IEEE Robotics and Automation Letters (RAL) paper presented at the
2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019

Probabilistic Appearance-Based Place Recognition Through Bag of Tracked Words

Konstantinos A. Tsintotas, Loukas Bampis and Antonios Gasteratos

***Abstract*— A key feature in robotics applications is to recognize whether the current environment observation corresponds to a previously visited location. Should the place be recognized by the robot, a Loop Closure Detection (LCD) has occurred. The letter in hand deploys a novel low complexity LCD method based on the representation of the route by unique Visual Features (VFs). Each of these VFs, referred to as "Tracked Word" (TW), is generated on-line through a tracking technique coupled with a guided-feature-detection mechanism and belongs to a group of successive images. During the robot's navigation, new TWs are added to the database forming a bag of tracked words. When querying the database seeking for loop closures, the new local-feature-descriptors are associated with the nearest neighboring TWs in the map casting votes to the corresponding instances. The system relies on a probabilistic method to select the most suitable loop closing pair, based on the number of votes each location polls. The proposed system depends solely on the appearance information of the scenes on the trajectory, without requiring any pre-training phase. The evaluation of the method is administered via a variety of tests with several community datasets, thus proving its capability of achieving high recall rates for perfect precision.**

I. INTRODUCTION

An appropriate representation on the environment is essential for a robot to be able to perform elaborated functions, such as path and task planning. Therefore, robotics scholars have put a tremendous effort in methods, approaches and techniques to map the world by means of several exteroceptive sensors [1], [2], [3]. In many cases, it is the scenario with which the robot would deal, that drives the map representation. Within the context of Simultaneous Localization and Mapping (SLAM) [1] while the robot navigates through the field a map of the surroundings is constructed; at the same time its position in the world is estimated. However, given the noisy sensor measurements, modeling inaccuracies and errors due to field abnormalities, even the most accurate pose estimators are prone to faults. Loop Closure Detection (LCD), i.e., the event in which a robot returns to a previously visited location and recalls it, constitutes a core component of SLAM systems and enables incremental pose drift to be rectified using visual information. The detection of numerous fault-free loop closure events constitutes a prime goal of modern autonomous systems.

Due to their low cost, camera sensors have become the key perception device in most recent robotic platforms. Place recognition approaches hinging on appearance aim to detect pre-visited locations only by means of visual sensory information [4]. As the camera stream enters the pipeline, the perceived instances are processed to produce a more compact representation. In the typical case, this process is consisted of two main procedures, viz., keypoint detection and description [5], [6], [7], [8]. When a query image (or the current robot view) is captured, comparisons are performed with all the frames in the sequence, seeking for the most suitable loop closing pairs. Towards this end, voting schemes are implemented [9], [10], [11], [12] to highlight database instances with the most common keypoint features. Other LCD pipelines [13], [14], [15], [16], [17], tackle the place recognition task based on the Bag of Words (BoW) model [18]. Quantizing the descriptors' space yields to Visual Words (VWs) and images represented by VW histograms. Thus, loop closure events are indicated by comparing similarities between such representations. The BoW approaches can offer high performance as well as computational frequency. Nevertheless, their success highly depends on the quality of the Visual Vocabulary (VV) and, in turn, the data the latter was trained with. In order to avoid a performance failure due to the generic construction of VV, incremental dictionaries which are built on-line are adopted [19], [20], [21], [22].

This paper presents a straightforward probabilistic appearance-based LCD framework which relies on an image-to-map voting scheme based on an incremental version of BoW methods avoiding any pre-trained technique. Feature tracking is performed using a guided-feature-detection technique and Kanade-Lucas-Tomasi (KLT) point tracker [23]. For each tracked feature, a Tracked Word (TW) is generated by averaging the instances of the corresponding descriptors. TWs are assigned to the map representing specific locations along the trajectory, while a Bag of Tracked Words (BoTW) is constructed during the navigation. Working with scale and rotation invariant local-features provides a built-in robustness towards view-point and velocity variations. At query time, local-feature-descriptors vote using a k-NN technique, while a binomial Probability Density Function (PDF) is adapted as a belief generator among the candidate loop closing pairs. The proposed method is evaluated on six different environments while compared against state-of-the-art methods.

The main contributions of this paper are:

- A fully probabilistic scene recognition pipeline with low computational complexity capable of detecting loop closure events through the distributed local-feature votes.
- An on-line BoTW database generation assigned to the traversed map, consisting of unique TWs.
- An experimental parameter estimation scheme based on two criteria:

Department of Production and Management Engineering, Democritus University of Thrace, 12 Vas. Sophias, GR-671 32, Xanthi, Greece
{ktsintot, lbampis, agaster}@pme.duth.gr

- The number of tracked features among consecutive frames, indicated as the parameter ν.
- The number of required tracked instances in order for a feature to be converted into a TW, pointed out as the parameter ρ.

The remainder of this work is structured as follows. In Section II, we present a review of related work on appearance-based LCD approaches. Section III describes in detail the proposed implementation. Our method's evaluation and experimental results are apposed in Section IV, while the last Section is devoted to conclusion and future work.

II. RELATED WORK

The concept of BoW has originally been applied to text retrieval [24]. In robot navigational models, such frameworks can be distinguished into two categories according to their VV construction procedure. Methods that utilize a pre-trained vocabulary are placed in the first category [13], [14], [15], [16], whereas algorithms which build their VV on the fly belong to the second one [3], [19], [20], [21], [22]. A probabilistic appearance-based pipeline, which uses a pre-trained VV of SIFT [6] descriptors is proposed in [13]. This approach additionally includes a Chow Liu tree to learn the co-occurrence probabilities among VWs [25]. Similarly, a binary VV coupled with geometrical and temporal checks can turn fault detections down [14], [15], while a binary feature-based tree is adopted in [26] to enhance the computational speed.

All these frameworks search the database for image associations in a greedy manner. On the contrary, the work presented in [16] adopted a probabilistic pipeline based on location models which are constructed along the map through features covisibility. During a query event, candidate loop closing places are retrieved from the map and evaluated through a Bayesian filter. Another set of LCD approaches with pre-trained VV utilize the visual information of image sequences. A representative example of such techniques can be found in [17], where sequences of instances are represented by a VW histogram and the candidate matches are enhanced through a quantitative interpretation of temporal consistency.

On the other hand, Angeli et al. [19] propose an incremental BoW algorithm for scene recognition. During navigation, two parallel VVs (one representing image-descriptors and the other color histograms) are generated and combined. Loop closing pairs are indicated via a Bayesian filter and validated by epipolar geometry constraints. In [3], a scalable and automatic BoW technique is presented using agglomerative clustering. Furthermore, Khan and Wollherr [20], propose a binary BoW method where VWs are incrementally generated through feature-tracking. Coupled with a likelihood function and temporal consistency checks, pre-visited locations are recognized in an on-line and real-time manner. An incremental visual dictionary built on a hierarchical structure of VWs is proposed in [22]. In [21], binary codewords with perspective invariance to the camera's motion are learned on-line from matched feature-pairs along consecutive instances. Integrated in an incremental BoW system, this technique provides reliable loop closure hypotheses.

Recent works [9], [10], [11], [12] tackle the LCD task through voting techniques directly on descriptors' space to potentially achieve more accurate results. In [9], a k-d tree is built from projected BRISK descriptors; by querying the nearest neighbor of each descriptor, loop closures are identified by means of statistical tests. Cieslewski et al. [10] search a neighborhood using a vocabulary tree to retrieve the appropriate match, while the voting score is normalized relatively to the observed landmarks. The authors in [11] propose a probabilistic approach to interpret the voting score. Loop closure events are computed by relying on the aggregated descriptor votes, while they verified by temporal and geometrical checks. Likewise, in our latest work [12] votes are distributed to places dynamically defined on the trajectory, whereas a probabilistic score indicates pre-visited locations.

Other contemporary approaches [27], [28], [29], make use of Convolutional Neural Networks (CNNs) to solve the place recognition problem. Specific convolutional layers behave as image-descriptors, whereas comparisons are performed among them. Despite their impressive results, these approaches are known for their excess demand in computational resources [30], thus remaining unsuitable for real-time applications in mobile robotics.

The majority of the aforementioned approaches in appearance-based place recognition rely on comparisons of a single or a sequence of images, represented by VW histograms. Yet, the proposed framework adapts an on-line, image-to-map voting scheme which relies on tracking features among consecutive instances, similarly to [20] and [31]. However, our method is fundamentally different due to the fact that the incrementally-constructed BoTW elements are unique. More specifically, each element is originated from a different local-feature along the trajectory, while a comparison regarding their similarity is avoided, either between local-features in the same camera measurement or the already constructed TWs. In such a way, a detailed representation of the traversed route is achieved. Following this principle, the proposed voting procedure is more robust, resulting into an accurate localization outcome.

III. METHODOLOGY

In order to carry out visual place recognition a pipeline of operations is implemented as outlined in Fig. 1. The workflow is comprised of two parts: i) Building the BoTW database, and ii) Querying the Database. Keypoint extraction, point tracking, guided-feature-detection, TW generation and BoTW are found in the first part, whereas voting scheme, probabilistic belief generator and geometrical check belong to the second one. The following sections describe the individual parts of the algorithm in detail.

A. Building the BoTW database

1) Point Tracking: Point tracking across consecutive images is achieved by a set of ν SURF Points (SP = $\{sp^1$,

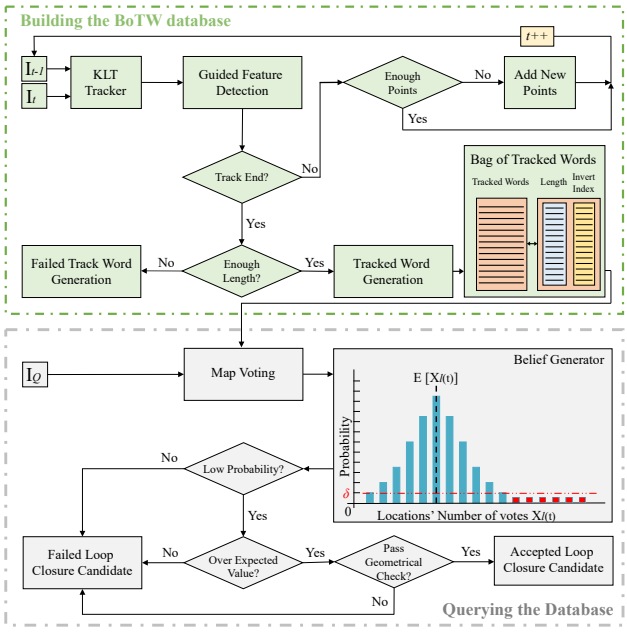


Fig. 1: An overview of the proposed pipeline. Points from the previous image I_{t-1} are tracked into the current frame I_t, by means of Kanade-Lucas-Tomasi (KLT) tracker [23]. Subsequently, tracked points are assigned to corresponding SURF [5] extracted in I_t by a guided-feature-detection mechanism. Points which lose their track during navigation are converted into tracked words and assigned to the corresponding locations along the map. When a query image I_Q arrives, its descriptors vote into the Bag of Tracked Words, according to a nearest-neighbor descriptor method. A binomial Probability Density Function (PDF), utilized as belief generator, indicates loop closing pairs. The red locations on the PDF are identified as candidate loop closure events since the probabilistic score threshold δ is satisfied. Finally, a geometrical verification step strengthens the results.

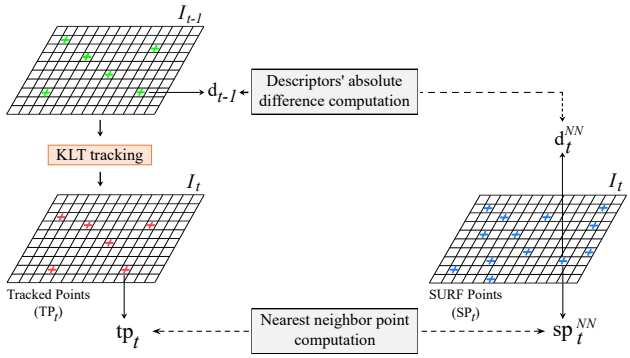


Fig. 2: Guided-Feature-Detection: Kanade-Lucas-Tomasi (KLT) tracker [23] computes the coordinates of the Tracked Points (TP = $\{\text{tp}^1, \text{tp}^2, \ldots \text{tp}^\nu\}$) from the previous I_{t-1} to the current image I_t, illustrated by green (+) and red (+) crosses, respectively. Subsequently, they are matched with their nearest neighboring SURF point ($\text{sp}_t^{NN} \in \text{SP}_t$), depicted by a blue cross (+), as per their points' coordinates and descriptors' distance.

$\text{sp}^2, \ldots \text{sp}^\nu\}$) from the previous image I_{t-1}, fed into a KLT point tracker along with the currently perceived camera measurement I_t. Additionally, we retain the corresponding set of description vectors ($\text{D}_{t-1} = \{\text{d}_{t-1}^1, \text{d}_{t-1}^2, \ldots \text{d}_{t-1}^\nu\}$) which are meant for the next step. Subsequently, the set of local keypoints SP_t in I_t is extracted and described producing the descriptors' set D_t, with a view to be matched with the ones from the tracker. Aiming to a reliable tracking system, points in I_t are browsed within 3 levels of resolution, around a 31×31 neighborhood, with a maximum bidirectional error of 3 pixels.

2) Guided Feature Detection: In order to produce unique TWs for map representation, we perform descriptor matching among different frames. The BoTW module constitutes the core component in our system, thus point descriptors need to be sufficiently accurate for a robust TW estimation and, consequently, the success of the method. A guided-feature-detection mechanism is deployed, to avoid the tendency of Tracked Points (TP = $\{\text{tp}^1, \text{tp}^2, \ldots \text{tp}^\nu\}$) to drift along the trajectory. A k-NN ($k = 1$) search is performed on the points' coordinate space between the TP_t elements and the ones extracted by SURF (SP_t). For each tracked point tp_t, the nearest sp_t^{NN} ($\text{sp}_t^{NN} \in \text{SP}_t$) is detected and evaluated by measuring the ℓ_2 distance between its descriptor d_t^{NN} and the one (d_{t-1}) corresponding to sp_{t-1} in the previous image I_{t-1} (Fig. 2). A point is accepted and its descriptor is considered to be a good match, providing that the following conditions are satisfied: (a) The Euclidean distance between tp_t and its corresponding sp_t^{NN} is lower than α:

$$\ell_2(\text{tp}_t, \text{sp}_t^{NN}) < \alpha \tag{1}$$

and (b) the descriptors' absolute difference is lower than β:

$$\ell_1(\text{d}_{t-1} - \text{d}_t) < \beta \tag{2}$$

During the guided-feature-detection, matched local-features are removed from the procedure, reducing the risk of repeatable identification between points. In the course of the robot's navigation, when a tracked feature ceases to exist (regardless of whether it forms a TW or not), it is replaced by a new one detected in I_t. Similarly, in cases where the system is unable to derive enough visual information (e.g. white plain or blurred images), the low-informative frame is skipped and the process keeps on with a new I_{t+1}.

3) Bag of Tracked Words: The final step of the BoTW database procedure is the descriptors' merging, so as to produce the TWs. When the tracking of a certain point is discontinued, its total length τ, measured in consecutive frames, determines whether a new word should be created ($\tau > \rho$). From the average of the tracked descriptors the representative TW is computed:

$$\text{TW}[i] = \frac{1}{\tau} \sum_{j=1}^{\tau} \text{d}_j[i] \tag{3}$$

where $\text{d}_j[i]$ denotes the i-th (SURF: $i \in [1, 64]$) dimension of the j-th ($j \in [1, \tau]$) descriptor-vector. In addition, two

important components are retained in the BoTW: i) the TW-length for each element, and ii) an invert indexing list for fast loop closure identification. Last, due to the nature of our system's belief generation mechanism, each TW is independent from any scoring technique, such as the "term frequency-inverse document frequency" (tf-idf) [18]. This is due to the fact that there is no straightforward approach to determine whether a specific word should be more defining than another since each TW corresponds to a different entity of the map.

B. Querying the Database

1) Map Voting: Regular LCD pipelines make use of the BoW model, in which images are compared through VW histograms. Yet, the proposed system adopts a voting scheme. At query time, the most recent instance I_Q directly projects its descriptors, formulated by guided-feature-detection, to the BoTW database via nearest neighbor search in a greedy manner. Votes are distributed into the map, whilst a database vote counter for each image increases in agreement with the contributing TWs. The vote density $x_l(t)$ of each database entry l plays a primary role in the loop closure belief generator. To avoid erroneous detections originated by the robot's varying velocity (e.g. when the platform remains still), our method seeks for database matches obtained earlier than frame I_w. In this case, w is computed by $w = t - 2c$, where c corresponds to the length of the longest active point track. This way the system is endowed with the certainty that I_t does not share any common feature with the database, while a commonly used heuristic timing threshold is avoided.

2) Probabilistic Belief Generator: To avoid the naive approach of applying a heuristic threshold over $x_l(t)$ for detecting potential loop closing candidates, a binomial PDF is adopted as a dynamic belief generator [11]. The method examines the rareness of an event and relies on the assumption that every time the robot traverses a location unseen hitherto, votes should be randomly distributed to TWs on the map, meaning that each location's vote density should be low. Ergo, the number of aggregated votes for each database entry should obey a binomial distribution (see eq. 4). Besides, when confronting a pre-visited environment the corresponding votes casted increase, which corresponds to a high vote density and represents a probabilistic event of low expectation, as depicted in Fig. 1. As a consequence, each time an increased vote score is observed, the system marks a candidate loop closure:

$$X_l(t) \sim Bin(n, p), n = N(t), p = \frac{\lambda_i}{\Lambda(t)} \tag{4}$$

where $X_l(t)$ represents the random variable for the number of accumulated votes of database location l at time t, N denotes the multitude of query's TP (number of points after the guided-feature-detection), λ is the number of TWs members in l, and Λ corresponds to the size of the BoTW list (without the rejected locations).

The binomial expected value on each location has to satisfy a loop closure threshold δ, so as to be accepted:

$$Pr(X_l(t) = x_l(t)) < \delta < 1 \tag{5}$$

while to avoid cases where a location accumulates unexpectedly few votes due to extreme dissimilarities, the following condition should hold:

$$x_l(t) > E[X_l(t)] \tag{6}$$

Through the usage of binomial PDF the system is capable of avoiding cases where the number of votes is not sufficient e.g., due to poor visual imagery information. Its robustness to identify loop closure candidates is highlighted in Fig.3.

However, since the distribution of votes affects a group of consecutive images, which should be able to satisfy the aforementioned conditions, the proposed method selects the one polling the largest number of votes (see Fig. 4).

3) Geometric Check: The aforementioned loop closing candidates are further evaluated through a geometrical check

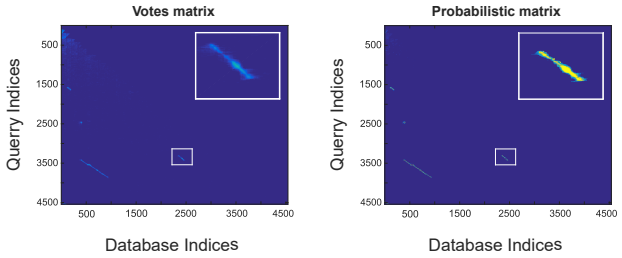


Fig. 3: Similarity matrices of KITTI 00 [32] dataset based on images' vote aggregation (left) and probability scores (right). Binomial probability shows to perform favorably, providing the system with an important intuition about pre-visited locations. For illustration purposes, each score in probabilistic matrix is plotted through -log10($Pr(X_l(t) = x_l(t))$.

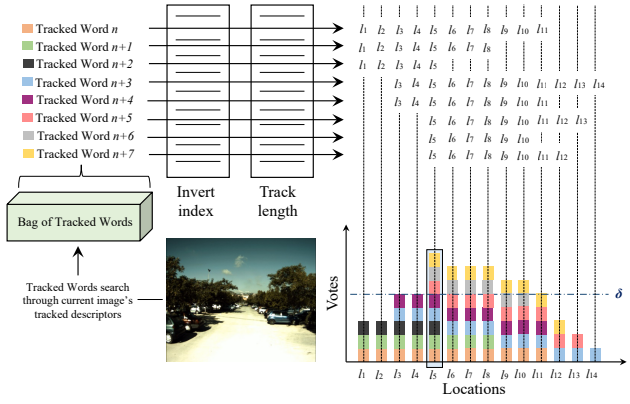


Fig. 4: Selecting a loop closing location: at query time, image-descriptors distribute votes to database locations where the nearest neighboring Tracked Words (TWs) are originated. Colored sticks indicate the corresponding votes casted by different TWs. Although locations which satisfy the probability function condition δ exist, the highlighted one is selected as the proper candidate owed to the fact that it accumulates the majority of the votes.

TABLE I: Details about used datasets

Dataset Label	Description	Image Characteristics	Sequence Size
KITTI 00 [32]	Urban environment consisting of houses, cars and trees.	1241 × 376 res., 10 Hz	4551 images apprx.11 km
KITTI 05 [32]	Urban environment consisting of houses, cars and trees.	1241 × 376 res., 10 Hz	2761 images apprx.7.5 km
Lip6 Outdoor [19]	Urban hand-held dataset surrounded by houses. Several loop closures are performed with variations in orientation and velocity.	240 ×192 res., 1 Hz	301 images apprx.1.5 km
Malaga 6L [33]	University parking containing mostly cars and trees.	1024 × 768 res., 7.5 Hz	3474 images apprx.1.2 km
EuRoC MH 05 [34]	An industrial environment of a machine hall. Several loop closures are made with variations in platform's velocity.	752 × 480 res., 20 Hz	2761 images apprx.100 m
New College [35]	College's grounds containing mostly trees, buildings and people.	512 × 384 res., 20 Hz	52480 images apprx.2.2 km

in order to ensure a robust place recognition system. Similarly to [14], the fundamental matrix is estimated by means of RANSAC, which is required to be supported by at least ϕ correspondences between the query I_Q and the matched image I_M. If the estimation fails, the selected instance is ignored, preventing the system from false detections. This straightforward but computational costly implementation relies on an exhaustive feature matching search on the chosen pair. Having said that, we exploit the extracted BoTW database to optimize this procedure. More specifically, when a new TW is added into the BoTW a list of its descriptors is stored coupled with an image index. Then, in order to obtain correspondences between I_Q and I_M, we perform comparisons only between the query's tracked features and the ones associated with the voted TWs of the matched instance. This technique accelerates the computations leading to a reduced execution time.

IV. Measuring the Performance

A. Experimental Protocol

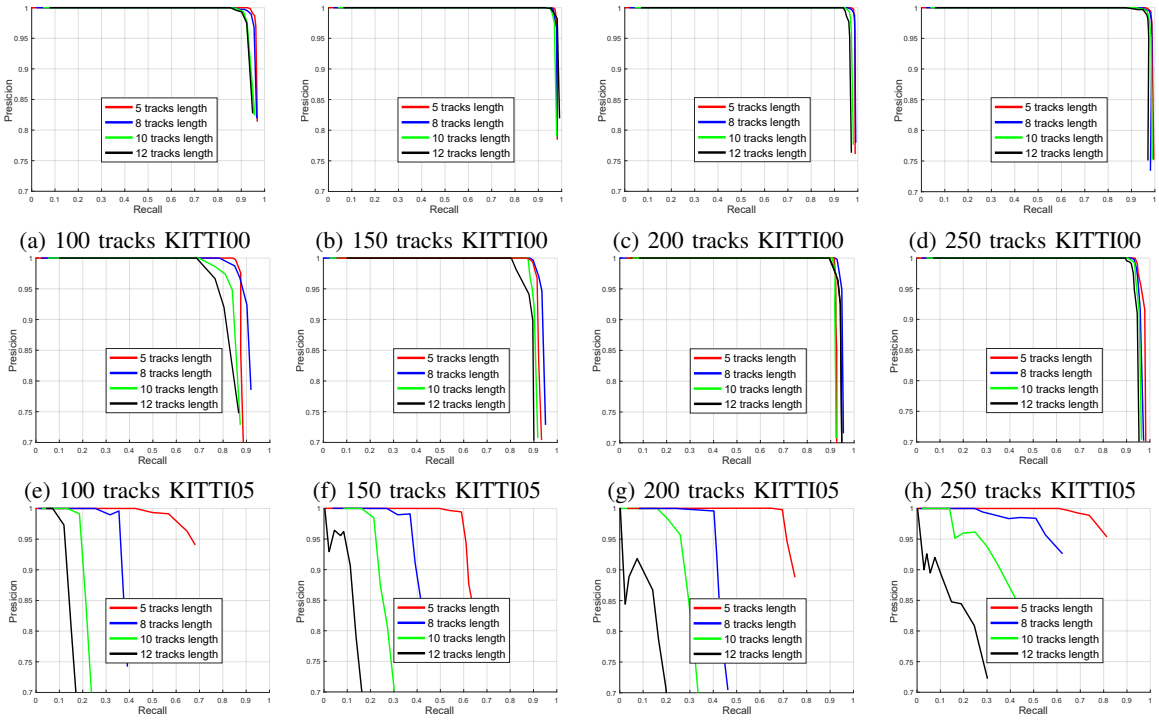
A total of six publicly-available image sequences were utilized to validate the proposed system. With the aim to adjust the parameters of the algorithm, three of these sequences were selected. The chosen environments represent outdoor and dynamic urban areas containing mostly street views. The estimated parameters are assessed on the three remaining datasets, which consist of an indoor industrial area, an outdoor university campus parking lot and a college's ground. Table I provides a summary of each sequence used. The presented approach is compared against state-of-the-art methods with incremental vocabulary, namely Angeli et al. [19], IBuILD [20], Zhang et al. [21], Gehrig et al. [11], iBoW-LCD [22], as well as our previous work [12]. For the sake of competitiveness we also compare the proposed method with approaches based on a pre-trained vocabulary, specifically FAB-MAP 2.0 [13], DBoW2 [14], DBoW2-ORB [15], PREVIeW [17]. All experiments were performed on an Intel i7-6700HQ 2.6 GHz processor with 8 GB RAM.

1) Evaluation Sequences: Two out of three image sequences belong to KITTI vision suite collection [32] while the third one to the Lip6 Outdoor sequence (L6O) [19]. Regarding KITTI datasets, the incoming visual stream is obtained by means of a stereo camera system mounted on a forward moving car. Sequences 00 (K00) and 05 (K05) have been selected since they provide meaningful loop closure examples, accurate odometry information and long-term operational conditions. Aiming to an appearance-based place recognition framework, only the left image stream was used for this evaluation. In L6O, visual information is provided by a hand-held camera encountering plenty of loop closures along the traversed route. The particular dataset is chosen for it includes both low camera resolution and frame-rate, making it essential for assessing the tracking mechanism of the proposed method.

2) Test Sequences: The EuRoC Machine Hall 05 (EuR5) sequence of the EuRoC MAV dataset [34] is selected and utilized to test the robustness of the system, as it provides strong variations in velocity along the trajectory and relevant examples of loop closure events with changes in illumination. Visual sensory information is provided by cameras mounted on a Micro Aerial Vehicle (MAV) with high acquisition frame-rate. Malaga 2009 Parking 6L (MLG) [33] and New College (NC) [35] have been recorded by means of the vision system of an electric buggy-typed vehicle and a robotic platform, respectively. They are incorporated since the characteristics of the corresponding recorded data are different by means of their evaluation datasets and contain a significant amount of loop closure examples. The incoming visual stream in every testing sequence is a stereo one, though we only utilized just the monocular sequences in our assessment. The frames of NC were resampled to 1Hz, due to the robot's low velocity and high camera frequency.

B. Performance Evaluation

To evaluate the system's performance against the selected datasets, the precision-recall metrics are displayed. Precision can be defined by the number of correct loop closing matches –true-positive detections– over the total method's identifications –true-positive plus false-positive detections–, whereas recall is the ratio of true-positive over the sum of true-positive and false-negative detections. A correct match is considered to be any identification which occurs within a small radius from the query location, while a false-positive detection lies outside this area. False-negative detections represent the locations that ought to have been recognized, but



(a) 100 tracks KITTI00 (b) 150 tracks KITTI00 (c) 200 tracks KITTI00 (d) 250 tracks KITTI00

(e) 100 tracks KITTI05 (f) 150 tracks KITTI05 (g) 200 tracks KITTI05 (h) 250 tracks KITTI05

(i) 100 tracks Lip6O (j) 150 tracks Lip6O (k) 200 tracks Lip6O (l) 250 tracks Lip6O

Fig. 5: Precision-recall curves evaluating the number of maximum tracked features against the minimum required length for a Tracked Word (TW) generation, tested on the KITTI [32] sequences, 00 (top), 05 (middle) and Lip6 Outdoor [19] (bottom) for the proposed method. As the number of Tracked Points (TP) grows, the performance is increased (recall rates for 100% precision) until it settles in the cases of 200 and 250. On the contrary, as the minimum allowed TW-length increases, the system's performance constantly decreases.

TABLE II: Method's Parameters

Number of maximum points fed into the tracker, ν	:	200
Minimum points' distance, α	:	5
Minimum descriptors' distance, β	:	0.6
Minimum track word length, ρ	:	5
RANSAC inliers, ϕ	:	9

the method failed to. The tolerance used for the evaluation is set to 10 neighboring locations and corresponds to a distance that allows the accurate estimation of fundamental matrix with RANSAC. The aim of a loop closure algorithm is to achieve the highest recall score possible for flawless precision. Ground Truth (GT) information is used to determine whether an image pair corresponds to a loop closure. GT is shaped in the form of a binary matrix whose rows and columns represent images with different time indices, while its elements are set to 1 in the case that a loop closure occurs (GT_{ij} = true) and 0 otherwise (GT_{ij} = false). For the MGL, EuR5, NC and KITTI sequences, GT was labeled manually in [12] based on the dataset odometry information. The evaluation of L6O is established through the GT data [19].

As the loop closure threshold δ is varied, we monitor the precision-recall scores obtained for different cases of maximum retained tracked features ($\nu = \{100, 150, 200, 250\}$). In addition, we assessed the TW's minimum allowed length ($\rho = \{5, 8, 10, 12\}$) for the achieved performance. In support thereof we observe that the recall rate in Fig. 5 increases

TABLE III: System's response for 4.5k images of KITTI 00 dataset (ms/frame)

		Average time	**Standard deviation**
Feature extraction	SURF detection	46.6	6.6
	SURF description	34.7	8.2
Features tracking	Kanade-Lucas-Tomasi	7.7	1.5
	Guided-Feature-Detection	4.3	5.9
Voting	Votes aggregation	96.3	62.7
	Probability score	1.8	3.7
Pairing	Image indexing	3.0	7.1
	Geometrical Check	2.0	5.6
Sum		196.4	101.3

with the number of TP, reaching more than 90% in KITTI sequences and almost 70% in L6O, while the precision remains at 100%. In L6O, where the acquisition rate is too low (1Hz), counter to the points' quantity the performance decreases as the TW-length gets longer, intensively reveling the effect of the lengthiness of TWs. This is owed to the fact that points which appear for a short time-period in the trajectory are discarded from the BoTW reducing the potential of a richer database and a more accurate voting procedure. Table II presents the parameters selected in order to achieve a reduced computational complexity, while still preserving increased recall rates.

In order to analyze the computational complexity of this work, we tested the proposed system in K00 since it is

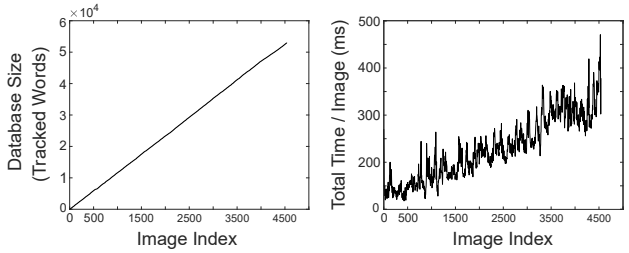


Fig. 6: Bag of tracked words evolution and computation times of the proposed system over the KITTI 00 dataset [32]. The generated tracked words along the traversed route prove the database's limited size (left). The total performance time per image with regard to the processed images depicts the evolution of computation speed (right).

the longest dataset included. In Table III, an extensive assessment of the corresponding response time per image is presented, while Fig. 6 illustrates the database evolution, as well as the system's overall performance. It is noteworthy that a total of $\sim$53K TWs are generated for a route of 11Km.

C. Comparative Results

Table IV and Table V show the precision-recall measurements for each approach as reported in the corresponding papers, while two scores are presented for the proposed pipeline. The first one indicates the performance of our system without any geometrical check, while the second one demonstrates the results generated after the impact of RANSAC. Comparisons were performed with the loop closure threshold $\delta = 2^{-11}$. This value was selected by considering the precision-recall curves from Section IV-B, with the aim to avoid redundant geometrical verification and preserve high recall rates at the same time. Nevertheless, different δ values can also be utilized to increase the achieved recall, with the cost of invoking more frequently the RANSAC-based check in order to retain perfect precision. In the comparative results presented the parameters of our method remain constant, so as to evaluate the adaptability of the approach. It is noteworthy that the proposed pipeline can achieve remarkable recall scores for perfect precision in all tested dataset.

When comparing the two KITTI datasets, the proposed pipeline exhibits over 90% of recall results. In the case of sequence 00, our method performs comparably to the rest of the approaches, reaching almost 98% in both cases (Pre-RANSAC, Post-RANSAC). In sequence 05, the proposed algorithm drops slightly, yet it retains high precision and recall scores. L6O constitutes a challenging dataset where our framework performs unfavorable against the other algorithms. This is mainly due to the selected loop closure threshold which obliges the system to deviate from its potential performance, but also to the fact that the system encounters a sequence of low textured images and frame-rate. This characteristic is highlighted in the post-RANSAC results, where the drop in recall is more intense as compared

TABLE IV: Comparisons with incremental methods

		K00	K05	L6O	MLG	EuR5	NC
iBoW-LCD [22]	P	100.0	n.a	100.0	n.a	n.a	100.0
	R	76.5		**85.2**			79.4
IBuILD [20]	P	100.0	n.a	100.0	100.0	n.a	n.a
	R	92.0		25.6	78.1		
Gehrig et al. [11]	P	100.0	100.0	n.a	n.a	100.0	n.a
	R	93.1	94.0			71.0	
Zhang et al. [21]	P	n.a	n.a	n.a	100.0	n.a	100.0
	R				82.6		59.2
Angeli et al. [19]	P	n.a	n.a	100.0	n.a	n.a	n.a
	R			71.0			
Tsintotas et al. [12]	P	100.0	100.0	n.a	100.0	100.0	100.0
	R	93.2	**94.2**		**87.9**	69.2	**88.0**
Proposed	P	99.87	100.0	100.0	97.52	92.32	97.53
	R	**97.7**	92.6	54.0	85.5	**85.9**	85.2
Proposed + RANSAC	P	100.0	100.0	100.0	100.0	100.0	100.0
	R	97.5	92.6	50.0	85.0	83.7	83.0

TABLE V: Comparisons with pre-trained methods

		K00	K05	L6O	MLG	EuR5	NC
PREVIeW [17]	P	100.0	100.0	100.0	100.0	n.a	100.0
	R	96.5	**97.3**	**58.3**	**87.5**		**92.7**
DBoW2 [14]	P	n.a	n.a	n.a	100.0	n.a	100.0
	R				74.7		55.9
DBoW2-ORB [15]	P	n.a	n.a	n.a	100.0	n.a	100.0
	R				81.5		70.3
FAB-MAP 2.0 [13]	P	100.0	n.a	n.a	100.0	n.a	100.0
	R	49.2			68.5		51.9
Proposed	P	99.87	100.0	100.0	97.52	92.32	97.53
	R	**97.7**	92.6	54.0	85.5	**85.9**	85.2
Proposed + RANSAC	P	100.0	100.0	100.0	100.0	100.0	100.0
	R	97.5	92.6	50.0	85.0	83.7	83.0

to the rest of the datasets, since some of the true positive detections are discarded as they fail to produce a valid fundamental matrix with enough inliers. Performance on the MLG and NC is pretty similar to the rest of the methods, reaching a score of 85% in both datasets, while holding high precision rates. Finally, in the case of EuR5, the system demonstrates a significant robustness against the platform's velocity variations which outperforms the other algorithms. It is also worth to note that geometrical verification does not significantly affect the system's recall rate, as per the experimental results.

Albeit the proposed system achieves high recall rates in every tested dataset, there are methods, both incremental and pre-trained ones, that outperform our framework. Regarding the incremental approaches, our previous work [12] performs better in datasets where the robot mainly follows parallel trajectory tracks when accounting pre-visited locations in which, the definition of absolute place boundaries is not crucial to the achieved accuracy. Despite the high recall rates it can achieve, this method is computational costly as compared to the proposed one. This is owed to the fact that a sequence segmentation along the trajectory is required in order to define places. Concerning the pre-trained methods, PREVIeW's performance is highly depended on its off-line learned temporal consistency filter, which enhances the validity of the system's proposed identifications. Even if a similar filter could have been adopted to our framework, we preferred not to implement it, since our main focus lays on the development of a completely on-line architecture. Concerning the computational efficiency, the size of dataset does not seem to effect our frameword to perform with high computational speed although an indexing technique is missing. An average of 196.4 ms/image proves the superiority of the proposed method in comparison with [22] where the

average time is 432.38 ms/image.

V. CONCLUSIONS

This paper proposes a novel pipeline for appearance-based place recognition. It makes use of a guided-feature-detection mechanism along with a KLT tracker to generate unique TWs, which are assigned to the navigated map. The framework is coupled with a nearest neighbor voting scheme to identify loop closing image pairs. A probabilistic score produced by a binomial PDF provides the means to recognize the proper locations, while a geometrical check determines the final decision. The presented method was designed to be computationally efficient and takes advantage of the scale- and rotation-invariance immersed in SURF descriptors. The algorithm does not require any prior training process, as it retains its robustness against different operational conditions, including changes in velocity and view-point, as demonstrated by the evaluation tests. In comparison to state-of-the-art techniques, the proposed approach achieves high recall rates while maintaining 100% precision.

VI. ACKNOWLEDGMENT

This research has been co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH – CREATE – INNOVATE (project code:T1EDK-00737). The paper was partially supported by project ETAA, DUTH Research Committee 81328.

REFERENCES

[1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age," *IEEE Trans. Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[2] I. Kostavelis and A. Gasteratos, "Semantic mapping for mobile robotics tasks: A survey," *Robotics and Autonomous Syst.*, vol. 66, pp. 86–103, 2015.

[3] T. Nicosevici and R. Garcia, "Automatic Visual Bag-of-Words for Online Robot Navigation and Mapping," *IEEE Trans. Robotics*, vol. 28, no. 4, pp. 886–898, 2012.

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

[5] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded Up Robust Features," in *Proc. European Conf. Comput. Vision*, 2006, pp. 404–417.

[6] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[7] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An Efficient Alternative to SIFT or SURF," in *Proc. IEEE Int. Conf. Comput. Vision*, 2011, pp. 2564–2571.

[8] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary Robust Invariant Scalable Keypoints," in *Proc. IEEE Int. Conf. Comput. Vision*, 2011, pp. 2548–2555.

[9] S. Lynen, M. Bosse, P. Furgale, and R. Siegwart, "Placeless place-recognition," in *Proc. IEEE Int. Conf. on 3D Vision*, 2014, pp. 303–310.

[10] T. Cieslewski, E. Stumm, A. Gawel, M. Bosse, S. Lynen, and R. Siegwart, "Point Cloud Descriptors for Place Recognition using Sparse Visual Information," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2016, pp. 4830–4836.

[11] M. Gehrig, E. Stumm, T. Hinzmann, and R. Siegwart, "Visual Place Recognition with Probabilistic Vertex Voting," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2017, pp. 3192–3199.

[12] K. A. Tsintotas, L. Bampis, and A. Gasteratos, "Assinging Visual Words to Places for Loop Closure Detection," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2018, pp. 5979–5985.

[13] M. Cummins and P. Newman, "Appearance-only SLAM at large scale with FAB-MAP 2.0," *Int. J. Robotics Research*, vol. 30, no. 9, pp. 1100–1123, 2011.

[14] D. Gálvez-López and J. D. Tardos, "Bags of Binary Words for Fast Place Recognition in Image Sequences," *IEEE Trans. Robotics*, vol. 28, no. 5, pp. 1188–1197, 2012.

[15] R. Mur-Artal and J. D. Tardós, "Fast Relocalisation and Loop Closing in Keyframe-Based SLAM," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2014, pp. 846–853.

[16] E. S. Stumm, C. Mei, and S. Lacroix, "Building Location Models for Visual Place Recognition," *Int. J. Robotics Research*, vol. 35, no. 4, pp. 334–356, 2016.

[17] L. Bampis, A. Amanatiadis, and A. Gasteratos, "Fast loop-closure detection using visual-word-vectors from image sequences," *Int. J. Robotics Research*, vol. 37, no. 1, pp. 62–82, 2018.

[18] J. Sivic and A. Zisserman, "Video Google: A Text Retrieval Approach to Object Matching in Videos," in *Proc. IEEE Int. Conf. Comput. Vision*, 2003, p. 1470.

[19] A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer, "Fast and Incremental Method for Loop-Closure Detection Using Bags of Visual Words," *IEEE Trans. Robotics*, vol. 24, no. 5, pp. 1027–1037, 2008.

[20] S. Khan and D. Wollherr, "IBuILD: Incremental Bag of Binary Words for Appearance Based Loop Closure Detection," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2015, pp. 5441–5447.

[21] G. Zhang, M. J. Lilly, and P. A. Vela, "Learning Binary Features Online from Motion Dynamics for Incremental Loop-Closure Detection and Place Recognition," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2016, pp. 765–772.

[22] E. Garcia-Fidalgo and A. Ortiz, "iBoW-LCD: An Appearance-Based Loop-Closure Detection Approach Using Incremental Bags of Binary Words," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3051–3057, 2018.

[23] C. Tomasi and T. Kanade, "Detection and Tracking of Point Features," *Carnegie Mellon Univ. Tech. Rep. CMU-CS-91-132*, 1991.

[24] R. Baeza-Yates, B. Ribeiro-Neto *et al.*, *Modern Information Retrieval*, 1999, vol. 463.

[25] C. Chow and C. Liu, "Approximating Discrete Probability Distributions with Dependence Trees," *IEEE Transactions on Information Theory*, vol. 14, no. 3, pp. 462–467, 1968.

[26] D. Schlegel and G. Grisetti, "HBST: A Hamming Distance Embedding Binary Search Tree for Feature-Based Visual Place Recognition," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3741–3748, 2018.

[27] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, 2016, pp. 5297–5307.

[28] F. Radenović, G. Tolias, and O. Chum, "CNN Image Retrieval Learns from BoW: Unsupervised Fine-Tuning with Hard Examples," in *Proc. European Conf. Comput. Vision*, 2016, pp. 3–20.

[29] N. Sünderhauf, S. Shirazi, F. Dayoub, B. Upcroft, and M. Milford, "On the Performance of ConvNet Features for Place Recognition," in *Proc. IEEE Int. Conf. Intelligent Robots and Syst.*, 2015, pp. 4297–4304.

[30] F. Maffra, Z. Chen, and M. Chli, "Viewpoint-tolerant Place Recognition combining 2D and 3D information for UAV navigation," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2018, pp. 2542–2549.

[31] M. Demir and H. I. Bozma, "Automated Place Detection Based on Coherent Segments," in *IEEE 12th Int. Conf. on Semantic Computing*, 2018, pp. 71–76.

[32] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, 2012.

[33] J.-L. Blanco, F.-A. Moreno, and J. Gonzalez, "A Collection of Outdoor Robotic Datasets with centimeter-accuracy Ground Truth," *Autonomous Robots*, vol. 27, no. 4, pp. 327–351, 2009.

[34] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The EuRoC micro aerial vehicle datasets," *Int. J. Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.

[35] M. Smith, I. Baldwin, W. Churchill, R. Paul, and P. Newman, "The New College Vision and Laser Data Set," *Int. J. Robotics Research*, vol. 28, no. 5, pp. 595–599, 2009.

[36] M. Labbe and F. Michaud, "Appearance-Based Loop Closure Detection for Online Large-Scale and Long-Term Operation," *IEEE Trans. Robotics*, vol. 29, no. 3, pp. 734–745, 2013.

[37] A. Babenko and V. Lempitsky, "The Inverted Multi-Index," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, 2012, pp. 3069–3076.