Effective Visual Place Recognition Using Multi-Sequence Maps

Olga Vysotska and Cyrill Stachniss

***Abstract*—Visual place recognition is a challenging task, especially in outdoor environments as the scenes naturally change their appearance. In this paper, we propose a method for visual place recognition that is able to deal with seasonal changes, different weather condition as well as illumination changes. Our approach localizes the robot in a map, which is represented by multiple image sequences collected in the past at different points in time. Our approach is also able to localize a vehicle in a map generated from Google Street View images. Due to the deployment of an efficient hashing-based image retrieval strategy for finding potential matches in combination with informed search in a data association graph, our approach robustly localizes a robot and quickly relocalizes it if getting lost. Our experiments suggest that our algorithm is an effective matching approach to align the currently obtained images with multiple trajectories for online operation.**

***Index Terms*—Localization, Visual Place Recognition**

I. Introduction

LOCALIZATION is a key building block for most robot navigation systems, as robots need to know where they are in order to take navigation decisions. One form of localization, which is also relevant for performing loop-closing within the simultaneous localization and mapping problem, is the ability to identify that the robot is currently at a place already observed in the past, also known as "weak" localization. Solving this global data association problem is especially challenging for robots operating in dynamic and changing environments. A robust localization system should be able to deal with the substantial appearance changes that occur in real-world outdoor environment due to seasonal change, weather, or modifications in the scene, see Fig. 1 for an example. Thus, the goal of all such localization systems is to relate the current observation with respect to a previously recorded one or a model of the environment.

Image matching-based localization, also under substantial scene changes, is an active research field and multiple approaches have been proposed [6], [7], [16], [21], [22]. One group of approaches relies on sequence information, i.e., they exploit the fact the images are not recorded in a random order but according to the motion of the robot or vehicle through the environment. This allows for handling certain types of changes in the appearance better than when considering single images. Sequence-based approaches, however, often face the challenge that they must provide means for efficient localization even if the robot has deviated from a previously taken route. Furthermore, the majority of approaches in this area considers only the matching of two trajectories although there are exceptions such as the experience-based navigation paradigm [6], which can fuse multiple trajectories into a graph combining topological and metric information.

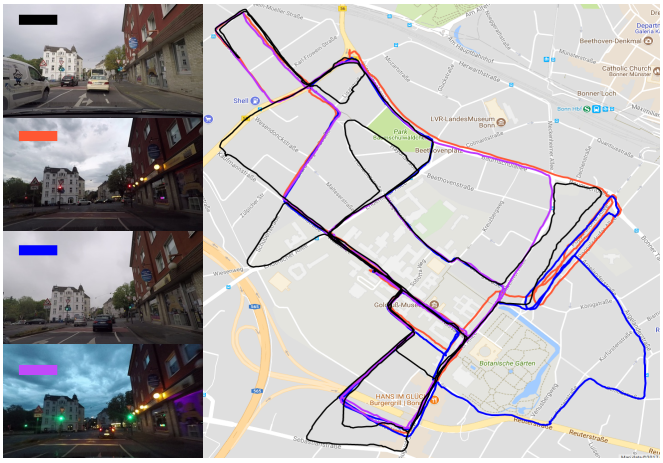

Fig. 1: Given a sequence of query images (black trajectory), our approach is able to localize the robot within multiple reference sequences (colored) of different length, shape, and visual appearance.

The main contribution of this paper is a new sequence-based visual place recognition system that localizes against a map consisting of multiple image sequences. Our system is particularly robust against drastic visual appearance changes due to a combination of expressive image features from a layer of convolutional neural network with adapted graph-based search in maps consisting of multiple trajectories of camera images. A property of our new method is that the reference sequences can be of arbitrary shape and represent places under different seasons or weather conditions. Additionally, there are no restrictions imposed on the length or amount of overlap between the trajectories as well as on synchronization, i.e., the reference trajectories do not need to be image-to-image synchronized. We address the problem of relocalization by deploying a hashing-based image retrieval with the search through a data association graph.

We evaluate our approach on different datasets to illustrate the following properties of our approach. First, it can recognize previously visited places within the map of multiple sequences. The map can consist of multiple independently collected

Manuscript received: September, 10, 2018; Revised December, 6, 2018; Accepted January, 3, 2019.

This paper was recommended for publication by Editor Cesar Cadena Lerma upon evaluation of the Associate Editor and Reviewers' comments. This work has partly been supported by the German Research Foundation under Germany's Excellence Strategy, EXC-2070 - 390732324 (PhenoRob).

The authors are with the University of Bonn, Germany, olga.vysotska@uni-bonn.de

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

sequences or be constructed from the publicly available data, like Google Street View. Second, the sequence can be collected using different setups, e.g., in cars or on bikes. Third, our matching approach is an online approach, which is fast enough to be executed on a mobile platform.

II. Related Work

Localization in real-world outdoor environments is an active field of research and one popular, image-based approach is FAB-MAP2 [7]. In order to deal with substantial variations in the visual input, however, it is useful to exploit sequence information for the alignment, compare [10], [14], [15], [16]. Another popular approach for visual place recognition proposed by Galvez-Lopez *et al.* [8] proposes a bag of words approach using binary features for fast image retrieval. Orthogonal to the exploitation of sequences, different types of features and their use for place recognition have been studied. Some approaches use variants of HOG features such as [16] or Bag of Words models optimized for seasonal changes [17]. More recently, multiple researchers apply learned features such as those proposed by Sermanet *et al.* [19] and suggested for place recognition by Chen *et al.* [5]. These CNN features yield a high matching quality but are rather high-dimensional, i.e., comparisons are computationally expensive. This motivates the binarizations of such features and efficient comparisons using the Hamming distance [3]. Another recent approach by Maffra *et al.* [12] proposes a system for viewpoint tolerant place recognition for UAV navigation. This approach uses 3D structure of the environment obtained from visual-inertia keyframe-based SLAM to be able to perform robust place recognition in presence of dramatic view-point changes. In our approach we proposed pure image based place recognition without building a 3D map of the environment.

The experience-based navigation paradigm [6] takes into account multiple sequences and stores multiple images/experiences for individual places. It extends the place model whenever matching the current images to previous ones becomes challenging. Extension of experience-based navigation targets large-scale localization by exploiting a prioritized collection of relevant experiences so that the number of matches can be reduced [10]. SeqSLAM [15] aims at matching image sequences under strong seasonal changes and computes an image-by-image matching matrix that stores similarity scores between the images in a query and database sequence. Milford *et al.* [14] present a comprehensive study about the SeqSLAM performance on low-resolution images. Related to that, Naseer *et al.* [16] focus on offline sequence matching using a network flow approach and Vysotska *et al.* [22] extended this idea towards an online approach with lazy data association and build up a data association graph online on demand. To not restrict query trajectory to follow the reference one all the time, we proposed in our paper [23] an extension that allows for flexible trajectories. To be able to achieve this flexibility, we proposed an efficient hashing based relocalization strategy as well as how to apply traditional hashing based techniques within the graph-search sequence matching framework.

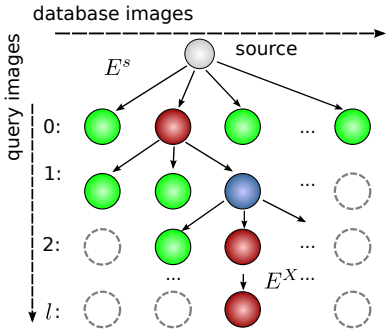

Fig. 2: Graph structure inherited from [22]. Nodes correspond to possible image matches, edges represent transitions between the matches. Green circles denote expanded nodes (for which two feature vectors are compared); red circle - match; blue - non match, but support the path hypothesis.

Typical approaches to visual place recognition start with collecting the datasets, sequences or experiences to recognize later on the places against. Collecting these maps is a time and resource consuming operation. Recently, there started to appear several approaches in the literature that tried to overcome the burden on collecting the reference dataset by exploiting already exciting sources, like Google Street View or other publicly available sources. Badino *et al.* [4] proposed a method for long term vehicle localization that can localize a vehicle with respect to previously collected topometric map as well as Google Street View. Their method deploy local keypoint features U-SURF and performs localization and tracking by applying discreet Bayes filter. To implement the state transition probability function, the authors assume to know the velocity of the robot at every point in time, whereas our approach only relies on maximal possible velocity or in other words maximum possible distance in frames (fanout). Another approach by Majdik *et al.* [13] uses Google Street View to localize a micro aerial vehicle. This setup imposes particular viewpoint challenge which they overcome by generating virtual views and match them against the street view images. Agarwal *et al.* [1] also uses the imagery from Google Street View to perform a metric robot localization. They compute rigid body transformation between input image stream from the monocular camera and geotagged rectilinear panoramic views. Afterwards, they perform a two phase nonlinear least square estimation to obtain the refined robot poses. The authors rely on an inaccurate GPS to preselect the set of panoramas to perform metric localization against. Our approach can directly provide the matching street view image to perform a more precise metric localization.

III. Our Approach

A. Graph-Based Sequence-to-Sequence Localization

Most related work on sequence matching consider one sequence to localize against. They seek to find for every image in a query sequence, e.g. the sequence of incoming images, the corresponding image in reference sequence, e.g. sequence recorded beforehand. In this subsection, we briefly summarize the main principles of graph-based image visual place recognition described by Vysotska and Stachniss in [22].

We preserve the sequential information by constructing a graph structure, where every node corresponds to the potential match between a pair of images, from query and reference sequences, and the edges represent possible transitions between the potential matches. Intuitively, the edges show if a pair of images is considered a match, where to look for the next matches, e.g. pair of matching images, see Fig. 2 for illustration. In some cases due to potentially severe visual appearance changes, for example, caused by glare or occlusions, the images that correspond to the same place produce low matching scores. To compensate for this issue the nodes in the graph are represented by two states real (red) / hidden (blue). A node becomes hidden if the cost of matching two images is higher than predefine non-matching cost m. This preserves the sequential behavior of the search, but discards visually bad matches. Given the proposed graph structure, the correspondence between images is found through the search for the shortest path in the graph. Furthermore, we adapted the proposed method to operate in online fashion, where for every incoming image, we expand only a fraction of the graph following the current best matching hypothesis and update the search. Due to the ideas of lazy data associations, we are able to track multiple hypotheses for the image matchings. For more details, please refer to [22].

One of the weaknesses of the described approach is its inability to efficiently handle flexible trajectories. We say that trajectories are "flexible" if they do not follow the same route. By using the described above graph construction procedure, we implicitly assume that both image sequences roughly follow the same physical trajectories. This means that for most of the incoming images there should be an image representing the similar place in the reference sequence. However, in real-world scenarios this is not always the case due to differently planned routes or specific map geometry. Typical problems within flexible trajectories are loops within reference trajectory and deviations of the query routes, since both cases violate the smoothness assumptions of the search. To tackle this issue, we have introduced additional edges in the graph structure that allows to handle loops within reference trajectories and proposed an efficient relocalization techniques for finding the re-entry point in case of query sequence detours. For more details please refer to [23].

In this paper, we propose an adaptation of our graph-based image sequence matching algorithm that handles multiple reference trajectories. Additionally, we describe how to leverage information available from Google Street View within our multi-sequence place recognition algorithm.

B. Multi-Trajectory Place Recognition

In realistic scenarios, one reference trajectory is typically not sufficient to cover the operational environment of the vehicle. Thus, we extend our approach to deal with multiple reference trajectories and in this way allow our system to grow a map and thus improve the coverage of the environment, both in terms of space and different appearances. In this subsection, we describe how to perform image sequence matching in the case of multiple reference sequences, also called "one-to-many" matching.

For consistency, we briefly repeat here the relocalization strategy. In previous paper, we proposed a relocalization technique based on the feature dimensionality analysis and inverted index structure. This method works better and faster than Multi-Probe Locality Sensitive hashing (LSH) [11] for very high-dimensional feature vectors, for example features from OverFeat convolutional neural network [19] or VGG-16 [20]. In this paper, we opted for newer smaller feature vectors, namely the feature vectors obtained from NetVLAD convolutional neural network [2]. The advantages of these features are comparably small size (4092) and robustness against visual appearance changes. Due to the vector size, we selected Multi-Probe Locality Sensitive hashing as a fast alternative to perform relocalization. Whenever the robot is lost, defined by the fact that there are more than 80% hidden nodes in the sliding window around current best match, we pick the top candidates from all the images using Multi-Probe LSH and select the most promising one. This matching candidate becomes a real node if the respective matching cost is lower than non-matching threshold m and a hidden node otherwise. Afterwards, we connect it to the current best matching hypothesis. We consider the relocalization to be successful if no more than 80% of the nodes within the sliding window of 5 frames in path hypothesis are hidden nodes.

The novelty of this work is the fact that the map consists of multiple sequences of the images collected in different points in time, recorded from different viewpoints, and with different frame rates. We may synchronize the sequences by performing the pairwise sequence-to-sequence matching, i.e. given our approach described so far, for all reference sequences. From this matching information, we can define an in-reference matching function $M(j,t)$ that returns for the image j from reference trajectory t all images (image index and trajectory index) that match to image j from t. If there are no corresponding images, the function returns the empty set. To enhance the localizability capabilities of the system, we have changed the representation of the map in comparison to [22] to be able to match against multiple image sequences. To incorporate this map of multiple reference sequences into our search procedure, we need to redefine the edges of the data association graph. This also leads to a slight change into the notation: Here, every node in "one-to-many" strategy is specified as x^i_{jt}, where i refers to the image id in the query sequence. The subscript jt refers to the image with id j in the reference sequence t, see Fig. 3 (Right) for visualization.

The search starts with constructing the source node x^s. Since from the beginning the robot has no information about its location, we perform a relocalization action, which includes hashing the first query image q_0 and retrieving potential candidates from a hash table $C(q_0)$, forming the first type of edges called E^s in the data association graph, given by:

$$E^s = \{(x^s, x^0_c)\}_{c \in C(q_0)} \tag{1}$$

During the search, every node that is *worth expanding* given the heuristic proposed in [22] is connected to its children within the same sequence using the set of edges E^X:

$$E^X = \{(x^i_{jt}, x^{i+1}_{kt})\}_{k=j-K,\dots,j+K} \tag{2}$$

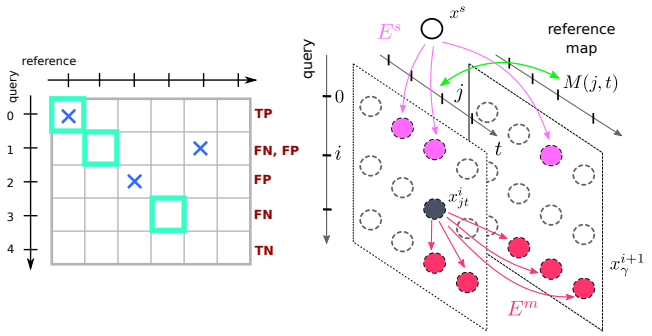

Fig. 3: Left: Evaluating per image data associations between query and reference sequences. Blue crosses denote the matches found by our algorithm. Green squares denote the ground truth solutions. TP-*true positive*, TN - *true negative*, FP - *false positive*, and FN - *false negative*. Right: E^s (pink circles) relocalization edges; E^m - correspondence edges; $M(j,t)$ (green arrow) corresponds to the images that represent the same place from different image sequences.

At the same time, we allow for transitions between reference sequences given the identified correspondences through the function $M()$. Thus, the set of edges E^m interconnects the images along the different reference trajectories, i.e.:

$$E^m = \{(x^i_{jt}, x^{i+1}_\gamma)\}_{\gamma=M(j,t)-K,\ldots,M(j,t)+K} \tag{3}$$

Finally, in case the vehicle looses track of its localization, either due to a failure or due to the fact that it had left the previously mapped area, the last node of the current best matching hypothesis x^i_* is connected to the set of the candidates obtained from our hashing scheme:

$$E^l = \{(x^i_*, x^{i+1}_c)\}_{c\in C(q_{i+1})} \tag{4}$$

Each edge $e \in E$, independently of type, has a weight $w(e)$ assigned to it. The weight is inverse proportional to the similarity score between the images. If an edge connects two nodes $(x^i_{jt}, x^{i+1}_{j't'})$, the weight $w(e) = 1/z^{i+1}_{j't'}$, where $z^{i+1}_{j't'}$ is the cosine distance between query image feature $i+1$ and reference image feature (j',t'). In this paper, we use NetVLAD features that were designed to be compared with Euclidean distance, so $w(e)$ is given by the Euclidean distance between two feature vectors now.

C. Leveraging Google Street View For Multi-trajectory Visual Place Recognition

As presented in previous section, our algorithm is able to recognize places within multiple image sequences. These reference sequences should be collected beforehand which may be a tedious work to do. Instead, we can also use already publicly available sources, like Google Street View, which provides panorama images from all over the world. Using Google Street View API, we query images from Street View given a GPS coordinate as well as a selected heading. To exploit this information within our place recognition framework, we perform a sequence of transformations that turn a set of unordered images into the map of image sequences.

We form image sequences by combining the Street View Images along streets. Then each individual street turns to be a reference image sequence. The synchronization of the reference sequences then comes naturally from incorporating the information about street crossings. Further, we will describe a way to arrange a set of images into sequences.

As a first step, we extract the GPS coordinates of the streets from OpenStreetMaps [18]. Obtaining the street coordinates from OpenStreetMaps requires only parsing the provided xml file. The OpenStreetMaps API provides for every street a set of GPS coordinates in form of street segments. The size of the line segment, e.g. the distance between the GPS coordinates, depends on the shape and curvature of the street. If the street is long and straight, we should expect small amount of GPS points with large distance between them and vice versa if the street is curvy, we get a lot of small segments that describe the physical shape of this street. Afterwards, for every segment we compute the heading of this street with respect to the North, since this is one of the parameters from Google Street View API. Heading defines basically which way the car is facing the street. Since the road segments can be quite long, we interpolate the points in between the segment endpoints to get more locations to query a panorama image from. Having GPS coordinates with associated headings for every street allows us to directly query images into sequences.

Performing visual place recognition against Street View imagery imposes several further challenges. In addition, to being collected in different point in time, with respect to query sequence as well as within the panoramas sequences itself, the frame rate of the panoramas is not constant. There are parts of the street where the density of panorama images is higher, which gives better place coverage in comparison to the places where the density is lower. Furthermore, the viewpoint change can get severe, firstly because the camera on the Google car was mounted on the poll on the rooftop, whereas the camera in our experiments is mounted inside of the car. Secondly, it is not guaranteed that the cars have taken the same lanes. This becomes particularly challenging for carrying out recognition tasks in the cities with wide streets (6 lanes), since the same place may look substantially different from different sides of the street.

IV. Experimental Evaluation

We designed the experimental evaluations to support the claims made in this paper, which are the following ones: Our approach for sequence-based visual place recognition in changing environments is able to efficiently relocalize against (i) multiple image sequences collected with similar camera setup, (ii) imagery coming from different modalities, sequences collected on bike as well as in the car, (iii) imagery from Google Street View, and (iv) we are able to localize an imagery from random YouTube video within the Google Street View. Note, there are no constraints on shape, length or visual change of the trajectories.

A. Evaluation Setup

To describe our evaluation setup, we first analyze the output of the matching algorithm. Our place recognition system reports for every query image if there is a matching image

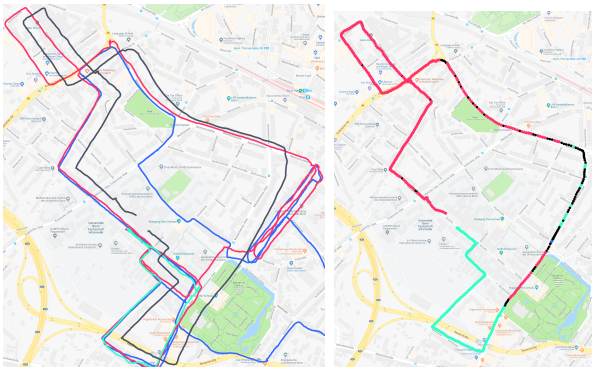

Fig. 4: Left: Query trajectory drawn in black and shifted artificially for better visibility, others are reference trajectory. Right: Query trajectory painted in the colors of the reference trajectories it was matched to.

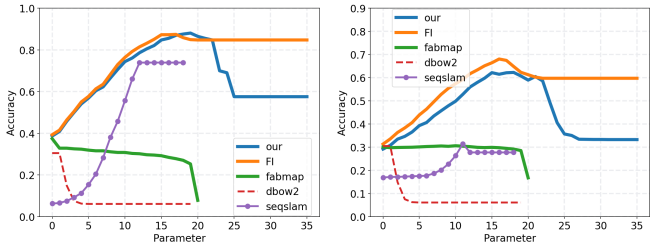

Fig. 5: Left: Accuracy plot for the dataset in Fig. 4. Right: accuracy for a larger query sequence against three reference trajectories, depicted in Fig. 1.

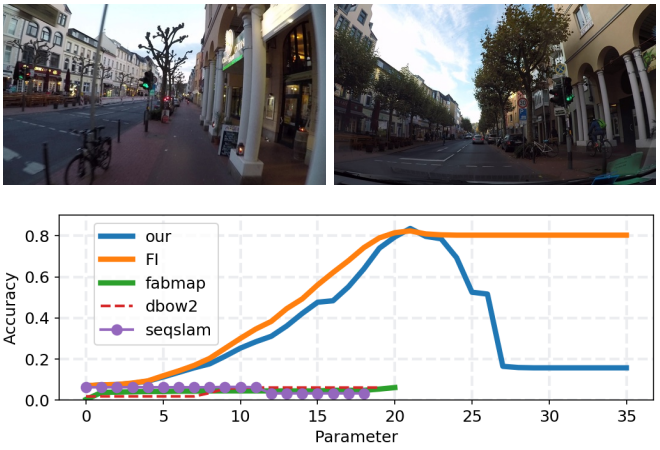

Fig. 6: Top: Matching image pair from query (bike) left and reference (car) right. Bottom: Accuracy curve. DBoW2 and SeqSLAM perform poorly on this dataset, since SeqSLAM was not designed to work with multiple sequences and DBoW2 works with descriptors that are unable to deal with seasonal changes.

in the reference dataset as well as what exactly that image is. We consider two images to match if their GPS coordinates lie within the 30 meters range. There are 5 types of situations that can happen while evaluating a match for a query image, see Fig. 3. First case, *true positive* (TP) occurs when the algorithm found a match that is in the set of ground truth matches as for the query image 0 in Fig. 3. Second case, *false negative* (FN) there is a match for a query image 3 in the dataset, but the algorithm failed to detect it. Third case, *false positive* (FP) when the algorithm has detected a match but there should be no match for a query image, as for image 2. This typically happens when the query trajectory makes a detour from the reference ones. Fourth case, *true negative* (TN) there is no match in the ground truth set and the algorithm has correctly not found it as in the image 4. The other possible situation is that there exists at least one ground truth image correspondence for a query image but the algorithm failed to detect it and found a wrong match instead as for the image 1. Then, by our definition this match is a false positive as well as false negative. To not penalize this situation twice, we increment the set of FP and FN not by 1 as for the other cases but by 0.5. Afterwards, we compute the accuracy for individual dataset as

$$\text{acc} = \frac{TP + TN}{TP + TN + FP + FN}. \tag{5}$$

Since the performance of our search algorithm depends on the non-matching parameter m, we vary this parameter to evaluate the behavior of the search. This allows to obtain the accuracy curve. For the last experiment, we do not provide the accuracy curve since we do not have GPS coordinates of the YouTube video footage.

To provide comparative evaluations, we use an open-source version of FABMAP [9] algorithm as well as open-source version of DBow2 [8]. To adjust it to our setup, we trained the vocabulary for both approaches on several extra datasets that exhibit similar visual conditions, like viewpoint changes, changes in environmental appearance, etc. We used the default provided parameters for both approaches. Since FABMAP and DBow2 do not explicitly work with reference data represented with multiple trajectories, throughout all our experiments we stacked reference trajectories into a single big trajectory. For FABMAP we select for every query image a matching image if it has the highest probability. Whenever the probability exceeds the predefined matching threshold, then the match is considered valid, otherwise the query match does not have a matching image in the reference dataset. To obtain the accuracy curve, we vary the matching probability threshold from 0 to 1. The same evaluation strategy holds for the DBow2 but there we threshold by the score and not by the probability.

Additionally, we compare our search strategy against the algorithm that compares every query image to every image in the reference dataset – a property that an online approach cannot have. This algorithm operates with the same features as ours, but selects the match with the smallest cost from all the reference sequences, making it a fully informed search, labeled as FI in the plots, also known as exhaustive search. A match for a query image is accepted if the matching cost is smaller than a non-matching cost parameter m. The curve is generated by varying the non-match parameter.

Furthermore, we have compared our approach to the state of the art approach in visual place recognition under dramatic visual appearance changes, SeqSLAM [15]. Since SeqSLAM is designed to match two sequences of images, we applied the same strategy as for FABMAP and DBow2 to convert our reference map of multiple sequences into one reference sequence. For clarification, the x-axis (Parameter) on all accuracy plots

Fig. 7: An example matching image pair from the car perspective (left) and from Google Street View (right).

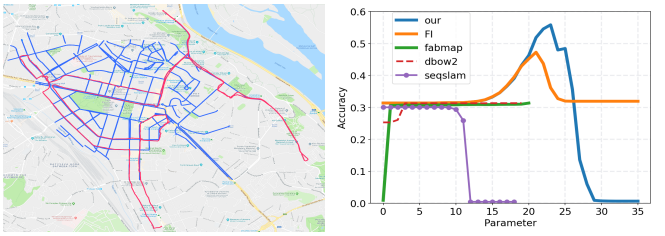

Fig. 8: Left: City streets for which panorama images were extracted (blue), query trajectory driven by a car (pink). Right: Corresponding accuracy plot.

correspond to the non-matching cost m for our algorithm and FI search, to the probability threshold for FABMAP and to weight threshold for DBoW2. The scale only shows how many parameters are used. To evaluate our approach we have collected several types of datasets. We used goPro Hero 6 camera with additional GPS for ground truth evaluations. The camera was mounted on the front window of a car or on the helmet of bicyclist. Throughout our experiments the images were extracted at 1 fps.

B. Experimental Results

In the first experiment, we show that our system is able to recognize previously visited places within multiple reference trajectories. The query trajectory consists of 636 images and was collected during the evening. There are three reference trajectories (around 3k images in total) of different shapes that were collected during the rainy morning, early and late evening respectfully. Fig. 4 shows the GPS trajectories of the reference sequences (pink, blue, cyan) as well as query (black) sequence. Fig. 4 (right) shows the trajectory of a query sequence drawn with the color of reference trajectory it was localized against. Black corresponds to the fact that no reference image was found. As can be seen most of the time, the query sequence is localized successfully against reference trajectories (pink or cyan) as well as almost no correct place associations made for the cases, where query trajectory deviates from any reference trajectories, the part where black trajectory deviates from all reference trajectories. Quantitative evaluations are shown in Fig. 5 (left). As can be seen, our approach shows similar accuracy to the fully informed matching (FI) and outperforms the FAB-MAP as well as DBow2 approaches. Fig. 5 (right) depicts accuracy results for another dataset depicted in Fig. 1 with query trajectory of 2,022 images and shows similar performance of our algorithm.

The second experiment is designed to show that our search approach is able to perform reliable visual place recognition for the cases when trajectories have been collected using

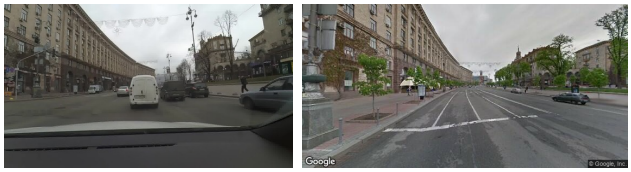

Fig. 9: A matching image pair found by our approach from a YouTube video (left) and from Google Street View (right).

camera on the dashboard of a car and on a helmet of the bicyclist. This setup imposes a particular viewpoint challenge, see Fig. 6 (top) for an example of a matching pair between a query (bike) image on the left and a reference (car) image on the right, successfully found by our algorithm. Fig. 6 (bottom) shows that our approach has a comparable performance to the FI with around 80% accuracy and they both outperform FAB-MAP, DBoW2, and SeqSLAM for this dataset.

The third experiment is designed to show that the ideas of place recognition against multiple trajectories can be successfully applied to relocalize against Google Street View. As was noted before, place recognition against street view is more challenging due to irregular frame rate of panorama images, partially drastic viewpoint changes on top of environmental visual appearance changes. The query trajectory in this experiment consists of 3,800 images whereas the total amount of extracted panorama images is 10,272. Fig. 7 shows a typical matching example from query and Street View. As can be seen from Fig. 8 (right) taking sequence information into account (our approach) outperforms the pure FI search and results at best with 58% accuracy versus 48% for informed search. Please note that DBoW2, FABMAP, and SeqSLAM were not designed to operate on multiple reference image sequences, which results in the poor performance of those algorithms in the challenging conditions tackled in this paper.

The forth experiment is designed to show that our approach can recognize places from a random street drive footage taken from YouTube. The particular challenge of this experiment lies in the fact that both query and reference sequences were collected with different cameras as well as different unknown to us positioning setup. Fig. 9 shows a matching pair that was successfully found by our approach. This experiment shows that our algorithm is robust to recognize places using images only from unknown camera setups. We do not provide the accuracy evaluations for this experiment due to the lack of exact positioning information from the YouTube video.

C. Timings

In this experiment, we confirm that the proposed algorithm allows for faster image matching than fully informed search. We performed the runtime measuring on all of the previously mentioned datasets by averaging the performance of individual datasets within the 10 runs and selecting the non-matching parameter that leads to the highest accuracy. Fig. 10 shows average matching time for a query image with respect to the size of reference dataset, e.g. total number of images in reference sequences. Since FI algorithm matches a query image to every image in the reference dataset, the time needed for finding a match grows with increasing dataset size, whereas

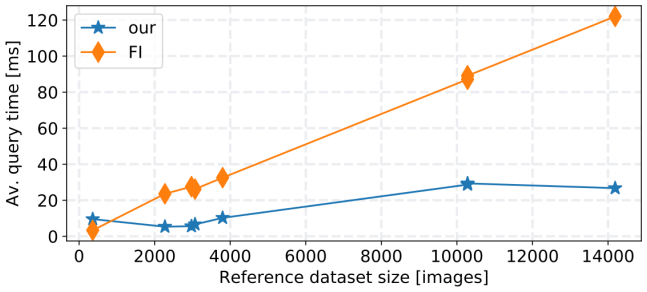

Fig. 10: Comparison of the running time for the proposed algorithm (our) and for the fully informed search (FI). Every point depicts the average time to find a match for a query image for the reference datasets of various sizes.

our approach experiences only slight increase in running time. In general, the performance of our algorithm is independent from the size of the reference dataset. To show this, we augmented the dataset that had 10,000 images with 4,000 additional ones and leaving the query trajectory the same. As can be seen the runtime of the FI algorithm for the dataset is increased whereas for our algorithm stayed almost the same. The relocalization step in our approach, however, may lead to an increase in runtime. Whenever the robot is lost, querying the candidate locations in performed via a variant of hashing, whose performance is directly influenced by the size of the dataset. Also matching capability of the features influence the search speed. The more distinctive the matching scores are, e.g., the bigger the score difference between the matching pairs and non-matching pairs is, the faster the search will reject unpromising candidates and thus runtime will decrease.

D. Limitations

Since the matching performance of our algorithm depends on the non-matching parameter m, selecting it correctly may be not an obvious thing to do. Also we observe a performance degradation whenever the visual appearance changes within the query sequence. For example, if the sequence starts at the evening and matching continues for a long time, so that it gets dark outside, the same non-matching parameter that reasonably described the non-matchiness of the sequence is no longer valid.

V. Conclusion

We presented a novel approach for quickly finding correspondences between a currently observed image stream and a map of several previously recorded image sequences given substantial appearance changes. Matching is performed through an informed search in a data association graph that is built incrementally. By deploying hashing technique, we are able to relocalize the robot if it is lost as well as between multiple image sequences. Additionally, we showed how to leverage publicly available Google Street View imagery within our framework. Our evaluations show that we can perform place recognition faster than offline, fully informed search with the comparable or better matching performance in presence of drastic visual appearance changes as well as viewpoint changes.

Acknowledgments

We gratefully acknowledge Oleg Vysotskyy for his support during data collection and Igor Bogoslavskyi for the fruitful discussions.

References

[1] P. Agarwal, W. Burgard, and L. Spinello. Metric Localization using Google Street View. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, Hamburg, Germany, 2015.

[2] 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.

[3] R. Arroyo, P.F. Alcantarilla, L.M. Bergasa, and E. Romera. Fusion and Binarization of CNN Features for Robust Topological Localization across Seasons. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2016.

[4] H. Badino, D. Huber, and T. Kanade. Visual topometric localization. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 794–799, 2011.

[5] Z. Chen, O. Lam, A. Jacobson, and M.Milford. Convolutional neural network-based place recognition. In *Proc. of the Australasian Conf. on Robotics and Automation (ACRA)*, 2014.

[6] W. Churchill and P. Newman. Experience-Based Navigation for Long-Term Localisation. *Intl. Journal of Robotics Research (IJRR)*, 2013.

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

[8] D. Galvez-Lopez and J. D. Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Trans. on Robotics (TRO)*, 28(5):1188–1197, Oct 2012.

[9] A. Glover, W. Maddern, M. Warren, S. Reid, M. Milford, and G. Wyeth. Openfabmap: An open source toolbox for appearance-based loop closure detection. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pages 4730–4735, 2012.

[10] 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 Intl. Conf. on Robotics & Automation (ICRA)*, 2015.

[11] Q. Lv, W. Josephson, Z. Wang, M. Charikar, and K. Li. Multi-probe lsh: efficient indexing for high-dimensional similarity search. In *VLDB*, pages 950–961, 2007.

[12] F. Maffra, Z. Chen, and M. Chli. Viewpoint-tolerant place recognition combining 2d and 3d information for uav navigation. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*. IEEE, 2018.

[13] A.L. Majdik, Y. Albers-Schoenberg, and D. Scaramuzza. Mav urban localization from google street view data. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 3979–3986, 2013.

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

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

[16] T. Naseer, L. Spinello, W. Burgard, and C. Stachniss. Robust Visual Robot Localization Across Seasons using Network Flows. In *Proc. of the Conf. on Advancements of Artificial Intelligence (AAAI)*, 2014.

[17] P. Neubert, N. Sunderhauf, and P. Protzel. Appearance Change Prediction for Long-Term Navigation Across Seasons. In *Proc. of the Europ. Conf. on Mobile Robotics (ECMR)*, 2013.

[18] OpenStreetMaps. https://www.openstreetmap.org.

[19] 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.

[20] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint*, abs/1409.1556, 2014.

[21] E. Stumm, C. Mei, S. Lacroix, and M. Chli. Location Graphs for Visual Place Recognition. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2015.

[22] O. Vysotska and C. Stachniss. Lazy Data Association for Image Sequences Matching under Substantial Appearance Changes. *IEEE Robotics and Automation Letters (RA-L)*, 2016.

[23] O. Vysotska and C. Stachniss. Relocalization under substantial appearance changes using hashing. In *Proc. of the IROS Workshop on Planning, Perception and Navigation for Intelligent Vehicles*, 2017.