




Tracking-DOSeqSLAM: A dynamic sequence-based visual place recognition paradigm

Konstantinos A. Tsintotas  | Loukas Bampis  | Antonios Gasteratos  | FIET

Laboratory of Robotics and Automation,
Department of Production and Management
Engineering, School of Engineering, Democritus
University of Thrace, Xanthi, Greece

Correspondence

Konstantinos A. Tsintotas, Laboratory of Robotics
and Automation, Department of Production and
Management Engineering, School of Engineering,
Democritus University of Thrace, Xanthi, Greece.
Email: ktsintot@pme.duth.gr

Abstract

Simultaneous localization and mapping (SLAM) refers to a process that permits a mobile robot to build up a map of the environment and, at the same time, to use it to compute its location. One of its most important components is its ability to associate the most recently perceived visual measurement to the one derived from previsited locations, a technique widely known as loop closure detection. In this article, we evolve our previous approach, dubbed as *'DOSeqSLAM'* by presenting a low complexity loop closure detection pipeline wherein the traversed trajectory (map) is represented by sequence-based locations (submaps). Each of these groups of images, referred to as place, is generated online through a point tracking repeatability check employed on the perceived visual sensory information. When querying the database, the proper candidate place is selected and, through an image-to-image search, the appropriate location is chosen. The method is subjected to an extensive evaluation on seven publicly available datasets, revealing a substantial improvement in computational complexity and performance over its predecessors, while performing favourably against other state-of-the-art solutions. The system's effectiveness is owed to the reduced number of places, which, compared to the original approach, is at least one order of magnitude less.

1 | INTRODUCTION

Nowadays, robotics researchers have put a tremendous effort in developing methods to map the world through several exteroceptive sensors [1–3]; the reason is the usefulness of an appropriate representation of the surroundings for the robot to be able to perform more elaborate tasks such as path and task planning. It is common though that the use case the robot should deal with impels the map representation. Within the scope of simultaneous localization and mapping (SLAM) methods [4], the robot should estimate its pose as it navigates through the working field. The importance of an efficient and robust estimation is vital for accurate navigation to be achieved. Thus, SLAM is sine qua non in any contemporary autonomous system. The ability to detect and identify a location that has previously been observed is referred to as place recognition. Due to noisy sensor measurements or field abnormalities, drifts occur on the robot's generated map. Such cases are minimised and improved pose

estimation is provided through the accurate detection of loop closures [5–10]. In many contemporary applications, such as in aerial or space robotics, computational resources are restricted. In such cases, efficient methods that provide low complexity, even at the expense of performance, are generally preferred [11–15].

Visual place recognition, the ability to identify a known location in the environment using vision as the main sensory input, is achieved using cameras that provide low-cost means for the generation of extremely rich and dense data [16]. Owing to the increased availability of computational power during the last years, cameras became the primary sensory unit in most autonomous mobile platforms where localisation is based solely on the appearance of the scene [17]. The main key components that constitute such a pipeline are the *image processing module*, the *map* and the *belief generator module*. During the agent's navigation, the received incoming visual measurements are interpreted by the *image processing module* and, subsequently, their arrangement in a topological or metric

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2021 The Authors. *IET Computer Vision* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology.

manner formulates the environment's *map*, which maintains the robot's knowledge about the explored world. Finally, the last module decides whether or not the agent re-encounters an already visited location. Depending on the way by which the system maps the environment, appearance-based systems are distinguished into two main categories, namely single- and sequence-based. Approaches belonging to the first category seek for the most similar location in the robot's traversed trajectory greedily, while methods of the second category search between submaps that is, groups of individual frames, defined as places. In both categories, the *belief generator module* performs data comparisons, aiming to get a score about image similarity based on the way the incoming images are processed. Sum of absolute differences (SAD), local feature vote density [18], bag of words (BoW) histograms [19] or feature vectors derived from convolutional neural networks (CNNs) [20] are the most common techniques. Thus, loop closure events are indicated by comparing similarities between such map representations.

SeqSLAM [21] constitutes one of the most recognised algorithm in sequence-based visual place recognition [22–31] exhibiting the system's performance improvement by comparing places to decide about its position into the world. Using a downsample scheme within the *image processing module*, along with the SAD metric, this framework achieves robust localisation through a *nearest-neighbour distance ratio* [32] technique. However, many challenges arise when breaking the map into places, including optimal submap size, submap overlap during database searching, consistent semantic map segmentation, data duplication and submap alignment [33].

To overcome these challenges, the majority of approaches, including SeqSLAM, use a predefined number of frames to break the map into places. Subsequently, through a sliding window scheme, these approaches look for every possible submap correlation. Although this technique improves the achieved performance, its functionality is computationally costly since the agent needs to seek and compare every possible group of images. As a result, the system's complexity increases since the comparisons are performed for images which might not exhibit the same semantics with their neighbouring ones. Having identified this drawback, in our previous work [34], we proposed an online sequence-based loop closure detection pipeline, wherein a feature matching technique between consecutive images provides the system with dynamically defined places. This approach searches into the traversed trajectory for similar group of images avoiding the sliding window approach of the initial framework, while its *belief module* is based on an average decision filter. The usage of dynamic submaps showed favourable performance against its predecessor; nevertheless, the extraction and matching process of local floating point features for each incoming camera measurement burdened the computational complexity. To build an efficient and independent from training procedure system, we evolve our preliminary framework presenting an appearance-based loop closure detection pipeline which relies on a dynamic place-to-place matching scheme. Local point extraction is performed on the perceived visual information, and through the

Kanade-Lucas-Tomasi (KLT) tracker [35], a new place, that is, a group of images, is defined when contained point tracking fails to advance in the following frame. This way the computationally costly local feature extraction and matching of DOSeqSLAM is avoided, while robustness is achieved regarding the place's size. Next, the camera measurements are subjected to the *image processing module*, used by both the initial approach and our previous work, where instances are downsampled and normalised. At query time, the latest constructed place seeks for the most similar candidate through the matching score produced by the *nearest-neighbour distance ratio*. Finally, the most suitable location is identified via an image-to-image association in the SAD domain avoiding the usage of the average decision filter of our preliminary work. The proposed framework is evaluated in seven different environments, while also compared with its ancestors and other state-of-the-art solutions.

The main contributions of this work are as follows:

- A low-computational place recognition pipeline capable of detecting loop closure events through place-to-place comparisons using almost two orders of magnitude less operations than the initial approach.
- A robust dynamic submapping for place definition based on the extension of the point tracking. Although this process could also be used to potentially detect the dataset keyframes, we define a place by all the frames belonging in the same submap, outlined by the point tracking.
- An exhaustive experimental parameter evaluation scheme based on the number of extracted points.

The remainder of this article is organised as follows. Section 2 provides a brief review of appearance-based place recognition approaches. In Section 3, the proposed pipeline is described in detail, while Section 4 presents evaluation and experimental results. In Section 5, the conclusions are provided.

2 | RELATED WORK

The loop closure detection pipeline proposed in [19] belongs to single-based methods and makes use of the BoW model [36] for representing the incoming image through a pretrained visual vocabulary generated by SIFT descriptors [32]. Additionally, a Chow-Liu tree learns the co-occurrence probabilities among visual words [37]. An improved approximation of this work allows the system to scale by more than two orders of magnitude [38], while 3D information further enhances the system [39]. In [40, 41], a binary vocabulary is proposed and utilised, which is accompanied by a geometrical verification step to enhance loop detection. To improve the speed of the matching process in methods based on binary vocabularies, an incremental feature-based tree is proposed in [42].

The latest works in visual loop closure detection are inspired by the great success of CNNs in several computer vision tasks [43–45]. These approaches address the place recognition task by using specific layers of the architecture to

represent an image and determine potentially revisited locations [46–56]. These layers are originally trained for object recognition; thus, they are tightly bounded to their learning example attributes. NetVLAD [49], an advanced version of VLAD (vector of locally aggregated descriptors [57]), which is commonly used for image retrieval, consists of two trainable end-to-end modules. The first is a CNN extracting the image features and the second is a fusion layer so as to form a descriptor that mimics the behaviour of VLAD. Improving upon CNN-based visual place recognition, Chen et al. [51] trained two neural networks on Specific Places Dataset (SPED), namely AMOSNet and HybridNet. The former is trained from scratch on SPED, while the latter uses the weights of the top 5 convolutional layers from CaffeNet [58] which is trained on ImageNet dataset [59]. CNN-based description of images that utilise only regions of interest (ROI) showed enhanced performance compared to whole-image descriptors. R-MAC (regions of maximum activated convolutions) [60] uses max pooling on cropped areas in CNN layer features to extract ROI. Khaliq et al. [53] combine VLAD with ROI to achieve robustness against appearance and viewpoint variations. However, CNNs require model training from large-scale labelled datasets from a multitude of environments, which is a practical limitation. Their intense computational nature constitutes a key limitation since higher run-time memory and feature encoding time are needed. Thus, despite the impressive results they produce, their high demand in computational resources makes these frameworks unsuitable for mobile robotic applications [61, 62]. More specifically, the above limitations raise deployability concerns on resource-constrained platforms (including battery-powered aerial, micro-aerial and ground vehicles), as identified in [63]. Furthermore, their feature extractor is viewpoint dependent since the topological information is not provided. As a result, they remain incompatible with the majority of SLAM applications in mobile robotics without the utilisation of extra computational power.

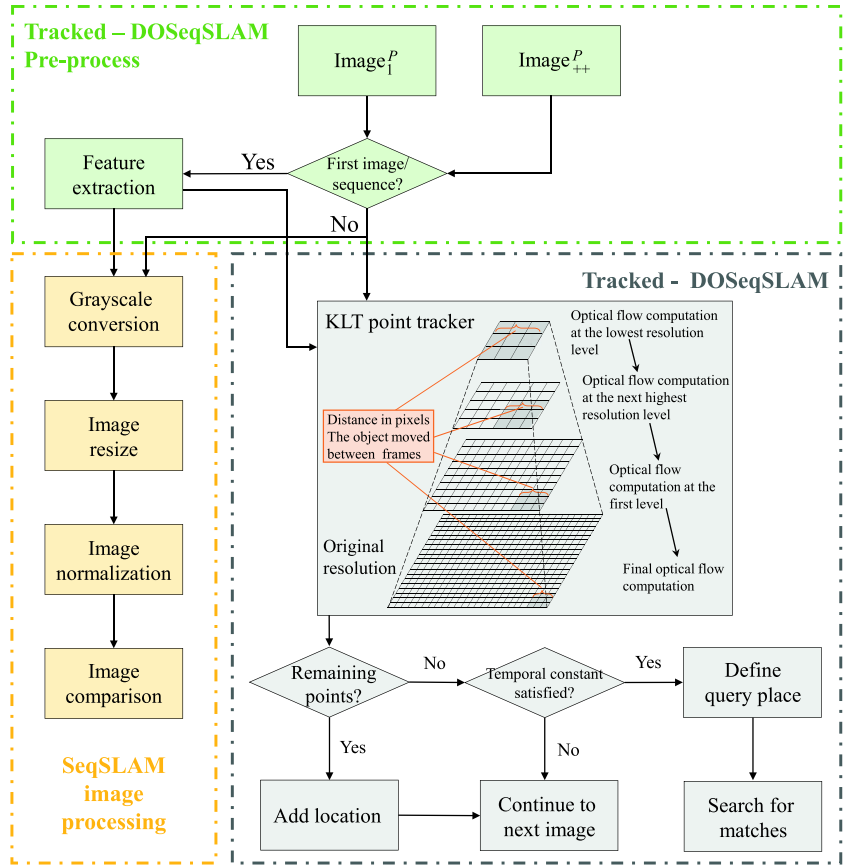
While the aforementioned methods address the place recognition task as a single instance matching process, the sequence-based matching frameworks aim to take advantage of the additional information provided by a group of images in a scene. In [64], a sequence-based algorithm is proposed where the distance between local scenes is used in order to find statistically pairings between places. Similarly, in [65], the incoming visual sensory information is segmented into fix-sized groups of images and represented by a common visual-word histogram. Using a quantitative interpretation of temporal consistency, place-to-place matches that are coherently advancing along time are enhanced [66]. Group of landmarks, formulated through local feature covisibility, generate location graph models [67], while in [68], additional geometrical information from the observed environment structure is used in order to increase the performance. Vysotska et al. [69] build up a data association graph exploiting GPS information to find a sequence of matching images in an offline fashion. An extension of this work exploits hashing

techniques to realise efficient re-localisation when the platform has left the previously mapped area [70].

Compared with the methods which are based on a pretrained description technique, incremental approaches ‘learn’ the environment during the navigation. Two visual vocabularies—one representing image descriptors and the other colour histograms—are generated online indenting to detect loop closures in a Bayesian filtering scheme in [71]. By following the incremental fashion in [72], a visual vocabulary is proposed where the words are generated using a modified version of agglomerative clustering. Since mobile robots have limited computing resources, an incremental loop closure detection approach for large scale and long term is proposed in [73]. Most of these frameworks tackle the loop closure detection task through the location polling of the distributed votes originated by the local feature descriptors. In [74], regions of high vote density are selected as loop closure candidates via a nearest neighbouring descriptor technique. The IBUILD algorithm [75] proposes a binary vocabulary wherein the extracted features are matched across consecutive images. In [76], images with similar visual properties are stored in groups formulating a hierarchical architecture of places, each of which is represented by a global descriptor. The method selects the candidate loop closing place through a query-to-database comparison of the global descriptors. Subsequently, the most likely match is retrieved through an extensive search in the local feature space. A new approach was recently introduced by the same authors [77], in which dynamic islands were used to group the images based on spatiotemporal similarity. Probabilistic voting schemes utilise the number of aggregated votes in the database to compute a score that indicates previsited locations [18]. A dynamic sequence segmentation is performed based on the image content proximity, while a clustering technique generates the visual words assigned to these specific places [78]. Temporal and geometrical checks are also included for the sake of performance improvement in the incrementally constructed vocabulary of tracked words [79]. In a similar manner, a modified version of growing self-organising maps [80] is proposed by [81]. Using Gist features [82] for representing places, the map is incrementally learnt, while the most active neuron is selected as loop closure candidate during query.

Working with sequence-based methods, many robotics scholars evolve the well-known SeqSLAM [23–27]. Querying the traversed trajectory, via a Bayes filter, a subset of candidate places are indicated, and through an extended evaluation on the selected sequences, a faster version of SeqSLAM is provided [23]. Likewise, the authors in [24] propose an efficient version, titled ‘Fast-SeqSLAM’. In this work, place matching is achieved using a histogram of gradients [83] to describe the downsampled images, along with a k -d tree [84] and a nearest neighbour classifier on the descriptor space. In the work of Wang et al. [25], a real-time framework is presented. The visual sliding window technique accompanied by odometry information provides loop closure candidates, while a multiscale

FIGURE 1 An overview of the proposed sequence-based loop closure detection framework. As the incoming visual sensory information arrives (I_1^p) to the pipeline, points are extracted via the speeded-up robust features (SURF) detection and description algorithm [88] each time a new place begins its generation procedure. Subsequently, the camera measurement follows the SeqSLAM's [21] processing steps where it is being downsampled and normalised before compared to the previously visited locations in the database. When the following image (I_{++}^p) enters the system, points are tracked through the Kanade-Lucas-Tomasi (KLT) method [35] to dynamically define places. Finally, when point tracking is lost and the temporal constant is satisfied, the database is queried with the last formulated place



search at the selected group of images indicates the best location match. By utilising the BoW model for representing images, SeqSLAM obtains robustness against scale and rotation variations, while the binary vocabulary generated by ORB descriptors [85] offers a low computational pipeline [26]. Dongdong et al. [27] proposed 'SeqCNNSLAM', wherein pretrained CNN output layers are utilised as image descriptors. Comparisons are performed among CNN feature vectors and sequence matches are accomplished by following the original version steps. In [30], a technique is proposed that combines properties from two existing visual place recognition methods, which do not depend on learning examples, that is, SeqSLAM and CoHOG [86]. A lightweight system wherein places are compressed and represented by compact codes is proposed in [29]. Finally, Chen et al. [46] used features from all layers of Overfeat Network [87] and integrated it into the spatial scheme of Seq-SLAM.

The majority of the aforementioned algorithms in sequence- and appearance-based place recognition relies on SeqSLAM coupled with a pretraining technique for describing the images or the addition of extra information along with the incoming visual sensory measurements. However, our previous work [34] focuses on the dynamic segmentation of the traversed path for defining places in order to avoid the sliding window scheme. Although our approach performs favourably against the initial pipeline,

submaps generated via feature matching tend to lose their local feature coherence sooner than expected, while the system's complexity remains high since feature extraction is implemented for every image. Yet, the proposed framework improves on its predecessor by adapting a point tracking technique among consecutively acquired images to determine places, while keeping its original online behaviour independent from any training procedure.

3 | METHODOLOGY

In this section, an extended description of the proposed loop closure detection pipeline is presented. As mentioned previously, the algorithm formulates each place dynamically through the KLT point tracker. To carry out the submap definition, local keypoints are extracted from the camera measurements. Subsequently, the SeqSLAM processing steps are performed with the data being downsampled and normalised. Each image is compared to the database through SAD, and when a temporal constant is satisfied, the database is searched for a candidate place match. Since the main algorithm follows the initial approach, a brief description of our previous work is provided. An outline of the proposed visual place recognition workflow is shown in Figure 1.

3.1 | DOSeqSLAM

3.1.1 | SeqSLAM procedure

For each image I entering the system, the visual data is converted into the greyscale equivalent and then is down-sampled into χ pixels. In addition, the resized ones are normalised in an N size local neighbourhood and comparisons with the traversed trajectory are performed by means of SAD:

$$D_{ij} = \frac{1}{R_x R_y} \sum_{x=0}^{R_x} \sum_{y=0}^{R_y} |\rho_{x,y}^i - \rho_{x,y}^j|, \quad (1)$$

where R_x and R_y denote the reduced dimensions of the images, while ρ represents each pixel's intensity value. A vector D_i for location i containing distance metric against every pre-visited one j is generated through SAD, resulting into comparison matrix D . Focussing on the comparison between sequences of images during the query procedure, a contrast enhancement process is performed on the D_i elements, which is analogous to a 1D patch normalisation in a local area of ϵ pixels:

$$\widehat{D}_i, \mu = \frac{D_{i,\mu} - \overline{D}_\epsilon}{\sigma_\epsilon}, \quad (2)$$

where \overline{D}_ϵ represents the local mean and σ_ϵ the local standard deviation around element μ .

3.1.2 | Dynamic sequences

In order to define a dynamic group of instances in DOSeqSLAM, local keypoints are detected via the SURF method [88] from each incoming visual sensory data. Utilising the full space, projected SURF descriptors (d_j) are temporarily extracted during the online operation as long as the place construction lasts. Through a feature matching coherence check, new places are determined along the robot's traversed path. Additionally, in cases where the input camera measurement is unable to produce enough visual information, for example, the system observes a blank plane, the pipeline skips those images avoiding the construction of inconsistent places.

More specifically, at time t , the incoming image stream $I_{(t-n)}, \dots, I_{(t-2)}, I_{(t-1)}, I_{(t)}$ is segmented when the correlation between the last n image descriptors ceases to exist:

$$\left| \bigcap_{i=0}^{i=n} d_{I_{(t-i)}} \right| \leq 1, \quad (3)$$

where $|S|$ denotes the cardinality of set S .

3.1.3 | Sequence-based search

In order to identify a previously visited location (database), searching is based on place comparisons. During the system's navigation, when the latest sequence Seq_N is created, querying the database is performed for the first frame in the previous generated place Seq_{N-1} . A number of trajectories are projected on the enhanced distance matrix \widehat{D} for every traversed location j . The trajectory lengths are proportional to the query place size. Each trajectory represents a possible velocity assumption corresponding to different robot velocities V . A number of multiple scores s_{do} are calculated for every trajectory assumption by averaging the accumulated values:

$$s_{do} = \frac{1}{seq_{Len}^Q} \sum_{I_1^Q}^{I_{end}^Q} \widehat{D}_k. \quad (4)$$

where I_1^Q and I_{end}^Q are the first and last image timestamps of the query sequence, respectively, seq_{Len}^Q is the query length and k denotes velocity assumption paths:

$$k = j + V(L - i + t), \quad (5)$$

where V is designated by multiple values within the range of $[V_{min}, V_{max}]$ (advancing by V_{step} each time step t) and L represents the sequence length.

The minimum score s_{do} is selected for each instance in the navigated path, yielding an S_{do} vector, wherein the lowest value is selected for the particular location I_j . Subsequently, this score is normalised over the second lowest value outside of a window W_{DOSeq} [34] resulting to γ . Lastly, an average weighted filter is applied for the final decision. A candidate loop closure sequence is determined when factor γ is satisfied and the system performs an additional greedy image-to-image search into the SAD submatrix for single-image associations.

3.2 | Tracking-DOSeqSLAM

Feature tracking is essential for several high-level computer vision tasks such as motion estimation [89], structure from motion [90] and image registration [91]. Since the earliest works, feature trackers have been used as a de facto tool for handling points in a video. We have chosen to use a tracker based on a floating point, local feature detection and description algorithm during the navigation procedure. Through point tracking we achieve to dynamically segment the incoming visual stream and determine a place. This way, the computationally demanding procedure of feature detection and description for every incoming frame, which is used in our previous work [34], is avoided.

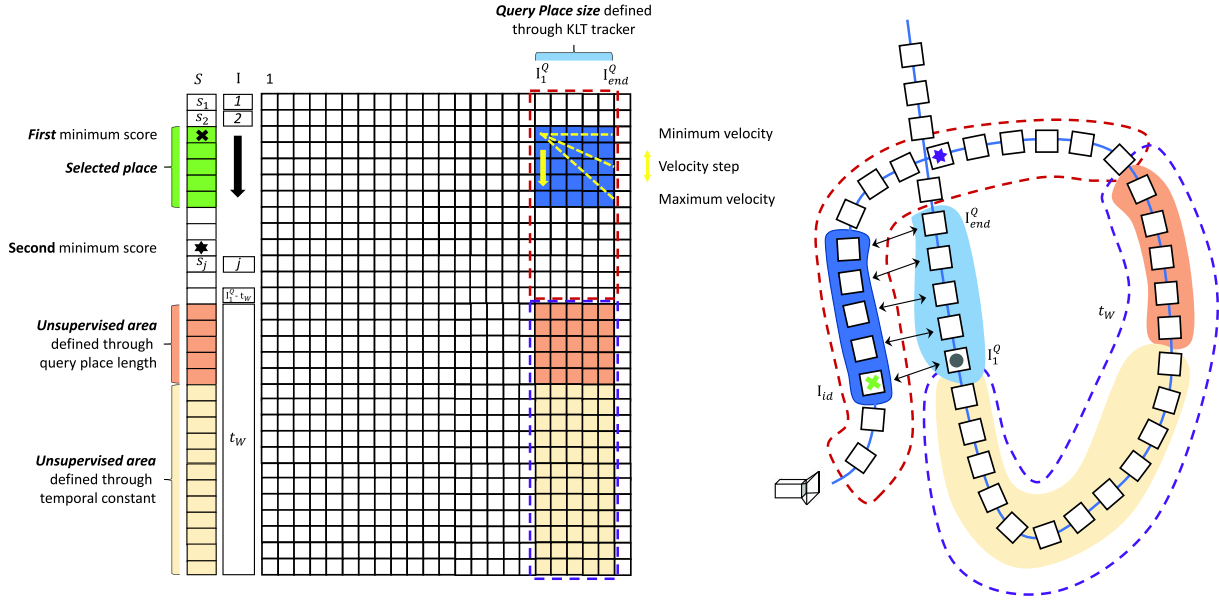


FIGURE 2 In order to define places in the proposed pipeline, local keypoints are extracted via the speeded-up robust features (SURF) [88] detector for the first image of each place (grey circle). Through the Kanade-Lucas-Tomasi (KLT) method [35], points are tracked along the traversed path, while a new place is determined when each tracked point is lost. The query process begins when a temporal window based on a time constant and query length is satisfied (beige and light orange area). The latest generated place (light blue area) seeks for similar places along the navigated path in a sequence-based scheme. Each visited location (j) is associated with a score (s_j) which indicates the nearest neighbouring trajectory assumption (yellow dashed line). The selected images (green box in S vector) point out the proper place and an image-to-image search is subsequently performed

Algorithm 1 Place definition

```

Input:  $I$ : Incoming image,  $P$ : Place index,
           $L^P$ : Place length
Output:  $P$ : Place index,  $L^P$ : Place length
1 if  $L^P == 0$  then
2    $SP_I = \text{detectSURF}(I)$  //extract SURF
   keypoints from  $I$ 
3    $TP_{I-1} = \text{KLT}(SP_I)$  //initialize tracked
   points
4    $\text{numTrackedPoints} = \text{sum}(TP_{I-1})$ 
5    $L^P++$ 
6 else
7    $TP_{I-1} = \text{KLT}(TP_I, I)$  //track points in  $I$ 
8    $TP_{I-1} = TP_I$  //set tracked points for next
   iteration
9    $\text{numTrackedPoints} = \text{sum}(TP_{I^P})$ 
10   $L^P++$ 
11 end
12 if  $\text{numTrackedPoints} < 1$  then
13    $P++$ 
14    $L^P = 0$ 
15 end

```

3.2.1 | Place definition through tracking

In contrast to our previous work, in the proposed framework, the place definition is based on a repeatability check of point occurrence between consecutive frames. A set of ξ SURF keypoints ($SP_{I^P} = \{sp_{I^P}^1, sp_{I^P}^2, \dots, sp_{I^P}^\xi\}$) is detected in the first location of each place (I_1^P). Subsequently, the points are fed into a KLT tracker along with the next perceived visual measurement (I_{++}^P), yielding to a set of tracked points ($TP_{I_{++}^P} = \{tp_{I_{++}^P}^1, tp_{I_{++}^P}^2, \dots, tp_{I_{++}^P}^\xi\}$). Points in I_{++}^P are browsed within three levels of resolution, around a 31×31 patch allowing our system to handle large displacements between frames. In such way, we achieve to generate robust places, even if occlusions occur due to moving objects, as evidenced by the experimental evaluation in Section 4.3. Furthermore, to propose a pipeline with low complexity, we avoid the computation of bidirectional error between points. In addition, as the algorithm progresses over time, points tend to gradually be lost due to lighting variation or out of plane rotation. At time t , when every point repeatability expires, the previous visual sensory stream $I_{(t-n)}, \dots, I_{(t-3)}, I_{(t-2)}, I_{(t-1)}$, is determined as a new place:

$$\left| \bigcap_{i=n}^{i=0} \text{TP}_{I_{(t-i)}}^P \right| \leq 1, \quad (6)$$

Finally, two important components are retained during navigation: (i) the place index P and (ii) its length L^P . Algorithm 1 summarises this process.

3.2.2 | Image modulation

Next, the pipeline follows the process described in Section 3.1.1 in order to keep the initial SeqSLAM characteristics. Incoming frames are scaled down to χ pixels and are then normalised. Comparisons between the query, that is, the current robot view, and the traversed locations are achieved via SAD, yielding to generation of distance matrix D . However, the contrast enhancement step is omitted in the proposed pipeline since it constitutes an essential component when the system confronts changing environments [66].

3.2.3 | Place-to-place association

When a place is determined, the query procedure starts. With the aim to perform reliable searching for similar submaps, the newly generated place P_Q should not share any common semantic information with the recently visited locations. This is due to the fact that a set of input frames obtained during a short time interval before I_t are expected to be similar without corresponding to actual loop closure events. To prevent our pipeline from detecting such cases, we consider a temporal window t_W , which rejects locations visited just earlier ($I_1^Q, \dots, I_1^Q - t_W$). We define this window based on a temporal constant ψ and the place length L^P :

$$t_W = \psi + L^P. \quad (7)$$

This way, the searching area spans among the first perceived location I_1 and the one determined by the temporal window $I_1^Q - t_W$ as depicted in Figure 2 by the red dashed line. The latest produced submap seeks into the navigated path for similar places via a sequence-based technique. For each database location I_j , belonging to the searching area, a difference score s is calculated (Equation (4)) for each velocity assumption (Equation (5)). These scores are based on the values the trajectory line passes through in travelling from I_1^Q to I_{end}^Q (Figure 2). The trajectory with the minimum s value is selected as the representing score s_j between the query place and the one starting from frame I_j . When all database images have been examined, a score vector is determined $S = \{s_1, s_2, \dots, s_{I_1^Q - t_W}\}$ and subsequently the minimum value is selected corresponding to the start location I_{id} of the candidate submap. Next, following the *nearest neighbour distance ratio* [32] the chosen score is normalised over the second lowest score (Figure 2) outside of a window range of

equal size with the place length L . The normalised score, which is the ratio between these scores, is calculated for each place, while one of the following conditions has to be satisfied before a submap is recognised as previsited. The recent score has to be lower than a threshold σ ($\sigma < 0.7$ [21]) or the score generated by the last two consecutive submaps to satisfy a threshold λ . This temporal consistency check is incorporated in the proposed pipeline since loop closure detection is a task submitting to a temporal order of the visited places along the navigation route. That is, if a place is identified as previsited, then it is highly probable that the following ones have also gone through. This way, we achieve to improve the system's performance, while we avoid to lose actual loop detections due to strict thresholding. Algorithm 2 illustrates this process.

Algorithm 2 Detecting loop places

Input: D : Difference matrix, P : Query place index, L : Query place length, f : dataset's frame rate

Output: id : Candidate index, $score$: Candidate score

```

1  $t_w = 40 * f + L$  //temporal window definition
2 for each image  $I_j$  in Database do
3    $T = \text{computeTrajectoryScores}(I, D, P, L)$ 
4    $t = \min(T)$ 
5    $S(I) = t$ 
6 end
7  $[id, score1] = \min(S)$  //find the minimum
   score and candidate index
8  $e = [I_{id-L/2}, \dots, I_{id+L/2}]$  //define images
   around  $I_{id}$ 
9  $S(e) = 1 \setminus \setminus$  reject images in  $e$ 
10  $[\sim, score2] = \min(S)$  //find the second
   minimum score excluding images in  $e$ 
11  $score = score1 / score2$  //compute the
   normalised score for  $I_{id}$ 

```

3.2.4 | Local best match

Up to this point, the proposed algorithm is capable of identifying a pre-visited place in the navigated map. Finally, an image-to-image correlation is performed between the query locations and the most similar members of the selected submap in the database. Hence, each place member is associated with the most similar from the corresponding ones in the matched database image through the SAD submatrix. Let us consider that at time t , the system correctly indicates a previously visited place by matching pair $\langle I_1^Q, I_{id} \rangle$. Our method defines a group of images which are the only set of database entries that are going to be evaluated through SAD metrics. In this paper, we determine this group to be of double the size of

TABLE 1 Description of the benchmark datasets used for evaluating and testing the proposed system

Dataset Label	Description	Camera position	Image resolution	Imagesnumber	Framesper second
KITTI 00 [92]	Urban environment obtained by means of a stereo camera system mounted on a forward moving car.	Frontal	1241 × 376	2761	10
KITTI 02 [92]	Urban environment obtained by means of a stereo camera system mounted on a forward moving car.	Frontal	1241 × 376	4551	10
KITTI 05 [92]	Urban environment obtained by means of a stereo camera system mounted on a forward moving car.	Frontal	1241 × 376	4661	10
Lip6 Outdoor [71]	Urban environment surrounded by houses recorded via a handheld camera.	Frontal	240 × 192	600	0.5
City Centre [19]	Public roads near the city featuring many dynamic objects such as traffic and pedestrians recorded via a mobile robotic platform.	Lateral	1024 × 768	1237	7
New College [93]	College's ground recorded by means of the vision system of a robotic platform.	Frontal	512 × 384	52,480	20
Malaga 2009 Parking 6L [94]	University parking recorded by means of the vision system of an electric buggy-typed vehicle.	Frontal	1024 × 768	3474	7.5

camera frequency κ , while it is centred around I_{id} for I_1^{PQ} , that is, $I_{(id-1)-\kappa}, \dots, I_{(id-1)+\kappa}$. However, for the following image in the query place I_2^{PQ} this area shifts by 1.

4 | EXPERIMENTAL SETUP

This section provides a description of the experimental procedure, an expansive evaluation of the proposed pipeline, as well as comparative results. A total of seven publicly available datasets are selected for assessing our method. The present approach is compared in terms of precision–recall metrics [38] with our previous work, the baseline version of SeqSLAM as well as other well-known place recognition solutions. All experiments were performed on an Intel i7-6700HQ 2.6 GHz processor with 8 GB of RAM.

4.1 | Datasets

The chosen environments represent outdoor, static and dynamic areas containing mostly urban views. In addition, a variety of different measurement properties are selected, for example, robot velocity, image resolution and frame rate, in order to examine the system's adaptation in different conditions. In Table 1, a summary of each data sequence used is provided. Three out of seven datasets belong to the KITTI visual collection [92], representing urban environments that mostly consist of houses, cars and trees. The perceived visual sensory information is obtained by a stereo camera system mounted on a car, while the recorded data offer considerable loop closure events, in addition to accurate odometry and high resolution images. Furthermore, image sequences 00 and 02 are selected in order to examine the algorithm in long-term operations, since the vehicle traverses a distance over 11 km. Lip 6 Outdoor [71] provides information perceived via a handheld camera

TABLE 2 Parameters utilised from the proposed pipeline. Most of the reported values come from the OpenSeqSLAM implementation, while the rest are selected by means of the experimentations.

Parameters	Symbol	Value
Downsampled image size	χ	2048
Patch normalisation length	N	8
Reduced image size	R_x, R_y	32, 64
Minimum velocity	V_{min}	0.8
Velocity step	V_{step}	0.1
Maximum velocity	V_{max}	1.2
Extracted SURF points	ξ	500
Search area time constant	ψ	40 sec. [78]

encountering mostly buildings, while a high amount of loop closures along the navigated path are presented. The particular dataset is chosen to test the robustness of the system since it includes variations in the orientation and velocity of the incoming image stream, as well as low camera resolution and frame rate. City Centre [19] and New College [93] have been registered by the vision system of a robotic platform. They refer to significantly different operational conditions (e.g. travelled distance, frame size, acquisition frequency, camera orientation), as presented in Table 1. However, they both contain a significant amount of loop closure examples. Note that the acquisition frequency of New College was resampled to one frame per second, from its initial 20 Hz rate, due to the robot's low velocity and high camera frequency. In Malaga 2009 [94], the Parking 6L (Malaga 6L) data sequence was selected. This environment mostly contains cars and trees, while the camera information is provided by means of a vision system mounted on an electric buggy-typed vehicle. Plenty examples of revisited locations are

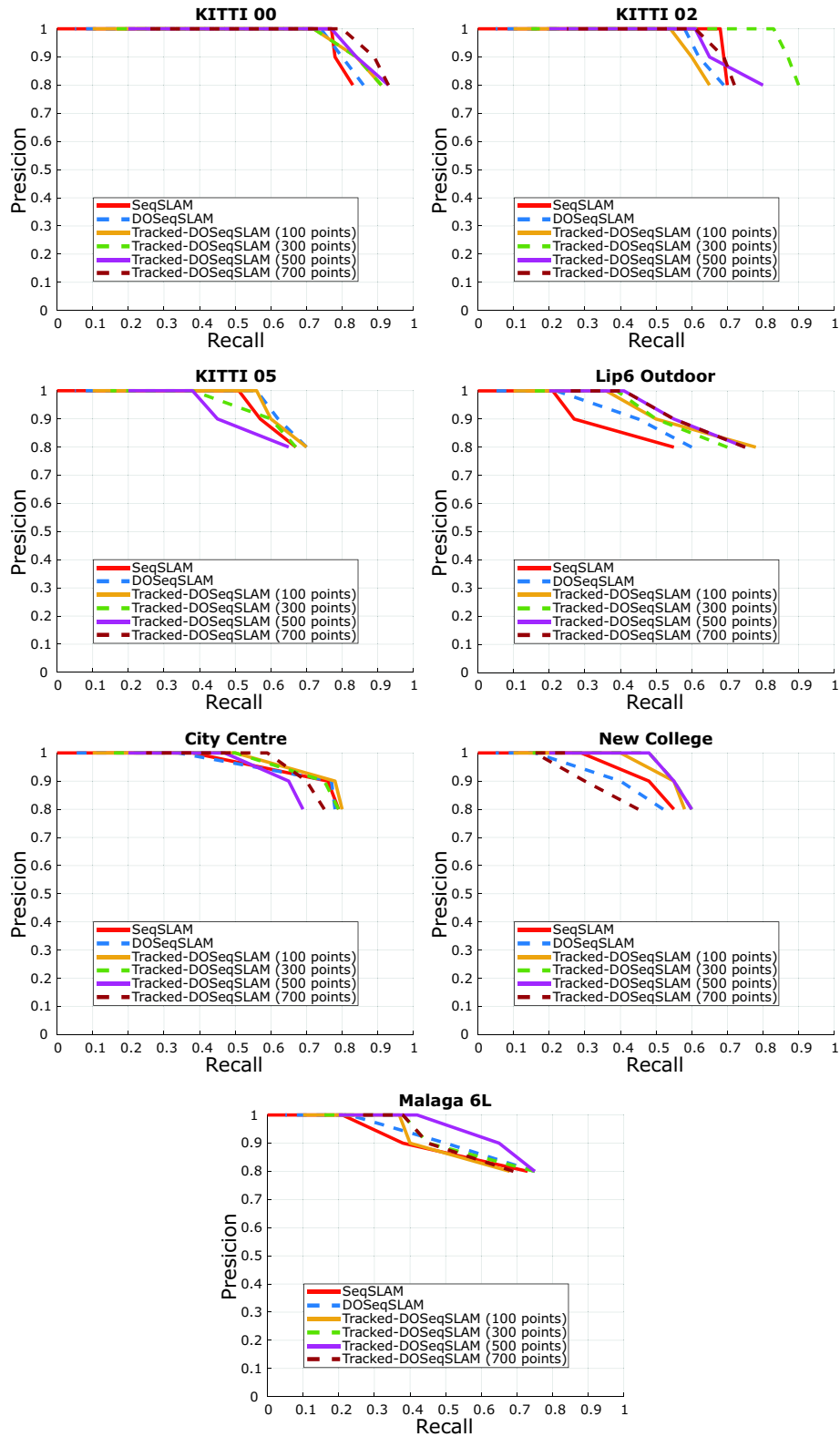


FIGURE 3 Precision–recall curves of the proposed pipeline evaluating the utilised number of extracted speeded-up robust features (SURF) [88] ξ against the previous approach [34] and the baseline solution of SeqSLAM [21]. Experiments are performed on the KITTI 00, 02 and 05 data sequences [92], Lip6 [71] Outdoor, City Centre [19], New College [93] and Malaga [94] 6L. As the number of detected points increases, our proposed system presents a slight improvement, reaching recall values of about 77% in case KITTI 00%, 85% in KITTI 02% and 56% in KITTI 05. In Lip6 Outdoor, a score of 50% is achieved, while a similar performance is observed for the rest datasets. However, the performance falls drastically when feature extraction exceeds the amount of 500, points as evidenced in KITTI 05 and New College. This is mainly owed to the resulting size of the generated submaps which fail to be matched with the ones in the traversed trajectory. The proposed system offers higher performance against its predecessors for the rest of the evaluated datasets

TABLE 3 Recall rates at 100% precision: a comparison of the proposed method against our previous work [34], as well as the baseline approach of SeqSLAM [21]. Bold values indicate the maximum performance per evaluated image sequence. As shown from the obtained results, the proposed pipeline outperforms the previous versions, while performance improvement is observed as the extracted set of points increases until a certain point. Aiming for an efficient system which preserves high recall scores for 100% precision, the case of 500 points is indicated.

Dataset	SeqSLAM [21]	DOSeqSLAM [34]	Tracking-DOSeqSLAM (100 points)	Tracking-DOSeqSLAM (300 points)	Tracking-DOSeqSLAM (500 points)	Tracking-DOSeqSLAM (700 points)
KITTI [92] 00	77.3	74.8	69.8	72.1	77.6	80.1
KITTI [92] 02	68.2	58.9	54.6	83.6	61.1	61.1
KITTI [92] 05	51.5	56.7	56.7	38.4	38.2	–
Lip6 [71] Outdoor	21.2	22.5	36.8	39.8	40.9	40.9
Oxford [19] City Centre	38.9	34.9	50.4	50.4	47.1	59.4
Oxford [93] New College	29.3	16.8	39.9	40.0	40.0	16.3
Malaga [94] Parking 6L	21.5	23.3	37.2	38.4	42.0	38.1

also presented. The incoming visual stream in most sequences is provided by a stereo camera rig; however, since our approach aims to an appearance-based pipeline, only the monocular capture was used. For City Centre, New College and Malaga 6L, the right visual stream was selected, while for the KITTI sequences the left one.

4.2 | Evaluation protocol

In this section, an evaluation protocol for the proposed framework is presented in detail. Precision–recall metrics along with the ground truth (GT) information are utilised in order to assess the algorithm performance. Comparisons were performed based on the parameters in Table 2. Those values remain constant for every tested environment, so as to prove the adaptability of the algorithm. It is notable that the proposed approach is able to achieve high recall rates for 100% precision than any of its predecessors on most of the evaluated datasets.

4.2.1 | Parameter discussion

In this section, we briefly discuss the system’s chosen parameters. In general, most of the proposed values, for example downsampled image size χ , image reduced size R_x, R_y , are defined similarly to the initial version of SeqSLAM [21]. Velocity properties $[V_{max}, V_{min}, V_{step}]$ come from the open source implementation of OpenSeqSLAM¹ [22], while the normalisation parameter N is defined based on the open-SeqSLAM2.0 MATLAB toolbox² [95]. Extracted SURF points ξ defined via the precision–recall metrics in Figure 3 with the aim to achieve a framework exhibiting high performance.

¹The OpenSeqSLAM algorithm can be found online in: <https://openslam.org/openslam.html>

²The OpenSeqSLAM2.0 toolbox can be found online in: <https://github.com/kadn/OpenSeqSLAM2.0>

4.2.2 | Ground truth

The binary matrix whose rows and columns correspond to different timestamps indicating the actual loop closure events occurring in a dataset is defined as ground truth. The presence of an $GT_{ij} = 1$ element denotes the existence of a loop and $GT_{ij} = 0$ otherwise. For the KITTI 00, 02, 05 and New College data sequences, the GT was manually generated in [79] through odometry information. In Lip6 Outdoor, this information is provided by the authors in [71]. Similarly, City Centre contains its own GT, while Malaga 6L was manually labelled by the authors in [52].

4.2.3 | Precision–recall metric

A true-positive detection concerns the correct match as indicated by the GT. As a correct match is considered any recognition occurs within a small radius from the query location. On the contrary, as false-positive detection is defined any identification occurs outside of this area, while false-negative detections are the ones that the loop closure detection system ought to have identified but failed to. The tolerance used for the evaluation is 40 m. Thus, precision is the ratio between true positives over the total system’s detections:

$$\text{Precision} = \frac{\text{True positive}}{\text{True positive} + \text{False positive}}, \quad (8)$$

whereas recall is defined as the number of true positives over the sum of loop closure events contained in GT:

$$\text{Recall} = \frac{\text{True positive}}{\text{True positive} + \text{False negative}}. \quad (9)$$

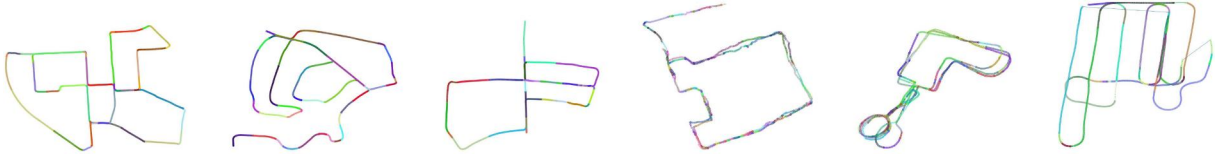


FIGURE 4 Submaps generated from the proposed dynamic segmentation of the incoming image stream using the parameters defined in Table 2. Images exhibiting time and content proximity are labelled by the same colour. From left to right, submaps are illustrated for KITTI data sequences [92] 00, 02, 05, City Centre [19], New College [93] and Malaga 6L [94]. 47, 52, 22, 151, 119 and 43 places are generated, respectively. As can be seen in most of the cases, the images are tagged with the same colour when the robot traverses a route which presents similar visual content. This is especially highlighted in the KITTI datasets, where the camera measurements arrive from a forward moving car, in contrast to City Centre’s lateral camera orientation

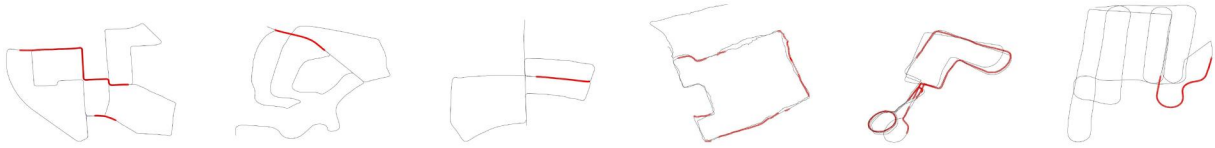


FIGURE 5 Loop closures detected by the proposed pipeline for each dataset trajectory. From left to right: KITTI [92] 00, 02, 05, City Centre [19], New College [93] and Malaga 6L [94]. Red dots indicate that the system closes a loop with another image in the database

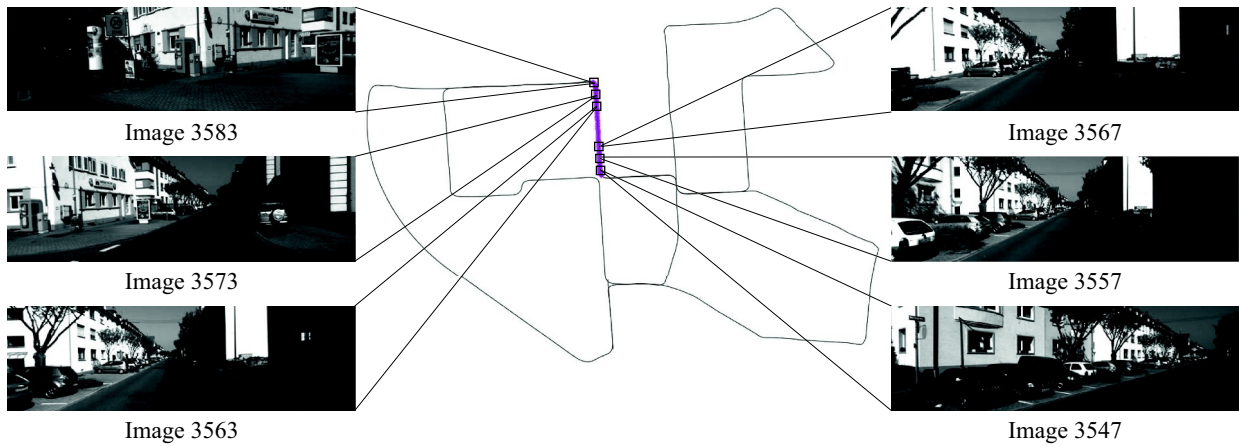


FIGURE 6 An illustrative example of our place generation technique based on point tracking. The respective camera poses corresponding to the same group of images are marked in magenta. A set of speeded-up robust features (SURF) [88] is detected in the first location of a newly formulated place (image 3547) and subsequently tracked along the trajectory. At time t (image 3583), the incoming visual sensory stream $I_{(t-n)}, \dots, I_{(2)}, I_{(1)}, I_{(t)}$, is finalised as a new submap since all the initial points cease to exist from the tracker

4.3 | Performance evaluation

By altering the loop closure decision parameter λ , precision–recall curves are monitored for different cases of image keypoints detection ($\xi = 100, 300, 500, 700$) in Figure 3. The system’s performance for the proposed dynamic place generation is evaluated and compared with the previous version of DOSeqSLAM, as well as the baseline approach of SeqSLAM. The latter is based on the open-source implementation of OpenSeqSLAM, while configured through OpenSeqSLAM2.0 toolbox [95]. The selected parameters remained constant over all datasets. However, aiming to a fair performance evaluation, the contrast enhancement step was avoided for both previous methods of SeqSLAM and DOSeqSLAM. Furthermore, a 40 s temporal window, similar to the proposed method, was applied

to reject early visited locations. For an easier understanding of the curves, best results at 100% of precision are also presented in Table 3. Our first remark is that the area under the curve of Tracking-DOSeqSLAM is higher than the corresponding curves for its predecessors, outperforming them in most of the evaluated datasets. As can be observed, DOSeqSLAM is usually able to obtain similar recall at perfect precision as SeqSLAM, except for New College, where the result drops to a rate of 17%. According to our experiments, the proposed pipeline shows especially high performance for Lip6 Outdoor, City Centre and Malaga 6L, for each case of the extracted keypoints, compared to the other solutions. Furthermore, the maximum scores for the other datasets are also high, while a high improvement is observed in KITTI 02 for a number of 300 keypoints, reaching a score of about 85% for perfect precision.

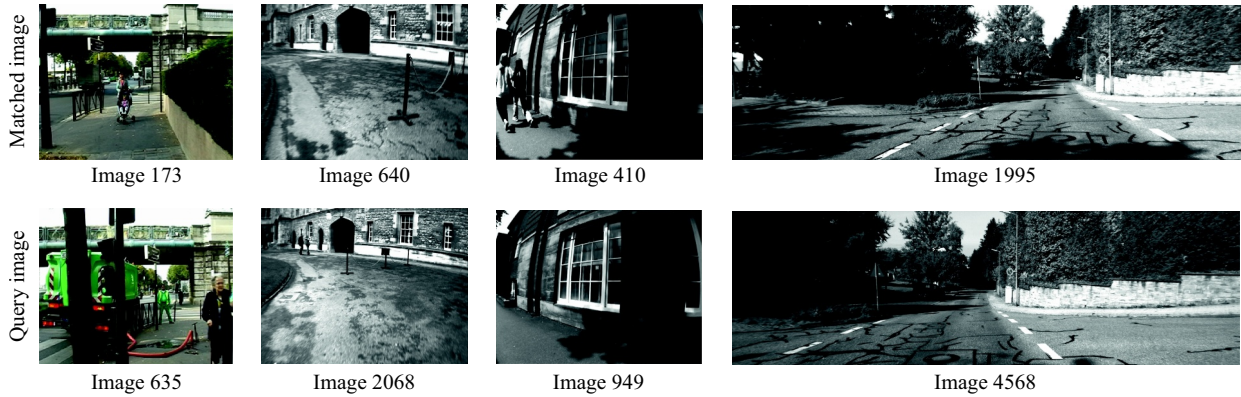


FIGURE 7 Some example images that are correctly recognised by our pipeline as loop closure events. The query frame is the image recorded by the vehicle at time t , whereas the matched frame is the corresponding one identified among the members of the chosen place. From left to right: Lip6 Outdoor [71], New College [93], City Centre [19] and KITTI 02 [92]

TABLE 4 Processing time per image (ms/query) of Tracking-DOSeqSLAM, as well as of its previous version [34] and the baseline approach [21], for the KITTI 00 data sequence. It is notable that the proposed pipeline requires less time due to its efficient matching process which is based on the image aggregation from the generated places.

	Average time (ms)		
	SeqSLAM	DOSeqSLAM	Tracking-DOSeqSLAM
SURF detection	–	42.67	0.01
SURF description	–	27.96	–
Points tracking	–	–	4.27
Feature matching	–	6.75	–
Resize	2.43	2.43	2.43
Patch normalisation	5.77	5.77	5.77
Image comparison (SAD)	42.90	42.90	42.90
Matching	67.66	64.18	0.71
Sum	118.76	192.66	56.20

Nevertheless, counter to most data sequences, where the increased keypoint extraction improves the performance, the evaluation of the proposed method in KITTI 05 and New College shows an instant drop in the recall rate. In the latter case, we observe a lower recall score, while in the former one our method does not recognise any revisited location. This is owed to the fact that places which are generated under those conditions fail to be matched with the query ones due to their extreme size. By considering the results presented in Table 3, the parameter ξ is selected at 500 in order to ensure a system that achieves high recall scores for 100% precision. Figure 4 shows the submaps formulated by Tracking-DOSeqSLAM for each dataset, while Figure 5 presents the detected loops for 100% precision. For each submap, a random colour has been assigned to highlight a distinct place across the traversed trajectory, and thus, every location associated to the same submap is labelled by the same colour. An example containing images from the same place defined by our algorithm based on point tracking is illustrated in Figure 6. Evidently, as soon as the robot turns to a route that represents a visually consistent area, the corresponding images that exhibit time and content

TABLE 5 Average computational times (T) and number of generated places for the proposed pipeline and its predecessors for all datasets. As can be observed, the improved version achieves substantially lower timings for each evaluated case, outperforming the rest of the solutions.

	SeqSLAM		DOSeqSLAM		Tracking-DOSeqSLAM	
	Places	T(ms)	Places	T(ms)	Places	T(ms)
KITTI 00	4554	118.76	155	192.66	47	56.20
KITTI 02	4661	123.04	378	209.02	52	58.99
KITTI 05	2761	58.62	192	132.90	22	38.66
Lip 6 Outdoor	600	22.13	177	37.56	51	20.87
City Centre	1237	26.96	211	71.41	151	25.47
New College	2624	55.51	323	94.70	119	36.28
Malaga 6L	3474	81.73	162	186.22	43	48.97

proximity are aggregated in the same group (place). Finally, in Figure 7, some accurately detected locations using the selected parameterisation are shown.

TABLE 6 Recall scores for 100% precision: comparison of the proposed place recognition pipeline against other state-of-the-art methods

Dataset	FAB-MAP [38] R (%)	DBow2 [40] R (%)	SeqSLAM [21] R (%)	DOSeqSLAM [34] R (%)	BoW-SeqSLAM [26] R (%)	FILD [52] R (%)	Kazmi and Mertsching [81] R (%)	Proposed R (%)
KITTI 00	61.2	72.43	77.3	74.8	89.0	91.2	90.3	77.6
KITTI 02	44.3	68.22	68.2	58.9	72.2	65.1	79.4	61.1
KITTI 05	48.5	51.97	51.5	56.7	91.0	85.1	81.4	38.2
Lip6 Outdoor	N/A	N/A	21.2	22.5	40.0	N/A	N/A	40.9
Oxford City Centre	40.1	30.6	38.9	34.9	38.9	66.4	75.5	47.1
Oxford New College	52.6	55.9	29.3	16.8	85.9	76.7	51.0	40.0
Malaga Parking 6L	21.8	31.0	21.5	23.3	39.1	56.0	50.9	42.0

TABLE 7 Average computational times (T) on the representative datasets for the proposed pipeline and the baselines The Tracking-DOSeqSLAM achieves substantially lower timings for each evaluated case.

Method	Central processing unit CPU	Graphics processing unit GPU	Memory RAM	KITTI 00 [92] T (ms)	Oxford city centre [19] T (ms)	Malaga parking 6L [94] T (ms)
FAB-MAP [38]	Intel i7 3.4 GHz	-	16 GB	388.1	259.7	526.6
DBow2 [40]	Intel Xeon 2.6 GHz	-	N/A	111.0	27.5	42.5
SeqSLAM [21]	Intel i7 2.6 GHz	-	8 GB	118.7	25.9	81.7
DOSeqSLAM [34]	Intel i7 2.6 GHz	-	8 GB	192.6	71.4	186.2
FILD [52]	Intel Xeon 2.6 GHz	Nvidia P40	N/A	92.6	40.2	68.1
Kazmi and Mertsching [81]	Intel i7 3.4 GHz	-	16 GB	96.9	95.4	95.4
Ours	Intel i7 2.6 GHz	-	8 GB	56.2	25.4	48.9

4.4 | System's response

To analyse the computational complexity of the proposed method, we ran each framework that is, SeqSLAM, DOSeqSLAM and Tracking-DOSeqSLAM on the KITTI 00 image sequence, which is the longest among the evaluated ones exhibiting a remarkable amount of loop closures. In Table 4, an extensive assessment of the corresponding response time per image is presented. The detection and description of SURF keypoints constitutes the feature extraction process presented in the evolution methods of SeqSLAM. Feature tracking corresponds to the time needed by the KLT technique, while feature matching the time for DOSeqSLAM to segment the incoming visual stream. Image resize and patch normalisation declare the processing steps of SeqSLAM, whilst image comparison constitutes the time for the SAD method. Lastly, matching denotes the time required for each method to search for similar places in the database. The results show that we can reliably detect loops, while maintaining low execution times. We observe that every involved step is notably fast except for the comparison process which exhibits the highest execution due to the utilised metric technique. The time for keypoint detection is

negligible since we search for new SURF elements at the beginning of a new place, while the timing for point tracking is also low.

4.5 | Comparative results

This section compares the proposed pipeline with other well-known state-of-the-art algorithms. Firstly, since Tracking-DOSeqSLAM is an evolution of our previous work [34] and SeqSLAM [21] as well, we present in Table 5 the final amount of generated places and the average processing time for each method. In this regard, we aim to show that the proposed modifications result in an improvement in terms of processing time and computational complexity. As the proposed method follows the baseline approach regarding the main processing steps (e.g, image downsample, comparison technique, etc.), the computational complexity mainly depends on the number of constructed places. As highlighted in Table 5, our system achieves the generation of an amount of places at least one order of magnitude less than SeqSLAM, while a significant decrease is also presented against our previous one. This results in notably fast associations between similar submaps

permitting our method to process in less time in contrast to the other previous versions, while presenting high recall scores for perfect precision. Furthermore, the impact in terms of recall is high and outperforming its predecessors in most of the tested datasets.

In addition, for the sake of completeness, we show the results of other modern methods with the aim to help the reader to identify the place of the proposed pipeline within the state-of-the-art. In Table 6, we compare our approach with well-known works in place recognition, namely FAB-MAP [19], DBoW2 [40], BoW-SeqSLAM [26], FILD [52] and Kazmi and Mertsching [81]. The maximum recall scores for perfect precision for each approach are based on the figures reported in the original papers. The term N/A denotes that the corresponding information is not available from any cited source. Furthermore, for the case of FAB-MAP 2.0 [38] and DBoW2 [40] along with FILD [52] where no actual measurements are provided regarding the used datasets, the presented results are obtained from the setup described in [55, 81], respectively. Most of the approaches (e.g. FAB-MAP, DBoW2, BoW-SeqSLAM) are based on pretrained visual vocabularies, while FILD uses deep features in order to represent the incoming sensory measurement.

Albeit the proposed system achieves high recall rates in every tested dataset, the difficulty to present higher scores against recent loop closure pipelines which utilise more sophisticated *image processing* techniques for the location representation is evident. This is owed to the inability of SAD to quantify the obtained frames visual properties. However, our key purpose is to demonstrate the achieved performance gain, over the original SeqSLAM versions, through a refined trajectory segmentation, while operating with the lowest possible complexity and avoiding any training procedure. Thus, a direct comparison of Tracking-DOSeqSLAM with the rest of the approaches is not informative; it is only included here as a performance indicator to better interpret the possible improvement margins. On the support of thereof, in Table 7, we compare the average execution time of the proposed framework with the baselines on three representative datasets. The time for each approach is based on the reported values presented in the aforementioned sources. It is noteworthy that the proposed pipeline can achieve the lowest timings in every tested dataset.

In KITTI data sequences, the proposed algorithm performs unfavourably against the other solutions. However, despite FILD achieving the highest recall rates, this method is computationally intensive since a graphics processing unit was used to extract deep features making it unsuitable for mobile robotic platforms. Moreover, SURF are used for verifying candidate pairs though the RANSAC technique, which is well known for its high complexity and ability to reject outliers. In a similar manner, BoW-SeqSLAM and Kazmi and Mertsching exploit the epipolar geometry between the chosen images to further enhance the system's performance. When comparing the Lip6 Outdoor, the proposed pipeline exhibits over 40% of recall results outperforming the other methods. In the case of

City Centre, New College and Malaga 6L, our algorithm drops, yet it retains better recall scores than its predecessors, while keeping the lowest complexity.

5 | CONCLUSIONS

The article in hand extends our previous work [34], presenting an appearance- and sequence-based loop closure detection method, which makes use of KLT tracking in order to efficiently fragment the robot's map into submaps defining dynamic places, dubbed as Tracking-DOSeqSLAM3. Following its ancestor's image representation and similarity comparison processes, the proposed pipeline highlights the system's ability to recognise revisited places using almost two orders of magnitude less operations. This way an efficient framework for autonomous robots with restricted computational resources is achieved. When the proper place is selected, an image-to-image search in the SAD domain determines the appropriate location. The system retains its ability to perform robustly against different operational conditions and works online without any training procedure. Compared with the initial version, the proposed approach achieves high recall rates for perfect precision in the most of the tested publicly available datasets, while still retaining a real-time performance.

ORCID

Konstantinos A. Tsintotas  <https://orcid.org/0000-0002-1808-2601>

Loukas Bampis  <https://orcid.org/0000-0001-7764-4646>

Antonios Gasteratos  <https://orcid.org/0000-0002-5421-0332>

REFERENCES

1. Kostavelis, I., Gasteratos, A.: Semantic mapping for mobile robotics tasks: a survey. *Robot. Auton. Syst.* 66, 86–103 (2015)
2. Erkent, Ö., Bozma, H.I.: Bubble space and place representation in topological maps. *Int. J. Robot. Res.* 32(6), 672–689 (2013)
3. Balaska, V., Bampis, L., Gasteratos, A.: Graph-based semantic segmentation. In: *International Conference on Robotics in Alpe-Adria Danube Region, Patras*, pp. 572–579 (2018)
4. Cadena, C., et al.: Past, present, and future of simultaneous localization and mapping: towards the robust-perception age. *IEEE Trans. Robot.* 32(6), 1309–1332 (2016)
5. Stewart, B., et al.: The revisiting problem in mobile robot map building: a hierarchical Bayesian approach. In: *Proceedings of the Conference on Uncertainty in Artificial Intelligence, Acapulco* (2003)
6. Ho, K.L., Newman, P.: Loop closure detection in SLAM by combining visual and spatial appearance. *Robot. Auton. Syst.* 54(9), 740–749 (2006)
7. Garcia-Fidalgo, E., Ortiz, A.: On the use of binary feature descriptors for loop closure detection. In: *IEEE Emerging Technology and Factory Automation, Barcelona*, pp. 1–8 (2014)
8. Erhan, C., et al.: Patterns of approximated localised moments for visual loop closure detection. *IET Comp. Vis.* 11(3), 237–245 (2016)
9. Tsintotas, K.A., et al.: Appearance-based loop closure detection with scale-restrictive visual features. In: *International Conference on Computer Vision Systems, Thessaloniki*, pp. 75–87 (2019)
10. Company, Corcoles, J.P., Garcia-Fidalgo, E., Ortiz, A.: LiPo-LCD: combining lines and points for appearance-based loop closure detection,

- 31st British Machine Vision Conference, Manchester (2020). <https://www.bmvc2020-conference.com/assets/papers/0789.pdf>
11. Kostavelis, I., et al.: SPARTAN: developing a vision system for future autonomous space exploration robots. *J. Field Robot.* 31(1), 107–140 (2014)
 12. Boukas, E., Gasteratos, A.: Modeling regions of interest on orbital and rover imagery for planetary exploration missions. *Cybern. Syst.* 47(3), 180–205 (2016)
 13. Lygouras, E., et al.: Unsupervised human detection with an embedded vision system on a fully autonomous UAV for search and rescue operations. *Sensors.* 19(16), 3542 (2019)
 14. Zhao, W., et al.: Review of slam techniques for autonomous underwater vehicles. In: *International conference on robotics, intelligent control and artificial intelligence*, Shanghai, pp. 384–389 (2019)
 15. Yang, Z., et al.: Gridding place recognition for fast loop closure detection on mobile platforms. *Electron Lett.* 55(17), 931–933 (2019)
 16. Lowry, S., et al.: Visual place recognition: a survey. *IEEE Trans. Robot.* 32(1), 1–19 (2016)
 17. García-Fidalgo, E., Ortiz, A.: Vision-based topological mapping and localization methods: a survey. *Robot Auton. Syst.* 64, 1–20 (2015)
 18. Gehrig, M., et al.: Visual place recognition with probabilistic voting. In: *IEEE International Conference on Robotics and Automation*, Marina Bay Sands, pp. 3192–3199 (2017)
 19. Cummins, M., Newman, P.: FAB-MAP: probabilistic localization and mapping in the space of appearance. *Int. J. Robot Res.* 27(6), 647–665 (2008)
 20. Sünderhauf, N., et al.: On the performance of ConvNet features for place recognition. In: *Proceedings IEEE/RSJ international conference on intelligent robots and systems*, Hamburg, pp. 4297–4304 (2015)
 21. Milford, M.J., Wyeth, G.F.: SeqSLAM: visual route-based navigation for sunny summer days and stormy winter nights. In: *IEEE international conference on robotics and automation*, Saint Paul, pp. 1643–1649 (2012)
 22. Sünderhauf, N., Neubert, P., Protzel, P.: Are we there yet? Challenging SeqSLAM on a 3000 km journey across all four seasons. In: *IEEE international conference on robotics and automation of workshop on long-term autonomy*, Karlsruhe, pp. 2013 (2013)
 23. Liu, Y., Zhang, H.: Towards improving the efficiency of sequence-based SLAM. In: *IEEE international conference on robotics and automation*, Karlsruhe, pp. 1261–1266 (2013)
 24. Siam, S.M., Zhang, H.: Fast-SeqSLAM: a fast appearance based place recognition algorithm. In: *IEEE international conference on robotics and automation*, Marina Bay Sands, pp. 5702–5708 (2017)
 25. Wang, Y., et al.: Improved SeqSLAM for real-time place recognition and navigation error correction. In: *7th International conference on intelligent human-machine systems and cybernetics*, Hangzhou, pp. 260–264 (2015)
 26. Tsintotas, K.A., et al.: SeqSLAM with bag of visual words for appearance based loop closure detection. In: *International conference on robotics of the Alpe-Adria Danube Region*, Patras, pp. 580–587 (2018)
 27. Dongdong, B., et al.: CNN feature boosted SeqSLAM for real-time loop closure detection. *IET Chin. J. Electron.* 27(3), 488–499 (2018)
 28. Rodrigues, F., et al.: Three level sequence-based loop closure detection. *Robot. Auton. Syst.* 133, 103620 (2020)
 29. Garg, S., Milford, M.: Fast, compact and highly scalable visual place recognition through sequence-based matching of overloaded representations—IEEE international conference on robotics and automation, Paris, pp. 3341–3348 (2020)
 30. Tomitā, M.A., et al.: ConvSequential-SLAM: a sequence-based, training-less visual place recognition technique for changing environments [arXiv:200913454](https://arxiv.org/abs/200913454) (2020)
 31. Chancán, M., et al.: A hybrid compact neural architecture for visual place recognition. *IEEE Robot. Automat. Lett.* 5(2), 993–1000 (2020)
 32. Lowe, D.G.: Distinctive image features from scale-Invariant keypoints. *Int. J. Comp. Vis.* 60(2), 91–110 (2004)
 33. Sibley, G., et al.: Vast-scale outdoor navigation using adaptive relative bundle adjustment. *Int. J. Robot. Res.* 29(8), 958–980 (2010)
 34. Tsintotas, K.A., Bampis, L., Gasteratos, A.: DOSeqSLAM: dynamic on-line sequence based loop closure detection algorithm for SLAM. In: *IEEE international conference on imaging systems and techniques*, Krakow, pp. 1–6 (2018)
 35. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: *IEEE international joint conference on artificial intelligence*, Vancouver, pp. 674–679 (1981)
 36. Sivic, J., Zisserman, A.: Video Google: a Text retrieval approach to object matching in videos. In: *IEEE international conference on computer vision*, Nice, pp. 1470–1477 (2003)
 37. Chow, C., Liu, C.: Approximating discrete probability distributions with dependence trees. *IEEE Trans. Inf. Theory.* 14(3), 462–467 (1968)
 38. Cummins, M., Newman, P.: Appearance-only SLAM at large scale with FAB-MAP 2.0. *Int. J. Robot. Res.* 30(9), 1100–1123 (2011)
 39. Paul, R., Newman, P.: FAB-MAP3D: topological mapping with spatial and visual appearance. In: *2010 IEEE international conference on robotics and automation*, Anchorage, pp. 2649–2656 (2010)
 40. Gálvez. López, D., Tardos, J.D.: Bags of binary words for fast place recognition in image sequences. *IEEE Trans. Robot.* 28(5), 1188–1197 (2012)
 41. Mur-Artal, R., Tardós, J.D.: Fast relocalisation and loop closing in key-frame-based SLAM. In: *IEEE international conference on robotics and automation*, Hong Kong, pp. 846–853 (2014)
 42. Schlegel, D., Grisetti, G.: HBS-T: a hamming distance embedding binary search tree for feature-based visual place recognition. *IEEE Robot. Autom. Lett.* 3(4), 3741–3748 (2018)
 43. Fu, H., et al.: Deep ordinal regression network for monocular depth estimation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, Salt Lake City, pp. 2002–2011 (2018)
 44. Cao, Z., et al.: OpenPose: realtime multi-person 2D pose estimation using part affinity fields. *IEEE Trans. Pattern Anal. Mach. Intell.* 43(1), 172–186 (2019)
 45. Kansizoglou, I., Bampis, L., Gasteratos, A.: An active learning paradigm for online audio-visual emotion recognition. *IEEE Trans. Affect. Comput.* 1–1 (2019). <https://doi.org/10.1109/taffc.2019.2961089>
 46. Chen, Z., et al.: Convolutional neural network-based place recognition Australasian conference on robotics and automation, Melbourne, Vol. 2 (2014)
 47. Hou, Y., Zhang, H., Zhou, S.: Convolutional neural network-based image representation for visual loop closure detection. In: *2015 IEEE international conference on information and automation*, Lijiang, pp. 2238–2245 (2015)
 48. Sünderhauf, N., et al.: Place recognition with convnet landmarks: view-point-robust, condition-robust, training-free. *Sci. Syst. Robot.* XI, 1–10 (2015)
 49. Arandjelovic, R., et al.: NetVLAD: CNN architecture for weakly supervised place recognition. In: *IEEE conference on computer vision and pattern recognition*, Las Vegas, pp. 5297–5307 (2016)
 50. Arroyo, R., et al.: Fusion and binarization of CNN features for robust topological localization across seasons. In: *IEEE/RSJ international conference on intelligent robots and systems*, Daejeon, pp. 4656–4663 (2016)
 51. Chen, Z., et al.: Deep learning features at scale for visual place recognition. In: *2017 IEEE international conference on robotics and automation (ICRA)*, Marina Bay Sands, pp. 3223–3230 (2017)
 52. An, S., et al.: Fast and incremental loop closure detection using proximity graphs. In: *IEEE/RSJ international conference on intelligent robots and systems*, Macau, pp. 378–385 (2019)
 53. Khaliq, A., et al.: A holistic visual place recognition approach using lightweight CNNs for significant viewpoint and appearance changes. In: *IEEE transactions on robotics*, pp. 561–569 (2020)
 54. Camara, L.G., Přeucil, L.: Spatio-semantic convnet-based visual place recognition. In: *2019 European conference on mobile robots (ECMR)*, Prague, pp. 1–8 (2019)
 55. An, S., et al.: Fast and incremental loop closure detection with deep features and proximity graphs [arXiv:201011703](https://arxiv.org/abs/201011703) (2020)
 56. Camara, L.G., Gäbert, C., Přeucil, L.: Highly robust visual place recognition through spatial matching of CNN features. In: *2020 IEEE international conference on robotics and automation (ICRA)*, Paris, pp. 3748–3755 (2020)

57. Jégou, H., et al.: Aggregating local descriptors into a compact image representation. In: 2010 IEEE Computer Society conference on computer vision and pattern recognition, San Francisco, pp. 3304–3311 (2010)
58. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. *Commun. ACM.* 60(6), 84–90 (2017)
59. Russakovsky, O., et al.: Imagenet large scale visual recognition challenge. *Intl. J. Comput. Vis.* 115(3), 211–252 (2015)
60. Tolias, G., Sicre, R., Jégou, H.: Particular object retrieval with integral max-pooling of CNN activations arXiv:151105879 (2015)
61. Maffra, F., Chen, Z., Chli, M.: Tolerant place recognition combining 2D and 3D information for UAV navigation. In: IEEE international conference on robotics and automation, Brisbane, pp. 2542–2549 (2018)
62. Maffra, F., et al.: Real-time wide-baseline place recognition using depth completion. *IEEE Robot. Autom. Lett.* 4(2), 1525–1532 (2019)
63. Zaffar, M., et al.: Are state-of-the-art visual place recognition techniques any good for aerial robotics? arXiv:190407967 (2019)
64. Newman, P., Cole, D., Ho, K.: Outdoor SLAM using visual appearance and laser ranging. In: IEEE international conference on robotics and automation, Stockholm, pp. 1180–1187 (2006)
65. Bampis, L., Amanatiadis, A., Gasteratos, A.: Encoding the description of image sequences: a two-layered pipeline for loop closure detection. In: IEEE/RSJ international conference on intelligent robots and systems, Daejeon, pp. 4530–4536 (2016)
66. Bampis, L., Amanatiadis, A., Gasteratos, A.: Fast loop-closure detection using visual-word-vectors from image sequences. *Int. J. Robot. Res.* 37(1), 62–82 (2018)
67. Stumm, E.S., Mei, C., Lacroix, S.: Building location models for visual place recognition. *Int. J. Robot. Res.* 35(4), 334–356 (2016)
68. Bampis, L., Amanatiadis, A., Gasteratos, A.: High order visual words for structure-Aware and viewpoint-Invariant loop closure detection. In: IEEE/RSJ international conference on intelligent robots and systems, Canada, Vancouver, pp. 4268–4275 (2017)
69. Vysotska, O., et al.: Efficient and effective matching of image sequences under substantial appearance changes exploiting GPS priors. In: 2015 IEEE international conference on robotics and automation (ICRA), Seattle, pp. 2774–2779 (2015)
70. Vysotska, O., Stachniss, C.: Relocalization under substantial appearance changes using hashing. In: IEEE/RSJ international conference intelligent on robots and systems workshop on planning, perception and navigation for intelligent vehicles, vol. 24. Canada, Vancouver (2017)
71. Angeli, A., et al.: Fast and incremental method for loop-closure detection using bags of visual words. *IEEE Trans. Robot.* 24(5), 1027–1037 (2008)
72. Nicosevici, T., Garcia, R.: Automatic visual bag-of-words for online robot navigation and mapping. *IEEE Trans. Robot.* 28(4), 886–898 (2012)
73. Labbe, M., Michaud, F.: Appearance-based loop closure detection for online large-scale and long-term operation. *IEEE Trans. Robot.* 29(3), 734–745 (2013)
74. Lynen, S., et al.: Placeless place-recognition. In: IEEE international conference on 3D vision, Tokyo, pp. 303–310 (2014)
75. Khan, S., Wollherr, D.: IBuILD: Incremental bag of binary words for appearance based loop closure detection. In: IEEE international conference on robotics and automation, Seattle, pp. 5441–5447 (2015)
76. Garcia, Fidalgo, E., Ortiz, A.: Hierarchical place recognition for topological mapping. *IEEE Trans. Robot.* 33(5), 1061–1074 (2017)
77. Garcia et al.: iBoW-LCD: an appearance-based loop-closure detection approach using incremental bags of binary words. *IEEE Robot. Autom. Lett.* 3(4), 3051–3057 (2018)
78. Tsintotas, K.A., Bampis, L., Gasteratos, A.: Assigning visual words to places for loop closure detection. In: IEEE international conference on robotics and automation, Brisbane, pp. 1–7 (2018)
79. Tsintotas, K.A., Bampis, L., Gasteratos, A.: Probabilistic appearance-based place recognition through bag of tracked words. *IEEE Robot. Autom. Lett.* 4(2), 1737–1744 (2019)
80. Alahakoon, D., Halgamuge, S.K., Srinivasan, B.: Dynamic self-organizing maps with controlled growth for knowledge discovery. *IEEE Trans. Neural Netw.* 11(3), 601–614 (2000)
81. Kazmi, S.A.M., Mertsching, B.: Detecting the expectancy of a place using nearby context for appearance-based mapping. *IEEE Trans. Robot.* 35(6), 1352–1366 (2019)
82. Oliva, A., Torralba, A.: Building the gist of a scene: the role of global image features in recognition. *Prog. Brain Res.* 155, 23–36 (2006)
83. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE international conference on computer vision and pattern recognition, San Diego, pp. 886–893 (2005)
84. Bentley, J.L.: Multidimensional binary search trees used for associative searching. *ACM Commun.* 18(9), 509–517 (1975)
85. Rublee, E., et al.: ORB: An efficient alternative to SIFT or SURF. *IEEE Int. Conf. Comput. Vision, Barcelona*, pp. 2564–2571 (2011)
86. Zaffar, M., et al.: Cohog: a light-weight, compute-efficient, and training-free visual place recognition technique for changing environments. *IEEE Robot. Autom. Lett.* 5(2), 1835–1842 (2020)
87. Sermanet, P., et al.: Overfeat: integrated recognition, localization and detection using convolutional networks arXiv:13126229 (2013)
88. Bay, H., et al.: Speeded-up robust features'. In: European Conference on Computer Vision, Berlin, pp. 404–417 (2006)
89. Thormahlen, T., et al.: Merging of feature tracks for camera motion estimation from video (2008)
90. De.Cubber, G., Sahli, H.: Partial differential equation-based dense 3D structure and motion estimation from monocular image sequences. *IET Comp. Vis.* 6(3), 174–185 (2012)
91. Ramli, R., et al.: Local descriptor for retinal fundus image registration. *IET Comp. Vis.* 14(4), 144–153 (2020)
92. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: IEEE international conference on computer vision and pattern recognition, Providence, pp. 3354–3361 (2012)
93. Smith, M., et al.: The new college vision and laser data set. *Int. J. Robot. Res.* 28(5), 595–599 (2009)
94. Blanco, J.L., Moreno, F.A., Gonzalez, J.: A collection of outdoor robotic datasets with centimeter-accuracy ground truth. *Auton. Robots.* 27(4), 327 (2009)
95. Talbot, B., Garg, S., Milford, M.: OpenSeqSLAM2.0: an open source toolbox for visual place recognition under changing conditions. In: IEEE/RSJ international conference intelligent on robots and systems, Madrid, pp. 7758–7765 (2018)

How to cite this article: Tsintotas KA, Bampis L, Gasteratos A. Tracking-DOSeqSLAM: A dynamic sequence-based visual place recognition paradigm. *IET Comput. Vis.* 2021;15:258–273. <https://doi.org/10.1049/cvi2.12041>