



Appearance-based loop closure detection combining lines and learned points for low-textured environments

Joan P. Company-Corcoles¹ · Emilio Garcia-Fidalgo¹ · Alberto Ortiz¹

Received: 20 April 2021 / Accepted: 20 December 2021 / Published online: 9 February 2022
© The Author(s) 2022

Abstract

Hand-crafted point descriptors have been traditionally used for visual loop closure detection. However, in low-textured environments, it is usually difficult to find enough point features and, hence, the performance of such algorithms degrade. Under this context, this paper proposes a loop closure detection method that combines lines and learned points to work, particularly, in scenarios where hand-crafted points fail. To index previous images, we adopt separate incremental binary Bag-of-Words (BoW) schemes for points and lines. Moreover, we adopt a binarization procedure for features' descriptors to benefit from the advantages of learned features into a binary BoW model. Furthermore, image candidates from each BoW instance are merged using a novel query-adaptive late fusion approach. Finally, a spatial verification stage, which integrates appearance and geometry perspectives, allows us to enhance the global performance of the method. Our approach is validated using several public datasets, outperforming other state-of-the-art solutions in most cases, especially in low-textured scenarios.

Keywords Appearance-based localization · Loop closure detection · Visual place recognition · Simultaneous localization and mapping

1 Introduction

Simultaneous Localization and Mapping (SLAM) is a fundamental task in robotics that allows an agent to build a map of an unknown environment while, at the same time, tracks its position within this map (Cadena et al. 2016). SLAM systems typically rely on Loop Closure Detection (LCD) methods to identify when the robot has returned to a previously visited place (Stewart et al. 2002). As it is well known, the accumulated drift can be greatly reduced using loop closure information. Additionally, LCD is also used for robot relocalization after a tracking failure (Williams et al. 2011). Cameras have been fundamentally used in the last decade, among other sensors that can be adopted for LCD, due to

the richness of imagery as well as the low cost of these devices nowadays. These methods are globally referred to as *appearance-based* LCD (Angeli et al. 2008; Milford and Wyeth 2012; Cummins and Newman 2011; Galvez-López and Tardos 2012; Khan and Wollherr 2015; Garcia-Fidalgo and Ortiz 2018; Bampis et al. 2018).

The performance of appearance-based LCD methods is mainly determined by: (1) the image description technique adopted; and (2) the ability of the approach to efficiently index previous images (Garcia-Fidalgo and Ortiz 2015). Concerning image description, hand-crafted point features have extensively been used for LCD tasks. Among these solutions, binary descriptors (Calonder et al. 2010; Rublee et al. 2011) have progressively emerged as an alternative to real-valued descriptors (Lowe 2004; Bay et al. 2006) given their demonstrated advantages in computational terms. More recently, the aforementioned techniques have successfully been replaced by methods based on Convolutional Neural Networks (CNN), which have proven to be more robust to illumination and viewpoint changes than hand-crafted features (DeTone et al. 2018). CNN-based methods have typically been used as holistic descriptors. Other approaches (DeTone et al. 2018; Dusmanu et al. 2019; Revaud et al. 2019) have been recently shown to be able to detect and

✉ Joan P. Company-Corcoles
joanpep.company@uib.es

Emilio Garcia-Fidalgo
emilio.garcia@uib.es

Alberto Ortiz
alberto.ortiz@uib.es

¹ Department of Mathematics and Computer Science, University of the Balearic Islands, and IDISBA (Institut d'Investigació Sanitària de les Illes Balears), Palma de Mallorca, Spain

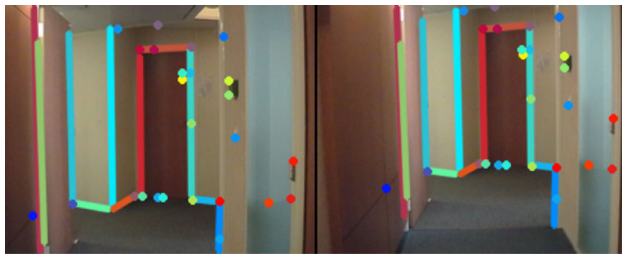


Fig. 1 Lines can support point features in low-textured scenarios: (left) query image, (right) loop-closing candidate image selected by our approach. Points and line segments are labelled as circles and lines, respectively. Matchings found are indicated by the same color in both images

describe simultaneously interest points in an image with adequate performance; none of them, however, produce binary descriptors. On the other side, while hand-crafted point features have shown impressive results working on well-textured scenarios (Mur-Artal and Tardós 2017), their performance decreases when dealing with low-textured scenarios, where the number of feature points detected results to be rather low (Pumarola et al. 2017). Precisely because of their nature, low-textured, human-made environments usually exhibit structural regularities that can be described using line features. All this has taken us to the combination of both visual perspectives, lines and point features, in order to enhance loop closing performance and widen the range of environments that can be addressed (Company-Corcoles et al. 2020). By way of illustration, Fig. 1 shows the benefits of combining both kinds of features in a low-textured scenario.

In reference to image indexing, arguably the most used scheme is the combination of the Bag of Words model (BoW) [50], Nister and Stewenius (2006) with an inverted file. In this model, descriptors are quantized into visual words, according to the available visual vocabulary. Images are then described by a histogram of occurrences of each visual word in the image. Depending on how this visual vocabulary is generated, the available solutions can be classified as either off-line or on-line (Garcia-Fidalgo and Ortiz 2015). The off-line approach needs a training phase to build the visual vocabulary, which can be particularly time-consuming, and besides makes the resulting solution highly dependent on the diversity of the training set (Nister and Stewenius 2006). As an alternative, the on-line approach builds the visual vocabulary incrementally, as the robot navigates, which helps to solve the aforementioned issues. A fusion strategy is additionally required to obtain a consolidated list of loop closure candidates when different sorts of features are involved, maybe by means of several BoW instances, one for each kind of feature (Zuo et al. 2017; Gomez-Ojeda et al. 2019; Han et al. 2021).

Standard BoW schemes do not account for the spatial distribution of features in the image, which tends to reduce

their accuracy under severe perceptual aliasing conditions (Galvez-López and Tardos 2012; Garcia-Fidalgo and Ortiz 2018). Due to this reason, a final spatial verification step is usually performed to check consistency at the geometric level for the resulting correspondences between the query and candidate images. In this regard, the most popular approach consists in using RANdom SAmple Consensus (RANSAC) (Fischler and Bolles 1981) to check whether the image features obey a specific motion model between images. Typically, this loop closing validation step degrades in performance when dealing with either non-rigid image transformations or a high number of outliers. To tackle this problem, some recent approaches check the consistency between images for local neighborhood structures around the correspondences found (Lowry and Andreasson 2018; Ma et al. 2019; Zheng and Doermann 2006; Bian et al. 2017).

Under this context, we introduce an appearance-based loop closure detection algorithm, named *Lines and POints for Low-Textured scenarios' Loop Closure Detection* (LiPo-LCD⁺⁺). In our approach, binarized CNN-based point descriptors and binary line descriptors are indexed using an incremental Bag-of-Binary-Words scheme and combined through a query-adaptive late fusion strategy in order to obtain an integrated list of loop closure candidates. After choosing an image as a final loop candidate, the loop closure is further validated performing a spatial verification procedure which naturally integrates point and line features. Our approach, as it is shown in the experimental results section, outperforms other solutions for generic environments, exhibiting a remarkable performance level for low-textured scenarios.

A preliminary version of LiPo-LCD⁺⁺ was introduced in Company-Corcoles et al. (2020). In this paper we extend this work with the following contributions:

- A binarization procedure to transform a CNN-based real-valued local descriptor (DeTone et al. 2018) into a binary descriptor. This binarization procedure allows us to combine the advantages of CNN-based feature descriptors with the benefits of a binary visual vocabulary.
- A query-adaptive late fusion approach to merge loop candidates retrieved using lines and points independently. This fusion strategy permits us to adapt our system dynamically to the environment by weighing automatically the candidates according to the images contents in terms of lines and point features, and, thus, increasing its performance for a wider range of scenarios.
- A novel spatial verification method which combines appearance and geometric information of feature points and lines. This method leads to the detection of a larger number of inliers when dealing with lines and attains higher accuracy in general. Moreover, we propose a fusion of feature points and lines involving local

neighborhood structures. This allows us to increase the efficiency of usual spatial verification stages for, particularly, low-textured environments.

Additionally, we report on an extensive set of experiments to validate the performance, adaptability and effectiveness of LiPo-LCD⁺⁺. As it is shown in the experimental results section, our solution compares favourably with other state-of-the-art methods in the field.

The rest of the paper is organized as follows: Sect. 2 overviews the most important works in the field; Sect. 3 describes the proposed approach, while Sect. 4 discusses on a set of experimental results to evaluate LiPo-LCD⁺⁺ performance; finally, Sect. 5 concludes the paper and suggests some future research lines.

2 Related work

Appearance-based loop closure approaches can be categorized according to their image description method (Garcia-Fidalgo and Ortiz 2015). In this regard, some works (Sünderhauf and Protzel 2011; Milford and Wyeth 2012; Arroyo et al. 2014) have opted for global descriptors for image description. These descriptors are usually very fast to compute, though, typically, they are more sensitive to viewpoint and illumination changes than local features. Due to these reasons, point local descriptors, either real-valued (Cummins and Newman 2008; Angeli et al. 2008; Cummins and Newman 2011; Tsintotas et al. 2019) or binary (Galvez-López and Tardos 2012; Mur-Artal and Tardós 2014; Khan and Wollherr 2015; Garcia-Fidalgo and Ortiz 2018), have been widely used in the literature during last decades. More recently, approaches based on CNNs have emerged as an alternative, motivated by their demonstrated robustness to visual appearance changes (Chen et al. 2014; Sünderhauf et al. 2015; Chen et al. 2017; Kenschimov et al. 2017; Lopez-Antequera et al. 2017; Yue et al. 2019). These solutions typically employ CNNs to extract a global descriptor of the image, what makes more difficult the implementation of a spatial verification stage.

The BoW model [50], Nister and Stewenius (2006), used along with an inverted file, is the most used indexing scheme for appearance-based LCD (Garcia-Fidalgo and Ortiz 2015; Lowry et al. 2016). As discussed earlier, depending on how the visual vocabulary is generated, LCD approaches are classified as either off-line or on-line methods. Off-line methods (Cummins and Newman 2011; Galvez-López and Tardos 2012; Mur-Artal and Tardós 2014; Bampis et al. 2018) generate the vocabulary during a training phase. Conversely, on-line methods (Angeli et al. 2008; Labbé and Michaud 2013; Khan and Wollherr 2015; Garcia-Fidalgo and Ortiz 2018; Tsintotas et al. 2018; Tsintotas et al. 2019) build the visual vocabulary as images are received.

Sequence-based matching is a well-known technique that has proven to be useful in LCD approaches (Milford and Wyeth 2012). Recently, Garg and Milford (Garg and Milford 2021) introduced the concept of temporal matching in a hierarchical scheme. In their solution, a learning-based sequence descriptor is first employed to select a number of place candidates. A learnt single image descriptor is next used to decide the final match. Unlike these works, our approach employs two types of features and involves temporal information in the decision after selecting a set of candidates. Additionally, we do not rely on global descriptors.

Within the context of LCD, binarizing CNN features has been adopted by multiple solutions. For instance, Arroyo et al. (2016) applies min-max normalization to the vector resulting from the concatenation of several convolutional layers. The final descriptor is next obtained by randomly selecting a specific set of features and thresholding on each component. More recently, Garg and Milford (Garg and Milford 2020) describe a coarse quantization-based hashing scheme, which employs PCA for dimensionality reduction. In Neubert et al. (2019), the authors propose a sparsified binary adaptation of Locality-Sensitive Hashing (LSH) using random projections, while other works opt for a CNN-based approach to directly generate a binary descriptor (Lin et al. 2016; Lin et al. 2019). The main difference between our approach and these solutions is that they result into a global binary descriptor for the whole image. Furthermore, unlike our solution, some of the aforementioned works (Arroyo et al. 2016; Garg and Milford 2020; Neubert et al. 2019) require a dimensionality reduction stage.

Some SLAM approaches (Pumarola et al. 2017; Zhang et al. 2019) have benefited from the combination of points and lines for image description. However, during the LCD stage, they only rely on point features, discarding line information that can be useful for some environments, unlike our approach (Company-Corcoles et al. 2020), and others that adopt a dual scheme for LCD (Gomez-Ojeda et al. 2019; Zuo et al. 2017; Han et al. 2021). Our approach resembles the idea of combining lines and points described in these works, though it differs in multiple aspects, such as (1) the point features that are employed, (2) how the visual vocabularies are built, (3) how image scores are calculated from the visual vocabularies and later combined, and (4) the spatial verification process. All these points are revised in the following:

- As for the generation of the vocabulary, all the above-mentioned works rely on an off-line method which, as commented previously, makes them highly dependent on the diversity of the training set used. This dependence does not appear in our method thanks to the on-line vocabulary acquisition approach.

- Moreover, all of them make use of ORB (Rublee et al. 2011) to detect and describe point features. Conversely, we use a CNN-based method which is more tolerant to appearance changes.
- Different weighing schemes have been proposed to merge lists of candidates. For example, Zuo et al. (2017) use static weights for each visual vocabulary, which are empirically adapted for each dataset. Since this does not consider the importance of each type of feature for every specific query image, Han et al. (2021); Gomez-Ojeda et al. (2019) propose dynamic weighing methods which can be adapted to each query image. More precisely, Han et al. (2021) computes weights from the entropy of the features found in the query image. Contrarily, Gomez-Ojeda et al. (2019) calculates the dynamic weights according to the fraction of features found, along with their distribution in the query image. Furthermore, dynamic weighing is combined with static weighing. Despite their good performance, these approaches evaluate the importance of each visual feature during feature extraction, which might not correspond to the actual importance, derived from the scores resulting from image candidates retrieval, as it is done in LiPo-LCD⁺⁺.
- Finally, regarding the final spatial verification step, Zuo et al. (2017); Gomez-Ojeda et al. (2019) require a previous pose estimation, as well as depth estimates for the features involved. Others, such as Han et al. (2021), adopt the most widely used spatial verification approach, based on a RANSAC-based estimator of the fundamental matrix. We instead adopt a strategy based on analyzing the preservation of the topological relationships between neighbouring features. Unlike these approaches, we only use the appearance to validate loop closures. Furthermore, the proposed procedure can tolerate a high number of outliers and non-rigid camera motions.

3 Proposed approach

Figure 2 illustrates LiPo-LCD⁺⁺. As can be observed, two sets of features, i.e. lines and points, are found in every new image that is processed. Each of these features is described by means of a binary descriptor, which is then used to: (1) obtain a list of the most similar images according to this visual cue; and (2) update the corresponding visual vocabulary. Next, the two lists are merged using a scored-voting procedure, giving rise to a combined list of loop-closing candidates. Subsequently, a temporal filter, based on the concept of *dynamic islands* (Garcia-Fidalgo and Ortiz 2018), groups those images close in time and prevent adjacent frames from competing among them as loop candidates. To validate the existence of a loop closure, the representative image of the best island is selected as the final loop candidate image and it is spatially assessed against the query image. We deal next with the details of every step.

3.1 Image description

As stated previously, LiPo-LCD⁺⁺ uses points and lines to describe images. The rationale behind this is that the combination of multiple and complementary descriptions improves the performance and robustness of LCD methods (Hausler and Milford 2020). By way of illustration, textureless scenes may be more distinctively described using lines than only points, while other environments that lack structure can be better described by means of point features. To this end, in our solution, the image I_t at time t is described by $\phi(I_t) = \{P_t, L_t\}$, being P_t a set of local keypoint descriptors and L_t a set of line-segment descriptors.

3.1.1 Point description

Given the robustness shown by CNN-based methods, we use the SuperPoint approach (DeTone et al. 2018) to obtain a set of keypoints from the image under consideration. SuperPoint is a learning-based feature detection and

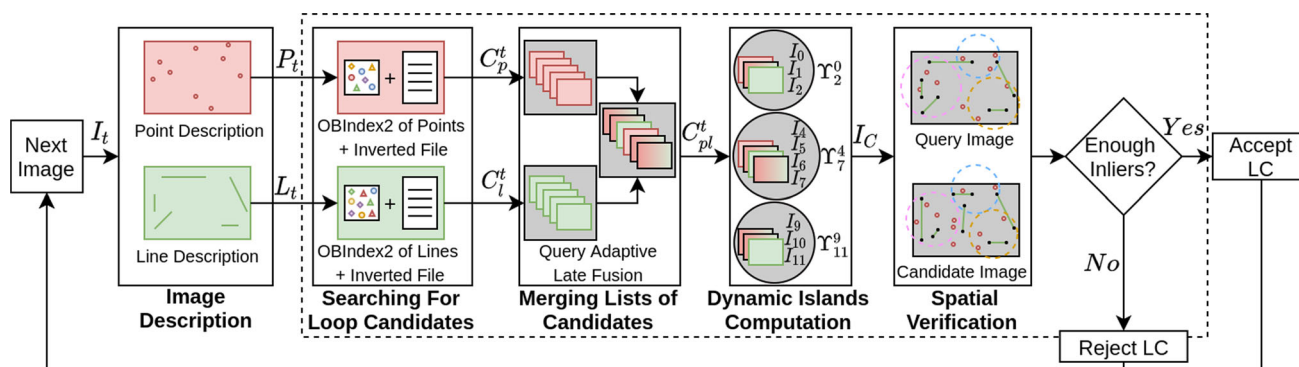


Fig. 2 General overview of the proposed approach



Fig. 3 Feature points found using: (left) ORB, 46; (center) SIFT, 61; and (right) SuperPoint, 87

description framework that is specially effective under illumination and viewpoint changes. Furthermore, compared to similar solutions such as R2D2 (Revaud et al. 2019) and D2-Net (Dusmanu et al. 2019), SuperPoint detects and describes features in an affordable time. In LiPo-LCD⁺⁺, we use the default model provided by the authors, which was trained using a wide range of scenarios. This is to enhance the performance of our approach in a larger range of environments, in contrast to other CNN-based approaches, which are typically trained using a specific scenario. Figure 3 shows an example of how SuperPoint detects, in an indoor scenario, a higher number of features than other hand-crafted features like ORB (Rublee et al. 2011) or SIFT (Lowe 2004). Besides, we have observed that SuperPoint tends to find keypoints in a more uniform way throughout the image than other detectors.

SuperPoint, similarly to other recent CNN-based approaches for keypoint detection, produces real-valued descriptors (DeTone et al. 2018; Dusmanu et al. 2019; Revaud et al. 2019). In this work, we are interested in binary descriptors to take advantage of their well-known computational benefits. To this end, our proposal inspires on other binary description methods (Calonder et al. 2010; Rublee et al. 2011), which perform pairwise comparisons between pixels within an image patch. In our case, we compare the values of different components of the real-valued descriptor. Formally, we arrange a D -dimensional real-valued descriptor $d \in \mathbb{R}^D$ as a concatenation of 8-component M subvectors: $d = [d_0, \dots, d_{M-1}]$, with $d_j = [d_{j,0}, \dots, d_{j,7}]$, $j \in \{0, M - 1\}$. In this way, each 8-component vector leads to an 8-bit string for each pair, which can be stored using a single byte. Given two subvectors d_x and d_y from d , their corresponding 8-bit string β is hence defined as

$$\beta(d_x, d_y) = \sum_{0 \leq i \leq 7} 2^i \tau(d_{x,i}, d_{y,i}), \tag{1}$$

where τ is a comparison test given by:

$$\tau(d_{x,i}, d_{y,i}) = \begin{cases} 0 & \text{if } d_{x,i} < d_{y,i} \\ 1 & \text{otherwise} \end{cases}. \tag{2}$$

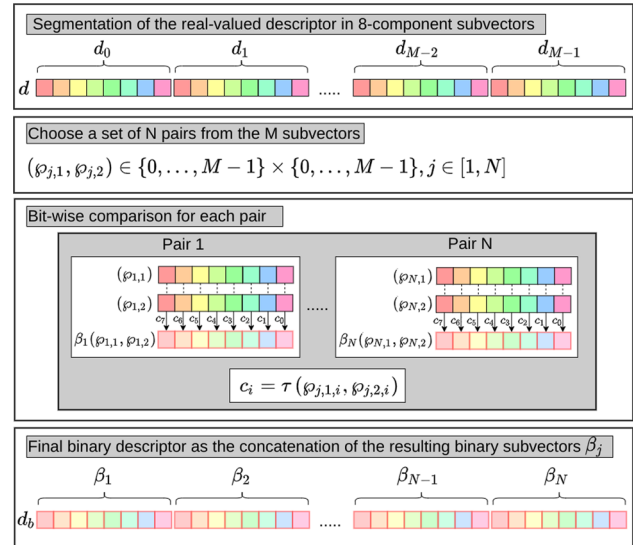


Fig. 4 Overview of the descriptor binarization procedure. The different colours denote the same position into the 8-component vector/8-bit string. Darker and lighter colours correspond to, respectively, real and binary values. In the drawing, $\beta_j = \beta(d_{\varphi_{j,1}}, d_{\varphi_{j,2}})$

The final binary descriptor d_b for d results from the 8-bit strings of a chosen set of N pairs $(\varphi_{j,1}, \varphi_{j,2}) \in \{0, \dots, M - 1\} \times \{0, \dots, M - 1\}$ from the M subvectors of d , as follows:

$$d_b = \bigoplus_{j=1}^N \beta(d_{\varphi_{j,1}}, d_{\varphi_{j,2}}), \tag{3}$$

where \bigoplus stands for the concatenation of binary descriptors. The binarization procedure is illustrated in Fig. 4.

SuperPoint generates a 256-dimensional descriptor, i.e. $D = 256$, resulting into $M = 32$ subvectors d_x , so that the set of $N \leq \binom{32}{2}$ φ_j pairs leads to a binary descriptor d_b of $N \times 8$ bits. In this work, we assess three criteria for selecting these pairs, since the chosen pairs affect the performance of the LCD approach. For each case, we generate $N = 32$ and $N = 64$ pairs, which result into, respectively, 256- and

512-bit binary descriptors. We briefly review each selection method in the following.

The first proposal applies a dimensionality reduction procedure to the original real-valued descriptor and, then, combines all-against-all possible pairs from the resulting subvectors. To this end, we make use of Gaussian Random Projections (Dasgupta 2013) to transform the original space into a simpler K -dimensional space. The conversion is performed by means of a projection matrix $R \in \mathbb{R}^{D \times k}$. Each entry is independently sampled from a standard normal distribution $N(0,1)$. In our experiments, we project SuperPoint descriptors onto 72 and 96 dimensions, which results into, respectively, 9 and 12 subvectors d_x per descriptor. We choose all the possible pairs on each case, respectively $\binom{9}{2} = 36$ and $\binom{12}{2} = 66$, discarding four combinations in the first case $-(0,1)$, $(0,2)$, $(6,8)$ and $(7,8)$ – to reach the 32 pairs of the final binary descriptor, and discarding two combinations $-(0,1)$ and $(10,11)$ – to end with 64 pairs in the second case. There is no particular reason for discarding these pairs, except for the fact of involving the first and last components of the descriptor in both cases.

As a second proposal, we consider one of the approaches assessed by the BRIEF descriptor authors in Calonder et al. (2010), which choose randomly the N pairs. In the seminal work, this random criterion seems to provide better results than the other pair selection schemes considered. We thus generate two random patterns of 32 and 64 pairs from all the possible combinations of M subvectors.

As a third alternative, we propose a novel but simple method to produce the corresponding N pairs. To generate a 256-bit descriptor, each subvector d_k is paired with its neighbours $\{d_{k-1}, d_{k+1}\}$. Similarly, the 512-bit descriptor is computed using the two nearest neighbours of d_k , i.e. $\{d_{k-2}, d_{k-1}, d_{k+1}, d_{k+2}\}$. For the 256-bit case, in practice, one has to consider pairs (d_k, d_{k+1}) , $k = 0..30$, together with pair (d_{31}, d_0) ,¹ for a total of 32 pairs. For the 512-bit case, one has to consider the previous 32 pairs and pairs (d_k, d_{k+2}) , $k = 0..29$, together with (d_{30}, d_0) and (d_{31}, d_1) , for a total of 64 pairs.

As is shown in Sect. 4.2, this last approach is the one leading to the best performance. To finish, we denote the final set of binarized descriptors as P_t .

3.1.2 Line description

Lines are detected in the image using the Line Segment Detector (LSD) (Grompone von Gioi et al. 2010). LSD is a linear-time high-precision line segment detector with subpixel accuracy without parameter tuning. However, LSD tends to break long line segments into shorter segments or even duplicated segments. The existence of these repli-

cas often affects the matching procedure and increases the storage requirements. To minimize the incidence of these misbehaviours, we perform a line merging step after segment detection. Formally, each line segment k is represented by its start point \mathbf{s}_k and its end point \mathbf{e}_k in homogeneous coordinates. The direction vector l for segment k is thus given by:

$$l_k = \frac{\mathbf{s}_k - \mathbf{e}_k}{\|\mathbf{s}_k - \mathbf{e}_k\|}. \quad (4)$$

For segment merging, we consider every possible pair of lines (l_i, l_j) . Lines are merged if the minimum Euclidean distance for the four possible combinations of end-points is smaller than a threshold and the angle between line-segments θ_{ij} , computed as

$$\theta_{ij} = \arctan \left(\frac{\|\vec{l}_i \times \vec{l}_j\|}{\vec{l}_i \cdot \vec{l}_j} \right), \quad (5)$$

is close to 0 or π radians.

Similarly to LiPo-LCD (Company-Corcoles et al. 2020), the lines found are described using the binary form of the Line Band Descriptor (LBD) (Zhang and Koch 2013), conforming the final set of binary line descriptors L_t .

3.2 Searching for loop closure candidates

Similarly to our earlier work (Company-Corcoles et al. 2020), we make use of a hierarchical tree structure to efficiently retrieve loop closure candidates, what allows us to manage an increasing number of visual words, i.e. online operation. To this end, we employ *OBIndex2* (Garcia-Fidalgo and Ortiz 2018) as an incremental BoW scheme for binary descriptors. In combination with an inverted index, it allows for fast image retrieval.

More precisely, LiPo-LCD⁺⁺ maintains two instances of *OBIndex2*, one for point features and another one for line features. Each instance builds an incremental visual vocabulary jointly with an inverted index of images. For a given query image I_t , a search is performed on each index to retrieve the most similar images with regard to each visual perspective. To this end, we search each visual feature descriptor within its corresponding dictionary to find the closest visual word. Next, using the inverted index, a list of images where the visual word has been observed is retrieved, and, for each one, a TF-IDF weight is added to the final score of the image. Candidate images are then sorted according to this final score. For a further explanation, the reader is referred to Garcia-Fidalgo and Ortiz (2018).

As a result, two lists are obtained: (1) the m most similar images using point features $C_p^t = \{I_{p_0}^t, \dots, I_{p_{m-1}}^t\}$ and (2) the n most similar images using line features $C_l^t =$

¹ Considering the SuperPoint descriptor as a circular vector.

$\{I_{l_0}^t, \dots, I_{l_{n-1}}^t\}$. Each list is sorted by the associated scores, respectively $s_p^t(I_t, I_j^t)$ and $s_l^t(I_t, I_j^t)$, which measure the similarity between the query image I_t and each candidate image I_j . Since the range of these scores varies depending on the distribution of visual words in each vocabulary, we map them onto a common range $[0, 1]$ by applying min-max normalization given by:

$$\tilde{s}_k^t(I_t, I_j^t) = \frac{s_k^t(I_t, I_j^t) - s_k^t(I_t, I_{min}^t)}{s_k^t(I_t, I_{max}^t) - s_k^t(I_t, I_{min}^t)}, \quad (6)$$

where $s_k^t(I_t, I_{min}^t)$ and $s_k^t(I_t, I_{max}^t)$ correspond, respectively, to the minimum and the maximum scores of an image candidate list, being $k \in \{p, l\}$. To keep control of the size of the list of candidates for a specific feature C_k^t , $k \in \{p, l\}$, images whose normalized score \tilde{s}_k^t is lower than a predefined threshold are discarded. Finally, the respective visual vocabularies are updated accordingly to the set of point and line binary descriptors resulting from the current image.

3.3 Fusion of the lists of candidates

The next step is to integrate both lists C_p^t and C_l^t , each one providing loop candidates from a different visual perspective, into a joint list of similar images C_{pl}^t .

3.3.1 Overview

Techniques that combine multimodal information for image retrieval can be generically classified into *early* or *late fusion* approaches (Bhowmik et al. 2014). *Early fusion* refers to the combination of the information at the feature descriptor level, before being processed by a retrieval system. Conversely, *late fusion* works with the outputs of different retrieval systems. Regarding the latter, a common approach is to aggregate multiple ranked lists, corresponding to different visual features, using a function that provides a global confidence measure. This function can be implemented in several ways: (1) using a ranked-voting procedure that considers the position of each candidate in the list for each retrieval system; or (2) using a score-based method that combines the candidates by weighing them according to the importance of each visual feature in the query image. In a previous work (Company-Corcoles et al. 2020), we adopted a ranked-voting scheme where each candidate was scored inversely proportional to their position in the list. Despite its good performance, this scheme did not take into account the relevance, or even the existence, of each visual feature in the query image, preventing the use of information that could certainly be used to fit the approach with dynamic adaptation to each case.

To this end, in this work, we adopt a *query-adaptive late fusion* approach which exploits a score-based method to

leverage the information coming from both sorts of visual features. Particularly, we propose a late fusion approach based on the work by Zheng et al. (2015), where the importance of a visual feature for a query image can be attributed to the shape of the curve of the resulting scores, sorted in descending order. To be more precise, they argue that this curve is L-shaped for a *good* feature, indicating that a small portion of the database images attain a high score, while a *bad* feature distributes the scores along more database images. LiPo-LCD⁺⁺ accounts for this behaviour essentially by means of the calculation of the Area Under the Curve (AUC) of the scores curve, to compute adaptively the importance of each feature for each query image. The details for this computation can be found next.

3.3.2 Computation of feature relevance

Following the aforementioned ideas, we propose to compute the importance of each visual feature, for a given query image, according to the AUCs of the sorted scores of lists C_k^t , $k \in \{p, l\}$. Formally, we denote f_k as the curve of the corresponding normalized scores \tilde{s}_k^t , i.e. $f_k(j) = \tilde{s}_k^t(I_t, I_j^t)$ for a specific image candidate j . As a first step, each curve f_k is analysed to detect and remove flat tails, i.e. detecting when the curve stops decreasing. To this end, we compute a linear approximation, from lower to higher scores, for each pair of consecutive points of the curve. Then, the analysis is finished when the magnitude of the resulting slope is above a threshold, discarding the remaining candidates and leading to a shorter curve \tilde{f}_k . Assuming that the final number of candidates is C_k , its AUC, denoted by A_k , is calculated by integrating \tilde{f}_k using the composite trapezoidal rule using equally-sized partitions over the interval $[0, C_k - 1]$:

$$A_k \approx \left(\sum_{n=1}^{C_k-2} \tilde{f}_k(n) + \frac{\tilde{f}_k(C_k-1) + \tilde{f}_k(0)}{2} \right). \quad (7)$$

Once the AUC for points A_p and lines A_l has been calculated, the weights for point features w_p and line features w_l are given by:

$$w_p = \frac{1}{\frac{1}{A_p} + \frac{1}{A_l}}, \quad w_l = 1 - w_p. \quad (8)$$

We use the inverse of A_k to score higher smaller areas and so capture better the L-shape.

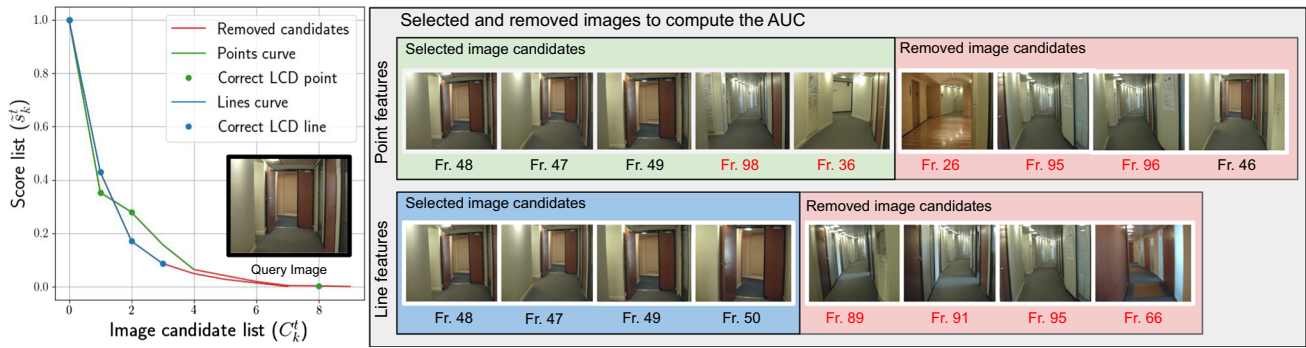


Fig. 5 (left) \tilde{f}_p and \tilde{f}_l curves, in, respectively, green and blue colour. The red segments correspond to the removed candidates for each case. (right) Selected and rejected frames to compute A_k on each case. Frame

numbers are in black for hits and in red for failures, and as green/blue dots on the left plots (Color figure online)

3.3.3 Combination of scores

Weights computed in the previous step are used to leverage scores of image candidates from the different visual perspectives, resulting into a global confidence measure. Among the different options considered to consolidate these scores in Zheng et al. (2015), we use the sum rule given its better tolerance to false positives. Therefore, we compute a joint similarity measure $S^t(I_t, I_j^t)$ between the query image I_t and a candidate image I_j^t as:

$$S^t(I_t, I_j^t) = \tilde{s}_p^t(I_t, I_j^t) \cdot w_p + \tilde{s}_l^t(I_t, I_j^t) \cdot w_l. \tag{9}$$

Finally, an integrated image list C_{pl}^t is obtained by sorting the resulting candidates according to their integrated scores $S^t(I_t, I_j^t)$.

Figure 5 illustrates the above-described merging procedure with an example. Curves f_p and f_l for, respectively, feature points and feature lines are shown on the left, while the respective images are shown on the right. The images involved in the calculation of A_p and A_l correspond to the ends of the blue and green segments of the \tilde{f}_k curves (for, respectively, lines and points). Removed image candidates for each case are shown as the ends of the red segments. Black captions below the images on the left denote correct loop closures (according to the ground truth), labelled with a coloured dot on the left curves, whereas captions for non-correct loop closures are shown in red. As a result of the whole process, for this particular case, $w_p = 0.46$ and $w_l = 0.54$, indicating that the importance of each feature, for the example query image, is almost the same, as the \tilde{f}_k curves already suggest. It can also be observed from this figure that discarded images are mostly incorrect loop closure candidates, while the selected frames to compute A_k are almost the same in both cases. It can also be noticed that, in this example, points are more affected by perceptual aliasing than lines.

3.4 Dynamic islands computation for loop candidates filtering

After merging the resulting lists C_p^t and C_l^t into C_{pl}^t , we filter C_{pl}^t to prevent that images coming from the same area compete among them as final loop candidates. To this end, we make use of the concept of *dynamic islands* (Garcia-Fidalgo and Ortiz 2018). A dynamic island \mathcal{Y}_n^m groups images whose timestamps range from m to n . The size of an island is not fixed, but it depends on the similarities between neighbouring images and the camera velocity, to adapt to the specific image stream. For each query image I_t , a set of islands Γ_t is calculated processing images in the list C_{pl}^t in ascending order by using the following procedure. Every image $I_i \in C_{pl}^t$ that lies in the $[m, n]$ interval is associated to an existing island \mathcal{Y}_n^m ; in such a case, the time interval of the island is updated to accommodate I_t and some previous and posterior frames. I_t is associated to a new island created around time t if I_t does not fit in any existing island. Note that this procedure relies on a sorted list of images.

After processing all images in C_{pl}^t , a global score g is computed for each island as:

$$g(\mathcal{Y}_n^m) = \frac{\sum_{i=m}^n S^t(I_t, I_i^t)}{n - m + 1}, \tag{10}$$

which is an indicator of how similar an area of the environment is with regard to the query image, taking into account the two visual perspectives. We also rely on the idea of *priority islands*, defined as the islands in Γ_t that overlap in time with the island selected at time $t - 1$, $\mathcal{Y}^*(t - 1)$. The rationale behind these islands is based on the fact that consecutive images should close loops with areas of the environment where previous images also closed a loop. Originally, the authors of Garcia-Fidalgo and Ortiz (2018) selected, as a

final island, the priority island with the highest score g , if any. However, they did not consider the spatial arrangement of the features and the fact that perceptual aliasing might produce incorrect island associations, especially in human-made environments. For this reason, as an alternative, LiPo-LCD⁺⁺ retains an island for the next time instant if the final selected loop candidate satisfies the spatial verification procedure, as explained in Sect. 3.5. Then, when the best island $\Upsilon^*(t)$ is determined, the image I_c with the highest combined weighted score S^t of $\Upsilon^*(t)$ is selected as its representative and passed to the next spatial verification stage.

3.5 Spatial verification

One of the main drawbacks of BoW schemes is that they ignore the spatial arrangement of visual words in the image. Under severe perceptual aliasing conditions, this might lead to produce incorrect loop associations which can be detected by checking whether some minimal physical constraints on the camera motion are satisfied or not. Therefore, a spatial verification step usually takes place to validate the loop candidate. The most widely used approach in this regard assumes a rigid motion from the camera and verifies if the motion of image features is compatible with epipolar geometry computing the fundamental matrix within a robust estimation framework, e.g. RANSAC-based, and requiring a minimum number of inliers. However, RANSAC performance decreases, either when the number of outliers becomes large, or when the camera motion cannot be modeled by means of the fundamental matrix (Ma et al. 2019). Other approaches (Zheng and Doermann 2006; Bian et al. 2017; Lowry and Andreasson 2018; Ma et al. 2019) tackle these limitations by the evaluation of neighbourhood points. Among them, Grid-based Motion Statistics (GMS) (Bian et al. 2017) and Locality Preserving Matching (LPM) (Ma et al. 2019) perform the matching task in an affordable time. However, unlike GMS, LPM avoids that various motion models, particularly when the number of detected features is low, affect the performance. This is the reason why we adopt LPM as a key component in the spatial verification step of LiPo-LCD⁺⁺.

LPM is based on the fact that the true matches from a list of candidate matches should maintain the spatial neighborhood relationships among image features, i.e. should preserve the topological structures of the scene. In this work, we modify LPM by incorporating line matches into the procedure.

3.5.1 Feature matching

The first step is to find a set of consistent matches between the query and the candidate images. In our approach, we perform a separate matching procedure for points and lines. On the one hand, matchings between the point descriptor sets P_c , for a candidate image I_c , and P_t , for the query image I_t , are val-

idated using the Nearest Neighbor Distance Ratio (NNDR) test (Lowe 2004). However, in our experiments, this approach has not led to adequate performance for lines when using a restrictive ratio, especially in low-textured environments. Instead, we propose, for lines, to combine appearance and geometrical information into the matching process. Firstly, segments are matched by their appearance. Next, geometric properties of lines are used to further filter the initial set of matches. The details can be found in the following.

Given L_c and L_t as the sets of line descriptors for, respectively, a candidate image I_c and the query image I_t , we first obtain an initial set of appearance-based line matches by applying the NNDR test, but with a more permissive threshold than for points. Next, we discard those matching pair segments whose lengths change significantly from one frame to the other. For this purpose, we calculate the ratio length μ_{ij} for every candidate matching pair (l_i, l_j) , as

$$\mu_{ij} = \frac{\max(\|s_i - e_i\|, \|s_j - e_j\|)}{\min(\|s_i - e_i\|, \|s_j - e_j\|)}. \quad (11)$$

Matching candidates with similar lengths, i.e. $\mu_{ij} \approx 1$, are accepted.

Subsequently, we assess whether the angle θ_{ij} between the two line segments, calculated using Eq. 5, is approximately 0 or π . To avoid that camera rotations between images affect this step, θ_{ij} is compensated using the global rotation between frames, as performed in Zhang and Koch (2013). Surviving line matches are considered for the local neighborhood evaluation.

3.5.2 Local Neighborhood Consistency Assessment

Once the set of putative matches for each feature are generated, we assess their consistency for their local neighborhood. For this purpose, we propose a modification of the LPM algorithm, which we will name as LP-LPM from now on, as it combines points and lines. This combination has shown to be able to enhance the matching performance in low-textured environments, where the performance of LPM usually drops. As it is explained in Ma et al. (2019), the distance between two feature points may vary under viewpoint changes, but local neighborhood structures should be similar. The generation of these local neighborhood structures unfortunately degrades when the number of detected features is low. Combining points and lines however allows for an increasing number of features over the image and leads to a better estimation of such structures.

For a start, point matches are incorporated into the LP-LPM scheme as usual, while line matches are introduced as point correspondences relating their particular endpoints s_i and e_i for, respectively, the start and the end of the line segment. First, for every line matching pair (l_i, l_j) , we deter-

Table 1 Features of the datasets involved in the experiments

Dataset	Description	Image size	Number of images
CityCentre (CC)	Urban, dynamic	640 × 480	1237
EuRoC Machine Hall 05 (EuR5)	Industrial, low-textured, highly dynamic	752 × 480	2761
KITTI 00 (K00)	Urban, dynamic	1241 × 376	4551
KITTI 05 (K05)	Urban, dynamic	1241 × 376	2761
KITTI 06 (K06)	Urban, dynamic	1241 × 376	1101
Lip6Indoor (L6I)	Indoor, low-textured, perceptual aliasing	240 × 192	388
Lip6Outdoor (L6O)	Urban, dynamic	240 × 192	1063
Malaga 2009 Parking 6L (MLG)	Parking, perceptual aliasing	1024 × 768	3474

mine the resulting point correspondences according to the value of the compensated angle θ_{ij} . If θ_{ij} is approximately 0, the line segments can be considered to have the same direction, and, therefore, we can add the point correspondences (s_i, s_j) and (e_i, e_j) to the LP-LPM framework. In case θ_{ij} is approximately π , the line directions result to be opposite vectors and, hence, pairs (s_i, e_j) and (e_i, s_j) are used as correspondences. Next, LP-LPM proceeds generating two sets of inliers. We consider a line match as an inlier if at least one of its matched endpoints has been regarded as an inlier by LP-LPM. Finally, the selected loop candidate I_c is validated as a loop closure if the total number of inliers, comprising the inliers for both features, is above a threshold. It is discarded otherwise.

4 Experimental results

In this section, we conduct a set of experiments on several publicly available datasets to evaluate the performance of LiPo-LCD⁺⁺. We also compare our solution with a representative set of state-of-the-art approaches. All experiments have been run on an Intel Core i7-9750H (2.60 GHz) processor with 16 GB RAM. In addition, an Nvidia GeForce GTX 980 GPU has been used to execute SuperPoint,² using a Python implementation on TensorFlow.

4.1 Methodology

We have selected 8 public datasets, which correspond to different environmental conditions: CityCentre (Cummins and Newman 2008) (CC), EuRoC Machine Hall 05 (Burri et al. 2016) (EuR5), KITTI 00 (Geiger et al. 2012) (K00), KITTI 05 (Geiger et al. 2012) (K05), KITTI 06 (Geiger et al. 2012) (K06), Lip6 Indoor (Angeli et al. 2008) (L6I), Lip6 Outdoor (Angeli et al. 2008) (L6O) and the Malaga 2009 Parking 6L (Blanco et al. 2009) (MLG). Table 1 provides a summary

² <https://github.com/rpautrat/SuperPoint>.

of these datasets. For each one, we use the ground truth provided by the original authors except for the KITTI sequences, for which we use the data referred by Arroyo et al. (2014), and the EuR5 and MLG datasets, for which we use the data available from Tsintotas et al. (2019).

Precision-recall (PR) metrics are used to evaluate the global performance of every approach. Given that the aim of an LCD method is to be used in a real SLAM solution, false positives are considered critical. For this reason, we are interested in the maximum recall at 100% precision.

Regarding LiPo-LCD⁺⁺ parameters, we have used either default values, or values which experimentally have resulted in good performance. OBIndex2 and dynamic islands have been configured as explained in Garcia-Fidalgo and Ortiz (2018), as well as in our previous work (Company-Corcoles et al. 2020). The maximum number of features for SuperPoint has been set to 1500. Line detection and description methods have been executed using the default parameters. The binarization of SuperPoint descriptors, apart from the desired size of the final descriptor, does not require any additional setup. Regarding the query-adaptive late fusion stage, the threshold for the segment slopes has been set to 0.025. We have also limited the maximum value of weights w_p and w_l to 0.8. Concerning the geometric spatial verification stage, 0.8 and 0.95 have been used as NNDR values for, respectively, points and lines. Line matches have been discarded if the ratio length μ_{ij} is above 2.5, and in case θ_{ij} deviates more than 30° from 0 or π . LPM runs using the default parameters provided by the authors.

4.2 Effectiveness of the binary descriptors

In this section, we assess the effectiveness of the binarization procedure. We empirically evaluate the three configurations for selecting pairs introduced in Sect. 3.1.1, denoted by, respectively, *All* (A), *Random* (R), and *Neighbors* (N). For each configuration, we consider the two possible versions of 32 and 64 pairs, resulting into binary descriptors of 256 and 512 bits, respectively. Additionally, we

Table 2 Maximum recall at 100% of precision for the 7 descriptors considered. Winners are indicated in bold face

	A-256	A-512	N-256	N-512	R-256	R-512	Q-256
L6I	83.72	90.23	89.62	94.34	89.86	90.70	81.27
EuR5	72.57	79.28	81.15	86.12	72.73	79.87	67.25

also consider a less sophisticated method that quantizes a 256-dimensional SuperPoint descriptor by thresholding each real-value component, hence resulting into a 256-bit descriptor. The threshold for each component is chosen as the average value of that component in the processed dataset. In the following, we denote this method as $Q-256$.

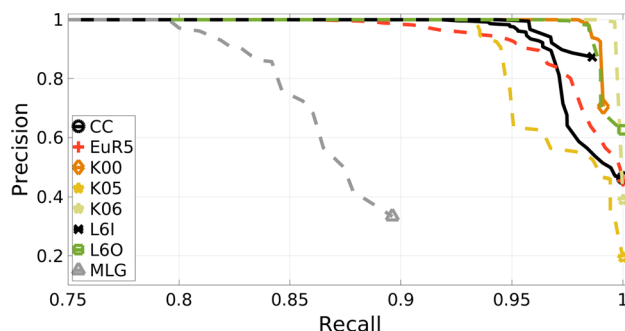
The L6I and the EuR5 datasets have been chosen for this experiment. L6I is a full indoor scenario which can be considered as a low-textured environment with high perceptual aliasing. Conversely, EuR5 is also a low-textured indoor environment which presents scenes under severe illumination and scale changes. To measure the effectiveness of each configuration, we compute the maximum recall that can be achieved by the whole system maintaining precision at 100%. Results are shown in Table 2. The highest recall has been obtained for the N-512 configuration for both datasets. As it is shown later, the average time difference required per frame using the N-256 and the N-512 configurations is approximately 7% in the EuR5 dataset. Taking into account both performance perspectives, we decide to adopt the configuration leading to best PR.

4.3 LCD performance breakdown

In this section, we evaluate the net effect of the proposed contributions to the final LCD performance. More precisely, we compute what is the maximum recall obtained by LiPo-LCD⁺⁺ at 100% of precision, but using different alternatives on each of the steps of our pipeline. Results are shown in Table 3. For the Feature Extraction (FE) stage, we consider ORB and our binarized SuperPoint descriptor (b-SP) as point descriptors, along with the LBD descriptor for lines. Regarding the Late Fusion (LF) options, we consider: (1) our previous Borda Count-based system (Company-Corcoles et al. 2020) and (2) the Query-Adaptive Late Fusion (QALF) method proposed in this work. Finally, RANSAC and the proposed LP-LPM are the choices studied for the Spatial Verification (SV) stage. The L6I dataset is selected for this experiment because it only contains indoor scenes and therefore it can be regarded as the most low-textured dataset considered in this work. As it can be observed, every contribution improves a specific part of the pipeline, which results into a slightly increase of the final recall. Overall, the new pipeline exhibits enhanced performance with regard to our previous contributions: 11.16% over the recall reported in Garcia-Fidalgo and Ortiz (2018), where only points were

Table 3 Maximum recall at 100% of precision for L6I. † and ◊ respectively refer to the solutions described in Garcia-Fidalgo and Ortiz (2018) and Company-Corcoles et al. (2020)

FE	LF	SV	Recall
ORB [†]	–	RANSAC	83.18
ORB + LBD [◊]	Borda Count	RANSAC	85.24
ORB + LBD	QALF	RANSAC	86.11
ORB + LBD	QALF	LP-LPM	90.61
b-SP	–	RANSAC	90.52
b-SP + LBD	Borda Count	RANSAC	90.81
b-SP + LBD	QALF	RANSAC	92.13
b-SP + LBD	QALF	LP-LPM	94.34

**Fig. 6** PR curves for each dataset. In all plots, P is 1 for R below 0.75

used, and 9.1% that reported in Company-Corcoles et al. (2020), which involved both point and line features.

4.4 General performance

This section focuses on the global performance of LiPo-LCD⁺⁺. We measure the accuracy of loop closure detection, and we report on the computational times as well as on the evolution of the vocabulary size.

4.4.1 LCD accuracy evaluation

Figure 6 reports on the accuracy of loop closure detection of LiPo-LCD⁺⁺ for all datasets. As usual, we employ PR curves, which result from modifying the threshold on the total number of inliers required to accept a loop. As can be observed, LiPo-LCD⁺⁺ exhibits a stable behaviour for all datasets, achieving high recall values while keeping precision at 100%.

Table 4 Average response time (ms) per image, calculated for each stage of the pipeline. These times were computed using the K00 dataset

	FE	VU	SC	SV
Points (P)	9.58	183.87	145.07	–
Lines (L)	17.60	22.60	18.69	–
P+L	18.98	201.32	161.45	24.77

4.4.2 Computational times

Average computational times are summarized in Table 4. These values have been computed from the K00 dataset, since it is the largest one considered in this work. Feature Extraction (FE) includes line and point detection and description, as well as the proposed binarization procedure for points using the N-512 configuration. On average, point FE requires about 9.58 ms per image, from which only 1.96 ms corresponds to the binarization of point descriptors. These times are remarkably low in contrast to the 457 ms required by, for instance, D2-Net (Dusmanu et al. 2019) and the 542 ms of R2D2 (Revaud et al. 2019), involving only feature detection and the generation of real-valued descriptors. The Vocabulary Update (VU) and Search for Candidates (SC) steps are usually slower for points than lines, due to the higher number of descriptors that have to be handled. Times required to merge candidate lists and select dynamic islands are included in SC because they are almost negligible. Finally, the Spatial Verification (SV) step only makes sense when both points and lines are jointly computed. The average response time per image of the whole pipeline turns out to be 406.52 ms in a multi-thread implementation, where points and lines are each processed within a separate thread.

The evolution of computational times as more images are processed is additionally illustrated in Fig. 7. We also use the K00 dataset for this assessment, for the same reason as above. As can be observed, times for VU and SC remain relatively stable over time (remember that our approach implements a dual incremental BoW scheme). SV times grow occasionally due to the amount of inliers required to validate these candidates within the LP-LPM scheme. If required for a specific application, the number of candidates can be reduced.

4.4.3 Evolution of the vocabulary size

As stated previously, LiPo-LCD⁺⁺ makes use of two vocabularies, one for point features and one for line features, that store visual words of, respectively, 512 and 256 dimensions. Figure 8 shows that (1) globally the number of visual words grows as more images have been processed, and (2) the amount of line features is considerably lower than the number of point features, both (1) and (2) as expected. (The

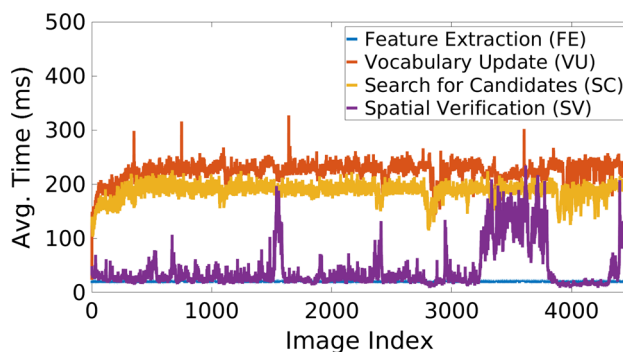


Fig. 7 Evolution of the computational times for each part of the pipeline using the K00 dataset

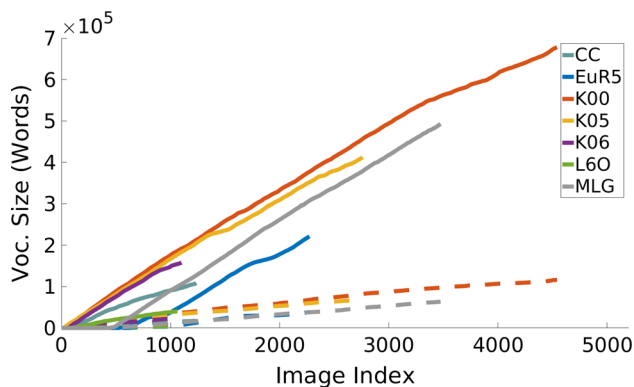


Fig. 8 Evolution of the vocabulary size for each feature. The amounts of point and line features are respectively denoted by continuous and dashed lines

L6I dataset is missing in the plot because of the significantly lower number of images it consists of.)

4.5 More on the computational requirements

In this section, we evaluate the effect of two strategies to save computational resources on the performance of LiPo-LCD⁺⁺, to contemplate the possibility of running it on computationally-restricted hardware (otherwise, one may be interested in achieving the highest performance in terms of recall). More precisely, among other alternatives (Zaffar et al. 2020), we first assess how reducing the map affects the performance of LiPo-LCD⁺⁺. To this end, we consider a simple method, denoted by Map Size Reduction (MSR), which, using the N-512 configuration as a basis, processes one out of every two images of the input dataset. A second strategy focuses on limiting the vocabulary size. For this purpose, we reduce the number of point features extracted from each image using the relevance measure available for SuperPoints; we consider 1000, 750, 500 and 250 features per image. Regarding the vocabulary for line features, we do not care for its size since it is negligible in contrast with that of point features. The results in terms of vocabulary size

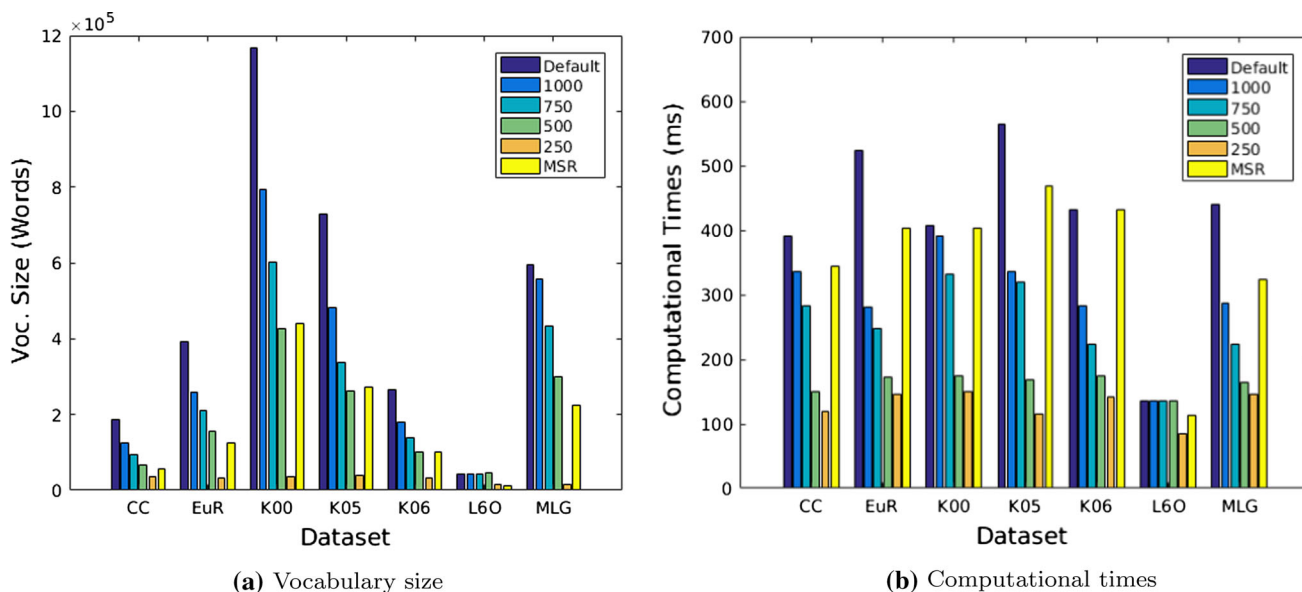


Fig. 9 Comparison of vocabulary size and running times for different strategies to reduce the computational requirements of LiPo-LCD⁺⁺. *Default* corresponds to the proposed approach using the N-512 descrip-

tor and 1500 point descriptors per image. *MSR* stands for Map Size Reduction, while the rest of cases refer to the maximum number of point descriptors requested per image

and computational times are detailed in, respectively, Fig. 9a and b. The corresponding recall values at 100% of precision for the different datasets can be found in Table 5.

As can be observed in Fig. 9a, the vocabulary size required by MSR is similar to the one resulting when using 500 point descriptors. However, the computational times reported in Fig. 9b demonstrate that MSR leads to similar running times than using 1000 or 1500 point descriptors per image.

Regarding the reduction of the map size reported in Table 5, it can be observed that the recall is dependent on the frame rate/camera speed, e.g. for datasets with higher frame rate or low camera speed, better performance is exhibited. On the other side, the recall tends to decrease when using less point features per image. However, despite this reduction, results for K00, K06 and L6O keep similar for all configurations. Another relevant observation is that using 750 or 1000 descriptors per image achieves similar performance than using the default configuration, i.e. 1500 descriptors, in most cases. Regarding L6O, the number of features that are detected is no more than 700, this is the reason why the recall is the same for the 750, 1000 and 1500-point features cases. As for L6I, no more than 250 features can be found, reason by which it is missing in Table 5.

4.6 Comparison with other solutions

In this section, we compare the performance of LiPo-LCD⁺⁺ against other state-of-the-art solutions. The maximum recall achieved at 100% of precision for each approach is summarized in Table 6. The reported results come from

Table 5 Maximum recall at 100% of precision for different configurations regarding computational complexity. *Default* corresponds to the proposed approach using the N-512 descriptor and 1500 point descriptors per image. *MSR* stands for Map Size Reduction, while the rest of cases refer to the maximum number of point descriptors requested per image

	CC	EuR5	K00	K05	K06	L6O	MLG
Default	92.22	86.12	98.08	93.68	99.62	96.81	79.96
1000	90.62	81.62	97.98	87.33	99.15	96.81	74.25
750	89.35	82.08	94.85	89.69	99.26	96.81	75.97
500	87.98	77.53	94.47	89.05	99.05	93.33	73.82
250	80.25	71.20	94.59	86.43	99.62	91.56	69.81
MSR	89.39	69.25	97.41	89.52	99.12	90.07	65.29

the original works, except for Gomez-Ojeda et al. (2019), which was executed by ourselves using the vocabularies and the default parameters provided by the authors, and Tsintotas et al. (2018); Tsintotas et al. (2019, 2021) for the L6I dataset. Non-available recall values are reported as n.a. Approaches that do not reach 100% precision for a specific dataset are indicated by a dash (–).

As can be observed, the combination of points and lines achieves the highest recall for almost all datasets. This is specially evident for the case of L6I and Eur5, which are regarded as the most low-textured scenarios considered in this work, which actually is the focus of our approach. Furthermore, the incorporation of lines in an LCD method not only improves the performance in low-textured scenarios, but

Table 6 Maximum recall at 100% precision for several off-line approaches (top), on-line approaches (middle) and LiPo-LCD⁺⁺ (bottom). The best and the second best scores are shown in bold and italic, respectively

	CC	EuR5	K00	K05	K06	L6I	L6O	MLG
Bampis et al. (2018)	71.14	23.10	96.53	97.30	80.10	52.22	58.32	87.56
Galvez-López and Tardos (2012)	31.61	n.a.	72.40	51.90	89.7	n.a.	n.a.	74.75
Mur-Artal and Tardós (2014)	43.03	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	81.51
Cummins and Newman (2011)	38.77	n.a.	49.20	48.50	55.34	n.a.	n.a.	68.52
Gomez-Ojeda et al. (2019)	n.a.	1.64	75.91	72.66	56.92	n.a.	n.a.	n.a.
Han et al. (2021)	33.45	n.a.	91.45	86.51	n.a.	n.a.	n.a.	72.51
Yue et al. (2019)	<i>91.00</i>	n.a.	97.00	93.00	98.00	n.a.	n.a.	90.00
Tsintotas et al. (2019)	20.00	83.70	97.50	92.60	<i>98.10</i>	–	50.00	85.00
Tsintotas et al. (2018)	16.30	69.20	93.20	94.20	86.00	–	12.00	<i>87.90</i>
Tsintotas et al. (2021)	36.00	<i>85.00</i>	97.70	<i>94.30</i>	<i>98.10</i>	4.09	78.00	<i>87.90</i>
Angeli et al. (2008)	n.a.	n.a.	n.a.	n.a.	n.a.	36.86	23.59	n.a.
Zhang et al. (2016)	41.18	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	82.60
Gehrig et al. (2017)	n.a.	71.00	93.10	94.00	n.a.	n.a.	n.a.	n.a.
Khan and Wollherr (2015)	38.92	n.a.	n.a.	n.a.	n.a.	41.74	25.58	78.13
Garcia-Fidalgo and Ortiz (2018)	88.25	76.28	76.50	51.86	95.53	83.18	85.24	61.23
Company-Corcoles et al. (2020)	89.30	81.94	<i>97.80</i>	91.73	97.38	<i>85.24</i>	97.31	75.73
LiPo-LCD ⁺⁺	92.22	86.12	98.08	93.68	99.62	94.34	<i>96.81</i>	79.96

also in the rest of datasets where line segments are noticeable. This can be observed when comparing either LiPo-LCD⁺⁺ or Company-Corcoles et al. (2020), which also combines point and line features, with the method proposed in Garcia-Fidalgo and Ortiz (2018), where only point features are used. Notice that LiPo-LCD⁺⁺ improves the performance attained by our previous solution (Company-Corcoles et al. 2020) for these low-textured datasets, and provides a competitive performance for the remaining datasets. Furthermore, LiPo-LCD⁺⁺ obtains better recall values almost for every dataset compared to the solution by Yue et al. (2019), which describes the images using a non-binarized version of the SuperPoint descriptor. Despite LiPo-LCD⁺⁺ achieves better recall values than our previous solution in the MLG dataset, the performance attained is rather low with regard to other more successful approaches. We have observed that this can be the result of wrong line detections between consecutive frames, which are due to the bad quality of the images and the corresponding perceptual aliasing that occurs among those lines, leading to a subsequent decrease in the pipeline performance. Notice that our proposal also outperforms (Gomez-Ojeda et al. 2019; Han et al. 2021), which are the solutions most similar to LiPo-LCD⁺⁺, given that they are the only ones that combine points and lines for loop closure detection.

To finish, in Table 7, we compare the vocabulary size of LiPo-LCD⁺⁺ with other state-of-the-art solutions in terms of memory consumption. As expected, the requirements for LiPo-LCD⁺⁺ are higher than the ones for a method that

only uses points (Garcia-Fidalgo and Ortiz 2018). This is obviously because of the two visual vocabularies, but also due to the fact that our binary point descriptor is 512-dimensional, which is larger than the ORB descriptor used by Garcia-Fidalgo and Ortiz (2018). Notice also that limiting the number of point features to 1000 - 250 we reduce the gap as for memory requirements with regard to Tsintotas et al. (2021) without compromising much our LCD performance (as reported in Table 5).

5 Conclusions and future work

This work has introduced LiPo-LCD⁺⁺, an appearance-based LCD approach that combines point and line features to increase the performance in low-textured environments. To search for loop candidates, our approach is based on two instances of an incremental BoW scheme specifically devised for binary descriptors, one instance for each feature. Besides, we develop a binarization procedure to incorporate SuperPoint into LiPo-LCD⁺⁺ as a feature point extractor, given the advantages of CNN-based methods recently reported. A query-adaptive late fusion approach is also adopted to merge lists of candidates obtained from the two visual perspectives. This fusion method is dynamically adapted to the current operating scenario, leveraging the final candidate scores according to the presence or absence of each kind of feature. Finally, we introduce a spatial verification method,

Table 7 Memory consumption (Mb) for several state-of-the-art approaches. Lowest values are shown in bold

	CC	EuR5	K00	K05	K06	MLG
Gehrig et al. (2017)	44.6	83.0	166.2	101.0	40.2	126.9
Bampis et al. (2018)	30.5	30.5	30.5	30.5	30.5	30.5
Tsintotas et al. (2021)	0.5	3.1	8.3	4.8	1.9	6.8
Garcia-Fidalgo and Ortiz (2018)	2.8	13.5	29.2	16.9	6.4	24.5
LiPo-LCD ⁺⁺ [Default]	11.4	23.9	71.22	44.4	16.2	36.2
LiPo-LCD ⁺⁺ [1000]	7.6	15.8	48.5	29.4	11.1	34.0
LiPo-LCD ⁺⁺ [750]	5.8	12.9	36.6	22.6	8.55	26.5
LiPo-LCD ⁺⁺ [500]	4.1	9.5	26.1	16.0	6.1	18.4
LiPo-LCD ⁺⁺ [250]	2.4	5.6	15.3	9.4	3.5	10.2
LiPo-LCD ⁺⁺ [MSR]	3.9	8.6	29.5	18.2	6.6	15.2

which jointly validates the appearance and the geometric consistency of points and lines together. LiPo-LCD⁺⁺ compares favourably with several state-of-the-art approaches for different public datasets and hence different environmental conditions.

Regarding future work, we plan to incorporate LiPo-LCD⁺⁺ into a SLAM framework and analyze its performance in low-textured scenarios. We also plan to explore the use of other geometric entities, e.g. planes, for loop closure detection.

Acknowledgements This work is partially supported by EU-H2020 projects BUGWRIGHT2 (GA 871260) and ROBINS (GA 779776), and by projects PGC2018-095709-B-C21 (funded by MCIU/AEI/10.13039/501100011033 and FEDER “Una manera de hacer Europa”), and PROCOE/4/2017 (Govern Balear, 50% P.O. FEDER 2014-2020 Illes Balears). This publication reflects only the authors views and the European Union is not liable for any use that may be made of the information contained therein.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Angeli, A., Filliat, D., Doncieux, S., & Meyer, J. A. (2008). A fast and incremental method for loop-closure detection using bags of visual words. *IEEE Transactions on Robotics*, 24(5), 1027–1037.
- Arroyo, R., Alcantarilla, P. F., Bergasa, L. M., Romera, E. (2016). Fusion and binarization of CNN features for robust topological localization across seasons. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4656–4663.
- Arroyo, R., Alcantarilla, P. F., Bergasa, L. M., Yebes, J. J., Bronte, S. (2014). Fast and effective visual place recognition using binary codes and disparity information. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3089–3094.
- Bampis, L., Amanatiadis, A., & Gasteratos, A. (2018). Fast loop-closure detection using visual-word-vectors from image sequences. *International Journal of Robotics Research*, 37(1), 62–82.
- Bay, H., Tuytelaars, T., Van Gool, L. (2006). Surf: Speeded up robust features. In: European Conference on Computer Vision, pp. 404–417.
- Bhowmik, N., González, R., Gouet-Brunet, V., Pedrini, H., Bloch, G. (2014). Efficient fusion of multidimensional descriptors for image retrieval. In: IEEE International Conference on Image Processing, pp. 5766–5770.
- Bian, J., Lin, W., Matsushita, Y., Yeung, S., Nguyen, T., Cheng, M. (2017). Gms: Grid-based motion statistics for fast, ultra-robust feature correspondence. In: International Conference on Computer Vision and Pattern Recognition, pp. 2828–2837.
- Blanco, J. L., Moreno, F. A., & Gonzalez, J. (2009). A collection of outdoor robotic datasets with centimeter-accuracy ground truth. *Autonomous Robots*, 27(4), 327.
- Burri, M., Nikolic, J., Gohl, P., Schneider, T., Rehder, J., Omari, S., Achtelek, M. W., & Siegwart, R. (2016). The EuRoC micro aerial vehicle datasets. *International Journal of Robotics Research*, 35(10), 1157–1163.
- Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., & Leonard, J. J. (2016). Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6), 1309–1332.
- Calonder, M., Lepetit, V., Strecha, C., Fua, P. (2010). Brief: Binary robust independent elementary features. In: European Conference on Computer Vision, pp. 778–792.
- Chen, Z., Jacobson, A., Sünderhauf, N., Upcroft, B., Liu, L., Shen, C., Reid, I., Milford, M. (2017). Deep learning features at scale for visual place recognition. In: IEEE International Conference on Robotics and Automation, pp. 3223–3230.
- Chen, Z., Lam, O., Jacobson, A., Milford, M. (2014). Convolutional neural network-based place recognition. In: Australasian Conference on Robotics and Automation, pp. 1–8.
- Company-Corcoles, J. P., Garcia-Fidalgo, E., Ortiz, A. (2020). LiPo-LCD: Combining lines and points for appearance-based loop closure detection. In: British Machine Vision Conference.
- Cummins, M., & Newman, P. (2008). FAB-MAP: Probabilistic localization and mapping in the space of appearance. *International Journal of Robotics Research*, 27(6), 647–665.
- Cummins, M., Newman, P. (2011). Appearance-only SLAM at large scale with FAB-MAP 2.0. *International Journal of Robotics Research* 30(9), 1100–1123.

- Dasgupta, S. (2013). Experiments with random projection. arXiv preprint [arXiv:1301.3849](https://arxiv.org/abs/1301.3849)
- DeTone, D., Malisiewicz, T., Rabinovich, A. (2018). Superpoint: Self-supervised interest point detection and description. In: International Conference on Computer Vision and Pattern Recognition, pp. 224–236.
- Dusmanu, M., Rocco, I., Pajdla, T., Pollefeys, M., Sivic, J., Torii, A., Sattler, T. (2019). D2-net: A trainable CNN for joint description and detection of local features. In: International Conference on Computer Vision and Pattern Recognition, pp. 8092–8101.
- Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), 381–395.
- Galvez-López, D., & Tardos, J. D. (2012). Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5), 1188–1197.
- García-Fidalgo, E., & Ortiz, A. (2015). Vision-based topological mapping and localization methods: a survey. *Robotics and Autonomous Systems*, 64, 1–20.
- García-Fidalgo, E., & Ortiz, A. (2018). iBoW-LCD: An appearance-based loop-closure detection approach using incremental bags of binary words. *IEEE Robotics and Automation Letters*, 3(4), 3051–3057.
- Garg, S., Milford, M. (2020). Fast, compact and highly scalable visual place recognition through sequence-based matching of overloaded representations. In: IEEE International Conference on Robotics and Automation, pp. 3341–3348.
- Garg, S., Milford, M. (2021). Seqnet: Learning descriptors for sequence-based hierarchical place recognition. *IEEE Robotics and Automation Letters*, 4305–4312.
- Gehrig, M., Stumm, E., Hinzmann, T., Siegwart, R. (2017). Visual place recognition with probabilistic voting. In: IEEE International Conference on Robotics and Automation, pp. 3192–3199.
- Geiger, A., Lenz, P., Urtasun, R. (2012). Are we ready for autonomous driving? The KITTI vision benchmark suite. In: International Conference on Computer Vision and Pattern Recognition, pp. 3354–3361.
- Gomez-Ojeda, R., Moreno, F., Zuñiga-Noël, D., Scaramuzza, D., & Gonzalez-Jimenez, J. (2019). PL-SLAM: A stereo SLAM system through the combination of points and line segments. *IEEE Transactions on Robotics*, 35(3), 734–746.
- Grompone von Gioi, R., Jakubowicz, J., Morel, J., Randall, G. (2010). LSD: A fast line segment detector with a false detection control. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32(4), 722–732.
- Han, J., Dong, R., Kan, J. (2021). A novel loop closure detection method with the combination of points and lines based on information entropy. *Journal of Field Robotics*, 38, 386–401.
- Hausler, S., Milford, M. (2020). Hierarchical multi-process fusion for visual place recognition. arXiv preprint [arXiv:2002.03895](https://arxiv.org/abs/2002.03895)
- Kenshimov, C., Bampis, L., Amirgaliyev, B., Arslanov, M., & Gasteratos, A. (2017). Deep learning features exception for cross-season visual place recognition. *Pattern Recognition Letters*, 100, 124–130.
- Khan, S., Wollherr, D. (2015). IBuLLD: Incremental bag of binary words for appearance based loop closure detection. In: IEEE International Conference on Robotics and Automation, pp. 5441–5447.
- Labbé, M., & Michaud, F. (2013). Appearance-based loop closure detection for online large-scale and long-term operation. *IEEE Transactions on Robotics*, 29(3), 734–745.
- Lin, K., Lu, J., Chen, C., Zhou, J. (2016). Learning compact binary descriptors with unsupervised deep neural networks. In: International Conference on Computer Vision and Pattern Recognition, pp. 1183–1192.
- Lin, K., Lu, J., Chen, C., Zhou, J., & Sun, M. (2019). Unsupervised deep learning of compact binary descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(6), 1501–1514.
- Lopez-Antequera, M., Gomez-Ojeda, R., Petkov, N., & Gonzalez-Jimenez, J. (2017). Appearance-invariant place recognition by discriminatively training a convolutional neural network. *Pattern Recognition Letters*, 92, 89–95.
- Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2), 91–110.
- Lowry, S., Andreasson, H. (2018). Logos: Local geometric support for high-outlier spatial verification. In: IEEE International Conference on Robotics and Automation, pp. 7262–7269.
- Lowry, S., Sünderhauf, N., Newman, P., Leonard, J. J., Cox, D., Corke, P., & Milford, M. J. (2016). Visual place recognition: A survey. *IEEE Transactions on Robotics*, 32(1), 1–19.
- Ma, J., Zhao, J., Jiang, J., Zhou, H., & Guo, X. (2019). Locality preserving matching. *International Journal of Computer Vision*, 127(5), 512–531.
- Milford, M. J., Wyeth, G. F. (2012). SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights. In: IEEE International Conference on Robotics and Automation, pp. 1643–1649.
- Mur-Artal, R., Tardós, J. D. (2014). Fast relocalisation and loop closing in keyframe-based SLAM. In: IEEE International Conference on Robotics and Automation, pp. 846–853.
- Mur-Artal, R., & Tardós, J. D. (2017). ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras. *IEEE Transactions on Robotics*, 33(5), 1255–1262.
- Neubert, P., Schubert, S., Protzel, P. (2019). A neurologically inspired sequence processing model for mobile robot place recognition. *IEEE Robotics and Automation Letters* 3200–3207.
- Nister, D., Stewenius, H. (2006). Scalable recognition with a vocabulary tree. In: International Conference on Computer Vision and Pattern Recognition, Vol. 2, pp. 2161–2168.
- Pumarola, A., Vakhitov, A., Agudo, A., Sanfeliu, A., Moreno-Noguer, F. (2017). PL-SLAM: real-time monocular visual SLAM with points and lines. In: IEEE International Conference on Robotics and Automation, pp. 4503–4508
- Revaud, J., Weinzaepfel, P., De Souza, C., Pion, N., Csürka, G., Cabon, Y., Humenberger, M. (2019). R2d2: Repeatable and reliable detector and descriptor. arXiv preprint [arXiv:1906.06195](https://arxiv.org/abs/1906.06195)
- Rublee, E., Rabaud, V., Konolige, K., Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. In: International Conference on Computer Vision, pp. 2564–2571.
- Sivic, Z. (2003). Video google: A text retrieval approach to object matching in videos. In: International Conference on Computer Vision, Vol. 2, 1470–1477.
- Stewart, B., Ko, J., Fox, D., Konolige, K. (2002). The revisiting problem in mobile robot map building: A hierarchical bayesian approach. In: Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence, p. 551–558.
- Sünderhauf, N., Protzel, P. (2011). BRIEF-Gist—closing the loop by simple means. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1234–1241.
- Sünderhauf, N., Shirazi, S., Dayoub, F., Upcroft, B., Milford, M. (2015). On the performance of ConvNet features for place recognition. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4297–4304.
- Tsintotas, K. A., Bampis, L., Gasteratos, A. (2018). Assigning visual words to places for loop closure detection. In: IEEE International Conference on Robotics and Automation, pp. 5979–5985
- Tsintotas, K. A., Bampis, L., & Gasteratos, A. (2019). Probabilistic appearance-based place recognition through bag of tracked words. *IEEE Robotics and Automation Letters*, 4(2), 1737–1744.

- Tsintotas, K. A., Bampis, L., & Gasteratos, A. (2021). Modest-vocabulary loop-closure detection with incremental bag of tracked words. *Robotics and Autonomous Systems*, 141, 103782.
- Williams, B., Klein, G., & Reid, I. (2011). Automatic relocalization and loop closing for real-time monocular slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9), 1699–1712.
- Yefeng, Z., & Doermann, D. (2006). Robust point matching for non-rigid shapes by preserving local neighborhood structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4), 643–649.
- Yue, H., Miao, J., Yu, Y., Chen, W., Wen, C. (2019). Robust loop closure detection based on bag of superpoints and graph verification. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3787–3793.
- Zaffar, M., Ehsan, S., Milford, M., Flynn, D., McDonald-Maier, K. (2020). VPR-Bench: An open-source visual place recognition evaluation framework with quantifiable viewpoint and appearance change. arXiv preprint [arXiv:2005.08135](https://arxiv.org/abs/2005.08135)
- Zhang, F., Rui, T., Yang, C., & Shi, J. (2019). Lap-SLAM: A line-assisted point-based monocular VSLAM. *Electronics*, 8(2), 243.
- Zhang, G., Lilly, M.J., Vela, P. A. (2016). Learning binary features online from motion dynamics for incremental loop-closure detection and place recognition. In: *IEEE International Conference on Robotics and Automation*, pp. 765–772.
- Zhang, L., & Koch, R. (2013). An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency. *Journal of Visual Communication and Image Representation*, 24(7), 794–805.
- Zheng, L., Wang, S., Tian, L., Fei, H., Liu, Z., Tian, Q. (2015). Query-adaptive late fusion for image search and person re-identification. In: *International Conference on Computer Vision and Pattern Recognition*, pp. 1741–1750.
- Zuo, X., Xie, X., Liu, Y., Huang, G. (2017). Robust visual SLAM with point and line features. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1775–1782.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Joan Pep Company-Corcoles is currently pursuing a Ph.D. degree at the University of the Balearic Islands (UIB). He received the B.Sc. and the M.Sc. degree from the Polytechnic University of Catalonia (UPC). His current research interests include Simultaneous Localization And Mapping (SLAM), place recognition, computer vision and unmanned aerial vehicles (UAVs).



Emilio Garcia-Fidalgo is a post-doctoral researcher at the Department of Mathematics and Computer Science, University of the Balearic Islands (UIB). He holds B.Sc., M.Sc. and Ph.D. degrees in Computer Science. His current research interests include localization and mapping, visual place recognition, computer vision, machine learning and unmanned aerial vehicles (UAVs).



Alberto Ortiz is Associate Professor at the Department of Mathematics and Computer Science of the University of the Balearic Islands (UIB). He holds B.Sc. and Ph.D. degrees in Computer Science. He is author and co-author of more than 160 publications related with computer vision, machine learning and mobile robotics. His current research interests are machine learning (deep and shallow) and its applications, motion estimation, localization and mapping, visual guidance of mobile robots, including obstacle detection and avoidance, and control architectures for mobile robots.