Vision-Based Markov Localization Across Large Perceptual Changes

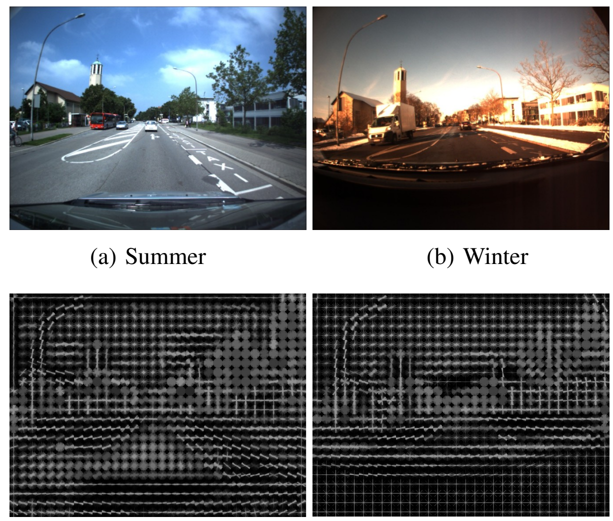
Tayyab Naseer Benjamin Suger Michael Ruhnke Wolfram Burgard
Autonomous Intelligent Systems Group, University of Freiburg, Germany
{naseer, suger, ruhnke, burgard}@informatik.uni-freiburg.de

***Abstract*—Recently, there has been significant progress towards lifelong, autonomous operation of mobile robots, especially in the field of localization and mapping. One important challenge in this context is visual localization under substantial perceptual changes, for example, coming from different seasons. In this paper, we present an approach to localize a mobile robot with a low frequency camera with respect to an image sequence, recorded previously within a different season. Our approach uses a discrete Bayes filter and a sensor model based on whole image descriptors. Thereby it exploits sequential information to model the dynamics of the system. Since we compute a probability distribution over the whole state space, our approach can handle more complex trajectories that may include same season loop-closures as well as fragmented sub-sequences. Throughout an extensive experimental evaluation on challenging datasets, we demonstrate that our approach outperforms state-of-the-art techniques.**

I. Introduction

Purely vision-based place recognition has made great advances in the last decade [7], [14], [15]. However, localizing robots over varying environmental conditions is still a challenging problem. Although using a monocular camera as the only sensor makes a robotic navigation system much cheaper, localization gets harder. The optimal place recognition system is robust towards changes in the scene caused by illumination, weather, time of the day and seasons. In this paper, we address the problem of robust place recognition across seasons. We focus on handling complex trajectories with partially overlapping routes and multiple loop closures in the database as well as the query sequence. Recent approaches perform well in handling large perceptual changes under special assumptions, e.g., manual image viewpoint alignments and linear motion models [15], [17]. Our approach can operate on raw images captured with a real robot driving in an urban environment at variable velocities and also taking different routes in each run. Furthermore, it does not require GPS information or other prior knowledge about the position of the robot and no dense temporal information. In contrast to the work of Pepperell *et al.* [18], our method can match two image sequences without any odometry information.

A keypoint-based description of an image often changes dramatically compared to the description of the image taken in a different season. As a consequence there remains only a sparse set of stable features, which makes it very hard to match such images. One way to overcome this limitation is to use a dense image description. Therefore, we partition an image into a grid and compute HOG [8] descriptors for each cell. Fig. 1 shows two example images with their corresponding grid-based feature description used in our work. We compute a similarity score for two images captured in different seasons by comparing the HOG descriptors for each cell. By doing this for all image pairs, we obtain a complete similarity score matrix.



(a) Summer (b) Winter

(c) Oriented gradients (d) Oriented gradients

Fig. 1: Local gradient information in the images is robust to seasonal changes and provides correct data associations.

Global image-based localization over the similarity matrix produces considerable false positives as it does not take into account the sequential nature of images captured with a moving robot. Therefore, we propose to use Markov localization-based temporal filtering over the similarity matrix to exploit the sequential information. This allows our method to handle revisits, cope with detours and process images captured at different frame rates. In our previous work [16], we build a data association graph over the similarity matrix and model image matching as minimum cost flow. The flow costs are based on the similarity scores and the path with minimum cost through the network is the optimal matching path hypothesis for one flow. To handle loop closures or not reachable states, it requires multiple flows to find all matching paths. As result we get a collection of independently estimated maximum likelihood

This work has been supported by the European Commission under the grant number ERC-AG-PE7-267686-LifeNav and FP7-610603-EUROPA2.

978-1-4673-9163-4/15/$31.00 © 2015 IEEE

paths. The main drawback of the network flow approach is the limited connectivity in the graph that restricts the transitions to the local neighborhood. Thus, the resulting estimate is not a joint maximum likelihood estimate over all path hypotheses. A fully connected graph that would overcome this limitation leads to a substantial memory requirement. This becomes unfeasible for long trajectories and still will be unable to handle loop closures in the database sequence in a single flow.

In contrast, our proposed method computes the joint probability densities for matching images with a discrete Bayes filter and performs sequential filtering on the resulting thresholded probabilities. The hypotheses are computed in a single pass instead of iteratively solving the full network flow problem. It is memory efficient for large trajectories as we do not need to build a graph and still can handle all potential state transitions. Therefore our method can handle more complex robot trajectories and can also be run online on a robot. Reducing the complexity of the path hypotheses estimation method might lead to overconfident estimates in certain scenarios. Although, we discuss such a scenario in our experimental evaluation, we demonstrate that our method outperforms the state-of-the-art on most real world datasets.

II. Related Work

Vision-based place recognition for autonomous robots has gained great importance in recent years [4], [7], [9], [11]. Robust localization is a vital part of a Visual Simultaneous Localization and Mapping (V-SLAM) framework. Traditional approaches for vision-based place recognition assume similar visual appearance across query and database images [3], [7]. The problem gets harder with large perceptual changes in the environment as compared to the robot's first visit to the place. These changes are caused by occlusions, variations in the illumination and seasonal changes like snow and foliage. Most of the approaches rely on keypoint-based image description for matching images [2], [13]. Keypoints tend to be unstable over longer periods of time and therefore the corresponding descriptors are not useful for matching images as shown in [16].

SeqSLAM [15] combines whole-image based descriptor matching and exploits sequential information for visual localization across day and night. The authors report a substantial improvement over keypoint-based global image localization under severe environmental changes. Furthermore, SeqSLAM assumes a linear trajectory and pre-aligned images to match viewpoints across two video sequences. These assumptions generally do not hold for typical real world robotic applications. The authors of [19] combined U-SURF image description of panoramic images with epipolar constraints to match images across seasons over a small database of 40 images. Glover *et al.* [11] combine RatSLAM [14] and FABMAP [7] to produce consistent maps over different times of the day. Both systems complement each other and produce less false matches under different visual conditions. Their approach is sensitive to longer detours and still suffers from unstable SURF features over different times of the day. Approaches like presented by Badino *et al.* [1] combine range and visual information and use it in a Bayesian framework to achieve robust localization across seasons whereas our approach is purely vision-based. Their approach is sensitive to longer detours with respect to the mapping run.

Training the system over a span of time, helps to learn the transformation of appearance changes in the scene. Neubert *et al.* [17] build vocabularies of superpixels and use it to predict the appearance in the new season. The approach assumes pixel aligned datasets for learning the visual vocabularies, an assumption which generally does not hold for typical real world robotic datasets. The authors of [5] proposed an experience-based navigation framework. They learn multiple appearances of the same place over time. For every subsequent visit, the query image is matched to all the appearances for the best match. It is a nice approach for handling extreme perceptual changes but requires longer learning periods and visual odometry. Hasen *et al.* [12] use dynamic time warping to find the most likely path through an environment by using global velocity constraints. The authors only report results for non-linear sequence-based matching and do not discuss path detours and multiple loop closures. Vysotska *et al.* [20] reduces the computational complexity of [16] and do not build up the full matching matrix by exploiting an uncertain GPS prior.

III. Visual Localization Utilizing Sequential Information

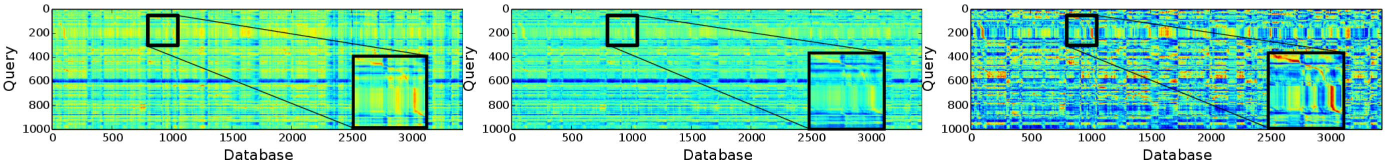
The goal of our approach is the robust localization of a mobile robot in changing environments as well as handling complex trajectories. The only sensor we use is a camera, taking images at low and variable frame rates, which is not adequate to compute visual odometry. The only prior knowledge we have is that the images are collected in a sequence, which is a reasonable and natural assumption. Each dataset we consider consists of two sets of images. We will refer to the first image sequence as the database, which is a temporally ordered set of images $\mathcal{D} = (d_1, \ldots, d_D)$ that constitutes the visual map of places with $D = |\mathcal{D}|$. The query set $\mathcal{Q} = (q_1, \ldots, q_Q)$ with $Q = |\mathcal{Q}|$ refers to the second image sequence which was recorded after a substantial perceptual change in the environment. Those changes may arise from different times of the day or different seasons. In order to compute correspondences between the two image sets, we use a discrete Bayes filter. We first give a brief description of the filter in Sec. III-A. The static transition model is depicted in Sec. III-B. Our sensor model is based on the cosine similarity of a whole image HOG-Descriptor as outlined in Sec. III-C. Subsequently, we explain the computation of the final belief matrix in Sec. III-D. Each row of the matrix is a probability distribution for an query image over all the correspondences with the database images. Finally, we describe our post processing scheme to filter the potential matching set for sequences In Sec. III-E.

A. Discrete Bayes Filter

We use Bayes Filters to represent the state at time t by a distribution of a random variable x_t, conditioned on the sensor data history.

$$Bel(x_t) = p(x_t | z_1, \ldots, z_t)$$

In the discrete case the domain of the random variable x_t can be mapped to a subset of $\mathbb{N}$. It is known that the complexity of the estimate for the belief grows exponential over time, making it computationally demanding [10]. If the dynamics of



(a) Raw scores (b) Scores normalized along columns (c) Scores after two pass normalization

Fig. 2: This figure shows the effect of two pass normalization used in our approach. We show a zoomed-in part of the score matrix for better visualization. Column wise normalization in the first pass helps to disambiguate the confusing database images which could match against all query images and provide high scores. Although the scores do not look highly discriminative, it greatly reduces the noise in the matrix. In the second pass we stretch the scores between 0 and 1 to achieve more distinctive scores.

the system, $p(x_t|x_{t-1})$, is known, the belief can be computed recursively and efficiently without loss of information.

$$Bel(x_t) = \eta_t p(z_t|x_t) \sum_{x_{t-1}} P(x_t|x_{t-1}) Bel(x_{t-1})$$

Where η_t is a normalization constant and the sum goes over all possible states of x_{t-1}. In our case, for all timesteps $t \in \{1, \dots, Q\}$, the domain of the random variable x_t is $\{1, \dots, D\}$. This means that we can match every image from the query set $\mathcal{Q}$ to every image of the database set $\mathcal{D}$.

B. State Transition Model

Whenever our robot traverses the environment the images are recorded in a sequence. Note that we do not know the topological connectivity of the images. Moreover it is possible, that the second sequence deviates to previously unseen parts of the environment or that places are revisited during the runs. Therefore, we choose the following transition model for forward, stationary and backward transitions that uses the parameters $c_f, c_b, c_s > 0$:

$$P(x_t|x_{t-1}) = \alpha_t * \begin{cases} c_f & \text{if } f > x_t - x_{t-1} > 0 \\ c_s & \text{if } x_t - x_{t-1} = 0 \\ c_b & \text{if } -b < x_t - x_{t-1} < 0 \\ 1 & \text{else} \end{cases}$$

Where α_t is a normalization factor to ensure a valid probability distribution. This means, that we have a range $b, f > 0$ which is the maximum step width we consider to be likely if we are within a sequence. A forward transition in this range is c_f times more likely than going to any other place outside the range $[-b, f]$. The same applies for going backward with factor c_b and staying at the same place with factor c_s. This transition model entails advantages compared to other state-of-the-art methods and is more flexible in handling complex and partially overlapping trajectories. In contrast, the flow network of our previous work ([16]) allows only local transitions when $0 \leq x_t - x_{t-1} < K$ for a constant value K. SeqSLAM [15] considers the transition model to be linear.

C. Robust Image Matching

Our approach is purely vision-based and does not integrate any additional information from other sensors than a monocular camera. Since we put our main focus on datasets where the two image sets are recorded in different seasons, we have to deal with large perceptual changes. We show that the gradient information provides a robust description of a scene even after large perceptual changes as shown in Fig. 1. We compute HOG descriptors on a grid of 32x32 pixels over the full image $\mathbf{I}$ of size 1024x768. We stack cell descriptors to form a whole-image descriptor and use them to compute a pairwise similarity score between two images. We compute the similarity between images $q_i \in \mathcal{Q}$ and $d_j \in \mathcal{D}$ with the cosine similarity of the two normalized image descriptors, respectively $\mathbf{I}_{q_i}$ and $\mathbf{I}_{d_j}$:

$$s_{ij} = \mathbf{I}_{q_i} \cdot \mathbf{I}_{d_j}, \tag{1}$$

where $s_{ij} \in [0, 1]$ and $s_{ij} = 1$ indicates full similarity. The similarity matrix $\mathbf{S}$ has a size of $Q \times D$ and consists of all s_{ij}. The idea behind the sensor model is that the likelihood for image q_i matching to image d_j is proportional to s_{ij}.

$$p(z = q_i | x_t = j) \propto s_{ij}$$

To obtain more distinctive scores, we perform a two step normalization on the raw cosine similarities s_{ij}. First, we normalize the similarity matrix column-wise such that every column has mean one.

$$\hat{s}_{ij} = s_{ij} \left(\frac{1}{Q} \sum_{j=1,\dots,Q} s_{ij} \right)^{-1}$$

In a second step, we normalize row-wise and spread the values between $[0, 1]$.

$$\tilde{s}_{ij} = \frac{\hat{s}_{ij} - \min\limits_{i=1,\dots,D} \hat{s}_{ij}}{\max\limits_{i=1,\dots,D} \left(\hat{s}_{ij} - \min\limits_{i=1,\dots,D} \hat{s}_{ij} \right)}$$

Fig. 2 illustrates the effect of the normalization procedure on the raw similarity matrix. Despite of the noise in the similarity matrix, the zoomed in view shows that the values are more distinctive, which supports the impact of the measurement model in the Bayes filter. We set the likelihood for our sensor model according to the normalized score

$$p(z = q_i | x_t = j) \propto \tilde{s}_{ij}.$$

Given this likelihood we can compute the belief for every query image with the discrete Bayes filter.

D. Forward Backward Propagation

We initialize the filter with a uniform distribution, $Bel(x_0 = j) = 1/D$ for $j = 1, \dots, D$. The final belief matrix, where each row is a probability distribution over the

Fig. 3: This figure shows pair of images matched using our approach. Left image of each pair is the query image and on the right is the image retrieved from database. Our localization method is robust to foliage color changes, occlusions and noise like sun glare.

correspondences, is computed by two passes of the filter. The first one, Bel_f, is computed in the direction of movement of the robot. Due to the nature of the recursive propagation within a sequence based model, the distribution gets more peaked the longer we are in a sequence. This may lead to an overconfident estimate in case of jumps. To account for such situations, we compute the probability distribution of a second filter on the same image set in reverse order. Currently, this makes it a batch approach where all the query images are processed for the final belief matrix. Given a more informed state estimate, ideally we would not require backward propagation and then the approach can easily be run online. The final belief matrix is computed as the normalized geometric mean of the two belief matrices,

$$Bel = \lambda \cdot \sqrt{Bel_f \cdot Bel_b}$$

Where the $\cdot$ is the element wise multiplication and the square root is evaluated element wise as well and λ is the $Q \times D$ normalization matrix.

E. Sequential Filtering

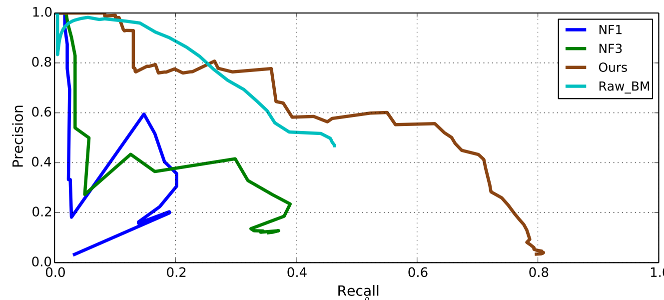
Neither the transition model nor the sensor model provide exact information and therefore the belief matrix still contains outliers. Still, we can compute a reliable estimate of matching trajectories by taking into account the sequential information. First, we neglect unlikely matches below a certain threshold and then cluster local peaks into a potential matching set. In this way, we represent a local neighborhood only with a single match, corresponding to the maximum likelihood of that neighborhood. In the final step, we search for sequences of local peaks in the matching set. The sequence search uses the following parameters: the minimum sequence length l_s, the maximum gap in rows g_r and columns g_c between two matches in a sequence. More formal, a set of row-wise ordered matches

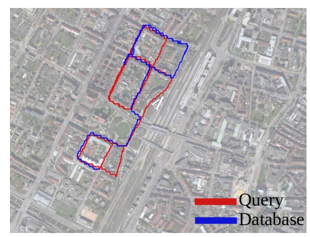
$$\mathcal{M} = \{(i_1, j_1), \ldots, (i_n, j_n) | \forall k = 1, \ldots, n-1 : i_k < i_{k+1}\}$$

with $\mathcal{M} \subset \mathcal{Q} \times \mathcal{D}$ is a sequence if and only if $n \geq l_s$ and for all $k = 1, \ldots, n-1$ the condition $i_{k+1} - i_k \leq g_r$ as well as $|j_{k+1} - j_k| \leq g_c$ is satisfied. This constraint ensures local sequences of a certain length and neglects short and isolated matches that often correspond to false positives. The final trajectory is then returned as the union of the sequences that passed the sequence test.

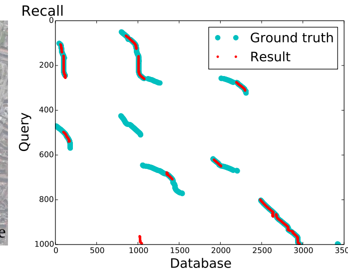
IV. Experiments

We evaluated our approach on four different real world datasets and compared it to our previous work [16], SeqSLAM [15] and a naive row-wise best match of the raw score matrix (RawBM) as baseline. We will refer to our previous approach as Network Flow, using x number of flows (NFx).





(a) GPS Trajectories



(b) Matching sub-sequences

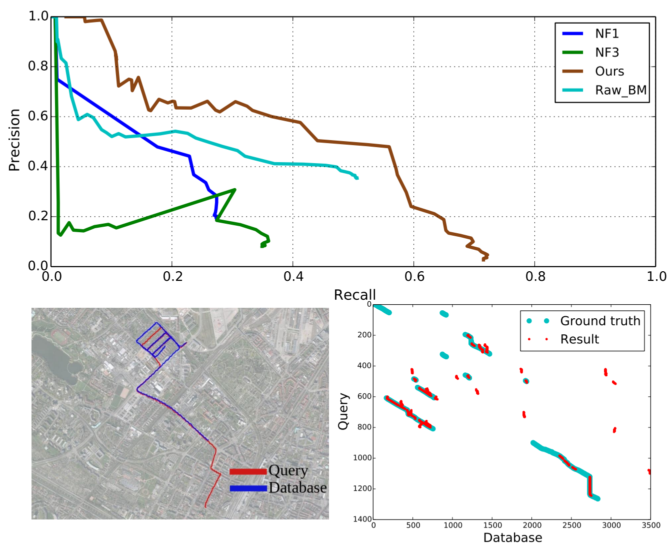
Fig. 4: Our method outperforms both NF and RawBM. NF requires multiple flows to retrieve all the matches because of the sparse graph connectivity, and also collects false positives in subsequent iterations. Constrained transition probabilities make it hard to retrieve all matching sub-sequences.

To quantify the results we calculate precision and recall, where we considered a match as true positive if and only if the distance in the ground truth was less than 6 frames in either direction. Given a speed of $10m/s$ and a frame rate of four Hz this corresponds to a radius of $\sim 15m$. To visualize the results we plot the precision-recall curves, since they are quite intuitive. To compare different precision-recall curves we use the maximum F_1-Score, which is the harmonic mean of precision and recall. For a positive real β the F_β-Score is defined as

$$F_\beta = (1 + \beta^2) \cdot \frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall}$$

Setting $\beta = 1$, like in our case, means that precision and recall are weighted equally.

Fig. 3 shows the matching pairs retrieved using our approach and highlights the challenging scenarios like different viewpoints, illumination and seasonal changes. This makes the datasets even more challenging, since all of these factors affect the matching quality. But it indicates the generality of the approach, since the two sequences may even be recorded by different platforms. The first dataset (Seq1) consists of



(a) GPS Trajectories (b) Matching sub-sequences

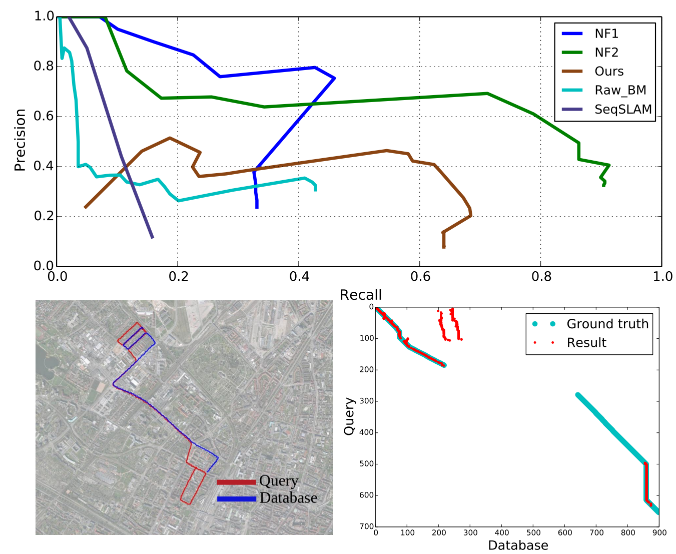
Fig. 5: Our approach performs the best for this challenging sequence as it contains multiple short overlapping routes and database loop closures. This dataset contains visually similar blocks as the car was driven in parallel streets which resulted in more false positives.

3431 images from sequence recorded in winter 2012 at four hz (database) and 1001 images from a sequence recorded in summer 2012 with one Hz (query). The trajectory includes loops in the database and non overlapping parts. The first row of Fig. 4 shows the precision-recall curve for the trajectory, where our approach outperforms the network flow-based sequential filtering as well as the best first match. The F_1-Scores for all dataset are given in Tab. I. This sequence consists of many loops and only short matching sequences. A setting that is difficult for the Network Flow, since it has to find very short sequences in multiple flows and might also take false positives on the way. Our approach instead is able to handle multiple matches with the more flexible transition model that allows transitions between all places, of course with different probabilities. Fig. 4(a) shows the GPS trajectories for both database and localization runs. Fig. 4(b) shows the temporal groundtruth trajectory and the retrieved matches using our approach.

The second dataset (Seq2) consists of 1,441 images in the localization run and 3,601 images in the database run. The car was driving around visually ambiguous street blocks. The precision-recall curve as well as the trajectory and our solution to it is shown in Fig. 5. Although this trajectory is challenging, our approach performs best among all the compared approaches.

Method	Seq1	Seq2	Seq3	NewCollege
Raw_BM	0.48	0.43	0.38	0.68
NF1	0.26	0.30	0.57	0.68
NF3/NF2*	0.35	0.31	**0.70***	0.79
Ours	**0.59**	**0.52**	0.51	**0.90**
SeqSLAM[1]	-	-	0.17	0.68

TABLE I: F1 Scores over all datasets.



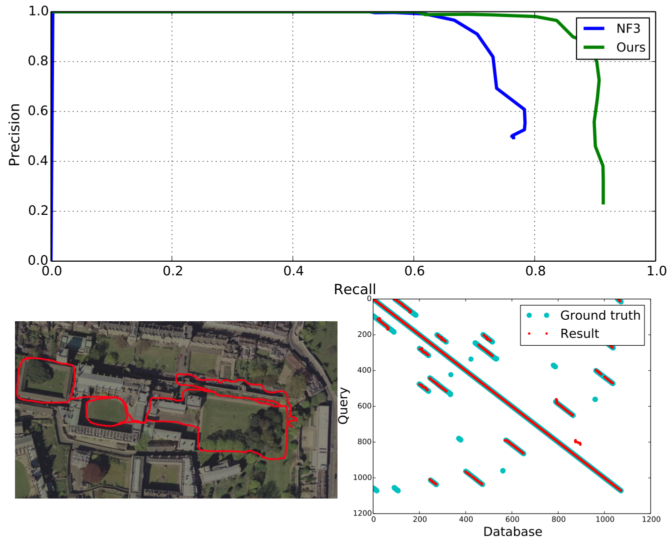
(a) GPS Trajectories (b) Matching sub-sequences

Fig. 6: Network flow outperforms our proposed method on this dataset as the best matching sequence remains globally consistent with the true matches and avoids false positives because of the restricted transition probabilities. Our approach suffers from a false positive sub-sequence with high matchings scores and as the belief is updated for all the possible states, these matches are considered as the part of the matching sub-sequence.

In our third experiment, we chose a dataset with 781 images for the localization sequence and match against 1328 images in the database (Seq3). This dataset consists of two longer matching sequences with a large gap in between them and no loops, see Fig. 6(b). Such a setting favors the Network Flow and therefore NF1 and NF2 outperform our approach. The general challenge is that, even after normalization, the score matrix is not fully distinguishable in large parts. This makes it hard to detect the non matching sub-sequence as true negative. In this case, our approach suffers from an overconfident probability distribution, see the false positive matches on the top left in Fig. 6(b). This also explains the unnatural shape of the precision-recall curve, since it loses the true positive sequence before it drops the false positive ones. The Network Flow can keep track on the real trajectory, since it is mostly consistent in the global path cost, but still needs two flows since the sub-sequences are too far apart.

In the last experiment, we chose the New College dataset [6], a typical single run mapping scenario with identical database and query sets. Nevertheless, we do not treat the dataset different from others, which means that we expect the approaches to detect the diagonal as well as the symmetric loop closures. In this scenario RawBM and NF1 are only able to match the diagonal, because the similarities there are equal to 1, which is the maximum possible score. Therefore, we did not plot the precision-recall curves for RawBM and NF1 in the top row of Fig. 7. Our approach is able to capture most of the matches, see Fig. 7(b). The overall performance is much better than in Seq1 - Seq3, but this is how one would expect

[1]The first two datasets were extremely challenging and no reasonable image matching was obtained using SeqSLAM.



(a) GPS Trajectories (b) Matching sub-sequences

Fig. 7: A dataset for same season loop closure. SeqSLAM and NF1 can only match the diagonal and are not plotted in the top row. NF3 achieves reasonable results and our approach performs best, finding most of the matching sub-sequences (b).

it, since here we are in the same season and the similarity scores are much more distinctive in this case. Taking a closer look at the solution of our approach (Fig. 7(b)) we see that we miss only some short sub-sequences. For the same reason NF3 provides reasonable results. The main drawback is that it can not handle multiple matches for a single pass. So the first flow finds the diagonal, the second flow the longer sub-sequences on the upper triangular part and the third flow does the same thing on the lower triangular part.

Throughout our experiments we demonstrated that our approach can cope with complex trajectories that contain loop closures in the database as well as in the query and jumps forward and backward in the database. Our previous work can not handle these situations, mainly caused by the restrictive transition probabilities. Furthermore, we showed that our method also outperforms our previous work in an inner season visual place recognition task.

V. Conclusion

We presented an approach for purely vision-based localization under extreme perceptual changes using Bayesian filtering. For each query image we consider transitions to every place in the database which yields higher precision and recall as compared to our previous work [16]. In this way, our approach can handle loops in the image sequences and also retrieve matches in a single pass without using any sort of position priors. In our experiments, we identified situations where one approach is superior to the other and vice versa and we also showed that our proposed approach outperforms existing state-of-the-art methods. Our current approach is fairly easy to implement, computationally light and provides more flexibility for the possible trajectories of the robot.

References

[1] H. Badino, D. Huber, and T. Kanade, "Real-time topometric localization," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 1635–1642.

[2] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 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*, 2006, pp. 143–157.

[4] C. Cadena, D. Gálvez-López, J. D. Tardós, and J. Neira, "Robust place recognition with stereo sequences," *Robotics, IEEE Transactions on*, vol. 28, no. 4, pp. 871–885, 2012.

[5] W. Churchill and P. Newman, "Practice makes perfect? managing and leveraging visual experiences for lifelong navigation," in *Proc. of the IEEE Int. Conf. on Robitics and Automation (ICRA)*, 2012.

[6] M. Cummins and P. Newman, "Fab-map: Probabilistic localization and mapping in the space of appearance," *The International Journal of Robotics Research (IJRR)*, vol. 27, no. 6, pp. 647–665, 2008.

[7] ——, "Highly scalable appearance-only SLAM - FAB-MAP 2.0," in *Proc. of Robotics: Science and Systems*, Seattle, USA, June 2009.

[8] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. of the IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[9] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, p. 2007, 2007.

[10] D. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello, "Bayesian filtering for location estimation," *IEEE pervasive computing*, vol. 2, no. 3, pp. 24–33, 2003.

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

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

[13] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[14] M. Milford and G. Wyeth, "Persistent navigation and mapping using a biologically inspired slam system," *Int. J. Rob. Res.*, vol. 29, no. 9, pp. 1131–1153, Aug. 2010.

[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 Int. Conf. on Robitics and Automation (ICRA)*, 2012.

[16] T. Naseer, L. Spinello, W. Burgard, and C. Stachniss, "Robust visual robot localization across seasons using network flows." in *AAAI Conference on Artificial Intelligence (AAAI)*, 2014.

[17] P. Neubert, N. Sünderhauf, and P. Protzel, "Superpixel-based appearance change prediction for long-term navigation across seasons," *Robotics and Autonomous Systems*, 2014.

[18] E. Pepperell, P. Corke, and M. Milford, "Towards persistent visual navigation using smart," in *Proc. of Australasian Conference on Robotics and Automation (ACRA)*. ARAA, 2013.

[19] C. Valgren and A. Lilienthal, "SIFT, SURF & Seasons: Appearance-based long-term localization in outdoor environments," *Robotics and Autonomous Systems*, vol. 85, no. 2, pp. 149–156, 2010.

[20] 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.