2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

Beyond ANN: Exploiting Structural Knowledge for Efficient Place Recognition

Stefan Schubert, Peer Neubert and Peter Protzel

Abstract— **Visual place recognition is the task of recognizing same places of query images in a set of database images. It is important for loop closure detection in SLAM and candidate selection for global localization. Many approaches in the literature perform computationally inefficient full image comparisons between queries and *all* database images. There is still a lack of suited methods for efficient place recognition that allow a fast, sparse comparison of only the most promising image pairs without any loss in performance. While this is partially given by approximate nearest neighbor (ANN) based methods, they trade speed for precision and additional memory consumption, and many cannot find arbitrary numbers of matching database images in case of loops in the database. In this paper, we propose a novel fast sequence-based method for efficient place recognition that can be applied online. It uses relocalization to recover from sequence losses, and exploits usually available but often unused intra-database similarities for a potential detection of *all* matching database images for each query in case of loops or stops in the database. We performed extensive experimental evaluations over five datasets and 21 sequence combinations, and show that our method outperforms two state-of-the-art approaches and even full image comparisons in many cases, while providing a good tradeoff between performance and percentage of evaluated image pairs. Code is available[1].**

I. INTRODUCTION

Visual place recognition is the task of recognizing places of query images Q in a set of database images DB, despite potential environmental condition changes due to time of day, weather or seasonal changes. It is required for loop closure detection in SLAM or candidate selection for global localization. Visual place recognition can be considered as an embedding of pure image retrieval into mobile robotics. This allows the exploitation of additional structural knowledge like spatio-temporal sequences in the database and query set.

Many approaches for place recognition perform a full pairwise comparison between *all* images of database and query, which is not only computationally expensive but potentially error-prone due to the risk of matching similar looking but different places. To circumvent the computational costs, some approaches use approximate nearest neighbor search (ANN) to find promising candidates. However, ANN-based approaches usually trade computation speed for precision, require additional memory for indexing, and do potentially not find multiple or even all matching database images for a single query image in case of loops or stops in the database. Finding multiple matches can be important for example in pose graph SLAM to match a query image not only to the most similar database image but also to another less similar database image with higher pose accuracy.

In this work, we propose a new method for efficient place recognition. It exploits sequences in database and query to compare only a small fraction of images between database and query, and avoids the usage of ANN-based methods. In addition, it builds upon yet often unused intra-database similarities S^{DB} for an exploitation of additional inherent structural knowledge without a need for additional sensors. Intra-database similarities contain information about loops and stops in the database. In summary, our proposed method

- can be used online
- is fast
- tremendously reduces the number of image comparisons
- preserves or even increases the performance of full image comparisons
- handles loops and stops in the database to find multiple matches for each query
- can cope with exploration in the query (no matching database images), when other sequence-based methods potentially fail
- auto-tunes two thresholds; this allows an application over many datasets with a single set of parameters

In the following Sec. II, we give an overview over related work. Next in Sec. III, we describe our proposed algorithm for efficient place recognition as well as a method for automatic threshold tuning and describe two relocalization strategies to recover from sequence loss. Subsequently in Sec. IV, we experimentally evaluate our method, compare it to two state-of-the-art methods for sequence-based and ANN-based efficient place recognition, show the benefit of using intra-database similarities, and consider the problem of exploration during the query run and the benefit of relocalization. Finally, we conclude our work in Sec. V.

II. RELATED WORK

Visual place recognition in changing environments is subject of active research. A more detailed introduction and an overview over existing methods is given in a survey from 2016 [1]. State-of-the-art methods for place recognition build upon deep learning image descriptors. Sünderhauf et al. [2] showed that the intermediate layers of CNNs trained on an image classification task can serve as a good holistic image descriptor for place recognition. They identified the conv3-layer of AlexNet [3] to be robust against condition and viewpoint changes. NetVLAD [4] is a holistic image

This work was supported by the German Federal Ministry for Economic Affairs and Energy.

All authors are with Faculty of Electrical Engineering and Automation Technology, Chemnitz University of Technology, Chemnitz, Germany {firstname.lastname}@etit.tu-chemnitz.de

[1] Code: https://www.tu-chemnitz.de/etit/proaut/prstructure

978-1-7281-9077-8/21/$31.00 ©2021 IEEE

descriptor that was specifically designed and trained for place recognition. The performance of such holistic descriptors can be further improved by using feature standardization [5]. Different to holistic image descriptors, methods like DELF [6] and D2-Net [7] extract local image descriptors for place recognition and localization. However, some local methods like D2-Net use holistic descriptors for an initial candidate selection due to computational efficiency [7].

For place recognition, many methods (e.g., [8][9]) perform a full pairwise comparison between all database and query descriptors. However, this approach is computationally inefficient and comes along with a higher risk of comparing similar looking but different places. There are at least three classes of methods in place recognition that can be used to avoid full image comparisons: sequence-based methods, methods that use approximate nearest neighbor search (ANN), and trained place classifiers.

Most sequence-based methods like SeqSLAM [8], HMM [10] or MCN [11] are designed to improve the place recognition performance, but are not intended to guide the image comparison for efficiency. For efficient, sparse image comparisons, Vysotska and Stachniss [12] proposed the sequence-based method Online Place Recognition (OPR) that performs a graph-based sequence search. Further, they investigated the incorporation of noisy GPS as location priors to handle loops in the database. Our approach builds upon sequences for efficiency as well, but performs a more greedy trajectory search for faster computation, while OPR (inefficiently) expands graph nodes iteratively. Moreover, OPR suffers from exploratory phases during the query run as illustrated in Fig. 5. To circumvent this problem, a relocalization procedure is integrated in our method. Moreover, we will show that we can handle loops even without GPS by using inherent, often available intra-database similarities S^{DB}, which were used as well in [13][14] for performance improvements.

ANN-based approaches for efficient place recognition trade computational speed for precision and additional memory consumption, and cannot find arbitrary numbers of matching database images. A recent survey [15] provides a comprehensive comparison of ANN methods on different tasks with an analysis of precision losses. They identified HNSW [16] to be particularly suited for high-dimensional data. FLANN [17] obtained low performance for place recognition in [12]. MILD [18] is an ANN-based method specifically designed for place recognition. ORB-SLAM2 [19] uses Bag of Words [20] to find nearest neighbors for place recognition.

A third class of methods that avoids full comparisons between database and query images use place classification. The database is used to train a deep neural network that treats each place as a class, which requires additional and often unavailable knowledge about the database. [21] used this approach during training to learn a holistic image descriptor for place recognition. PlaNet [22] uses this technique for geo-localization also during inference. For a general overview over global geo-localization techniques please refer to [23].

Handling loops in the database allows the matching of a query to multiple database images. For example in graph-based SLAM [24], this allows the connection of a new query node to nodes with lower visual similarity but higher pose accuracy. Searching multiple matches also increases the risk for wrong matches. However, since the advent of robust graph optimization for SLAM [25], wrongly matched places can be robustly detected. Multiple loop closures are for example used in some pose graph SLAM approaches (e.g., [24, p.40]). MILD [18] was specifically designed to return multiple loop closure detections for this purpose.

III. ALGORITHMIC DESCRIPTION

In this section, we present an efficient place recognition approach (EPR) for sparse image comparisons between database and query. EPR is designed to be fast and to work online while it preserves or improves the performance of full image comparisons. It can handle loops (and stops) in the database to potentially find all matches between database and a query.

The key idea of EPR is the exploitation of spatio-temporal sequences in the database (DB) and query (Q) set to identify candidates for a subsequent comparison with an incoming query image: Matching candidates are selected from the K highest similarities of the previous timestep and supplemented by their v successors in DB. These found candidates are used to extract additional similar database candidates from the intra-database similarities S^{DB} to handle loops (and stops) for a final comparison with the new query.

A detailed algorithmic description of EPR is presented in Algorithm 1. The involved parameters are introduced in Sec. III-A. The following Sec. III-B provides an overview over the algorithmic steps in Algorithm 1. Sec. III-C proposes an automatic tuning of the required threshold θ_s^{DB} and the optional threshold θ_s^{reloc} in case of event-based relocalization. Sec. III-D proposes two relocalization strategies to recover from sequence loss.

A. Parameters

EPR requires two parameters and one optional parameter to be set:

1) K ... the initial number of matching candidates
2) v ... the number of added subsequent database descriptors of the matching candidates in DB. This parameter exploits the sequences in DB and Q
3) (optional) T^{reloc} ... timesteps until the next relocalization in case of periodic relocalization (PR). See Sec. III-D for both relocalization strategies.

B. Algorithmic overview

Input to EPR is an already computed set of image descriptors $db_i \in DB$ and $q_t \in Q$ for every database and new query image. The algorithm can be roughly divided into the following steps:

1) Compute the intra-database similarity $S^{DB} \in \mathbb{R}^{|DB| \times |DB|}$ by comparing all descriptors $db_i \in DB$. For better performance, feature standardization [5] over all db_i could be applied (line 1)

2) Perform a relocalization for the first incoming query descriptor q_1 (line 2)
3) According to Sec. III-C, auto-tune θ_s^{DB} (and θ_s^{reloc} in case of event-based relocalization (ER), see Sec. III-D) (lines 3-4)
4) For every new incoming query descriptor q_t (line 5):
   a) Select matching candidates db_c with $c \in C^{DB}$ based on the K-best comparisons from the previous timestep $t-1$, their v successors in DB and their most similar database images with $S^{DB} \geq \theta_s^{DB}$ (lines 6-10)
   b) Compare all matching candidates db_c with $c \in C^{DB}$ to the current query descriptor q_t (line 11)
   c) If $relocalize(\cdot) \leftrightarrow true$ (see Sec. III-D), perform a relocalization by comparing query q_t to all database descriptors $db_i \in DB$ (lines 12-13)
   d) Else if $relocalize(\cdot) \leftrightarrow false$, find the database descriptors db_c with the K-highest similarities S_{ct} for the current timestep t and add their most similar database descriptors $db_{\hat{c}}$ if $S^{DB}_{\hat{c}c} \geq \theta_s^{DB}$. Compare these candidates to the current query q_t to compute similarities S_{ct} (lines 14-18)

C. Automatic tuning of θ_s^{DB} and θ_s^{reloc}

The two thresholds θ_s^{DB} and θ_s^{reloc} in Algorithm 1 are particularly dependent on the dataset and the used image descriptor. In order to apply EPR over a wider range of datasets with a single set of parameters, both thresholds have to be tuned automatically.

Our proposed unsupervised method tries to model the distribution of similarities of different places to exclude them from all similarities with a certain probability. As we cannot divide similarities from descriptor comparisons into same and different places without labels, we assume that there are more different places than same places. This assumption is usually valid for larger-scale datasets. Fig. 1 shows a typical distribution of the intra-database similarities S^{DB} and the similarities $S_{:1}$ between database and the first query.

The automatic parameter tuning fits a normal distribution

$$\mathcal{N}(\mu, \sigma^2) \sim p(s \mid \text{different places}) \tag{1}$$

to the similarity values. Because of the contained outliers (the same places) in S^{DB} and $S_{:1}$, a robust parameter estimation [26] for normal distributions is used for S^{DB}

$$\mu \approx median(S^{DB}) \tag{2}$$

$$\sigma \approx median(|S^{DB} - median(S^{DB})|)/0.675 \tag{3}$$

and similarly for $S_{:1}$. For σ we use the normalized median absolute deviation (MADN) [26].

Given the model in Eq. 1 and the estimated parameters from Eq. 2 and 3, we determine the parameters θ_s^{DB} and θ_s^{reloc} that satisfy

$$p(s < \theta_s^{DB} \mid \text{different places}) = 1 - 10^{-6} \tag{4}$$

$$p(s < \theta_s^{reloc} \mid \text{different places}) = 0.95 \tag{5}$$

Algorithm 1: Algorithmic description of our proposed Efficient Place Recognition (EPR) method

```
Data: $db_i \in DB$, $q_t \in Q$
Input: parameters $K$, $v$, ($T^{reloc}$ in case of periodic
       relocalization, see Sec. III-D
Result: $S \in \mathbb{R}^{|DB|\times|Q|}$
// compute intra-database similarities $S^{DB}$ from
   (standardized) database descriptors
1  $S^{DB}_{ij} = \frac{db_i^T \cdot db_j}{\|db_i\|\cdot\|db_j\|} \quad \forall i,j = 1,\dots,|DB|$
// initial relocalization
2  $S_{i1} = \frac{db_i^T \cdot q_1}{\|db_i\|\cdot\|q_1\|} \quad \forall i = 1,\dots,|DB|$
// auto-tune $\theta_s^{DB}$ (and $\theta_s^{reloc}$) according to Sec. III-C
3  $\theta_s^{DB} = autotune(S^{DB})$
4  $\theta_s^{reloc} = autotune(S_{:1})$
// for all incoming query images
5  for $t = 2,\dots,|Q|$ do
       // find $K$ highest similarities from previous timestep $t-1$
6      $C^{DB} :=$ K-$\underset{c}{\text{argmax}}(S_{c,(t-1)},\ K)$
       // for all candidates in $C^{DB}$ find database images in the
          intra-database similarities $S^{DB}$ with similarity $\geq \theta_s^{DB}$
7      for $\forall c \in C^{DB}$ do
8          $C^{DB} := C^{DB} \cup \{\hat{c} \mid S^{DB}_{\hat{c}c} \geq \theta_s^{DB}, \hat{c} \notin C^{DB}\}$
       // exploit the sequence assumption by adding the $v$
          consecutive images in $DB$ for all candidates in $C^{DB}$
9      for $\forall c \in C^{DB}$ do
10         $C^{DB} := C^{DB} \cup \{c+i,\ \forall i = 1,\dots,v \mid c+i \notin C^{DB}\}$
       // compare query $q_t$ to all candidates in $C^{DB}$
11     $S_{ct} = \frac{db_c^T \cdot q_t}{\|db_c\|\cdot\|q_t\|} \quad \forall c \in C^{DB}$
       // periodic or event-based relocalizaiton given either
          $\theta_s^{reloc}$ or $T^{reloc}$ (see Sec. III-D)
12     if relocalize($\cdot$) then
13         $S_{it} = \frac{db_i^T \cdot q_t}{\|db_i\|\cdot\|q_t\|} \quad \forall i = 1,\dots,|DB|$
14     else
           // find loops in intra-database similarities $S^{DB}$ for $K$
              highest similarities in current timestep
15         $C^{DB} :=$ K-$\underset{c}{\text{argmax}}(S_{ct},\ K)$
16         for $\forall c \in C^{DB}$ do
17             $C^{DB} := C^{DB} \cup \{\hat{c} \mid S^{DB}_{\hat{c}c} \geq \theta_s^{DB}, \hat{c} \notin C^{DB}\}$
18         $S_{ct} = \frac{db_c^T \cdot q_t}{\|db_c\|\cdot\|q_t\|} \quad \forall c \in C^{DB}$
```

Similarities above these thresholds are likely to belong to same places. We can use a higher probability for the intra-database similarities, since place descriptors perform better under more constant conditions.

Fig. 1 compares the result of a robust and regular normal distribution fitting. While the robust fitting performs well, the regular fitting (with mean and standard deviation) rather fails. Fig. 2 shows an intra-database similarity matrix S^{DB} thresholded with the auto-tuned θ_s^{DB}.

D. Relocalization strategy

EPR performs image comparisons locally in the similarity matrix S: Candidate descriptors in the database for the

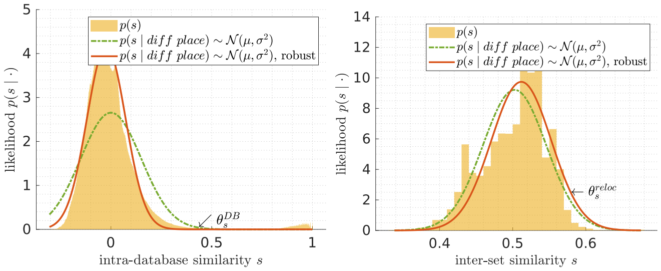

Fig. 1. Distribution of intra-database similarities S^{DB} (left) and of similarities $S_{:1}$ between all database images and the first query image (right). The solid red lines show the robustly fitted normal distribution model and the dashed green lines the non-robustly fitted normal distribution model. The parameters θ_s^{DB} and θ_s^{reloc} were finally determined from the red line. See Sec. III-C for a detailed description.

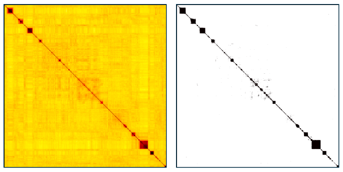

Fig. 2. Intra-database similarities S^{DB} (left) and thresholded with the auto-tuned θ_s^{DB} according to Sec. III-C (right)

current timestep t are selected from neighboring descriptors (with similar indices) from the previous timestep. Merely the intra-database similarities allow the choice of more distant candidates (much higher or smaller indices) in the database. While this sequence-based approach allows a sparse image comparison, it can also cause sequence loss, especially after exploration during the query run. In case of sequence loss, new query images could be compared to arbitrary non-matching database images despite existing matching database images. This behavior can be observed in Fig. 5.

In order to recover from a sequence loss, we use a relocalization that compares the current query q_t to all database descriptors $db_i \in DB$ with

$$S_{it} = \frac{db_i^T \cdot q_t}{\|db_i\| \cdot \|q_t\|} \quad \forall i = 1, \ldots, |DB| \tag{6}$$

To be more efficient than this full comparison, ANN-based methods could be used. However, these methods would involve additional properties that have to be evaluated, and are therefore subject of future work.

We propose two relocalization strategies: 1) periodic relocalization and 2) event-based relocalization. The qualitative performance of both strategies on a single dataset can be seen in Fig. 5.

1) Periodic relocalization (PR): A relocalization is performed periodically every T^{reloc} timesteps with

$$relocalize(\cdot) = \begin{cases} true, \text{ if } t \bmod T^{reloc} \leftrightarrow 0 \\ false, \text{ otherwise} \end{cases} \tag{7}$$

2) Event-based relocalization (ER): Given the already computed similarities S_{it} for the current timestep t, a relocalization is performed if none of the similarities exceed threshold θ_s^{reloc} with

$$relocalize(\cdot) = \begin{cases} true, \text{ if } \nexists i \text{ s.t. } S_{it} \geq \theta_s^{reloc} \\ false, \text{ otherwise} \end{cases} \tag{8}$$

IV. EXPERIMENTAL RESULTS

A. Experimental setup

NetVLAD [4] and AlexNet [3] are used as CNN-descriptors in the following experiments. For NetVLAD, we use the author's implementation trained on the Pitts30k dataset with VGG-16 and whitening that returns a normalized 4096-dimensional descriptor. For AlexNet, we use the flattened 64896-dimensional conv3-layer output of Matlab's ImageNet model. The area under the precision-recall curve (AUC) is used as performance metric over all experiments. The cosine similarity is used to measure the similarity between two descriptors.

For EPR, we used the parameters $K = 5$, $v = 5$ and $T^{reloc} = 100$ for PR. Feature standardization [5] on the database descriptors was used before the computation of intra-database similarities.

EPR is compared to OPR and HNSW: 1) Online Place Recognition (OPR) [12] is a sequence-based approach. We used the author's C++ implementation with parameters $K = 5$ and $\alpha = 0.6$. 2) Hierarchical Navigable Small World graphs (HNSW) [16] is an ANN method that was shown to be particularly suited for high-dimensional data [15]. We used the author's C++ implementation with parameters $ef = 20$ and $M = 40$, and search for $K = 5$ nearest neighbors. AlexNet descriptors were downsampled to 6490 dimensions before HNSW using Gaussian random projection.

B. Datasets

Our evaluation is based on five datasets with different characteristics regarding environment, appearance changes, single or multiple visits of places, possible stops, or viewpoint changes. We basically used the same five datasets **Nordland** [27], **StLucia** (Various Times of the Day) [28], **CMU** Visual Localization [29], **Gardens Point** Walking [30], and **Oxford** RobotCar [31] as described in our publication [11].

For Oxford, we use a new set of sequences sampled at 1Hz with the recently published accurate ground truth data [32]. For Nordland, we sampled the first two hours of each season including stops at 1Hz and removed tunnels.

C. Qualitative performance

Fig. 3 visualizes the qualitative performance of EPR with periodic relocalization (PR) together with ground truth and full image comparisons. The sparse similarity matrix S from EPR-PR clearly emphasizes the high efficiency compared to full image comparisons and its ability to correctly match multiple database images to a current query. The periodic relocalization can be seen as thin vertical lines.

D. Performance comparison

To measure the quantitative performance of our method EPR with periodic (PR) and event-based (ER) relocalization, we conduced experiments over five datasets with 21 sequence combinations. We compare it with the *full* image comparisons and with OPR and HNSW. We measured the single-matching performance which evaluates for each query image only the highest similarity to a database image as

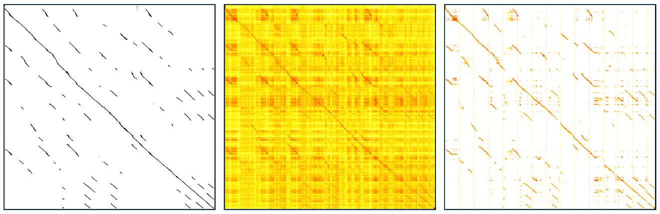

Fig. 3. Evaluated similarities $S \in \mathbb{R}^{|DB| \times |Q|}$ between database and query images for StLucia 08:45. White pixels represent uncompared image pairs. (left) ground truth, (mid) full image comparisons, (right) EPR-PR

well as the multi-matching performance that evaluates all similarities from all compared image pairs per query image. The multi-matching performance evaluates the capability of a method to find all matches for a query in case of loops in the database.

Table I shows the achieved absolute performance for each evaluated method. Fig. 4 illustrates their relation to the percentage of evaluated image pairs. The single-matching performance of the full image comparisons can be maintained by most methods. Only OPR performs worse on approximately half of the datasets, because it often loses track of the sequences due to low similarities, exploration during the query run and stops in the database (see also Sec. IV-F). EPR-ER with AlexNet fails on a single Oxford dataset with query images at night, since it cannot infer the need for a relocalization.

The multi-matching performance is often improved by our proposed methods, because of their ability to derive multiple matches from the intra-database similarities S^{DB} while ignoring similar looking image pairs of different places. OPR, HNSW and EPR-PR without S^{DB} fail on many datasets. HNSW can achieve high performance on some datasets with few loops and no stops (while FLANN [17] as an ANN-based method achieved only low performance in the literature on place recognition [12]).

EPR-PR without an exploitation of intra-database similarities (EPR-PR w/o S^{DB}) fails on the multi-matching task while it shows a surprisingly good single-matching performance. In conclusion, our proposed methods achieved the most performance gains, and performed best on most datasets for both the single- and multi-matching use-case.

E. Percentage of evaluated image pairs

Table II shows the corresponding percentages of evaluated image pairs over all datasets and methods. Fig. 4 illustrates the relation between the achieved performance gain over full image comparisons and the percentage of evaluated image pairs for our best performing method EPR-PR as well as the two methods OPR and HNSW.

Our method EPR-PR achieves a good tradeoff between performance gain or preservation and percentage of comparisons for single and multi matching, while OPR partially fails. HNSW by design achieves the lowest percentage, but partially fails on multi-matching.

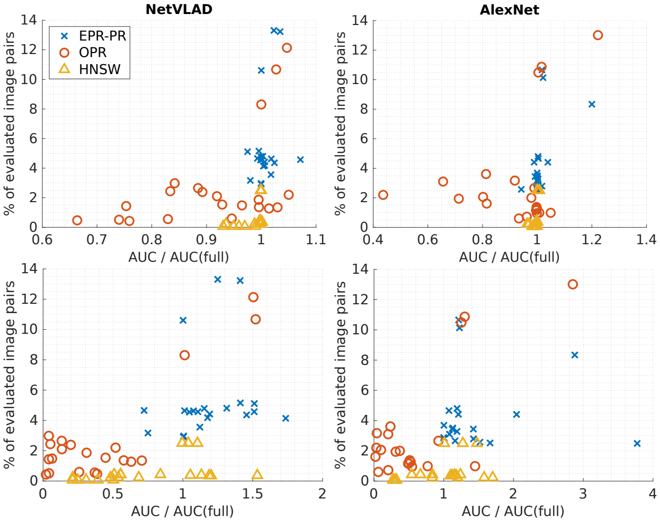

Fig. 4. Relative performance improvement compared to full image comparisons vs percentage of evaluated image pairs over all datasets from Table I and II for single-matching performance (top) and multi-matching performance (bottom)

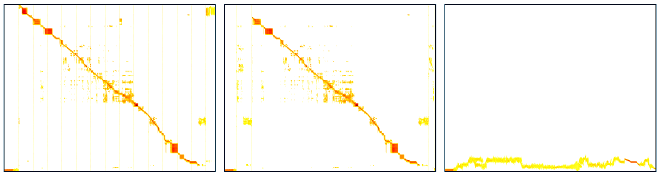

Fig. 5. Visualization of the exploratory query problem in $S \in \mathbb{R}^{|DB| \times |Q|}$: Without a relocalization strategy, sequence-based methods are likely to fail if the first or intermediate query images do not show places that are represented in the database. While our methods EPR-PR (left) and EPR-ER (mid) can recover due to relocalization, OPR (right) cannot

F. Influence of relocalization

Fig. 5 shows the problem of exploration during the query run and the benefit of relocalization: The shown dataset starts with exploratory queries. While EPR-PR can recover right after the beginning of the actual sequence, EPR-ER cannot immediately detect the need for a relocalization, but after a short period of time. OPR without a relocalization procedure completely fails to recover from the sequence loss.

G. Runtime

We implemented our proposed methods in Matlab and measured the runtimes on a notebook with Intel i7-8550U CPU. On the largest datasets with 6862 database and 6862 query images, our method EPR-PR required 7sec without the runtime for descriptor computations and the in average 330 image comparisons per query. On the same dataset, OPR required 142sec without descriptor computations and image comparisons despite its C++ implementation.

V. CONCLUSION

The experimental evaluation in this paper showed that the two state-of-the-art methods OPR and HNSW can considerably reduce the number of comparisons. However, each approach also revealed cases where the place recognition performance decreased compared to an exhaustive full comparison. This became obvious in multi-matching scenarios which are important in online scenarios, e.g., for SLAM. In

TABLE I

Performance (AUC) of the evaluated algorithms with NetVLAD or AlexNet descriptor. Best values per row and descriptor are bold. The colored arrows indicate large (≥25% better/worse) or medium (≥10%) deviation compared to "full".

			NetVLAD						AlexNet					
Dataset	**Database**	**Query**	**full**	**EPR-PR**	**EPR-ER**	**EPR-PR w/o S^{DB}**	**OPR**	**HNSW**	**full**	**EPR-PR**	**EPR-ER**	**EPR-PR w/o S^{DB}**	**OPR**	**HNSW**
			Single-Matching Performance											
Nordland	fall	spring	0.74	0.72 →	0.73 →	**0.90** ↗	0.61 ↘	0.69 →	**1.00**	**1.00** →	**1.00** →	**1.00** →	0.93 →	0.99 →
	fall	winter	0.68	0.68 →	0.67 →	**0.87** ↑	0.50 ↓	0.65 →	0.95	0.96 →	**0.97** →	**0.97** →	0.91 →	0.93 →
	spring	winter	0.75	0.75 →	0.71 →	**0.85** ↗	0.57 ↘	0.72 →	0.95	0.94 →	0.98 →	0.96 →	**1.00** →	0.94 →
	summer	spring	0.76	0.75 →	0.76 →	**0.90** ↗	0.50 ↓	0.72 →	0.99	**1.00** →	**1.00** →	**1.00** →	**1.00** →	0.99 →
	summer	fall	0.96	0.96 →	0.96 →	**1.00** →	0.91 →	0.95 →	**1.00**	**1.00** →	**1.00** →	**1.00** →	**1.00** →	**1.00** →
StLucia	100909-0845	190809-0845	0.95	**0.96** →	**0.96** →	0.95 →	**0.96** →	0.95 →	**0.99**	**0.99** →	**0.99** →	0.98 →	**0.99** →	**0.99** →
	100909-1000	210809-1000	0.96	0.96 →	**0.97** →	0.93 →	0.95 →	0.95 →	**0.98**	**0.98** →	**0.98** →	**0.98** →	**0.98** →	**0.98** →
	100909-1210	210809-1210	0.97	**0.98** →	**0.98** →	**0.98** →	0.89 →	0.97 →	**1.00**	**1.00** →	**1.00** →	0.99 →	0.97 →	**1.00** →
	100909-1410	190809-1410	0.95	**0.97** →	**0.97** →	0.91 →	**0.97** →	0.95 →	**0.99**	**0.99** →	**0.99** →	**0.99** →	**0.99** →	**0.99** →
	110909-1545	180809-1545	0.92	**0.94** →	**0.94** →	0.91 →	0.82 ↘	0.92 →	**0.99**	**0.99** →	**0.99** →	0.98 →	0.98 →	**0.99** →
CMU	20110421	20100901	**0.97**	**0.97** →	0.96 →	**0.97** →	0.85 ↘	**0.97** →	0.95	0.96 →	0.94 →	**0.97** →	0.62 ↓	0.95 →
	20110421	20100915	**0.97**	0.96 →	**0.97** →	0.96 →	0.81 ↘	**0.97** →	0.96	**0.97** →	**0.97** →	**0.97** →	0.78 ↘	0.96 →
	20110421	20101221	0.92	**0.98** →	0.92 →	**0.98** →	0.96 →	0.91 →	0.93	0.97 →	0.97 →	**0.98** →	0.92 →	0.93 →
	20110421	20110202	**0.99**	**0.99** →	**0.99** →	**0.99** →	0.83 ↘	**0.99** →	**0.98**	0.97 →	**0.98** →	0.93 →	0.90 →	**0.98** →
GardensPoint	day-right	day-left	**1.00**	**1.00** →	**1.00** →	**1.00** →	**1.00** →	**1.00** →	0.98	**1.00** →	**1.00** →	**1.00** →	**1.00** →	0.98 →
	day-right	night-right	0.97	0.99 →	0.99 →	**1.00** →	**1.00** →	0.97 →	0.96	0.98 →	0.97 →	**1.00** →	0.97 →	0.96 →
	day-left	night-right	0.94	0.97 →	0.96 →	**1.00** →	0.98 →	0.93 →	0.76	0.92 ↗	**0.93** ↗	0.92 ↗	**0.93** ↗	0.77 →
Oxford	2014-12-09-13-21-02	2015-05-19-14-06-38	**1.00**	**1.00** →	**1.00** →	0.99 →	0.99 →	**1.00** →	**0.98**	**0.98** →	0.95 →	0.95 →	0.79 ↘	**0.98** →
	2014-12-09-13-21-02	2015-08-28-09-50-22	**0.98**	0.97 →	0.96 →	0.95 →	0.91 →	0.97 →	**0.92**	0.87 →	0.91 →	0.88 →	0.40 ↓	0.89 →
	2014-12-09-13-21-02	2014-11-25-09-18-32	**1.00**	**1.00** →	0.99 →	0.99 →	0.96 →	**1.00** →	**0.99**	0.98 →	0.93 →	0.98 →	0.80 ↘	0.98 →
	2014-12-09-13-21-02	2014-12-16-18-44-24	0.81	0.83 →	0.78 →	**0.85** →	0.61 ↘	0.76 →	0.84	**0.85** →	0.07 ×	0.81 →	0.60 ↓	0.81 →
			Multi-Matching Performance											
Nordland	fall	spring	0.39	0.58 ↑	**0.61** ↑	0.08 ×	0.14 ↓	0.12 ↓	**1.00**	**1.00** →	**1.00** →	0.08 ×	0.06 ×	0.27 ↓
	fall	winter	0.21	**0.36** ↑	0.26 ↗	0.05 ×	0.01 ×	0.10 ↓	0.63	0.90 ↑	**0.92** ↑	0.08 ×	0.13 ↓	0.20 ↓
	spring	winter	0.45	**0.59** ↑	0.50 ↗	0.05 ×	0.01 ×	0.10 ↓	0.52	0.74 ↑	**0.80** ↑	0.07 ×	0.76 ↑	0.16 ↓
	summer	spring	0.43	0.60 ↑	**0.61** ↑	0.09 ×	0.17 ↓	0.17 ↓	0.99	**1.00** →	**1.00** →	0.09 ×	0.76 ↘	0.28 ↓
	summer	fall	0.96	**0.97** →	**0.97** →	0.10 ↓	0.25 ↓	0.24 ↓	**1.00**	**1.00** →	**1.00** →	0.08 ×	0.54 ↓	0.32 ↓
StLucia	100909-0845	190809-0845	0.41	0.48 ↗	0.48 ↗	0.24 ↓	0.26 ↓	**0.49** ↗	0.59	0.64 →	**0.71** ↗	0.29 ↓	0.29 ↓	0.65 ↗
	100909-1000	210809-1000	0.47	**0.54** ↗	0.51 →	0.21 ↓	0.27 ↓	0.53 ↗	0.57	0.67 ↗	**0.71** ↗	0.32 ↓	0.29 ↓	0.65 ↗
	100909-1210	210809-1210	0.51	**0.61** ↗	0.60 ↗	0.31 ↓	0.07 ×	0.53 →	0.54	0.61 ↗	**0.65** ↗	0.31 ↓	0.20 ↓	0.64 ↗
	100909-1410	190809-1410	0.38	0.56 ↑	**0.57** ↑	0.15 ↓	0.27 ↓	0.46 ↗	0.61	0.61 →	0.64 →	0.34 ↓	0.30 ↓	**0.66** →
	110909-1545	180809-1545	0.27	0.29 →	0.32 ↗	0.15 ↓	0.05 ×	**0.41** ↑	0.60	**0.68** ↗	0.65 →	0.36 ↓	0.31 ↓	0.67 ↗
CMU	20110421	20100901	0.73	**0.76** →	0.65 ↘	0.49 ↓	0.10 ↓	0.40 ↓	0.44	**0.52** ↗	0.43 →	0.32 ↓	0.09 ×	0.29 ↓
	20110421	20100915	0.77	0.56 ↓	**0.80** →	0.32 ↓	0.04 ×	0.39 ↓	0.59	**0.64** →	0.48 ↘	0.31 ↓	0.14 ↓	0.32 ↓
	20110421	20101221	0.56	**0.85** ↑	0.61 →	0.45 ↘	0.29 ↓	0.31 ↓	0.34	**0.69** ↑	0.68 ↑	0.33 →	0.31 →	0.28 ↘
	20110421	20110202	0.61	**0.68** ↗	**0.68** ↗	0.52 ↘	0.03 ×	0.51 ↘	0.33	0.40 ↗	**0.41** ↗	0.24 ↓	0.01 ×	**0.41** ↗
GardensPoint	day-right	day-left	0.97	0.97 →	0.97 →	**1.00** →	0.98 →	0.97 →	0.58	0.71 ↗	0.68 ↗	**0.87** ↑	0.76 ↑	0.59 →
	day-right	night-right	0.51	0.63 ↗	0.66 ↑	0.70 ↑	**0.77** ↑	0.56 →	0.51	0.62 ↗	0.58 ↗	**0.86** ↑	0.64 ↑	0.65 ↑
	day-left	night-right	0.40	0.56 ↑	0.54 ↑	**0.78** ↑	0.60 ↑	0.42 →	0.11	0.30 ↑	**0.37** ↑	0.35 ↑	0.30 ↑	0.16 ↑
Oxford	2014-12-09-13-21-02	2015-05-19-14-06-38	**0.78**	**0.78** →	**0.78** →	0.38 ↓	0.24 ↓	0.42 ↓	0.24	**0.41** ↑	0.39 ↑	0.19 ↘	0.02 ×	0.29 ↗
	2014-12-09-13-21-02	2015-08-28-09-50-22	**0.60**	0.45 ↘	0.32 ↓	0.16 ↓	0.27 ↓	0.29 ↓	0.11	0.16 ↑	**0.23** ↑	0.12 →	0.00 ×	0.17 ↑
	2014-12-09-13-21-02	2014-11-25-09-18-32	**0.87**	**0.87** →	0.76 ↘	0.57 ↓	0.06 ×	0.60 ↓	0.42	**0.49** ↗	0.47 ↗	0.24 ↓	0.01 ×	0.35 ↘
	2014-12-09-13-21-02	2014-12-16-18-44-24	0.55	**0.62** ↗	0.60 →	0.10 ↓	0.02 ×	0.12 ↓	0.07	**0.27** ↑	0.01 ×	0.07 ×	0.02 ×	0.12 ↑

TABLE II

Percentage of evaluated image pairs between database and query. The ground truth (GT) shows the percentage of required (min) and allowed (max) matches (allowed matches due to small visual overlaps between consecutive images)

			GT		NetVLAD						AlexNet					
Dataset	**Database**	**Query**	**min**	**max**	**full**	**EPR-PR**	**EPR-ER**	**EPR-PR w/o S^{DB}**	**OPR**	**HNSW**	**full**	**EPR-PR**	**EPR-ER**	**EPR-PR w/o S^{DB}**	**OPR**	**HNSW**
Nordland	fall	spring	1.27	2.11	100	5.11	4.36	1.20	0.56	0.07	100	2.87	4.23	1.18	0.61	0.07
	fall	winter	1.27	2.11	100	4.14	3.94	1.23	0.51	0.07	100	2.79	5.42	1.18	0.72	0.07
	spring	winter	1.27	2.11	100	4.81	5.43	1.22	0.43	0.07	100	3.44	11.80	1.18	0.99	0.07
	summer	spring	1.27	2.11	100	5.15	4.33	1.22	0.47	0.07	100	2.82	2.83	1.19	0.99	0.07
	summer	fall	1.27	2.11	100	4.64	3.72	1.21	0.60	0.07	100	2.88	2.04	1.17	0.95	0.07
StLucia	100909-0845	190809-0845	0.17	1.24	100	4.18	3.26	2.06	1.28	0.35	100	3.10	2.27	2.01	1.16	0.35
	100909-1000	210809-1000	0.16	1.26	100	4.79	3.90	2.18	1.36	0.39	100	3.28	2.51	2.11	1.37	0.39
	100909-1210	210809-1210	0.14	1.29	100	4.42	3.57	2.21	2.10	0.39	100	3.43	2.67	2.20	2.00	0.39
	100909-1410	190809-1410	0.17	1.28	100	4.36	3.38	2.12	1.35	0.36	100	3.69	2.87	2.01	1.17	0.36
	110909-1545	180809-1545	0.14	1.24	100	4.63	3.69	2.17	2.39	0.37	100	3.51	3.38	2.03	1.28	0.37
CMU	20110421	20100901	0.59	1.97	100	4.55	4.06	2.11	2.65	0.43	100	4.80	5.41	2.21	3.10	0.43
	20110421	20100915	0.47	1.66	100	4.66	4.66	2.04	2.44	0.43	100	4.66	5.23	2.11	3.61	0.43
	20110421	20101221	0.57	1.96	100	4.57	4.19	2.20	2.20	0.43	100	4.41	4.79	2.20	2.66	0.43
	20110421	20110202	0.32	1.97	100	4.57	3.89	2.15	2.98	0.43	100	4.41	6.34	2.27	3.17	0.43
GardensPoint	day-right	day-left	0.50	8.32	100	10.61	9.66	6.60	8.31	2.50	100	10.66	13.20	7.35	10.87	2.50
	day-right	night-right	0.50	8.32	100	13.31	14.06	7.78	10.67	2.50	100	10.14	16.53	7.30	10.48	2.50
	day-left	night-right	0.50	8.32	100	13.23	15.40	7.69	12.14	2.50	100	8.34	31.08	7.02	13.01	2.50
Oxford	2014-12-09-13-21-02	2015-05-19-14-06-38	0.13	0.93	100	2.91	2.02	1.60	1.87	0.23	100	2.52	1.56	1.60	2.06	0.23
	2014-12-09-13-21-02	2015-08-28-09-50-22	0.07	0.82	100	3.17	2.59	1.63	1.54	0.23	100	2.57	2.93	1.59	2.20	0.23
	2014-12-09-13-21-02	2014-11-25-09-18-32	0.07	0.93	100	2.97	2.54	1.58	1.48	0.23	100	2.66	1.38	1.61	1.61	0.23
	2014-12-09-13-21-02	2014-12-16-18-44-24	0.14	0.92	100	3.56	2.77	1.69	1.44	0.23	100	2.49	1.40	1.64	1.94	0.23

this paper, we proposed a new method, EPR, for efficient place recognition. The approach exploits structural knowledge, which is inherent to many practical place recognition setups, for a reduction of descriptor comparisons. Our experiments demonstrated that a single set of parameters could be successfully used for a wide range of datasets. This is partially due to the ability to auto-tune required thresholds. The comparison with the state-of-the-art approaches showed a good balance between performance and efficiency. Although there still exist rare cases where the performance decreased, in significantly more cases the presented approach was able to even improve the place recognition performance compared to a full comparison. This is based on the ability to use structural information to prevent the evaluation of false positive matchings. A key element of the presented approach to use structural knowledge is the ability to relocalize and to detect the need for relocalization. We presented two different relocalization strategies. In principle, they also allow the combination with ANN-approaches for an application on very large datasets. A particularly interesting open research question are more robust event-based relocalization strategies for immediate relocalization.

REFERENCES

[1] S. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, "Visual place recognition: A survey," *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 1–19, 2016.

[2] N. Sünderhauf, S. Shirazi, F. Dayoub, B. Upcroft, and M. Milford, "On the performance of convnet features for place recognition," in *International Conference on Intelligent Robots and Systems (IROS)*, 2015.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2012. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[4] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," *Trans. on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, 2018.

[5] S. Schubert, P. Neubert, and P. Protzel, "Unsupervised learning methods for visual place recognition in discretely and continuously changing environments," in *International Conference on Robotics and Automation (ICRA)*, 2020.

[6] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han, "Large-scale image retrieval with attentive deep local features," in *International Conference on Computer Vision (ICCV)*, 2017.

[7] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler, "D2-net: A trainable cnn for joint description and detection of local features," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[8] M. J. Milford and G. F. Wyeth, "Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights," in *International Conference on Robotics and Automation (ICRA)*, 2012.

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

[10] P. Hansen and B. Browning, "Visual place recognition using HMM sequence matching," in *International Conference on Intelligent Robots and Systems (IROS)*, 2014.

[11] P. Neubert, S. Schubert, and P. Protzel, "A neurologically inspired sequence processing model for mobile robot place recognition," *IEEE Robotics and Automation Letters (RA-L)*, vol. 4, no. 4, pp. 3200–3207, 2019.

[12] O. Vysotska and C. Stachniss, "Lazy data association for image sequences matching under substantial appearance changes," *IEEE Robotics and Automation Letters (RA-L)*, vol. 1, no. 1, pp. 213–220, 2016.

[13] S. Schubert, P. Neubert, and P. Protzel, "Graph-based non-linear least squares optimization for visual place recognition in changing environments," *IEEE Robotics and Automation Letters (RA-L)*, vol. 6, no. 2, pp. 811–818, 2021.

[14] P. Neubert, S. Schubert, and P. Protzel, "Resolving place recognition inconsistencies using intra-set similarities," *IEEE Robotics and Automation Letters (RA-L)*, vol. 6, no. 2, pp. 2084–2090, 2021.

[15] W. Li, Y. Zhang, Y. Sun, W. Wang, M. Li, W. Zhang, and X. Lin, "Approximate nearest neighbor search on high dimensional data experiments, analyses, and improvement," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 8, pp. 1475–1488, 2020.

[16] Y. A. Malkov and D. A. Yashunin, "Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 4, pp. 824–836, 2020.

[17] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *International Conference on Computer Vision Theory and Applications*, 2009.

[18] L. Han and L. Fang, "Mild: Multi-index hashing for appearance based loop closure detection," in *International Conference on Multimedia and Expo (ICME)*, 2017.

[19] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[20] D. Galvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, 2012.

[21] Z. Chen, A. Jacobson, N. Sünderhauf, B. Upcroft, L. Liu, C. Shen, I. Reid, and M. Milford, "Deep learning features at scale for visual place recognition," in *International Conference on Robotics and Automation (ICRA)*, 2017.

[22] T. Weyand, I. Kostrikov, and J. Philbin, "Planet - photo geolocation with convolutional neural networks," in *European Conference on Computer Vision (ECCV)*, 2016.

[23] J. Brejcha and M. Cadik, "State-of-the-art in visual geo-localization," *Pattern Analysis and Applications*, vol. 20, no. 3, pp. 613–637, 2017.

[24] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based slam," *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, 2010.

[25] N. Sünderhauf and P. Protzel, "Switchable constraints for robust pose graph slam," in *International Conference on Intelligent Robots and Systems (IROS)*, 2012.

[26] R. Maronna, *Robust Statistical Methods*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 1244–1248.

[27] N. Sünderhauf, P. Neubert, and P. Protzel, "Are we there yet? challenging seqslam on a 3000 km journey across all four seasons," *International Conference on Robotics and Automation (ICRA) Workshop on Long-Term Autonomy*, 2013.

[28] A. J. Glover, W. P. Maddern, M. J. Milford, and G. F. Wyeth, "Fabmap + ratslam: Appearance-based slam for multiple times of day," in *International Conference on Robotics and Automation (ICRA)*, 2010.

[29] H. Badino, D. Huber, and T. Kanade, "Visual topometric localization," in *Intelligent Vehicles Symposium (IV)*, 2011.

[30] *A. Glover. "Day and Night with Lateral Pose Change Datasets". (2014). https://goo.gl/tqmWyq.*

[31] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The oxford robotcar dataset," *The International Journal of Robotics Research*, vol. 36, no. 1, pp. 3–15, 2017.

[32] W. Maddern, G. Pascoe, M. Gadd, D. Barnes, B. Yeomans, and P. Newman, "Real-time kinematic ground truth for the oxford robotcar dataset," 2020.