Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Modest-vocabulary loop-closure detection with incremental bag of tracked words

Konstantinos A. Tsintotas *, Loukas Bampis, Antonios Gasteratos

Laboratory of Robotics and Automation, Department of Production and Management Engineering, School of Engineering, Democritus University of Thrace, Xanthi GR-67132, Greece

ARTICLE INFO

Article history:
Received 12 December 2020
Received in revised form 25 February 2021
Accepted 7 April 2021
Available online 16 April 2021

Keywords:
Loop-closure detection
Mapping
Recognition
SLAM
Visual-based navigation

ABSTRACT

A key feature in the context of simultaneous localization and mapping is loop-closure detection, a process determining whether the current robot's environment perception coincides with previous observation. However, in long-term operations, both computational efficiency and memory requirements involved in an autonomous robot operation in uncontrolled environments, are of particular importance. The majority of approaches scale linearly with the environment's size in terms of storage and query time. The article at hand presents an efficient appearance-based loop-closure detection pipeline, which encodes the traversed trajectory by a low amount of unique visual words generated on-line through feature tracking. The incrementally constructed visual vocabulary is referred to as the "Bag of Tracked Words." A nearest-neighbor voting scheme is utilized to query the database and assign probabilistic scores to all visited locations. Exploiting the inherent temporal coherency in the loop-closure task, the produced scores are processed through a Bayesian filter to estimate the belief state about the robot's location on the map. Also, a geometrical verification step ensures consistency between image matches. Management is also applied to the resulting vocabulary to reduce its growth rate and constraint the system's computational complexity while improving its voting distinctiveness. The proposed approach's performance is experimentally evaluated on several publicly available and challenging datasets, including hand-held, car-mounted, aerial, and ground trajectories. Results demonstrate the method's adaptability, which retains high operational frequency in environments of up to 13 km and high recall rates for perfect precision, outperforming other state-of-the-art techniques. The system's effectiveness is owed to the reduced vocabulary size, which is at least one order of magnitude smaller than other contemporary approaches. An open research-oriented source code has been made publicly available, which is dubbed as "BoTW-LCD."

1. Introduction

Visual place recognition, an autonomous robot's ability to recognize a familiar place in the environment using vision as its primary sensing modality, remains a demanding and well-known challenge in the research community [1,2]. It constitutes a fundamental building block in frameworks such as Simultaneous Localization and Mapping (SLAM), where the system needs to build a map of its surroundings while at the same time identify its position within the environment [3]. Since the importance of an efficient and robust estimation is vital for achieving accurate navigation, a wide variety of methods has been proposed that aim to map the world through exteroceptive sensors [4,5]. Despite their tremendous effort, it is proven that even the most accurate pose estimators are prone to errors given the noisy sensory measurements, modeling inaccuracies, or field abnormalities. However, every time the robot returns to a previously visited location and recalls it, there is the potential to rectify the incremental pose drifts within a cost-function minimization scheme. The identification of such an event is widely known as loop-closure detection [6]. Loop-closure differs from localization since it regards the task of discriminating between an already visited location and a new one; hence, it plays a pivotal role in map rectification. On the other hand, localization considers the challenge of determining the robot's pose in an existing map. The detection of consistently accurate loop-closure events constitutes a primary goal of modern autonomous systems.

In appearance-based approaches, which consider image-to-image matching, the perception system (camera) usually attempts to construct a map of the environment based on unique and distinctive visual local features [7–13]. The detection and

* Corresponding author.
E-mail addresses: ktsintot@pme.duth.gr (K.A. Tsintotas), lbampis@pme.duth.gr (L. Bampis), agaster@pme.duth.gr (A. Gasteratos).
URLs: http://robotics.pme.duth.gr/ktsintotas (K.A. Tsintotas), http://robotics.pme.duth.gr/bampis (L. Bampis), http://robotics.pme.duth.gr/antonis (A. Gasteratos).

https://doi.org/10.1016/j.robot.2021.103782
0921-8890/© 2021 Elsevier B.V. All rights reserved.

description of the regions-of-interest can provide a compact and detailed representation of the incoming visual stream. The utilization of such key-points has been proven highly robust against image deformations (e.g., scale, rotation, and viewpoint change) or occlusions [14]. When the query camera measurement (i.e., the current robot view) is recorded, the system tries to match the observed location with the database (i.e., the currently formulated map). This is achieved by searching the most similar entry based on the visual information provided by the extracted regions-of-interest. However, nowadays, loop-closure detection algorithms have to provide robust navigation for an extended period. Hence, the computational efficiency and the storage requirements are vital factors for recognizing previously visited areas during long-term and large-scale SLAM operations. As most pipelines scale linearly with the environment's size, constituting a limitation for resource-constrained platforms, such as Unmanned Aerial Vehicles (UAVs), especially in the case that raw local descriptors are used for image representation. Therefore, there has been great interest in developing compact appearance representations that demonstrate sub-linear scaling in computation time and storage requirements.

The Bag of Words (BoW) model, applied initially to text retrieval [15], is established as a primary tool to tackle the place recognition task [16–25]. Quantizing a batch of key-point descriptors through a training procedure, typically via k-means clustering [26], yields the vocabulary's visual words [27]. A fixed-size vocabulary is produced, which is then used as a vector quantizer to categorize extracted descriptors of both query and database images. Loop-closure events are indicated by comparing similarities between such histogram representations. BoW approaches offer high performance, as well as computational efficiency. Nevertheless, their success is highly dependent on the quality of the visual vocabulary and, in turn, the sufficiency of the available training data. Even if a generic learning set is available, with an abundance of different visual cues, false detections can still arise when applied to environments with fine visual characteristics which are not distinguishable due to the finite number of present visual words. In those cases, the vocabulary needs to be trained from scratch using images captured in the same environment. Various available techniques rely on the formulation of a purely incremental visual vocabulary, which is computed each time the robot attempts a new mission to obviate such dependencies and cope with these issues [28–37]. Such pipelines utilize voting schemes to highlight pre-visited locations by identifying map entries with the most common correspondences owing to their database construction nature. Such algorithms typically reduce the storage footprint, and consequently, memory consumption, in the expense of computational complexity to achieve high performance. As a final note, contemporary appeared methods [38–40] use Convolutional Neural Networks (CNN), which are initially trained for object recognition, to address the visual place recognition task; thus, tightly bound to their learning examples' attributes [41]. Specific network layers are treated as image descriptors, whereas matches are detected based on their distance metric. Despite their high performances, these approaches are known for their excess demand in computational resources [42], while their feature extractor is viewpoint-dependent since the spatial information is excluded from global descriptors. As a result, they remain incompatible with most real-time SLAM applications over resource-constrained platforms (including battery-powered aerial, micro-aerial, and ground vehicles), as indicated by [43].

Our interest lies in developing a low-complexity and probabilistic appearance-based loop-closure detection framework that identifies previously seen locations thanks to an incremental BoW method based on feature tracking, viz., the Bag of Tracked Words (BoTW), thus avoiding any pre-training procedure. The well-known Kanade–Lucas–Tomasi (KLT) point tracker [44] is utilized to formulate such a vocabulary, accompanied by a guided feature selection technique. Each point whose track ceases to exist is transformed into a tracked word used to describe every key-point element contributing to its formulation. The query's tracked descriptors seek for the Nearest-Neighboring (NN) words into the vocabulary to detect loop-closures, distributing votes across the traversed path. Each voted location is assigned with a similarity score through a binomial Probability Density Function (PDF), which is meant to indicate candidate matches.

We further introduce modeling mechanisms that significantly improve our framework's memory usage and computational complexity compared to other modern solutions. Our approach's implementation blocks are discussed in detail to justify its effectiveness better and enhance its reproducibility. The unchecked generation of new elements in incremental vocabulary methods affects the systems' performance since these new entries reduce their distinctive ability, especially in cases where the robot traverses pre-visited locations. Our method applies a map management scheme that restricts similar words during the vocabulary construction to address such a deficiency.

Moreover, loop-closure detection is a task submitting to a temporal order of the visited places along the navigation route. If a location is identified as pre-visited, then it is highly probable that the following ones have also gone through. This property is explored in the proposed approach by employing a Bayes filter, accompanied by a temporal consistency constraint, over the probabilistic scores produced through the binomial PDF. We specifically exploit the temporal information of the incoming visual stream to decide about the appropriate belief state. This way, an improvement in recall rate is achieved since locations within a known area are not excluded even if they present a lower similarity than the defined threshold. Lastly, a geometrical verification step is performed over the most similar candidates. The proposed method is tested on nine different environments in a broad set of conditions and compared against various state-of-the-art methods (both incremental and pre-trained).

As a final note, intending to serve as a benchmark for the research community, an open-source implementation of the presented work is available, under the title "Bag of Tracked Words-mapping for Loop-Closure Detection" (BoTW-LCD).[1] From the user's perspective, the framework consists of two major parts: (1) the vocabulary-building block that takes raw visual sensory data and maps the environment and (2) the query procedure, where the system searches for loop-closure events.

The remainder of this work is organized as follows. Section 2 discusses the related studies in the field of appearance-based loop-closure detection. In Section 3, the proposed vocabulary construction is described in detail, while the recognition process is clarified in Section 4. Section 5 presents the method's experimental evaluation and comparative results against other state-of-the-art approaches. Conclusions and possible extensions to our approach are outlined in Section 6.

2. Related work

In this section, a focused discussion on the most representative techniques in the field of appearance-based place recognition is presented. According to the vocabulary generation process, the literature is distinguished into two main categories: off-line and on-line, to guide the reader in placing the proposed method within the state-of-the-art.

[1] The reader can download and review an implementation of BoTW-LCD at: https://github.com/ktsintotas/BoTW-LCD. Following an extensible and modular design, the algorithm's components are organized in Matlab functions.

2.1. Off-line approaches

Comparisons between local features inevitably increase any system's computational cost, particularly for mobile robots, where the incoming visual stream captures highly textured environments [45]. As a result, most of the literature addresses the challenge of appearance-based loop-closure detection by adopting the BoW model. This is owed to its proven effectiveness in computational speed, primarily when combined with an inverted indexing file system [46]. Fast Appearance-Based MAPping (FAB-MAP), a standard for loop detection, tackles the problem by utilizing a vocabulary generated by Scale-Invariant Feature Transform (SIFT) descriptors [47]. Also, a Chow Liu tree learns the co-occurrence probabilities among visual words [48]. Continuous Appearance-based Trajectory Simultaneous Localization and Mapping (CAT-SLAM) extends the FAB-MAP by utilizing odometry information [49], while the method proposed by [50] adopts an improved scoring technique for BoW approaches known as Term Frequency - Inverse Document Frequency (TF-IDF). Aiming to discriminate the vocabulary's entries, description vectors for each location are created, where each element is proportional to the ratio between the number of word occurrences within that location and the total of words in the entire learned bag. Comparisons between images are made by finding their TF-IDF vectors' distance. An improved approximation was introduced later [16], allowing the system to scale by more than two orders of magnitude [18]. By grouping landmarks through the local features covisibility during navigation, location graph-models are created in [21]. During a query event, the graph is browsed for clusters of landmarks that share substantial similarity with the query, while evaluation is performed through a Bayes filter. This method exhibits an inherent ability to cope with variations in the robot's trajectory, including irregular changes in speed, direction, and viewpoint. [19] presented a method based on a binary vocabulary, successfully constructed through k-means++ clustering [51]. The system's false detections are significantly reduced due to geometrical and temporal checks. Similarly, this approach's extension utilizes rotation invariant and scale-aware local features on a keyframe-based SLAM system [20].

While the methods mentioned above address the loop-closure detection task as a single image matching process, recent frameworks introduced the concept of sequence-to-sequence matching [52–55]. These techniques take advantage of the additional information provided by a group of images in a scene, treating each sequence as an aggregation of image description vectors or visual words. In [22], the incoming visual sensory information is segmented into fixed-size groups of images and represented by a common visual-word-histogram. Using a quantitative interpretation of temporal consistency, sequence-to-sequence matches that are coherently advancing along time are enhanced [23]. When employing the BoW model in SeqSLAM [56], one of the most acknowledged methods in the field of sequence-based place recognition, a robust system against scale and rotation variations is provided [57].

2.2. On-line approaches

Off-line vocabulary building methods typically use clustering algorithms, which require various heuristic parameters (e.g., the number of clusters in the vocabulary or some sort of distance threshold to define different sample groups). Finding adequate parameters for an optimum vocabulary is a tedious task that generally involves multiple trial-and-error iterations. For instance, a vocabulary with too many elements would not satisfy abstraction properties to measure similarities between images correctly, whist a vocabulary with not enough words would inevitably merge visual information from different entities due to the wide quantization intervals. Incrementally building a visual vocabulary entails "learning" the environment in which the agent acts. This concept was introduced by [28], who described a place recognition system, the vocabulary of which was generated on-line to address the localization task. In an extension of this work, two visual vocabularies (one representing image descriptors and another for color histograms) were created incrementally to detect loop-closures in a Bayesian filtering scheme [29]. In this method, the candidate matches were validated when the epipolar geometry constraint was satisfied. An on-line vocabulary was also proposed by [30], which utilized a sliding window over the image stream to match between SIFT features. [31] followed the incremental fashion, and proposed a visual vocabulary whose words were generated using a modified version of agglomerative clustering. A loop-closure detection approach for large-scale and long-term autonomy, entitled Real-Time Appearance-Based Mapping (RTAB-Map), was proposed by [32], with direct application in SLAM systems. Since mobile robots have limited computing resources, this solution's main contribution was related to a memory management mechanism that constrained the map's size, allowing the detections to be established under a fixed time limit. In the IBuILD algorithm [33], Binary Robust Invariant Scalable Key-points (BRISK) were matched across consecutive images yielding a binary vocabulary. A likelihood function is maximized based on the visual words' occurrence frequency in images during query time, while inconsistent loop-closure hypotheses were filtered out through a temporal consistency check. Similarly, binary codewords were learned from adjacent images via linear discriminant analysis [34]. Integrated into the incremental BoW pipeline of IBuILD, this technique provided reliable loop hypotheses. In the Hierarchical Topological Mapping (HTMap) approach of [58], images with similar visual properties were stored in groups according to places, formulating a hierarchical architecture. Each place was represented by a global descriptor, which summarized the visual information of the traversed locations. Firstly, the algorithm selected the candidate loop-closing place by comparing the query's global descriptor with the incrementally created map. The most likely match was retrieved through an extensive search in the local features space, followed by a geometric consistency test. The same authors recently introduced a new approach in which dynamic islands were used to group the images based on Spatio-temporal similarity [36]. For efficient indexing, the local features' binary description space in [12] was successively clustered in the form of a tree. Likewise, in our previous work, a dynamic sequence segmentation was performed based on the images' content proximity resulting in the places' formulation [35]. The accumulated descriptors were processed into the growing neural gas clustering mechanism for the corresponding visual words' generation [59]. A voting scheme was adopted to highlight the most similar database instances, while a probabilistic score indicated the candidate matching place [60]. The most appropriate detection was selected from an image-to-image search through temporal and geometrical checks, providing a higher level of discrimination. Places' representation was incrementally learned in [61] using a modified version of growing self-organizing maps [62], along with gist features. During the query, the maximally active neuron was searched and retrieved as a loop-closure candidate. The Fast and Incremental Loop-closure Detection (FILD) method presented an incremental graph-based CNN feature vocabulary [63]. The method was proposed for a SLAM architecture, and local key-points are also extracted to support the system, while the graphics processing unit was utilized to cope with the computationally demanding deployment of the CNN.

3. Bag of tracked words

3.1. Overview

Unlike most of the approaches mentioned above, the proposed one does not require any training process or environment-specific parameter tuning since the map is built on-line in the course of the robot's navigation. As the construction of the vocabulary plays the primary role in the proposed visual loop-closure detection pipeline, it has to be as discriminable and detailed as possible. Our trajectory mapping is based on the observation that the traversed path is associated with unique visual words generated on-line. On the contrary, through the BoTW scheme, each codeword is initiated by a local key-point tracked along the trajectory in consecutive camera frames. An algorithm with scale- and rotation-invariant properties has been adopted to obtain a robust and accurate description against image deformations. Overall, the map representation during the robot's navigation consists of four individual parts: (i) feature tracking, (ii) guided feature selection, (iii) tracked word generation, and (iv) merging words.

3.2. Feature tracking

Feature tracking is essential for several high-level computer vision tasks such as motion estimation, structure from motion, and image registration. Since the earliest works, feature trackers have been used as a standard tool for handling feature points in a sequence of images. We have chosen to map the trajectory, through a tracker based on the Speeded-Up Robust Features (SURF) [9]. Each SURF element has a detection response that quantifies its distinctiveness among the rest of the image's content. This property is used to select the most prominent local key-points in the image. Thus, intending to promote computational efficiency, we limit the number of features to be used to the ν most prominent. Those key-points ($\mathrm{P}_{t-1} = \{\mathrm{p}^1_{t-1}, \mathrm{p}^2_{t-1}, \dots, \mathrm{p}^{\nu}_{t-1}\}$) from the previous image I_{t-1}, along with the current camera frame I_t, are utilized within a KLT point tracker, to obtain their projected location, which we refer to as Tracked Points ($\mathrm{TP}_t = \{\mathrm{tp}^1_t, \mathrm{tp}^2_t, \dots, \mathrm{tp}^{\nu}_t\}$). Additionally, we retain the corresponding set of description vectors ($\mathrm{D}_{t-1} = \{\mathrm{d}^1_{t-1}, \mathrm{d}^2_{t-1}, \dots, \mathrm{d}^{\nu}_{t-1}\}$) that are meant to be matched with the corresponding ones (D_t) in I_t.

3.3. Guided feature selection

Although KLT is sufficiently effective in estimating a detected point's flow between successive frames (e.g., I_{t-1} and I_t), accumulative errors within the entire image-sequence may drift the Tracked Points. Furthermore, as the algorithm progresses over time, points can be lost due to lighting variations, out-of-plane rotations, or articulated motions. Points have to be periodically redetermined to track features over a long period. Having apprehended these challenges, we adopt a guided feature selection technique [64–67] (Fig. 1) that, along with the KLT's flow estimation, also detects new SURF key-points ($\mathrm{P}_t = \{\mathrm{p}^1_t, \mathrm{p}^2_t, \dots, \mathrm{p}^{\mu}_t\}$) and computes the corresponding description vectors ($\mathrm{D}_t = \{\mathrm{d}^1_t, \mathrm{d}^2_t, \dots, \mathrm{d}^{\mu}_t\}$) from the most recent frame I_t. Note that we retain only the μ most prominent detected feature points with a response higher than Φ, to reduce computational complexity further. A NN search is performed between the Tracked Points' coordinate space (TP_t) detected in image I_t and the ones in P_t. Thus, for each tracked point tp^i_t, the nearest $\mathrm{p}^{NN}_t \in \mathrm{P}_t$ is accepted as a proper extension-member of the track, providing that the following conditions are satisfied:

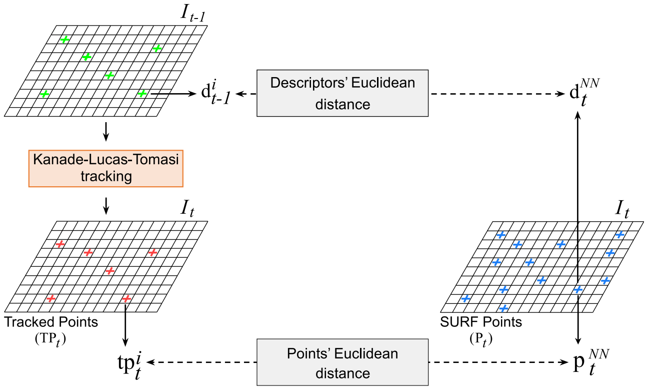

I_{t-1}
d^i_{t-1}
Descriptors' Euclidean distance
d^{NN}_t
Kanade-Lucas-Tomasi tracking
I_t
I_t
Tracked Points (TP_t)
SURF Points (P_t)
tp^i_t
Points' Euclidean distance
p^{NN}_t

Fig. 1. Guided feature selection over the points being tracked. The Kanade–Lucas–Tomasi [44] tracker estimates the expected coordinates for each of the Tracked Points ($\mathrm{TP}_t = \{\mathrm{tp}^1_t, \mathrm{tp}^2_t, \dots, \mathrm{tp}^{\nu}_t\}$), originated from the previous image I_{t-1}, to the current one I_t, (the green and red crosses (+), respectively). Their nearest-neighboring points $\mathrm{p}^{NN}_t \in \mathrm{P}_t$, detected via speeded-up robust features [9], are evaluated as per their points' coordinates and descriptors distance for the proper feature selection using Eqs. (1) and (2). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

– the Euclidean distance between tp^i_t and its corresponding p^{NN}_t is lower than α:

$$\ell_2(\mathrm{tp}^i_t, \mathrm{p}^{NN}_t) < \alpha, \tag{1}$$

– the Euclidean distance between its descriptor d^{NN}_t and the d^i_{t-1}, corresponding to p^i_{t-1} in the previous image I_{t-1}, is lower than β:

$$\ell_2(\mathrm{d}^i_{t-1}, \mathrm{d}^{NN}_t) < \beta. \tag{2}$$

If at least one of the above conditions is not met, the corresponding track point ceases to exist, and it is replaced by a new one detected in I_t, ensuring a constant number of ν TP_t members. Similarly, aiming to preserve a constant set of points during the robot's navigation, when a tracked feature is discontinued (regardless of whether it forms a tracked word or not), it is replaced by a new one, fished out from I_t. This way, the computationally costly brute force local features' matching as tracking scheme is avoided, while a robust trajectory mapping is achieved.

3.4. Tracked word generation

The next step of the BoTW procedure is the descriptors' merging, which, in turn leads to the formulation of the visual codewords. When the tracking of a certain point terminates, its total length τ, measured in consecutive frames, determines whether a new word should be created ($\tau > \rho$). Describing part of the environment, the representative tracked word is computed as the median of the tracked descriptors:

$$\widetilde{\mathrm{TW}}[i] = \mathrm{median}(\mathrm{d}_1[i], \mathrm{d}_2[i], \dots, \mathrm{d}_j[i]), \tag{3}$$

where $\mathrm{d}_j[i]$ denotes the element in the ith (SURF: $i \in [1, 64]$) dimension of the jth ($j \in [1, \tau]$) description vector. Note that, we refer to the tracked word set as a visual vocabulary since each codeword is created through an average representation, as it evident in Eq. (3) of [68], which is also the norm for a typical BoW representation. In general, new codewords are generated through averaging the corresponding descriptors, yet in the proposed approach, the median is selected since it provides better performance with lower computational cost as evidenced by the experimental evaluation in Section 5. Finally, an indexing list *Idx* is retained that includes the positions upon which each tracked word is located in the respective image.

3.5. Merging words

Finally, to provide a discriminative visual vocabulary, we avoid adding new visual elements into the vocabulary without comparing their similarity to the database. In the proposed system, an additional preliminary step is incorporated. For each newly generated element, a one-vs-all scheme computes the pairwise distances against the database's ones. Subsequently, the *Nearest-Neighbor Distance Ratio* [7] is applied, indicating two visual elements as similar when a distance ratio value lower than 0.5 is satisfied. The tracked descriptors of the newly created element and the vocabulary's chosen word are merged based on Eq. (3), and the new codeword is ignored. However, in Section 4.6, we further propose a vocabulary management scheme in which visual words corresponding to already visited locations are discarded, resulting in an overall reduced memory footprint.

4. Probabilistic loop-closure detection pipeline

In this section, our probabilistic framework for the identification of loops within BoTW-LCD is presented. The voting procedure is being described as the first step of the proposed on-line pipeline. Subsequently, we show how the locations are assigned with a probabilistic score through the binomial PDF, while the derivation of the Bayes filtering scheme used for the estimation of the loop-closure state is also detailed. Finally, we focus on the additional implementation details we adopted for incorporating geometrical verification and visual vocabulary management.

4.1. Searching the database

With the aim to perform reliable searching during query, the newly acquired frame I_Q should not share any common features with recently visited locations. This is owed to the fact that a set of input images obtained within a short time interval before grabbing I_Q are expected to be similar to it, yet they should not be considered as loop-closure events. To prevent our pipeline from detecting such cases, we consider a temporal window w, which rejects locations visited just earlier $(I_{Q-1}, I_{Q-2}, \ldots, I_{Q-w})$. We define this window as $w = t - 4c$, where c corresponds to the length of the longest active point track, as indicated by the retained τ values. In this way, it is guaranteed that I_Q will not share any visual information with the recently created database entries, while at the same time, we avoid the use of a fixed timing threshold that is typically selected by environment-specific experimentation.

Due to the lack of description histograms, the proposed appearance-based framework adopts a probabilistic voting scheme to infer pre-visited locations. At query time, the most recent incoming sensory data I_Q directly distributes its descriptors – formulated by guided feature selection – to the database via a k-NN ($k = 1$) search among the available database tracked words in a brute force manner. In order to accelerate the matching process, many approaches build a k-d tree [69]. While offering an increased computational performance when applied to a low dimensional descriptor space, the tree is unsuitable for on-line developed vocabularies. This is owed to possible unbalanced branches and the addition of new descriptors after the tree construction, impairing the performance [70]. Moreover, during on-line navigation, the complexity concerning the tree building will eventually prevent real-time processing, especially in large-scale environments containing thousands of images [71]. Besides, our descriptor has a 64-dimensional feature vector, and the k-d tree is unable to provide speedup over the exhaustive search for more than about 10-dimensional spaces [7]. A valid alternative for high dimensional descriptors, as well as for larger vocabularies, is the inverted multi-index file system [72]. This technique is multiple times faster compared to a k-d tree while offering similar performance. However, it needs to be trained beforehand, impractical for incremental approaches within a SLAM framework. Aiming to improve the descriptor matching speed, an incremental feature-based tree is proposed by [73], which is still incompatible with our framework due to its boolean structure. Even though the availability of indexing approaches, our work (see Section 4.6) aims to map the environment efficiently. Therefore, we focus on the significant reduction of the vocabulary's size, as well as the rate of its increment, reaching a footprint of one order of magnitude shorter than other state-of-the-art techniques. As our results in Section 5.5 suggest, using such a small vocabulary renders the complexity of an exhaustive search inferior to the overhead of retaining a dynamic indexing file system.

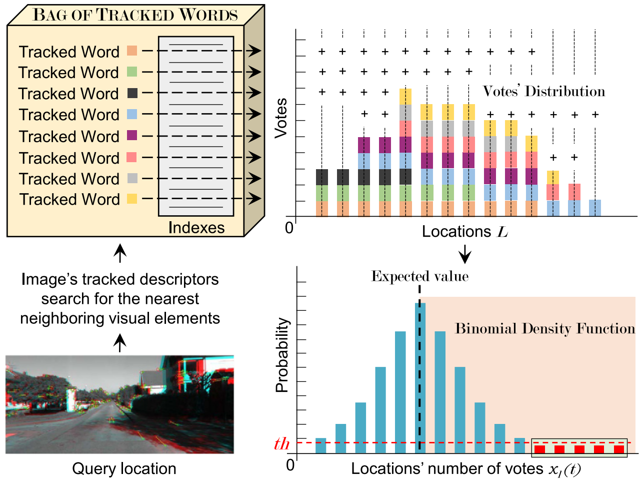


Fig. 2. Probabilistic appearance-based loop-closure detection. During a query event, the most recently obtained image directly distributes its descriptors, formulated by guided feature selection, to the Bag of Tracked Words list via a greedy nearest-neighbor search. Votes are assigned to the map L, whilst a vote counter for each location $l \in L$ increases relatively to the contributing words (colored cubes). Finally, candidate locations are indicated via a binomial density function according to their vote density $x_l(t)$. Highlighted with red, instances of votes' count correspond to locations that are intended for a geometrical check since they satisfy the rareness limit th of a loop-closure, while also exceeding the expected vote aggregation value. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.2. Navigation using probabilistic scoring

During the matching process among the query features from I_Q and the vocabulary, votes are distributed into the map L under the tracked words' indexing list Idx, as depicted in Fig. 2. A database vote counter $x_l(t)$ for each traversed location $l \in [1, t - 4c]$ increases in agreement with the associated words. In order to decide whether a location corresponds to a possible loop-closure event, most voting schemes use a threshold over the accumulated votes. This is a straightforward method where a single threshold value is selected through extensive experimentation. Although such techniques could also be applicable here, it is uncertain how the system would behave in cases where the number of votes is not sufficient (e.g., due to low textured visual information). To avoid the aforementioned simplified approach, a binomial PDF is adopted to assign a score over each location based on the votes' density:

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

where $X_l(t)$ represents the random variable regarding the number of accumulated votes of each database location l at time t, N denotes the multitude of query's tracked words (the cardinality of TP_Q after the guided feature selection), λ is the number of visual elements included in l (the cardinality of TW_l) and $\Lambda(t)$ corresponds to the size of the generated BoTW list until t (excluding the locations inside the window w). The nature of the binomial function seeks for the rareness of an event. In cases where the robot traverses a hitherto unseen location, votes should be randomly distributed to their NN words in the database even if they are not accurately associated to an actually similar one. This constitutes a common event with high probability, meaning that the locations' vote density should be low. Ergo, the number of aggregated votes for each database entry should obey a binomial distribution (see Eq. (4)). Contrariwise, when confronting a pre-visited environment, the corresponding votes casted for a specific location increase. Thus, the random vote distribution expected from the binomial function would be violated. As a consequence, the event would be considered of low probability with an increased voting score. The binomial expected value of a location l has to satisfy a loop-closure threshold th, so as to be accepted:

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

where $x_l(t)$ corresponds to the respective location's aggregated votes. However, to avoid cases where a location accumulates unexpectedly few votes due to extreme dissimilarities, the following condition should also hold:

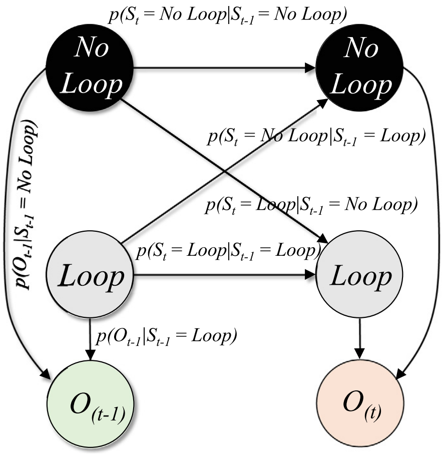
$$x_l(t) > E[X_l(t)]. \tag{6}$$

Conditions (5) and (6) of binomial PDF are depicted in Fig. 2 through the light green and light orange areas, respectively. The robustness of this metric to highlight loop-closure events is demonstrated in our previous works [35,68]. In addition, with the aim to avoid the redundant computation of probabilistic scores for each traversed location (e.g., for completely unvoted entries), we propose to compute the binomial-based score only for locations gathering more than 1% of the votes distributed by the tracked descriptors.

4.3. Location estimation via recursive Bayes rule

In a previous work of ours [35], a location is accepted as a loop-closure detection when the system meets specific conditions for a certain sequence of consecutive measurements. However this technique presents the disadvantage that many loop hypothesis belonging at the starting point of a pre-visited area are ignored until the temporal check is satisfied. With a view to tackle this drawback, we take advantage of the temporally consistent acquisition of images within the loop-closure task and adopt a Bayesian scheme. Even though this approach can be considered to be a standard practice in the field [16,29,32,58,61,74] our system differs in the aspect that we chose to apply a simple temporal model which maintains the decision factor between consecutive observations, rather than to compute a probability score for each database entry. The discrete Bayes filter allows us to deal with noisy measurements and ensures temporal coherency between consecutive predictions, integrating past estimations over time. Despite the presence of the Bayesian filter, locations captured in a sequence of loop-closing images are processed for further evaluation without being affected by their binomial-based score.

A proper filtering algorithm needs to maintain only the past state's estimates and updating them, rather than going back over the entire history of observations for each update. In other words, given the filtering result up to time $t-1$, the agent needs to compute the posterior (filtering) distribution $p(S_t \mid O_t)$ for t using the new observation O_t. Let $S_t = \langle No\ Loop, Loop\rangle$ be the state variable representing the event that I_t closes a loop, while O_t is the binomial response $Pr(X_l(t) = x_l(t))$ between I_Q and the database. Following the Bayes' rule and under the Markov assumption, the posterior can be decomposed into:



p(S_t = No Loop|S_{t-1} = No Loop)
No Loop
No Loop
p(S_t = No Loop|S_{t-1} = Loop)
p(S_t = Loop|S_{t-1} = No Loop)
p(O_{t-1}|S_{t-1} = No Loop)
Loop
p(S_t = Loop|S_{t-1} = Loop)
Loop
p(O_{t-1}|S_{t-1} = Loop)
O_(t-1)
O_(t)

Fig. 3. State machine representation of the proposed Hidden Markov Model (HMM) for loop-closure detection. Observations (O_{t-1}, O_t) are based on the system's binomial response $Pr(X_l(t) = x_l(t))$ among the database locations after the voting process. The light green observation indicates the existence of locations l which satisfy the binomial function's conditions ($\exists\, l \in L : O_t < th$), while the light orange examples correspond to the ones that do not ($\forall\, l \in L : O_t > th$). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$p(S_t|O_t) = \eta\, \underbrace{p(O_t|S_t)}_{\text{Observation}} \underbrace{\sum_{S_{t-1}} \underbrace{p(S_t|S_{t-1})}_{\text{Transition}} p(S_{t-1}|O_{t-1})}_{\text{Belief}}, \tag{7}$$

where η is a normalization constant. The recursive estimation is being composed by two parts: firstly, the current state distribution is projected forward (prediction) from $t-1$ to t; then, it is updated using the new evidence O_t.

4.3.1. Prediction

Between $t-1$ and t, the posterior is updated according to the robot's motion through the transition model $p(S_t|S_{t-1})$, which is used to predict the distribution of S_t given each state of S_{t-1}. The combination of the above with the recursive part of the filter $p(S_{t-1}|O_{t-1})$ comprises the belief of the next event. Depending on the respective values of S_t and S_{t-1}, this probability is set with one of the following values, which are further discussed in Section 5.3:

- $p(S_t = No\ Loop \mid S_{t-1} = No\ Loop) = 0.975$, the probability that no loop-closure event occurs at time t is high, given that no loop-closure occurred at time $t-1$.
- $p(S_t = Loop \mid S_{t-1} = No\ Loop) = 0.025$, the probability of a loop-closure event at time t is low, given that no loop-closure occurred at $t-1$.
- $p(S_t = No\ Loop \mid S_{t-1} = Loop) = 0.025$, the probability of the event "*No Loop*" at time t is low, given that a loop-closure occurred at time $t-1$.
- $p(S_t = Loop \mid S_{t-1} = Loop) = 0.975$, the probability that a loop-closure event occurs at time t is high, given that a loop also occurred at time $t-1$.

4.3.2. Bayes update

The sensor model $p(O_t|S_t)$ is evaluated using the locations' binomial probability score. Aiming to categorize the generated binomial scores into filter observations, the value range is split into two parts based on the probability threshold *th*:

$$p(O_t|S_t = No\ Loop) = \begin{cases} 1.00, \text{ if } O_t > th, \forall l \in L. \\ 0.00, \text{ if } O_t < th, \exists l \in L. \end{cases} \quad (8)$$

$$p(O_t|S_t = Loop) = \begin{cases} 0.46, \text{ if } O_t > th, \forall l \in L. \\ 0.54, \text{ if } O_t < th, \exists l \in L. \end{cases} \quad (9)$$

As shown, our observation model seeks into the set of locations L for the existence of database entries l which satisfy the binomial conditions. Notably, the system's initialization probabilities are set to a no loop-closure belief $p(S_0) = \langle 1, 0 \rangle$, which derives from our confidence that such detection cannot occur at the beginning of any trajectory. The proposed model is summarized in the diagram in Fig. 3, while a discussion regarding the selected probability values is offered in Section 5.3.

4.4. A new or an old location?

Posterior, in the probabilistic context, means "after taking into account the relevant observation related to the examined cases". After $p(S_t|O_t)$ has been updated and normalized, the highest hypothesis is accepted as full posterior, i.e., if the loop-closure hypothesis $p(S_t = No\ Loop \mid O_t)$ is higher than 50%, the system adds a new location to the database, otherwise a "*Loop*" is detected.

4.5. Location matching

Since the votes' distribution affects a group of consecutive images, the 10 most similar candidate loop-closing locations are considered among the database entries that satisfy the conditions in Section 4.2. In addition, when the perceived query camera measurement performs a loop in the trajectory, while none of the database observation scores satisfy the aforementioned conditions ($O_t > th, \forall l \in L$), a temporal consistency constraint is adopted so as to determine the candidates images. In order to cope with possible false positive detections, owed to potential perceptual aliasing in the environment (e.g., when different places contain similar visual cues), the selected camera frames are subjected to a geometrical check. In such a way, image pairs that cannot be correlated by a transformation matrix are rejected independently from their visual similarity. An image-to-image correlation is performed between the query I_Q and the accepted candidates. Computations are executed until a valid matrix is estimated through an ascending binomial score order.

4.5.1. Temporal consistency

Let us consider that at time $t-1$, the system correctly indicates a previously visited location by matching pair $\langle I_{Q-1}, I_{M-1} \rangle$ and that at time t, the filter also indicates a loop; however, none of the locations satisfy the binomial threshold ($O_t > th, \forall l \in L$). The temporal constrain defines a group of images, which are the only set of database entries to be further examined as loop-closures. In this paper, we determine this window as of size $2\kappa + 1$ centered around $I_M - 1$, i.e., $[I_{(M-1)-\kappa}, \ldots, I_{(M-1)+\kappa}]$. Nevertheless, locations which are not assigned with a binomial score are excluded.

4.5.2. Geometrical verification

A fundamental matrix is estimated, through a RANSAC-based (RANSAC stands for RANdom SAmple Consensus) scheme, which is required to be supported by at least ϕ point inliers between the query I_Q and the matched image I_M. To compute these correspondences, tracked features are compared with the descriptors from the chosen location. It is also worth noting that during the above geometrical verification, our original approach achieved reduced computational complexity by using only the database features that contributed to the formulation of tracked words. Nevertheless, this characteristic also leads to fewer feature associations impairing the computation of a valid fundamental matrix and, in turn, to the rejection of valid loop-closing image pairs. To cope with such cases, we extract a set of SURF descriptors the cardinality of which is twice as big as the ones of each frame's TP, thus, offering an efficient balance between accuracy and computational complexity. It is worth mentioning that the key-point matches and the estimated fundamental matrix for a valid loop-closure event can be provided to a SLAM architecture, to be used within a bundle adjustment framework or for re-localization, without any additional cost.

4.6. Visual vocabulary management

The goal of this process is to effectively handle the increasing rate of the vocabulary, which, until this stage, adds new elements no matter if a similar entry already exists. The objective is to remove multiple codewords of repetitive pattern representing the same environmental element at different time-stamps. On top of the database size and computational complexity reduction, this unmonitored development also results in a voting ambiguity. This issue is mostly evident when the agent revisits a certain route more than twice, in which case the query image will distribute votes from the same physical location to multiple ones, decreasing the system's discriminability.

Algorithm 1: Vocabulary management

```
Input: $I_Q$: Incoming image, $I_M$: Matched image, $Idx$:
       Location indexing list, $W_n$: Newly generated tracked
       word, $d_n$: Newly generated tracked word's
       descriptors, $RL$: Reference list
Output: $Idx$: Updated location indexing list, $BoTW$: Visual
        vocabulary
for each newly generated tracked word $W_n$ do
    $id$ = find(max($RL$, $W_n$)) // select the most voted word in
      database based on the $W_n$ descriptors' polling history
    $dist$ = norm($W_n$ - $W_{id}$) // euclidean distance
    $member$ = $Idx$($id$, $I_M$) // matched image contains most
      voted word
    if $dist$ < 0.4 and $member$ == true then
        $W_{id}$ = median($W_{id}$, $d_n$) // refresh the word's
          description
        $Idx$ = update($Idx$) // refresh the location indexing list
          based on the generated word's map position
    else
        $BoTW$ = add($W_n$) // add newly generated word since
          it does not exists in dictionary
    end
end
```

Thus, during navigation, we create a reference list based on the matching process, which indicates tracked words being voted by the query descriptors. When a loop-closure is detected, each newly generated word is checked for a descriptors-to-word correspondence, to determine if the new element needs to be further processed or not. For each sequence of tracked descriptors, the most voted word in the database is indicated. Then, a similarity comparison based on Eq. (2) is applied on the chosen words' pair ⟨newly generated, corresponding most voted⟩, in which tracked words are considered to be similar if their distance is lower than 0.4. However, despite this check, the corresponding vocabulary's entry needs to satisfy a location condition check, meaning that the selected word is ignored if it is not associated with the

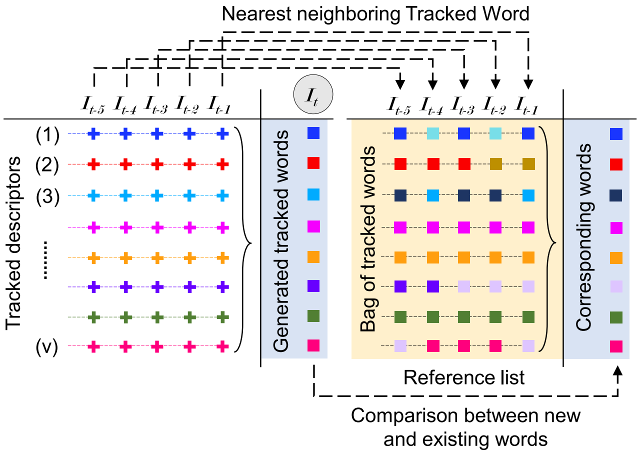

Nearest neighboring Tracked Word
I_{t-5} I_{t-4} I_{t-3} I_{t-2} I_{t-1}
I_t
I_{t-5} I_{t-4} I_{t-3} I_{t-2} I_{t-1}
(1)
(2)
(3)
(v)
Tracked descriptors
Generated tracked words
Bag of tracked words
Corresponding words
Reference list
Comparison between new and existing words

Fig. 4. The process of vocabulary management. As the trajectory escalates $(\ldots, I_{t-3}, I_{t-2}, I_{t-1}, I_t)$ along with voting procedure, a reference list regarding the tracked descriptors (block of crosses) and their nearest-neighboring tracked words (block of squares) is maintained. When the query location I_t is identified as a loop-closure, the most recently generated tracked words are checked with the most reported ones indicated via the reference list, in order to decide if they should be accumulated into the existing vocabulary.

chosen loop-closing image. Subsequently, tracked descriptors of the generated word and the one existing in the database are merged according to Eq. (3). Finally, the vocabulary's indexing list *Idx*, regarding the tracked word's locations, is updated to include the images corresponding to the merged word. A representative example is depicted in Fig. 4, while Algorithm 1 details this process.

5. Experimental evaluation

This section starts with an introduction of the experimental methodology followed to evaluate the proposed framework. Then, an extensive set of tests on several community datasets are demonstrated.

5.1. Methodology

To evaluate the BoTW-LCD performance, precision–recall metrics are utilized [18]. Precision is defined as 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 loop-closure detections to the total of ground truth loop-closures (sum of true-positive and false-negative). A correct match is considered to be any identification that occurs within a small radius from the query location, while any false-positive detection lies outside this area. False-negative detections correspond to the locations that ought to have been recognized, but the method failed to. This ground truth information is shaped in the form of a matrix whose rows and columns represent images with different time indices, while its boolean elements are set to 1 in the case that a loop-closure exists and 0 otherwise. A loop-closure algorithm aims to achieve the highest possible recall score for a flawless precision (i.e., with no false-positives). The need to maximize the recall score highly depends on the SLAM method associated with the loop-closure detection task and used during the autonomous mission. If a metric SLAM with very accurate odometry is used, then the recall ratio can be low at 100% precision. However, within the scope of this work, we assume the general case of a less accurate odometry, which makes it essential to achieve the highest possible recall metrics.

Table 1
BoTW-LCD parameters.

Parameter		Value
SURF point response, Φ	:	400.0
Maximum # of Tracked Points, ν	:	150
Minimum points' distance, α	:	5
Minimum descriptors' distance, β	:	0.6
Minimum tracked word's length, ρ	:	5
Minimum RANSAC inliers, ϕ	:	8
Temporal consistency, κ	:	8

An entry-level system with an Intel Core i7-6700HQ (2.6 GHz) processor and 8 GB RAM was used in all of the following experiments, while BoTW-LCD was configured by using the parameters summarized in Table 1. The parameters regarding the visual vocabulary generation (e.g., points' distance α, descriptors' distance β and tracked word's length τ) are set in accordance to the evaluation in [68] and remained the same over all datasets to demonstrate the system's adaptability. The rest of the proposed parameters are extensively evaluated in Section 5.4. The performance of the presented approach is compared against existing state-of-the-art methods which are based on an incrementally generated visual vocabulary, while for the sake of completeness, we also compare the proposed framework against several well-known pre-trained ones.

5.2. Community datasets

A total of nine publicly-available datasets were chosen to validate the proposed system. With a view to assess the parameter's effect over the achieved performance, three of these datasets were selected as evaluation cases, containing mostly ground-level views from outdoor and dynamic urban areas. With a purpose to highlight the system's adaptability and its capacity to generalize, the identified parameters are then fixed and assessed on the remaining six sets, treated as testing cases, which represent different environments from the evaluation ones. Table 2 provides a brief description of each dataset used.

5.2.1. Evaluation datasets

The KITTI vision suite collection [75] constitutes a widely-known benchmark environment in the robotics community as it provides a broad range of traversed routes, accurate odometry data, long-term operational conditions and high-resolution visual information (with respect to both image-size and frame-rate). The incoming visual stream is obtained by means of a stereo camera system, which is mounted on a forward-moving car. We considered only the left camera stream from courses 00 and 05 since, compared to the rest, they provide substantial loop-closure examples. The third evaluation dataset was selected from Lip6 Outdoor (Lip6O) [29]. The visual information is provided by a hand-held camera encountering plenty of loop events in the traversed path. An important characteristic of this set is the fact that the camera visits some of the recorded locations more than twice, making it ideal for the assessment of the proposed vocabulary management mechanism. Also, due to the sensor's low resolution and frame-rate, this dataset constitutes one of the most challenging cases among our tests. Regarding the related loop-closure data, Lip6O contains its own ground truth information, as provided by the authors. About the KITTI courses, this information was manually extracted through the dataset's odometry data by [79].

Table 2
Properties of the utilized datasets.

Dataset	Environment characteristics	Number of frames	Traversed distance	Image size & frequency	Camera orientation
[75] KITTI 00	Outdoor, urban, dynamic	4551	≈ 12.5 km	1241 × 376, 10 Hz	Frontal
[75] KITTI 02	Outdoor, urban, dynamic	4661	≈ 13.0 km	1241 × 376, 10 Hz	Frontal
[75] KITTI 05	Outdoor, urban, dynamic	2761	≈ 7.5 km	1241 × 376, 10 Hz	Frontal
[75] KITTI 06	Outdoor, urban, dynamic	1101	≈ 3.0 km	1241 × 376, 10 Hz	Frontal
[29] Lip 6 Outdoor	Outdoor, urban, highly dynamic	1063	≈ 1.5 km	240 × 192, 1 Hz	Frontal
[76] EuRoC MH 05	Indoor, static	2273	≈ 0.1 km	752 × 480, 20 Hz	Frontal
[77] Malaga 6L	Outdoor, static	3474	≈ 1.2 km	1024 × 768, 7 Hz	Frontal
[78] New College	Outdoor, dynamic	2624	≈ 2.2 km	512 × 384, 1 Hz	Frontal
[16] City Centre	Outdoor, urban, dynamic	1237	≈ 1.9 km	1024 × 768, 7 Hz	Lateral

5.2.2. Testing datasets

Since the proposed pipeline needs to be adaptable to a variety of different environments, six datasets are selected with diverse visual properties, which are widely used in visual SLAM research and, in particular, in evaluating loop-closure detection. The tested courses consist of three outdoor, urban environments, an indoor industrial area recorded by an aerial vehicle, an outdoor university campus parking lot registered from a ground-moving vehicle and a college's campus park which was recorded via a wheeled robot. Among the above, the KITTI sequences 02 and 06 are selected aiming to evaluate our vocabulary's evolution size as they provide trajectories wherein loops are not frequently presented. The EuRoC Machine Hall 05 (EuRoC MH 05) part of the EuRoC Micro Aerial Vehicle (MAV) dataset [76] is also utilized, as it provides rapid velocity variations along the trajectory and multiple examples of loop-closure events with slight fluctuations in illumination. Visual respective sensory information is provided by cameras mounted on a MAV with a high acquisition frame-rate. Malaga 2009 Parking 6L (Malaga 6L) [77], New College [78] and City Centre [16] have been registered by the vision system of an electric buggy-typed vehicle and a robotic platform, respectively. They refer to significantly different operational conditions (e.g., traveled distance, frame size, acquisition frequency, camera orientation), as presented in Table 2. However, they both contain a significant amount of loop-closure examples. Note also that New College's incoming visual data were resampled to 1 Hz, from its initial 20 Hz rate, due to the robot's low velocity and high camera frequency, simulating a more representative example of modern robotic platforms. The incoming visual stream in the aforementioned datasets is provided through a stereo system, yet only the right monocular data are utilized here. The ground truth information used in the experiments for EuRoC MH 05, Malaga 6L and New College are as in [68], City Centre contains its own.

5.3. Parameter discussion

In this subsection, we briefly discuss the temporal parameters. In general, the performance of BoTW-LCD relies on the transition $p(S_t|S_{t-1})$ and observation $p(O_t|S_t)$ probabilities (see Section 4.3). The framework in which we work is quite simple, including two states for the transition model and the observation. The transition probabilities follow the loop-closure principle which indicates that a belief state would follow its previous one. Thus, their values are appropriately attributed to almost 98% in both cases (see Section 4.3.1). The observation model is the one which plays the primary role in shaping the final decision. Aiming to highlight the probabilities produced by the binomial density function, we have chosen a high level of confidence $p(O_t|S_t = No\ Loop) = 0.00$ when the loop-closure threshold is satisfied (Eq. (8)) since its efficiency in identifying pre-visited locations has been well-established [35,45,60,68]. On the contrary, to avoid losing a possible detection in a sequence of loop events, due to the lack of satisfying the condition if $O_t > th, \forall l \in L$, its probability is defined at 46% (Eq. (9)), allowing the system to correct its belief in the following observations while maintaining its high performance. These parameters are estimated empirically while a level of confidence about their values is attributed through the Hidden Markov Model (HMM) estimation algorithm proposed by [80]. All our experiments were performed under the same set of probability filtering values.

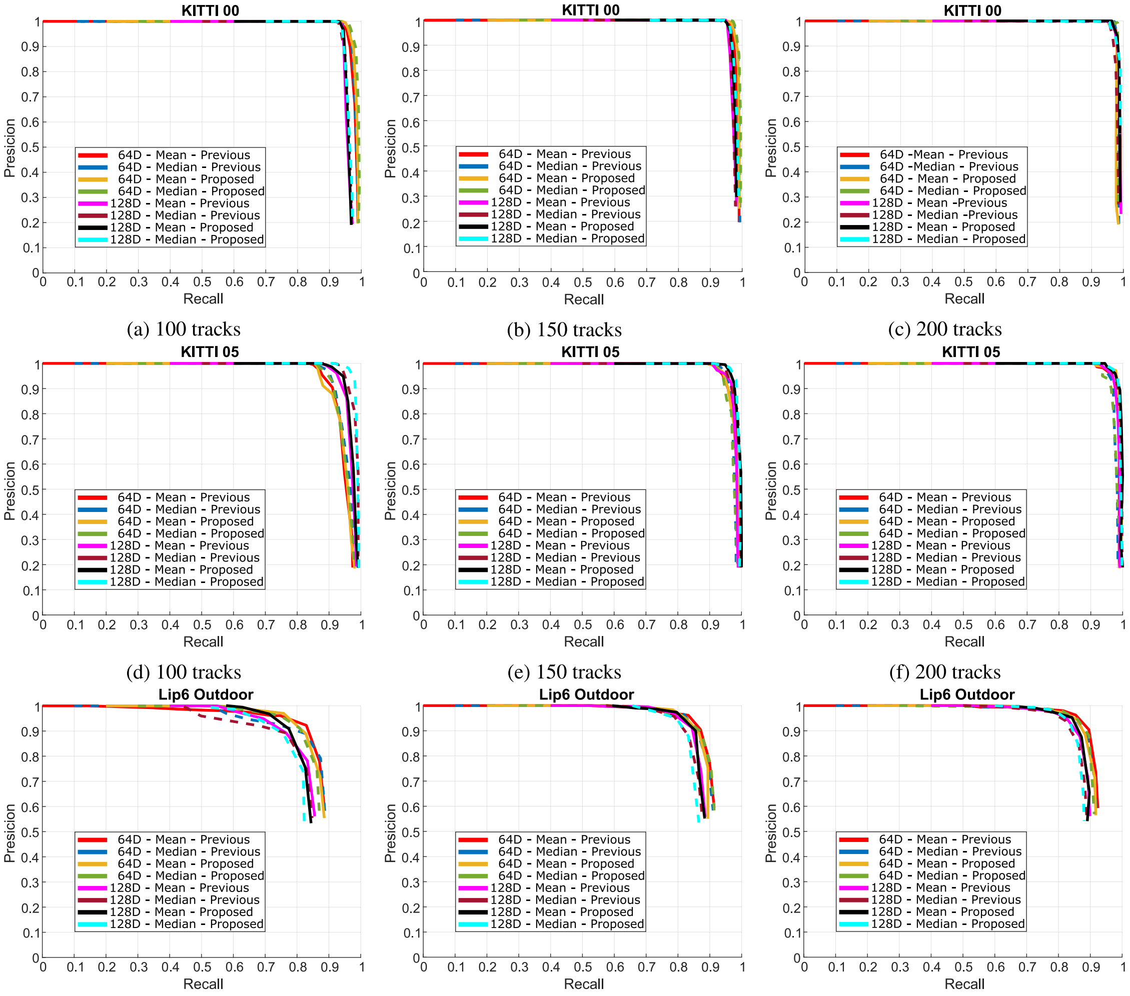
5.4. Performance evaluation

We illustrate the precision–recall rates for different cases of maximum retained tracked features ($\nu = 100, 150, 200$). In addition, we assess the tracked words' merging approach (mean, median) as well as the description method accuracy (SURF - 64D, SURF - 128D) for the achieved performance.

5.4.1. A modest vocabulary loop-closure detection

By deactivating the temporal filter and the geometrical verification check, the proposed visual vocabulary management technique is evaluated and its results are compared against our previous work [68] in Fig. 5 for each of the aforementioned cases. Our first remark is that each of the produced curves presents high recall rates on the evaluation datasets. As one can observe, the system offers a very competent performance for 150 and 200 tracked features, approaching 95% recall in both KITTI courses, while keeping perfect precision. We observe that the median achieves similar performance to the mean-based. Furthermore, the 128D version of SURF shows higher recall rates for a lower number of tracked features, exhibiting its description accuracy for both the mean and median merging methods. In Lip6O, which is evidently the most challenging image-sequence due to its low acquisition frame rate, visual resolution and rapid viewpoint variations, the recall extends to almost 90%, whilst maintaining high precision scores. It is notable that the recall curves corresponding to the proposed method, which incorporates vocabulary management, performs better in this dataset. This is owed to the fact that the specific image stream records the same route more than twice and the voting ambiguity originated from the arbitrary generation of new words is avoided.

In support thereof, we present a quantitative evaluation of the generated words in Table 3. Since our management technique is affected by the system's performance to detect loop-closures, the recorded number of words is obtained for the highest recall rate at 100% precision. A words' reduction of about 10% is observed for each case ($\nu = 100, 150, 200$) for both merging methods and descriptors dimensions. In addition, more words are ignored as the number of tracked features increases, indicating that a higher number of elements are generated and remain in the database affecting the system's discrimination capabilities. Regarding the mean and the median versions, the results show a similar output with small fluctuations. Finally, although the description accuracy for the 128D version of SURF offers a lower amount of tracked words, we argue that this fact is not decisive for our approach since its memory footprint would be double the size of the 64D one.



(a) 100 tracks (b) 150 tracks (c) 200 tracks

(d) 100 tracks (e) 150 tracks (f) 200 tracks

(g) 100 tracks (h) 150 tracks (i) 200 tracks

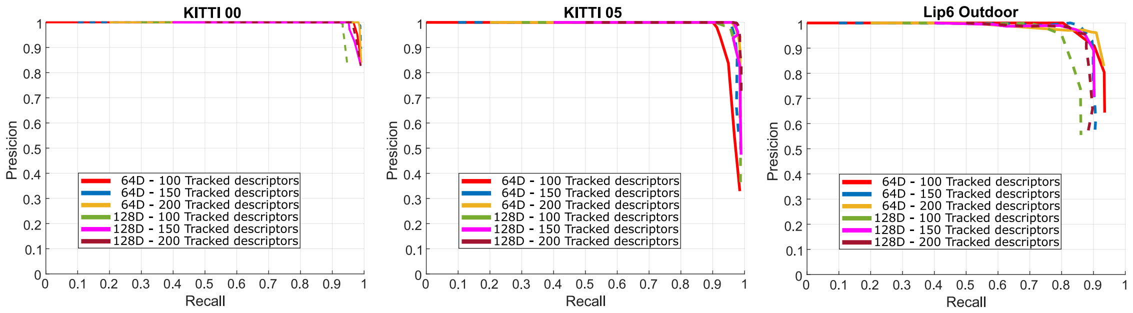
Fig. 5. Precision–recall curves evaluating the number of maximum tracked features ν against the previous approach [68] and the proposed one using the vocabulary management technique. For each version, the tracked word generation method is presented along with the different descriptor version (64 & 128 dimension space of Speeded-Up Robust Features (SURF) [9]). Experiments are performed on the KITTI sequences [75], 00 (top), 05 (middle) and Lip 6 Outdoor [29] (bottom). The proposed system seems to offer higher discrimination at voting procedure permitting similar recall rates for 100% precision between the cases of 150 and 200. The 128D SURF descriptors exhibit their robustness when a lower amount of tracked features is used as depicted in every of the evaluated dataset for the case of 100 tracks.

Table 3
Total of generated tracked words for each evaluated dataset.

Dataset	Tracked points	Previous (64D)	Proposed (64D)	Previous (128D)	Proposed (128D)
		Description method Mean/Median	Description method Mean/Median	Description method Mean/Median	Description method Mean/median
[75] KITTI 00	100	25 603/25 603	22 930/22 898	23 577/23 577	21 092/22 160
[75] KITTI 00	150	38 170/38 170	34 196/34 170	34 951/34 951	31 514/31 722
[75] KITTI 00	200	50 510/50 510	45 741/45 731	45 976/45 976	41 669/41 989
[75] KITTI 05	100	14 853/14 853	13 432/13 431	13 832/13 832	12 448/12 508
[75] KITTI 05	150	22 199/22 199	20 135/20 100	20 515/20 515	18 548/18 642
[75] KITTI 05	200	29 391/29 391	26 728/26 659	27 009/27 009	24 516/24 687
[29] Lip 6 Outdoor	100	4206/4206	3717/3724	3145/3145	2748/2788
[29] Lip 6 Outdoor	150	5776/5776	5066/5085	4309/4309	3768/3803
[29] Lip 6 Outdoor	200	6956/6956	6138/6236	5145/5145	4499/4576

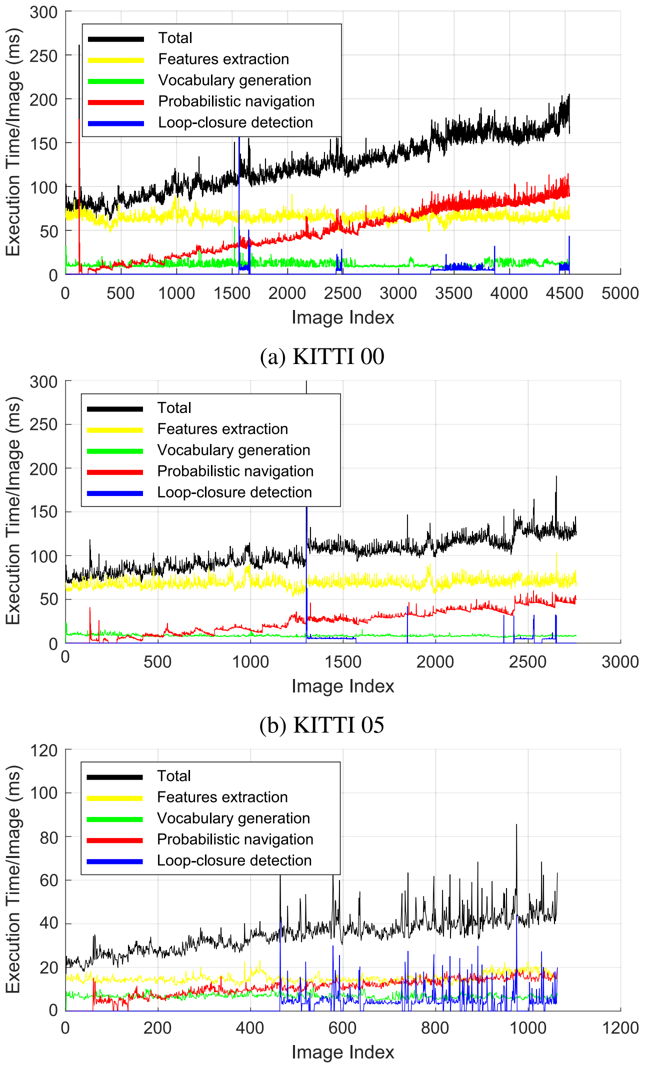
5.4.2. Bayesian filtering

We now present our evaluation of the Bayesian filtering approach which uses temporal context in Fig. 6. To exhibit the method's performance based on the binomial PDF and Bayes filter, we illustrate the precision–recall rates following the same methodology for different loop-closure threshold *th* values. The experiments have been performed on the same evaluation datasets using the median approach for generating tracked words, while the geometrical verification was not activated. As we gradually evaluate individual frames, the posterior probability for non-loop and all possible loop-closure events at each query location is evaluated based on the loop hypothesis in Section 4.4. Table 1 presents the parameters selected in order to achieve a reduced computational complexity, while still preserving increased



(a) 100 tracks (b) 150 tracks (c) 200 tracks

Fig. 6. Evaluating the proposed system's performance though the utilization of the Bayes filter. Using the median approach during the generation of tracked words, precision and recall curves are illustrated for different speeded-up robust features' dimensions (64 & 128) [9] and maximum number of tracked features ν. High recall rates are obtained for each evaluated image stream. This is owed to the exploitation of the visited locations' temporal consistency along the navigation route in combination with the binomial probabilistic scoring.



(a) KITTI 00

(b) KITTI 05

(c) Lip 6 Outdoor

Fig. 7. Execution time per image of the KITTI [75] sequences 00, 05 and Lip 6 Outdoor [29] for each of the main processing stages of the proposed algorithm.

recall rates. Concerning the overall performance, we observe that an improved score is achieved by the BoTW-LCD in every image-sequence, reaching high recall rates at 100% precision. However, it is notable that in Lip6O, an improvement in performance permits the system to reach a score of about 85% when 150 tracks are employed. When the binomial score does not satisfy condition (5), temporal consistency prevents the system from detecting false-positive events in a different, though similar, area than the one where the previous loop-closure occurs. This way, a higher recall score is attained for both descriptor versions, allowing us to avoid the 128D method since it is computation-wise and memory-wise demanding.

Table 4
Processing time per image (ms/query) for the KITTI 00, 05 and Lip 6 Outdoor datasets.

		Average time		
		KITTI		Lip 6
		00	05	Outdoor
Feature extraction	Point detection	41.4	45.5	7.6
	Point description	21.0	23.0	7.4
Vocabulary generation	Kanade–Lucas–Tomasi	8.9	6.4	5.6
	Guided feature selection	2.0	2.0	1.1
	Merging words	2.9	2.6	1.5
Probabilistic navigation	Database search	46.4	23.4	9.6
	Binomial scoring	0.8	0.8	1.0
Loop-closure detection	Geometrical verification	1.3	1.0	2.7
	Vocabulary management	1.5	0.6	2.2
Whole pipeline		126.2	105.3	38.7

5.5. System's complexity

Our method's average timing results per image are shown in Fig. 7. To measure the execution time, we ran our framework on each of the evaluation datasets. Among them, the KITTI 00 set is the longest ones exhibiting a remarkable amount of loop-closure events. For this group of experiments, a total of 4551 images are processed, yielding 126.2 ms per query image on average. Table 4 shows an extensive timing documentation for each stage. The *features extraction* process involves the computation of SURF key-points detection and description, while the *vocabulary generation* is split into three steps: the points' tracking though the KLT method, the guided feature selection and the words' merging. The *probabilistic navigation* includes both the exhaustive database search and the binomial probabilistic score computations. Lastly, *Loop-closure detection* denotes the time required for the verification step through the calculation of the corresponding fundamental matrices and the words' update due to the vocabulary management.

Table 5
Comparison with our previous approach.

Dataset	Tsintotas et al. [68]			BoTW-LCD		
	SURF	R (%)	T (ms)	SURF	R (%)	T (ms)
[75] KITTI 00	51K	97.5	173.5	**34K**	**97.7**	**126.2**
[75] KITTI 02	52K	80.0	190.2	**37K**	**81.5**	**133.0**
[75] KITTI 05	29K	92.6	130.1	**20K**	**94.0**	**105.3**
[75] KITTI 06	12K	98.1	98.7	**8K**	**98.1**	**90.1**
[29] Lip 6 Outdoor	7K	50.0	37.1	**5K**	**78.0**	**38.7**
[76] EuRoC MH 05	20K	83.7	90.8	**13K**	**85.0**	**82.6**
[77] Malaga 6L	41K	85.0	171.8	**28K**	**85.2**	**146.7**
[78] New College	18K	83.0	82.1	**10K**	**87.0**	**67.5**
[16] City Centre	3K	20.0	65.0	**2K**	**36.0**	**68.4**

Table 6
Comparison with the baseline approach.

Dataset	Gehrig et al. [60]			BoTW-LCD		
	SURF	R (%)	T (ms)	SURF	R (%)	T (ms)
[75] KITTI 00	681K	92,8	920.3	**34K**	**97.7**	**126.2**
[75] KITTI 02	699K	80,2	990.7	**37K**	**81.5**	**133.0**
[75] KITTI 05	414K	86,0	572.8	**20K**	**94.0**	**105.3**
[75] KITTI 06	165K	**98,5**	185.9	**8K**	98.1	**90.1**
[29] Lip 6 Outdoor	159K	**85,5**	232.9	**5K**	78.0	**38.7**
[76] EuRoC MH 05	340K	53,8	310.4	**13K**	**85.0**	**82.6**
[77] Malaga 6L	520K	64,0	770.0	**28K**	**85.2**	**146.7**
[78] New College	394K	84.7	672.6	**10K**	**87.0**	**67.5**
[16] City Centre	183K	**74,0**	232.7	**2K**	36.0	**68.4**

Table 7
Comparison with a state-of-the-art approach.

Dataset	iBoW-LCD [36]			BoTW-LCD		
	ORB	R (%)	T (ms)	SURF	R (%)	T (ms)
[75] KITTI 00	958K	76,5	400.2	**34K**	**97.7**	**126.2**
[75] KITTI 02	950K	72,2	422.3	**37K**	**81.5**	**133.0**
[75] KITTI 05	556K	53,0	366.5	**20K**	**94.0**	**105.3**
[75] KITTI 06	212K	95,5	385.1	**8K**	**98.1**	**90.1**
[29] Lip 6 Outdoor	121K	**85,2**	228.0	**5k**	78.0	**38.7**
[76] EuRoC MH 05	443K	25,6	350.4	**13K**	**85.0**	**82.6**
[77] Malaga 6L	806K	57,4	440.8	**28K**	**85.2**	**146.7**
[78] New College	254K	73.1	383.7	**10K**	**87.0**	**67.5**
[16] City Centre	67K	**88,2**	336.2	**2K**	36.0	**68.4**

The results in Table 4 show that we can reliably detect loops in datasets that expand for 11 km while maintaining low execution times. We observe that all the involved steps are notably fast, considering the fact that we utilize a floating-point descriptor through the SURF algorithm. BoTW-LCD is able to rapidly process images using a reduced set of visual words due to its innovative visual word management process. In contrast to the *binomial scoring*, the *database search* stage exhibits the highest execution time, due to the lack of an indexing scheme, followed by the *feature* extraction stage, which is known as the bottleneck point for many loop-closure approaches. The execution time for *vocabulary generation* is highly depended on the number of points and the tracker's parameters (e.g., pyramid levels, neighborhood area, maximum bidirectional error), while the required time for the *guided feature selection* and the *words' merging* is low. The *loop-closure detection* stage is also negligible. As shown in Fig. 7, the proposed system achieves to estimate a valid fundamental matrix very fast reaching a value of 3 computations, on average, between the query I_Q and the accepted candidates.

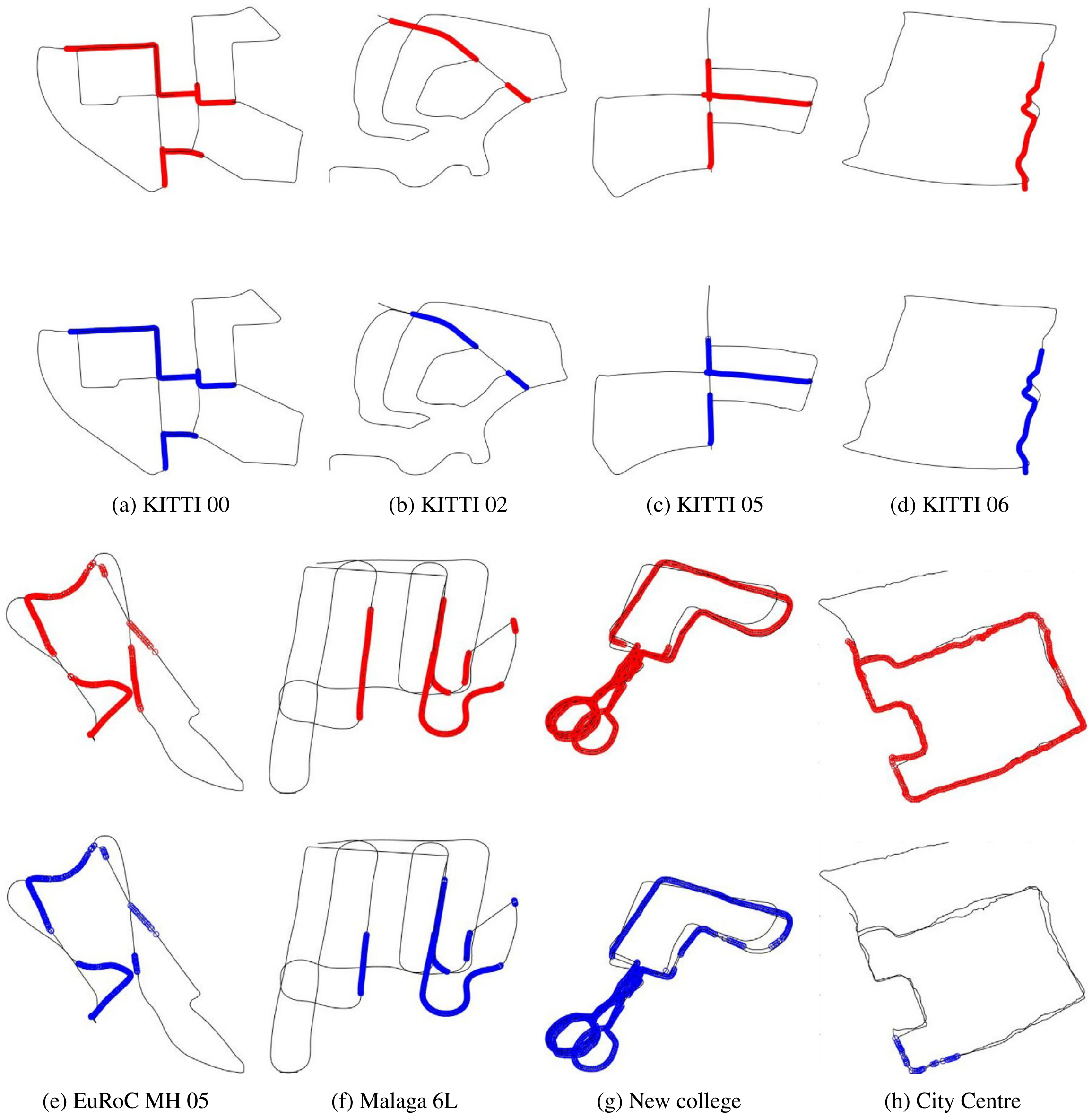
5.6. Comparative results

This section compares BoW-LCD against other state-of-the-art solutions. In this regard, Table 5 contains the final mapping size (BoTW), the maximum recall at 100% of precision (R) and the average response time per image (T) obtained for each approach and dataset. The performance of our system was measured by using a generic loop-closure threshold of $th = 2^{-9}$, which was obtained by the precision–recall curves in Fig. 6. This value was selected since it allows the system to achieve high recall rates in every evaluation dataset. During this experiment, our geometrical verification and vocabulary management modules were active, while the parameters remained constant so as to evaluate the adaptability of the approach. As can be observed, the impact in terms of recall is minimum and, in general, quite similar. However, BoTW-LCD is able to process an image in lessen time using a reduced set of visual words. We argue that this fact is mainly due to the new visual word managing process. Note that a comparison with off-line BoW schemes regarding their respective complexities is not presented since a direct analogy with methods based on a pre-trained vocabulary would not be meaningful. Following the results presented in Table 5, Fig. 8 illustrates the detections provided by BoTW-LCD at 100% precision for each image-sequence. The top path of each dataset presents the corresponding ground truth, i.e., the trajectory which should be recognized in case the framework detects every loop-closure. When a loop is detected, the image triggering this event is labeled by a blue cycle. Note that in most cases, the loops are successfully detected, especially in the sequences of KITTI dataset.

Subsequently, since a source code regarding the approach of Gehrig et al. [60] is not publicly available, we implemented a SURF-based[2] version aiming to offer a more thorough view about the impact of our mapping technique, as apposed in Table 6. This version utilizes the same amount of SURF elements as the Tracked Points ν used by the proposed method to describe the incoming frame. Furthermore, a 40 s temporal window is included for rejecting early visited locations [35]. Similarly, for searching the database and aggregating votes, $k = 1$ nearest-neighbor was selected, while the parameterization of the geometrical check between the chosen pair was also based on our presented work. The best-performing loop-closure threshold for each assessed case was evaluated according to the literature and the selected parameters remained constant over all datasets. In addition, in Table 7 we compare the proposed pipeline with the state-of-the-art framework iBoW-LCD [36], which uses binary codewords for generating the vocabulary. Its evaluation come from the open source implementation.[3] Notice the high reduction of the final mapping size (BoTW against SURF and ORB) and the timings offered by the proposed approach in comparison to the ones in the implemented methods of [36,60]. Nevertheless, as shown in both Tables 6 and 7, building a map through tracked words does not always imply higher recall values. However, it consistently reduces the computational times and the size of the final map. It is worth to mention that both in Lip6O and City Centre, which are two challenging image-sequences (e.g., due to the cameras' orientation), the other approaches perform better since are tend to work as image-retrieval methods having a distinct representation for each incoming frame. Moreover, with the aim to enrich the comparative analysis, Table 8 presents the memory consumption in each trajectory mapping for some of the most acknowledged methods that aim for a real-time and lightweight implementation. As shown, BoTW-LCD achieves the lowest footprint in every dataset. Note that the low memory usage of the iBoW-LCD vocabulary is mainly due to its binary form. Similarly, PREVIeW,

[2] The implementation of Gehrig et al. [60] SURF-based pipeline can found at: https://github.com/ktsintotas/probabilistic_voting.

[3] The iBoW-LCD [36] open-source implementation can be found at https://github.com/emiliofidalgo/ibow-lcd.


(a) KITTI 00 (b) KITTI 02 (c) KITTI 05 (d) KITTI 06

(e) EuRoC MH 05 (f) Malaga 6L (g) New college (h) City Centre

Fig. 8. Loop-closures generated from the proposed pipeline for every evaluated dataset using the parameters defined in Table 1. In each trajectory, red cycles indicate ground truth information, while the blue ones illustrate the system's detections. The top row presents the KITTI data [75] sequences 00, 02, 05 and 06, whilst EuRoC MH 05 [76], Malaga 6L [77], New College [78] and City Centre [16] are depicted in the bottom row. As can be seen in most of the cases, BoTW-LCD achieves to recognize locations when the robot traverses a route which presents similar visual content. This is especially highlighted in the KITTI datasets, where the frames are captured from a forward facing camera, in contract to City Centre's lateral sensor orientation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

which uses a binary dictionary of 1M visual words, utilizes only 30.5 Mb of memory.

Furthermore, in Table 9 our approach is compared against the most representative works in visual place recognition using online learning techniques based on both local and global descriptors, namely Angeli et al. [29], Miford and Wyeth (seqSLAM) [56], Khan and Wollherr (IBuILD) [33], Zhang et al. [34], Tsintotas et al.[4] [35], Kazmi and Mertsching [61], An et al. (FILD) [63] and Tsintotas et al.[5] [81]. In addition, for the sake of completeness, comparisons are also presented against approaches based on a pre-trained vocabulary with the aim to help the reader to identify the place of the proposed pipeline within the state-of-the-art. More specifically, Cummins and Newman (FAB-MAP 2.0) [18], Gálvez-López and Tardós (DBoW2) [19], Mur-Artal and Tardós (DBoW2-ORB) [20], Bampis et al. (PREVIeW) [23] and Yue et al. are chosen. The maximum recall scores achieved at 100% precision for each approach are based on the figures reported in the original papers for image-sequences with the ground truth provided by the respective authors. The term N/A indicates that the corresponding information is *not available* from any cited source, while the dash (–) designates that the approach fails to reach a recall score for perfect precision. Regarding PREVIeW[6] and FILD,[7] evaluation occurred based on the open source implementations, with the default parameter configurations provided in the

[4] The implementation of Tsintotas et al. [35] can be found at https://github.com/ktsintotas/assigning-visual-words-to-places.

[5] The implementation of Tsintotas et al. [81] can be found at https://github.com/ktsintotas/tracking-DOSeqSLAM.

[6] The PREVIeW open-source implementation can be found at https://github.com/loukbabi/PREVIeW.

[7] The FILD open-source implementation can be found at https://github.com/AnshanTJU/FILD.

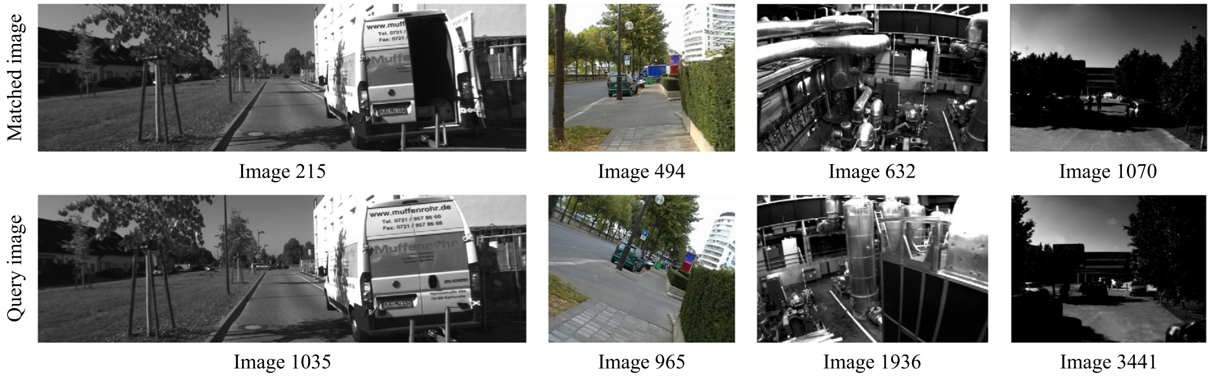


Fig. 9. Example images which are correctly identified by the proposed framework as loop-closure detections. The query frame I_Q is the image recorded by the robot at time t, whereas the matched frame I_M corresponds to the chosen location. From left to right: KITTI 02 [75], Lip 6 Outdoor [29], EuRoC MH 05 [76] and Malaga 6L parking [77].

respective codes. OpenSeqSLAM[8] has been configured through OpenSeqSLAM2.0[9] toolbox [82] except the sequence length of the images (ds) which is, according to the author [83], the most critical parameter of the algorithm. Longer sequence lengths usually perform better in terms of precision–recall, but in some datasets such as KITTI, our experiments showed the opposite behavior. Since we want to increase the performance of each approach, this parameter was experimentally set to 20, that maximized the recall in all environments. Furthermore, a 40 s temporal window, similar to [35], was applied to reject early visited locations. The best results at 100% of precision are chosen for each dataset. The authors in [61] performed the presented evaluations based on our ground truth information. In addition, for the case of FAB-MAP 2.0 [18] and DBoW2 [19], where no actual measurements are provided regarding the used datasets, the presented performance is obtained from the setup described by [61] and [84], respectively.

By examining Table 5, one can observe the significantly high score achieved by our method in the Lip6O dataset. We succeed to excel among our preliminary work, highlighting the importance of the temporal information across the trajectory. Nonetheless, our framework performs unfavorably against iBoW-LCD and Kazmi and Mertsching as shown in Table 9. This is due to the geometrical verification parameterization which strengthens our framework's precision accuracy in the cost of missing some of its potential performance, but also due to the fact that the system encounters a route of low textured images (avg. features/image), as shown in Table 2, impairing our feature tracking procedure. This characteristic drops the recall rate when the geometrical verification step is applied since some of the true-positive detections are discarded as they fail to produce a valid fundamental matrix with enough inliers. However, we also need to stress out that our method's performance is able to reach even higher recall rate than the ones in Table 9 (as illustrated in the precision–recall curves in Fig. 6), yet our aim is to present a system with a homogeneous set of parameters that can be used in any environment. Thus, the adopted probability threshold and the RANSAC inliers are selected and fixed so as to maintain high scores for 100% precision across every evaluated dataset. In the KITTI 00 set, the proposed framework exhibits over 97% of recall results, while compared to the rest of the sequences of the KITTI suite, it outperforms most of the competitors. Moreover, in the testing cases, our framework demonstrates a significant improvement to the obtained recall. In EuRoC MH 05, Malaga 6L, and New College a score of 85% is reached on each dataset, while holding a high precision rate. It is noteworthy that in EuRoC, where the system confronts an environment of low illumination, the binary description methods, adopted in PREVIeW and iBoW-LCD, are unable to perform competitively compared to the floating point features. Similarly, global descriptors utilized in [56,63] and [81] present low recall scores. This results in a high divergence in terms of recall scores against the proposed pipeline. Finally, in the case of City Centre, our system fails to follow the performance of the other solutions. This fact implies that our mapping procedure performs better when the camera's orientation is frontal, allowing the formulation of prolonged word tracks. In Fig. 9, some accurately detected locations are shown.

6. Discussion and future work

In this article, an improved loop-closure detection approach is presented under the BoTW-LCD framework. The proposed solution comprises a completely incremental and on-line appearance-based loop-closure detection architecture, which allows the trajectory to be mapped by a significantly lower number of words extracted from tracked features. The method does not require any training process regarding its visual vocabulary leading to a set of words which are fully adapted to each individual environment. We have maintained the system's complexity as low as possible, avoiding redundant accumulation of new points as long as a feature's tracking holds. This essentially leads to the generation of a representative vocabulary which is significantly shorter than any other cited work. Thus our method achieves to incrementally map the environment in real-time (up to 20 frames-per-second) and has a lower run-time memory footprint during deployment. It is noteworthy that less than 37K visual words are totally produced for a route of 13 km using 8.3 Mb of memory. The proposed pipeline can achieve high recall scores for perfect precision in all tested datasets outperforming existing state-of-the-art methods while maintaining low execution times.

By applying a probabilistic voting scheme when searching for pre-visited locations into the map, a high degree of confidence about the images' similarity is achieved. Furthermore, a Bayes filter exploits the temporal aspect of the data gathered along the traversed path and finally, a geometrical verification step is performed to reject possible remaining outliers. Using a vocabulary management technique, tracked words are further merged when a pre-visited location is detected reducing the computational time and the vocabulary's size while improving the system's accuracy. Though our extensive experimentation, we showed that 64D SURF descriptors are able to outperform the 128D ones achieving higher accuracy and lower memory consumption. As

[8] The OpenSeqSLAM open-source implementation can be found at http://openslam.org/openseqslam.html.

[9] The OpenSeqSLAM2.0 open-source implementation can be found at https://github.com/kadn/OpenSeqSLAM2.0.

Table 8
Memory usage for different state-of-the-art systems. Bold values indicate minimum consumption per evaluated dataset.

Method	KITTI				Lip 6	EuRoC	Malaga	New	city
	00	02	05	06	Outdoor	MH 05	6L	College	Centre
	(Mb)	(Mb)	(Mb)	(Mb)	(Mb)	(Mb)	(Mb)	(Mb)	(Mb)
[68] Tsintotas et al.	12.4	12.6	7.0	2.9	1.7	4.8	10.0	4.3	0.7
[60] Gehrig et al.	166.2	170.6	101.0	40.2	38.8	83.0	126.9	96.1	44.6
[36] Garcia-Fildago and Ortiz (iBoW-LCD)	29.2	28.9	16.9	6.4	3.6	13.5	24.5	7.7	2.8
[23] Bampis et al. (PREVIeW)	30.5	30.5	30.5	30.5	30.5	30.5	30.5	30.5	30.5
BoTW-LCD	**8.3**	**9.0**	**4.8**	**1.9**	**1.2**	**3.1**	**6.8**	**2.4**	**0.5**

Table 9
Comparison of maximum recall at 100% precision. Bold values indicate maximum performance per evaluated dataset.

	KITTI				Lip 6	EuRoC	Malaga	New	City
	00	02	05	06	Outdoor	MH 05	6L	College	Centre
Off-line approaches									
[18] Cummins and Newman (FAB-MAP2.0)	61.2	44.3	48.5	64.5	N/A	N/A	21.8	52.6	40.1
[19] Gálvez-López and Tardós (DBoW2)	72.4	68.2	51.9	89.7	N/A	N/A	74.7	47.5	30.6
[20] Mur-Artal and Tardós (DBoW2-ORB)	N/A	N/A	N/A	N/A	N/A	N/A	81.5	N/A	43.3
[23] Bampis et al. (PREVIeW)	96.5	72.0	**97.3**	80.1	58.3	23.1	33.9	80.8	71.1
[85] Yue et al.	97.4	N/A	93.0	98.0	N/A	N/A	N/A	N/A	**90.5**
On-line approaches									
[29] Angeli et al.	N/A	N/A	N/A	N/A	71.0	N/A	N/A	N/A	N/A
[56] Milford and Wyeth (SeqSLAM)	74.8	63.8	52.1	95.6	26.0	12.1	20.5	41.7	85.0
[33] Khan and Wollherr (IBuILD)	92.0	N/A	N/A	N/A	25.6	N/A	78.1	N/A	38.9
[34] Zhang et al.	N/A	N/A	N/A	N/A	N/A	N/A	82.6	N/A	41.1
[35] Tsintotas et al.	93.1	76.0	94.2	86.0	12.0	69.2	87.9	88.0	16.3
[61] Kazmi and Mertsching	90.3	79.4	81.4	97.3	**84.9**	26.8	50.9	51.0	75.5
[63] An et al. (FILD)	91.2	65.1	85.1	93.3	0.3	–	56.0	76.7	66.4
[81] Tsintotas et al. (Tracking-DOSeqSLAM)	77.6	61.1	38.2	–	40.9	–	42.0	40.0	47.1
BoTW-LCD	**97.7**	**81.5**	94.3	**98.1**	78.0	**85.0**	**87.9**	**89.2**	36.0

shown in Table 9, the recall rates are particularly high in most of the assessed image-sequences. Regarding the case of Lip6O, wherein the camera traverses the same trajectory three times, the accuracy was increased by over 20% in contrast to our previous work. Compared to the state-of-the-art, our algorithm achieved high performance and real-time behavior on routes as large as 13 km.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

The authors would also like to thank S. M. Ali Musa Kazmi for evaluating their pipeline created for loop-closure detection based on our ground truth matrices for each of the utilized dataset.

References

[1] E. Garcia-Fidalgo, A. Ortiz, Vision-based topological mapping and localization methods: A survey, Robot. Auton. Syst. 64 (2015) 1–20.

[2] S. Lowry, N. Sünderhauf, P. Newman, J.J. Leonard, D. Cox, P. Corke, M.J. Milford, Visual place recognition: A survey, IEEE Trans. Robot. Autom. 32 (2016) 1–19.

[3] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, J.J. Leonard, Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age, IEEE Trans. Robot. Autom. 32 (2016) 1309–1332.

[4] I. Kostavelis, A. Gasteratos, Semantic mapping for mobile robotics tasks: A survey, Robot. Auton. Syst. 66 (2015) 86–103.

[5] V. Balaska, L. Bampis, I. Kansizoglou, A. Gasteratos, Enhancing satellite semantic maps with ground-level imagery, Robot. Auton. Syst. (2021).

[6] K.L. Ho, P. Newman, Loop closure detection in SLAM by combining visual and spatial appearance, Robot. Auton. Syst. 54 (2006) 740–749.

[7] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2004) 91–110.

[8] E. Rosten, T. Drummond, Fusing points and lines for high performance tracking, in: Procceding of the IEEE International Conference on Computer Vision, 2005, pp. 1508–1515, http://dx.doi.org/10.1109/ICCV.2005.104.

[9] H. Bay, A. Ess, T. Tuytelaars, L.V. Gool, Speeded-Up Robust Features (SURF), Comput. Vis. Image Underst. 110 (2008) 346–359.

[10] M. Calonder, V. Lepetit, C. Strecha, P. Fua, BRIEF: Binary Robust Independent Elementary Features, in: Proceedings of the European Conference on Computer Vision, 2010, pp. 778–792, http://dx.doi.org/10.1007/978-3-642-15561-1_56.

[11] S. Leutenegger, M. Chli, R.Y. Siegwart, BRISK: Binary robust invariant scalable keypoints, in: Proceedings of the IEEE International Conference on Computer Vision, Barcelona, Spain, 2011, pp. 2548–2555, http://dx.doi.org/10.1109/ICCV.2011.6126542.

[12] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, ORB: An efficient alternative to SIFT or SURF, in: Proceedings of the IEEE International Conference on Computer Vision, Barcelona, Spain, 2011, pp. 2564–2571, http://dx.doi.org/10.1109/ICCV.2011.6126544.

[13] P.F. Alcantarilla, A. Bartoli, A.J. Davison, KAZE features, in: Proceedings of the European Conference on Computer Vision, 2012, pp. 214–227, http://dx.doi.org/10.1007/978-3-642-33783-3_16.

[14] H. Zhang, BoRF: Loop-closure detection with scale invariant visual features, in: Proceedings of the IEEE International Conference on Robotics and Automation, Shanghai, China, 2011, pp. 3125–3130, http://dx.doi.org/10.1109/ICRA.2011.5980273.

[15] R. Baeza-Yates, B. Ribeiro-Neto, et al., Modern Information Retrieval, Vol. 463, 1999.

[16] M. Cummins, P. Newman, FAB-MAP: Probabilistic localization and mapping in the space of appearance, Int. J. Robot. Res. 27 (2008) 647–665.

[17] C. Mei, G. Sibley, P. Newman, Closing loops without places, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 2010, pp. 3738–3744, http://dx.doi.org/10.1109/IROS.2010.5652266.

[18] M. Cummins, P. Newman, Appearance-only SLAM at large scale with FAB-MAP 2.0, Int. J. Robot. Res. 30 (2011) 1100–1123.

[19] D. Gálvez-López, J.D. Tardós, Bags of binary words for fast place recognition in image sequences, IEEE Trans. Robot. Autom. 28 (2012) 1188–1197.

[20] R. Mur-Artal, J.D. Tardós, Fast relocalisation and loop closing in keyframe-based SLAM, in: Proceedings of the IEEE International Conference on Robotics and Automation, Hong Kong, China, 2014, pp. 846–853, http://dx.doi.org/10.1109/ICRA.2014.6906953.

[21] E.S. Stumm, C. Mei, S. Lacroix, Building location models for visual place recognition, Int. J. Robot. Res. 35 (2016) 334–356.

[22] L. Bampis, A. Amanatiadis, A. Gasteratos, Encoding the description of image sequences: A two-layered pipeline for loop closure detection, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2016, pp. 4530–4536, http://dx.doi.org/10.1109/IROS.2016.7759667.

[23] L. Bampis, A. Amanatiadis, A. Gasteratos, Fast loop-closure detection using visual-word-vectors from image sequences, Int. J. Robot. Res. 37 (2018) 62–82.

[24] V. Balaska, L. Bampis, M. Boudourides, A. Gasteratos, Unsupervised semantic clustering and localization for mobile robotics tasks, Robot. Auton. Syst. (2020) 103567.

[25] I.T. Papapetros, V. Balaska, A. Gasteratos, Multi-layer map: Augmenting semantic visual memory, in: Proceedings of the International Conference on Unmanned Aircraft Systems, Athens, Greece, 2020, pp. 1206–1212, http://dx.doi.org/10.1109/ICUAS48674.2020.9213923.

[26] J. MacQueen, et al., Some methods for classification and analysis of multivariate observations, in: Proceedings of the Berkeley Symposium on Mathematical Statistics and Probability, 1967, pp. 281–297.

[27] J. Sivic, A. Zisserman, Video Google: A text retrieval approach to object matching in videos, in: Proceedings of the IEEE International Conference on Computer Vision, Nice, France, 2003, pp. 1470–1477, http://dx.doi.org/10.1109/ICCV.2003.1238663.

[28] D. Filliat, A visual bag of words method for interactive qualitative localization and mapping, in: Procedings of the IEEE International Conference on Robotics and Automation, 2007, pp. 3921–3926, http://dx.doi.org/10.1109/ROBOT.2007.364080.

[29] A. Angeli, D. Filliat, S. Doncieux, J. Meyer, Fast and incremental method for loop-closure detection using bags of visual words, IEEE Trans. Robot. Autom. 24 (2008) 1027–1037.

[30] A. Kawewong, N. Tongprasit, S. Tangruamsub, O. Hasegawa, Online and incremental appearance-based SLAM in highly dynamic environments, Int. J. Robot. Res. 30 (2011) 33–55.

[31] T. Nicosevici, R. Garcia, Automatic visual bag-of-words for online robot navigation and mapping, IEEE Trans. Robot. Autom. 28 (2012) 886–898.

[32] M. Labbé, F. Michaud, Appearance-based loop closure detection for online large-scale and long-term operation, IEEE Trans. Robot. Autom. 29 (2013) 734–745.

[33] S. Khan, D. Wollherr, IBuILD: Incremental bag of binary words for appearance based loop closure detection, in: Proceedings of the IEEE International Conference on Robotics and Automation, Seattle, WA, USA, 2015, pp. 5441–5447, http://dx.doi.org/10.1109/ICRA.2015.7139959.

[34] G. Zhang, M.J. Lilly, P.A. Vela, Learning binary features online from motion dynamics for incremental loop-closure detection and place recognition, in: Procedings of the International Conference on Robotics and Automation, Stockholm, Sweden, 2016, pp. 765–772, http://dx.doi.org/10.1109/ICRA.2016.7487205.

[35] K.A. Tsintotas, L. Bampis, A. Gasteratos, Assigning visual words to places for loop closure detection, in: Proceedings of the IEEE International Conference on Robotics and Automation, Brisbane, QLD, Australia, 2018, pp. 5979–5985, http://dx.doi.org/10.1109/ICRA.2018.8461146.

[36] E. Garcia-Fidalgo, A. Ortiz, iBoW-LCD: An appearance-based loop-closure detection approach using incremental bags of binary words, IEEE Robot. Autom. Lett. 3 (2018) 3051–3057.

[37] J.P. Company-Corcoles, E. Garcia-Fidalgo, A. Ortiz, Towards robust loop closure detection in weakly textured environments using points and lines, in: Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation, 2020, pp. 1313–1316, http://dx.doi.org/10.1109/ETFA46521.2020.9212133.

[38] N. Sünderhauf, S. Shirazi, F. Dayoub, B. Upcroft, M. Milford, On the performance of convnet features for place recognition, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2015, pp. 4297–4304, http://dx.doi.org/10.1109/IROS.2015.7353986.

[39] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, J. Sivic, NetVLAD: CNN architecture for weakly supervised place recognition, in: Proceeding of the IEEE Conference Computer Vision and Pattern Recognition, 2016, pp. 5297–5307, http://dx.doi.org/10.1109/TPAMI.2017.2711011.

[40] F. Radenović, G. Tolias, O. Chum, CNN image retrieval learns from BoW: Unsupervised fine-tuning with hard examples, in: Proceedings of the European Conference on Computer Vision, 2016, pp. 3–20, http://dx.doi.org/10.1007/978-3-319-46448-0_1.

[41] I. Kansizoglou, L. Bampis, A. Gasteratos, Deep feature space: A geometrical perspective, 2020, arXiv preprint arXiv:2007.00062.

[42] F. Maffra, Z. Chen, M. Chli, Tolerant place recognition combining 2D and 3D information for UAV navigation, in: Proceedings of the IEEE International Conference on Robotics and Automation, 2018, pp. 2542–2549, http://dx.doi.org/10.1109/ICRA.2018.8460786.

[43] F. Maffra, L. Teixeira, Z. Chen, M. Chli, Real-time wide-baseline place recognition using depth completion, IEEE Robot. Autom. Lett. 4 (2019) 1525–1532.

[44] B.D. Lucas, T. Kanade, An iterative image registration technique with an application to stereo vision, in: Proceedings of the International Joint Conference on Artificial Intelligence, San Francisco, CA, USA, 1981, pp. 674–679.

[45] K.A. Tsintotas, P. Giannis, L. Bampis, A. Gasteratos, Appearance-based loop closure detection with scale-restrictive visual features, in: Proceedings of the International Conference on Computer Vision Systems, 2019, pp. 75–87, http://dx.doi.org/10.1007/978-3-030-34995-0_7.

[46] J. Zobel, A. Moffat, Inverted files for text search engines, ACM Comput. Surv. 38 (2006) 6.

[47] M. Cummins, P. Newman, Probabilistic appearance based navigation and loop closing, in: Proceedings of the IEEE International Conference on Robotics and Automation, IEEE, 2007, pp. 2042–2048.

[48] C. Chow, C. Liu, Approximating discrete probability distributions with dependence trees, IEEE Trans. Inform. Theory 14 (1968) 462–467.

[49] W. Maddern, M. Milford, G. Wyeth, Cat-slam: probabilistic localisation and mapping using a continuous appearance-based trajectory, Int. J. Robot. Res. 31 (2012) 429–451.

[50] D. Hiemstra, A probabilistic justification for using tf×idf term weighting in information retrieval, Int. J. Digit. Libr. 3 (2000) 131–139.

[51] D. Arthur, S. Vassilvitskii, k-means++: The advantages of careful seeding, in: Proceedings of the ACM-SIAM Symposium on Discrete Algorithms, 2007, pp. 1027–1035.

[52] P. Hansen, B. Browning, Visual place recognition using HMM sequence matching, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2014, pp. 4549–4555.

[53] R. Arroyo, P.F. Alcantarilla, L.M. Bergasa, E. Romera, Towards life-long visual localization using an efficient matching of binary sequences from images, in: Proceedings of the IEEE International Conference on Robotics and Automation, 2015, pp. 6328–6335.

[54] K.A. Tsintotas, L. Bampis, A. Gasteratos, DOSeqSLAM: dynamic on-line sequence based loop closure detection algorithm for SLAM, in: Proceedings of the IEEE International Conference on Imaging Systems and Techniques, IEEE, 2018, pp. 1–6.

[55] P. Neubert, S. Schubert, P. Protzel, A neurologically inspired sequence processing model for mobile robot place recognition, IEEE Robot. Autom. Lett. 4 (2019) 3200–3207.

[56] M.J. Milford, G.F. Wyeth, SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights, in: Proceedings of the IEEE International Conference on Robotics and Automation, 2012, pp. 1643–1649, http://dx.doi.org/10.1109/ICRA.2012.6224623.

[57] K.A. Tsintotas, L. Bampis, S. Rallis, A. Gasteratos, SeqSLAM with bag of visual words for appearance based loop closure detection, in: Proceedings of the International Conference on Robotics in Alpe-Adria Danube Region, Patras, Greece, 2018, pp. 580–587, http://dx.doi.org/10.1007/978-3-030-00232-9_61.

[58] E. Garcia-Fidalgo, A. Ortiz, Hierarchical place recognition for topological mapping, IEEE Trans. Robot. Autom. 33 (2017) 1061–1074.

[59] B. Fritzke, A growing neural gas network learns topologies, in: Proceedings of the International Conference on Neural Information Processing Systems, 1994, pp. 625–632.

[60] M. Gehrig, E. Stumm, T. Hinzmann, R. Siegwart, Visual place recognition with probabilistic voting, in: Proceedings of the IEEE International Conference on Robotics and Automation, Singapore, Singapore, 2017, pp. 3192–3199, http://dx.doi.org/10.1109/ICRA.2017.7989362.

[61] S.M.A.M. Kazmi, B. Mertsching, Detecting the expectancy of a place using nearby context for appearance-based mapping, IEEE Trans. Robot. Autom. 35 (2019) 1352–1366.

[62] D. Alahakoon, S.K. Halgamuge, B. Srinivasan, Dynamic self-organizing maps with controlled growth for knowledge discovery, IEEE Trans. Neural Netw. 11 (2000) 601–614.

[63] S. An, G. Che, F. Zhou, X. Liu, X. Ma, Y. Chen, Fast and incremental loop closure detection using proximity graphs, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2019, pp. 378–385.

[64] T. Senst, V. Eiselein, T. Sikora, Robust local optical flow for feature tracking, IEEE Trans. Circuits Syst. Video Technol. 22 (2012) 1377–1387.

[65] X. Lan, A.J. Ma, P.C. Yuen, Multi-cue visual tracking using robust feature-level fusion based on joint sparse representation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1194–1201.

[66] G. Zhang, H. Liu, Z. Dong, J. Jia, T.-T. Wong, H. Bao, Efficient non-consecutive feature tracking for robust structure-from-motion, IEEE Trans. Image Process. 25 (2016) 5957–5970.

[67] R. Mur-Artal, J.D. Tardós, Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras, IEEE Trans. Robot. Autom. 33 (2017) 1255–1262.

[68] K.A. Tsintotas, L. Bampis, A. Gasteratos, Probabilistic appearance-based place recognition through bag of tracked words, IEEE Robot. Autom. Lett. 4 (2019) 1737–1744.

[69] J.L. Bentley, Multidimensional binary search trees used for associative searching, Commun. ACM 18 (1975) 509–517.
[70] S. Lynen, M. Bosse, P. Furgale, R. Siegwart, Placeless place-recognition, in: Proceedings of the International Conference on 3D Vision, Vol. 1, 2014, pp. 303–310.
[71] Y. Liu, H. Zhang, Indexing visual features: Real-time loop closure detection using a tree structure, in: Proceedings of the IEEE International Conference on Robotics and Automation, 2012, pp. 3613–3618.
[72] A. Babenko, V. Lempitsky, The inverted multi-index, IEEE Trans. Pattern Anal. Mach. Intell. 37 (2014) 1247–1260.
[73] D. Schlegel, G. Grisetti, HBST: A hamming distance embedding binary search tree for feature-based visual place recognition, IEEE Robot. Autom. Lett. 3 (2018) 3741–3748.
[74] A.C. Murillo, G. Singh, J. Kosecká, J.J. Guerrero, Localization in urban environments using a panoramic gist descriptor, IEEE Trans. Robot. Autom. 29 (2012) 146–160.
[75] A. Geiger, P. Lenz, R. Urtasun, Are we ready for autonomous driving? The KITTI Vis. benchmark suite, in: Proceedings of the IEEE Conference on Computer Vision and Rattern Recognition, 2012, pp. 3354–3361, http://dx.doi.org/10.1109/CVPR.2012.6248074.
[76] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M.W. Achtelik, R. Siegwart, The EuRoC micro aerial vehicle datasets, Int. J. Robot. Res. 35 (2016) 1157–1163.
[77] J.-L. Blanco, F.-A. Moreno, J. Gonzalez, A collection of outdoor robotic datasets with centimeter-accuracy ground truth, Auton. Robots 27 (2009) 327.
[78] M. Smith, I. Baldwin, W. Churchill, R. Paul, P. Newman, The new college visual and laser data set, Int. J. Robot. Res. 28 (2009) 595–599.
[79] R. Arroyo, P.F. Alcantarilla, L.M. Bergasa, J.J. Yebes, S. Bronte, Fast and effective visual place recognition using binary codes and disparity information, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2014, pp. 3089–3094, http://dx.doi.org/10.1109/IROS.2014.6942989.
[80] R. Durbin, S.R. Eddy, A. Krogh, G. Mitchison, Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids, 1998.
[81] K.A. Tsintotas, L. Bampis, A. Gasteratos, Tracking-DOSeqSLAM: A dynamic sequence-based visual place recognition paradigm, IET Comput. Vis. (2021).
[82] B. Talbot, S. Garg, M. Milford, OpenSeqSLAM2.0: An open source toolbox for visual place recognition under changing conditions, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2018, pp. 7758–7765, http://dx.doi.org/10.1109/IROS.2018.8593761.
[83] M. Milford, Vision-based place recognition: how low can you go? Int. J. Robot. Res. 32 (2013) 766–789.
[84] S. An, H. Zhu, D. Wei, K.A. Tsintotas, Fast and incremental loop closure detection with deep features and proximity graphs, 2020, arXiv preprint arXiv:2010.11703.
[85] H. Yue, J. Miao, Y. Yu, W. Chen, C. Wen, Robust loop closure detection based on bag of superpoints and graph verification, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Macau, China, 2019, pp. 3787–3793, http://dx.doi.org/10.1109/IROS40897.2019.8967726.

Konstantinos A. Tsintotas received the bachelor's degree from the department of automation engineering, Technological Education Institute of Central Greece, Psachna, Greece and the master's degree in mechatronics from the department of electrical and computer engineering, University of Western Macedonia, Kozani, Greece, in 2010 and 2015, respectively. Since 2016, he has been a Ph.D. candidate in the field of robotic vision at the laboratory of robotics and automation (LRA), department of production and management engineering in the Democritus University of Thrace, Xanthi, Greece. His research interests include vision-based place recognition methods for simultaneous localization and mapping applications in mobile, autonomous robots.

Loukas Bampis received the diploma in electrical and computer engineering and PhD degree in machine vision and embedded systems from the Democritus University of Thrace (DUTh), Greece, in 2013 and 2019, respectively. He is currently a postdoctoral fellow in the laboratory of robotics and automation (LRA), department of production and management engineering, DUTh. His work has been supported through several research projects funded by the European Space Agency, the European commission and the Greek government. His research interests include real-time localization and place recognition techniques using hardware accelerators and parallel processing.

Antonios Gasteratos received the M.Eng. and Ph.D. degrees from the department of electrical and computer engineering, Democritus University of Thrace (DUTh), Greece. He is a professor and head of department of production and management engineering, DUTh, Greece. He is also the director of the laboratory of robotics and automation (LRA), DUTh and teaches the courses of robotics, automatic control systems, electronics, mechatronics and computer vision. During 1999–2000 he was a visiting researcher at the laboratory of integrated advanced robotics (LIRALab), DIST, University of Genoa, Italy. He has served as a reviewer for numerous scientific journals and international conferences. He is a subject editor at electronics letters and an associate editor at the international journal of optomechatronics and he has organized/co-organized several international conferences. His research interests include mechatronics and robot vision. He has published more than 220 papers in books, journals and conferences. He is a fellow member of IET and a senior member of the IEEE.