NOSeqSLAM: Not only Sequential SLAM

Jurica Maltar[1], Ivan Marković[2], and Ivan Petrović[2]

[1] Department of Mathematics, University of Osijek, Croatia
[2] University of Zagreb, Faculty of Electrical Engineering and Computing, Croatia
jmaltar@mathos.hr, {ivan.markovic, ivan.petrovic}@fer.hr

Abstract. The essential property that every autonomous system should have is the ability to localize itself, i.e., to reason about its location relative to measured landmarks and leverage this information to consistently estimate vehicle location through time. One approach to solving the localization problem is *visual place recognition*. Using only camera images, this approach has the following goal: during the second traversal through the environment, using only current images, find the match in the database that was created during a previously driven traversal of the same route. Besides the image representation method – in this paper we use feature maps extracted from the OverFeat architecture – for visual place recognition it is also paramount to perform the scene matching in a proper way. For autonomous vehicles and robots traversing through an environment, images are acquired sequentially, thus visual place recognition localization approaches use the structure of sequentiality to locally match image sequences to the database for higher accuracy. In this paper we propose a not only sequential approach to localization; specifically, instead of linearly searching for sequences, we construct a directed acyclic graph and search for any kind of sequences. We evaluated the proposed approach on a dataset consisting of varying environmental conditions and demonstrated that it outperforms the SeqSLAM approach.

Keywords: Visual place recognition, localization, SeqSLAM, deep convolutional neural networks

1 Introduction

Localization, i.e., reasoning about own's location given a set of measurements from one or mulitple sensors, is a prerequisite capability for any autonomous system. It can be approached in different ways depending on the used sensor setup, and by relying on onboard sensors many approaches have been developed for mobile agents based on laser range sensors, cameras (mono, stereo or depth), ultrasonic sensor and even radars. Localization is often performed in a known environment; however, it can be challenging when the environment is highly dynamic both in the sense of having dynamic objects and changing its own appearance through time. Simply during a single day, visual appearance of the same location can change drastically due to day and night conditions. Nevertheless, although challenging, since images contain a rich set of information,

they can be leveraged to recognize places even during strong appearance changes such as day/night or seasonal changes.

Visual place recognition, as the name suggests, tackles the problem of recognizing a previously seen location given a single or multiple images captured by a camera, thus it can be seen as a specific approach to solving the localization problem. More formally, given a *query* image $I_{q_i} \in \mathcal{Q}$, taken as the vehicle traverses its route *on the fly*, we are trying to find the most plausible match from the labeled database of *reference* images $\mathcal{D}$ [11]. Such a match, often denoted with $I^*_{d_j} \in \mathcal{D}$, represents the current position hypothesis. Furthermore, it is an instance of a well-known computer vision problem – *visual instance retrieval* [18] where given a query image, we are trying to find all the possible matches that correspond to the category of this instance. However, subtle differences exist, the most prominent one being that both $\mathcal{Q}$ and $\mathcal{D}$ in visual place recognition are sequentially ordered. This insight into data sequentiality can help us build more robust systems [12], [16] and, as it will be seen, the proposed method strongly relies on it. Given that, in order to make visual place recognition a robust localization method, it should be developed to be *view-point invariant*, i.e., to be able to recognize the same location from different viewpoints, and *condition invariant*, i.e., to be able to recognize the same location irrespective of the time-of-the-day or season. For example, $\mathcal{D}$ could have been captured on a stormy winter evening, while $\mathcal{Q}$ is captured on a sunny autumn noon. Consequently, two main design aspects regarding visual place recognition are *image representation* and *image matching* - as the appropriate image representation is obtained, the goal is to find the appropriate match.

To represent an image of a place, we can employ classical computer vision approaches consiting of image feature extractors and descriptors - e.g., SURF[2] was used by [5], while ORB [3], [19] was used by [13]. *Histogram of oriented gradients* is often used as a global image descriptor [16], [14], [15], [25]. However, given the development of deep convolutional neural networks, research has been directed into utilizing feature maps obtained by passing an image through the network and using them as a global description of an image. Sünderhauf et al. [22] have concluded that those feature maps extracted from middle layers of the AlexNet architecture [10] behave better for *condition variations* while feature maps from higher layers are more suitable when view-point variance occurs. The same authors propose a system [23] that uses *an object proposal* method in order to achieve even stronger *view-point invariance*. Another notable application of DCNNs is NetVLAD [1] by Arandjelović et al. who have modified the original VLAD [9] by replacing the indicator function with softmax. Inspired by NetVLAD, Garg et al. [7] propose another descriptor called LoST which aggregates residuals of semantic categories. Fetching the appropriate representation of an image from DCNN, Hausler et al. [8] filter out "bad" slices from feature maps extracted from some k-th convolution layer. Chen et al. [4] obtain the representation by *multi-scale pooling* where feature map is divided into $S \times S$ subregions and the maximum activation is pooled resulting in a more compact representation for each feature map.

Regarding image matching, often used approach is SeqSLAM [12], that searches for the locally optimal sequence match - a sequence that bears the information about the vehicle traversing in a local scope. Siam and Zhang [21] upgraded SeqSLAM such that N approximate nearest neighbors (ANN) of the query image I_T were taken. Yin et al. [27] incorporate particle filter within SeqSLAM in order to reduce computational complexity. SMART [17] improves SeqSLAM by incorporating the odometry information from *wheel encoders*. Improved variants of SeqSLAM search methods (*cone-based* and *hybrid* method) can be found in [24]. In [6] for each query image N most similar matches from reference database are fetched. Thereafter, by using another system that approximates the depth from an image, authors find the reference image with the most plausible neighborhood. Nasser et al. [16] addressed the traversal matching by using *data association graph*. Each node within this graph represents route match between $\mathcal{Q}$ traversal and $\mathcal{R}$ traversal, while both traversals consist of sequence of images. When data association graph is constructed, an appropriate traversal A is obtained by solving *min-cost flow* problem. The work of Vysotska and Stachniss [26] can be considered as follow-up to [16] where the improvement is manifested in the fashion of search. System proposed in [16] operates in *offline fashion* meaning that both $\mathcal{D}$ and $\mathcal{Q}$ are first obtained and thereafter appropriate associations are found while this system operates in *online fashion* which means that right after the last query image is obtained, the appropriate match is found. As emphasized in [16] solving for min-cost flow problem is equivalent to the *shortest path* problem in *directed acyclic graph*. Nodes of the shortest path in their formulation represent match hypotheses.

In this paper we propose a not only sequential approach to localization; specifically, instead of linearly searching for sequences, as SeqSLAM does, for image matching we construct a directed acyclic graph and search for any kind of sequences. Thus instead of using the shortest path as route hypothesis, we use shortest paths to measure the association of the matches between $\mathcal{Q}$ and $\mathcal{D}$. For image representation, we use deep learning feature maps extracted from the OverFeat architecture [20], since it was constructed for localization and detection tasks. We evaluated the proposed approach on the *Bonn dataset* [26] consisting of varying environmental conditions and compared it to SeqSLAM. To evaluate quantitatively, we constructed precision-recall curves and computed the area under the curve. The results show that on the tested database the proposed approach outperforms SeqSLAM. Source code of our approach is available online[3].

2 Proposed Visual Place Recognition

The general scheme for visual place recognition, given in Fig. 1, is as follows:

1. Image representation for each image is obtained by pre-processing, feature extraction, description and dimensionality reduction

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

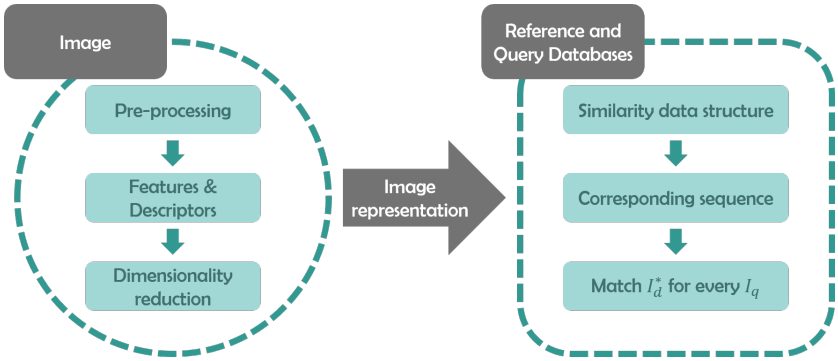


Fig. 1. Visual place recognition scheme.

2. Similarity s_{I_q,I_d} is calculated and by sequence matching, the best match $I_d^* \in \mathcal{D}$ for query image $I_q \in \mathcal{Q}$ is taken.

As stated above, more robust matching between $I_{q_i} \in \mathcal{Q}$ and $I_{d_j} \in \mathcal{D}$ can be found by incorporating data sequentiality, thus by observing an ordered local neighborhood around this match in contrast to the naive approach

$$I_{d_j}^* = \underset{I_{d_j} \in \mathcal{D}}{\operatorname{argmax}} \, s_{I_{q_i},I_{d_j}} \tag{1}$$

where

$$s_{I_{q_i},I_{d_j}} = \cos(\theta) = \frac{I_{q_i} I_{d_j}}{\|I_{q_i}\| \|I_{d_j}\|}. \tag{2}$$

Because images are represented as vectors we can measure the similarity by taking their cosine similarity according to (2). However, as noted by [15], "matching images just according to the best similarity score produces considerable false positives [...]".

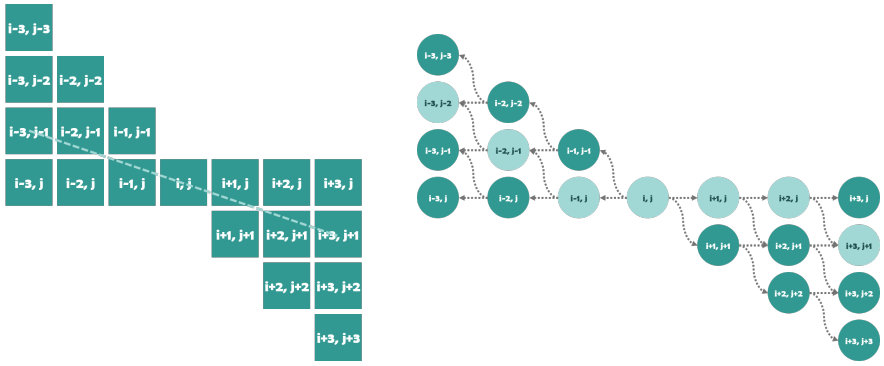
2.1 SeqSLAM

Let T denote the current time and d_s denote the number of matches some sequence is consisted of and let M denote the difference matrix where the j-th column $M[:,j] = \hat{D}^j$, $j \in \{T - \lfloor \frac{d_s}{2} \rfloor, \ldots, T + \lfloor \frac{d_s}{2} \rfloor\}$ is the vector of differences between j-th query image I_{q_j} and the reference database. Then, given a certain velocity V we can calculate the sequence weight

$$S_{j,T,V} = \sum_{t=T-\lfloor \frac{d_s}{2} \rfloor}^{T+\lfloor \frac{d_s}{2} \rfloor} \hat{D}_k^t \tag{3}$$

where $k = j + V(t - T)$. The most appropriate sequence for I_T and I_{d_j} is the one that minimize the weight of sequence by velocity, therefore

$$S_{j,T} = \min_V S_{j,T,V}. \tag{4}$$



(a) SeqSLAM **(b)** NOSeqSLAM

Fig. 2. (a) SeqSLAM searches for the optimal linear sequence passing through (I_{q_i}, I_{d_j}) while (b) NOSeqSLAM searches for the optimal single-source shortest path from root (I_{q_i}, I_{d_j}) to the left subgraph and from root to the right subgraph.

The procedure operates in the same manner if we are taking similarity between images, but then we maximize by V because *minimizing the difference* is equivalent to *maximizing the similarity* [11]. Another relevant fact is that SeqSLAM is agnostic regarding image representation. In the original work no features from images were extracted, thus original *human-understandable* downsampled image is used while the difference between two images is measured via *sum of absolute differences*.

2.2 NOSeqSLAM

Our method differs from SeqSLAM in such a way that the appropriate association between I_{q_i} [4] and I_{d_j} is not found by measuring the weight of optimal linear sequence, but generally by measuring the weight of any kind of sequence passing through the match of I_{q_i} and I_{d_j}. Henceforth, we denote this match with (I_{q_i}, I_{d_j}). For illustration of similarities and differences between the two methods confer Fig. 2.

By *linear sequence* we mean pure linear correlation between indices in difference matrix M just as it is illustrated in Fig. 2(a). From a physical point of view this means a vehicle should traverse same subroute in both $\mathcal{Q}$ and $\mathcal{D}$ with linear correlation in acceleration/deceleration. One special case of this condition is to traverse same subroute in $\mathcal{Q}$ and $\mathcal{D}$ without acceleration at all. This is a limiting factor, as a vehicle is akin to accelerate/decelerate all the time. We therefore model our system so that it searches for any kind of sequences - not only linear, and for that reason we name it NOSeqSLAM, where NO is acronym for *"not only"*.

[4] We substitute T with q_i for the sake of clarity.

Similar to difference matrix M, we place similarities between matches in matrix A, where $A[j,i] = s_{I_{q_i},I_{d_j}}$. For each match (I_{q_i}, I_{d_j}) we construct *directed acyclic graph* $G_{(i,j)}$ rooted at (I_{q_i}, I_{d_j}). This root is thereafter expanded in the left and the right direction with respect to i-th row resulting in left and right subgraphs. We build graph iteratively until the depth of $\lfloor \frac{d_s}{2} \rfloor$ is reached. As the graph expands, each node in the graph will be predecessor for η_{exp} nodes. This procedure can be parallelized on two threads, one for the left and one for the right subgraph of a DAG. Our system thereby depends upon two parameters: d_s (sequence length) and η_{exp} (expansion rate). The example for $d_s = 7$ and $\eta_{exp} = 2$ is shown in Fig. 2(b).

The weight of an edge from (I_{q_i}, I_{d_j}) to (I_{q_k}, I_{d_l}) is defined as

$$w\left[(I_{q_i}, I_{d_j}), (I_{q_k}, I_{d_l})\right] = 1 - A[l,k]. \tag{5}$$

Naseer et al. [16] use $1/s_{I_{q_k},I_{d_l}}$ in this situation; however, (5) is also reasonable – as association approaches one, weight approaches zero, and vice-versa. Therefore, it is appropriate to reach some node reciprocal to its similarity measure.

As the graph for (I_{q_i}, I_{d_j}) with associated weight is constructed, we still have to measure association for (I_{q_i}, I_{d_j}), thus, how good I_{d_j} fits for I_{q_i}. In NOSeqSLAM this measure is defined as the sum of similarities of those nodes that lie on the minimal of the shortest paths in $U_{(i,j)}$ that connect the leaf from left subgraph to the leaf in the right subgraph passing through (I_{q_i}, I_{d_j}), where $U_{(i,j)}$ is undirected version of $G_{(i,j)}$. Although this was our initial thought about the formulation, a simpler way to achieve this is to construct $G_{(i,j)}$, find the minimal of the shortest paths in left subgraph $l^*_{(i,j)}$ and the minimal of the shortest paths in right subgraph $r^*_{(i,j)}$, to sum similarity measures through this minimal shortest paths together with similarity between I_{q_i} and I_{d_j} what yields the following association measure:

$$S_{j,i} = A[j,i] + \sum_{(k,l)\in l^*_{(i,j)}} A[l,k] + \sum_{(k,l)\in r^*_{(i,j)}} A[l,k]. \tag{6}$$

Pseudocodes for both the proposed method and SeqSLAM are shown in Alg. 1 and Alg. 2.

3 Experimental Results

The first step to experimental evauluation was to build up the association matrix A which truly reflects the relationship between $\mathcal{Q}$ and $\mathcal{D}$. As mentioned, the key idea is to build this matrix up *row-by-row* whenever a new $I_{q_i} \in \mathcal{Q}$ is captured. This I_{q_i} is then compared with every $I_{d_j} \in \mathcal{D}$. The result is depicted in Fig. 3.

3.1 Dataset

For evaluation purposes we used the *Bonn dataset* contained with the publicly available implementation of [26]. The route is driven in an urban area and is

Algorithm 1 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

Algorithm 2 SeqSLAM

$V_{steps} = linspace(V_{min}, V_{max}, V_{step})$
for each $(I_T, I_{d_j}) \in \mathcal{Q} \times \mathcal{D}$ **do**
$S_{j,T} = \infty$
for each $V \in V_{steps}$ **do**
$s = S_{j,T,V}$
if $s < S_{j,T}$ **then**
$S_{j,T} = s$
end if
end for
end for

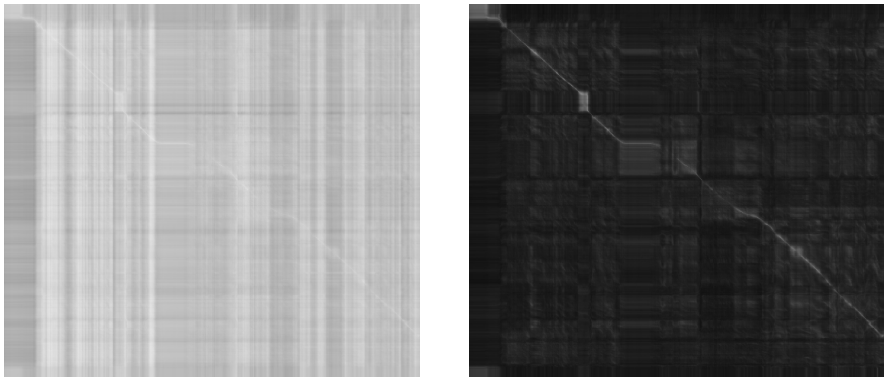
captured in two different environmental conditions. When driven for the first time (and therefore saved as $\mathcal{D}$), route is captured in the evening, while for the second time (saved as $\mathcal{Q}$) it is captured on a gloomy day. Although there is no record about the distance traveled throughout this route, by observing sequences alongside the route and from the cardinality of both datasets ($|\mathcal{D}| = 488$, $|\mathcal{Q}| = 544$) we can assert that the route is 1–2 km long.

View-point variance is not that accentuated in this dataset because, the vehicle stays in the same track for both traversals. However, variations in condition are severe as can be seen in Fig. 4. Not only illumination is different, but also various moving objects appear throughout both traversals.

3.2 Image representation

Besides raw images, the dataset comes with the feature maps extracted from the OverFeat architecture [20] for both $\mathcal{D}$ and $\mathcal{Q}$. This architecture is nearly the same as the AlexNet [10], but besides classification, it was constructed also for other tasks such as *localization* and detection.

In order to choose the best image representation algorithm, we conducted an experimental comparison. First, we constructed the HOG representation, but for which it can be seen from Fig. 3(a), that it does not quite yield discriminative associations for $\mathcal{Q}$ with respect to $\mathcal{D}$. Generally, the trend of replacing *handcrafted* features and descriptors with representations extracted from DCNN architectures is also present in image representation. Second, beside the OverFeat map that was readily available, we have also extracted feature maps from AlexNet



(a) HOG **(b)** OverFeat

Fig. 3. Plotting the association matrix A when using different image representations reflects their quality. The more accentuated is the contrast, the better.

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

Fig. 4. Different environmental conditions and occlusions at each traversal. Images taken from [26].

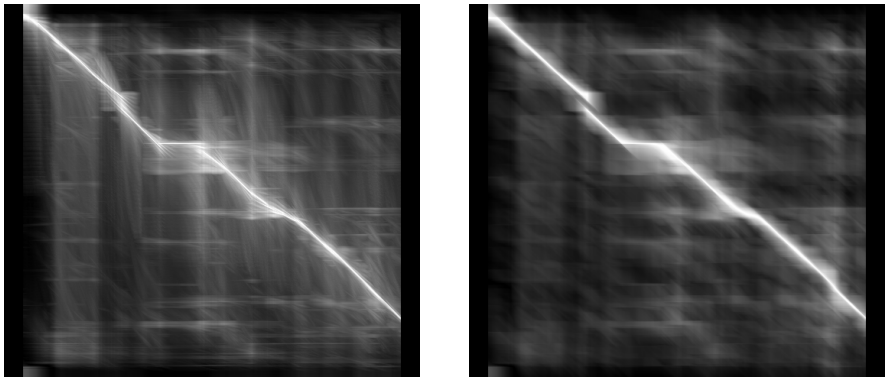
`conv3` layer as it was reported that this architecture is suitable for image representation too [22]. Moreover, therein it is also asserted that when the network is trained on *a scene-centric* training set (in contrast to *an object-centric* training set), even more accurate results can be obtained. For that purpose, we employ another AlexNet network trained on a scene-centric dataset Places365 [28].

Given that, we tested the suitability of the aforementioned representations. Even though the contrast of the association matrix plot, as in Fig. 3(a), can act as a qualitative indicator whether a representation is fine, a quantitative measure is needed. For that purpose we calculated precision and recall for different representations using $d_s = 7$ in combination with $\eta_{exp} = 3$ and accumulated the results in order to obtain the *area under the curve* (AUC) measure. No significant improvement when using AlexNet trained on Places365 ($AUC = 0.89193$) with respect to the one trained on ImageNet ($AUC = 0.88786$) was noticed. The HOG representation, as expected, had the lowest score ($AUC = 0.77529$). Finally, similar results were obtained with OverFeat ($AUC = 0.91425$) and AlexNet representa-

tions, which was expected since OverFeat is almost identical in its convolutional layers. Finally, given the results, we decided to use OverFeat feature maps in the ensuing experiments both for NOSeqSLAM and SeqSLAM.

3.3 Comparison of NOSeqSLAM and SeqSLAM

In the next experiment we compared the proposed NOSeqSLAM and SeqSLAM for visual place recognition. By focusing on Fig. 5, we can see that the association matrix reflects the designs of each approach, i.e., SeqSLAM leaves traces in the form of linear sequences distributed through the path, while at the same time, NOSeqSLAM traces are softer.



(a) SeqSLAM **(b)** NOSeqSLAM

Fig. 5. Association matrices of $\mathcal{Q}$ and $\mathcal{D}$ for SeqSLAM and NOSqeSLAM. Differences are visible as the two algorithms assign different associations.

Fig. 5 shows that that the sequence (no matter if linear or non-linear) of length d_s does not fit into first $\lfloor \frac{d_s}{2} \rfloor$ indices and last $\lfloor \frac{d_s}{2} \rfloor$ indices of $\mathcal{Q}$ (notice lateral black areas). This implicates that no viable match can be found for first $\lfloor \frac{d_s}{2} \rfloor$ and last $\lfloor \frac{d_s}{2} \rfloor$ query images and matches for those $2 \lfloor \frac{d_s}{2} \rfloor$ images will be declared as false negatives. From *precision-recall* curves in Fig. 6(a) we can see that in general NOSeqSLAM performs better regardless of chosen d_s and that the maximal recall is roughly the same depending on d_s no matter what method is used. For fair comparison we group the results according to d_s and show the results in Table 1, from which we can see that in terms of the AUC measure, NOSeqSLAM outperforms SeqSLAM. In visual place recognition we are striving to achieve the AUC measure as large as possible - ideally equal to 1. This, amongst others, means that no false positives have been encountered at all, i.e., each match is consistent with the ground truth.

Given the association matrix A, NOSeqSLAM takes $\Theta\left(|\mathcal{Q}||\mathcal{D}|d_s^2\eta_{exp}^2\right)$ asymptotic running time while SeqSLAM operates in $\Theta\left(|\mathcal{Q}||\mathcal{D}|d_s V_{steps}\right)$. If we diminish

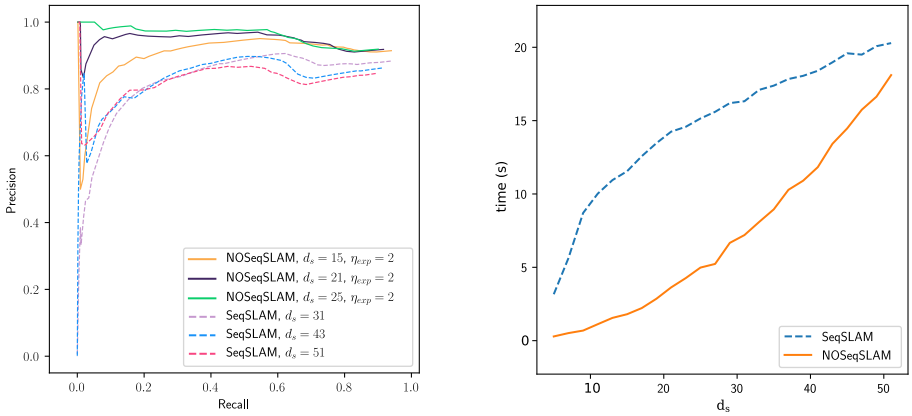


Fig. 6. (a) Precision-recall plots with $d_s \in \{31, 43, 51\}$. (b) Running time as a function of the sequence length $d_s \in \{5, 7, \ldots, 51\}$.

the role of factors $d_s^2\eta_{exp}^2$ and $d_s V_{steps}$, thus $\Theta\left(d_s^2\eta_{exp}^2\right) = \Theta\left(d_s V_{steps}\right) = \Theta\left(1\right)$, we can say that both algorithms operate in $\Theta\left(|\mathcal{Q}||\mathcal{D}|\right)$ asymptotic time. In terms of *real-time* performance measured on a `i7@2.8GHz` laptop processor, NOSeqSLAM evaluation takes 0.29s and SeqSLAM evaluation takes 3.16s for $d_s = 5$. For $d_s = 51$, NOSeqSLAM evaluation takes 18.1s, while SeqSLAM evaluation takes 20.29s. This is approximately 33ms per query image which is readily employable for autonomous vehicles. In general, for $d_s \in \{5, 7, \ldots, 51\}$ NOSeqSLAM operates faster than SeqSLAM as can be seen in Fig. 6(b). Although NOSeqSLAM running time will exceed SeqSLAM running time once when d_s is sufficiently large, we think that $d_s = 51$ is more than enough to describe a local neighborhood so there is no need for larger d_s.

Table 1. SeqSLAM (denoted with S) and NOSeqSLAM (denoted with N) AUC results.

d_s	alg.	η_{exp}	AUC
51	S	-	0.73113
	N	3	0.84503
	N	2	0.86518

d_s	alg.	η_{exp}	AUC
43	S	-	0.75650
	N	3	0.85559
	N	2	0.86630

d_s	alg.	η_{exp}	AUC
31	S	-	0.77137
	N	3	0.83986
	N	2	0.84798

d_s	alg.	η_{exp}	AUC
19	S	-	0.84974
	N	3	0.87886
	N	2	0.88981

d_s	alg.	η_{exp}	AUC
15	S	-	0.87877
	N	3	0.89566
	N	2	0.90672

d_s	alg.	η_{exp}	AUC
11	S	-	0.90616
	N	3	0.90744
	N	2	0.91058

d_s	alg.	η_{exp}	AUC
7	S	-	0.90482
	N	2	0.91319
	N	3	0.91425

d_s	alg.	η_{exp}	AUC
5	S	-	0.91714
	N	2	0.92258
	N	3	0.92288

4 Conclusion

In this paper we have presented a not only sequential approach to visual place recognition. This design objective has been achieved with the powerful tool of graph theory - *shortest path* on a directed acyclic graph in such a way that the accumulated similarity throughout the shortest path represents the plausibility of the match. By using state-of-the-art image representations extracted from DCNNs, we made our system view-point and condition invariant.

From experiments conducted on Bonn dataset, we have shown that our system can operate in a rather demanding urban area with strong appearance changes. Not only that the proposed approach is capable of achieving this objective, but it has outperformed SeqSLAM in terms of both precision-recall and execution time on the tested dataset. Given the results we have observed, this system may be used for other purposes such as simultaneous localization and mapping *loop closing detection* and *relocalization*.

References

1. Arandjelović, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: NetVLAD: CNN architecture for weakly supervised place recognition **70**(5), 641–648 (2015)
2. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-Up Robust Features (SURF). Comput. Vis. Image Underst. **110**(3), 346–359 (2008)
3. Calonder, M., Lepetit, V., Strecha, C., Fua, P.: BRIEF: Binary robust independent elementary features. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (2010)
4. Chen, Z., Jacobson, A., Sünderhauf, N., Upcroft, B., Liu, L., Shen, C., Reid, I., Milford, M.: Deep learning features at scale for visual place recognition. Proceedings - IEEE International Conference on Robotics and Automation **1**, 3223–3230 (2017)
5. Cummins, M., Newman, P.: FAB-MAP: Probabilistic localization and mapping in the space of appearance. International Journal of Robotics Research **27**(6), 647–665 (2008)
6. Garg, S., Babu, M., Dharmasiri, T., Hausler, S., Sünderhauf, N., Kumar, S., Drummond, T., Milford, M.: Look No Deeper: Recognizing Places from Opposing Viewpoints under Varying Scene Appearance using Single-View Depth Estimation (2019)
7. Garg, S., Sünderhauf, N., Milford, M.: LoST? Appearance-Invariant Place Recognition for Opposite Viewpoints using Visual Semantics (2018)
8. Hausler, S., Jacobson, A., Milford, M.: Feature Map Filtering: Improving Visual Place Recognition with Convolutional Calibration (2018)
9. Jegou, H., Douze, M., Schmid, C., Perez, P.: Aggregating local descriptors into a compact image representation. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 3304–3311. IEEE (2010)
10. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS (2012)
11. Lowry, S., Milford, M.J.: Supervised and Unsupervised Linear Learning Techniques for Visual Place Recognition in Changing Environments. IEEE Transactions on Robotics **32**(3), 600–613 (2016)

12. Milford, M.J., Wyeth, G.F.: SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights. In: Proceedings - IEEE International Conference on Robotics and Automation, pp. 1643–1649. IEEE (2012)
13. Mur-Artal, R., Montiel, J.M.M., Tardós, J.D.: {ORB-SLAM:} a Versatile and Accurate Monocular {SLAM} System. CoRR **abs/1502.0** (2015)
14. Naseer, T., Burgard, W., Stachniss, C.: Robust Visual Localization Across Seasons. IEEE Transactions on Robotics **34**(2), 289–302 (2018)
15. Naseer, T., Ruhnke, M., Stachniss, C., Spinello, L., Burgard, W.: Robust visual SLAM across seasons. IEEE International Conference on Intelligent Robots and Systems **2015-Decem**, 2529–2535 (2015)
16. Naseer, T., Spinello, L., Burgard, W., Stachniss, C.: Robust visual robot localization across seasons using network flows. Proceedings of the AAAI Conference on Artificial Intelligence pp. 2564–2570 (2014)
17. Pepperell, E., Corke, P.I., Milford, M.J.: All-environment visual place recognition with smart. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 1612–1618 (2014)
18. Razavian, A.S., Sullivan, J., Carlsson, S., Maki, A.: Visual Instance Retrieval with Deep Convolutional Networks (June 2017) (2014)
19. Rosten, E., Porter, R., Drummond, T.: Faster and better: A machine learning approach to corner detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **32**(1), 105–119 (2010)
20. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., Lecun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. In: International Conference on Learning Representations (ICLR2014), CBLS, April 2014 (2014)
21. Siam, S.M., Zhang, H.: Fast-seqslam: A fast appearance based place recognition algorithm. In: 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 5702–5708 (2017)
22. Sünderhauf, N., Shirazi, S., Dayoub, F., Upcroft, B., Milford, M.: On the performance of ConvNet features for place recognition. IEEE International Conference on Intelligent Robots and Systems **2015-Decem**, 4297–4304 (2015)
23. Sünderhauf, N., Shirazi, S., Jacobson, A., Dayoub, F., Pepperell, E., Upcroft, B., Milford, M.: Place Recognition with ConvNet Landmarks: Viewpoint-Robust, Condition-Robust, Training-Free. In: Robotics: Science and Systems XI. Robotics: Science and Systems Foundation (2015)
24. Talbot, B., Garg, S., Milford, M.: OpenSeqSLAM2.0: An Open Source Toolbox for Visual Place Recognition Under Changing Conditions. IEEE Robotics and Automation Letters **1**(1), 213–220 (2018)
25. Vysotska, O., Naseer, T., Spinello, L., Burgard, W., Stachniss, C.: Efficient and effective matching of image sequences under substantial appearance changes exploiting gps priors. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 2774–2779 (2015)
26. Vysotska, O., Stachniss, C.: Lazy Data Association For Image Sequences Matching Under Substantial Appearance Changes. IEEE Robotics and Automation Letters **1**(1), 213–220 (2016)
27. Yin, P., Srivatsan, R.A., Chen, Y., Li, X., Zhang, H., Xu, L., Li, L., Jia, Z., Ji, J., He, Y.: MRS-VPR: a multi-resolution sampling based global visual place recognition method (2019)
28. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: A 10 million image database for scene recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (2017)