Visual Place Recognition using Directed Acyclic Graph Association Measures and Mutual Information-based Feature Selection

Jurica Maltar[a], Ivan Marković[b], Ivan Petrović[b]

[a] *J. J. Strossmayer University of Osijek, Department of Mathematics, Trg Ljudevita Gaja 6, HR-31000, Osijek, Croatia*
[b] *University of Zagreb Faculty of Electrical Engineering and Computing, Laboratory for Autonomous Systems and Mobile Robotics, Unska 3, HR-10000, Zagreb, Croatia,*

Abstract

Visual localization is a challenging problem, especially over the long run, since places can exhibit significant variation due to dynamic environmental and seasonal changes. To tackle this problem, we propose a visual place recognition method based on directed acyclic graph matching and feature maps extracted from deep convolutional neural networks (DCNN). Furthermore, in order to find the best subset of DCNN feature maps with minimal redundancy, we propose to form probability distributions on image representation features and leverage the Jensen-Shannon divergence to rank features. We evaluate the proposed approach on two challenging public datasets, namely the Bonn and the Freiburg datasets, and compare it to the state-of-the-art methods. For image representations, we evaluated the following DCNN architectures: AlexNet, OverFeat, ResNet18 and ResNet50. Due to the proposed graph structure, we are able to account for any kind of correlations in image sequences, and therefore dub our approach NOSeqSLAM. Algorithms with and without feature selection were evaluated based on precision-recall curves, area under the curve score, best recall at 100% precision score and running time, with NOSeqSLAM outperforming the counterpart approaches. Furthermore, by formulating the mutual information-based feature selection specifically for visual place recognition and by selecting the feature percentile with the best score, all the algorithms, and not just NOSeqSLAM, exhibited enhanced performance with the reduced feature set.

Keywords: Visual place recognition, localization, deep convolutional neural networks, mutual information-based feature selection, SeqSLAM

1. Introduction

One of the fundamental building blocks of an autonomous mobile robot or a vehicle is the ability to reason about its location in a given environment. This challenge can be tackled by various approaches and in the current paper we focus on visual place recognition. As defined in [1], visual place recognition is "[...] the problem of a mobile robot identifying its current location from a database of previously visited locations, using vision as the primary or only sensor". In other words, the mobile robot or a vehicle drives through a previously traversed and labeled route and tries to find the corresponding match for a previously visited place. During both traversals, the robot captures places in the form of images and thereafter, each place is represented by an image. Images of the previous route traversal are stored in the *reference database*, $\mathcal{D}$, while images of the current traversal are stored in the *query database*, $\mathcal{Q}$. Given the notation, we can now formulate a more precise definition of visual place recognition:

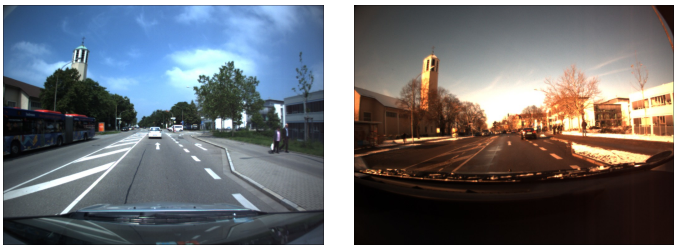


Figure 1: (a) $I_d \in \mathcal{D}$ captured during the first traversal. (b) $I_q \in \mathcal{Q}$ captured during the second traversal. Both images represent the same place, but, as can be seen, intensive variation of conditions is present. Visual place recognition deals with how to match these images.

given a query image $I_q \in \mathcal{Q}$ the task is to find the corresponding database image $I_d^* \in \mathcal{D}$. Illustration of the visual place recognition problem is given in Fig. 1.

Visual place recognition is related to the *visual instance retrieval* [41] problem; however, subtle differences exist. First, in visual instance retrieval both the reference database $\mathcal{D}$ and the query database $\mathcal{Q}$ are not periodically ordered, i.e., they are not *sequential*, while in visual place

Email addresses: jmaltar@mathos.hr (Jurica Maltar), ivan.markovic@fer.hr (Ivan Marković), ivan.petrovic@fer.hr (Ivan Petrović)

recognition, conversely, this is the case and we exploit the *sequentiality* in order to gain even more information about a specific place. SeqSLAM [32] and the proposed approach rely heavily on this fact. Second, *view-point variance*, i.e., variation of the camera reference frame between $\mathcal{D}$ and $\mathcal{Q}$, is more emphasized in visual instance retrieval than visual place recognition. Besides view-point variance, *condition variance* is another challenge to be addressed. This encompasses all the environmental factors that contribute to the difference in the visual representation of an image – weather, season (emphasized in Fig. 1), moment of the day, moving objects etc. It is thus the ultimate objective of the visual place recognition system to be *condition invariant* and *view-point invariant*, and to achieve this, design aspects that can be investigated are *image representation*, *image matching* and *feature selection*. Given that, visual place recognition can be seen as a specific instance of visual instance retrieval with its characteristic challenges and constraints that can be leveraged to yield better tailored solutions.

Regarding image representation, although some works use human-understandable representation of an image [1, 32], a more common approach is to propagate the image through the computer vision techniques in order to obtain its *features* (salient regions) and the corresponding descriptors that discriminatively describe this region. Handcrafted features have been used in robotics since the advent of computer vision in general [33]. In the context of visual place recognition SURF [4] is used in FAB-MAP [12], while ORB features [6, 42] are used in ORB-SLAM [34]. These local descriptors are sometimes used in combination with methods that yield *global descriptors* such as *bag-of-words* (BoW) [17] or vector of locally aggregated descriptors (VLAD) [24], e.g., [15] uses SIFT [29] in combination with VLAD. *Histogram of oriented gradients* (HOG) is another notable global descriptor used in [35, 36, 37, 50]. However, as it has been shown in previous works, and as it will be shown in this paper, handcrafted features can be replaced by features extracted from deep convolutional neural networks (DCNNs) in the favor of achieving better performance.

Feature maps extracted from deep convolutional neural networks (DCNNs) have been extensively used as global image descriptors. Sünderhauf et al. [45] wondered if these feature maps are appropriate for the problem of visual place recognition by using the AlexNet architecture [26]. Therein, authors concluded that DCNN-extracted features perform better than the handcrafted ones and, specifically, feature maps extracted from the middle and higher convolutional layers perform better than those extracted from the earlier layers. Furthermore, feature maps extracted from middle layers are more suitable when condition variance occurs, as these layers encode elementary visual entities, while feature maps extracted from higher layers perform better when view-point variance is emphasized, as they encode semantic information. The authors also claim that architecture trained on a *scene-centric dataset* performs even better than the one trained on an *object-centric dataset*. Naseer et al. [36] used AlexNet and GoogLeNet [47] architectures, while Vysotska et al. [50, 51] used features extracted from OverFeat [43]. Besides using the whole image and passing it through a neural network, some authors first find more "salient" regions in the image and then process them further. For example, Sünderhauf et al. [46] use an *object proposal* method that finds significant objects in the image, and later on those regions are propagated through the neural network yielding an even more view-point invariant representation. Arandjelović et al. proposed NetVLAD [2] by modifying the original VLAD, where the indicator function is replaced with *softmax* and therefore obtain representation that can be trained in an *end-to-end* manner specifically for visual place recognition. Similar to NetVLAD, Garg et al. [18] propose *local semantic tensor* (LoST) which aggregates residuals of semantic categories. Maximum activations from feature maps are used for image representation too. Chen et al. [7] represent each slice in feature map with 30-dimensional vector obtained by *multi-scale pooling* where slice is divided into $k \in \{1, 2, 3, 4\}$ squares and maximum activations from each k-subdivision were concatenated. In a similar manner, Hausler et al. [20] process a slice with *maximum spatial pooling* obtaining 5-dimensional feature vector from each slice. In their follow-up work [21], the authors propose to remove bad slices in the earlier layers of DCNN architecture used for feature extraction. These weights and biases in the specific earlier layer are annulated and then feature maps are further propagated. Removing the bad slices can be considered as a feature selection technique in the context of visual place recognition. In addition to ordinary image representations (i.e. RGB or grayscale images), other sensor modalities can be used. Cupec et. al [13] represent a place using planar surface segments and line segments obtained from depth images since such a representation is more condition-invariant.

Regarding image matching for visual place recognition, SeqSLAM [32] is often used where matches between $\mathcal{Q}$ and $\mathcal{D}$ are obtained by observing the local neighborhood of the corresponding images. In Section 3 we will examine this work more closely as it has influenced our work presented in this paper. Siam et al. [44] improved SeqSLAM using *approximate nearest neighbor* (ANN) in order to obtain N nearest images from $\mathcal{D}$ that correspond to a specific time instance image $I_T \in \mathcal{Q}$. Thereafter, only K of N nearest neighbors were considered for the possible matching candidates and each pair between I_T and the particular candidate is processed as in the ordinary method. Alongside reduction in the space of reference candidates, the obtained velocity for I_T image match is used in order to append one more candidate for image I_{T+1} which is implicated by the fact that a vehicle tends to move at the constant velocity between two sampled query images. Yin et al. [52] proposed to reduce SeqSLAM computational complexity by using particle filtering. Particles are re-sampled across reference database and SeqSLAM procedure is evaluated

for those pairs that contain a particle. Simultaneously, the number of particles is halved while the frame rate of $\mathcal{Q}$ and $\mathcal{D}$ is doubled. As the authors claim, this provides moderate computational effort regardless of the number of particles and frame rate. Pepperell et al. [39] integrate the odometry from wheel encoders with SeqSLAM. By using odometry, both $\mathcal{Q}$ and $\mathcal{D}$ are captured, not with the same sampling frequency, but with the same longitudinal distance, meaning that the vehicle has traveled x meters between two captured images. This way, SeqSLAM considers only those velocities that yield linear sequences in correlation between query and reference indices, that form an angle of $\varphi \in \{40°, 45°, 50°\}$ horizontally when observing *the difference matrix*. Another interesting approach, when it comes to matching by observing the local neighborhood, is by Garg et al. [19]. Based on their previous work [18], for each query image N candidates from the reference database are selected. Thereafter, the depth from an image is approximated and for each query image, the reference image with the most appropriate neighborhood is selected. Le et al. [27] use a binary tree in order to achieve sub-linear $\mathcal{O}(\log|\mathcal{D}|)$ memory complexity. For each query image $I_{q_i} \in \mathcal{Q}$ they obtain the corresponding index for an image in $\mathcal{D}$ by using classifiers trained on the different levels of a tree. In the literature, Bayesian inference for matching is used too [40, 12, 15, 35]. By using network flows, Naseer et al. [35] addressed the problem of visual place recognition. Min-cost flow problem used for matching is later reduced to find the single source shortest path of the directed acyclic graph (DAG) which is efficiently solved with topological sorting. Work of Vysotska et al. [51] is a follow-up to [35] where the system operates in an *online* fashion. The shortest path in these works represents a general route hypothesis, while our work uses shortest path in order to measure the association between each $I_{q_i} \in \mathcal{Q}$ and $I_{d_j} \in \mathcal{D}$. Thus, we build a directed acyclic graph for each $(I_{q_i}, I_{d_j}) \in \mathcal{Q} \times \mathcal{D}$ and thereafter we find its shortest path in order to measure the association, while [35, 51] build one global DAG for a route hypothesis.

In this paper we propose a visual place recognition system based on directed acyclic graph matching with mutual information feature selection and deep convolutional neural networks. The proposed method is the extension of our preliminary work published in a conference paper [31]. The contributions of this paper are threefold. First, we propose an image matching algorithm called NOSeqSLAM which uses directed acyclic graphs between image similarities in order to measure their association. In addition to NOSeqSLAM, we developed a so called *on-the-fly relaxation* algorithm which replaces usual *single-source shortest path* algorithms and significantly improves the running time of NOSeqSLAM. Second, we formulate a *mutual information-based feature selection* in the context of visual place recognition, conduct the training, and apply feature selection on an existing image representation yielding an even more robust representation. Third, we evaluate the proposed algorithm using various image representations either without or with the proposed feature selection on two publicly available and challenging datasets, and outperform current visual place recognition state-of-the-art both in terms of the accuracy and the running time. The source code of our approach is also available online [1].

The paper is organized as follows. In Section 2, we present the theoretical background for mutual information-based feature selection in order to apply it in the proposed method. Then, in Section 3 we present the proposed image matching algorithm and explain how the feature selection is formulated specifically for visual place recognition. In Section 4 we present the experimental results, while Section 5 concludes the paper.

2. Mutual Information-based Feature Selection

Since visual place recognition relies on feature-based image representation, it would be beneficial to analyze which of the features bear the most information and then select the most informative subset, especially for the case of DCNNs which can yield large feature maps. Theoretical background on how to construct a probability distribution on image features and compute mutual information is described in this section, while Section 3 describes details of the proposed application to visual place recognition.

As *mutual information* measures the dependence between two random variables, it is used extensively for feature selection. In general, these methods select features iteratively, where in m-th step feature x_m is selected with

$$x_m = \underset{x}{\operatorname{argmax}}\{I(y;x) - R_y(x, S_{m-1})\}, \tag{1}$$

where $I(y;x)$ is the relevance of the feature x for class y, $R_y(x, S_{m-1})$ represents the redundancy between feature x and S_{m-1} with respect to the class y, while S_{m-1} represents the optimal subset of $m-1$ features for which the mutual information of the class y and this subset is maximal [9]. As can be seen from (1), the idea is to pick the feature x_m that maximizes the relevance, while simultaneously minimizing the redundancy. In the literature, (1) is derived and numerous methods have been proposed [16, 23, 28, 38]. For example, in [10, 11], an optimization technique is used in order to obtain features, as it is usual in *sparse feature selection* methods formulated with

$$\begin{aligned} &\min_{\beta \in \mathbb{R}^N} \|\mathbf{y}^T - \beta^T \mathbf{X}\|_2^2 \\ &\text{s.t.} \quad \|\beta\|_0 = k, \end{aligned} \tag{2}$$

where $(\mathbf{X}, \mathbf{y})$ is the training data while β^T holds the regression coefficients. This is an NP-hard problem and therefore various approximations for (2) exist [11, 10, 53, 49, 55, 30].

In our paper, we build on the feature selection method from [9]. From this work we have taken a measure for feature quality; however, we extend the formulation especially

[1] `https://bitbucket.org/unizg-fer-lamor/noseqslam/`

to obtain *the target features* for visual place recognition as will be shown in Section 3. In general, this approach uses tools from *information theory* in order to measure which subset of features is optimal, i.e., which are the most informative features. That is why *probability distributions* for each feature have to be built while the informativeness between features can be measured with the Jensen-Shannon divergence – a general information theoretic measure of the difference between probability distributions. In order to obtain probability distributions, in [9] authors proposed to build graphs for each feature as follows. Let $f^{(k)} \in \mathbb{R}^{M\times 1}$ denote a feature from the following set of features $\mathcal{S} = \{f^{(1)}, \ldots, f^{(k)}, \ldots, f^{(N)}\}$. For $f^{(k)}$ we can build the corresponding *undirected graph* $G_{f^{(k)}} = (V_{f^{(k)}}, E_{f^{(k)}})$, with V and E being sets of nodes and edges, respectively, such that each node $v_l^{(k)} \in V_{f^{(k)}}$ represents a component $f^{(k)}[l]$ of the feature vector. We can further define a weight function $w : E_{f^{(k)}} \to \mathbb{R}_0^+$ as the Euclidean distance

$$w(v_l^{(k)}, v_m^{(k)}) = \sqrt{(f^{(k)}[l] - f^{(k)}[m])^2}. \tag{3}$$

In order to assign a probability distribution to a graph $G_{f^{(k)}}$ *the steady state random walk*[3] is used. For each node $v_l^{(k)}$ of a graph, probability is defined as

$$p(v_l^{(k)}) = \frac{\deg(v_l^{(k)})}{\sum\limits_{v_m^{(k)} \in V_{f^{(k)}}} \deg(v_m^{(k)})}, \tag{4}$$

where

$$\deg(v_l^{(k)}) = \sum_{v_m^{(k)} \in V_{f^{(k)}}} w(v_l^{(k)}, v_m^{(k)}) \tag{5}$$

is the degree of a node. It is visible that (4) truly meets the requirements for the definition of probability, i.e. $p(v_l^{(k)}) \geq 0, \forall v_l^{(k)} \in V_{f^{(k)}}$ and $\sum_{v_l^{(k)} \in V_{f^{(k)}}} p(v_l^{(k)}) = 1$.

It therefore remains to determine how does one feature $f^{(k)}$ correspond to another feature $f^{(l)}$. As we have a formulation to obtain probability distributions for features, we can now evaluate the Jensen-Shannon divergence

$$\begin{aligned} JSD(p(f^{(k)}), p(f^{(l)})) =& H_S\left(\frac{p(f^{(k)}) + p(f^{(l)})}{2}\right) \\ &- \frac{H_S(p(f^{(k)})) + H_S(p(f^{(l)}))}{2}, \end{aligned} \tag{6}$$

where

$$H_S(p(f^{(k)})) = -\sum_{v_m^{(k)} \in V_{f^{(k)}}} p(v_m^{(k)}) \log p(v_m^{(k)}). \tag{7}$$

is the Shannon entropy.

Equation (6) measures the difference between probability distributions and therefore, in [9] a feature similarity measure between the feature distributions was defined

$$FS(p(f^{(k)}), p(f^{(l)})) = \exp(-JSD(p(f^{(k)}), p(f^{(l)}))). \tag{8}$$

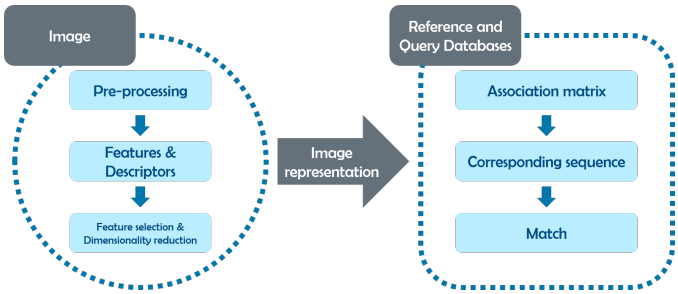


Figure 2: A general visual place recognition scheme. First, image representation is obtained by pre-processing, feature extraction and description, feature selection and dimensionality reduction. Then image matching process starts.

As a negative value of (6) measures the similarity between features, and exponential function is monotonically increasing, thus the order is preserved and (8) measures the similarity too.

3. Proposed Visual Place Recognition and Feature Selection

As stated in the introduction, design aspects of a visual place recognition system concern *image representation* and *image matching*, while its general scheme, shown in Fig. 2, is as follows:

1. Image representation is obtained by pre-processing, extraction of features and descriptors and, optionally, feature selection and dimensionality reduction
2. An association matrix is built and the corresponding sequence measures how much a reference image fits the query.

The design of the proposed system also follows the scheme depicted in Fig. 2. In Subsection 3.1 we review SeqSLAM algorithm for completeness and since it has influenced our approach. Thereafter, in Subsection 3.2 we present the proposed approach and how it differs from SeqSLAM. Subsection 3.3 presents the proposed formulation of the mutual information-based feature selection in the context of visual place recognition.

3.1. Sequential SLAM (SeqSLAM)

A naïve approach to image matching would exclusively measure either the distance or the similarity between a fixed query image $I_{q_i} \in \mathcal{Q}$ and $I_{d_j} \in \mathcal{D}, \forall\, d_j \in \{1, \ldots, |\mathcal{D}|\}$. For example, when measuring the similarity, the most appropriate match for I_{q_i} would be

$$I_{d_j}^* = \underset{I_{d_j}}{\operatorname{argmax}}\{s_{I_{q_i}, I_{d_j}}\}, \tag{9}$$

where

$$s_{I_{q_i}, I_{d_j}} = \frac{I_{q_i}^T I_{d_j}}{\|I_{q_i}\| \|I_{d_j}\|} \tag{10}$$

is the cosine similarity. However, Naseer et al. [36] noticed that "matching images just according to the best similarity score produces considerable false positives [...]".

In contrast to the naïve approach, SeqSLAM considers the local neighborhood around the fixed image pair $(I_{q_i}, I_{d_j}) \in \mathcal{Q} \times \mathcal{D}$ and this way we gain more information about the location. A neighborhood is obtained as a linear sequence centered in (I_{q_i}, I_{d_j}). Let $\hat{D}$ denotes the *difference matrix* where $\hat{D}[j, i]$ is the sum of absolute differences between I_{q_i} and I_{d_j} and let d_s denotes the number of image matches in the neighborhood (also called the *sequence length*). Then the correspondence between I_{q_i} and I_{d_j} can be measured as

$$S_{j,i} = \min_{v} \sum_{t=i-\lfloor \frac{d_s}{2} \rfloor}^{i+\lfloor \frac{d_s}{2} \rfloor} \hat{D}[k, t], \tag{11}$$

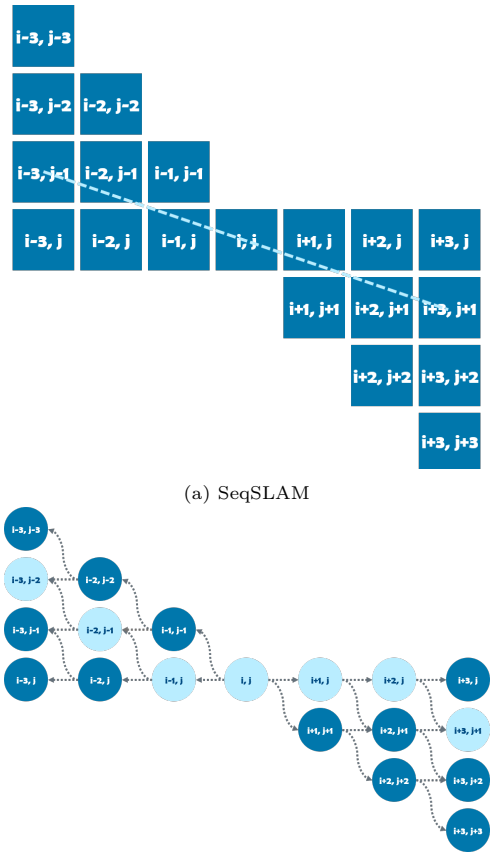
where $k = j + v(t - i)$ and v is the velocity by which we minimize (11). The lower $S_{j,i}$ the better the correspondence between I_{q_i} and I_{d_j}.

As mentioned, $\hat{D}$ represents the matrix of differences between image representations. In the case of SeqSLAM, *downsampled, human-readable* image representation has been used, although it is agnostic in terms of representation (e.g. in this paper we evaluate SeqSLAM using feature maps extracted from a DCNN). Moreover, it is also agnostic in terms of the measure between image representations (e.g. sum of absolute differences can be replaced by cosine similarity). Besides the original algorithm that measures the association as a linear sequence of a minimal weight, other variants have been introduced called the *cone-based method* and the *hybrid method* [48]. In the cone-based method for SeqSLAM, the association between images is as good as the number of minimal differences for each query image that fall in a cone, while hybrid method is, as the name suggests, a blend between the original and the cone-based method.

Indices k and t from (11) form a linear sequence centered at the indices pair (i, j). One can recognize that the expression $k = j + V(t - i)$ is truly the basic equation for the position of an object moving at constant velocity V. If the vehicle drives both query and reference traversals with linear correlation in the acceleration (e.g. if the vehicle does not accelerate at all during both traversals), then linear sequences will be visually manifested in the difference matrix and this is the scenario that SeqSLAM perfectly fits. However, vehicles in general tend to move with different accelerations and therefore we need an algorithm that can capture also this kind of scenario.

3.2. Not Only Sequential SLAM (NOSeqSLAM)

The proposed algorithm, in turn, considers any kind of correlation between indices – not only the linear ones. In similar manner as SeqSLAM, we measure the association between each query and reference image, but instead of using equation that yields linear sequences in the difference matrix, we employ *graph theory*. For illustration of similarities and differences between the two methods, please confer Fig. 3.



(a) SeqSLAM

(b) NOSeqSLAM

Figure 3: (a) SeqSLAM searches for the optimal linear sequence passing through (i, j) while (b) NOSeqSLAM searches for the optimal single-source shortest path from the root (i, j) to the left subgraph and to the right subgraph. Hence, SeqSLAM considers those sequences of matches whose indices are one after the other linearly correlated, while NOSeqSLAM considers any kind of correlation between indices.

Instead of the difference matrix $\hat{D}$ we build an *association matrix* $A \in [0, 1]^{|\mathcal{D}| \times |\mathcal{Q}|}$ such that

$$A[j, i] = s_{I_{q_i}, I_{d_j}}, \forall (I_{q_i}, I_{d_j}) \in \mathcal{Q} \times \mathcal{D}. \tag{12}$$

For a fixed indices pair (i, j) we construct a directed acyclic graph (DAG) $G_{(i,j)} = (V_{(i,j)}, E_{(i,j)})$ rooted at node (i, j) and iteratively expand this graph until the depth of $\lfloor \frac{d_s}{2} \rfloor$ is reached in a way that it is expanded in the direction of the previous query and reference indices (*left wing*) and in the direction of the following query and reference indices (*right wing*). Each non-root and non-leaf node has η_{exp} children, e.g., DAG shown in Fig. 3(b) is constructed with the parameters $d_s = 7$ and $\eta_{exp} = 2$. The corresponding weight function $w : E_{(i,j)} \to [0, 1]$ is defined as

$$w\left((k, l), (m, n)\right) = 1 - A[n, m]. \tag{13}$$

As $A[j, i]$ measures how much does I_{q_i} fit I_{d_j} and vice versa, we can also obtain the measure of the difference be-

Algorithm 1 On-the-fly relaxation

```
for i_off = 1 to ⌊d_s/2⌋ do
    for j_off = 0 to i_off · (η_exp − 1) do
        i_l = i − i_off
        j_l = j − j_off
        if i_l ∈ {0, ..., |Q| − 1} and j_l ∈ {0, ..., |D| − 1} then
            for each (i_{p_l}, j_{p_l}) ∈ predecessors(i_l, j_l) do
                relax((i_{p_l}, j_{p_l}), (i_l, j_l))
            end for
        end if
        i_r = i + i_off
        j_r = j + j_off
        if i_r ∈ {0, ..., |Q| − 1} and j_r ∈ {0, ..., |D| − 1} then
            for each (i_{p_r}, j_{p_r}) ∈ predecessors(i_r, j_r) do
                relax((i_{p_r}, j_{p_r}), (i_r, j_r))
            end for
        end if
    end for
end for
```

tween them by subtracting 1 with $A[j,i]$. Naseer et al. [35] use $1/s_{i_{q_i},i_{d_j}}$ for dissimilarity measure, but we consider (13) more numerical friendly and this equation in the context of an optimal sequence tells us that a node (i,j) is suitable for a sequence proportionally to its similarity measure.

Once DAG $G_{(i,j)}$ is constructed we measure the corresponding association between I_{q_i} and I_{d_j} as

$$S_{j,i} = A[j,i] + \sum_{(k,l)\in l^*_{(i,j)}} A[l,k] + \sum_{(k,l)\in r^*_{(i,j)}} A[l,k], \quad (14)$$

where $l^*_{(i,j)}$ and $r^*_{(i,j)}$ are the shortest paths in the left and right subgraph from the root to some leaf, respectively. In general, the *single source shortest path* problem can be solved either by using the Bellman-Ford algorithm [5] with the running time of $\Theta(|V||E|)$, or by using the Dijkstra algorithm [14] with the running time of $\Theta(|V|\lg|V| + |E|)$ when weights are positive [8]. However, when a graph is specifically a DAG, we can use the topological sort [25] which runs in $\Theta(|V|+|E|)$. Moreover, as we have to build each graph iteratively, we can do the relaxation on-the-fly as nodes are expanded and accumulate weights simultaneously. Although the topological sort has the lowest asymptotic running time, we developed a novel on-the-fly relaxation algorithm listed in Algorithm 1 which considerably lowered the running time of the NOSeqSLAM algorithm that is listed in Algorithm 2. Later, in Section 4 we will compare their running times which turn affect the running time of NOSeqSLAM.

Considering image representation, DCNNs can be used as *feature maps* extractors, since their convolutional layers are intended exactly for this purpose. As befits to the state-of-the-art, we also choose to use image representations extracted from deep convolutional neural networks. In general, regardless of the used image representation,

Algorithm 2 NOSeqSLAM

for each $(I_{q_i}, I_{d_j}) \in \mathcal{Q}\times\mathcal{D}$ **do**

$G_{(i,j)} = DAG(I_{q_i}, I_{d_j}, d_s, \eta_{exp})$

$l^*_{(i,j)} = \min_m SP(G_{(i,j)}, (I_{q_i}, I_{d_j}), (I_{q_{i-\lfloor \frac{d_s}{2} \rfloor}}, I_{d_m}))$

$r^*_{(i,j)} = \min_m SP(G_{(i,j)}, (I_{q_i}, I_{d_j}), (I_{q_{i+\lfloor \frac{d_s}{2} \rfloor}}, I_{d_m}))$

$S_{j,i} = A[j,i] + \sum_{(k,l)\in l^*_{(i,j)}} A[l,k] + \sum_{(k,l)\in r^*_{(i,j)}} A[l,k]$

end for

DAG - directed acyclic graph
SP - shortest path

we can apply the feature selection procedure presented in the following subsection which in turn can even further improve the results.

3.3. Feature Selection for Visual Place Recognition

We formulate feature selection method for visual place recognition according to the work presented in [9] where feature selection is performed for the sake of a classification problem. Therefore, we modify this formulation, and to the best of our knowledge, this is the first application to the visual place recognition problem.

Lets build a matrix $\mathcal{Q}_M \in \mathbb{R}^{|\mathcal{Q}|\times K}$ of image representations from the database $\mathcal{Q}$ as

$$\mathcal{Q}_M = \begin{bmatrix} I_{q_1} \\ I_{q_2} \\ \vdots \\ I_{q_{|\mathcal{Q}|}} \end{bmatrix} = \begin{bmatrix} I_{q_{11}} & I_{q_{12}} & \dots & I_{q_{1K}} \\ I_{q_{21}} & I_{q_{22}} & \dots & I_{q_{2K}} \\ \vdots & \vdots & \ddots & \vdots \\ I_{q_{|\mathcal{Q}|1}} & I_{q_{|\mathcal{Q}|2}} & \dots & I_{q_{|\mathcal{Q}|K}} \end{bmatrix}. \quad (15)$$

For each query image I_{q_i} we assume that there exists at least one "ground-truth" image from the reference. Ground-truth images are images that represent the same visual place as the one in the query image, either being hand-labeled or obtained from GPS data. We can choose the most similar ground-truth image $I_{d_{i^*}}$ for each query image I_{q_i} according to

$$I_{d_{i^*}} = \underset{I_{d_j}\in GT(I_{q_i})}{\operatorname{argmax}} \; s_{I_{q_i},I_{d_j}}, \quad (16)$$

and build a matrix $\mathcal{D}_M \in \mathbb{R}^{|\mathcal{Q}|\times K}$ defined as

$$\mathcal{D}_M = \begin{bmatrix} I_{d_{1^*}} \\ I_{d_{2^*}} \\ \vdots \\ I_{d_{|\mathcal{Q}|^*}} \end{bmatrix} = \begin{bmatrix} I_{d_{1^*1}} & I_{d_{1^*2}} & \dots & I_{d_{1^*K}} \\ I_{d_{2^*1}} & I_{d_{2^*2}} & \dots & I_{d_{2^*K}} \\ \vdots & \vdots & \ddots & \vdots \\ I_{d_{|\mathcal{Q}|^*1}} & I_{q_{|\mathcal{Q}|^*2}} & \dots & I_{q_{|\mathcal{Q}|^*K}} \end{bmatrix}. \quad (17)$$

What corresponds to the term *feature* in [9] are columns of matrices (15) and (17) and accordingly we define features in our case as

$$f^{(k)}_{\mathcal{Q}} = \mathcal{Q}_M[:,k], \; f^{(k)}_{\mathcal{D}} = \mathcal{D}_M[:,k]. \quad (18)$$

where $f^{(k)}_{\mathcal{D}}$ specifically called *the target feature*.

A good feature is similar in both query and reference images and therefore we measure the quality of the particular k-th feature as

$$q(k) = FS(p(f_{\mathcal{Q}}^{(k)}), p(f_{\mathcal{D}}^{(k)})). \tag{19}$$

When (19) is evaluated $\forall k \in \{1, \ldots, K\}$ and feature qualities $\{q(1), \ldots, q(K)\}$ are obtained, we pick all those k' indices such that $q(k') \geq c$ where c is a p-th quantile of $\{q(1), \ldots, q(K)\}$. Therefore, in a vector that represents an image, we pick these components that have such index k'.

4. Experimental Results

In this section we present the experimental evaluation of our approach. First, we present the evaluation datasets, followed by image representation analysis, and finally, comparison of the proposed algorithm with state-of-the-art – both with and without feature selection.

4.1. Evaluation datasets

For evaluation purposes we used the Bonn and Freiburg datasets that are part of the publicly available implementation of [51] containing routes driven in an urban area. The main challenge stems from condition variance, while view-point variance is moderate, because sequences were captured by a vehicle driving on the road.

When driven for the first time (and therefore saved as $\mathcal{D}$), a route in Bonn is captured in the evening, while for the second time (saved as $\mathcal{Q}$) it is captured on a gloomy day (Fig. 4(a), Fig. 4(b)). Due to different illumination sources (daylight vs. streetlight), images in $\mathcal{Q}$ and $\mathcal{D}$ differ significantly. Although there is no record about the distance traveled throughout this route, by observing sequences alongside the route and from the cardinality ($|\mathcal{D}| = 488$, $|\mathcal{Q}| = 544$) we can assert that the route is 1–2 km long.

A query route driven in Freiburg was captured on a sunny day, while the reference database contains images from a sunny winter afternoon ((Fig. 4(c), Fig. 4(d))). Although both traversals were captured in daylight, illumination differs, and the snow appears in the reference traversal. In [35] it is stated that this route is about 3 km long. View-point variance is not accentuated in either of the datasets, because the vehicle stays in the same lane for the both the query and the reference traversal.

Both datasets come with the ground-truth files where for each query image there exists at least one match with an image from the reference database. The same way as described in [50], we use the localization radius of ± 3 indices, i.e., a match is considered *a true positive* if it deviates up to 3 indices from the ground-truth. On the other hand, *score thresholding* [48] is used in order to reject poor match proposals.

(a) $\mathcal{Q}$ (b) $\mathcal{D}$

(c) $\mathcal{Q}$ (d) $\mathcal{D}$

Figure 4: Significant condition variance in Bonn dataset (a),(b) and Freiburg dataset (c),(d). View-point variance is not accentuated because the vehicle stays in the same lane.

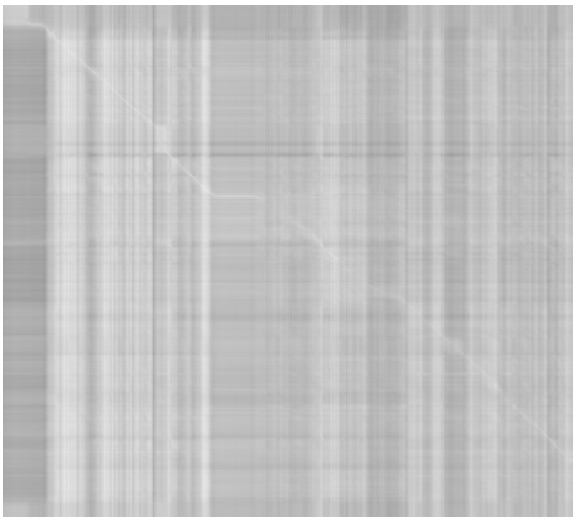
4.2. Image representation with DCNN feature maps

Given the evaluation datasets, we analyzed different image representations. As a qualitative measure, we can observe the association matrix for various representations in Fig. 5, where, as expected, the more accentuated contrast indicates a good representation.

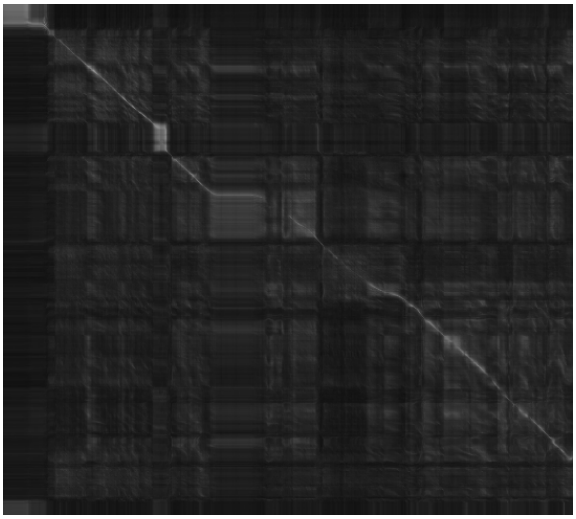
Representation with HOG yields poor contrast, probably due to the design of the technique which exclusively examines orientations of the image gradients. Besides HOG, in our experiments we also evaluated feature maps extracted from neural networks, beginning with the AlexNet architecture. First, we extracted features from the vanilla ImageNet-trained AlexNet architecture concluding `conv3` and `conv4` layers as [45] reported their suitability. Moreover, the same authors claim that feature maps extracted from a network trained on a scene-centric dataset perform better then the ones extracted from a network trained on an object-centric dataset. Therefore, we also extracted feature maps from AlexNet trained on Places365 [54].

Thereafter, we extracted feature maps from the 10th convolutional layer of the OverFeat architecture. Besides classification, it is also used for localization and detection tasks. Moreover, we decided to perform our experiments on two other state-of-the-art architectures - ResNet18 and ResNet50 [22] - where each architecture is trained on both ImageNet and Places365. We extracted features from their last convolutional layers. In total, we used 10 various image representations.

Next, we incorporated the aforementioned technique for feature selection into our experiments. First, training was performed on the Freiburg dataset, and when the appropriate feature qualities were obtained, we further used these qualities in order to select features for evaluation on the Bonn dataset. Vice versa, the training was performed on the Bonn dataset in order to evaluate on the Freiburg



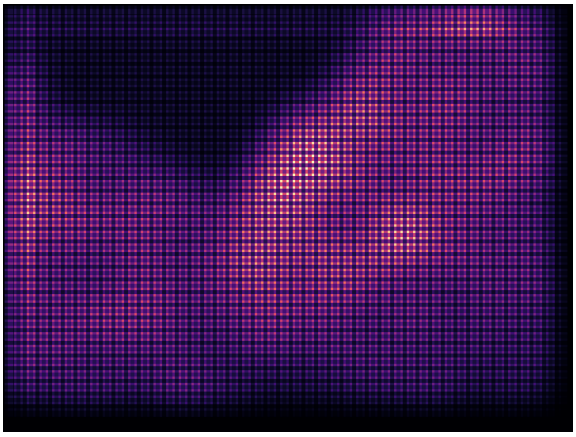
(a) HOG



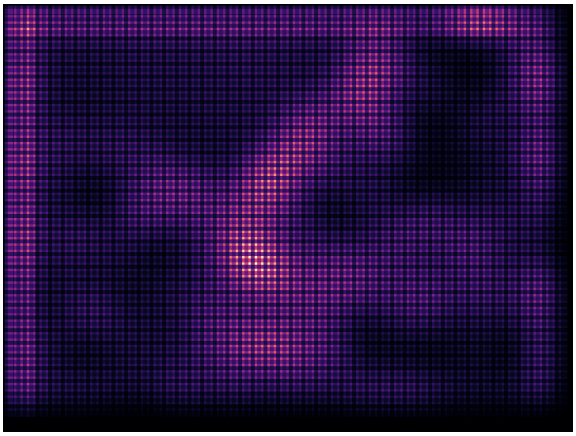
(b) OverFeat

Figure 5: Plotting the association matrix A with different image representations reflects the quality of image representation. The more accentuated is the contrast, the better.

(a) Original image



(b) Feature maps without feature selection



(c) Feature maps with feature selection ($p = 0.95$)

Figure 6: The effect of feature selection. In (b) activations are uniformly distributed, while in (c) activations are more sparsely distributed and focused on discriminative objects (e.g. weak activations for moving objects such as the cyclist, and simultaneously strong activations on and around the road). Feature maps from `conv3` AlexNet architecture are shown in this figure.

dataset with selected features. We find this approach to be the most transparent in contrast to both training and evaluating selected features on the same dataset. Even though the full feature maps are of high dimensions, we did not apply any dimensionality reduction techniques in order to concentrate solely on the visual place recognition performance of the proposed and other algorithms, as well as on the feature selection performance.

Alongside the quantitative performance observation in the upcoming subsection, it is important to examine how does feature selection truly select features, i.e., which features survive after the selection is performed? This question can be answered by plotting feature maps as in Fig. 6. Without feature selection (Fig. 6(b)), feature maps activate uniformly throughout the spatial locations taking into account both relevant and irrelevant objects. After feature selection (Fig. 6(c)), activations are more sparsely distributed throughout spatial locations. It is visible that feature maps are not focused anymore on moving objects

such as the cyclist. On the other hand, feature maps are more concentrated on the road. The footage of an entire traversal can be found online[2].

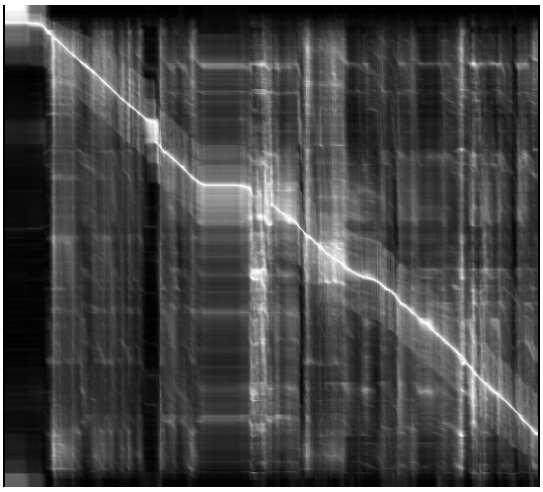
4.3. NOSeqSLAM comparison to other algorithms

In this section we conducted experiments that measure both the qualitative and quantitative performance of the sequence-based methods: NOSeqSLAM, SeqSLAM and cone-based SeqSLAM. Additionally, we will quantitatively compare the method of Vysotska et al. [51] with the aforementioned methods. Unlike the sequence-based methods, this method is not a sequence-based, i.e., it does not depend on d_s. Qualitative performance consists of observing the impact of algorithm has on the association matrix A – the higher contrast in the matrix A the better place distinction is achieved. Quantitative peformance is further examined by plotting the precision-recall curves, computing the area under a curve (AUC) and recall at 100% precision (R@100%P) scores.

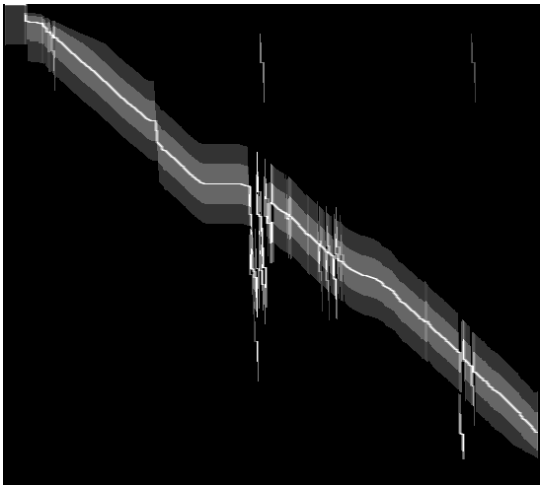
By analyzing Figs. 7(a) and 7(c), we can see that both SeqSLAM and NOSeqSLAM perform similarly in situations when d_s is small, because in this scenario the shortest paths tend to mimic linear sequences. The cone-based SeqSLAM (Fig. 7(b)) produces the best contrast, clearly indicating appropriate matches, although the route at several places is more disconnected. For $d_s = 51$ in Fig. 8(a) it is visible that SeqSLAM leaves linear "traces" throughout the association matrix, the same as its cone-based counterpart (Fig. 8(b)). Moreover, cone-based SeqSLAM algorithm enhances the contrast even more, so we could hypothesize that it performs better. NOSeqSLAM does not leave linear traces in the difference matrix, but in turn, due to its not only linear design, "smudges" the matrix a bit. As d_s increases, each algorithm achieves better distinctiveness, as can be seen from Figs. 7 and 8.

Quantitatively, we obtained the best result for each sequence-based method among image representations in terms of AUC and R@100% scores with and without feature selection, and then we grouped the results by the sequence length d_s. In Fig. 9(a), where the results are sorted by AUC, NOSeqSLAM achieves the best performance on the Bonn dataset for the major part of the curve, while SeqSLAM outperforms the cone-based SeqSLAM. Moreover, we observe that feature selection improves the performance for each of the sequence-based methods.

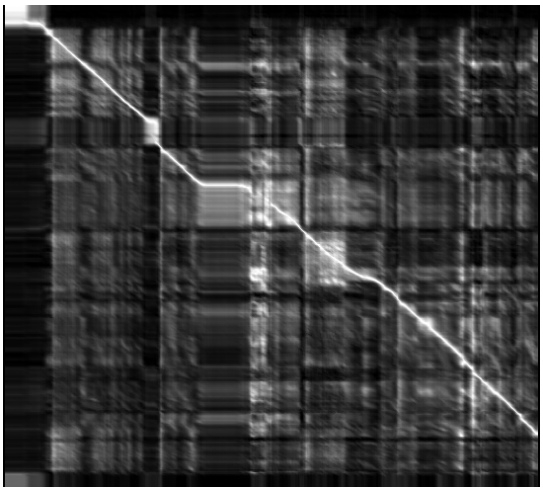
On the other hand, the method[3] by Vysotska et al. [51] achieves good performance interweaving with SeqSLAM. However, the sequence-based methods are unable to achieve 100% recall due to their design where the first $\lfloor \frac{d_s}{2} \rfloor$ and last $\lfloor \frac{d_s}{2} \rfloor$ query images cannot be paired. This negative effect is diminished when the query database is large enough because then the number of unpaired queries



(a) SeqSLAM



(b) SeqSLAM (cone)

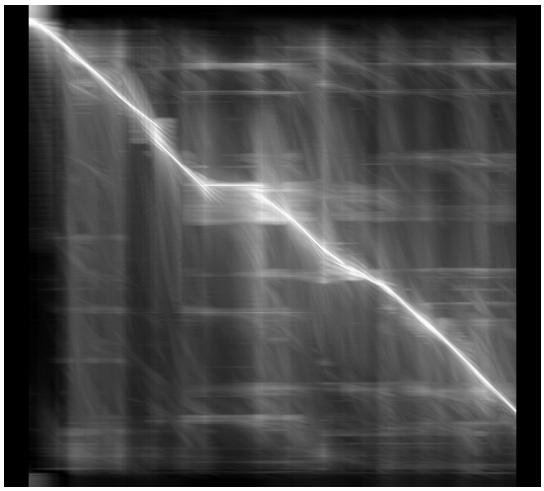


(c) NOSeqSLAM

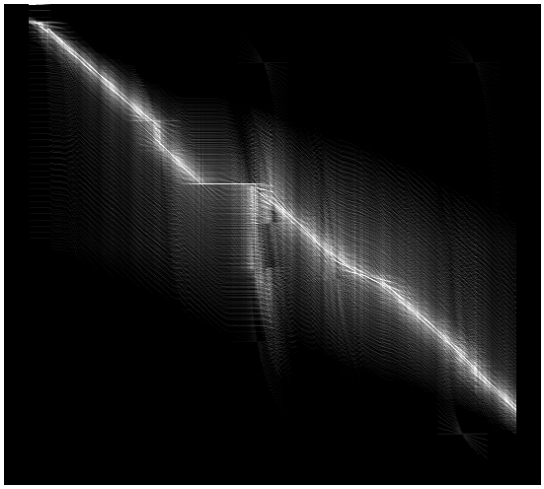
Figure 7: Qualitative performance of (a) SeqSLAM, (b) cone-based SeqSLAM and (c) NOSeqSLAM on the Bonn dataset for $d_s = 5$. (a) and (c) perform similarly because shortest path mimics linear sequence, while (b) more clearly indicates the route hypothesis.

[2] `https://youtu.be/MtaNUlZWtRg`

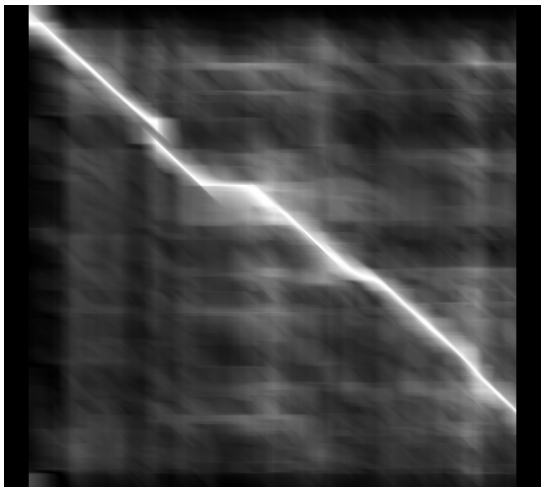
[3] No feature selection was applied. Feature maps extracted from the OverFeat `conv10` layer.



(a) SeqSLAM

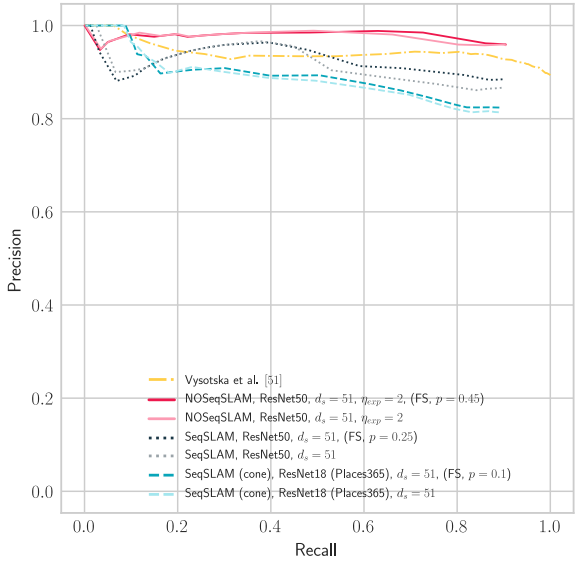


(b) SeqSLAM (cone)

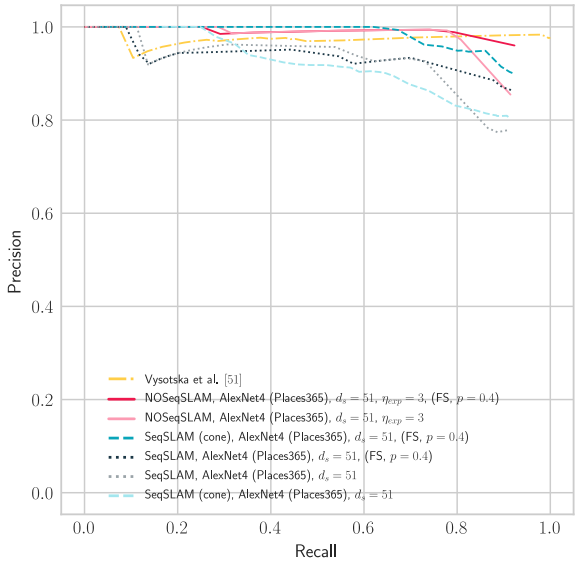


(c) NOSeqSLAM

Figure 8: Qualitative performance of (a) SeqSLAM, (b) cone-based SeqSLAM and (c) NOSeqSLAM on the Bonn dataset for $d_s = 51$. As the sequence length d_s increases, distinctiveness between algorithms is more pronounced.



(a) Bonn



(b) Freiburg

Figure 9: Precision-recall curves for (a) Bonn and (b) Freiburg. For each sequence-based method we pick the best results among image representations both with and without feature selection. For completeness, we plot a curve for the method by Vysotska et al. [51] too.

is negligible in comparison with $|\mathcal{Q}|$. A more detailed examination about this topic is provided in [39]. In Fig. 9(b) which represents the performance on the Freiburg dataset, we see that feature selection improves the performance moderately at the end of the curves for NOSeqSLAM and SeqSLAM while for cone-based SeqSLAM the improvement is more accentuated.

In Tables 1 and 2 the best AUC and R@100% scores are listed for the Bonn dataset. We omit the method by Vysotska at al. [51] (although we will reports its performance in this paragraph), as it does not depend on d_s and therefore we can not group it in tables. By measuring AUC, ResNet

Table 1: AUC scores of SeqSLAM, cone-based SeqSLAM and NOSeqSLAM for the Bonn dataset with feature selection (FS column with percentile value) and without (–).

d_s	Alg.	Repr.	Dim.	η_{exp}	FS	Score
31	SC	O (conv10)	153600	–	–	0.83521
	SC	O (conv10)	46080	–	0.7	0.83683
	S	R18 (conv17)	200192	–	–	0.89153
	S	R18 (conv17)	140134	–	0.3	0.90005
	N	R50 (conv49)	800768	2	–	0.91501
	N	R50 (conv49)	680653	2	0.15	**0.91890**
43	SC	R18 (conv17, P)	200192	–	–	0.81094
	SC	R18 (conv17, P)	190182	–	0.05	0.81692
	S	R50 (conv49)	800768	–	–	0.87506
	S	R50 (conv49)	40038	–	0.95	0.88787
	N	R50 (conv49)	800768	2	–	0.89647
	N	R50 (conv49)	120116	2	0.85	**0.89798**
51	SC	R18 (conv17, P)	200192	–	–	0.78836
	SC	R18 (conv17, P)	180172	–	0.1	0.79570
	S	R50 (conv49)	800768	–	–	0.82185
	S	R50 (conv49)	600576	–	0.25	0.83078
	N	R50 (conv49)	800768	2	–	0.88189
	N	R50 (conv49)	440424	2	0.45	**0.88592**

N - NOSeqSLAM, S - SeqSLAM, SC - cone-based SeqSLAM
A - AlexNet, O - OverFeat, R18 - ResNet18, R50 - ResNet50
P - trained on Places365

Table 2: R@100%P scores of SeqSLAM, cone-based SeqSLAM and NOSeqSLAM for the Bonn dataset with feature selection (FS column with percentile value) and without (–).

d_s	Alg.	Repr.	Dim.	η_{exp}	FS	Score
31	N	A (conv4)	360448	2	–	0.01103
	S	R18 (conv17)	200192	–	–	0.10846
	S	R18 (conv17)	140134	–	0.3	0.13419
	SC	R50 (conv49, P)	800768	–	–	0.13603
	SC	R50 (conv49, P)	360347	–	0.55	0.14522
	N	A (conv4)	36045	2	0.9	**0.14706**
43	S	A (conv4)	360448	–	–	0.01103
	N	A (conv4)	360448	2	–	0.04044
	SC	R18 (conv17)	200192	–	–	0.11029
	S	A (conv4)	54068	–	0.85	0.12684
	SC	R18 (conv17)	120116	–	0.4	0.12868
	N	A (conv4)	126157	2	0.65	**0.17647**
51	N	A (conv4)	360448	3	–	0.00000
	S	A (conv4)	360448	–	–	0.01654
	SC	R18 (conv17)	200192	–	–	0.11397
	SC	R18 (conv17)	150144	–	0.25	0.13603
	S	A (conv4)	54068	–	0.85	0.15257
	N	A (conv4)	36045	3	0.9	**0.17647**

N - NOSeqSLAM, S - SeqSLAM, SC - cone-based SeqSLAM
A - AlexNet, O - OverFeat, R18 - ResNet18, R50 - ResNet50
P - trained on Places365

Table 3: AUC scores of SeqSLAM, cone-based SeqSLAM and NOSeqSLAM for the Freiburg dataset with feature selection (FS column with percentile value) and without (–).

d_s	Alg.	Repr.	Dim.	η_{exp}	FS	Score
31	SC	A (conv4, P)	360448	–	–	0.63907
	SC	A (conv4, P)	216268	–	0.4	0.85604
	S	A (conv4, P)	360448	–	–	0.87314
	S	A (conv4, P)	216268	–	0.4	0.88080
	N	A (conv4, P)	360448	3	–	0.92429
	N	A (conv4, P)	216268	3	0.4	**0.93119**
43	SC	A (conv4, P)	360448	–	–	0.66569
	S	A (conv4, P)	360448	–	–	0.86373
	S	A (conv4, P)	216268	–	0.4	0.86775
	SC	A (conv4, P)	216268	–	0.4	0.87361
	N	A (conv4, P)	360448	3	–	0.91016
	N	A (conv4, P)	216268	3	0.4	**0.92177**
51	SC	A (conv4, P)	360448	–	–	0.67670
	S	A (conv4, P)	360448	–	–	0.84598
	S	A (conv4, P)	216268	–	0.4	0.85850
	SC	A (conv4, P)	216268	–	0.4	0.87534
	N	A (conv4, P)	360448	3	–	0.89907
	N	A (conv4, P)	216268	3	0.4	**0.91346**

N - NOSeqSLAM, S - SeqSLAM, SC - cone-based SeqSLAM
A - AlexNet, O - OverFeat, R18 - ResNet18, R50 - ResNet50
P - trained on Places365

Table 4: R@100%P scores of SeqSLAM, cone-based SeqSLAM and NOSeqSLAM for the Freiburg dataset with feature selection (FS column with percentile value) and without (–).

d_s	Alg.	Repr.	Dim.	η_{exp}	FS	Score
31	S	R50 (conv49)	800768	–	–	0.10207
	SC	A (conv4, P)	360448	–	–	0.23669
	S	R50 (conv49)	520499	–	0.35	0.41864
	SC	A (conv4, P)	216268	–	0.4	0.57101
	N	R50 (conv49, P)	800768	3	–	0.62574
	N	R50 (conv49, P)	480459	3	0.4	**0.63166**
43	S	A (conv4, P)	360448	–	–	0.05917
	SC	A (conv4, P)	360448	–	–	0.26036
	S	A (conv4, P)	18023	–	0.95	0.28107
	N	A (conv4, P)	360448	3	–	0.30325
	SC	A (conv4, P)	216268	–	0.4	0.58432
	N	A (conv4, P)	180224	3	0.5	**0.69822**
51	S	A (conv4, P)	360448	–	–	0.11243
	SC	A (conv4, P)	360448	–	–	0.24852
	N	A (conv4, P)	360448	3	–	0.28698
	S	A (conv4, P)	36045	–	0.9	0.32101
	SC	A (conv4, P)	216268	–	0.4	0.61982
	N	A (conv4, P)	162202	3	0.55	**0.74112**

N - NOSeqSLAM, S - SeqSLAM, SC - cone-based SeqSLAM
A - AlexNet, O - OverFeat, R18 - ResNet18, R50 - ResNet50
P - trained on Places365

architectures prevail as the best representations. With respect to the sequence length d_s, NOSeqSLAM outperforms in both scenarios with and without feature selection. The method by Vysotska et al. [51] achieves greater AUC score (0.94309) due to the problem of the maximum possible recall for the sequence-based methods [39]. By measuring R@100%P, NOSeqSLAM has the highest score, while the improvements with feature selection are more emphasized regardless of the used algorithm and representation while the method by Vysotska et al. [51] achieves poor R@100%P (0.06250). The same can be said for Tables 3 and 4 which list the results for the Freiburg dataset, although the prevailing architecture is AlexNet trained on Places365. Once again the method by Vysotska et al. [51] achieves greater AUC score (0.97471). When R@100%P is measured, NOSeqSLAM outperforms other sequence-based algorithms as well as the method by Vysotska et al. [51] (0.07396), while feature selection highlights the difference even more.

From a theoretical point of view, given a constructed association matrix A, SeqSLAM and cone-based SeqSLAM take $\Theta\left(|\mathcal{Q}||\mathcal{D}|d_s V_{steps}\right)$ asymptotic running times. NOSeqSLAM takes $\Theta\left(|\mathcal{Q}||\mathcal{D}|d_s^2\eta_{exp}^2\right)$ asymptotic running time because on-the-fly relaxation (Algorithm 1) takes $\Theta\left(d_s^2\eta_{exp}^2\right)$ asymptotic running time and must be executed for each image pair. In case of NOSeqSLAM, the topolog-

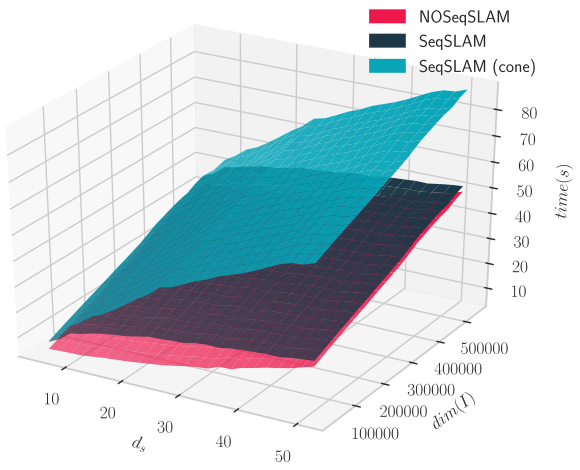


Figure 10: Running time as a function of the sequence length $d_s \in \{5, 7, \ldots, 51\}$ and the number of features $dim(I)$. Although SeqSLAM and cone-based SeqSLAM have better asymptotic times, NOSeqSLAM achieves lower running times for practical sequence lengths.

ical sort takes $\Theta\left(d_s^2 \eta_{exp} + \eta_{exp}^{d_s}\right)$ asymptotic running time; however, replacement of the topological sort with on-the-fly relaxation lowers the NOSeqSLAM running time approximately 200 times.

Empirically, running times were measured with a standard `i7@2.8GHz` laptop processor on the Bonn dataset for various sequence lengths $d_s \in \{5, 7, \ldots, 49, 51\}$ and the various number of features[4] $dim(I)$. Each running time includes: feature selection, the construction of the association matrix and then the evaluation of an algorithm. The results are shown in Fig. 10, from which we can see that NOSeqSLAM operates faster than SeqSLAM and cone-based SeqSLAM. On the other hand, SeqSLAM methods have better asymptotic times and therefore once when d_s is large enough, they will overtake NOSeqSLAM. However, we find that $d_s \in \{5, \ldots, 51\}$ are more than reasonable sequence lengths that consistently describe the neighborhood of a place and that longer sequences are most likely unnecessary.

5. Conclusion

In this paper we have proposed a method for visual place recognition based on directed acyclic graph matching and DCNNs, coupled with mutual information based feature selection. Due to the ability to account for not only linear correlations in the association matrix, we dubbed our approach NOSeqSLAM. In order to find the best subset of DCNN feature maps with minimal redundancy for visual place recognition, we proposed to form probability distributions on those features using steady-state random walk and leverage Jensen-Shannon divergence to rank features. We evaluated the proposed approach on two public datasets, namely the Bonn and Freiburg dataset, and compared it to SeqSLAM, cone-based SeqSLAM and the method by Vysotska et al. [51] For image representations HOG as well as feature maps from AlexNet, OverFeat, ResNet18 and ResNet50 were used.

Results suggest that image representation obtained from DCNNs offers better performance than the hand-crafted HOG. Furthermore, we did not notice significant difference when a neural network was trained on object-centric or scene-centric dataset – feature maps trained on an object-centric dataset performed exceptionally good for the task of visual place recognition on the tested datasets. AlexNet (`conv4`) has proved to be the best on the Freiburg dataset for both AUC and R@100%P. Moreover, it was the best on the Bonn dataset for R@100%P, while ResNet50 (`conv49`) prevailed in terms of AUC. When comparing NOSeqSLAM with regular and cone-based SeqSLAM, we have shown that our approach outperforms them on both datasets in terms of the AUC score, R@100%P score, and running time for practical sequence lengths. Moreover, it outperforms the method by Vysotska et al. [51] in terms of R@100%P. Since all the algorithms are *representation agnostic*, we have the freedom to choose the appropriate representation, thus by formulating mutual information-based feature selection specifically for visual place recognition and by selecting the feature percentile with the best score, we have shown that all the algorithms, and not just NOSeqSLAM, exhibit enhanced performance with the reduced feature set.

[4] Measurements taken on feature maps from the AlexNet `conv3` ranging from 27034 up to 540672 features.

6. References

[1] Supervised and Unsupervised Linear Learning Techniques for Visual Place Recognition in Changing Environments. *IEEE Transactions on Robotics*, 32(3):600–613, 2016.

[2] Relja Arandjelović, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. 70(5):641–648, nov 2015.

[3] Lu Bai, Luca Rossi, Horst Bunke, and Edwin R. Hancock. Attributed graph kernels using the Jensen-Tsallis q-differences. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014. ISBN 9783662448472. doi: 10.1007/978-3-662-44848-9_7.

[4] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded up robust features. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 3951 LNCS(4):404–417, 2006.

[5] Richard Bellman. On a routing problem. *Quarterly of Applied Mathematics*, 1958. ISSN 0033-569X. doi: 10.1090/qam/102435.

[6] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. BRIEF: Binary Robust Independent Elementary Features. In *Proceedings of the 11th European Conference on Computer Vision: Part IV*, ECCV'10, pages 778–792, Berlin, Heidelberg, 2010. Springer-Verlag.

[7] Zetao Chen, Adam Jacobson, Niko Sünderhauf, Ben Upcroft, Lingqiao Liu, Chunhua Shen, Ian Reid, and Michael Milford. Deep learning features at scale for visual place recognition. *Pro-*

ceedings - IEEE International Conference on Robotics and Automation, 1:3223–3230, 2017. ISSN 10504729.

[8] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition, 2009. ISBN 0262033844, 9780262033848.

[9] Lixin Cui, Lu Bai, Yue Wang, Xiao Bai, Zhihong Zhang, and Edwin R Hancock. P2P Lending Analysis Using the Most Relevant Graph-Based Features. pages 3–14. 2016.

[10] Lixin Cui, Lu Bai, and Edwin R Hancock. Fused Lasso for Feature Selection using Structural Information. (NeurIPS):1–10, 2019.

[11] Lixin Cui, Lu Bai, Zhihong Zhang, Yue Wang, and Edwin R. Hancock. Identifying the most informative features using a structurally interacting elastic net. *Neurocomputing*, 336:13–26, 2019. ISSN 18728286.

[12] Mark Cummins and Paul Newman. FAB-MAP: Probabilistic localization and mapping in the space of appearance. *International Journal of Robotics Research*, 27(6):647–665, 2008.

[13] Robert Cupec, Emmanuel Karlo Nyarko, Damir Filko, Andrej Kitanov, and Ivan Petrovi. Place recognition based on matching of planar surfaces and line segments. *International Journal of Robotics Research*, 34(4-5):674–704, 2015. ISSN 17413176. doi: 10.1177/0278364914548708.

[14] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1959. ISSN 0029599X. doi: 10.1007/BF01386390.

[15] Anh-Dzung Doan, Yasir Latif, Tat-Jun Chin, Yu Liu, Thanh-Toan Do, and Ian Reid. Scalable Place Recognition Under Appearance Change for Autonomous Driving. 2019.

[16] Francois Fleuret. Fast binary feature selection with conditional mutual information. *Journal of Machine Learning Research*, 2004. ISSN 15337928.

[17] D. Galvez-López and J. D. Tardos. Bags of Binary Words for Fast Place Recognition in Image Sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, oct 2012. ISSN 1552-3098.

[18] Sourav Garg, Niko Suenderhauf, and Michael Milford. Lost? appearance-invariant place recognition for opposite viewpoints using visual semantics. 2018. ISSN 1871756X.

[19] Sourav Garg, Madhu Babu, Thanuja Dharmasiri, Stephen Hausler, Niko Suenderhauf, Swagat Kumar, Tom Drummond, and Michael Milford. Look no deeper: Recognizing places from opposing viewpoints under varying scene appearance using single-view depth estimation. 2019.

[20] Stephen Hausler, Adam Jacobson, and Michael Milford. Feature Map Filtering: Improving Visual Place Recognition with Convolutional Calibration. 2018.

[21] Stephen Hausler, Adam Jacobson, and Michael Milford. Filter Early, Match Late: Improving Network-Based Visual Place Recognition. pages 2–9, 2019.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. 19(2):107–117, dec 2015.

[23] Gunawan Herman, Bang Zhang, Yang Wang, Getian Ye, and Fang Chen. Mutual information-based method for selecting informative feature sets. *Pattern Recognition*, 46(12):3315–3327, 2013. ISSN 00313203.

[24] H. Jegou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid. Aggregating Local Image Descriptors into Compact Codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1704–1716, sep 2012. ISSN 0162-8828.

[25] A. B. Kahn. Topological sorting of large networks. *Communications of the ACM*, 1962. ISSN 15577317. doi: 10.1145/368996.369025.

[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[27] Huu Le, Tuan Hoang, and Michael Milford. BTEL: A Binary Tree Encoding Approach for Visual Localization. 2019.

[28] Dahua Lin and Xiaoou Tang. Conditional infomax learning: An integrated framework for feature extraction and fusion. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2006. ISBN 3540338322.

[29] D.G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, pages 1150–1157 vol.2. IEEE, sep 1999. ISBN 0-7695-0164-8.

[30] Shuangge Ma, Xiao Song, and Jian Huang. Supervised group Lasso with applications to microarray data analysis. *BMC Bioinformatics*, 2007. ISSN 14712105.

[31] Jurica Maltar, Ivan Marković, and Ivan Petrović. NOSeqSLAM: Not only Sequential SLAM. In Manuel F Silva, José Luís Lima, Luís Paulo Reis, Alberto Sanfeliu, and Danilo Tardioli, editors, *Robot 2019: Fourth Iberian Robotics Conference*, pages 179–190, Cham, 2020. Springer International Publishing. ISBN 978-3-030-35990-4.

[32] Michael J. Milford and Gordon F. Wyeth. SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 1643–1649. IEEE, may 2012.

[33] Hans Moravec. Obstacle avoidance and navigation in the real world by a seeing robot rover. Technical Report CMU-RI-TR-80-03, Carnegie Mellon University, Pittsburgh, PA, September 1980.

[34] Raul Mur-Artal, J. M.M. Montiel, and Juan D. Tardos. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[35] Tayyab Naseer, Luciano Spinello, Wolfram Burgard, and Cyrill Stachniss. Robust visual robot localization across seasons using network flows. *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2564–2570, 2014.

[36] Tayyab Naseer, Michael Ruhnke, Cyrill Stachniss, Luciano Spinello, and Wolfram Burgard. Robust visual SLAM across seasons. *IEEE International Conference on Intelligent Robots and Systems*, 2015-Decem:2529–2535, 2015. ISSN 21530866.

[37] Tayyab Naseer, Wolfram Burgard, and Cyrill Stachniss. Robust Visual Localization Across Seasons. *IEEE Transactions on Robotics*, 34(2):289–302, apr 2018.

[38] Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005. ISSN 01628828.

[39] E. Pepperell, P. I. Corke, and M. J. Milford. All-environment visual place recognition with smart. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1612–1618, May 2014.

[40] Fabio Ramos, Ben Upcroft, Suresh Kumar, and Hugh Durrant-Whyte. A Bayesian approach for place recognition. *Robotics and Autonomous Systems*, 60(4):487–497, apr 2012. ISSN 09218890.

[41] Ali Sharif Razavian, Josephine Sullivan, Stefan Carlsson, and Atsuto Maki. Visual Instance Retrieval with Deep Convolutional Networks. (June 2017), 2014. ISSN 2186-7364.

[42] Edward Rosten, Reid Porter, and Tom Drummond. Faster and Better: A Machine Learning Approach to Corner Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:105–119, 2010.

[43] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Robert Fergus, and Yann Lecun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *International Conference on Learning Representations (ICLR2014), CBLS, April 2014*, 2014.

[44] S. M. Siam and H. Zhang. Fast-seqslam: A fast appearance based place recognition algorithm. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5702–5708, May 2017.

[45] Niko Sünderhauf, Sareh Shirazi, Feras Dayoub, Ben Upcroft, and Michael Milford. On the performance of ConvNet features for place recognition. *IEEE International Conference on Intelligent Robots and Systems*, 2015-Decem:4297–4304, 2015. ISSN 21530866.

[46] Niko Sünderhauf, Sareh Shirazi, Adam Jacobson, Feras Dayoub,

Edward Pepperell, Ben Upcroft, and Michael Milford. Place Recognition with ConvNet Landmarks: Viewpoint-Robust, Condition-Robust, Training-Free. In *Robotics: Science and Systems XI*. Robotics: Science and Systems Foundation, jul 2015. ISBN 9780992374716.

[47] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going Deeper with Convolutions. sep 2014.

[48] Ben Talbot, Sourav Garg, and Michael Milford. OpenSeqSLAM2.0: An Open Source Toolbox for Visual Place Recognition Under Changing Conditions. *IEEE Robotics and Automation Letters*, 1(1):213–220, apr 2018.

[49] Robert Tibshirani. Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 1996. ISSN 0035-9246.

[50] 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 *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2774–2779, May 2015.

[51] Olga Vysotska and Cyrill Stachniss. Lazy Data Association For Image Sequences Matching Under Substantial Appearance Changes. *IEEE Robotics and Automation Letters*, 1(1):213–220, 2016. ISSN 23773766.

[52] Peng Yin, Rangaprasad Arun Srivatsan, Yin Chen, Xueqian Li, Hongda Zhang, Lingyun Xu, Lu Li, Zhenzhong Jia, Jianmin Ji, and Yuqing He. MRS-VPR: a multi-resolution sampling based global visual place recognition method. 2019.

[53] Zhihong Zhang, Yiyang Tian, Lu Bai, Jianbing Xiahou, and Edwin Hancock. High-order covariate interacted Lasso for feature selection. *Pattern Recognition Letters*, 2017. ISSN 01678655.

[54] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[55] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 2005. ISSN 13697412.