

TextPlace: Visual Place Recognition and Topological Localization Through Reading Scene Texts

Ziyang Hong¹, Yvan Petillot¹, David Lane¹, Yishu Miao², Sen Wang^{1*}

¹Edinburgh Centre for Robotics, Heriot-Watt University ²University of Oxford
{zh9, y.r.petillot, d.m.lane, s.wang}@hw.ac.uk, yshu.miao@gmail.com

Abstract

Visual place recognition is fundamental for many vision based applications. Sparse feature and deep learning based methods have been successful and dominant over the decade. However, most of them do not explicitly leverage high-level semantic information to deal with challenging scenarios where they may fail. This paper proposes a novel visual place recognition algorithm, termed TextPlace, based on scene texts in the wild. Since scene texts are high-level information invariant to illumination changes and very distinct for different places when considering spatial correlation, it is beneficial for visual place recognition tasks under extreme appearance changes and perceptual aliasing. It also takes spatial-temporal dependence between scene texts into account for topological localization. Extensive experiments show that TextPlace achieves state-of-the-art performance, verifying the effectiveness of using high-level scene texts for robust visual place recognition in urban areas.

1. Introduction

Visual place recognition (VPR) is one of the fundamental elements of many computer vision and robotics applications and has attracted considerable attention over the decades [15]. Its goal is to recognize revisited places accurately by using visual information.

Most of the established visual place recognition systems rely on local visual features and the so-called Bag of Visual Words (BoVWs) technique [20, 27]. Some of them have shown impressive place recognition performance for large-scale environments by using a single image matching [4, 5]. However, for long-term operation, their performance may degrade due to extreme appearance variations caused by day-night cycles, weather and seasonal conditions. For example, as shown in Fig. 1, the visual appearance of a



Figure 1: Place recognition with scene texts. Top: Same place with different visual appearance at day and night. Bottom: Different places with strong perceptual aliasing. Note scene texts are consistent for the same place and distinct for different ones.

same place can be very different at day and night (top), while different places may look visually similar with strong perceptual aliasing (bottom). Therefore, sparse local feature based place recognition approaches may be fragile when encountering a large appearance variation due to illumination and season changes. To overcome these problems, sequence based methods [17, 19] use global image descriptors and a sequence of images for matching. Current state-of-the-art VPR systems mainly rely on deep learning techniques [2, 7] to learn a latent representation for place recognition in challenging scenarios. However, most of them do not explicitly leverage high-level semantic information.

Human is remarkably good at the visual place recognition task. One of the reasons is that human can make full use of high-level semantic information for scene understanding and place recognition. Texts, as one of main types of high-level semantics and the fruit of human’s wisdom, have been long used for reading maps, giving directions and finding destinations. However, they have not been fully exploited for visual place recognition research although scene texts pervasively exist in man-made environments and urban areas, e.g., road signs, street names and shop signage. Revisit the example in Fig. 1. It can be seen that the scene texts can

*Corresponding author

be powerful to deal with the challenging perceptual problems since they are consistent for a same place (top) and distinct for different ones (bottom).

In this paper, we explore how to leverage high-level textual information for visual place recognition and topological localization. A novel visual place recognition system, termed *TextPlace*, is proposed. Given scene texts detected in the wild, TextPlace represents places by textual descriptors and builds topological maps to encapsulate spatial coherence of the texts. Our main contribution is threefold. First, to the best of our knowledge, TextPlace is the first visual place recognition system which mainly uses scene texts as descriptors to deal with place recognition in challenging scenarios, e.g., serious illumination changes, perceptual aliasing, dynamic occlusion and variant viewpoints. Second, we develop a whole pipeline on using high-level textual information for topological metric localization. Last, we demonstrate the first time that texts can be effectively utilized to tackle visual place recognition and topological localization in the aforementioned challenging scenarios.

The rest of this paper is organized as follows. Section 2 reviews related work, followed by algorithm description on TextPlace in Section 3. Experimental results are given in Section 4. Conclusion is drawn in Section 5.

2. Related Work

Visual place recognition research can be roughly categorized into sparse feature, sequence and deep learning based techniques.

2.1. Sparse Feature based Place Recognition

Sparse feature based place recognition algorithms shown great success and was dominant a decade ago. They rely on local features, e.g., SURF [3] and SIFT [14], and BoVW technique. A classic method is FAB-MAP, a probabilistic localization and mapping model based on visual appearance [4]. Similarly, [6] uses BoVW model combining ORB features for fast place recognition. Vector of Locally Aggregated Descriptors (VLAD) [11] and DenseVLAD [28] compute the sum of the residuals between each visual word and the corresponding clustering center. However, the sparse feature based methods may be problematic when facing dramatic illumination changes and dynamic scenes.

2.2. Sequence based Place Recognition

Some work has exploited sequential information for place recognition. Milford and Wyeth [17] propose SeqSLAM which uses a sequence of global image descriptor for place recognition. It was one of the first attempts to address the problems raised by extreme perceptual changes, through exploiting the spatial and temporal information. In [18, 19, 29], a graph based model is used to represent the data association between the query and map sequences in

which the cost of network flow within the graph is minimized to find a match.

2.3. Deep Learning based Place Recognition

Since the high-level features learned from deep neural networks may implicitly encapsulate some semantic information, they can be used for place recognition in changing environments. In NetVLAD [2], a set of local descriptors of a single image is learned by a convolutional neural network, and the compact form of the local descriptors is computed similar to VLAD [11]. In [9], a three-stream siamese network is trained for image retrieval. Whereas in [22], 3D models are constructed for each cluster of images and used to guide for image retrieval. Both in [21] and [1], they perform image-to-image translation from the source domain to the target domain by using Generative Adversarial Network (GANs) [8] so that they can conduct simpler matching in the target domain.

2.4. Text based Localization and Place Recognition

Since scene texts, such as street signs, road markings, billboards and shop signage, usually carry extensive discriminative information, they can be considered as landmarks for localization. Textual information is proposed for robot navigation in [30], where a conjunction text feature is used to encode text information as landmark for loop closure detection. Radwan et al. [23] present a global localization approach by using multiple texts observed on a map. Ranganathan et al. [24] pair the road markings with precise GPS position during mapping and perform localization by matching the detected road markings with the map. The recent advance of text detection in the wild [10, 13] paves the way for using textual information for localization and place recognition in open, challenging environments. Next we explore how to use this high-level information in the wild for place recognition and localization.

3. Proposed Algorithm

In this section, the proposed TextPlace algorithm is described. Its goal is to track a camera robustly with respect to a text augmented topological map by text based place recognition and topological localization.

3.1. System Overview

The TextPlace system contains two major stages: 1) Mapping stage. Given a sequence of images and their visual odometry estimates, TextPlace builds a topological metric map of scene texts recognized. 2) Place recognition and localization stage. A sequence of query images or a new camera is localized with respect to the topological map built, relying on scene texts and their spatial-temporal coherence in an environment. Fig. 2 shows the system overview. The details of the system are explained next.

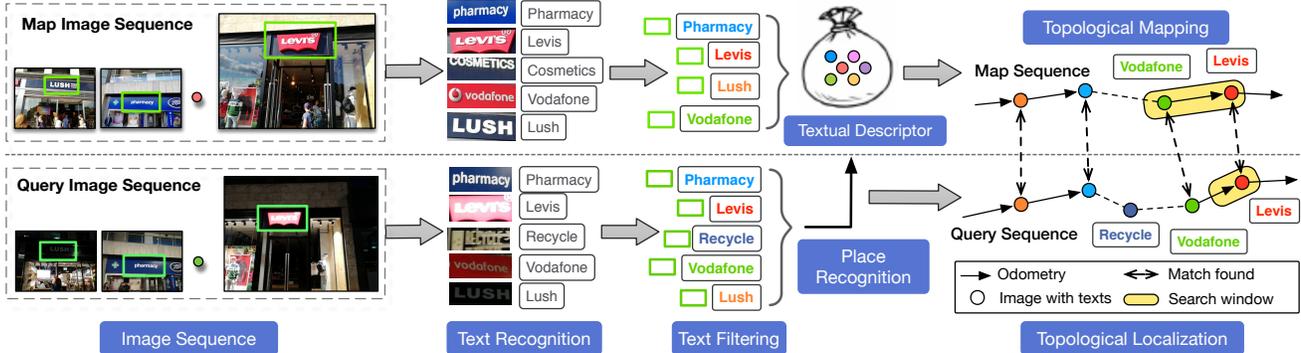


Figure 2: System overview (Top: mapping stage. Bottom: place recognition and localization stage). Scene texts are detected and recognized on both map and query image sequences. Some of the text strings are filtered by matching with the words in a pre-defined dictionary. At the mapping stage, a textual descriptor augmented topological map is produced. It is then used for text based place recognition and topological localization.

3.2. Text Detection and Recognition

Different from traditional VPR methods which use sparse or global image features, TextPlace extracts high-level textual information from the scenes and uses it as the primitive for place recognition. To spot the texts in the wild, TextBoxes++ [13], a deep learning based text detection and recognition model, is used. It can predict multiple texts appear in an image and their bounding boxes. The texts are then recognized as a sequence of text strings by the deep neural network model proposed in [26]. In this work, we use a pre-trained model of TextBoxes++ to extract texts and predict their bounding box positions.

3.3. Textual Descriptor

3.3.1 Text Filtering

Some texts in the wild can be distinct and informative when considering their geographical correlation, such as street names and shop signage. On the other hand, determiner words, like demonstratives (e.g., “this”) and possessives (e.g., “my”) provide very limited geo-information. Therefore, TextPlace uses a pre-defined dictionary to specify salient texts which are considered during the mapping and localization stages. In this work, a dictionary containing street names, shop signage and billboards is automatically built by crawling an online map. Since these texts are reasonably stable for a long period, it makes sense to maintain a pre-defined dictionary. Note the dictionary does not need to be comprehensive or latest thanks to the spatial-temporal dependence introduced by the topological mapping and localization. We will discuss this later.

In practice, some text recognition results can be incorrect due to poor image quality, unusual font styles, etc. Therefore, it is necessary to filter out these noisy texts, without considering them at the mapping stage. To this end, we measure the Levenshtein distance [12] between a rec-

ognized text and the words in the pre-defined dictionary. Levenshtein distance is the minimum operations, including deletions, insertions and substitutions, to take for correcting string A to string B. For example, the Levenshtein distance between “sitting” and “kitten” is 3 because three operations are needed to convert “kitten” to “sitting”: replacing “k” with “s”, replacing “e” with “i” and inserting a “g” after “n”. Denote $|A|$ and $|B|$ as the lengths of string A and B, respectively. The Levenshtein distance d between them is equal to $d_{A,B}(|A|, |B|)$ where

$$d_{A,B}(i, j) = \begin{cases} \max(i, j) & \text{if } \min(i, j) = 0, \\ \min \begin{cases} d_{A,B}(i-1, j) + 1 \\ d_{A,B}(i, j-1) + 1 \end{cases} & \text{otherwise.} \\ d_{A,B}(i-1, j-1) + 1_{A_i \neq B_j} \end{cases}$$

$1_{A_i \neq B_j}$ means the indicator function equal to 0 when A_i is the same as B_j and equal to 1 otherwise, and $d_{A,B}(i, j)$ is the distance between the first i character of A and the first j character of B. We therefore can filter out the incorrect text recognition results whose minimum Levenshtein distances across the whole dictionary are larger than a threshold.

3.3.2 Textual Descriptor

After the text filtering, each image frame which has some remaining texts is represented by a textual descriptor, facilitating the similarity check (details in Section 3.5.2) for place recognition. The textual descriptor Y of an image includes the set of the N remaining text strings $S = \{s_1, \dots, s_N\}$ and their bounding box positions $B = \{b_1, \dots, b_N\}$.

3.4. Topological Mapping

In order to reflect the spatial-temporal dependence of the detected texts in the environments, a topological metric map is adopted in TextPlace. Specifically, each node of the topo-

logical map represents an image, containing its textual descriptor and camera pose. The edge between two neighbor nodes denotes a relative transformation estimated by visual odometry, which can be accumulated to estimate the camera pose. Once the camera moves a certain distance, termed as displacement between nodes, a new node is added in the map. The mechanism of this topological metric map models the spatial coherence of the scene texts and forms distinct combinations of textual descriptors in a graph for place recognition.

3.5. Place Recognition

Once a topological metric map is built, we can perform place recognition and topological localization with respect to it. Similar to the mapping stage, each image frame is also processed with the text detection, recognition and filtering as described in Section 3.2 and 3.3.

The place recognition is formulated as how to best associate a query frame with a map node (representing an image) in the topological metric map, considering both the spatial-temporal constraints and the textual descriptor.

3.5.1 Spatial-Temporal Dependence

Scene texts are usually geographically distributed with a strong spatial correlation in the environments. For example, three adjacent store signs can be distinct in an area. Therefore, TextPlace utilizes a search window to confine the map nodes to match with for each query image. The window has an adaptive size, which depends on the uncertainty of the camera pose (details in Section 3.6.2). This significantly improves the efficiency and robustness of the place recognition, especially for large-scale environments, because the similarity matching can be conducted only with the map nodes in the window instead of the whole topological map.

3.5.2 Similarity Matching

We define a similarity function to match the textual descriptor of a query image with the map images within the search window. Fig. 3 gives an example on how to compute each similarity term. It mainly considers the following two metrics.

Levenshtein distance Denote the textual descriptors of a query and its i th map node in the search window as Y_q and Y_m^i , respectively. The normalized total Levenshtein distance $D(Y_q, Y_m^i)$ between their sets of text strings can be computed as

$$D(Y_q, Y_m^i) = \frac{1}{N} \sum_{k=1}^N \frac{\max(L_k - d_k^i, 0)}{L_k} \quad (1)$$

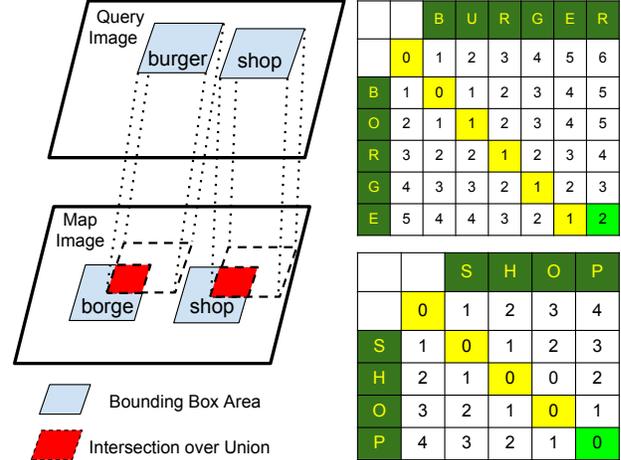


Figure 3: An example on similarity measurement. Left: IoU measurement. Right: Grids showing Levenshtein distance measurement between two pairs of query and map strings. The bottom right cell (green) is the final Levenshtein distance for each case.

where N is the number of text strings in the query descriptor Y_q , L_k is the length of the k th text string in Y_q , and d_k^i is the minimum Levenshtein distance between this k th string and all the strings in Y_m^i .

IoU Once the Levenshtein distance between a pair of query and map text strings is below an association threshold, the Intersection-over-Union (IoU) of their corresponding bounding boxes is computed. The mean IoU between Y_q and Y_m^i is defined as

$$U(Y_q, Y_m^i) = \frac{1}{N} \sum_{k=1}^N \frac{A_k \cap A_k^i}{A_k \cup A_k^i} \quad (2)$$

where A_k is the bounding box area of the k th text string in Y_q , A_k^i is the bounding box area of its matched string in Y_m^i , and \cap and \cup denote intersection and union of the two areas, respectively.

Finally, the similarity score for the i th map node in the search window is the weighted sum of these two metrics:

$$S_{sim}(Y_q, Y_m^i) = w_d \cdot D(Y_q, Y_m^i) + w_u \cdot U(Y_q, Y_m^i) \quad (3)$$

where w_d and w_u ($w_d + w_u = 1$) are weighting factors for the Levenshtein distance and IoU, respectively. Therefore, the map node whose similarity score is maximum is chosen as the recognized match for the query image.

3.6. Topological Localization

Based on the place recognition, we can achieve topological localization by modelling the temporal dependence of a camera and its motion estimation. It includes 3 main modules: 1) initialization, 2) pose tracking and update and 3) re-localization.

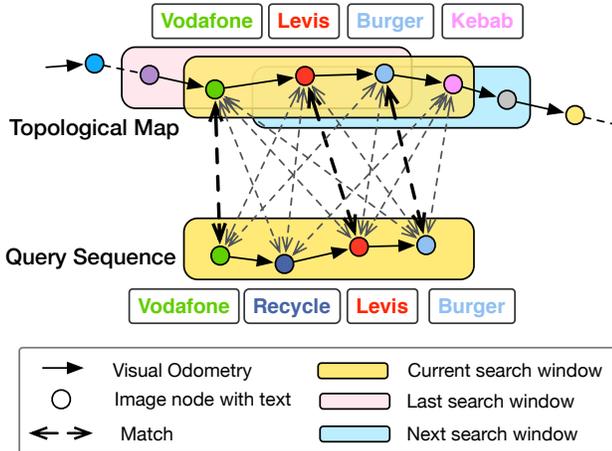


Figure 4: Example of a global searching. A sliding window of size 4 moves along the topological map, computing a set of similarity scores between query and map nodes to find the best match sequence which maximizes the sum of similarity score.

3.6.1 Initialization

Since a camera can start with an unknown initial position with respect to the topological map, we use the place recognition in the last section to initialize its location. Specifically, once a number of matches continuously having high similarity scores, the sequence of these query images is matched across the whole topological map as a sliding window. The camera is then initialized as the pose of the last node of the best matched sequence in the topological map. The search window is also set to the neighbor map nodes. An example of global searching is given in Fig. 4.

3.6.2 Pose Tracking and Update

Once the camera’s pose is initialized on the topological map, its location can be tracked through visual odometry while matching with map nodes. Since visual odometry accumulates drift over time, the uncertainty of the camera pose increases. The size of the search window on the topological map grows accordingly as it is designed to associate with the pose uncertainty. For matches whose similarly scores above a threshold, TextPlace selects the map node with the highest score. Then the pose of the chosen map node is used to update the camera’s pose in the framework of Extended Kalman Filter. After pose fusion, the drift is corrected and the size of the search window shrinks.

3.6.3 Re-Localization

The camera may lose tracking in reality due to a long period of severe occlusion and dynamic objects. In order to recover from losing tracking, a re-localization mode is introduced for the upcoming frame. Similar to the initialization step,

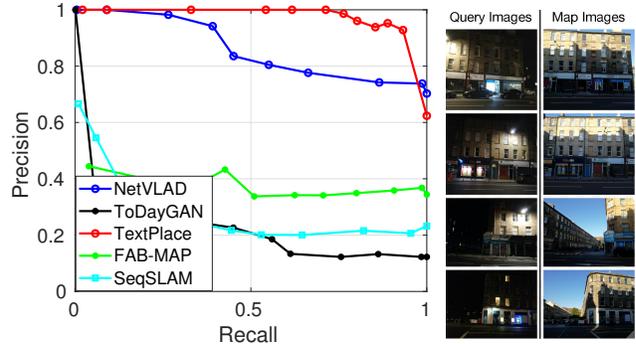


Figure 5: PR curve and sample images of Street 1 dataset. It is a day-night setting.

a sliding window is formed starting from the first succeeding frame which has texts detected. A global search is then performed to find the best local match within an enlarged search region defined by the predicted pose from the visual odometry. If the timeout period of re-localization expires, the localization system will be re-initialized.

4. Experimental Result

In this section, TextPlace is evaluated by comparing with the state-of-the-art place recognition algorithms in various scenarios.

4.1. Baselines and Evaluation

The visual place recognition algorithms chosen as baselines include ToDayGAN [1], NetVLAD [2], FAB-MAP [4] and SeqSLAM [17]. They cover the spare feature (FAB-MAP), global feature and sequence (SeqSLAM) and the state-of-the-art deep learning (ToDayGAN and NetVLAD) based methods. We use the open-source implementations of FAB-MAP¹, SeqSLAM², ToDayGAN³ and NetVLAD⁴. For NetVLAD, the top 1 match hypothesis is selected by comparing cosine distance of NetVLAD descriptors. This is different from the results in NetVLAD’s original paper where top 5 matches are reported. Note ToDayGAN is only compared with night-day setting since it focuses on synthesizing daytime images from night ones. Similar to most of the visual place recognition work, we use precision and recall (PR) as the main performance measurement for evaluation [15].

4.2. Public Available Dataset

We evaluate our algorithm on two public available datasets: ETH V4RL Urban Place Recognition Dataset [16] and SYNTHIA dataset [25]. V4RL is captured for place

¹<https://github.com/arrenglover/openfabmap>

²<https://openslam-org.github.io/openseqslam.html>

³<https://github.com/AAanoosheh/ToDayGAN>

⁴<https://github.com/Relja/netvlad>

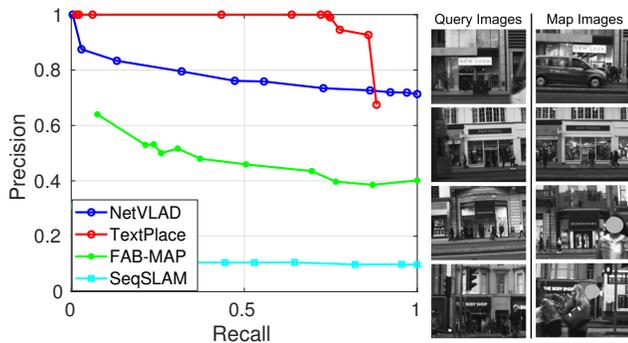


Figure 6: PR curve and sample images of Street 2 dataset.

recognition tasks from a flying drone or a pedestrian with a large variation on viewpoint, while [25] is synthetic dataset across different seasons and day-night conditions for a car driving scenario. We denote the summer-and-night dataset as SYNTHIA 1, and summer-and-winter dataset as SYNTHIA 2.

4.3. Self-Collected Dataset

To evaluate the performance of TextPlace against extreme perceptual changes, high dynamics and random occlusions, we collected 3 pairs of map and query sequences in outdoor streets and an indoor shopping mall, using a standard smart phone. Due to the severe illumination changes, occlusions and high dynamics (cars, buses and pedestrians), the visual appearance of the same place can vary dramatically, making the dataset very challenging. Some sample images are given in Fig. 1, Fig. 5 and Fig. 6. The datasets are named Street 1, Street 2 and Mall, respectively. All the self-collected datasets will be made publicly available⁵.

4.4. Performance on Precision and Recall

This section discusses the PR performance on different settings.

Day-night setting. The PR curve and some sample images of Street 1 dataset are shown in Fig. 5. It can be seen that TextPlace outperforms other methods when the recall is less than 0.9. It can achieve 100% precision when the recall is 0.7 on this day-night setting. NetVLAD also demonstrates strong performance in this scenario. But its precision drops below 100% while its recall is higher than 0.28. Since the environments have serious perceptual aliasing, ToDayGAN does not show high performance.

Day-day setting Three datasets are used for the day-day setting: Street 2, Mall and the sequence 1 in V4RL. The PR curve of Street 2 is given in Fig. 6. It can be seen that the precision of SeqSLAM is seriously degraded due to severe occlusions occurred occasionally. NetVLAD’s result is also slightly affected. Since TextPlace relies on high-level

⁵Datasets available at: <https://github.com/ziyanghong/dataset>

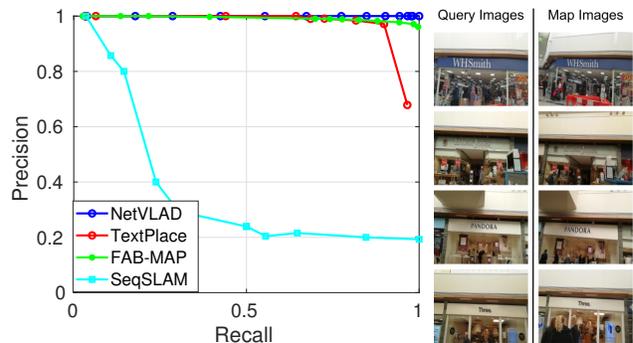


Figure 7: PR curve and sample images of Mall dataset.

texts, it can still achieve relatively high precision. Fig. 7 presents the PR results on the Mall dataset. Since the data was collected in 2 close periods, it is less challenging. FAB-MAP achieves competitive result with NetVLAD. Similar to previous experiments, TextPlace’s precision drops drastically once the recall is set higher than 0.9. This is because the threshold for the Levenshtein distance needs to be set big to have very high recall, which introduces many mismatches. The sequence 1 of the V4RL dataset has similar performance as shown in Fig. 9 (left).

Viewpoint setting. The sequence 2 of the V4RL dataset has big changes on viewpoint and image exposure between the query and map sequences, making it more challenging for visual place recognition. As shown in Fig. 9 (right), TextPlace achieves good robustness. This is because the high-level textual information used by TextPlace remains even with different viewpoints and image exposures. A good example is the fifth query (column) in Fig. 8. Since the query image is captured from a top-down view, the 2nd floor of the store in the map image is no longer preserved. Neither of NetVLD, SeqSLAM or FAB-MAP matches the correct map image for this query.

Synthetic setting. SYNTHIA 1 evaluates the match between the summer and night sequences, while SYNTHIA 2 covers winter and summer sequences. Results are shown in Fig. 10. It can be seen that TextPlace achieves comparable performance to SeqSLAM and NetVLAD.

Discussion. The previous experimental results verify that high-level textual information can be beneficial for visual place recognition in urban areas, even with challenging illumination changes and serious occlusions. Among all the compared algorithms, NetVLAD achieves excellent performance on various datasets. However, it may produce incorrect matching when the matched image highly resembles to the query in terms of the environment layout and structure. The second query (column) in Fig. 8 is a good example. TextPlace relies on the high-level texts and its spatial-temporal dependence to handle this problem.



Figure 8: Query and matching examples. Each column represents a query and matched images of various algorithms. Images with green frames are correct matches, while the ones with red frames are incorrect matches. ToDayGAN is not compared for the sequences in day-day setting.

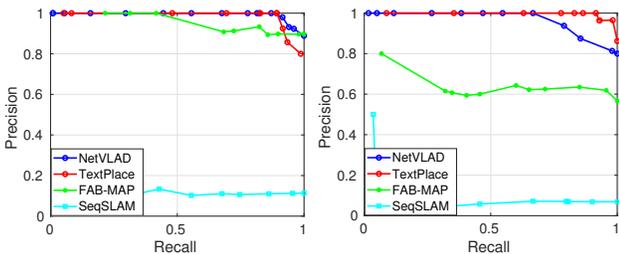


Figure 9: PR curve of V4RL dataset. Left: Sequence 1. Right: Sequence 2.

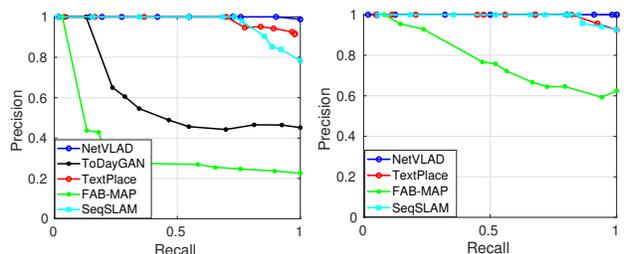


Figure 10: PR curve of SYNTIA dataset. Left: SYNTIA 1. Right: SYNTIA 2.

4.5. Comparison of Global and Prior Methods

The prior knowledge of TextPlace is a topological pose graph built online through VO and updated by texts, rather than strong global prior (e.g., GPS). Although leveraging spatial-temporal dependence (topological pose graph) is one of the novelties of TextPlace, separated evaluations on prior and global methods are provided for further comparison. We conducted 2 sets of experiments:

1. Adding position prior to NetVLAD, FAB-MAP and ToDayGAN. The search spaces of these methods are set to 10, 20 and 30 meters w.r.t the query image position. Table 1 show the results on Street 1 dataset. It can be seen that TextPlace has highest precision except when the recall is bigger than 0.9.
2. Adapting TextPlace to global search based method. The VO tracking and topological graph of TextPlace

are deactivated and, for each query, a global search is performed across the whole map. As shown in Table 1, TextPlace outperforms other global methods.

4.6. Challenging Scenarios with Occlusion

Occlusion can happen frequently for vision based applications or robotics in reality, especially in urban areas where a large amount of buses, cars, trucks and pedestrians appear occasionally. Persistent operation under serious occlusion is important to improve the robustness of a vision based algorithm. Fig. 11 shows some challenging cases successfully matched by TextPlace. It can be seen that the occlusions caused by the dynamic objects significantly change the image representations of a same place at different times. Therefore, it is demanding to request a single image based place recognition system to work in these cases. TextPlace resolves this problem by using high level semantic informa-

Table 1: Precision-Recall on day-night setting (Street 1 dataset).

Prior Methods	Recall				
	0.2	0.4	0.6	0.8	0.9
TextPlace	1	1	1	0.96	0.91
NetVLAD-10	1	1	1	0.95	0.93
NetVLAD-20	1	1	1	0.91	0.87
NetVLAD-30	1	1	0.97	0.85	0.83
ToDayGAN-10	0.5	0.55	0.58	0.57	0.56
ToDayGAN-20	0.4	0.4	0.4	0.38	0.38
ToDayGAN-30	0.26	0.24	0.24	0.25	0.24
FAB-MAP-10	0.79	0.69	0.67	0.65	0.63
FAB-MAP-20	0.76	0.69	0.67	0.63	0.6
FAB-MAP-30	0.68	0.67	0.67	0.62	0.58
SeqSLAM	0.3	0.24	0.18	0.13	0.13

Global Methods	Recall				
	0.2	0.4	0.6	0.8	0.9
TextPlace (Global)	1	1	1	0.92	0.84
NetVLAD	0.98	0.94	0.8	0.76	0.74
ToDayGAN	0.31	0.24	0.13	0.12	0.12
FAB-MAP	0.4	0.42	0.34	0.34	0.34

Table 2: Precision comparison with different system parameter settings for TextPlace on sequence 2 of V4RL dataset.

LD	DBN (meter)				SSW			
	0.5	1	1.5	3	5	8	15	20
0	0.963	0.981	1	1	0.596	0.596	0.981	1
1	0.876	0.95	0.987	1	0.685	0.577	0.95	1
2	0.883	0.988	1	0.86	0.622	0.848	0.988	1
3	0.883	0.976	0.988	0.872	0.689	0.825	0.976	0.988

tion, i.e., scene texts, which is independent of changes on visual appearance.

4.7. Different System Parameter Settings

In this section, we study how different system parameter settings of TextPlace affect the place recognition performance. The parameters studied here are the Levenshtein distance threshold (LD), the displacement between nodes (DBN) and the size of search window (SSW). The various combinations of these parameters represent different recalls. When testing the displacements between nodes, the window size is fixed to 15. The displacement between nodes is set to 1.5 meters when varying the window size.

Table 2 shows the precisions of different system parameter settings on the V4RL dataset. It is clear that TextPlace’s performance improves as the size of search window increases. Meanwhile, the influence of the threshold on the Levenshtein distance can be significantly mitigated if the window size is large. This suggests that a large search window size is preferred. In practice, window size of 15 to 20 usually gives a reasonably decent performance. It can also be seen that dense map nodes (small DBN) does not bring significant benefit, which means the density of the nodes in the topological map can be moderate.



Figure 11: Some challenging cases successfully matched by TextPlace. For each example, left and right columns show query and map images, respectively. Green line connects the matched texts. Note serious occlusions might occur in both map and query.

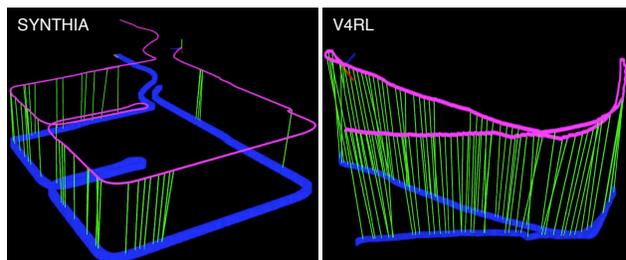


Figure 12: Left: Result of topological localization on SYNTHIA 1. Right: Result of topological localization on V4RL Sequence 1. The blue and pink trajectories represent the map and query trajectories, respectively. The green lines between them are the matches between map and query nodes.

4.8. Localization Performance

Apart from performing place recognition task, TextPlace is also adequate to topological localization. Fig. 12 shows the results of topological localization on SYNTHIA 1 and the sequence 1 of V4RL. It can be seen that the query images or the camera can be accurately localized with respect to the topological map. The matches found (green links) between the query and the map are with high precision, correcting the drifts of visual odometry for localization. A good example is the pose update in SYNTHIA 1 after having no match for a long period.

5. Conclusions

In this work, we propose a novel place recognition system and demonstrate the feasibility of utilizing high-level semantic information, i.e., scene texts, to solve the place recognition and topological localization problems in urban areas. Extensive experiments on various environmental settings verify that the proposed TextPlace can achieve state-of-the-art performance against extreme changes on visual appearance and perceptual aliasing.

Acknowledgements: This work was supported by EPSRC Robotics and Artificial Intelligence ORCA Hub (grant No. EP/R026173/1) and EU H2020 Programme under EU-MarineRobots project (grant ID 731103).

References

- [1] Asha Anooosheh, Torsten Sattler, Radu Timofte, Marc Pollefeys, and Luc Van Gool. Night-to-day image translation for retrieval-based localization. *arXiv preprint arXiv:1809.09767*, 2018. [2](#), [5](#)
- [2] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: CNN architecture for weakly supervised place recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5297–5307, 2016. [1](#), [2](#), [5](#)
- [3] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). In *European Conference on Computer Vision*, pages 404–417, 2008. [2](#)
- [4] Mark Cummins and Paul Newman. Fab-map: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 27(6):647–665, 2008. [1](#), [2](#), [5](#)
- [5] Mark Cummins and Paul Newman. Appearance-only slam at large scale with fab-map 2.0. *The International Journal of Robotics Research*, 30(9):1100–1123, 2011. [1](#)
- [6] Dorian Gálvez-López and Juan D Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, 2012. [2](#)
- [7] Sourav Garg, Niko Suenderhauf, and Michael Milford. Lost? appearance-invariant place recognition for opposite viewpoints using visual semantics. *Robotics: Science and Systems*, 2018. [1](#)
- [8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014. [2](#)
- [9] Albert Gordo, Jon Almazan, Jerome Revaud, and Diane Larlus. End-to-end learning of deep visual representations for image retrieval. *International Journal of Computer Vision*, 124(2):237–254, 2017. [2](#)
- [10] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Reading text in the wild with convolutional neural networks. *International Journal of Computer Vision*, 116(1):1–20, 2016. [2](#)
- [11] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3304–3311, 2010. [2](#)
- [12] Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710, 1966. [3](#)
- [13] Minghui Liao, Baoguang Shi, and Xiang Bai. Textboxes++: A single-shot oriented scene text detector. *arXiv preprint arXiv:1801.02765*, 2018. [2](#), [3](#)
- [14] David G. Lowe. Object recognition from local scale-invariant features. In *IEEE International Conference on Computer Vision*, pages 1150–1157, 1999. [2](#)
- [15] Stephanie Lowry, Niko Sünderhauf, Paul Newman, John J Leonard, David Cox, Peter Corke, and Michael J Milford. Visual place recognition: A survey. *IEEE Transactions on Robotics*, 32(1):1–19, 2016. [1](#), [5](#)
- [16] Fabiola Maffra, Zetao Chen, and Margarita Chli. Viewpoint-tolerant place recognition combining 2d and 3d information for uav navigation. In *IEEE International Conference on Robotics and Automation*, pages 2542–2549, 2018. [5](#)
- [17] Michael J Milford and Gordon F Wyeth. Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In *IEEE International Conference on Robotics and Automation*, 2012. [1](#), [2](#), [5](#)
- [18] Tayyab Naseer, Wolfram Burgard, and Cyrill Stachniss. Robust visual localization across seasons. *IEEE Transactions on Robotics*, 34(2):289–302, 2018. [2](#)
- [19] Tayyab Naseer, Luciano Spinello, Wolfram Burgard, and Cyrill Stachniss. Robust visual robot localization across seasons using network flows. In *AAAI Conference on Artificial Intelligence*, 2014. [1](#), [2](#)
- [20] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007. [1](#)
- [21] Horia Porav, Will Maddern, and Paul Newman. Adversarial training for adverse conditions: Robust metric localisation using appearance transfer. In *IEEE International Conference on Robotics and Automation*, pages 1011–1018, 2018. [2](#)
- [22] Filip Radenović, Giorgos Tolias, and Ondřej Chum. CNN image retrieval learns from bow: Unsupervised fine-tuning with hard examples. In *European Conference on Computer Vision*, pages 3–20, 2016. [2](#)
- [23] Noha Radwan, Gian Diego Tipaldi, Luciano Spinello, and Wolfram Burgard. Do you see the bakery? leveraging geo-referenced texts for global localization in public maps. In *IEEE International Conference on Robotics and Automation*, pages 4837–4842, 2016. [2](#)
- [24] Ananth Ranganathan, David Ilstrup, and Tao Wu. Lightweight localization for vehicles using road markings. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 921–927, 2013. [2](#)
- [25] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3234–3243, 2016. [5](#), [6](#)
- [26] Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(11):2298–2304, 2017. [3](#)
- [27] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *IEEE International Conference on Computer Vision*, pages 1470–1477, 2003. [1](#)
- [28] Akihiko Torii, Relja Arandjelovic, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 24/7 place recognition by view synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1808–1817, 2015. [2](#)
- [29] Olga Vysotska and Cyrill Stachniss. Relocalization under substantial appearance changes using hashing. In *IEEE/RSJ*

International Conference on Intelligent Robots and Systems Workshop, volume 24, 2017. 2

- [30] Hsueh-Cheng Wang, Chelsea Finn, Liam Paull, Michael Kaess, Ruth Rosenholtz, Seth Teller, and John Leonard. Bridging text spotting and slam with junction features. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3701–3708, 2015. 2