Exploiting intra Database Similarities for Selection of Place Recognition Candidates in Changing Environments

Peer Neubert*
TU Chemnitz

Stefan Schubert*
TU Chemnitz

Peter Protzel
TU Chemnitz

firstname.lastname@etit.tu-chemnitz.de

Abstract

Visual place recognition is an important field in mobile robotics. A typical setup contains a given database of images (the map) and the robot consecutively tries to find matchings between this database and the actual visual input (query images). For real world applications the runtime of this setup can be significantly reduced by selecting only few matching candidates from the database for the current query image. In this paper, we propose a novel approach to reduce the computational effort in the online stage of place recognition by processing the database in advance. The method can be used with any matching scheme if the following assumption on the dataset is fulfilled: Given database and query in form of image sequences, we assume directly neighbouring places in the world to be represented by images sharing similar appearance. Based on this assumption, we propose a selection of matching candidates for the next query image based on the current matchings and intra database similarities. We provide an algorithmic approach that can deal with different camera trajectories between database and query, repeatedly revisited places and kidnapped robot situations. Further, we provide preliminary results on the Nordland validation dataset and show that the proposed approach can save in many configurations about 90 % comparisons at the online stage without considerable place recognition performance loss. Finally, we discuss the intuitive parameters of the approach and evaluate them experimentally.

1. Introduction

Visual place recognition describes the problem of matching images (or image sequences) recorded at the same place at different times. Accordingly, the revisited places can look differently due to changing lighting conditions, changing weather conditions or seasonal appearance changes. Recognizing these places despite their different appearances is an important requirement, e.g. for factor graph based Simultaneous Localization and Mapping (SLAM) in such environments. It is a challenging and active research field for researchers on both mobile robotics and computer vision.

In addition to the challenges induced by the changing appearances, the generic problems of place recognition apply - i.e. the growing runtime with increasing number of seen places. A naive approach is to evaluate each possible combination between query (the current image) and database images. This results in completely filled similarity (or distance) matrices as can be seen in the left part of Fig. 1. Moreover, the larger the database and the more diverse the contained places, the higher is the fraction of possibly avoidable comparisons between images that show completely different places. We propose a novel method to reduce the number of comparisons during runtime to a subset of promising candidates. An example for a resulting sparse comparison matrix is shown in the right part of Fig. 1.

The underlying idea is to divide the place recognition task and its computational efforts into two phases:

1. In a first *offline* step, the database of known places can be analyzed and preprocessed.
2. In the time critical *online* phase where incoming query images are going to be matched to the preprocessed database, we can now exploit the results of the preprocessing.

In this paper, we propose a novel approach for such an offline preprocessing that can significantly reduce the number of required image comparisons in the online phase and approximately maintains the place recognition performance obtained by the exhaustive approach. Our method can be used with any matching scheme, if the following assumption on the dataset is fulfilled: Given database and query in form of image sequences, we assume directly neighbouring places in the world to be represented by images sharing similar appearance. Thus, consecutive images should be similar and different images from multiple visits of the

[*] Both authors contributed equally to this work.

same place should be similar *inside* the database. We want to emphasize that they need not be similar across changing environmental conditions. E.g. given a database of summer images and query images from winter, we expect the summer images of nearby places to be similar to each other.

The proposed approach can be summarized as follows: In an offline phase, we compare the whole dataset with itself to determine similar places within the database. This information can then be used in the online phase to select promising matching candidates. Thus, we select the matching candidates for an incoming query image Q_t based on successful matchings of the previous query image Q_{t-1} to the database. Therefore, we select the m database images that matched best to Q_{t-1} and additionally all other database images with an appearance distance $\leq dist_{max}$ to one of these database images. The parameter m corresponds to the number of simultaneously kept hypotheses of matchings to the database. Subsequently, Q_t is compared to the set of matching candidates. This corresponds to a sparse row vector in the right comparison matrix of Fig. 1. The m best matching database images for Q_t are then the base for the candidate set for the next query image Q_{t+1}.

To cope with situations where none of the m hypotheses yield a successful matching (i.e. if the prerequisite on the dataset is violated or in a kidnapped robot case), we add a relocalization step: If the appearance distance between the query image and each matching candidates exceeds a threshold $dist_{QDB,max}$, we compare Q_t to all database images. This adds a completely filled row vector to the comparison matrix. From this row, we can then select the m best matchings as basis for the matching candidate set.

After related work in the following Sec. 2, we will give more details on the proposed approach in Sec. 3. Sec. 4 presents preliminary results on the Nordland validation dataset [12]. We show that the proposed approach can save in many configurations about 90 % comparisons at the online stage without considerable place recognition performance loss. The final Sec. 5 discusses directions for future work, in particular in the context of changing environments.

2. Related Work

There are at least three approaches on dealing with changing environments: 1. We can try to find descriptors that can match places despite their appearance change. 2. One can try to learn systematic changes, e.g. by predicting changes [12] or by learning co-occurrences of features [8]. 3. Alternatively, we can accept the fact that we cannot match all appearances and try to organize them [3, 4]. Each approach may benefit from the proposed method.

Our experiments are based on images of the Nordland dataset. To decide whether two images show the same place or not, local keypoint features like SIFT or SURF can be used. For example, the bag of words based FAB-MAP [5] showed good performance on place recognition in large environments. However, the repeatability of local keypoint features across severe appearance changes, e.g. induced by seasonal changes, is limited [17]. I.e. results in [12] reveals that FAB-MAP has severe problems in the presence of seasonal changes.

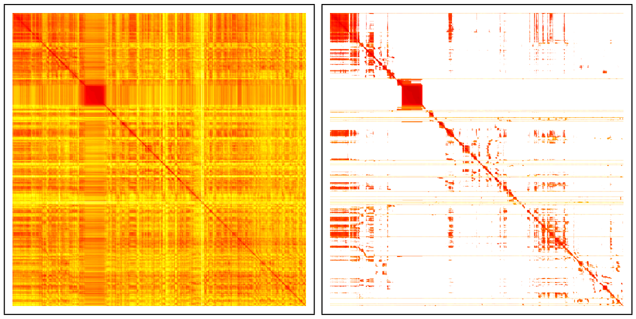

Figure 1. Two distance matrices after place recognition on the Nordland validation dataset [12] with seasonal appearance changes (here: spring-winter). Each row represents a query image in winter and a column represents a database image in spring. The left matrix is calculated by a naive approach comparing all possible query-database combinations. The right image depicts the matrix using the proposed approach. The white space represents image pairs which are not compared, thin horizontal lines indicate a relocalization. More exactly, the sparse matrix is filled with less than 9% of the possible comparisons, however the corresponding place recognition performance is almost identical (see Fig. 4 (top)). The parameters (see Sec. 3) are $m = 20$, $dist_{max} = 0.25$ and $dist_{QDB,max} = 0.5$.

To overcome these limitations, holistic image descriptors are often used for place recognition in changing environments. Holistic image descriptors are for instance Gabor-Gist [14], BRIEF-Gist [16] and WI-SURF [1]. In [13] the sum of absolute differences across corresponding pixels for several image scales is used as a holistic descriptor. Recently, Sünderhauf et al. [15] investigated the performance of a convolutional neural network (CNN) [7] for place recognition in changing environments. They showed that higher layers are better suited for view changes whereas lower layers show better performance in the presence of appearance changes. For our experiments, we decided to also use pretrained CNN features presented in [2].

In order to improve the performance of place recognition, various sequence based approaches have been developed. SeqSLAM [10] searches for partially linear sequences inside the query-database similarity matrix. In [9] the computational efficiency of SeqSLAM could be improved by applying a particle filter. These algorithms make assumptions on the similarity of camera motions across multiple visits of corresponding places. Our proposed approach relaxes this assumption to similar appearances of neighbouring places, independently from any motion constraints.

An existing approach for selection of matching candidates is based on an idea for finding more reliable matches presented in [6]. They suggest the usage of a weighted graph for GraphSLAM - its edges are weighted with log likelihoods. The branch with the lowest sum of likelihoods is further expanded. In [11] this idea was adopted for an offline network flow approach. However, its limitation is still the necessity for a completely computed query-database similarity matrix. In order to improve the efficiency Vysotska et al. [18] extended this approach by using a GPS prior. As a result, they could reduce the number of required image comparisons between query and database. By using a CNN for image comparison and a modified graph search algorithm they could reduce the number of required comparisons without additional external sensors (i.e. without GPS) [19].

3. Reducing the Number of Required Image Comparisons

Currently, most place recognition approaches are using the naive method of matching every incoming query image with all existing database images with the consequence of high computational effort. Accordingly, each query image is matched with each database image, although they might be very different. In this section we first provide a method for detecting similarities inside the database (Sec. 3.1), followed by a description how to use these similarities to avoid unnecessary image matchings between query and database images (Sec. 3.2). Finally, we propose a simple method to relocalize in case of position loss (Sec. 3.3). The experimental setup and corresponding results are then presented in the following Sec. 4.

3.1. The Offline Step: Finding similar places inside the database

In order to avoid unnecessary comparisons at the online stage, we suggest a preparation step computing a similarity matrix

$$\mathbf{D}_{DB\times DB} \in \mathbb{R}^{|DB|\times|DB|} \tag{1}$$

containing all combinations of image comparisons within the database. The computations can be done in advance, furthermore it is sufficient to calculate a triangle matrix since $\mathbf{D}_{DB\times DB}$ is a symmetric matrix. The comparison method can be the same as for comparisons between query and database. However, since the variance of the environmental conditions in the set of database images may be much smaller. Therefore, simpler, faster, and possibly more robust matching strategies may be used. An example matrix computed from our dataset is depicted in Fig. 3.

3.2. The Online Step: Comparing incoming query images with a reasonable subset of database images

We propose the method described in Algorithm 1 (see below) in order to receive a subset of database images for each time step during the place recognition phase by using the comparison matrix $\mathbf{D}_{DB\times DB}$ presented in the previous Sec. 3.1. The method is based on the assumption, that consecutive query images belong to places that share similar appearance in the corresponding database images.

The steps of our proposed algorithm are illustrated in Fig. 2. This figure shows six steps, each with the upper matrix $\mathbf{D}_{DB\times DB}$ and the lower matrix $\mathbf{D}_{Q\times DB}$. Since $\mathbf{D}_{DB\times DB}$ is computed in advance, it is completely filled. In contrast, the lower matrix $\mathbf{D}_{Q\times DB}$ starts from a single row corresponding to a first query image, further rows are appended when subsequent query images are processed.

Fig. 2a): A first query image Q_t has unknown corresponding images inside the database DB. Hence, we start a relocalization by comparing Q_t with all images inside DB. The algorithm performs $|DB|$ comparisons which results in a complete row in the lower matrix. The key idea of the proposed approach is to select just a small number of highly similar places as basis for matching candidates for the next query image. These selected places are shown in red in the lower matrix of Fig. 2a). In order to work with multiple hypotheses for different appearances of the query place in the database we select $m \in \mathbb{N}$ as a branching factor to proceed with the best m hypotheses DB_m.

Fig. 2b) and c): The m best hypotheses DB_m are used to search inside $\mathbf{D}_{DB\times DB}$ for the most similar database images DB_s. DB_{s_j} is similar to DB_{m_j} if their distance is $\leq dist_{max} \in \mathbb{R}$. The set DB_s is shown in blue.

Fig. 2d) and e): The set of similar database images DB_s (blue) is the set of matching candidates for the newly incoming query image Q_{t+1}. If the above assumption on the similarity of the images of neighbouring places holds, than the true database matching for the next query image is amongst the images similar to matchings of the previous query image and thus in the candidate set DB_s. The query image is now matched with all candidate images in DB_s. This can be seen as new row in $\mathbf{D}_{Q\times DB}$. Since $|DB_s| < |DB|$, the number of comparisons and thus the computational effort is reduced.

Fig. 2f): Similar to Fig. 2b)) the matching of the next incoming query image is prepared. The m best hypotheses DB_m are used to search inside $\mathbf{D}_{DB\times DB}$ for the most similar database images DB_s. The algorithm is repeated with every incoming Q_{t+i}.

3.3. Relocalization

The algorithm described in Sec. 3.2 assumes that database images DB_{j+t} neighbouring to DB_j are similar

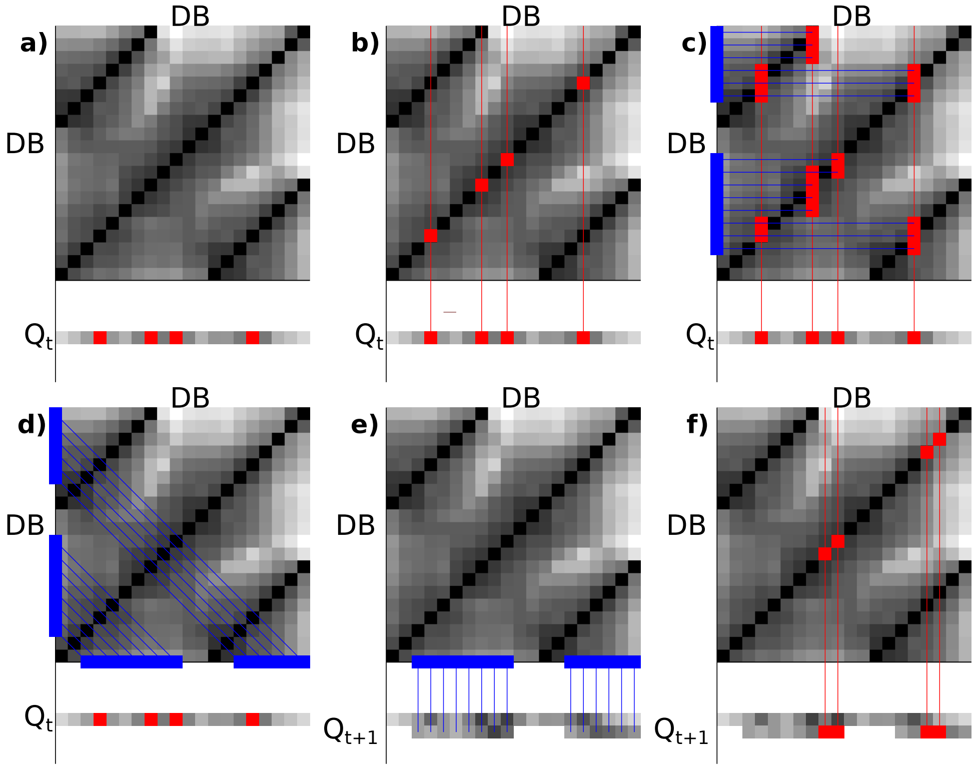

Figure 2. Illustration of our approach for reducing the number of necessary comparisons. Every step is represented by two comparison matrices $\mathbf{D}_{DB\times DB}$ and $\mathbf{D}_{Q\times DB}$. See Sec. 3.2 for details.

enough, i.e. their distance is less than $dist_{max}$. This assumption may be violated if the above assumption on the dataset is violated or in a kidnapped robot case. Fig. 3(left) shows the similarity matrix $\mathbf{D}_{DB\times DB}$ from our experimental setup. The salient yellow lines indicate some unique places inside the database which can lead to position loss.

Therefore, we introduce a third parameter $dist_{QDB,max} \in \mathbb{R}$. If the matching distance between Q_t and all DB_{m_j} is more than $dist_{QDB,max}$, a relocalization is carried out. Thus, Q_t is matched to all database images DB in order to find the m best matches representing better recognized places. In our experimental setup (Fig. 1, right), the relocalization can be seen as thin horizontal lines.

4. Experiments

4.1. Dataset

We use the Nordland validation dataset from [12] for our experiments. The dataset shows images of a train ride in Norway from all four seasons. However, for first experiments we use only the spring-winter configuration. Spring images are used as database images DB, i.e. these locations can be seen as previously visited places. Accordingly, winter images are used as query images Q which have to be matched to places in the database. The validation dataset contains 30 minutes of the spring and winter rides. The dataset is sampled with a frame rate of one frame per second resulting in 1800 frames per season. Since the image sequences are time synchronized, the ground truth comparison matrix is a diagonal matrix with additional blocks at the diagonal for situations where the train stopped. The image sequences at a frame rate of 1 FPS fulfil the assumption of consecutive images to share similar appearances.

4.2. Image comparison

Each image comparison is done with a holistic image descriptor and a subsequent cosine distance calculation. The holistic image descriptor is the vectorized output of the third convolutional layer (*conv3*) of the VGG-M network [2]. The database-database comparison (see Sec. 3) is also carried out with the CNN descriptor, although there is no need for a descriptor applicable for severe seasonal appearance changes. Rather, methods using scale invariant features could be applied (e.g. FAB-MAP [5]) in order to get image comparisons more robust to e.g. viewpoint changes. An example for a intra database similarity matrix is shown in Fig. 3. This corresponds quite well to the underlying ground truth.

```
Data: $Q$, $DB$
$DB_{candidate} = \emptyset$;
for $\forall q \in Q$ do
    if $DB_{candidate} == \emptyset$ then
        $DB_{candidate} = get\_m\_best\_db(dist(q, DB))$;
    else
        $DB_{query} = \emptyset$;
        for $\forall db_{candidate} \in DB_{candidate}$ do
            for $\forall db \in DB$ do
                if $dist(db_{candidate}, db) < dist_{max}$
                then
                    $DB_{query} = DB_{query} \bigcup \{db\}$;
                end
            end
        end
        $DB_{all\_candidate} =$
        $get\_m\_best\_db_{query}(dist(q, DB_{query}))$;
        $DB_{candidate} = \emptyset$;
        for $\forall db_{all\_candidate} \in DB_{all\_candidate}$ do
            if
            $dist(q, db_{all\_candidate}) < dist\_query_{max}$
            then
                $DB_{candidate} =$
                $DB_{candidate} \bigcup \{db_{all\_candidate}\}$;
            end
        end
    end
end
```

Algorithm 1: Procedure of our proposed approach.

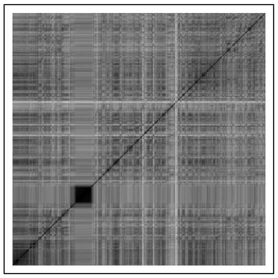

Figure 3. Symmetric comparison matrix $\mathbf{D}_{DB \times DB}$ between database images in spring.

4.3. Results

Fig. 4 (top) shows precision-recall curves for the experimental setup of one frame per second. To create the different curves, we varied the parameter $dist_{max} = 0.05 \ldots 0.4$. This parameter controls the range of similar database images that are included in the candidates set for each of the m hypotheses from the previous query image matching. In our experiments, the influence of the parameter m was negligible and it was fixed to $m = 20$. This might be different for other datasets with larger visual ambiguity. However, $m = 20$ enables to maintain 20 different corresponding appearances of a single query image in the database, which should be sufficient for many datasets. The results also showed to be insensitive to changes of the third parameter $dist_{QDB,max}$ that toggles the relocalization behaviour (curves not shown). For the shown curves, the parameter was fixed to $dist_{QDB,max} = 0.5$.

The precision-recall curves of our approach converge to the curve corresponding to the exhaustive comparison of the naive approach with increasing $dist_{max}$, since the parameter controls the threshold whether a database image is similar to another database image or not. Accordingly, the number of similar images increases with the distance. As a result, the number of necessary comparisons rises, which can be seen in the legend of the diagram. For instance, a result similar to the naive approach can be received with $dist_{max} = 0.25$ with a remaining number of comparisons of approximately 8.8%.

In Section 3 we emphasize that our approach requires similarity between consecutive images. In order to investigate the influence of a decreasing similarity, we reduce the frame rate from $1fps$ to $\frac{1}{2}fps$ and $\frac{1}{3}fps$ for query and dataset image sequences. Resulting curves can be seen in the middle and bottom parts of Fig. 4. The parameter $dist_{max}$ is varied identical for all three frame rates. For corresponding parameter settings, a decreasing place recognition performance can be observed for decreasing frame rate, since the similarity between consecutive images also decreases with lower frame rates. For instance, the above mentioned parameter set with $dist_{max} = 0.25$ results in lower place recognition performance with decreasing frame rates. Simultaneously, the number of comparisons rises, because the amount of relocalizations is higher. As a further consequence of the relocalization, the lowest three curves of $\frac{1}{2}fps$ are improved for $\frac{1}{3}fps$, since bad place recognitions cause a relocalization more often, improving the place recognition performance but increasing the number of comparisons.

The influence of $dist_{max}$ and frame rate is depicted more clearly in Fig. 5. The upper diagram shows the F_1 score as a function of $dist_{max}$ and the second diagram shows the number of comparisons as a function of $dist_{max}$. A higher $dist_{max}$ results in a higher F_1 score, however the corresponding number of comparisons increases. Furthermore, if $dist_{max}$ is constant, the number of comparisons is higher if the frame rate is lower. For the frame rate of $\frac{1}{3}fps$ the relocalization effect can be seen again as a better F_1 score in the left part of the upper diagram.

5. Discussion and Future Work

We presented a technique for place recognition in changing environments reducing the number of required image comparisons between an incoming query image and the

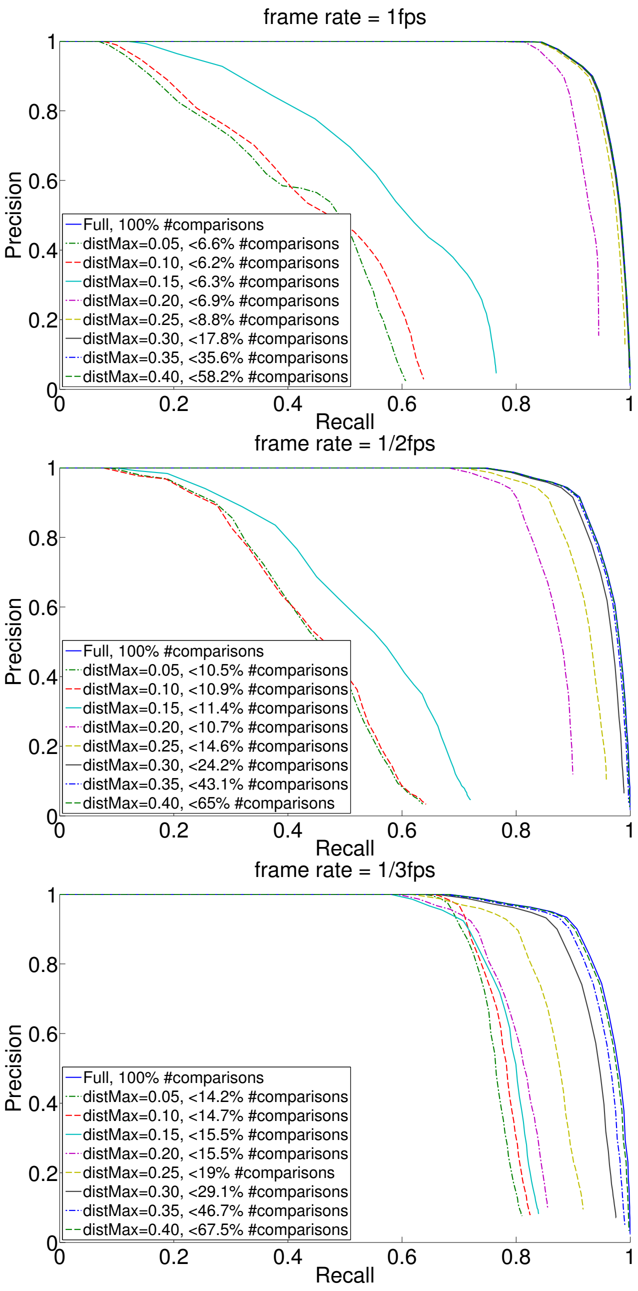

Figure 4. Precision-recall curves at three different frame rates with the constant parameters $m = 20$, $dist_{QDB,max} = 0.5$, and different values for $dist_{max}$.

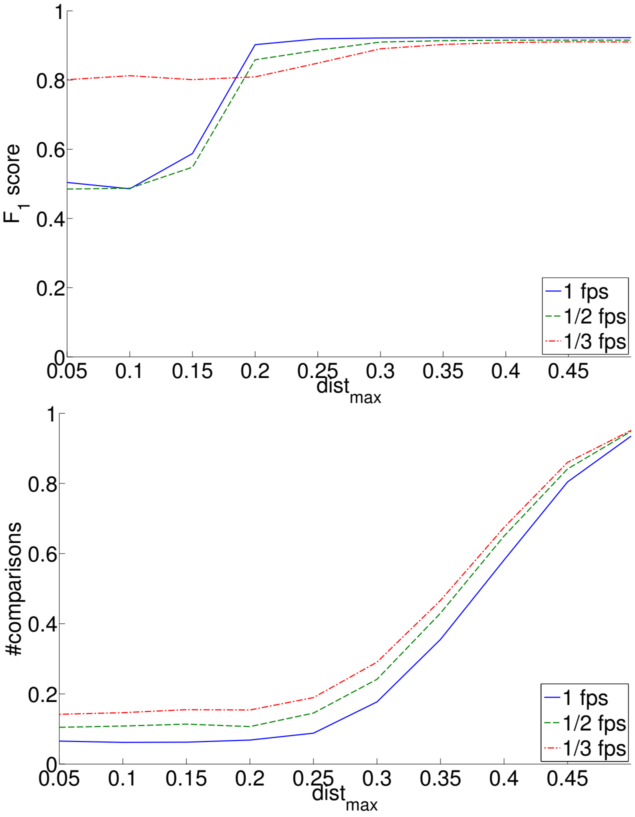

Figure 5. Illustration of the influence of $dist_{max}$ to the F_1 score and the number of comparisons, respectively, at three different frame rates.

database. Our approach shifts the major computational time into an offline phase. In this offline phase, similarities between all database images are computed. Subsequently, in an online phase these similarities can be used to choose only a subset of database images which have to be applied for comparisons with the next incoming query image.

Our preliminary results in Sec. 4 showed that for suitable parameter choices the place recognition performance is similar to a naive approach comparing all possible query-database image combinations. Simultaneously, the absolute number of comparisons was lower than 9% in our example.

In order to get multiple hypotheses for the current location, we introduced a parameter m to keep m hypotheses. Especially, if a dataset contains multiple loops or partially similar places at different locations, the parameter is required.

Furthermore, a parameter $dist_{max}$ is required which represents the threshold to decide whether a database image is similar to another database image or not. We figured out (e.g. in Fig. 5) $dist_{max}$ is a sensitive parameter determining whether a good place recognition performance is received.

As a last parameter we introduced $dist_{QDB,max}$ which is a threshold deciding whether a query image assigned to a database image really belongs to it. If this distance is exceeded, a relocalization is conducted. A suitable choice of $dist_{QDB,max}$ avoids position loss in the online phase.

Additionally, we investigated the influence of decreasing frame rates, since our algorithm has the requirement that adjacent query images are similar due to the similarity of their corresponding database images. It could be seen that the place recognition performance decreases for constant parameters with lower frame rate. However, by adjusting the parameters and by applying the relocalization more often, the performance could be re-established with an increasing number of comparisons.

In the particular case of recognizing places between changing environments, the proposed approach has some interesting properties if we consider the usage of different matching algorithms for the different types of comparisons: Assume we have a database of summer images and query images are from winter. We reduce the number of possibly expensive summer-winter comparisons by performing "cheap" summer-summer comparisons. Even in setups where we do not have a given database, this may be worthwhile. Moreover, we can e.g. use a matching scheme robust to seasonal changes for summer-winter comparisons and another that is robust to viewpoint changes for summer-summer comparisons. On the one hand this may help to relax our assumption on the intra sequence similarities. On the other hand this may even contribute to improve the overall performance by reducing the visual ambiguity of winter images by selecting only matching candidates that are chosen based on the summer data.

For our first experiments, we used only the spring-winter combination of the Nordland validation dataset. However, for verification of our approach we intend to run further experiments with datasets containing multiple loops and view changes.

Moreover, it is an interesting direction for future work to investigate whether not only the runtime but the place recognition performance can also benefit from this approach. Similar to how SeqSLAM exploits sequences in time to improve place recognition performance, the proposed approach could use "appearance sequences". These are sequences of consecutively selected matching candidates, that would appear as connected sequences in the similarity matrix. Since this matrix is sparse, one might obtain these sequences directly without further assumptions like constant velocity.

References

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

[2] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference*, 2014.

[3] W. Churchill and P. Newman. Practice makes perfect? managing and leveraging visual experiences for lifelong navigation. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 4525–4532, May 2012.

[4] W. Churchill and P. Newman. Experience-based navigation for longterm localisation. *Int. Journal of Robotics Research*, 2013.

[5] M. Cummins and P. Newman. Appearance-only slam at large scale with fab-map 2.0. *Int. Journal of Robotics Research*, 30(9):1100–1123, Aug. 2011.

[6] D. Hähnel, W. Burgard, B. Wegbreit, and S. Thrun. Towards lazy data association in slam. In *In Proc. of the Int. Symposium of Robotics Research (ISSR)*, 2003.

[7] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, MM '14, pages 675–678, New York, NY, USA, 2014. ACM.

[8] E. Johns and G.-Z. Yang. Feature co-occurrence maps: Appearance-based localisation throughout the day. In *ICRA*, pages 3212–3218. IEEE, 2013.

[9] Y. Liu and H. Zhang. Towards improving the efficiency of sequence-based slam. In *Mechatronics and Automation (ICMA), 2013 IEEE International Conference on*, pages 1261–1266, Aug 2013.

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

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

[12] P. Neubert, N. Sünderhauf, and P. Protzel. Superpixel-based appearance change prediction for long-term navigation across seasons. *Robotics and Autonomous Systems*, 69(0):15 – 27, 2015.

[13] E. Pepperell, P. Corke, and M. Milford. Towards vision-based pose- and condition-invariant place recognition along routes. In *In Proceedings of the Australasian Conference on Robotics and Automation 2014*, 2014.

[14] G. Singh. Visual loop closing using gist descriptors in manhattan world. In *in Omnidirectional Robot Vision workshop, held with IEEE ICRA*, 2010.

[15] N. Sünderhauf, F. Dayoub, S. Shirazi, B. Upcroft, and M. Milford. On the performance of convnet features for place recognition. *CoRR*, abs/1501.04158, 2015.

[16] N. Sünderhauf and P. Protzel. Brief-gist - closing the loop by simple means. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 1234–1241, Sept 2011.

[17] C. Valgren and A. J. Lilienthal. SIFT, SURF & seasons: Appearance-based long-term localization in outdoor environments. *Robotics and Autonomous Systems*, 58(2):149–156, Feb. 2010.

[18] 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 *In Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2015.

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