

Appearance-based Along-route Localization for Planetary Missions

Iris Grix

Philipp Schulz

Wolfgang Stürzl

Rudolph Triebel

Abstract—We propose an appearance-based along-route localization algorithm that relies on robust place recognition by matching image sequences instead of individual frames. Our approach extends state of the art place recognition framework SeqSLAM in several aspects to realize real-time localization along routes for autonomous navigation. First, our method is online in that we only rely on the recently observed image frames. Second, we provide a homing mechanism based on rotations computed from frame matches. And third, we use a more flexible mechanism to search for matching locations, not restricting the search to straight lines in the cost matrix as in SeqSLAM, but allowing for a wide variety of route traversal conditions such as varying velocities or rotational and translational viewpoint differences. We investigate different image preprocessing steps as well as image similarity metrics wrt. their influence on illumination and viewpoint invariance for a more robust place recognition. On a new challenging dataset, recorded in real world experiments with a planetary rover, in the course of a Moon-analogue mission on Sicily’s Mount Etna, we show the feasibility of our direct, sequence-based approach to along-route localization.

Index Terms—mobile robotics, field robotics, place-recognition, autonomous route navigation

I. INTRODUCTION

Autonomous Navigation is the key capability of any mobile robotic system, and it is particularly challenging if the environment is unstructured, large-scale, and GPS-denied (see Fig. 1). Modern robotic platforms are facing these challenges more and more often, and in addition they are required to robustly navigate for longer and longer time frames. Some highly relevant example applications where these challenges arise are search-and-rescue scenarios and planetary exploration. Autonomously navigating mobile platforms are also used in factories and homes. Usually, these platforms employ some form of SLAM (Simultaneous Localization and Mapping) to create a (possibly) metric and globally consistent map of their environment and at the same time self-localizing in that map. However, autonomous following of a desired route is a different problem which is not addressed by SLAM methods. There exist many possible scenarios where a mobile robot is required to travel along the same route repeatedly, e.g. mail delivery, automated transport in factories or between landing sites and fixed stations on the moon or Mars, inspection of pipelines, fences or sample return missions.

Visual route following is usually realized using *appearance-based* navigation, also often referred to as

All authors are with the Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Wessling, Germany. Email: firstname.lastname@dlr.de. R. Triebel is also with the Dep. Computer Science, TU Munich, Germany trieb1@in.tum.de.

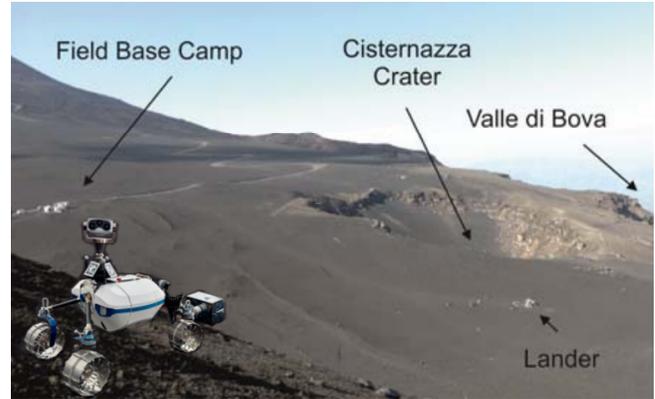


Fig. 1: The LRU lightweight rover on Mt. Etna during a moon analogue mission. Due to its lack of distinguishable features and its size, this environment is particularly challenging for an autonomous navigation system. Note that, while GPS signals are in principal available on Mt. Etna, we do not use them to obtain a more realistic treatment of the exploration scenario on the moon.

teach-and-repeat, where the current location of the camera is inferred by comparing the latest camera frame to a database of previously recorded images taken along the same route, which are organized in some form of topological or hybrid metric-topological map. Typically, these methods require a camera or stereo camera as the only sensor. Appearance-based navigation methods [1], [2], [3], [4] have recently received a lot of attention, mainly due to important achievements in terms of accuracy and the size of the maps. Nevertheless, for a long-term and large-scale application in unstructured environments, one of the main problems is still a reliable place recognition component with the main difficulty being changing illumination conditions in teach and query runs. Recently, the widely condition invariant place recognition framework SeqSLAM was introduced by Milford and Wyeth [5] which utilizes the sequence information inherent in the observations collected while traveling along a route to achieve a more robust localization. In the work presented here we adopt this idea to create an appearance-based route following method which is able to cope with changing environments.

Our method uses the principles of SeqSLAM for self-localization, but we make some adaptations in order to address the challenges of online along route localization which mainly originate from camera viewpoint changes as the robot repeatedly navigates along the route. First, our algorithm is an *online* method in contrast to the original

batch formulation, i.e. we only use the currently observed image frames for localization and navigation and not the entire sequence. Second, we provide a navigation component that relies on a *homing vector*, which we compute from the rotational difference between the current observations and the map frames. And third, we employ a more flexible matching technique based on Dynamic Time Warping (DTW).

We show that our method performs well on the datasets published with SeqSLAM. We also introduce a new challenging dataset, which was acquired with our planetary rover, the LRU (*Lightweight Rover Unit*) [6], [7] during a moon analogue mission in 2017 on Mt. Etna, Sicily as part of the ROBEX (Robotic Exploration of Extreme Environments) project [8], [9].

The rest of this paper proceeds as follows: Section II will give a short overview over previous work concerning the field of appearance-based navigation and place recognition. To facilitate understanding the inner workings of our localization algorithm and the applied adaptations to SeqSLAM, Section III introduces the key aspects of the route-based place recognition algorithm which we adapted. Section IV describes the proposed method in detail and Section V shows results of both our method and SeqSLAM on different datasets. Finally we conclude with an outlook regarding future and ongoing work towards autonomous route navigation in Section VI.

II. PREVIOUS WORK

There exists a large body of publications for appearance based navigation going back to Matsumoto [10]. Zhang and Kleeman [11] use cross correlation on omni-camera images in the Fourier domain to both localize the robot along the route and recover rotation around the z -axis. Royer et al. [4] and Šegvić [12] propose methods for route following using monocular vision. Both compute a 3D map by structure from motion and localization is realized through feature matching of the current frame to keyframes in the map. Paton et al. [3] use color constant images from stereo cameras for localization to improve performance in changing lighting conditions for long-term autonomy. Churchill and Newman [13] use SIFT features to match stereo frames. They address the problem of long-term navigation in changing environments by storing what they term a new experience whenever localization to the existing data is not possible. In [14] appearance-based navigation is applied to a UAV.

Concerning place recognition methods we will focus on SeqSLAM and its variants. For a recent summary of place recognition methods, see [15]. Variations of SeqSLAM have been published extending the original algorithm to exhibit slightly higher tolerance to viewpoint changes and traversal speed variations [16], increasing its computational efficiency [17], predicting seasonal environment changes for better matching [18], and using image features along with SeqSLAM [19], [20]. All mentioned publications apply the algorithm to images captured from a guided vehicle, and do not apply as an on-line localization as would be required by an autonomously navigating system. Always, two complete

runs along a route are matched and no route following strategy is implemented. Also, place recognition performance is low for route traversals with varying viewpoints, as would be encountered on an autonomously navigating mobile robot. In [19], using a global HOG descriptor to represent the images provides a better performance in this case, but the method requires multiple teaching runs for varying lighting conditions.

In the next section we will give a short overview over SeqSLAM, since our along-route method is essentially based on the ideas introduced there.

III. SUMMARY: SEQSLAM

SeqSLAM [5] is a visual place recognition algorithm that matches segments of routes to find images of the same location even under vast visual changes caused by differing times of day, weather conditions, or seasons. Implementations of the original SeqSLAM algorithm are freely available in Matlab [21], C++ [22] and Python [23]. We will shortly summarize the main ideas, including notes on where our method diverges.

First, two sequences of images I_1, I_2, \dots are recorded with a video camera along two passes of the same route, where one set $\mathcal{R} = (I_0^m, I_1^m, \dots)$, serves as the map and the other as the query sequence $\mathcal{L} = (I_0^q, I_1^q, \dots)$ to be matched with \mathcal{R} . From the two sets, images which correspond to the same physical location must be selected to obtain two sequences of the same size, where I_i^m corresponds to I_i^q . Our method differs here, as we only collect the map set, and self-localization will be performed on-line during every following traversal of the route. Consequently, we do not assume the database and the localization sequences to have the same size nor manually pick images of the same physical location.

In a subsequent preprocessing step all images are downsampled to an arbitrary small size (e.g. 64×32 pixels) and further divided into 8×8 pixels patches which are normalized to cover the full range of pixel values from 0 to 255 each (further called patch normalization).

$$I'(x, y) = \frac{I(x, y) - \mu(x, y)}{\sigma(x, y)}, \quad (1)$$

where $I'(x, y)$ is the new value of a pixel at (x, y) , $I(x, y)$ is its original value and $\mu(x, y)$ and $\sigma(x, y)$ are the mean and standard deviation of the pixel values of the surrounding 8×8 pixels patch. We apply in principle the same normalization, but in a sliding window filter manner (see eq. (3) in Section IV), which gives a higher invariance against rotations and translations of the images.

Downsampling and patch normalization both increase invariance to visual changes and lighting conditions between the passes.

Next, for every query frame I_j^q , the sum of absolute differences (SAD) to every database frame $[I_0^m, \dots, I_{N-1}^m]$ is computed. For every query frame, the result is stored in a vector of length N , called *image difference vector*. One key idea of SeqSLAM is that every element of the image

difference vector \mathbf{d}_j is in turn normalized across a window of n neighboring entries to emphasize the influence of the image sequence:

$$\hat{d}_{ji} = \frac{d_{ji} - \bar{d}_n}{\sigma_n}, \quad (2)$$

where \bar{d}_n and σ_n are the local mean and standard deviation of the n neighboring entries d_{ji} . We set this value to be (± 10 frames). Every normalized image difference vector forms a row in the cost matrix \mathbf{C} ,

$$\mathbf{C} = \begin{bmatrix} \hat{\mathbf{d}}_0^T \\ \hat{\mathbf{d}}_1^T \\ \hat{\mathbf{d}}_2^T \\ \vdots \\ \hat{\mathbf{d}}_{N-1}^T \end{bmatrix}$$

To find the current best matching location, linear searches through the matrix are performed to find a sequence of images with high similarity to the database images, leading to the most likely current position (see Fig. 3c). In our approach, the matrix \mathbf{C} is empty at the start, as the query sequence has not been acquired yet, and a new row is appended for every frame that is used for self-localization. Also, we replace the linear searches with a more flexible approach which we call *Online Dynamic Time Warping* (see IV-A). For a more in-depth description of SeqSLAM, the reader is referred to [5].

IV. OUR LOCALIZATION

Due to its good performance and robustness to environmental changes, we investigate its use for place recognition during in-operational self-localization along a previously traversed route. The route is represented by an ordered (sparse) sequence of images (see Sec. III). The localization wrt. the mapped images on a subsequent traversal will be performed in "real-time" on the camera video stream while the robot moves along the route a second time. While SeqSLAM performs well over vast appearance changes, it assumes that two images of the same place were taken from the same viewpoint and has very little tolerance to viewpoint changes. Also, in the datasets of [5] and [24], test and query sequences were manually aligned so that frames showing the same place are at equal positions in the map and query sequences. Our use case, self-localization of an autonomously navigating mobile robot introduces new challenges to place recognition. The query sequence (i.e. the currently observed scene) will almost never be aligned with any image in the map. The algorithm will have to be able to deal with rotational and/or lateral offsets in viewpoint and viewing direction in addition to appearance changes caused by varying illumination. Due to this additional uncertainty and the goal of creating an online localization method, we revisit key aspects of the SeqSLAM algorithm, namely the image preprocessing (for more illumination and rotation invariance), the image difference metric and the matrix search for the corresponding map image (to allow varying speeds of the mobile platform

including phases of zero motion as well as recovery from incorrect localization). In summary, the additional challenges to be addressed are

- online (in-operation) localization
- invariance to changes in viewpoint and illumination
- non-aligned map and query image sequences

How these issues were addressed is detailed in the following section. Results of our method in comparison with the original SeqSLAM, performed on the datasets used in [5] and [24], as well as our own dataset from the Mt. Etna moon analogue site are presented in section V-A.

A. Online implementation

The publicly available SeqSLAM implementations assume two full traversals of a given route are already available and both hold the same number of images. If $\mathcal{R} = (I_0^m, I_1^m, \dots, I_{N-1}^m)$, and $\mathcal{L} = (I_0^q, I_1^q, \dots, I_{N-1}^q)$ are the map and query sequences, the cost Matrix \mathbf{C} is therefore of size $N \times N$.

For online localization, the cost matrix does not exist from the beginning but needs to be computed incrementally as new frames are processed (every processed frame adds a new row to \mathbf{C}). To keep memory consumption constant, we implemented \mathbf{C} as a ring buffer holding the similarity scores of only the K most recent frames, so \mathbf{C} is at max $K \times N$. In addition, to further reduce computation time, we only compute the similarity scores to a set of map frames around the current location instead of all map frames.

In SeqSLAM, the best matching image to every frame is determined through searching on linear paths through the image difference matrix, starting at a position that lies in the future (since both traversals are already fully available) and leads to the current best matching image. For online use Wang et al. [25] use the same sequence matching but change it so that the search is started at a previously recorded location and ends at the current frame. We propose to replace the linear search with an online variant of the well known Dynamic Time Warping (DTW) algorithm, where the location of the best matching map frame depends only on the location of the previous match. This reduces the computational complexity while keeping sequence information. It also adds more flexibility to deal with non-aligned image sequences and variations in vehicle speed. The online DTW is described in more detail in section IV-C.

B. Greater invariance to changes in viewpoint and illumination

SeqSLAM uses patch normalization as described in eq. (1) to gain robustness to illumination changes between subsequent route traversals. While this is an effective approach for images taken from the same viewpoint, quantization of the image into patches may yield undesired effects if the two images to be compared are rotated wrt. each other. To increase invariance for rotated views of the same physical location, we replace patch normalization with a sliding window filter from [26], which implements a variation of

”divisive contrast normalization” as follows:

$$I'(x, y) = \frac{I(x, y) - \hat{\mu}(x, y)}{1 + \sigma_{1/2}^{-1} \hat{\sigma}(x, y)}, \quad (3)$$

where $\hat{\mu}(x, y)$ and $\hat{\sigma}(x, y)$ are the Gaussian weighted mean and standard deviation of pixel values surrounding (x, y) , and $\sigma_{1/2}^{-1}$ is a parameter allowing to adjust the strength of normalization. For more information, please refer to [26].

With this, pixel values vary more smoothly than after patch normalization and images which are rotated about the z -axis wrt. each other can be compared (using a pixel similarity measure) by shifting one image against the other. Figure 2 shows an example result of patch normalization and normalization using our filter approach. The rotational alignment is realized by computing a pixel similarity cost between a rectangular portion of the current frame which is shifted stepwise over a range of $\pm m$ pixels to every map frame. We use the sum of absolute differences (SAD) as a similarity measure. We also experimented with Census transform as the cost as it is robust wrt. illumination changes, but due to the normalization, which already reduces the influence of varying illumination, this did not lead to an observable increase in accuracy. The rotation leading to the smallest difference is then stored as additional information to the image difference vector. This information can be used to correct the robot’s heading to stay on the route. We make two moderate assumptions: firstly, we assume that the map images were taken with the camera facing in the direction of movement. Further, the ground is considered to be flat, but for use on routes with more topography, a pan tilt unit could be used to hold the camera in a horizontal position. In the case of our Mt. Etna dataset, the body of the rover was held in a horizontal position by active bogeys. This homing method has been shown to converge in the case of a lateral displacement from the route as well as a purely rotational offset in [11].

Additionally, we added sky blackening, as proposed in [16] to eliminate false information from e.g. moving clouds and make outdoor and day/night images more similar. This leads to highly increased recognition performance, especially if large portions of the images show the sky with clouds.

C. Dealing with non-aligned image sets

In SeqSLAM, sequences with images taken at different vehicle speeds or camera frame rates cannot be matched accurately as corresponding images do not form a straight line in the cost matrix. This is visible from Figure 3, where we compare the localization performance of SeqSLAM and our proposed more flexible online Dynamic Time Warping (online DTW) on an exemplary route where the camera was stopped and accelerated during the query run. The linear search through the cost matrix will not result in a correct localization if the database and query images are not well aligned.

The authors of [16] have used odometry information to align the images by selecting frames from equally spaced camera locations. Since odometry information is not always

reliable or even available, we developed a new localization strategy, calculating a minimum cost path through the cost matrix, similar to SeqSLAM but more flexible. It is not dependent on aligning the database and query frames spatially through odometry information and can deal with non-aligned image sequences resulting from stopping, accelerating, or using different frame rates during mapping and query runs.

Dynamic time warping (DTW) is a well established algorithm to align time series

$$\begin{aligned} X &:= (x_0, x_1, \dots, x_{N-1}) \\ Y &:= (y_0, y_1, \dots, y_{M-1}) \end{aligned}$$

with the goal of minimizing some local distance measure $c(x, y)$. Evaluating the local cost for every pair of elements in X and Y , one obtains a $N \times M$ cost matrix

$$\mathbf{C}(n, m) := c(x_n, y_m)$$

Note, that this corresponds to the original sequence search through the image difference matrix in SeqSLAM.

The goal is then to find a minimum cost path $p = (p_0, p_1, \dots, p_{L-1})$ through the matrix that minimizes the distance of both series and satisfies the following conditions:

$$p_0 = (0, 0) \text{ and } p_{L-1} = (N - 1, M - 1)$$

$$n_0 \leq n_1 \leq \dots \leq n_{L-1} \text{ and } m_0 \leq m_1 \leq \dots \leq m_{L-1}$$

$$p_{l+1} - p_l \in \{(1, 0), (0, 1), (1, 1)\} \text{ for } l \in [0 : L - 2]$$

called boundary, monotonicity and step size conditions. From \mathbf{C} an $N \times M$ accumulated cost matrix \mathbf{D} is computed.

Using dynamic programming, the minimum cost path is calculated in $O(NM)$ time, traversing the accumulated cost matrix in reverse order starting at $\mathbf{D}(N-1, M-1)$.

For application in navigation, where the query series is not fully known from the beginning, we adapted DTW in such a way that the accumulated cost matrix \mathbf{D} is computed incrementally as a new frame arrives and the path yielding the minimum cost is likewise calculated incrementally and in forward direction through the matrix. We altered the step size condition to

$$p_{l+1} - p_l \in \{(1, 0), (1, 1), (1, 2), \dots, (1, s)\}.$$

with s being the maximum step size, to allow for velocity differences of v_{map} and v_{query} in the range $v_{\text{query}} \in [0 : n * v_{\text{map}}]$, with $n \in [0 : s]$. Figure 4 shows a schematic of how the accumulated cost matrix and the resulting minimum cost path are calculated incrementally.

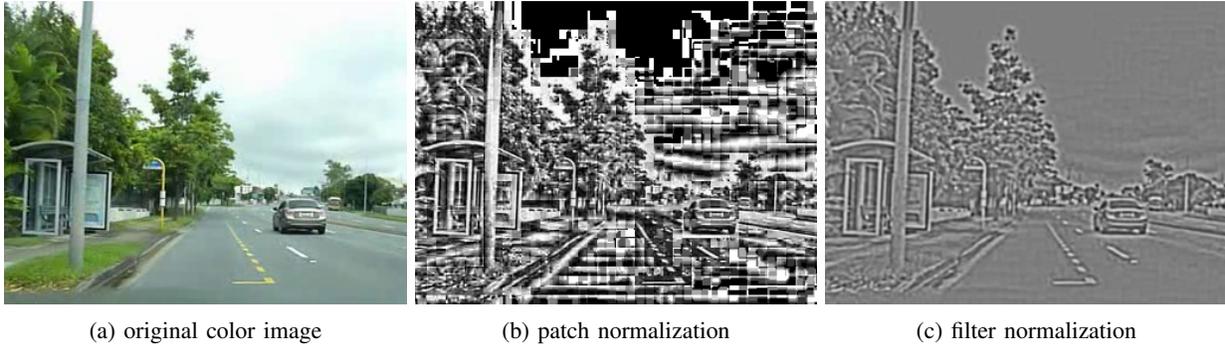


Fig. 2: Example images from the Alderley dataset showing the effect of patch normalization with 8×8 pixel patches as in SeqSLAM and the sliding window filter normalization introduced in our implementation (eq. (3)), with the normalization constant $\sigma_{1/2}^{-1} = 0.1$.

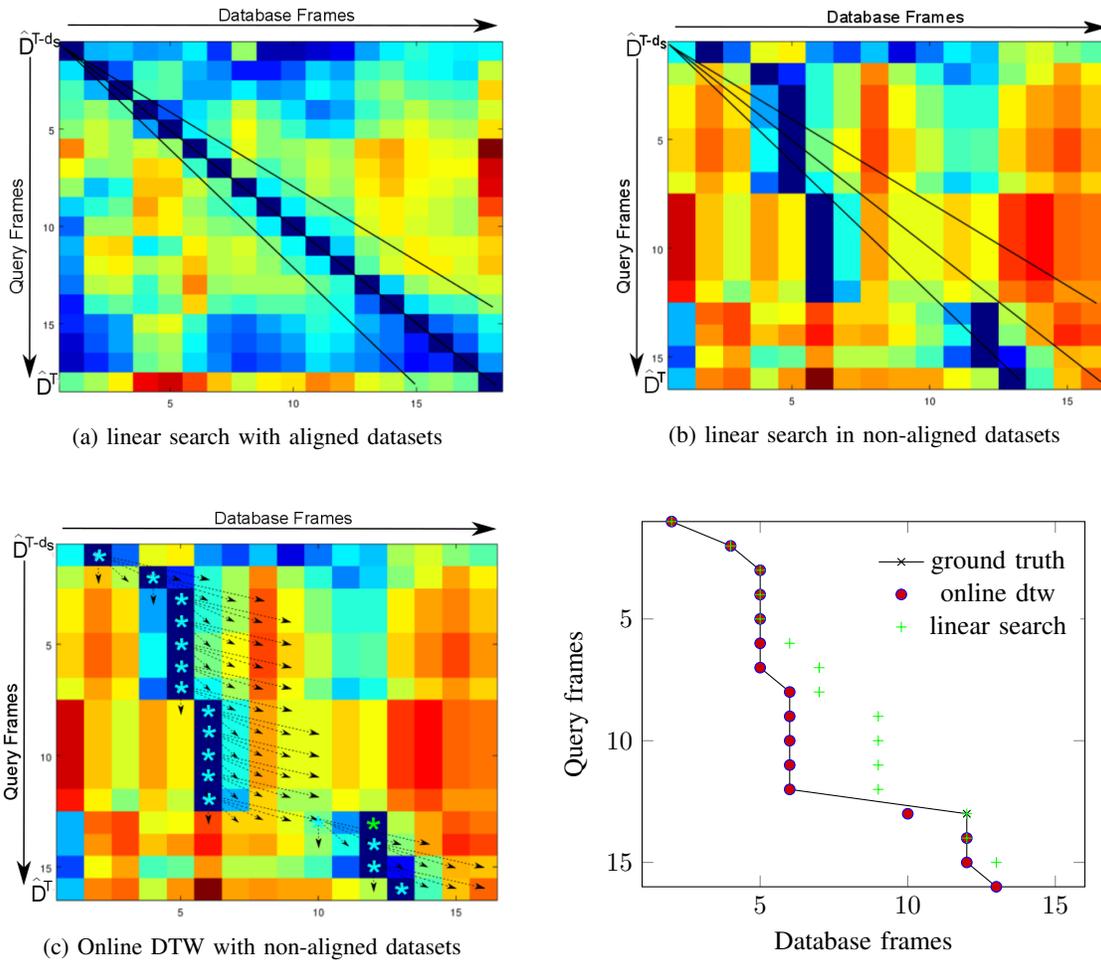


Fig. 3: Comparison of localization using linear search vs. online DTW with a maximum step size of four on small datasets of 16 map and query images for the case of aligned (matching database and query frame numbers, (a)) and non-aligned (b) and (c) data. The misalignment was introduced by stopping and accelerating the camera on the query run wrt. the map run. Subfigures a, b and c show the accumulated cost matrix $\hat{\mathbf{D}}$ with the search paths overlaid. The colors represent the accumulated cost, lowest values are dark blue, highest are red. Localization results and ground truth for the non-aligned case are shown in the plot.

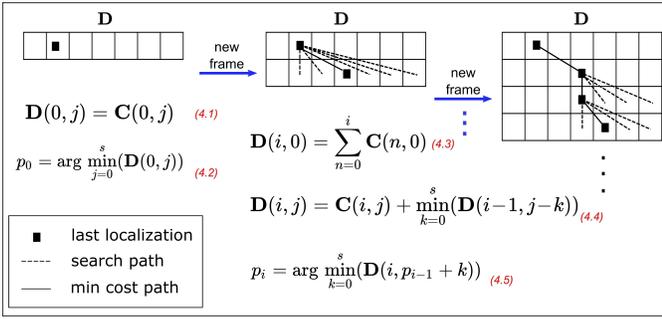


Fig. 4: Online Dynamic Time Warping. As a new frame arrives, it is preprocessed and the SAD to every map frame is calculated. The resulting difference vector forms a new row in the cost matrix \mathbf{C} . From this, the corresponding new row of the accumulated cost matrix \mathbf{D} is derived considering the maximum allowed step size s (eqs. (4.1), (4.3) and (4.4)). Next, a new localization is found as the path yielding the lowest cost wrt. the last localization (eqs. (4.2) and (4.5)), again considering s .

V. EXPERIMENTS AND RESULTS

A. Datasets

We evaluated our localization method in comparison with SeqSLAM on two datasets published in [5] and [24] as well as on our own dataset collected with the LRU rover on Mt. Etna. All datasets are available for download from [27], [28], [29] and [30].

The first dataset "Nordland" was recorded from a camera mounted in the front cabin of a train going the same route during Spring, Summer, Fall and Winter respectively. The image size is 64×32 pixels.



Fig. 5: Example images from the Nordland Summer (top left) and Spring (top right) dataset, the Alderley day (center left) and night (center right) dataset and the Etna dataset first run (bottom left) and second run (bottom right).

The second dataset was recorded through the windshield of

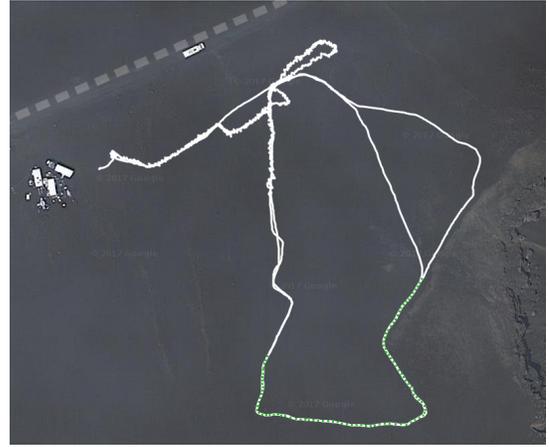


Fig. 6: The GPS track of the full longrange test run on Etna. The part which was used as dataset for our experiments is marked with green dots. The control center (two containers and a truck) can be seen on the left. A large crater, the "Cisternazza" lies to the right of the plotted track.

a car, on two rounds through a suburb of Alderley (Australia), one on a sunny day, the other in a rainy night. We created a dataset of aligned images for both day and night runs. The image size is 640×256 pixels and was downsampled to 64×32 pixels to match the experiments in [5]. In both datasets the viewpoints are aligned along the routes and not rotated between runs. The challenge to place recognition arises from the partly severe changes in appearance between seasons (Nordland) and day/night (Alderley). Finally, we evaluate the performance on our own Etna dataset which was acquired on a partially sunny and cloudy day with the LRU planetary rover during a moon analogue mission in 2017 on Mt. Etna, Sicily. We extracted the dataset from the video stream of one of the stereo cameras, recorded during a longrange navigation experiment with the rover. LRU was manually steered for two rounds of partially the same route. We used D-GPS as ground truth for the rover's location during the experiment. Figure 6 shows the GPS track of the full run plotted on an aerial image of the test site. The ROBEX mission control center is also visible on the top left of the image. The environment of the Etna dataset is challenging in many aspects. As can be seen from Figures 1, 5 and 6, it is largely (locally) monotonous with a lack of salient features, which increases the risk of spatial aliasing. Additionally, the changing lighting conditions from cloud shadows aggravate place recognition. In contrast to the other two datasets, the extracted frames exhibit offsets both along route and rotationally between the two traversals which only allows for an approximate localization to one of the map frames. To simulate an online localization, we chose the map frames to have a larger spatial separation than the query frames with a ratio of about 2:1. For the presented experiments, 190 map frames and 353 query frames, downsampled to 50×50 pixels were used.

In all experiments except for the last, SeqSLAM was run

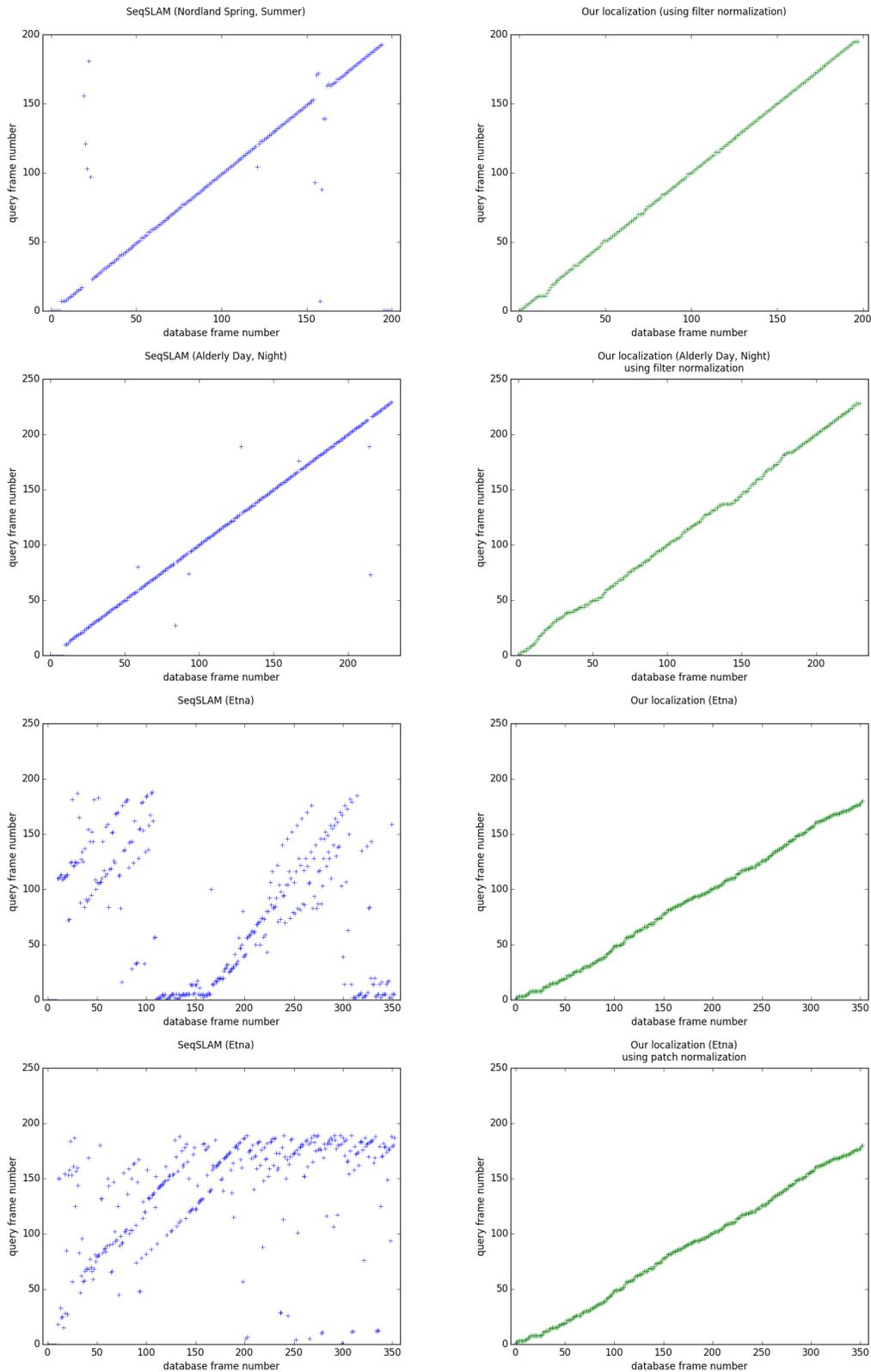


Fig. 7: Matching results of SeqSLAM (left) and our method (right) for the Nordland Spring/Summer (top row), the Alderley day/night (center row) and the Etna (bottom two rows) datasets. In the first three experiments, SeqSLAM was run with default parameters (patch size 8, matching distance 30 frames) and our method used the filter normalization. In the last experiment, we ran SeqSLAM with the filter normalization and our method with the patch normalization.

with a patch size of 8x8 pixels and a matching distance of 30 frames which corresponds to the default parameters. Our method was run using our sliding window filter normalization and the online DTW with a step size of $[0 : 2]$ frames. The results of the experiments, running SeqSLAM and our online localization on all three datasets are summarized in figure 7.

On the least challenging Nordland Spring/Summer images, both methods perform well. SeqSLAM slightly outperforms our method on the Alderley day/night sequence. We accredit this to the online DTW which trades a part of the sequence information for flexibility to find matches over a wider range in the sequence matrix. The result for the Etna dataset is harder to interpret, since the map and query frames do not correspond one to one and the query frames generally lie between two map frames and thus could be matched to either. Nevertheless, the spatial ratio of query and map frames is very close to 1:2, making the correct slope in the plots 0.5. Clearly, SeqSLAM fails to find correct matches while our method can adapt well to the given circumstances. This robustness results from limiting the search for corresponding map images to 0, 1 or 2 frames around the latest localization while SeqSLAM searches the full map sequence. In the last experiment, we switched the image normalization for both methods. Surprisingly, this did not influence the performance of our method noticeably while leading to very different but not better localization results in SeqSLAM.

VI. CONCLUSIONS AND FUTURE WORK

We conclude that the sequence based direct approach of SeqSLAM in combination with the more flexible online DTW is interesting for along route localization in visually challenging scenarios and could be used to guide an autonomously navigating robot along a desired path or as a method to correct accumulated errors in autonomous route following. Invariance to appearance changes can be achieved to a great extent by appropriate image preprocessing and exploitation of the sequence information in route datasets. Additional viewpoint invariance for all axes is not as easily achieved. Here, using point features or image patches to compute the relative pose of the current frame to candidate corresponding map images may lead to a more robust and exact localization as well as provide homing information for an autonomous platform to stay on or find back to the desired path. Investigating these options for further improvement of our along route localization is work in progress.

REFERENCES

- [1] M. Cummins and P. Newman, "FAB-MAP: Probabilistic localization and mapping in the space of appearance," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008.
- [2] D. Prasser, M. Milford, and G. Wyeth, "Outdoor simultaneous localization and mapping using RatSLAM," in *Field and Service Robotics*, 2006, pp. 143–154.
- [3] P. Furgale and T. D. Barfoot, "Visual teach and repeat for long-range rover autonomy," *Journal of Field Robotics*, vol. 27, no. 5, pp. 534–560, 2010.
- [4] E. Royer, J. Bom *et al.*, "Outdoor autonomous navigation using monocular vision," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2005, pp. 1253–1258.
- [5] M. J. Milford and G. F. Wyeth, "SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 1643–1649.
- [6] A. Wedler, B. Rebele *et al.*, "LRU – lightweight rover unit," in *13th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*, 2015.
- [7] M. J. Schuster, C. Brand *et al.*, "The LRU rover for autonomous planetary exploration and its success in the spacebotcamp challenge," in *International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, 2016, pp. 7–14.
- [8] A. Wedler, M. Vayugundla *et al.*, "First results of the ROBEX analog mission campaign: Robotic deployment of seismic networks for future lunar missions," *68th International Astronautical Congress (IAC)*, 2017.
- [9] "Robex webpage," www.robex-allianz.de/en, [Accessed: 2018-07-29].
- [10] Y. Matsumoto, M. Inaba, and H. Inoue, "Visual navigation using view-sequenced route representation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 1996, pp. 83–88.
- [11] A. M. Zhang and L. Kleeman, "Robust appearance based visual route following for navigation in large-scale outdoor environments," *The International Journal of Robotics Research*, vol. 28, no. 3, pp. 331–356, 2009.
- [12] S. Šegvić, A. Remazeilles *et al.*, "A mapping and localization framework for scalable appearance-based navigation," *Computer Vision and Image Understanding*, vol. 113, no. 2, pp. 172–187, 2009.
- [13] W. Churchill and P. Newman, "Experience-based navigation for long-term localisation," *The International Journal of Robotics Research*, vol. 32, no. 14, pp. 1645–1661, 2013.
- [14] T. Nguyen, G. K. I. Mann *et al.*, "Appearance-based visual-teach-and-repeat navigation technique for micro aerial vehicle," *Journal of Intelligent & Robotic Systems*, vol. 84, no. 1, pp. 217–240, Dec 2016. [Online]. Available: <https://doi.org/10.1007/s10846-015-0320-1>
- [15] S. Lowry, N. Sünderhauf *et al.*, "Visual place recognition: A survey," *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 1–19, 2016.
- [16] E. Pepperell, P. I. Corke, and M. J. Milford, "All-environment visual place recognition with smart," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 1612–1618.
- [17] Y. Liu and H. Zhang, "Towards improving the efficiency of sequence-based SLAM," in *IEEE International Conference on Mechatronics and Automation (ICMA)*, 2013, pp. 1261–1266.
- [18] N. Sünderhauf, P. Neubert, and P. Protzel, "Predicting the change – a step towards life-long operation in everyday environments," *Robotics Challenges and Vision (RCV)*, 2013.
- [19] T. Naseer, L. Spinello *et al.*, "Robust visual robot localization across seasons using network flows," in *AAAI Conference on Artificial Intelligence*, 2014, pp. 2564–2570.
- [20] E. Johns and G.-Z. Yang, "Feature co-occurrence maps: Appearance-based localisation throughout the day," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 3212–3218.
- [21] N. Sünderhauf, "OpenSeqSLAM," <https://openslam.org/openslam.html>, [Accessed: 2017-09-14].
- [22] S. Okita, "OpenSeqSLAM," <https://github.com/subokita/OpenSeqSLAM>, [Accessed: 2017-09-14].
- [23] T. Madl, "pySeqSLAM," <https://github.com/tmadl/pySeqSLAM>, [Accessed: 2017-09-14].
- [24] N. Sünderhauf, P. Neubert, and P. Protzel, "Are we there yet? Challenging SeqSLAM on a 3000 km journey across all four seasons," in *Workshop on Long-Term Autonomy, IEEE International Conference on Robotics and Automation (ICRA)*, 2013.
- [25] Y. Wang, X. Hu *et al.*, "Improved Seq SLAM for real-time place recognition and navigation error correction," in *7th International Conference on Intelligent Human-Machine Systems and Cybernetics*, 2015, pp. 260–264.
- [26] W. Stürzl and J. Zeil, "Depth, contrast and view-based homing in outdoor scenes," *Biological Cybernetics*, vol. 96, pp. 519–531, 2007.
- [27] N. Sünderhauf, "Nordland dataset," <http://www.tu-chemnitz.de/etit/proaut/datasets/nordland/64x32-grayscale-1fps.tar>, [Accessed: 2017-09-14].
- [28] M. Milford, "Alderley day video," <https://www.youtube.com/watch?v=OicCfIB1L7k>, [Accessed: 2017-09-14].
- [29] —, "Alderley night video," <https://www.youtube.com/watch?v=K0npaUJB6hY>, [Accessed: 2017-09-14].
- [30] I. Grix, "Etna dataset," <http://www.robex-allianz.de/datasets/route-following/>, [Accessed: 2018-07-27].