2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Congress Center Hamburg
Sept 28 - Oct 2, 2015. Hamburg, Germany

Robust Visual SLAM Across Seasons

Tayyab Naseer Michael Ruhnke Cyrill Stachniss Luciano Spinello Wolfram Burgard

***Abstract*— In this paper, we present an appearance-based visual SLAM approach that focuses on detecting loop closures across seasons. Given two image sequences, our method first extracts one descriptor per image for both sequences using a deep convolutional neural network. Then, we compute a similarity matrix by comparing each image of a query sequence with a database. Finally, based on the similarity matrix, we formulate a flow network problem and compute matching hypotheses between sequences. In this way, our approach can handle partially matching routes, loops in the trajectory and different speeds of the robot. With a matching hypothesis as loop closure information and the odometry information of the robot, we formulate a graph based SLAM problem and compute a joint maximum likelihood trajectory.**

I. Introduction

There has been a tremendous progress in the area of visual Simultaneous Localization and Mapping (SLAM) over the last couple of years, especially in the context of appearance-based place recognition, which is an essential building block to successfully find loop closures in the context of SLAM [6], [17], [18]. Knowing that a robot revisits the same place enables it to reduce the accumulated drift in its pose estimates along a trajectory and leads to more consistent maps. State-of-the-art systems perform well in detecting loop closures under minor perceptual changes in the environment but most of them fail to find loop closures across seasons. Often, feature descriptors change drastically over time. Reliably detecting loop closures, however, is essential to successfully perform life-long navigation, since it allows to relate all available information of a mobile robots operating across seasons. In this paper, we focus on computing consistent trajectories over longer periods of time and aim at achieving robust place recognition across seasons. In the related work [19], the authors crop an image into three parts and use one of these cropped regions for image matching. In contrast to this, our approach does not require any image cropping or the selection of an image subset. As consequence it can operate on raw data captured with a mobile robot, driving in an urban environment at different velocities. Furthermore, we do not require GPS information or other prior knowledge about the position of the robot.

One of the main reasons for current state-of-the-art systems to fail is that feature descriptors of the same place, captured in different seasons can change drastically. This makes image-matching across seasons a hard problem. To approach this problem, we need a descriptor that can cope with a high variance in appearance. In this paper, we use global image features from Deep Convolutional Neural Networks (DCNNs), which have shown to outperform traditional features for image classification and object detection tasks [13]. We explore the robustness of these features throughout extensive experiments and present a comparison to Histogram of Oriented Gradients (HOG), used in our previous work [20].

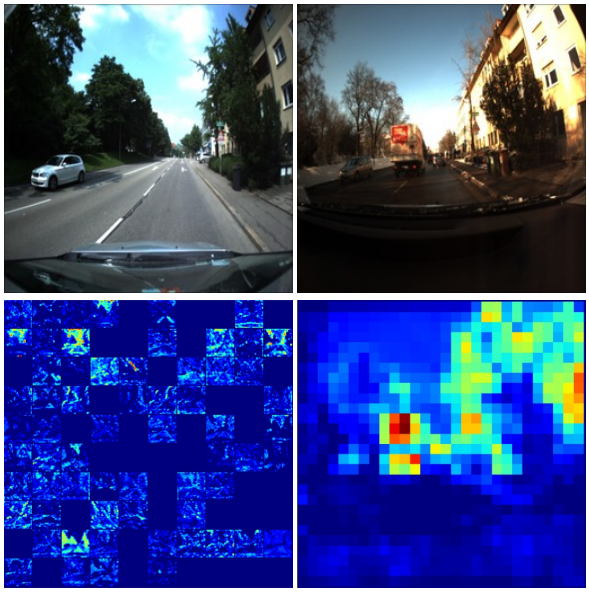


Fig. 1: The top two images show an example of the same location captured in summer and winter to illustrate typical perceptual changes across seasons. These changes are caused by occlusions and changes in illumination, vegetation, and perspective. To robustly match sequences across seasons we use the responses of a pre-trained deep convolutional neural network as feature. Below, we see the responses of the 96 filters for the summer image at the interception layer of GoogLenet and one of these 96 filters response for the winter image that in this case corresponds to its saliency map.

Matching images just according to the best similarity score produces considerable false positives which might result in inconsistent trajectories. Therefore, we build a directed data association graph over the similarity matrix to leverage sequential information. This allows our method to handle revisits, deal with occlusions and process images captured at different frame rates. As a result, our method achieves more robust loop closure detections. In the final step, we use these data associations together with the odometry information of

Tayyab Naseer, Michael Ruhnke, Luciano Spinello and Wolfram Burgard are with the Department of Computer Science, University of Freiburg, Germany. Cyrill Stachniss is with the Department of Photogrammetry, University Bonn, Germany. This work has been partly supported by the European Commission under the grant numbers ERC-AG-PE7-267686-LifeNav and FP7-610603-EUROPA2

the robot to formulate a graph-based SLAM problem and compute a joint maximum likelihood trajectory estimate. Our experimental results demonstrate that our approach is useful for estimating consistent trajectories across seasons.

II. Related Work

Long term visual localization has emerged as one of the vital aspects for lifelong autonomy. Over a course of time, robots experience variations in their environment. These can be introduced by either man-made changes or natural changes. A great amount of research has been dedicated to appearance-based mapping [8], [6], [7]. Perceptual changes caused by seasons and weather conditions make the problem harder. Image-based localization under these scenarios have recently gained importance. Traditional approaches use keypoint-based descriptors which tend to be unstable when the appearance changes. Valgren *et al.* [25] achieve robust localization using SURF [3] features along with geometrical constraints. Glover *et al.* [9] combine RatSLAM [17] and FABMAP [6] to produce consistent maps over different times of the day. Both systems complement each other and produce less false matches for better mapping under different visual conditions.

Recently, featureless sequence-based SLAM has shown a great improvement over feature-based global image localization, as presented by Milford *et al.* [16]. Their approach assumes the same route for each run, therefore requires preprocessing of the datasets. Approaches like presented by Badino *et al.* [2] combine range and visual information and use it in a Bayesian framework to achieve robust localization across seasons. Their approach is sensitive to longer detours with respect to the mapping run. Vysotska *et al.* [26] reduces the computational complexity of [20] by exploiting a rough GPS prior and do not build up the full matching matrix.

The approach presented by Churchill *et al.* [5] learns visual appearances of the same place over time. Every place not recognized in the previous experiences is added as a new experience of that place. It requires learning over long periods of time before it can cope with all perceptional changes. There exist various features to encode global or local information of images. Recently, features from DCNNs have shown to outperform traditional feature-based methods for image retrieval, object detection, and image classification tasks [13]. These networks automatically learn millions of parameters as feature representation using a huge amount of training data. A very recent approach has used these features to perform visual place recognition [4], but does not address matching images across seasons, and assumes linear trajectories. A recent approach evaluates the performance of DCNNs for place recognition across seasons [22]. It evaluates the system on the Nordland dataset, which has pixel aligned images, linear trajectories, and was recorded in a non-urban environment. Both approaches use networks pre-trained on the ImageNet database [21]. Whereas, we use a network trained on the Places [27] database and show that it performs favorable over ImageNet networks in most of our experiments, including the Nordland dataset. The main intuition behind this is that the Places database consists of images which are more suitable for outdoor localization.

We extended our previous approach [20] towards Simultaneous Localization and Mapping (SLAM) across seasons. In contrast to tessellated HOG feature descriptors in our previous work, we use global image features from DCNNs. These features provide more robust image description and we achieve higher accuracy in our localization tasks. Furthermore, our extended approach performs image matching in real-time using a GPU implementation instead of offline processing. As result, we can perform image matching over substantially longer trajectories for more robust data associations. Furthermore, we evaluate the performance of DCNNs trained with different datasets to highlight our choice. Finally, we use the resulting data associations together with the robot odometry, within a metric SLAM framework to produce consistent trajectory estimates across seasons.

III. Robust Visual SLAM Across Seasons

The goal of our approach is to compute a joint trajectory estimate for at least two datasets. Each dataset consists of a set of images with corresponding odometry information. We will refer to the first image sequence as 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 set $\mathcal{Q} = (q_1, \ldots, q_Q)$ with $Q = |\mathcal{Q}|$ refers to the query sequence that was recorded in a different season or after a substantial scene change. To obtain a joint least-square estimate for the individual trajectories, the SLAM solution, we first have to compute the full similarity matrix between all query and database images which is explained in III-A. In the second step, our approach leverages the sequence information by constructing a flow network graph to find the most likely image sequences according to the similarity matrix, explained in III-B. As result of this procedure, we obtain a set of loop closure constraints that we insert into a least-square optimization problem together with the provided odometry and utilize a state-of-the-art graph-based optimization framework to estimate a consistent joint trajectory, which we discuss in III-C.

A. Robust Image Matching

Place recognition with considerable perceptual changes pose problems for conventional keypoint-based image matching approaches, e.g. SURF [3], and SIFT [15]. Image characteristics change drastically over seasons. Lifelong visual mapping in urban environments encounters changes in appearance, illumination and structures. Roads are covered with snow in winter, and the illumination and color characteristics of a place undergo extreme variations as illustrated in Fig. 1. Keypoint-based descriptors change under these circumstances and eventually provide a non-coherent image representation [20]. Recently, the image feature representation from DCNNs like AlexNet [13] have gained great attention from the computer vision community for various image recognition and classification tasks and they outperform traditional feature based methods for these tasks.

DCNNs learn various features from millions of training images for classification tasks. They consist of convolutional layers in its early stages and inner product layers in the final stages. Feature representation towards the final stages are biased to the training data and the early stages provide very generic and large dimensional feature representation. For our place recognition task, we compare the ImageNet [21] and the Places [27] database. We use GoogLenet [24] and Alexnet architectures inside the Caffe framework [12]. In our approach, we use pre-trained Caffe models and extract features from the middle stages of the network. We chose to use the interception module 1 (icp1) layer of GoogLenet, trained on the Places database. Since the authors of Snderhauf *et al.* [22] have reported Conv3 of Alexnet trained on ImageNet to behave more robust to seasonal changes on their datasets, we present a detailed comparison in Section IV. As input to our framework, we use the full RGB image without cropping. We resize all images to 256x256x3. Both Conv3 and the icp1 layer provide a full image descriptor with dimensions 65536 and 75264 respectively. It is comparable to our previous dense HOG implementation with 65536 dimensions. But in contrast the HOG feature descriptor was computed on a grid of 32x32 pixels over the full image of size 1024x768, and the descriptors were matched cell wise to compute a pairwise similarity score. 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 $\mathbf{I}_{d_j}$:

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

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}, i.e., the cosine similarities between all images of $\mathcal{Q}$ and $\mathcal{D}$, computed according to Eq. (1). For our next task of finding the best match for each query image, we leverage the sequential information as explained in the following section.

B. Sequence Matching

Just using the best score for every query image with respect to the database in the similarity matrix $\mathbf{S}$ often leads to false data associations since the best match might not be the true positive. A more robust data association can be computed by also incorporating the sequence information. Therefore, we formulate the sequential matching as a minimum cost flow problem [1]. It seeks to find a path through a network which contains the highest number of highly scored matches between two sequences according to our similarity matrix. It is formulated as a Directed Acyclic Graph (DAG) between a source node and a sink node. Every entry in the similarity matrix corresponds to a node in the graph and is connected to its neighbors with edges. Each edge has a capacity c and a weight w. To handle non-matching subsequences, we introduce additional nodes in the graph as hidden states. To highlight our minor extension in the graph connectivity we briefly revisit our graph model. A detailed description can be found in our previous work [20].

Our DAG consists of $\mathcal{G} = (\mathcal{X}, \mathcal{E})$, where $\mathcal{X}$ are the nodes and $\mathcal{E}$ the edges. The set $\mathcal{X}$ contains four types of nodes:

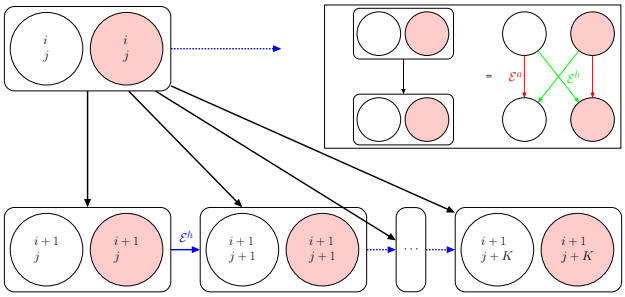


Fig. 2: Edge connections between matching nodes (white) and corresponding hidden nodes (red) in $\mathcal{G}$. For two connected nodes, both the matching state and the hidden state from the first node can either reach the matching or the hidden state of the second node. Nodes are only connected horizontally in the same row of the corresponding similarity matrix or the following row. The edges and the corresponding notations of the edge sets are colored accordingly for better understanding of the connections.

the *source* x^s, the *sink* x^t, the *matching nodes* x_{ij}, and so-called *hidden nodes* $\breve{x}_{ij}$. The total amount of flow F travels from x^s to x^t node. A node x_{ij} represents a match between the i-th image in $\mathcal{Q}$ and the j-th image in $\mathcal{D}$, which is triggered by the similarity score in $\mathbf{S}$ and corresponds to the hypothesis that the image pair i, j corresponds to same place in both sequences. For every node there is also an implicit hidden node $\breve{x}_{ij}$. This allows non matching sub-sequences, e.g., whenever a robot takes a different route that is not part of the database.

The connectivity in the graph is defined by the edge set $\mathcal{E} = \{\mathcal{E}^s, \mathcal{E}^t, \mathcal{E}^a, \mathcal{E}^b, \mathcal{E}^h\}$. The first set $\mathcal{E}^s$ connects the source to a matching node or to a hidden node. It defines that the first query image can be matched with any image in the $\mathcal{D}$, which implies that the robot can start from anywhere in the map. The second set of edges, $\mathcal{E}^t$, models all outgoing edges to the sink node. This models the matching or non-matching of the last query image. The set $\mathcal{E}^a$ establishes connections between matching nodes as well as between hidden nodes. These edges allow us to find sequences of matching images or sequences of unmatched query images respectively. The set $\mathcal{E}^b$ of edges connects hidden and matching nodes. The edges in $\mathcal{E}^b$ are included in the path whenever a robot recognizes a place in the database as the match while traversing a non matching route, see Fig. 2 for an illustration of the edges in $\mathcal{E}^a$ and $\mathcal{E}^b$. Compared with our previous approach, we extend the edge set with $\mathcal{E}^h$ that connect nodes horizontally. In this way we can account for cases in which the robot either stops or takes a shortcut in the database sequence. This reduces the overall number of flows to achieve the same sequence matching performance compared to our previous approach. Fig. 2 depicts the modified graph connections.

Now that we have the flow graph connectivity, we need edge costs to complete the graph model. The cost of reaching every matching node x_{ij} in the graph is $w_{ij} = \frac{1}{s_{ij}}$, where s_{ij} is the corresponding entry in the similarity matrix. We normalize the scores along the direction of query images to eliminate the effect of ambiguous database images. All

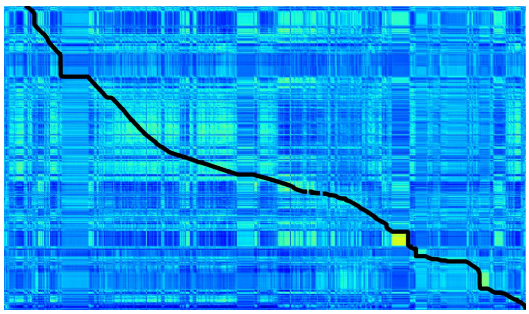


Fig. 3: Our localization result overlaid on the similarity matrix. We achieve localization with high recall on the Pittsburgh dataset.

outgoing edges from $\mathcal{E}^s$, incoming edges to $\mathcal{E}^t$, and edges in $\mathcal{E}^h$ have cost 0. All edges have capacity 1, and K controls the spread of outgoing edges to account for different frame rates and changing speeds of the robot, set as 4 in our approach. Finding the cheapest path through the DAG using topological sorting can be done in $\mathcal{O}(|\mathcal{X}| + |\mathcal{E}|)$. Fig. 3 illustrates the result of our graph matching on the Pittsburgh dataset. With the horizontal edges our path hypothesis can also deal with stops in the database sequence and requires fewer flows than our previous work to achieve the same performance.

C. Least Square Optimization

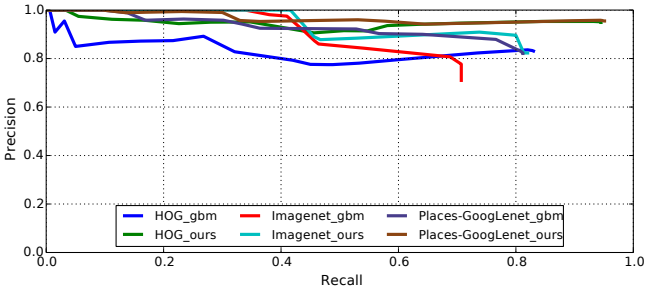
Based on the computed graph connectivity, we can construct a graph-based SLAM optimization problem in which we relate the robot trajectories collected in different seasons to each other. In such a pose graph, we model each pose $\mathbf{x}_i$ along a trajectory with a node. The nodes of consecutive poses in a sequence are connected with an odometry edge which takes the relative transformation according to odometry between $\mathbf{x}_i$ and $\mathbf{x}_j$ as measurement $\mathbf{z}_{ij}$. For two sequences this gives us two unconnected pose graphs. To relate them, we use the sequence matching information as described in the previous section and add loop closure constraints between corresponding poses of the different seasons. We assume that two poses which are connected via a loop closure edge are at the same location and that the relative transformation $\mathbf{z}_{ij}$ is zero. The appearance based image matching in the current state does not provide any relative geometric information. The added loop closure edges allow us to reduce errors in the odometry pose estimates of the individual sequences and compute a joined maximum likelihood trajectory. For that we define an error function for edges $\mathbf{z}_{ij}$ as:

$$\mathbf{e}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{z}_{ij}) = \mathbf{z}_{ij} \ominus (\mathbf{x}_i \ominus \mathbf{x}_j). \tag{2}$$

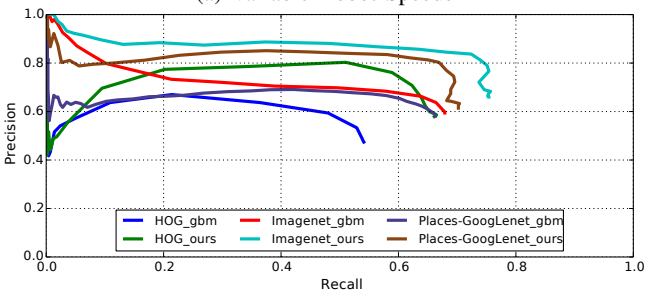
Since we model the uncertainties of spatial loop closure constraints and odometry constraints differently we use the following weighted error function

$$\mathrm{e}_{ij} = \mathbf{e}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{z}_{ij})^\intercal \Omega_{ij} \mathbf{e}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{z}_{ij}), \tag{3}$$

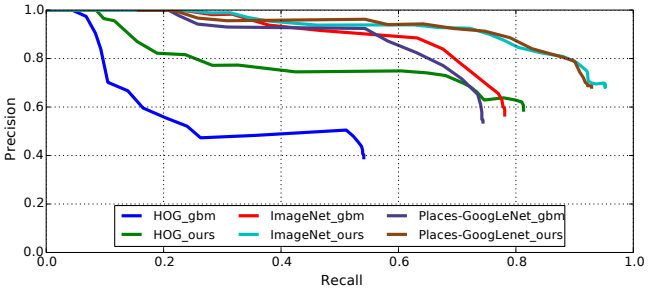
where Ω_{ij} denotes the information matrix for the error. In our current implementation, we down-weight the error for loop closure edges compared to odometry to account for potential spatial offsets between matching images. Based



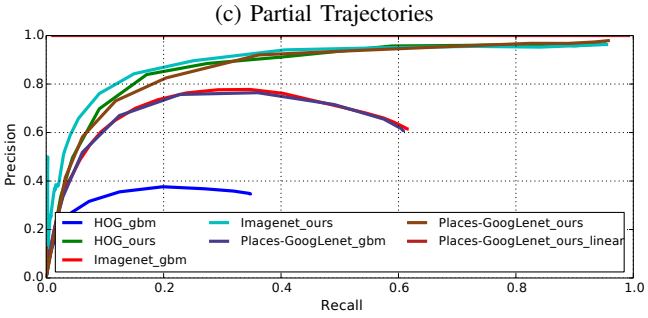
(a) Variable Robot Speeds



(b) VPRiCE Dataset



(c) Partial Trajectories



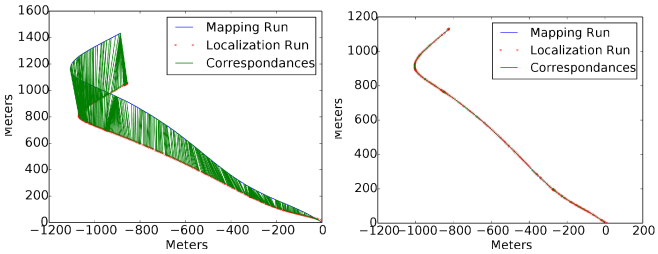
(d) Nordland

Fig. 4: Precision Recall curves for cases with partial trajectories, loops and variable robot speeds, and Nordland dataset using features from DCNN and dense HOG.

on the described error function we construct the following minimization problem

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_{i,j} \mathrm{e}_{ij}. \tag{4}$$

Potential false positives in the sequence matching can have a drastic impact on the solution of Equation 4. To deal with potentially wrong loop closure edges in the constructed graph, we use the Pseudo Huber [11] cost function for loop closure edges. In this way, we ensure a certain robustness in presence of a small number of false positives in the matching result. To efficiently solve the minimization problem, we



(a) Data associations (b) Optimized

Fig. 5: Our approach provides robust data associations across images with large perceptual variations and produces a coherent trajectory over larger periods of time and provides.

employ the g^{2}o framework [14].

IV. Experiments

We carried out an extensive set of experiments to evaluate the performance of our multi season SLAM system. Therefore, we collected two datasets in the same area, where one dataset was recorded during summer and the other one during winter in Freiburg, Germany. The dataset pair consists of a 50 km long trajectory with a total of $\sim$ 38,000 images, recorded in summer and winter of 2012. The odometry information is simulated as the real odometry information was not available for this dataset. We calculated these odometry measurements from the relative movements between GPS positions and added a substantial amount of noise to these relative movements. The GPS information was used in no other place of our experiments.

A. Robust Visual Localization

We carried out four experiments for a quantitative evaluation of the features. We use F1 scores as a measure to compare the overall performances of the features combined with sequential information. These experiments demonstrate the effectiveness of the features from a DCNN which is trained on a suitable database for matching images across seasons. In all figures, 'gbm' and 'ours' corresponds to descriptor based global best match and our approach which adds sequential information for each query respectively. The retrieved locations within ± 3 frames of the query location are considered to be true matches throughout our experiments.

The first experiment (Variable Robot Speed) consists of a image sequence recorded on a trajectory of 1.7 km. The database consists of 322 images while during the localization run, 676 images were recorded. GoogLenet-Places performs better than HOG based on descriptor-based image similarities only. With the sequential information, they perform equally well on this dataset as shown in Fig. 4(a). This shows that although the DCNN features improve the overall performance, the major performance comes from the sequential information in this sequence. The quantitative evaluation is shown in Table I.

The second sequence is the VPRiCE-dataset[1] which consists of 4,022 localization images and 3,756 database images.

[1]The VPRiCE challenge 2015 visual place recognition in changing environments. https: //roboticvision.atlassian.net/wiki/pages/viewpage.action?pageId= 14188617

TABLE I: F1 Scores for all the datasets.

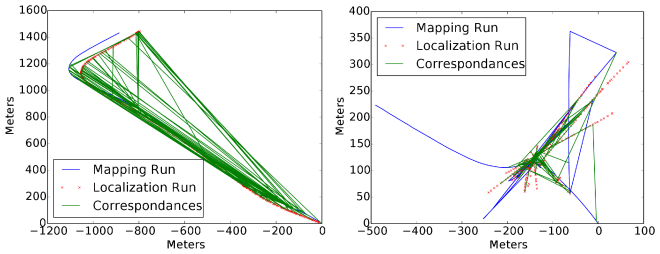
	HOG	ImageNet	Places
Variable Speed	0.95	0.85	**0.95**
VPRiCE	0.66	**0.78**	0.73
Partial Trajectories	0.70	0.84	**0.84**
Nordland	0.96	0.96	**0.97**

It contains images with extreme seasonal, viewpoint and daytime variations. It also contains scenarios where the robot is traveling in the opposite direction, we do not consider those images as groundtruth locations as it is out of the scope of this paper. For this dataset, Imagenet performs best as shown in Fig. 4(b). As this dataset is a mixture of various environment types, appearance and viewpoint differences, it requires further investigation to generalize in which scenarios a DCNN trained on Imagenet performs better.

The third experiment (Partial Trajectories) consists of 781 localization images and 1,328 database images. It contains visits to new places and scenes which are ambiguous for HOG features, whereas the feature representation of DCNN is quite distinctive and produces a lower number of false positives. Places-GoogLenet and Imagenet performs equally well on this dataset as shown in Table I. The fourth experiment was carried out on the publicly available Nordland dataset, which has a 728 km long trajectory. The gain of DCNNs' descriptor based best match localization is evident in Fig. 4(d). However, after adding the sequential information, the difference to HOG is minimal. The peculiar behaviour of the dataset showing low recall and low precision is related to the fact that many images with high similarity are false positives. These images correspond to locations in which the train travels through tunnels. For easier ground truth generation and evaluation, we only remove images corresponding to the train stoppages. In [22], they remove all train stoppages, images in tunnels and for some reason half an hour of the trajectory, whereas we keep all these images for evaluation. This experiment shows that although the gain from descriptor based performance increases, the sequential information has a larger impact on the performance. Furthermore, the assumption of a linear trajectory, as done by most related work, greatly simplifies the data association problem across seasons. To highlight this fact, we also evaluated the same dataset with our approach forcing a linear trajectory, i.e., K=1. While being less general, we achieved 100% precision for recall values of 0.0114 and higher but most robotic real world applications do not follow linear trajectories.

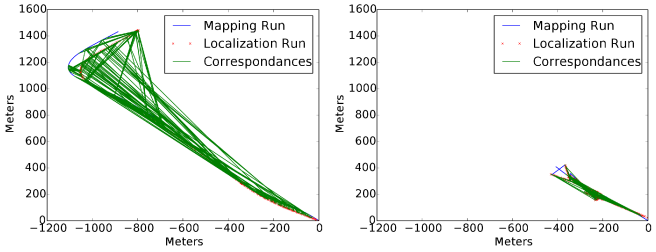
B. SLAM

Next we present the results of our across season SLAM framework. As first experiment, we take the sequence mentioned for Fig. 4(a). We used the odometry and the results of our sequence matching approach to construct a least square optimization problem as described in the previous section and computed a joint maximum likelihood trajectory. The resulting loop closure constraints are shown in Fig. 5(a) before optimization and the optimized result is shown in Fig. 5(b).



(a) Data associations (b) Pseudo Huber kernel result

Fig. 6: FABMAP being a feature-based approach produces considerable false positives in data associations which result in an overall inconsistent trajectory.

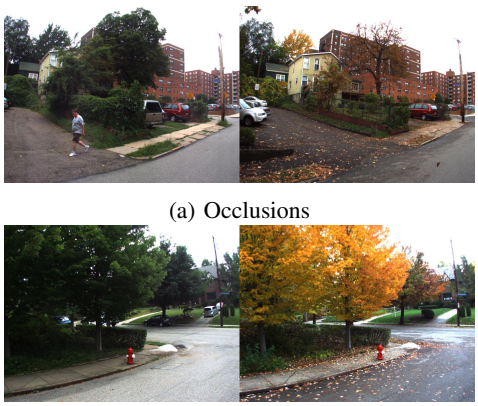


(a) Data associations (b) Pseudo Huber kernel result

Fig. 7: SeqSLAM data associations when used in our pose optimization graph framework degrades the performance because of false correspondences. This is because our datasets are not preprocessed and do not follow a linear trajectory.

We compare our results to state-of-the-art approaches for appearance based localization and mapping. We use the open source implementations of FABMAP [10] and SeqSLAM [23]. Fig. 6 and Fig. 7 show the resulting trajectories by using the image matching results for FABMAP and SeqSLAM in our pose graph optimization scheme. Notice how the false positives lead to poor results in the estimated trajectory. FABMAP fails in this case because it relies on feature based interest points which do not remain stable over seasonal changes. SeqSLAM can handle perceptual changes but assumes well aligned images for both runs, which makes it hard to collect real world data. Therefore, it requires to crop the images to match viewpoints, and then calculate pixel based differences, which performs poorly if applied on uncropped images. Secondly, it only accounts for linear trajectories and cannot deal with different frame rates or robot speeds.

The second experiment was carried out on the Pittsburgh dataset used by Badino *et al.* [2]. This experiment consists of an 8 km long trajectory recorded in July, 2011 and October, 2010. In this dataset, the images were collected with a sideways looking camera. With this setup the camera observes more of the structured environment, which makes image matching easier in urban areas. The localization result using our approach is shown in Fig. 3, where you can see the stoppages in the database as horizontal black paths. Some of the localized images using our approach are shown in Fig. 9 under various perceptual changes like foliage color variations, and occlusions. The data associations for this trajectory are shown in Fig. 8 (a) using our approach. The final merged trajectory after optimization is shown in Fig. 8 (c).

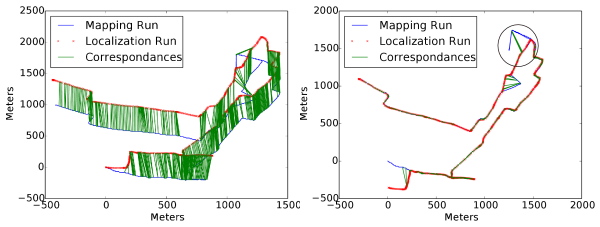


(a) Occlusions

(b) Seasonal foliage changes

Fig. 9: Localized images from the Pittsburgh dataset

For the third experiment, we recorded another trajectory in September, 2014, which is shown in Fig. 10, and take the corresponding part of the dataset from winter 2012 as the mapping run. It includes several non matching routes in the trajectory. The data associations for this trajectory are shown in Fig. 10 (a) using our approach. The final merged trajectory after optimization is shown in Fig. 10 (b).



(a) Data Associations (b) Pseudo Huber kernel result

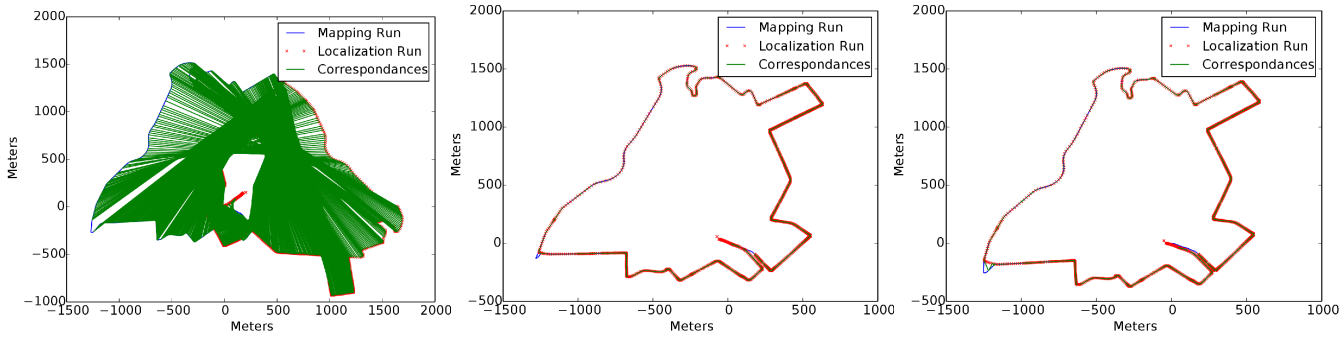
Fig. 10: We achieve consistent trajectories using our framework in this 5 km long trajectory. The encircled region in (b) shows a different path being taken in the two trajectories. There are very few false correspondences and as the error for odometry edges are weighted higher, it accounts for the spatial offset between these false matches.

C. Runtime Comparisons

In this section, we compare the runtime of our previous work [20] and the proposed approach for matching image descriptors. Previously, we extracted and matched HOG features on CPU where matching features consumed a huge amount of time, and proved to be a bottleneck. Therefore, we use a GPU based implementation for both GoogLenet and Alexnet and implemented image descriptor matching on GPU. It resulted in real-time image matching even for large datasets up to 48,000 images at 15 frames per second as shown in Table II.

V. Conclusions

In this paper, we presented a novel appearance-based visual SLAM approach that successfully detects loop closures in datasets recorded in different seasons. Given two



(a) Data Associations (b) Pose-graph optimization (c) Pseudo Huber kernel result

Fig. 8: We achieve a consistent trajectory using our Visual SLAM framework over the 8 km long Pittsburgh dataset.

Processing Unit	Feature Extraction per Image [ms]	Matching per Image Pair [ms]
GPU	**12**	**0.0014**
CPU	31	33

TABLE II: Feature processing times of CPU and GPU.

image sequences, our method extracts global image features from DCNNs on all images and computes a full similarity matrix. Based on this matrix we formulate a flow network problem and compute a matching sequence hypothesis. We use the matching sequence as loop closure information and formulate a graph-based SLAM problem including odometry information and solve it with a standard least square optimization framework to compute a joint trajectory estimate. Through extensive experiments on several challenging datasets we evaluated our method and provided a comparison with the open source implementations of two state-of-the-art appearance-based image matching approaches. Our approach outperforms them in terms of trajectory consistency on the evaluated datasets.

REFERENCES

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network flows: theory, algorithms, and applications*. Prentice hall, 1993.

[2] H. Badino, D. Huber, and T. Kanade, "Real-time topometric localization," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012.

[3] 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.

[4] Z. Chen, O. Lam, A. Jacobson, and M. Milford, "Convolutional neural network-based place recognition," *CoRR*, vol. abs/1411.1509, 2014.

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

[7] ——, "Fab-map: Appearance-based place recognition and mapping using a learned visual vocabulary model." in *Int. Conf. on Machine Learning (ICML)*, 2010.

[8] 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.

[9] 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.

[10] A. J. Glover, W. P. 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 Int. Conf. on Robitics and Automation (ICRA)*, 2012.

[11] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[12] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, 2012, pp. 1097–1105.

[14] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011.

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

[16] M. Milford, E. Vig, W. J. Scheirer, and D. D. Cox, "Vision-based simultaneous localization and mapping in changing outdoor environments," *Journal of Field Robotics (JFR)*, vol. 31, September 2014.

[17] 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.

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

[19] M. J. Milford and G. F. Wyeth, "Mapping a suburb with a single camera using a biologically inspired slam system," *Robotics, IEEE Transactions on*, vol. 24, no. 5, pp. 1038–1053, 2008.

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

[21] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *arXiv preprint arXiv:1409.0575*, 2014.

[22] N. Sünderhauf, F. Dayoub, S. Shirazi, B. Upcroft, and M. Milford, "On the performance of convnet features for place recognition," *arXiv preprint arXiv:1501.04158*, 2015.

[23] N. Sunderhauf, P. Neubert, and P. Protzel, "Are we there yet? challenging seqslam on a 3000 km journey across all four seasons," in *Proc. of the ICRA Workshop on Long-Term Autonomy*, 2013.

[24] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *arXiv preprint arXiv:1409.4842*.

[25] 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.

[26] 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 and Automation (ICRA)*, 2015.

[27] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Advances in Neural Information Processing Systems*, 2014, pp. 487–495.