Global Image Signature for Visual Loop-Closure Detection

Pep Lluis Negre Carrasco Francisco Bonin-Font Gabriel Oliver-Codina

July 13, 2015

Abstract

This work details a new method for loop-closure detection based on using multiple orthogonal projections to generate a global signature for each image of a video sequence. The new multi-projection function permits the detection of images corresponding to the same scene, but taken from different point of views. The signature generation process preserves enough information for robust loop-closure detection, although it transforms each image in a simple and compact representation. Thanks to these characteristics, a real-time operation is possible, even for long sequences with thousands of images. In addition, it has proved to work on very different scenarios without the need either to change the parameters or to perform an onffline training stage, which makes it very independent on the environment and camera configuration.

Results of an extensive set of experiments of the algorithm on several datasets, both indoors and outdoors and including underwater scenarios, are presented. Furthermore, an implementation, named HALOC, is available at a public repository[1] as a C++ library for its use under the BSD license.

1 Introduction

In the context of autonomous mobile robotics, an accurate estimation of the robot pose is essential to succeed in the programed missions. This problem is known as *localization* and it has been addressed from different perspectives, ranging from simple *dead reckoning* to methods based on complex representations of the environment. Traditionally, localization approaches have relied on range sensors, like laser on terrestrial environments or sonar in underwater media. Nevertheless, modern cameras offer higher spatial and temporal resolutions, reduced size and consumption at very competitive prices, thus their application in robotics is extensively used nowadays.

Loop-closure detection is a crucial issue in visual self-localization techniques, such as SLAM (*Simultaneous Localization And Mapping*) or topological mapping and localization approaches. This ability consists in detecting when the robot is returning to a previously visited area. Detected loop closings impose additional pose constraints which are essential for the accurate correction of the robot global pose. In vision-based systems, this loop-closure detection is performed by registering images currently grabbed, with other frames gathered previously. In long trajectories, this procedure can be highly costly when the query image is compared with all the previous ones. A way to reduce effectively the time for the loop-closure detection consists in running the image registration only over a set of image candidates that are most likely to overlap with the query. As the length of the image sequence grows, the candidate selection process time can grow boundless, preventing the method to be applied online on long trajectories [45]. Thus, it is essential to use a fast and efficient technique to select the candidate images from the grabbed sequence.

In this paper we introduce a fast method to accurately retrieve candidate images for loop-closure detection. The proposed algorithm uses a multi-projection function to convert every image of a video sequence into a single signature by using original image features. Moreover, the reduced size of this signature allows the execution of the loop-closure detection in an online system, even for long sequences.

[1] See `https://github.com/srv/libhaloc`

2 Related Work

Many visual SLAM approaches use advanced data association techniques [5] to match visual features of the last acquired image with those features stored in the map. This type of techniques are slow and limit the features used for loop-closure detection to those included in the map. Moreover, many systems using this scheme are not designed to work in real time when they are applied to long trajectories [44]. Other techniques based on topological maps are also contributing with highly accurate results. Garcia-Fidalgo and Ortiz present in [13] an extensive collection of methods and approaches to detect loop closures, in the context of topological mapping. Some of these approaches rely on image-to-image voting methods which implement a feature matching process over all the stored images and usually have an expensive computation time [20] [39]. The approach given in [12] extends the state of the art of appearance-based topological mapping methods and relies on the detection of loop closures by a process of indexing invariant features.

Bag-of-words (BoW) methods are extensively used to perform global localization and loop-closure detection in an image classification scheme. In this context, images are represented as vectors that account for the number of occurrences of local image features taken from a dictionary. This dictionary is performed by clustering similar keypoints [7] [6]. Despite there are recent techniques for building this vocabulary online [11] [1], most of the implementations present on the literature need a previous training stage to construct the dictionary which will be used subsequently in the localization process. The creation of this dictionary is the major weakness of most BoW approaches since it introduces the appearance assumption of the environment and, therefore, the navigation using different cameras or in a totally different environment becomes inefficient. On the other hand, one of the major advantage of BoW is the capacity to obtain a solution for loop closing that is independent of the map size in terms of computational time. In contrast, the literature widely reports that most of the BoW-based approaches introduce perceptual aliasing [1] [34]. Perceptual aliasing is the problem of having locations that might be perceived as the same but they should not (such as two visually equal cars -same manufacturer, model and color- present in different street locations). This problem is a serious weakness for a localization system that corrects its pose according to the loop closures since the new computed positions would be totally inconsistent. Other authors have incorporated the depth information (when working with RGB-D cameras or laser) in the SIFT extracted descriptors to enhance their distinctiveness properties. These enhanced descriptors obtained by fusing color and depth information are applied mostly in human gesture identification and activity recognition and they have revealed promising results when combined with bag-of-words [42] [43] [19].

Locality Sensitive Hashing (LSH) have been recently used by Shahbazi and Zhang [34] to transform the image descriptors into hash tables. Then, for every query image, its descriptors are hashed and compared with the existing ones into the tables by applying a distance ratio [25]. Images with a larger number of matches are treated as candidates for closing a loop. Shahbazi and Zhang also demonstrate that their algorithm has an almost linear running time with a constant term that they affirm to be small. Moreover, they compare the proposed algorithm with the state of the art of BoW to carry out a quantitative comparison which demonstrates that BoW is faster but less accurate than LSH (which is not fast enough to run online at 5Hz). The most important contribution of [34] is the use of a hash function directly over the image descriptors instead of using image histograms [15] or textures [21] to retrieve the candidates to close a loop. However, the LSH algorithm used by [34] has a high computational cost because a family of hash functions is applied on each descriptor separately.

Other techniques for loop closing detection in localization applications are based on image global descriptors. These descriptors usually represent the image by small vectors (i.e. 20 bytes in [18]), simplifying the image matching process in terms of time and computational resources. However, these simpler techniques sometimes compromise seriously the success ratios. For example, Liu and Siegwart proposed in [22] [23] a new unique descriptor to characterize a whole image, based on the average of the U-V color space values of the pixels enclosed in different areas of the image, delimited by vertical edges. This approach shows excellent results only when imaging indoor environments with omnidirectional cameras, and it uses color-based global descriptors, which are always more sensitive in different scenes with sim-

ilar colors. GIST [30] is another outstanding and well known global descriptor based on a set of perceptual and spatial structure of the scenes. The characterization of images is based on the diverse spectral properties of the scene (Energy Spectrum, Fourier Transforms, Windowed Fourier Transforms, Discriminant Spectral Templates -DST-, Windowed Discriminant Spectral Template -WDST-, etc.) combined properly and approximated as a set of Gabor filters. Gabor-GIST global descriptors have been used basically for scene categorization, and very occasionally for loop closure detection. Liu and Zhang presented in [24] a new generic framework to detect loop closings that incorporates reduced Gabor-GIST global descriptors. This framework is integrated in a particle filter to perform SLAM that exploits the temporal coherence in image sequences. BRIEF-Gist [38] is a simple global descriptor which consists in generating a BRIEF descriptor [4] on a downsampled (e.g 60×60 pixels) version of the original image. Another version of BRIEF-Gist concatenates, in a single binary vector, the BRIEF descriptor of each of the m $\times$ m tiles in which the original image is partitioned. The calculation and comparison of the BRIEF-Gist descriptors is extremely fast, and the exposed experiments show a precision of 100% in the detection of loop closings. However, there are some relevant restrictions when detecting loop closings using BRIEF-Gist, related to the point of view from which scenes that close a loop are re-viewed.

Another global descriptor that has shown very high performance in the similarity image search in large databases is VLAD [18]. This image global representation is obtained in two steps: first, the image descriptors are aggregated to a vector, based on their proximity to the center of any of the clusters that sectors the feature space. Second, a PCA projection is performed to reduce the dimensionality of the vector, resulting in a compact global image descriptor that can be used for fast image comparison. Recent studies [2] demonstrate that VLAD outperforms other popular global descriptors, such as GIST, in terms of accuracy, when applied on public image retrieval benchmarks. Moreover, as bag-of-words, VLAD requires a visual dictionary computed by, for example, K-means clustering in an offline training stage, causing the same disadvantages discussed above. In general, most of these global descriptors have demonstrated to be relatively robust in image recognition, scene categorization and, in some occasions, for loop closure detection, but none of them, to the best of the authors knowledge, have been tested in complex scenarios, such as underwater environments with no clear dominant spatial structures or semantic categories, and with extremely textured areas.

Our proposal is framed in the global descriptors context but, unlike the methods listed above, we apply a multi-projection function directly to the entire array of descriptors providing the following advantages:

1. Two images with similar descriptor matrices (i.e. having descriptors matching) will result in similar signatures due to the projection properties.

2. The query image and the obtained candidates to close a loop mostly have significant overlap, since the signature represents the whole descriptor matrix.

In general terms, the main contributions of this paper are:

1. We present an algorithm for fast loop-closure detection based on a global image signature. This approach reduces the computational time dedicated to search for image candidates to close a loop with a query, with respect to other approaches.

2. A wide set of experiments show how the method proposed here outperforms in scene recognition other important works based on the main techiques: BoW-based ([16], [1]), LSH-based ([34]) and global descriptors based [18]. Moreover, our method does not require a training stage that can negatively affect the results in the normal execution of the localization process.

3. The presented architecture has been tested in terrestrial (indoor and outdoor) and underwater datasets to demonstrate the non-dependence on the environment type and the parameter set. All experiments were performed using the default parameters provided in the public implementation, without any change or adjustment. Notice that, underwater is one of the most challenging environments for detecting loop closures due to the repeated patterns and textures. Is in these particular scenarios where our approach outdoes others extendedly used nowadays.

The rest of the paper is structured as follows: Section 3 outlines the procedure for calculating the new image signature; the whole loop-closing algorithm including the search for candidates and the subsequent validation stage is detailed in Section 4; experimental results are given in Section 5 and Section 6 is dedicated to the conclusions and future work.

3 Signature Generation

Generating a signature that describes the image information (also known as global image descriptor) can also be seen as a mathematical hash function, also known as media hash [32]. The principal purpose of a hash function is to extract a fixed-length string or bit vector from a large data structure [31] [28] [32] (a text message, a document file or an image, for example). Hash functions perform many-to-one mappings and can be defined as:

$$h = \Psi(\rho) \tag{1}$$

where Ψ is the hash function, ρ is the input data to be hashed and h is the computed hash string.

A key feature of conventional hashing algorithms, such as MD5 and SHA-1, is that they are extremely sensitive to small perturbations in the original input, causing large differences in the resulting hash. However, most of the visual applications (scene recognition or image categorization, for instance), need that two images with high perceptual similarities have resembling signatures. Some studies [41] [26] investigate how to prevent significant changes on image signatures that have suffered controlled modifications (p.e. rotation from 2 to 5 degrees, cropping from 10% to 20% of image area, etc.).

Signature generation functions can be roughly classified in two main groups depending on the input data used to generate the global descriptor:

1. **Appearance-based**: the signature is calculated from texture, color information, transformations in the frequency space or matrix factorizations [41] [21] [36] [27].

2. **Feature-based**: the signature is calculated from the image keypoints and their descriptors [34] [18] [32] [28].

Our work is framed in the feature-based group and aims to create a unique signature for each set of features that characterize an image. With this approach we want to avoid the perceptual aliasing problem present in clustering algorithms (i.e. BoW) that use the same visual words for different visual features. The objective is that computed signatures result similar for images that represent the same scene (i.e have matching features), but grabbed from different points of view.

3.1 Descriptor Matrix Projection

Let us define the image *descriptor matrix* $D_{n\times m}$, where n is the number of visual keypoints (features) found in the current image and m is the length of their corresponding descriptors. The signature function proposed in this work is based on the projection of each column of $D_{n\times m}$ onto a set of k directions defined by unit vectors of n dimensions. For a certain image set, m will be always the same (i.e. determined by the descriptor type), while n will vary for each image in the sequence. The aim of this process is to transform the variable descriptor space size into a fixed size: $\mathbb{R}^{n\times m} \rightarrow \mathbb{R}^{1\times k\cdot m}$. The value of k has to provide an acceptable trade-off between low size and high performance. Details of k setting are exposed at the end of this section.

The projection of the descriptor matrix onto a single ($k = 1$) unit vector $\widehat{u}$ of random direction is defined as:

$$\rho_i = \sum_{j=1}^{n} D(j,i) \cdot \widehat{u}(j) \tag{2}$$

where $1 \leq i \leq m$. Hence, the final sequence that forms the signature h will be:

$$h = \rho_1 \oplus \rho_2 \oplus ... \oplus \rho_m \tag{3}$$

where $\oplus$ is the concatenation operator.

Projections have already been used as hash functions in [15] and the advantages of introducing a random factor in the generation of the image signature has been proven in [37] and [26], among others.

The rows of the descriptor matrix corresponding to overlapping images that are rotated, scaled or translated (i.e. a possible loop-closure), can be unsorted or the size n can differ considerably. In principle and theoretically,

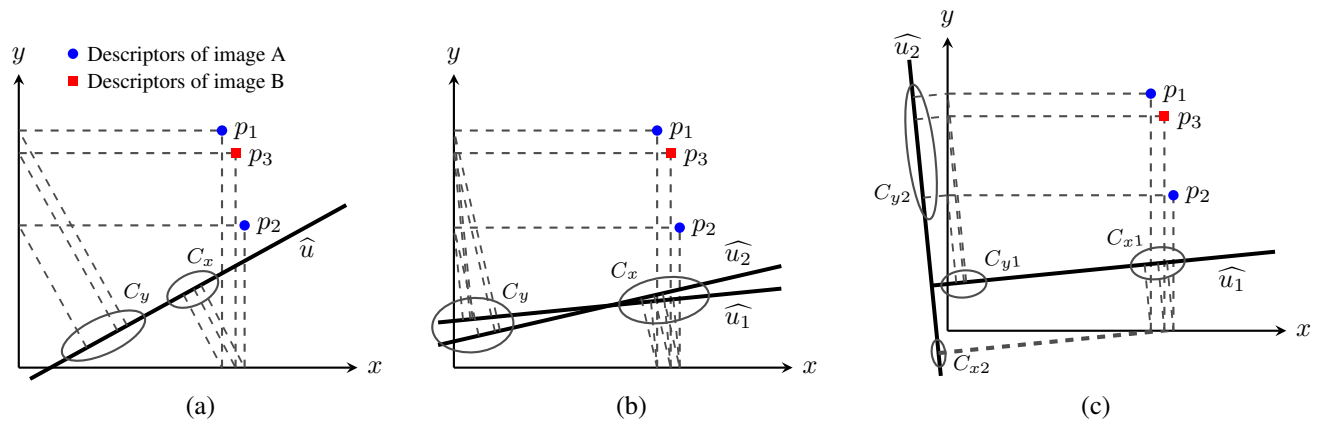


Figure 1: Ideal case (a) where similarities and differences in the descriptor space are translated to the projected points. Random vector projections (b) and orthogonal projections (c).

this might lead to distant values in equation 2 for both images, when they should be similar. To solve these problems, first the bucketing mechanism [14] is applied to keep a constant number of keypoints n in every image and spread them uniformly over the image domain. Secondly, the vectors used in the projections ($\widehat{u}$) are unitary. This circumstance causes the signature computed from the descriptors matrix (D) to be very similar to the case in which the columns of D were projected onto a vector with a constant value in each of its components (as shown in section 5.1), which would make the signature invariant to the order of the features in D. However, the $\widehat{u}$ component values need to be different enough so that descriptor changes are reflected in the projections. In consequence, when using a projective unit vector and a descriptor invariant to image rotation, scale and translation changes, this new approach has a considerable tolerance to these effects as it has been widely observed empirically, and shown in section 5.1 .

The direction of the unit vector $\widehat{u}$ is defined at the beginning of the process and must be the same to compute the signature of all the images of a certain sequence or trajectory. Moreover, the length of $\widehat{u}$ must be adaptable to the number of rows n of the descriptor matrix corresponding to each image of the sequence. The maximum value of n is unknown a priori, but the dimension of $\widehat{u}$ can be set large enough. Experiments demonstrated that a 5% of the image size in pixels is an appropriate value.

Figure 1-(a) illustrates the effects of projecting the descriptor components of two different images, A and B, onto a unit vector of random direction. For an easy understanding, let us assume a two-dimensional orthogonal space ($m = 2$), i.e. each descriptor has two components: x and y. Features of image A are plotted with blue circles and features of image B with red squares. In plot 1-(a), the projections of the x-components onto $\widehat{u}$ (C_x in the figure) are not useful for discriminating the descriptor p_2 from p_3. This is because the distance between p_2 and p_3 is not preserved on the x-projection. However, the projections of the y-components onto $\widehat{u}$ (C_y in the figure) perfectly reflect the differences of the descriptors space into the projected values. Plot 1-(b) illustrates a case where none of the two possible random directions of the unit vector ($\widehat{u_1}$ or $\widehat{u_2}$) is able transfer the differences of the descriptor space into the projected components. In these cases, significant differences in the x and y components of the descriptors are nearly negligible in the projected values. It is obvious that a single projection may not be enough to discriminate the descriptors of both images.

Next section offers a suitable solution to this problem by using more than one projection.

3.2 Multiple Orthogonal Projections

In order to reinforce the performance, the descriptor matrix is projected on several directions, and the different

projections are concatenated to form the final signature.

Let us define h_l as the result of projecting the *descriptor matrix* onto the l^{th} direction $\widehat{u}_l$. Then, equation 3 can be rewritten as:

$$h_l = \bigoplus_{i=1}^{m} \left(\sum_{j=1}^{n} D(j,i) \cdot \widehat{u}_l(j) \right) \quad (4)$$

and the final signature is defined as the concatenation of the k projections on all the different directions $\widehat{u}_l$:

$$H = \bigoplus_{l=1}^{k} h_l \quad (5)$$

The set of vectors $\widehat{u}_l$ ($1 \leq l \leq k$) must be defined offline, previously to the global localization process. If the k unit vectors were randomly generated, some of them could result in a similar direction, providing a repetitive and useless information and causing inefficiency in the hashing process. This is, for example, the case shown in figure 1-(b).

A way to generate an efficient and distinctive set of vectors $\widehat{u}$ is to calculate their directions so that they are mutually orthogonal [15]: $((\widehat{u}_1 \perp \widehat{u}_2) \perp \widehat{u}_3) \perp ...\widehat{u}_k$.

See the example of figure 1-(c) where the directions of $\widehat{u}_1$ and $\widehat{u}_2$ are orthogonal. Projections on direction $\widehat{u}_1$ are not distinctive enough, but projections on $\widehat{u}_2$ (mostly the y-component, C_{y2} in the figure) clearly differentiate the features p_2 and p_3. For a signature composed of k projections, the first unit vector $\widehat{u}_1$ is generated randomly, and the successive $k-1$ unit vectors are computed to be orthogonal between them by forcing their inner product equal to zero. E.g. if $k = 2$:

$$\widehat{u}_1(1) \cdot \widehat{u}_2(1) + \widehat{u}_1(2) \cdot \widehat{u}_2(2) + ... + \widehat{u}_1(n) \cdot \widehat{u}_2(n) = 0 \quad (6)$$

Since $\widehat{u}_1$ is known, it is possible to give random values to $\widehat{u}_2(1), \widehat{u}_2(2), ..., \widehat{u}_2(n-1)$ and then solve for the last index, $\widehat{u}_2(n)$, to make $\widehat{u}_1 \perp \widehat{u}_2$. For $k > 2$ a recursive solution in the form of a linear matrix equation $Ax = B$ can be defined forcing each vector to be orthogonal to the others using any of the existing decomposition procedures such as QR (*Orthogonal-Triangular Matrix*) factorization or SVD (*Singular Value Decomposition*) among others.

Preliminary experiments demonstrate that selecting $k = 3$ provides a good trade-off between signature size and accuracy in the detection of loop closures. So, from now on, all the analysis and experiments are performed with this value.

The current implementation of the signature calculation supports integer (p.e. SIFT [25]) and float (p.e. SURF [3]) descriptors.

As will be shown later, the present global image descriptor method for loop-closure detection outperforms the previous related approaches in two main aspects, reliability and computation time:

- First, scene recognition techniques supported on the comparison of color histogram or texture-based hashes [15] [41] introduce the problem of perceptual aliasing, where different scenes are wrongly labeled to represent the same.

  Although experimental results of the approach described in [1] show a certain robustness to the perceptual aliasing, this problem is commented and treated extendedly in this reference as one of the main challenges when using BoW. In [9], the perceptual aliasing is overcome to a certain extent, using the techniques proposed in the approach. Scene features of different places with high perceptual aliasing can be labeled with a low probability to close a loop, but, still, if they really close a loop, the results will be the same. According to [1] [9] (among many others), this is due to the use of a pre-computed training set of visual words which could have no relation with the current trajectory. Instead, our multi-projection function is applied directly over the image descriptors and no precomputed or training information is assumed. This reduces considerably the perceptual aliasing problem, since descriptors and their comparisons rely uniquely on each particular scene of the current trajectory, increasing the reliability of the system.

- Second, the multi-projection function proposed transforms the unknown size (n) of the descriptor matrix $D_{n \times m}$ to a fixed-length signature of size $W = k \cdot m$. This implies that the computation time will be bounded and considerably reduced. Contrarily, [34] applies a hash function over every particular feature obtaining a large set of hash indices for each image, increasing its complexity and, thus, causing

higher computing time.

4 Loop Closing Algorithm

The main purpose of this work is to build a fast algorithm applicable online that allows to find possible loop closures between the last received image and all the images acquired previously during a robot mission. Our efforts are focused on speeding up this process as much as possible, while maintaining a high reliability.

The process of image signature generation exposed in the previous section is used to search and retrieve the best candidates to close a loop with the most recently grabbed image. Afterwards, successive stages of image similarity validation are performed in order to accept or reject these pre-selected candidates. The proposed algorithm can be divided in 2 main stages:

1. **Similarity Image Search**. For every new image in the trajectory, compute its signature and store it into a table. The latest received image is treated as a query and compared with all the existing records (except itself and its neighbors) of the table in order to find the best candidates to close loop with it. The best candidates will be those with similar signatures. Finally, a Bayes filter is applied to improve the search for candidates by using the past trajectory information.

2. **Loop-Closure Validation**. For every candidate to close a loop obtained in the previous stage, its feature descriptors are matched with the feature descriptors of the query image using the classic Knn-Match (*K-nearest neighbor Match*). The existence of a considerable number of matchings reinforces the assumed image coincidence. Then, the epipolar geometry is imposed between the candidates and the query, and outliers are discarded applying RANSAC. If the epipolar condition is consistent, the loop-closure is confirmed.

The whole loop-closing process is illustrated in figure 2 and all the steps are detailed in the next sections.

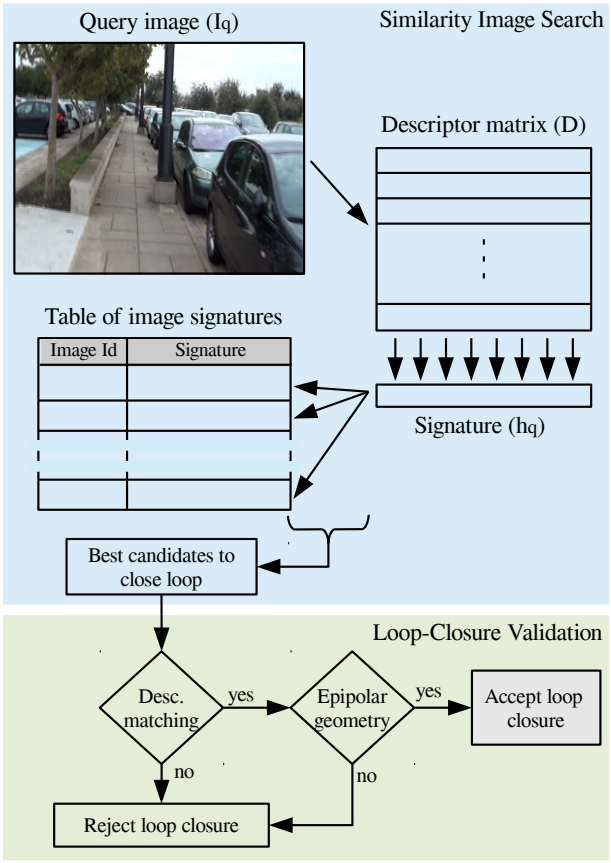


Figure 2: Overall loop-closing process diagram.

4.1 Similarity Image Search

For every new image in the trajectory, the algorithm extracts its descriptors and then applies eq. 5 to calculate its signature. This signature is saved alongside its index into a table to be a candidate for future queries. Moreover, descriptors of every image are also stored into another database to be accessible for a validation of a possible loop-closure. The frequency of accesses to this database is much lower than the number for the table of image signatures, thus it is implemented in a second priority memory level.

Every new signature is treated as a query (h_q) and compared with the signatures previously stored into the table with the purpose of finding possible candidates to close loops with it. Signatures for the corresponding s-nearest precedent neighbors of the query image are not considered as candidates to avoid closing loops between consecutive images, so h_q is only compared to $\{h_1, h_2...h_{q-s}\}$. The s parameter is adjusted depending on the frame rate and the velocity of the camera motion.

The comparison of image signatures is performed by applying the l_1-norm between the query and the candidate: $\|h_q - h_c\|_1$ where h_c is the candidate signature. The reason of using l_1 instead of, for example, l_2 is that, the experiments performed on the Webseek project [35] demonstrate that l_1 is marginally better than l_2 for signature-based similarity image search. It is expected that the images with significant overlap have a high number of feature matchings, giving rise to similar signatures and, thus, a small l_1 value.

At this moment, best candidates (i.e. those with a minimum value of l_1) could be taken and passed to the next step for its validation and loop-closing confirmation or rejection. However, it is possible to improve the quality of these candidates by taking advantage of the trajectory information. If, for example, the comparison made using the l_1-norm for the latest h_q indicates that the best candidate to close loop is h_i then, the neighbors of h_i have a high probability to be also candidates for the subsequent queries. A common way to introduce this probability over the candidates and their neighbors consists in using a normal distribution weighting function [10]; let S_q be the random variable representing the loop-closure hypotheses for the current query image (I_q), then the event $S_q = i$ is the event that the query image I_q closes a loop with a previous image I_i, while $S_q = -1$ is the event that no loop-closure has been found for the query. We define I_c as one of the candidate images (obtained after the l_1-norm comparison) and I^s as the set of its s-neighbor images plus I_c (i.e. $I^s = \{I_{c-s}, ..., I_{c+s}\}$). In a Gaussian probabilistic context, similarly to [1], we formulate the probability that the current query image I_q closes a loop with a certain image in the neighborhood of I_c as follows:

$$p(S_q|I_b^s) = \eta \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(I_b^s - I_c)^2}{2\sigma^2}} \tag{7}$$

where I_b^s is the b-nth element of I^s, i.e. $b \in [c - s, ..., c + s]$. η is a normalization factor used to weight the resulting probability according to the result of the l_1-norm comparison. σ is the amplitude of the normal distribution (standard deviation) and depends on the overlap between images. For example, in a side-looking sequence like [8], if the frame rate causes 3 consecutive images to overlap $\sigma = 3$.

Through eq. 7 each image of I^s is labeled with a probability of closing a loop with the query I_q. Therefore, the probability that the set of images I^s closes a loop with I_q is:

$$p(S_q|I^s) = \left[p(S_q|I_{c-s}^s), ..., p(S_q|I_{c+s}^s)\right] \tag{8}$$

Since it is desirable to have more than one candidate to close a loop, let us define Z_p as the set of p-best image candidates obtained after the l_1-norm comparison: $Z_p = \left\{I_{c_0}, I_{c_1}, ..., I_{c_{p-1}}\right\}$. The set of the corresponding neighbors of Z_p can be expressed as $Z_p^s = \left\{I^{s_0}, I^{s_1}, ..., I^{s_{p-1}}\right\}$.

So, the probability that a query image I_q closes a loop with some image in the neighborhood of the p-best candidates can be expressed as follows:

$$p(S_q|Z_p^s) = \sum_{j=0}^{p-1} p(S_q|I^{s_j}) \tag{9}$$

where j represents the candidate image index. Figure 3 illustrates the result of applying eq. 9 (taking $p = 5$ and $s = 10$) over a certain query image of one of the datasets used in the experimental results. The x-axis represents all the image indices of the trajectory and the y-axis is the probability that the query image (I_q) closes a loop with any of the other images.

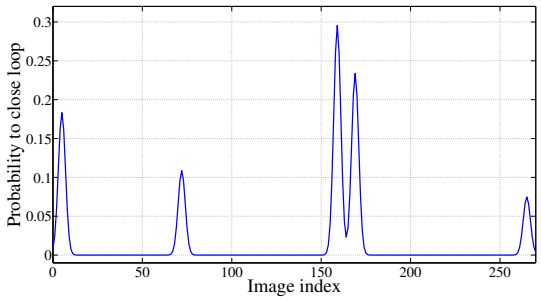


Figure 3: Probability vector example. Illustrates the probability (y-axis) of every image index (x-axis) to close loop with the query image. In this example the five best candidates ($p = 5$) have been taken.

As we mentioned above, the probability obtained in eq. 9 can be used on consecutive image queries to refine the results. Thus, let $Z_p^s(q)$ be the set of neighbors of all the candidates to close a loop with the latest query image I_q. Then, it is possible to take into account the probability information of the previous queries to compute the probability that the latest query image closes a loop with any other image (I_i), applying:

$$p(S_q|I^I) = \sum_{\omega=0}^{\upsilon} p(S_q|Z_p^s(q-\omega)) \qquad (10)$$

where $I^I = \{I_0, I_1, ...I_{q-s}\}$ and υ is the number of prior queries to take into account for the calculation of the final probability.

The result of eq. 10, for $\upsilon = 0$, is a probability vector similar to the one shown in fig. 3.

Finally, the similarity image search algorithm returns a vector with the image indices of the p-best candidates, which correspond to the p-highest peaks of the result of eq. 10. Then, these candidates are passed to the next stage where they will be rejected or confirmed as valid loop closures. In that sense, p is a configurable parameter that must be set properly, in order to get a trade-off between efficiency and precision. Large values of p increases the computational time since the steps of descriptor matching and epipolar geometry can be repeated p times, but the probability of finding a valid loop-closure is higher. Otherwise, small values of p significantly reduce the runtime but decreases the probability of loop-closing. Experiments demonstrate that values of p in the range of 2-5 produce the desired results.

4.2 Loop-Closure Validation

For each p-best candidate images selected in the previous step, its descriptor matrix is recovered from the descriptors database to perform a matching with the feature descriptors of the query image. If the number of descriptor matchings is above a pre-defined threshold, outliers are discarded by imposing the epipolar geometry constraint ($xFx^{'} = 0$ where F is the Fundamental Matrix, x is the vector containing the features of the query and $x^{'}$ is the vector containing the features of the candidate) and applying RANSAC [17]. Two frames whose features fulfill the epipolar constraint are very likely to represent the same scene viewed from different viewpoints. If this process ends consistently, that loop-closure is definitely accepted.

The execution time of the loop-closure validation stage, which is the task that requires more computational resources, depends only on the selected number of candidates (p), the descriptor type, and the image size. Therefore, this time is bounded and does not depend on the length of the robot trajectory or the total number of processed images. On the other hand, the similarity image search stage compares the signature of the query image with all the previously stored signatures, thus its complexity is linear with $\mathcal{O}(kmT)$ where T is the size of the hash table (i.e. the amount of images already processed). However, the execution time of this stage is more than 7 orders of magnitude lower than for the validation (using SIFT descriptors and $p = 5$). This means that, to have an execution time of the similarity image search stage higher than the execution time of the validation stage, a table with more than 10^7 entries is needed. Experiments on a commercial laptop show that one signature-to-signature comparison with $k = 3$ projections and SIFT descriptors ($m = 128$) takes about $10ns$. Thus, for example, a trajectory comprising 10.000 images would spend approximately $0.1ms$ on the similarity image search stage for a given query.

5 Experimental Results

The experiments presented in this section are aimed to validate, independently, the two main parts of the proposed algorithm.

First, the *Similarity Image Search* method is evaluated and compared with three well known alternatives: (a) OpenFABMAP2 [16] which is one of the most popular approaches of image retrieval for loop-closure detection based on BoW, (b) VLAD [18] as one of the most outstanding global descriptors in the literature; the implementation of VLAD included in the public VLFeat library [40] was used in all the following experiments, and, (c) an efficient LSH algorithm for loop-closure detection recently presented in [34]. To compare with OpenFABMAP2 and VLAD, we use four different datasets taken at two completely different environments, two outdoors and two underwater. Regarding the comparison with the third approach, although there is no public implementation of the Shahbazi and Zhang's method, we use the data given by the authors in their original work [34] setting their proposal against BoW.

Secondly, the *Loop-Closure Validation* is evaluated together with the overall loop-closing algorithm and compared to the work of Angeli et al. [1], which is one of the most outstanding and accepted contributions to the loop-closure detection and confirmation using BoW. Moreover, the authors provide the image datasets (one indoor and one outdoor) with which the experiments were performed in the original paper, thus the comparison is straightforward.

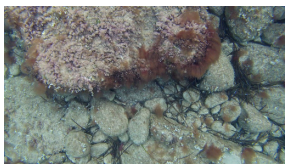
For an easier understanding and to short, from now on, our algorithm, as described in fig. 2, will be referred to as HALOC, while its *Similarity Image Search* part will be named *sisHALOC*. On the other hand, OpenFABMAP2 will be cited as BoW, [34] as LSH and [1] as FIBoW.

5.1 Similarity Image Search Results

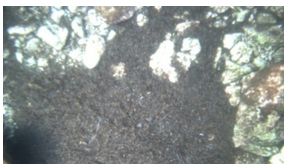
One of the most important contributions of the present proposal lies in the similarity image search stage. In the first part of this section, the performance of sisHALOC is compared to BoW and VLAD. Two outdoor datasets and two underwater datasets are used in this experiment (Fig. 4 shows sample images of these datasets). The outdoor datasets correspond to two public image sets of the Oxford city center and Oxford new college provided by the Oxford Mobile Robotics group [8]. Images were grabbed every 1.5 meters with a resolution of 640 $\times$ 480 pixels. The city center dataset contains a total of 2474 images while the new college dataset has 2146. The two other datasets were recorded underwater, in the Valldemossa (Mallorca, Spain) harbor and beach respectively. A GoPro camera moving at a constant depth with a resolution of 512 $\times$ 288 pixels was used to grab these datasets. The harbor dataset has a total of 682 images while the beach dataset has 738 images. Ground truth files are available for all these datasets indicating the total number of loop closures found in the trajectory and which images are involved in each one of them.

(a) Sample image of Oxford city center dataset.

(b) Sample image of Oxford new college dataset.



(c) Sample image of Valldemossa harbor dataset.



(d) Sample image of Valldemossa beach dataset.

Figure 4: Sample images from the outdoor (a and b) and underwater (c and d) datasets.

For BoW and VLAD training, we used two additional datasets to reproduce the situation in which a visual vocabulary is built from images taken in a certain environment, but used for loop-closure detection in another one. The UCID database [33] was used to train the Oxford sequences and an underwater dataset available to our group from previous research [46] was used for the training of underwater sequences.

The outputs of BoW, VLAD and *sisHALOC* are directly comparable since all of them return one array with the best loop-closure candidates ordered by its probability to close loop with the query image. This probability

is also returned by BoW and *sisHALOC* so the final user can implement a threshold to decide if a candidate will be processed or not. In order to not depend on this threshold when measuring the quality of the candidates provided by the algorithms, in this experiment we only take into account the real loop closures (i.e. those appearing in the ground truth). The experimental setup is as follows:

1. We used SIFT features for all methods: BoW, VLAD and *sisHALOC*. All the other parameters were set to the default for all the algorithms, except, a) the cluster radius of BoW that was properly configured to produce a vocabulary of 10K words and, b) the VLAD number of clusters to $k = 64$ and the vector dimension after the PCA dimensionality reduction $D' = 64$ [18]. We did not change any parameter in the execution of these algorithms over the four datasets, since one of the priorities was to demonstrate their independence with respect to the image and environment type.

2. Each image of the sequence was used as query to be compared with all the previously processed, except the 10 precedent ($s = 10$) to avoid closing loops between consecutive images.

3. For each query, 5 loop-closure candidates were requested for all the algorithms ($p = 5$).

4. For every position in the returned vector of candidates, we computed its percentage of valid loop closures by validating every candidate with the ground truth.

Figure 5 shows the accuracy of the three algorithms over the outdoor and underwater datasets by means of the percentage of valid loop closures for the top 5 candidates of all the queries. From these graphs we can highlight two main aspects: first, *sisHALOC* has the highest probability to close loops, regardless of the environment type (outdoor and underwater) and number of selected candidates. On the other hand, BoW and VLAD have a high dependency on the environment type, since these algorithms alternate good results depending on the dataset. Second, the percentage of valid loop closures of BoW and VLAD increase substantially as the number of candidates are increased, while *sisHALOC* for $p = 4$ and $p = 5$ behaves similar than for $p = 3$. This means that, choosing $p = 3$, the overall loop closing procedure will spend much less time confirming or rejecting the candidates, without implying a substantial loss of accuracy.

	Performance		Speed up over BoW	
p	LSH	*sisHALOC*	LSH	*sisHALOC*
1	12%	33%		
2	7%	39%	0.1	1.1
5	4%	26%		

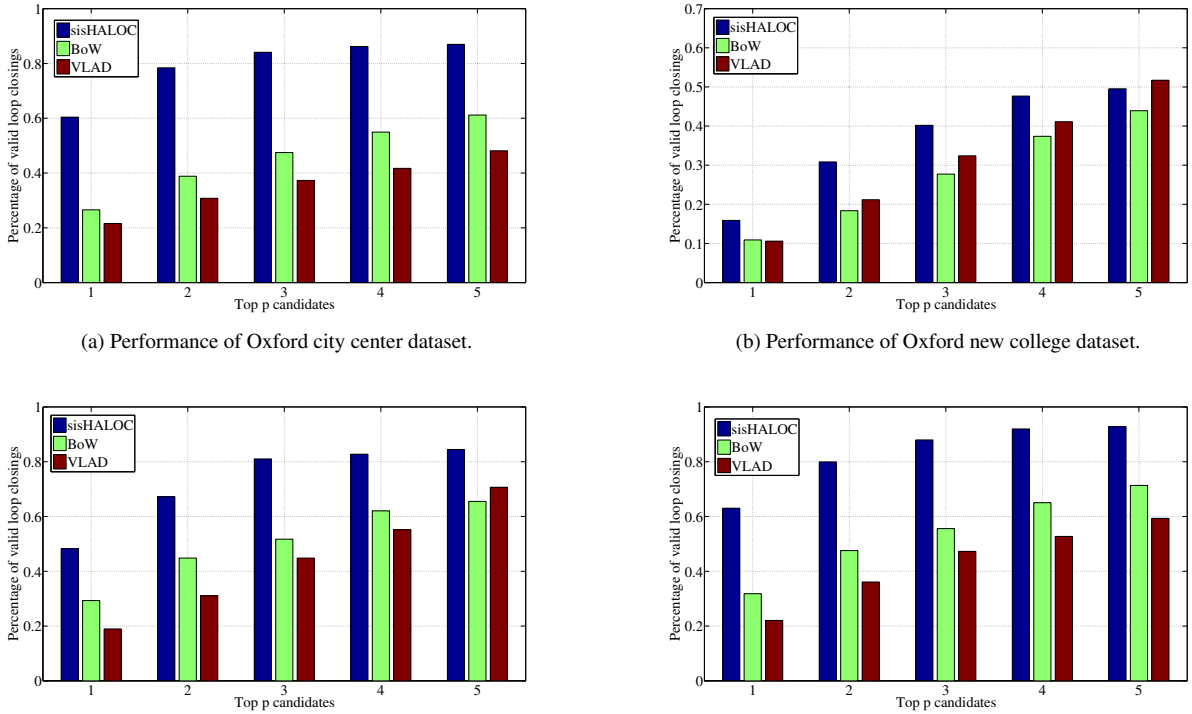
Table 2: Comparison of performances and runtime between LSH and *sisHALOC*.

Table 1 illustrates the algorithm mean runtimes for every dataset. This is the mean of the time that the algorithm spend between a new query image arrives and the potential candidates to close loop with it are calculated. Therefore, this time has two parts: the calculation of the global signature for the current query image (bounded) and the comparison/search for possible loop closing candidates using the past images (unbounded). From Table 1, BoW and *sisHALOC* have similar execution times, being *sisHALOC* slightly better in all cases. Furthermore, VLAD runs on one order of magnitude larger than BoW and *sisHALOC*. This is because of the calculation of VLAD itself and not the search for candidates, which is very fast. Although the VLAD implementation used is optimized at runtime, calculating VLAD is expensive. It is worth noting that, in the original work of VLAD [18], the authors do not consider the calculation of VLAD vector in the runtime evaluation, i.e. the execution time they present only takes into account the search for candidates.

Comparing *sisHALOC* to LSH is not as direct, since neither the datasets nor the implementation are provided by the authors. Fortunately, a detailed comparison of the LSH and BoW methods is given in [34], using an outdoor dataset. Thus, we use this information to calculate the relative improvement of LSH respect to BoW and do the same with *sisHALOC*. To be as fair as possible, we use the outdoor Oxford city center dataset and the same set of parameters from [34] to configure BoW.

The experimental setup is as follows:

1. From [34] we take the results of LSH2 configuration, which has a good trade-off between performance and



(a) Performance of Oxford city center dataset.

(b) Performance of Oxford new college dataset.

(c) Performance of Valldemossa harbor dataset.

(d) Performance of Valldemossa beach dataset.

Figure 5: Performance results: the percentage of valid loop closures for every candidate index for the outdoor datasets (a and b) and underwater datasets (c and d).

	Runtime (ms)			
	Ox. city	Ox. college	Vall. harbor	Vall. beach
sisHALOC	6.54	7.37	1.88	1.81
BoW	8.72	10.03	2.87	2.74
VLAD	70.79	70.86	32.85	31.39

Table 1: Runtime results: the mean runtime of the candidate search for every image in the trajectory on all datasets.

execution time.

2. BoW with a clustering size of 1000 and *sisHALOC* are both executed using SIFT descriptors for $p = 1, 2, 5$ and all the other parameters are set to default.

3. For every candidate index we compute the percentage of valid loop closures as the sum of correct matchings (validated thanks to the ground truth) divided by the total number of loop closures into the trajectory (as in [34]).

4. The execution time of BoW and *sisHALOC* is the mean of all the executions for $p = 1, 2, 5$.

Table 2 details the comparison of LSH and *sisHALOC* (in terms of performance and execution time) to BoW. For $p = 1, 2, 5$ the table shows the increased percentage in the number of loop closures found using LSH and *sisHALOC* compared to BoW. The speed up of LSH and *sisHALOC* over BoW is also illustrated. LSH always outperforms BoW detecting at least 4% more loop closures (for $p = 5$), but it requires a runtime 10 times greater than BoW. In the original paper [34], 0.18 seconds are required to execute LSH for the search of image candidates to close loop with a query, without taking into account the time needed for the descriptor extraction process and for the subsequent validation stage. This implies that LSH could not be implemented in an online localization system that needs loop-closure corrections at, at least, 5Hz. On the other hand, *sisHALOC* highly improve the results of BoW detecting a 39% more loop closures for $p = 2$, for example. Moreover, its execution time is one order of magnitude faster than LSH.

5.2 Loop-Closure Validation Results

The last stages of HALOC algorithm (shown in figure 2) accept or reject the image candidates provided by the similarity image search part. In this section, the performance of the overall process is analyzed using a set of public image datasets different from those of Section 5.1 and that have already been used in FIBoW [1]: two public sequences (one indoor and one outdoor) grabbed with a monocular handheld camera with a resolution is 240 $\times$ 192 pixels. The indoor dataset has 388 images while the outdoor has 531 images. These sequences are called Lip6Indoor and Lip6Outdoor respectively and both are provided with a ground truth file just as in the previous experiments.

The performance of the loop-closing process is evaluated with the following metrics:

$$Precision = \left(\frac{TP}{TP + FP}\right); Recall = \left(\frac{TP}{TP + FN}\right); \quad (11)$$

where TP (*true positives*) is the number of images correctly accepted as loop closures, FP (*false positives*) is the number of images wrongly accepted as loop closures and FN (*false negatives*) is the number of images wrongly discarded as loop closures. A high $Recall$ value is desirable since it means that most of the available loop closures have been detected. On the other hand, a $Precision$ of 100% is imperative as this implies no false loop closures have been accepted. The inclusion of false loop closures in the localization cycle would cause important errors in the robot pose estimation.

The assessment of the loop-closing procedure is performed for each dataset as follows:

1. SIFT features and default parameters are set for HALOC.

2. Each image of the sequence is used as query to be compared with all the previous ones, except the 10 precedent ($s = 10$) to avoid closing loops between consecutive images.

3. For each query, the 5 best candidates are taken to close a loop with it ($p = 5$).

4. Every candidate is passed to the validation stage (descriptor matching and epipolar geometry) to be confirmed or rejected.

5. When a loop-closure is confirmed by both HALOC and the ground truth, it is accounted as a TP. If the ground truth does not confirm the loop-closure, the candidate is labeled as a FP.

6. Finally, all the candidates not detected by HALOC but marked as loop closings in the ground truth are labeled as FN.

			FIBoW			HALOC			
Sequence	#img	Length	Prec.	Recall	CPU	Prec.	Recall	CPU i7	CPU Core2Duo
Lip6Indoor	388	6m28s	100%	68%	1m33s	100%	74%	0m19s	0m39s
Lip6Outdoor	531	17m42s	100%	70%	6m48s	100%	76%	1m08s	3m04s

Table 3: Comparison of loop-closure detection performances for Lip6 datasets.

The experiments detailed here have been executed on two different machines: an Intel Core i7 at 2.20 GHz and 8GB of RAM and an Intel Core2Duo at 1.8GHz and 2GB of RAM, to facilitate the comparison of HALOC against FIBoW, where the experiments were processed on an Intel Core2Duo at 2.33GHz.

Table 3 shows the results of the assessment for the sequences Lip6Indoor and Lip6Outdoor, compared with the results obtained by FIBoW, using the same datasets. Table 3 shows the name of the image set, the number of images of each trajectory (#img), the recording time (Length) of the trajectory and the corresponding $Precision$ (Prec.), $Recall$, and execution time (CPU) for both approaches. Our method gives better recalls than the fastest approach (SIFT without histograms) of FIBoW and it is 58% faster for the indoor dataset and 55% faster for the outdoor one (taking times of Core2Duo machine).

For all the experiments, our approach achieved a 100% of $Precision$, which indicates the best rate of correct detections. Also, high rates of $Recall$ indicate that most of the available loop closures are detected, which endows the localization process with a high reliability.

Moreover, these experiments show how the global signature proposed in this paper is useful, when using descriptors invariant to scale changes, translations and rotations, to detect loop closings of the same area viewed from different translated/rotated viewpoints or at different distance to the camera. Table 4 contains some of the loop closures detected in the Lip6Indoor dataset for rotated, scaled and translated images (see figure 6). This is the consequence of that, a) the bucketing method explained in section 3 forces descriptor matrices with the same dimension, and, b) the vectors used in the projections are unitary. Table 5 shows the first 5 values (from ρ_0 to ρ_4) of the new image signature of three image samples and for two different types of projections. The columns named $\widehat{u}$ correspond to signature values generated using a random unit vector (as explained in section 3) while columns named CST correspond to values generated using a vector with all the components containing a constant value equal to the mean of the unit random vector. The resulting signatures are very similar so that the order of descriptors in the descriptors matrix has a low influence in the calculation of the signature, i.e. rotation, scale and translation effects inherent to SIFT or SURF features are preserved, to a certain extent, in the global signature. However, they are also different enough so that descriptor changes are reflected in the different projections.

Query Image	Retrieved Candidates
201	58 (r)
202	60 (r,s)
203	60 (r,s)
204	59 (r,s), 63 (r,s,t)
207	64 (r)
213	65 (r,s,t), 71(r,s)

Table 4: Some of the loop-closure candidates provided by *sisHALOC* when executed over rotated, scaled and translated images of the Lip6Indoor dataset. Image numbers correspond to the image filenames of the dataset. The indices *r*, *s* and *t* indicate the following transformations between query and candidate: rotated, scaled and translated respectively.

Assessment tools used in this section are provided in conjunction with the HALOC C++ library in the public repository [29].

6 Conclusions and Future Work

This paper proposes a new global descriptor-based image signature especially designed to detect candidates for loop closings, together with a complete procedure to validate them. The aim of this new approach is to characterize images in a fast but reliable manner, in such a way that, im-

(a) Query image (204). (b) Candidate image 1 (59). (c) Candidate image 2 (63).

(d) Query image (213). (e) Candidate image 1 (65). (f) Candidate image 2 (71).

Figure 6: Two examples (one per row) of a loop-closure detection for rotated, scaled and translated images of the Lip6Indoor dataset. Left column represents the query image while center and right column are the first and second candidate to close loop with the respective query.

	H1		H2		H3	
	$\widehat{u}$	CST	$\widehat{u}$	CST	$\widehat{u}$	CST
ρ_0	0.695	0.648	0.596	0.613	0.708	0.669
ρ_1	0.291	0.267	0.327	0.352	0.611	0.647
ρ_2	0.474	0.344	0.660	0.701	0.433	0.483
ρ_3	0.315	0.306	0.674	0.630	0.307	0.323
ρ_4	1.169	1.245	0.779	0.676	0.261	0.228

Table 5: Three samples of the first five signature values for two types of projections: unit vector ($\widehat{u}$) and constant value (CST).

ages that overlap present signatures with small differences and images that do not overlap present distant signatures. The proposed image signature is obtained by projecting the components of the visual features of every image onto multiple orthogonal directions, generating a small vector easy and fast to operate with. In this way, the perceptual aliasing present in clustering algorithms caused by using the same visual words for different visual features is avoided. Images with similar signatures are proposed as candidates for loop-closing, and are passed to a second stage where the existence of a loop-closure is confirmed or rejected.

The performance of the proposed new method is compared to three of the most popular approaches to similarity image search: OpenFABMAP2, VLAD and LSH. The assessment has been done by comparing four datasets, two terrestrial and two underwater. The results of these experiments have shown that HALOC algorithm outperforms OpenFABMAP2, VLAD and LSH on all the environments, using the same set of parameters for each different technique. Although it is possible to adapt the configuration of BoW and VLAD to the type of images and carry out a tuning process to improve performance for a given dataset, this paper demonstrates its high dependence on the environment. Otherwise, HALOC performs well in all the scenarios with the default parameters (which are in fact very few) and with no training stage.

The overall algorithm used to confirm or reject the possible loop closings provided by the similarity image search stage has been assessed with two more datasets and compared to one of the most popular BoW-based loop-closure detection systems on the literature. $Recall$ and $Precision$ result in comparable or even better figures, but with a notably reduced execution time. The experimental assessment also shows an excellent performance when using SIFT features and detecting loop closures that present changes on scale, rotation and translation.

In conclusion, HALOC algorithm has shown to be a multi-environment plug and play loop closure detector. It achieves high performances for datasets grabbed indoor, outdoor and underwater with a minor runtime than other alternatives, without the need of tuning the parameters for every scene and without an offline training stage. Furthermore, HALOC preserves, to a certain extent, the properties of features invariant to scale changes, rotations and translations.

Finally, the implementation of HALOC is available to the scientific community in a public C++ repository [29]. This package has been designed and coded focusing our efforts on getting the maximum performance while making an efficient use of memory and CPU.

Future work includes the integration of the technique into a visual SLAM procedure to execute long trajectories online.

References

[1] A. Angeli, D. Filliat, S. Doncieux, and J. A. Meyer. Fast and incremental method for loop-closure detection using bags of visual words. *Robotics, IEEE Transactions on*, 24(5):1027–1037, 2008.

[2] R. Arandjelovic and A. Zisserman. All about vlad. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1578–1585. IEEE, 2013.

[3] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *Proceedings of the European Conference on Computer Vision (ECCV), 2006.*, pages 404–417. Springer, 2006.

[4] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. In *Proceedings of IEEE European Conference on Computer Vision (ECCV)*, 2010.

[5] L. A. Clemente, A. J. Davison, I. D. Reid, J. Neira, and J. D. Tardós. Mapping large loops with a single

hand-held camera. In *Robotics: Science and Systems*, 2007.

[6] M. Cummins and P. Newman. Accelerated appearance-only slam. In *Robotics and automation, 2008. ICRA 2008. IEEE international conference on*, pages 1828–1833. IEEE, 2008.

[7] M. Cummins and P. Newman. Fab-map: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 27(6):647–665, 2008.

[8] M. Cummins and P. Newman. The oxford mobile robotics group datasets, 2008. `http://www.robots.ox.ac.uk/~mobile/IJRR_2008_Dataset/data.html`.

[9] M. Cummins and P. Newman. Fab-map: Appearance-based place recognition and mapping using a learned visual vocabulary model. In *Proceedings of the International Conference on Machine Learning*, 2010.

[10] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part i. *Robotics Automation Magazine, IEEE*, 13(2):99–110, June 2006.

[11] D. Filliat. A visual bag of words method for interactive qualitative localization and mapping. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 3921–3926. IEEE, 2007.

[12] E. Garcia-Fidalgo and A. Ortiz. Vision-based topological mapping and localization by means of local invariant features and map refinement. *Robotica*, FirstView:1–25, 5 2014.

[13] E. Garcia-Fidalgo and A. Ortiz. Vision-based topological mapping and localization methods: A survey. *Robotics and Autonomous Systems*, In Press, 2015.

[14] A. Geiger, J. Ziegler, and C. Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *IEEE Intelligent Vehicles Symposium*, Baden-Baden, Germany, June 2011.

[15] A. Gionis, P. Indyk, R. Motwani, et al. Similarity search in high dimensions via hashing. In *VLDB*, volume 99, pages 518–529, 1999.

[16] A. Glover, W. Maddern, M. Warren, S. Reid, M. Milford, and G. Wyeth. Openfabmap: An open source toolbox for appearance-based loop closure detection. In *The International Conference on Robotics and Automation*, St Paul, Minnesota, 2011. IEEE.

[17] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521623049, Cambridge, UK, 2003.

[18] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3304–3311. IEEE, 2010.

[19] J. Konečnỳ and M. Hagara. One-shot-learning gesture recognition using hog-hof features. *The Journal of Machine Learning Research*, 15(1):2513–2532, 2014.

[20] J. Košecká, F. Li, and X. Yang. Global localization and relative positioning based on scale-invariant keypoints. *Robotics and Autonomous Systems*, 52(1):27–38, 2005.

[21] S. Lin, M. T. Ozsu, V. Oria, and R. Ng. An extendible hash for multi-precision similarity querying of image databases. In *In Proceedings of International Conference of Very Large Data Bases (VLDB)*, volume 1, pages 221–230, 2001.

[22] M. Liu and R. Siegwart. Dp-fact: Towards topological mapping and scene recognition with color for omnidirectional cameras. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 3503–3508, May 2012.

[23] M. Liu and R. Siegwart. Topological mapping and scene recognition with lightweight color descriptors for an omnidirectional camera. *IEEE Transaction on Robotics*, 30(2):310–324, April 2014.

[24] Y. Liu and H. Zhang. Visual loop closure detection with a compact image descriptor. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1051–1056, 2012.

[25] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[26] M. K. Mıhçak and R. Venkatesan. New iterative geometric methods for robust perceptual image hashing. In *Security and privacy in digital rights management*, pages 13–21. Springer, 2002.

[27] M. Milford and G. Wyeth. Visual route-based navigation for sunny summer days and stormy winter nights. In *The International Conference on Robotics and Automation*, 2012.

[28] V. Monga and B. L. Evans. Perceptual image hashing via feature points: Performance evaluation and tradeoffs. *IEEE Transactions on Image Processing*, 15(11):3453–3466, November 2006.

[29] P. L. Negre and F. Bonin-Font. libhaloc. `https://github.com/srv/libhaloc`, 2014. [Online; published May 2014].

[30] A. Oliva and A. Torralba. Modelling the shape of the scene: A holistic representation of the spatial envelope. *International Journal on Computer Vision.*, 42(3):145–175, 2001.

[31] S. Roy and Q. Sun. Robust hash for detecting and localizing image tampering. In *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, volume 6, pages VI–117. IEEE, 2007.

[32] S. Roy, X. Zhu, J. Yuan, and E. C. Chang. On preserving robustness-false alarm tradeoff in media hashing. In *Visual Communications and Image Processing 2007. SPIE Proceedings*, volume 6508. SPIE, 2007.

[33] G. Schaefer and M. Stich. Ucid - an uncompressed colour image database. In *Proc. SPIE, Storage and Retrieval Methods and Applications for Multimedia*, pages 472–480, San Jose, USA, 2004.

[34] H. Shahbazi and H. Zhang. Application of locality sensitive hashing to realtime loop closure detection. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems(IROS)*, pages 1228–1233, San Francisco (USA), 2011.

[35] J. R. Smith and S. Chang. Visually searching the web for content. *IEEE multimedia*, 4(3):12–20, 1997.

[36] N. Sunderhauf and P. Protzel. Brief-gist-closing the loop by simple means. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 1234–1241. IEEE, 2011.

[37] A. Swaminathan, Y. Mao, and M. Wu. Robust and secure image hashing. *Information Forensics and Security, IEEE Transactions on*, 1(2):215–230, 2006.

[38] N. Sünderhauf and P. Protzel. Brief-gist - closing the loop by simple means. In *Proceedings of IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2011.

[39] I. Ulrich and I. Nourbakhsh. Appearance-based place recognition for topological localization. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 2, pages 1023–1029. Ieee, 2000.

[40] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/, 2008.

[41] R. Venkatesan, S. M. Koon, M. H. Jakubowski, and P. Moulin. Robust image hashing. In *Image Processing, 2000. Proceedings. 2000 International Conference on*, volume 3, pages 664–666. IEEE, 2000.

[42] J. Wan, Q. Ruan, W. Li, G. An, and R. Zhao. 3d smosift: three-dimensional sparse motion scale invariant feature transform for activity recognition from rgb-d videos. *Journal of Electronic Imaging*, 23(2):023017–023017, 2014.

[43] J. Wan, Q. Ruan, W. Li, and S. Deng. One-shot learning gesture recognition from rgb-d data using bag of features. *The Journal of Machine Learning Research*, 14(1):2549–2582, 2013.

[44] B. Williams, M. Cummins, J. Neira, P. Newman, I. Reid, and J. Tardós. An image-to-map loop closing method for monocular slam. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ*

International Conference on, pages 2053–2059. IEEE, 2008.

[45] B. Williams, M. Cummins, J. Neira, P. Newman, I. Reid, and J. Tardós. A comparison of loop closing techniques in monocular slam. *Robotics and Autonomous Systems*, 57(12):1188–1197, 2009.

[46] S. Wirth, P.L. Negre Carrasco, and G. Oliver. Visual odometry for autonomous underwater vehicles. In *Proceedings of the IEEE Oceans*, Bergen, Norway, 2013.