of the similarity of two images has to be done often and thus we are in general interested in a fast computation. Nevertheless, the quality of the similarity

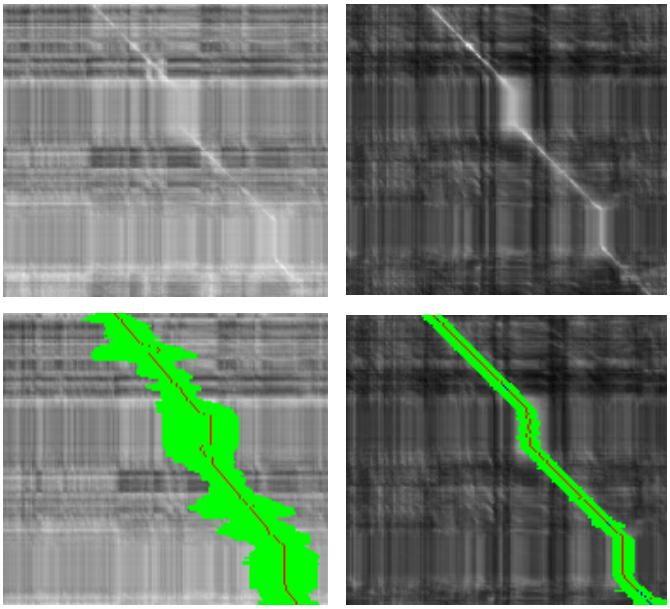

Fig. 4: Similarity matrix computed using tessellated HOGs as in [22] (left) and OverFeat features (right). As can be seen in the first row, the OverFeat features yield more distinct similarity values. This leads to a smaller number of nodes that are instantiated in the data association graph (green), as depicted in the second row.

function is of high importance. The more distinct the value of z_{ij} are for images taken from the same places vs. from different places, the easier the data association problem. As a result, the more distinct such values are, the better the performance of our graph search algorithm as less nodes will need to be expanded.

In our previous works [22], [30], we computed the tessellated HOG descriptor for every image and then compared them using the cosine distance. The obtained cost difference between the best match and the worse match was sufficient to find good solution with an exhaustive search. In the context of our lazy data association approach with a search heuristic, we experience problems to find matching sequences reliably without expanding the majority of the nodes in the graph. Therefore, we changed the image descriptors in this work to the pre-trained deep convolutional neural network OverFeat as proposed by Sermanet *et al.* [25] due to its better matching performance. OverFeat is built using a network trained on the ImageNet dataset consisting of 1.2 million images. We used the 10^{th} layer as a global image feature as suggested by Chen *et al.* [5]. Using OverFeat features instead of HOG directly improves the performance of our algorithm and supports the lazy approach. To give an intuition about the matching similarity, Figure 4 depicts the similarity of comparing all possible combinations of images from database $\mathcal{D}$ and query $\mathcal{Q}$ computed with the tessellated HOG descriptor (left) and OverFeat (right). Brighter values indicate a higher similarity. As can be seen from the images, the OverFeat features lead to more distinct values (higher contrast).

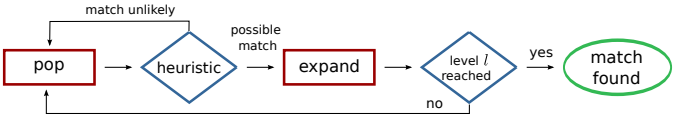

Fig. 5: Illustration of searching for a match for an input image.

C. Image Sequence Matching through Graph Search

The sequence of matching images between $\mathcal{Q}$ and $\mathcal{D}$ can be computed by a path search from the start node x^s to any node x_{l*}, with $*$ referring to any index in $\mathcal{D}$ and l being the most recent image in $\mathcal{Q}$. Every node that is a part of the shortest path corresponds to a selected data association, i.e. a match.

The computationally most demanding process for building and searching in such a data association graph is instantiating all nodes as a large number of possible matches has to be computed. For online localization, we are interested in keeping the computational efforts small and in avoiding the creation of nodes that do not represent matches. To address this issue, we propose an approach that limits the number of image comparisons and results in an efficient algorithm.

Our work is motivated by the ideas of lazy data associations in the context of SLAM proposed by Hähnel *et al.* [13] for constructing pose graphs. Hähnel *et al.* build up a data association tree and expand in each round the node with the highest log likelihood of representing a match between laser range scans. This is similar to a greedy search in a data association tree.

In our case, we go a step further and seek to performing an *informed* search through the graph, while the graph is built on the fly. One popular way to perform an informed search is the A* algorithm using a heuristic, which allows for estimating the cost from the currently expanded node to the goal node. For our matching problem, that means we need to predict how well the images that we will receive in the future will match our database images—this is in general a difficult task. Furthermore, A* requires that the heuristic is a predefined function and does not change during the search. In contrast to that, we take a different approach and try to predict the matching cost based on the images that we have received so far. This means, our heuristic is updated *during the search*, which prevents the application of A*. Our search procedure takes into account the estimated matching cost and works as follows.

Similar to A*, we use an open-list F of nodes that are still under consideration. This open-list is realized through a priority queue. In contrast to A*, the key of our priority queue for a node x_{ij} is the cost $g(x_{ij})$ of reaching x_{ij} from the source x^s. Our search and simultaneous graph construction starts with creating the source node x^s and connecting it to the matching nodes according to Eq. (1). This step requires to instantiate $|\mathcal{D}|$ nodes if no further information about the first possible match is provided.

For a new incoming image referred to as l, we use the following procedure to update the graph as well as the matching sequence (see Figure 5 for a brief illustration): Whenever a new image is obtained, we pop a node from F. We use

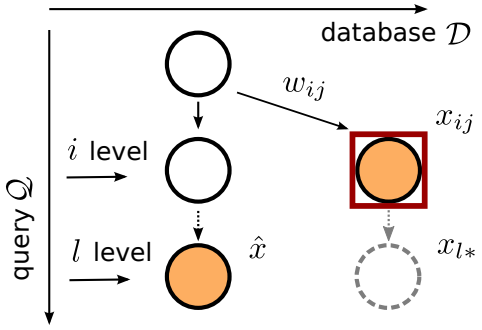

Fig. 6: Illustration for the graph expanding procedure. Orange nodes are nodes in the F. The red square indicates that the element x_{ij} will be the next one in F. The dashed gray line represent nodes and edges not computed yet.

our heuristic, which will be described in the remainder of this section, to estimate if the popped node x_{ij} is worth expanding or if it is unlikely to be part of the matching sequences. If the node is unlikely to be part of the matching sequences, we continue with the next node in F. Otherwise, we expand the node x_{ij} by computing the matching costs for its children $\mathrm{ch}(x_{ij})$ and connecting the node x_{ij} with $\mathrm{ch}(x_{ij})$ using the edges define in E^X, see Eq. (2). If a node in $\mathrm{ch}(x_{ij})$ lies on the l^{th} level of the graph, it represents the so far best match for the most recent image and the search terminates for this input image. Otherwise, we proceed expanding nodes from F.

The above described method relies on a heuristic to estimate the sum of matching costs for reaching the l^{th} level (the most recent query image). The key problem here is that defining an *effective* and *admissible* heuristic is hard due to the small amount of background information that can be exploited to predict future image similarities. Therefore, we take an alternative approach to come up with a heuristic that provides a good estimate of the cost but is not guaranteed to be admissible in the sense of A^* search. We take a statistical approach and approximate an *expected* lower bound for the *average* cost of the unexpanded and thus unknown nodes. We do so by using the average cost of the *best* path found so far as a prediction of the cost of individual matches along a new path. Furthermore, we exploit the fact that we know the number of images *obtained so far* and we know that the shortest path will have $l+1$ nodes (start node plus one matching node for each image). This allows us to formulate the expected cost $f(x_{ij})$ for a node x_{ij} to a node on the l^{th} level, i.e. x_{l*}, as the computed cost from x^s to x_{ij} expressed through $g(x_{ij})$ plus the estimate cost as:

$$f(x_{ij}) = g(x_{ij}) + \underbrace{\alpha(l-i)\mu_{\text{cost}}(\hat{x})}_{\text{heuristic}}, \tag{4}$$

where $\alpha \in [0,1]$ is a factor to trade off the quality of the solution and the number of nodes that needs to be expanded. For $\alpha = 0$, we obtain a greedy search behavior and for $\alpha = 1$, we may not expand enough nodes to find a good solution. The term $(l-i)$ is the number of images that have to be matched to end the sequence and $\mu_{\text{cost}}(\hat{x})$ is the average cost of the best path found so far, see also Figure 6.

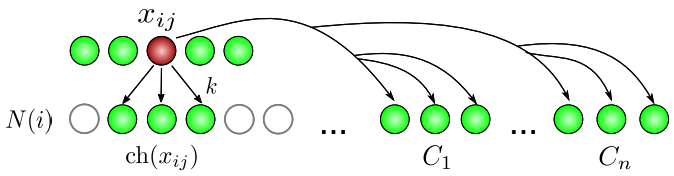

Fig. 7: Keeping connectivity though additional edges when using location priors. Green: nodes that are expended and added to the graph; gray: potential neighboring nodes according to the prior, but not encountered in the graph search and thus not instantiated.

In sum, the data association graph constructed in the proposed way using OverFeat features allows us to design a useful, but not guaranteed to be admissible heuristic for the search for data associations. This in turn means that we are not guaranteed to find the optimal solution but enables a fast search for image matching across seasons that can be executed online.

D. Exploiting Location Priors for Online Matching

In case a rough location prior, for example from a noisy GPS, is available, we can further improve the matching procedure and can also better deal with loops in the database as well as query sequences, i.e. place revisits.

The graph construction described in Section III-A can naturally be extended to account for location prior information. The overall procedure of constructing the graph stays the same but an additional location prior allows us to identify for every query image i the set of possible neighboring images $N(i)$ as

$$N(i) = \{j \mid j \in \mathcal{D} \wedge \mathrm{dist}(i,j) < d_{max}\}, \tag{5}$$

where $\mathrm{dist}(i,j)$ is the distance between the location at which the images with indices i and j have been taken according the location prior. All elements not in $N(i)$ will be discarded based on the prior information and thus several matching hypotheses do not need to be computed.

As reported in [30], incorporating such location prior information may lead to disconnected graph components. For an incremental search, it is however easy to connect different components by extending the definition of the children $\mathrm{ch}(x_{ij})$ of a node by

$$\mathrm{ch}(x_{ij}) \leftarrow \mathrm{ch}(x_{ij}) \cup C_1 \cup \ldots \cup C_n, \tag{6}$$

see Figure 7 for an intuitive definition of the components $C_1, \ldots, C_n$. Thus, we ensure that if the path stays in the same component, the procedure of building the graph is not changed. If, however, the path may "jump" to another component, we account for this possibility given the prior.

IV. Experimental Evaluation

Our evaluation is designed to illustrate the performance of our approach and to support the three main claims made in this paper. These three claims are: (i) our approach has the ability to run in an incremental fashion so that only few nodes are expanded so that online localization is possible, (ii) our

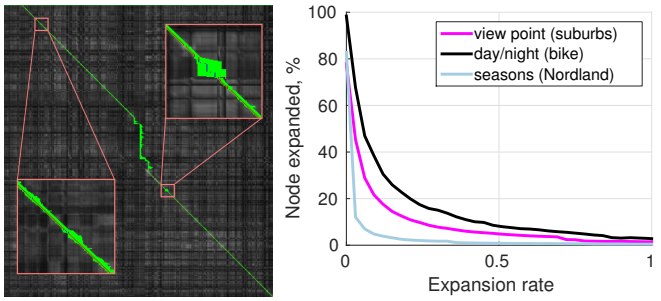

Fig. 8: Left: visualization of the graph structure for the dataset with dramatic seasonal changes (Nordland sequence from VPRiCE). The algorithm compares the images only for the nodes marked with green. Other nodes are computed for visualization only. Right: Plot of the dependency between the expansion rate α and the number of matching cost computations, expressed in percentage from total number of nodes.

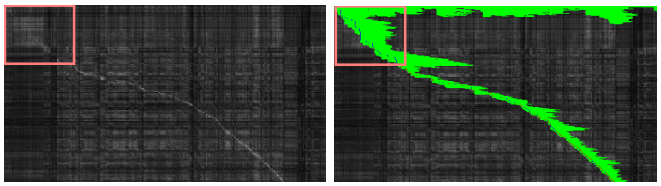

Fig. 9: Full matching matrix (left) and the nodes expanded by our algorithm (green nodes in the right image). The similarity matrix is computed for visualization only. The squares highlights an area in which most images are hard to distinguish, which leads to a larger node expansion.

heuristic is well-suited to find a competitive solution in most real world situations and (iii) our algorithm is able to exploit additional location prior information and can in this case also handle loops in robot's trajectories.

Throughout our evaluation, we rely on multiple publicly available datasets, see Figure 1. We use the summer-winter dataset from [22], [30], referred to as Freiburg and the Nordland dataset, which is a four season train ride through Norway. We also use the Alderley dataset [19] recorded during a sunny day and a rainy night. Finally, we used the datasets that have been selected for the VPRiCE Challenge 2015. The challenge consists of 4022 query and 3756 database images organized as a single sequence although it resembles multiple different datasets stitched together.

Besides setting the performance of our online method in relation to full offline matching, we compare it to OpenSeqSLAM as well as to a baseline approach that uses approximate

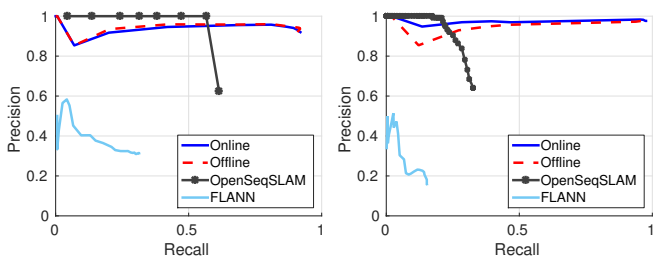

Fig. 10: Precision-recall plots for the datasets Nordland (left) and Freiburg (right).

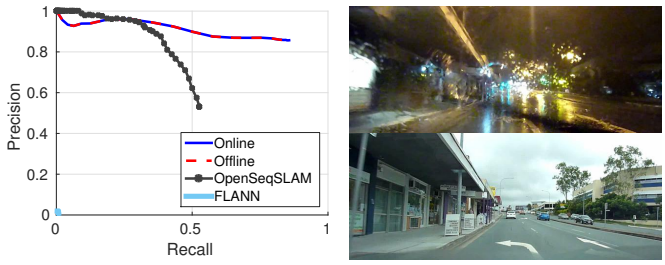

Fig. 11: Performance evaluation on the Alderley dataset.

Euclidean nearest neighbor search to find the most similar image according to the OverFeat features using the FLANN library. This approach is called FLANN in the remainder of this work and is, as we will see later on, not well-suited to solve the across season matching problem.

A. Matching Performance

The first experiment is designed to illustrate the capabilities of our approach. Figure 8 depicts the full matching matrix from a subset of the VPRiCE dataset with strong seasonal changes. Our algorithm compares $29,317$ image pairs out of $5,693,135$ possible matching that approaches such as [22] would expand. This yields a reduction of the computation cost of 99.5%, while obtaining a comparable matching performance. We obtain reductions of more than 95% for most datasets. In general, the larger the dataset the bigger the savings. Also the distinctiveness of the similarity score plays a role for our algorithm. As it can be seen in Figure 9, the block of the similarity matrix in the upper left corner shows no distinct matching pattern. As a result, our approach expands a comparably large number of nodes, indicated by the green elements in the right image. Note that our algorithm does not compute the full similarity matrix as it is shown here, we depict it for visualization purposes only.

The second set of experiments is designed to show how the proposed heuristic influences the matching performance based on the Freiburg, Nordland, and Alderley datasets. We compare the matches of our online method with those of our previous offline approach [30] using the full matching matrix but replacing the previously used HOG features by OverFeat. The results in Figure 10 and Figure 11 illustrate that our heuristic leads to comparable matching results while the number of image comparisons that need to be performed drops dramatically with increasing expansion parameter α, see Figure 12. As all precision-recall plots illustrate, we outperform OpenSeqSLAM as well as the FLANN baseline.

The performance of our algorithm has also been evaluated within the place recognition challenge VPRiCE 2015 conducted at ICRA 2015 and CVPR 2015 workshops. The evaluation has been performed by the challenge organizers. Our online algorithm achieved 3^{rd} place with precision 0.680 and recall 0.755 in the test settings. The approach that scored first [20] is an offline method and the second place [12] focuses on the design of new features for image comparisons and thus could even be combined with our method as they are rather orthogonal.

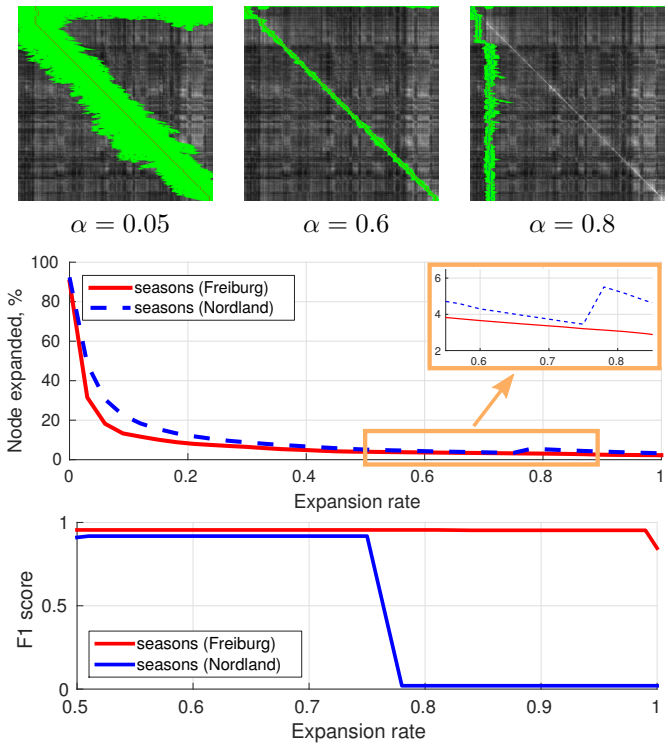

Fig. 12: In overall selecting the bigger expansion parameter α leads to a decrease in node expansion, while preserving the accuracy of the solution. The middle plot also shows that constraining α close to 1 may prevent finding the correct path. This leads to degradation in accuracy (bottom) and may lead to increase in node expansion, depending on the underlying data (middle).

B. Node Expansion

The third experiment is designed to evaluate the expansion of nodes in the data association graph in more detail. The evaluation illustrates that we can achieve online performance as only a comparably small number of nodes in the data association graph get expanded in every step. The two major factors that influence how the graph expands are the distinctiveness of matching costs and the expansion parameter α. We varied the expansion parameter of our heuristic in Eq. (4) between 0 and 1. Zero basically leads to a greedy search, while $\alpha = 1$ approximates the expected cost by the average cost of the best path. Figure 12 (middle) shows the dependency between the graph size and the applied expansion rate for the Freiburg and Nordland dataset. Roughly speaking, the closer the expansion parameter α is to 1 the smaller the resulting graph will be and vice versa. On the other hand, constraining α to the values close to 1, is likely to prevent the algorithm from finding the optimal path, see Figure 12 (top, right) for an example. In this figure, the matching nodes, which are computed using our approach, are colored in green. All other are depicted for illustration purposes only and do not need to be computed in practice. In these cases, the *reductions* in the number of expanded nodes are $\{65\%, 95.7\%, 95\%\}$ (from left to right). The figure also shows the F1 score illustrating that too large values for α can lead to a decay in matching performance. In all our experiments, we select $\alpha = 0.6$ as a good trade off between matching performance and computational savings.

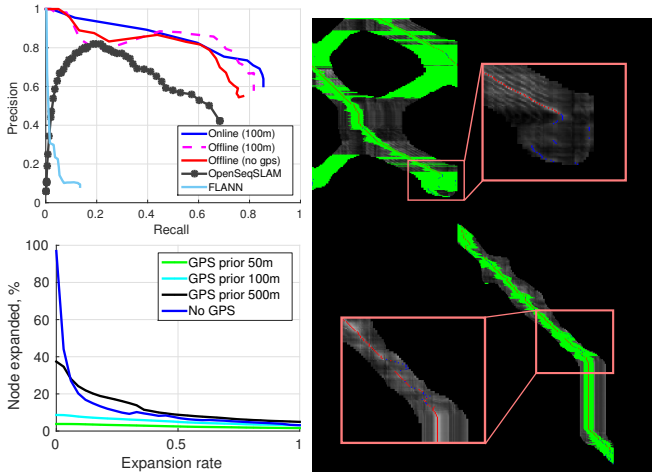

Fig. 13: Exploiting location priors enables handling the loops in image sequences. Right: Example of the similarity matrix constrained with 100m GPS prior and overlaid graph search results. Top left: comparison using precision-recall plots. Bottom left: node reductions relative to the uncertainty of the location prior.

C. Exploitation of Additional Location Priors

The last experiment is designed to show that in presence of additional but potentially noisy location priors, our algorithm is able to handle loops in the query and database trajectories as well as deviations from the database, i.e., visiting unknown areas. Figure 13 (right) depicts a similarity matrix between a query and database sequence for the scenario in which the location of the robot is known up to $100\,m$, for example obtained from a GPS receiver operating under suboptimal conditions. The green area corresponds to the nodes that are instantiated in the graph construction and search, while black areas correspond to the nodes excluded due to the location prior. Also in this settings, our algorithm is able to avoid instantiating unnecessary nodes while correctly finding the path.

Exploiting such location priors furthermore allows us to handle loops, see for example Figure 13 (top). There is almost no decrease in performance in comparison to the offline method also using OverFeat. Additionally, we outperform SeqSLAM and FLANN. Figure 13 (bottom) shows that the gain in node reduction is smaller the better the pose is known from the prior, which is an expected result.

D. OverFeat vs. HOG Features

We also analyzed the performance of the matching approaches using HOG features, as used in [22], [30] and the pre-trained OverFeat features by Sermanet *et al.* [25].

We found that the OverFeat features outperform the HOG features for place recognition under strong appearance changes as they provide more distinctive matching scores, see also Figure 4. For the HOG features, the ratio between the best match and the worse match using the cosine distance is 1.46 compared to 4.28 for OverFeat. Thus, using HOGs would be less effective for the search method presented in this paper as a larger number of nodes of the data association graph would be expanded (see green area in the last row of Figure 4). In

this sense, we confirm the results by Chen *et al.* [5] that the 10^{th} layer is well-suited for place recognition tasks.

E. Timing

Our approach can run online with around 1 fps on a standard notebook. Breaking down the timings of the individual components shows that computing the OverFeat descriptor takes the largest amount of time with approx. 500 ms. Expanding a single node, i.e., comparing two descriptors, takes 8 ms. The incremental update of the shortest path takes around 40 ms on average.

V. Conclusion

We proposed an incremental approach to visual image sequence matching under substantial appearance changes for online operation. The key idea is to apply a lazy data association approach and to define a heuristic for the search in the data association graph that estimates the similarity of images. This enables us to achieve online performance for image sequence matching under substantial appearance changes. We furthermore illustrated that noisy location priors can be exploited during online search. We implemented and tested our approach using real world image sequences acquired in summer and in winter as well as under different weather conditions. Our comparisons to other methods as well as the results from the VPRiCE 2015 place recognition challenge suggest that our approach provides competitive results and avoids expanding large portions of the data association graph or building a large matching matrix.

Acknowledgment

The authors would like to gratefully thank Luciano Spinello for fruitful discussions and feedback.

References

[1] M. Agrawal and K. Konolige. Frameslam: From bundle adjustment to real-time visual mapping. *IEEE Transactions on Robotics*, 24(5), 2008.

[2] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (SURF). *Comput. Vis. Image Underst.*, 110(3):346–359, 2008.

[3] M. Bennewitz, C. Stachniss, W. Burgard, and S. Behnke. Metric localization with scale-invariant visual features using a single perspective camera. In *European Robotics Symposium*, pages 143–157, 2006.

[4] P. Biber and T. Duckett. Dynamic maps for long-term operation of mobile service robots. In *Proc. of Robotics: Science and Systems*, pages 17–24, 2005.

[5] Z. Chen, O. Lam, A. Jacobson, and M.Milford. Convolutional neural network-based place recognition. *Australasian Conference on Robotics and Automation 2014*.

[6] W. Churchill and P. Newman. Experience-based Navigation for Long-term Localisation. *Int. Journal of Robotics Research*, 2013.

[7] M. Cummins and P. Newman. Highly scalable appearance-only SLAM - FAB-MAP 2.0. In *Proc. of Robotics: Science and Systems*, 2009.

[8] A.J. Davison, I.D. Reid, N.D. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 29, 2007.

[9] J. Fuentes-Pacheco, J. Ruiz-Ascencio, and J.M. Rendón-Mancha. Visual simultaneous localization and mapping: a survey. *Artificial Intelligence Review*, pages 1–27, 2012.

[10] P.T. Furgale and T.D. Barfoot. Visual teach and repeat for long-range rover autonomy. *Int. J. Field Robotics*, 27:534–560, 2010.

[11] A.J. Glover, W.P. Maddern, M. Milford, and G.F. Wyeth. FAB-MAP + RatSLAM: Appearance-based slam for multiple times of day. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 3507–3512, 2010.

[12] 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.

[13] D. Hähnel, W. Burgard, B. Wegbreit, and S. Thrun. Towards lazy data association in slam. In *Proc. of the Int. Symposium of Robotics Research (ISRR)*, pages 421–431, Siena, Italy, 2003.

[14] E. Johns and G.-Z. Yang. Feature co-occurrence maps: Appearance-based localisation throughout the day. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2013.

[15] C. Linegar, W. Churchill, and P. Newman. Work smart, not hard: Recalling relevant experiences for vast-scale but time-constrained localisation. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2015.

[16] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.

[17] Y. Matsumoto, M. Inaba, and H. Inoue. Visual navigation using view-sequenced route representation. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 1996.

[18] M. Milford. Vision-based place recognition: how low can you go? *Int. Journal of Robotics Research*, 32(7):766–789, 2013.

[19] M. Milford and G.F. Wyeth. Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012.

[20] D. Mishkin, M. Perdoch, and J. Matas. Place recognition with wxbs retrieval. In *Proc. of the CVPR 2015 Workshop on Visual Place Recognition in Changing Environments*, 2015.

[21] T. Naseer, M. Ruhnke, C. Stachniss, L. Spinello, and W. Burgard. Robust visual slam across seasons. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2015.

[22] T. Naseer, L. Spinello, W. Burgard, and C. Stachniss. Robust visual robot localization across seasons using network flows. In *Proc. of the AAAI Conference on Artificial Intelligence*, 2014.

[23] P. Neubert, N. Sunderhauf, and P. Protzel. Appearance change prediction for long-term navigation across seasons. In *Proc. of the European Conference on Mobile Robotics (ECMR)*, 2013.

[24] G. Pascoe, W. Maddern, A.D. Stewart, and P Newman. Farlap: Fast robust localisation using appearance priors. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2015.

[25] P. Sermanet, D. Eigen, Z. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *Intl. Conf. on Learning Representations (ICLR)*, 2014.

[26] C. Stachniss and W. Burgard. Mobile robot mapping and localization in non-static environments. In *Proc. of the AAAI Conference on Artificial Intelligence*, pages 1324–1329, 2005.

[27] E. Stumm, C. Mei, S. Lacroix, and M. Chli. Location graphs for visual place recognition. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2015.

[28] 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. In *Proc. of Robotics: Science and Systems*, 2015.

[29] C. Valgren and A.J. Lilienthal. SIFT, SURF & Seasons: Appearance-based long-term localization in outdoor environments. *Robotics and Autonomous Systems*, 85(2):149–156, 2010.

[30] O. Vysotska, T. Naseer, L. Spinello, W. Burgard, and C. Stachniss. Efficient and effective matching of image sequences under substantial appearance changes exploiting gps priors. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2015.

[31] O. Vysotska and C. Stachniss. Lazy sequences matching under substantial appearance changes. In *Workshop on Visual Place Recognition in Changing Environments at the IEEE Proceedings of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2015. Short paper.