Journal Title
XX(X):1–22
©The Author(s) 2016
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

Fast Loop Closure Detection using Visual-Word-Vectors from Image Sequences

Loukas Bampis, Angelos Amanatiadis and Antonios Gasteratos

Abstract

In this paper, a novel pipeline for loop closure detection is proposed. We base our work on a bag of binary feature words and we produce a description vector capable of characterizing a physical scene as a whole. Instead of relying on single camera measurements, the robot's trajectory is dynamically segmented into image sequences according to its content. The visual word occurrences from each sequence are then combined to create Sequence-Visual-Word-Vectors and provide additional information to the matching functionality. In this way, scenes with considerable visual differences are firstly discarded, while the respective image-to-image associations are provided subsequently. With the purpose of further enhancing the system's performance, a novel temporal consistency filter (trained off-line) is also introduced to advance matches that persist over time. Evaluation results prove that the presented method compares favorably with other state-of-the-art techniques, while our algorithm is tested on a tablet device verifying the computational efficiency of the approach.

Keywords

Loop Closure Detection, Image Sequences, Visual SLAM, Mobile Robotics, Low-power Embedded Systems.

1 Introduction

The problem of visual Place Recognition (vPR) refers to the ability of a system to recognize a scene based on visual sensing and it has been used during the last decade in order to address many challenges in mobile robotics. As part of a Simultaneous Localization And Mapping (SLAM) system, vPR has been applied in a variety of forms and alterations such as the Loop Closure Detection (LCD) and the Re-Localization (RL) procedures. An LCD engine is responsible for detecting revisited trajectory regions and creating additional edge constraints between the current and earlier pose nodes on graph-based SLAM systems (Folkesson and Christensen 2004; Thrun and Montemerlo 2006; Grisetti et al. 2010). Those additional edge constraints provide supplementary information regarding the measurements' arrangement in the 3D space, and they can be used to further improve the SLAM output in an on-line and/or post-processing manner (Mur-Artal et al. 2015; Latif et al. 2013; Strasdat et al. 2010; Mei et al. 2009). Moreover, an RL system utilizes the visual information in order to recover the robot's position in an already known environment (the problem of kidnapped robot) or in localization failure scenarios (Konolige et al. 2010; Wolf et al. 2005; Mur-Artal and Tardós 2014). Even though these challenges refer to different applications, they share the same basic functionality of identifying a previously visited scene and thus they can be addressed by common solutions.

During the past decade a plethora of vPR techniques have been presented in the literature. Williams et al. (2009) distinguished the approaches into three main categories with respect to the type of data they associate. In the first category, referred to as Map-to-Map, correspondences are found between features taking into account both their appearance and their relative location inside the world. Furthermore, Image-to-Map methods aim to recognize places by associating features between the latest acquired frame and a retained spatial representation of the already seen world. Finally, Image-to-Image matching approaches (or appearance-based techniques) detect correspondences between the images themselves and present better scaling capabilities in long trajectory cases.

Department of Production and Management Engineering, Democritus University of Thrace

Corresponding author:

Loukas Bampis, Department of Production and Management Engineering, Democritus University of Thrace, 12 Vas. Sophias, GR-671 32, Xanthi, Greece

Email: lbampis@pme.duth.gr

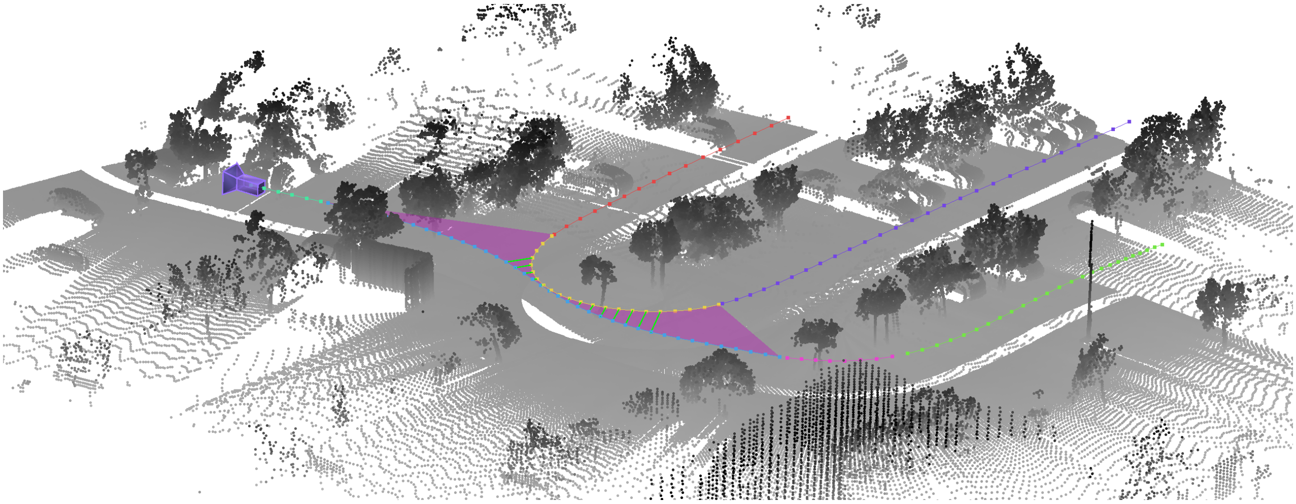


Figure 1. A 3D representation of the proposed loop closure detection pipeline tested on Malaga 2009 Parking 6L (Blanco et al. 2009) dataset. As the robot moves, the executed trajectory is segmented into intervals/sequences (illustrated with different colors). The formulated S-VWVs are used to detect sequence matches (marked with the magenta plain) and signal the existence of loop closing frames. The individual image-to-image associations (marked with green links) are provided via the individual I-VWVs.

The most common approach for addressing appearance-based LCD tasks refers to the characterization of each individual frame by an aggregation of local image descriptors. As the robot moves, revisited places are detected by measuring content similarities between the current input frame (query) and all the previous ones (database). In order to provide efficiency in the implementation, the model of Bag of Visual Words (BoVW) can be utilized as a means of quantizing the extracted descriptors' space. In the general case, every input frame is assigned with one Image-Visual-Word-Vector (I-VWV). The entries of this vector correspond to a weighted frequency of occurrence for every visual word in the given image (histogram). The created I-VWVs are treated as image descriptors, thus loop closing pairs of camera poses are recognized by calculating similarity metrics between them. The aforementioned approach was initially inspired by image retrieval techniques (Sivic and Zisserman 2003), yet in some vPR algorithms, measurements obtained from close-in-time instances are summed in order to enhance the results. Finally, it has been proven that the representation of the created BoVW with a tree structure (vocabulary tree) improves significantly the computational efficiency (Nister and Stewenius 2006).

In this work, we present an improved pipeline for appearance-based LCD which combines the visual information from multiple frames in order to describe a physical scene as a total. As the robot moves, the input camera stream is dynamically segmented into intervals (image sequences), based on the scene's content variations. For each image sequence, the extracted feature descriptors are converted into visual words and combined to produce one global Sequence-Visual-Word-Vector (S-VWV) as well as the individual I-VWVs. Thus, the revisited trajectory regions are detected on a first level by measuring the similarities between all S-VWVs in the database, while the loop closing frames are determined using the individual I-VWVs only for the associated sequences' image-members. A typical example of the aforementioned procedure is illustrated in Fig. 1. Note that henceforth, the term "sequence" will refer to "sequence of images" for brevity.

The main contributions of the paper in hand can be summarized as follows:

- Using a description vector capable of characterizing an image sequence as a whole, our method provides more information to the matching functionality advancing the LCD performance. Additionally, since we rely our pipeline on such a descriptor, rather than accumulating the similarities between multiple I-VWVs, the system's performance is not restricted by a per-frame perception of the environment.
- With the view to further enhance the produced sequence similarity measurements, our algorithm introduces a temporal consistency filter over the similarity matrix entries. The corresponding kernel's coefficients are calculated using a cost function minimization scheme on a set of training samples.
- The proposed methodology entails a reduced computational complexity as compared to other vPR techniques since our first level of sequence-to-sequence matching excludes the trajectory regions that are absolutely different in the general view.

In addition, the nature of our pipeline provides an efficient way to further reduce the visual word votes by considering only the entries that persist during the sequence formulation. An implementation of the proposed algorithm is tested on a mobile device utilizing the parallel execution capabilities of the ARM-NEON co-processor and proving its ability to run in real-time (in a sense of processing the input faster or in equal time with the execution frequency of a modern key-frame SLAM system) even in the case of a less powerful machine.

A preliminary version of presented work was presented in (Bampis et al. 2016). In this paper, we advance the system's performance by adopting a rotation and scale invariant local feature descriptor and a dynamical sequence identification technique, while additionally, we address the temporal consistency filtering as a classification problem operating on the sequence similarity scores. Furthermore, we provide a complete justification of the benefits offered by a unified VWV and extend our experiments to fully evaluate the performance of our algorithm. Finally, extensive comparative results are presented against other state-of-the-art sequence-based vPR techniques, proving the capabilities of the S-VWV based description.

The following section contains a discussion about the related work on the field of vPR and subsequently introduces the advantageous matching properties of the introduced S-VWVs. Section 3 describes in details our on-line pipeline together with the preprocessing steps for the vocabulary tree formulation and filter training. In Section 4, our experimental evaluation and comparative results against other state-of-the-art approaches are presented. The computational benefits of the proposed approach, together with the employed parallelization techniques and implementation details of the tested mobile application, are summarized and assessed in Section 5. Finally, Section 6 draws our final conclusions by describing our algorithm's potentials and contributions.

2 From Image to Sequence Description

In this section, we discuss some of the most representative techniques in the field of appearance-based vPR with the aim to lead our reader to the comprehension of the proposed sequence description method. For an extended survey of vPR, the reader can refer to the work of Lowry et al. (2016).

2.1 *Single-Image based Visual Place Recognition*

Probably the most acknowledged method on the field of vPR is FAB-MAP (Cummins and Newman 2008). According to that method, co-currency probabilities between observed visual words are used to perform appearance-based vPR. Although FAB-MAP constitutes the foundation for a plethora of later methodologies, it suffers in terms of performance, when repetitive patterns are accounted (Piniés et al. 2010), and execution time, due to the expensive extraction and matching of SURF features (Bay et al. 2006). In a later work, the same authors introduced an improved sparse approximation of their original technique, called FAB-MAP 2.0 (Cummins and Newman 2011), allowing their system to scale by more than two orders of magnitude. Another representative approach was proposed by Angeli et al. (2008), where the description relied on two visual vocabularies (one from SIFT descriptors (Lowe 2004) and another from local color histograms). Using a Bayesian filter, the detection was enhanced taking into account the matching probability of previously obtained measurements. Schindler et al. (2007) provided a more sophisticated representation of the visual vocabulary with a tree structure addressing city-scale vPR challenges. In their work, the Greedy N-Best Paths (GNP) algorithm was used so as to cluster the feature descriptors incrementally.

More recent techniques have been deviated from the aforementioned probabilistic approach of detecting loop closures with floating-point descriptors, like SIFT or SURF, offering faster but still competitive results (Mur-Artal and Tardós 2014; Gálvez-López and Tardós 2012; Khan and Wollherr 2015). More specifically, visual words from binary features, found in every camera measurement, are used in order to create image description vectors (I-VWVs). Thus, the detection of revisited places is achieved by obtaining similarity metrics, based on L1/L2 norm, between the individual I-VWVs. Gálvez-López and Tardós (2012) proposed a typical technique for this approach with the DBoW2 algorithm. Since in their case the Bayesian filtering was not included, the matching candidates were forced to follow a temporal consistency constraint. Mur-Artal and Tardós (2014), enforced DBoW2 by exploring the usage of a more sophisticated binary descriptor (ORB (Rublee et al. 2011)) and provided a real-time vPR, RL and LCD system.

Additionally, since the off-line formulation of a visual vocabulary is not suitable for every application, some methods suggested the on-line development of a BoVW by estimating an average representation of repetitive descriptors. For instance, Labbe and Michaud (2013) for large scale environments proposed the formulation of an on-line vocabulary based on a randomized forest of kd-trees achieving exquisite performance. Although their technique is capable of recognizing revisited places in constant-time, independently of the traversed trajectory's length, the computationally expensive SURF feature extraction and the constant updates of their vocabulary render the approach less appealing for normal scale scenarios (such as $20K$-$30K$ input frames). Aiming for an immediate reduction of the execution time, Khan and Wollherr (2015) proposed the on-line creation of a binary vocabulary based on the insertion of new visual words each time an unfamiliar descriptor is obtained. Since their method (IBuILD) utilizes efficient binary operations, it constitutes a more attractive solution in terms of computational complexity.

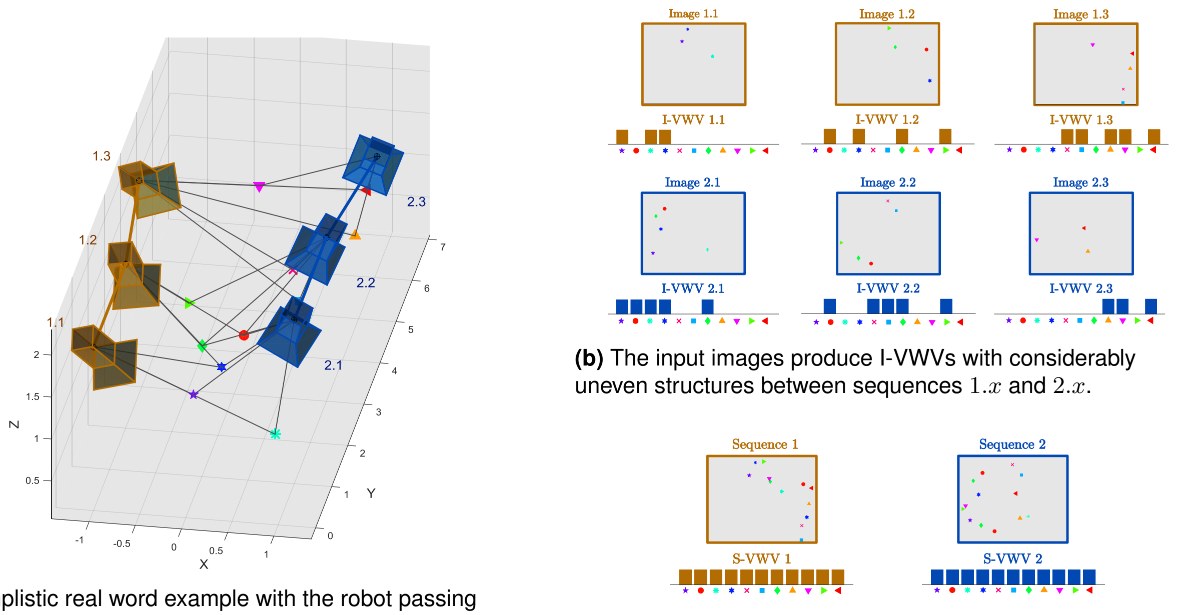
Recently, the concept of sequence-to-sequence matching has also been introduced in the literature. Newman et al. (2006) in their work for outdoor SLAM applications, pointed out the advantages of matching sequences instead of individual frames using an accumulative version of similarity matrices. The same notion appears (even on some abstract level) on other techniques as well (e.g. (Mur-Artal and Tardós 2014; Gálvez-López and Tardós 2012)) proving that the LCD performance can be strengthened when the visual information from more than one camera measurements is considered. Even though those techniques aim to take advantage of the additional information from the entire scene, they treat each sequence as an aggregation of image description vectors rather than visual words, subjecting their matching procedure to a per-frame view of the environment (visual words are redundantly clustered by the camera measurements). On the contrary, our method reformulates the process of creating VWVs and consider the whole sequence as a single “super-frame”. This approach offers invariance to the visual words’ distribution over the camera measurements, and it will be further analyzed in the following subsection.

Finally, sequence-based techniques have been reported addressing the vPR task under extreme environmental changes originated from different lighting conditions (day and night) or year seasons (Milford and Wyeth 2012; Arroyo et al. 2015). Even though the choice of more traditional local feature descriptors is avoided (due to their inability of matching under such intense environmental changes (Valgren and Lilienthal 2010)), the usage of global sequence descriptors is proven to be crucial for the achieved performance. Most lately, condition-invariant vPR techniques have been presented based on the classification characteristics of Convolution Neural Networks (CNNs). Methods like the ones presented by Sünderhauf et al. (2015a,b) and Arroyo et al. (2016) treat the output of particular CNN layers, initially trained for object detection tasks, as image descriptors and address the vPR problem by measuring distances between them. Even though CNN-based techniques offer superior retrieval performances, they are still decoupled from the LCD and SLAM functionalities. Sizikova et al. (2016) and Fei et al. (2016) in their respective works, accurately pointed out the CNN’s dependence over viewpoint-invariant surface appearances and the lack of topological information at the higher networks’ levels, which characterize them as suboptimal for LCD tasks. On the contrary, local feature-based techniques are widely used in visual SLAM applications (Mur-Artal et al. 2015; Lim et al. 2014; Davison et al. 2007; Klein and Murray 2007; Cieslewski and Scaramuzza 2017) and they can be efficiently combined with an illumination invariant image representation technique (e.g. (Shakeri and Zhang 2016; Maddern et al. 2014)) to further improve their robustness over the potential environmental changes. However, such an application is beyond the scope of this paper and thus it is not further discussed.

2.2 Establishing the Necessity of Sequence-Visual-Word-Vectors

Given an actual pair of loop closing images, there is no guarantee that a sufficient subset of common visual words will be detected in every case since a single image can be subject to aliasing, contain noise and/or moving objects, etc. Thus, it is expected that an absolute thresholding, over the similarity scores between single instances, would fail in detecting some of the trajectory’s loops, or it would also result in many false-positive detections (when a tolerant thresholding is applied). Many existing techniques (e.g. (Mur-Artal and Tardós 2014; Gálvez-López and Tardós 2012; Newman et al. 2006; Milford and Wyeth 2012)) choose to support their detection by accumulating the similarity metrics ($F_I(I_1, I_2)$) from many images acquired close-in-time. In the general case, succeeding frames are treated as sequences of multiple I-VWVs. These groups of I-VWVs (for instance S_1 and S_2) are then compared to the database and assigned with a additive score of $F_S(S_1, S_2) = \sum_{i,j}^{i \in S_1, j \in S_2} F_I(I_i, I_j)$. Although this approach produces effective results, it is limited to a per-frame representation of the environment rather than offering a description of the whole sequence. Considering a simple example as the one presented in Fig. 2a, each visual word of a given scene may not constantly be inside the camera’s frustum or, for any reason, not found by the feature detector (Fig. 2b). This inconsistency entails I-VWVs with considerably uneven values even though they refer to the same scene. As a result, the $F_S(S_1, S_2)$ scores often lead to a false interpretation of the actual sequences’ similarity in a variety of operational scenarios.

With the above notion in mind, the aforementioned approaches can be characterized as sequence matching techniques rather than sequence descriptive ones. As opposed to treating a sequence as the summation of individual matching scores, our method achieves a description vector that contains every visual word found in the scene. Using a computationally efficient approach, for a given sequence of images, the visual words found in every camera measurement are gathered and vote on the respective bin of a common description vector (S-VWV). Note that multiple instances corresponding to the same visual word (i.e. a visual word observed by multiple frames)



(a) A simplistic real word example with the robot passing through the same scene twice (camera pose sets $1.x$ and $2.x$). A different subset of the scene's visual words is observed by each pose.

(b) The input images produce I-VWVs with considerably uneven structures between sequences $1.x$ and $2.x$.

(c) The proposed S-VWVs contain the vocabulary entries found during each sequence in a common description vector as if they were observed by two "super-frames".

Figure 2. The efficiency of the proposed loop closure detection approach with a simplified real word scenario.

are treated as one since they refer to the same feature inside the world. A worth noticing realization here is that the proposed S-VWV to S-VWV matching would present the same results with the earlier approaches only under the false assumption that the used similarity metrics preserved the additive property of linear mapping. As it can be seen in Fig. 2c, our method produces description vectors with better matching properties as it is confirmed by our experimental evaluation (see Section 4).

A similar evaluation of such a unified description has been reported by MacTavish and Barfoot (2014). In their work, sequence-based descriptors were assessed for a variety of different but fixed sized image-groups, while the matching was achieved using a FAB-MAP based probabilistic scheme. Parallel to this notion, our previous work (Bampis et al. 2016) deviated from the probabilistic matching solution and introduced a temporal consistency filtering to further improve the results. As mentioned before, the method in hand incorporates a dynamical sequence distinction technique, allowing for sequences of varying size to be formulated, while additionally, addresses the filtering as a classification procedure. Finally, a unified description was also achieved by the work of Lynen et al. (2014). Although their system allowed for the detected features to be matched against the whole database (regardless the image they belonged to), the method was restricted to operate off-line, after the conclusion of the full trajectory, while the sequence formulation was performed at query time. On the contrary, here the sequence distinction and matching is achieved on-line, as the trajectory escalates, quantizing the searching space through the means of BoVW model.

3 Proposed Methodology

Our on-line LCD algorithm is divided into two main steps while the vocabulary and the filter's kernel coefficients are learned off-line through a training scheme. In the first step of the proposed on-line pipeline, sequence matches are detected, while the individual image associations are extracted in the second one.

3.1 Vocabulary Training

In order to quantize the feature descriptors' space, a visual vocabulary needs to be created. Aiming to offer a real-time implementation, we choose to utilize the binary description of ORB. In an off-line step, a generic set of training descriptors is provided as input to a k-median hierarchical clustering, with k-means++ seeding (Arthur and Vassilvitskii 2007) and Hamming as the distance metric. In accordance to the conclusions drawn by Nister and Stewenius (2006) and Gálvez-López and Tardós (2012) we formulate a vocabulary tree with $L = 6$ levels and $K = 10$ branches per level leading to a total set of $W =$

K^L discrete visual words w_i ($i \in [1, W]$). Two different kinds of multisets need to be defined here, namely $\mathbb{N}_i^D$ and $\mathbb{N}^D$, corresponding to the i-th word's occurrences and the total visual words occurrences in the training dataset, respectively.

3.2 Creating Sequence and Image Descriptors

The main objective of our sequence distinction functionality does not refer to the actual semantics of the observed environment, but rather to identify groups of frames that share common features. To achieve a dynamical partition of the image stream, we utilize the variance of the obtained visual words. At a time instant t, during the sequence's S_t formation, an occupancy vector $V_{S_t}^O$ is retained to keep track of the already seen visual words. This binary vector, represented as $V_{S_t}^O = [w_1^o, \dots, w_i^o, \dots, w_W^o]$, shares the same length with the vocabulary, while each value w_i^o declares the existence ($w_i^o = 1$) or absence ($w_i^o = 0$) of the corresponding word w_i in the current sequence. As the robot moves, the N_f most prominent ORB features are extracted using the oFAST algorithm (the orientation invariant alteration of FAST (Rosten and Drummond 2006) proposed by Rublee et al. (2011)) from every input image. The descriptors are then mapped into visual words through the created vocabulary and marked as NEW (firstly seen during S_t) or OLD (already seen during S_t) by checking their indexes with vector $V_{S_t}^O$. Thus, using a "visual word variance" metric defined as $\sigma_v = N_{NEW} / \left(N_{NEW} + N_{OLD}\right)$, we signal the completion of the current S_t and the beginning of a new S_{t+1} each time $\sigma_v > r_v$, with r_v being a visual word variance above which, the input frame does not share enough of visual words with the rest of the sequence. N_{NEW}, N_{OLD} denote the number of visual words marked as NEW or OLD, respectively. Using the above metric, a new sequence is instigated when the percentage of NEW visual words dominates the entire set of the input image's features. Then, the new vector $V_{S_{t+1}}^O$ is initialized to zero, and the same procedure is repeated for the next sequence. In the case of $\sigma_v \leq r_v$, the $V_{S_t}^O$ vector is updated with the marked as NEW visual words and the following input image is characterized as a member of the current S_t. We additionally force a maximum and minimum visual words' capacity for the sequences, preventing their uncontrolled growth and allowing the $V_{S_t}^O$ vectors to initialize at least some elements. Finally, images that do not contain a minimum number of visual words are rejected as less informative.

Having a completed sequence S with M image-members I_m ($m \in [1, M]$) we now proceed to its description. The following visual words multisets need to be defined. Multisets $\mathbb{N}_i^S$ and $\mathbb{N}_i^{I_m}$ are defined as the i-th visual word's occurrences in sequence S and image I_m, respectively. Additionally, $\mathbb{N}^S$ and $\mathbb{N}^{I_m}$ are defined as the total visual word's occurrences in S and I_m, respectively. The aforementioned multisets are governed by:

$$\mathbb{N}_i^S = \bigcup_{m=1}^{M} \mathbb{N}_i^{I_m} \tag{1}$$

$$\mathbb{N}^S = \bigcup_{m=1}^{M} \mathbb{N}^{I_m} \tag{2}$$

The widely used "term frequency - inverse document frequency" (tf-idf) (Sivic and Zisserman 2003) was selected as a means of defining each visual word's participation and creating the following VWVs: (i) one S-VWV ($\bar{\boldsymbol{v}}^{(S)}$) describing the whole observed visual content of the sequence's respective area and (ii) M I-VWVs ($\bar{\boldsymbol{v}}^{(I_m)}$) for the individual image-members. These descriptors –$\bar{\boldsymbol{v}}^{(S)} = \left[v_1^{(S)}, \dots, v_i^{(S)}, \dots, v_W^{(S)}\right]$ and $\bar{\boldsymbol{v}}^{(I_m)} = \left[v_1^{(I_m)}, \dots, v_i^{(I_m)}, \dots, v_W^{(I_m)}\right]$– are calculated via:

$$v_i^{(S)} = \frac{N_i^S}{N^S} \log \frac{N^D}{N_i^D} \tag{3}$$

$$v_i^{(I_m)} = \frac{N_i^{I_m}}{N^{I_m}} \log \frac{N^D}{N_i^D} \tag{4}$$

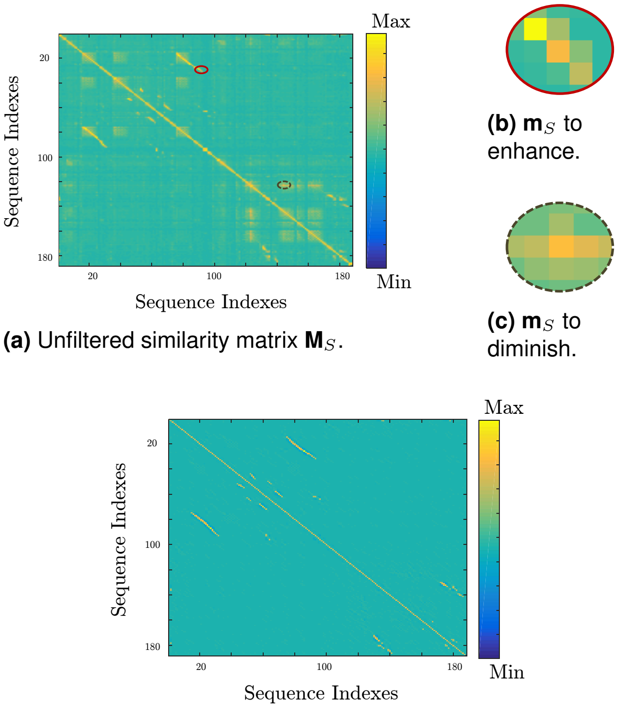
where $N_i^S = \left|\mathbb{N}_i^S\right|$, $N_i^{I_m} = \left|\mathbb{N}_i^{I_m}\right|$, $N^S = \left|\mathbb{N}^S\right|$, $N^{I_m} = \left|\mathbb{N}^{I_m}\right|$, $N_i^D = \left|\mathbb{N}_i^D\right|$ and $N^D = \left|\mathbb{N}^D\right|$, with the notation $|\mathbb{X}|$ representing the cardinality of multiset $\mathbb{X}$. Equation 4 refers to the description of the individual frames, while using eq. 3 we are able to create a global description with better sequence matching capabilities, as described in Section 2.2. Note that the additional computational burden for producing two versions of VWVs is negligible, since the most time consuming part of the process (the tree traversal) is executed only once per visual word.

Finally, to restrict the matching search only between S-VWVs that include mutual visual information, inverted indexing is applied (Jegou et al. 2008). A set of W lists (one for every visual word w_i) is retained, keeping track of sequence indexes whose S-VWVs contain common visual words. Thus, sequence similarity scores are calculated through the inverted indexing list, achieving a reduction of the computational complexity.

3.3 Sequences-to-Sequence Matching

In order to match the individual sequences, we make use of a similarity metric based on $L2$-norm. More specifically, using an $L2$-score similarity between a query (S_q) and a database (S_d) sequence that the inverse indexes indicate:

$$L2\left(\bar{\boldsymbol{v}}_q^{(S)}, \bar{\boldsymbol{v}}_d^{(S)}\right) = 1 - 0.5 \left| \frac{\bar{\boldsymbol{v}}_q^{(S)}}{\left|\bar{\boldsymbol{v}}_q^{(S)}\right|} - \frac{\bar{\boldsymbol{v}}_d^{(S)}}{\left|\bar{\boldsymbol{v}}_d^{(S)}\right|} \right| \tag{5}$$



(a) Unfiltered similarity matrix $\mathbf{M}_S$.

(b) $\mathbf{m}_S$ to enhance.

(c) $\mathbf{m}_S$ to diminish.

(d) Filtered similarity matrix $\mathbf{M}_S^F$.

Figure 3. Impact of the proposed consistency filter on the sequence similarity matrix. The filtered similarity entries corresponding to loop closure events are easily separable from the non-loop closing ones. Note that $\mathbf{M}_S$ and $\mathbf{M}_S^F$ are fully formulated only for visualization purposes. During the on-line algorithm execution, the matrices are only partially computed due to the incorporated inverse indexing.

we obtain a metric that produces higher values as the vectors become more similar. As the trajectory escalates, the calculated $L2$-scores can be arranged to incrementally formulate a similarity matrix $\mathbf{M}_S$ as the one presented in Fig. 3a. This matrix is symmetric with each element containing a corresponding normalized (Gálvez-López and Tardós 2012) $L2\left(\bar{\boldsymbol{v}}_i^{(S)}, \bar{\boldsymbol{v}}_j^{(S)}\right)$ measurement.

A naive approach to detect loop closing sequences would be to apply an absolute thresholding over the values of matrix $\mathbf{M}_S$. With the view to further enhance the cases of sequence matches with indexes that advance concurrently along time ($S_{i\pm k}$-to-$S_{j\pm k}$, $k = 0, 1, 2, ...$), we propose a novel temporal consistency filtering, the coefficients of which are trained in an off-line step. Quantitatively interpreting the temporal constrain, we expect this filter to advance a sequence similarity score $L2\left(\bar{\boldsymbol{v}}_i^{(S)}, \bar{\boldsymbol{v}}_j^{(S)}\right)$ proportionally to the values of $L2\left(\bar{\boldsymbol{v}}_{i\pm k}^{(S)}, \bar{\boldsymbol{v}}_{j\pm k}^{(S)}\right)$. In the same fashion, the filter should penalize the $L2\left(\bar{\boldsymbol{v}}_i^{(S)}, \bar{\boldsymbol{v}}_j^{(S)}\right)$ proportionally to the scores of $L2\left(\bar{\boldsymbol{v}}_{i\pm k_1}^{(S)}, \bar{\boldsymbol{v}}_{j\pm k_2}^{(S)}\right)$, with $(k_1 \neq k_2)$. In other words, the resulting similarity measurement between sequences S_i and S_j will tend to become higher as the respective sub-matrix ($\mathbf{m}_S$) of $\mathbf{M}_S$ –centered around the (i, j) entry– comes closer to a diagonal view (e.g. Fig. 3b) and by analogy lower in cases of temporal inconsistency (e.g. Fig. 3c). Considering an example of window size $w_F = 3$ (corresponding to $k = 1$), those two notions can be efficiently combined into a filter kernel with the following structure:

$$F = \begin{bmatrix} \alpha_0 & -\alpha_1 & -\alpha_2 \\ -\alpha_3 & \alpha_4 & -\alpha_5 \\ -\alpha_6 & -\alpha_7 & \alpha_8 \end{bmatrix} \tag{6}$$

with $\alpha_i \geq 0$. The correlation operation of F with the $\mathbf{M}_S$ matrix results to a more intelligible interpretation ($\mathbf{M}_S^F$), as shown in Fig. 3d. In order to avoid the manual selection of the F coefficients and its size, an off-line supervised training scheme based on cost function minimization is formulated. Another way to consider our consistency filter is as a classifier that separates the loop closing (class LC) similarity measurements from the non-closing ones (class N-LC). For each tested sequence pair $<S_i, S_j>$, this classifier uses the corresponding $\mathbf{m}_S$ neighborhood (i.e. a window around $\mathbf{M}_S\,(i, j)$ of $w_F \times w_F$ size), as a descriptor and decides whether it should fall into category LC or N-LC. Thus, we adopt a logistic-regression approach and we search for a first order multivariate polynomial, with coefficients $\bar{\boldsymbol{\theta}} = [\theta_0, \theta_1, ..., \theta_n]^T$, for which $\bar{\boldsymbol{x}} \cdot \bar{\boldsymbol{\theta}} \geq 0$ indicates the detection of a sequence loop closure event. Note that $\bar{\boldsymbol{x}} = [1, \hat{x}_1, ..., \hat{x}_n]$ denotes the rearrangement of a $\mathbf{m}_S$ sub-matrix's entries into a normalized feature vector format and $n = w_F^2$. The normalization $\hat{x}_i = x_i / max\,(x_i)$, $x_i \in \mathbf{m}_S$ provides the required invariance over any similarity scale. Consequently, the values of θ_1 to θ_n correspond to the filter's coefficients (eq. 6), while $r_s = -\theta_0$ can be characterized as a threshold that should be applied over the $\mathbf{M}_S^F$ entries to identify the loop closing sequences. The final cost function minimization scheme is governed by:

$$\bar{\boldsymbol{\theta}} = \underset{\bar{\boldsymbol{\theta}}}{\operatorname{argmin}}\, J(\bar{\boldsymbol{\theta}}) \tag{7}$$

$$\begin{aligned} J(\bar{\boldsymbol{\theta}}) = -\frac{1}{l_{tr}} \sum_{i=1}^{l_{tr}} \Big[& y_{tr}^{(i)} \log h_\theta(\bar{\boldsymbol{x}}_{tr}^{(i)}, \bar{\boldsymbol{\theta}}) + \\ & + \left(1 - y_{tr}^{(i)}\right) \log \left(1 - h_\theta(\bar{\boldsymbol{x}}_{tr}^{(i)}, \bar{\boldsymbol{\theta}})\right) \Big] \end{aligned} \tag{8}$$

$$h_\theta(\bar{\boldsymbol{x}}, \bar{\boldsymbol{\theta}}) = \frac{1}{1 + e^{\bar{\boldsymbol{x}} \cdot \bar{\boldsymbol{\theta}}}} \tag{9}$$

In eq. 8, l_{tr} denotes the size of the learning set, while $\bar{\boldsymbol{x}}_{tr}^{(i)}$ and $y_{tr}^{(i)}$ denote the individual training feature vectors and their corresponding loop closure ground-truth, respectively.

Since two classes are used, we assign $y_{tr}^{(i)} = 1$ to the training LC elements and $y_{tr}^{(i)} = 0$ to the N-LC ones. The above set of equations corresponds to a standard binary logistic-regression formulation. Looking for a hypothesis vector $\bar{\boldsymbol{\theta}}$ with the characteristics explained before, the sigmoid function of eq. 9 maps the range $\mathbb{R}$ of $\bar{\boldsymbol{x}} \cdot \bar{\boldsymbol{\theta}}$ output into the interval $(0, 1)$. Then, the first summation term of eq. 8 quantifies the cost of $\bar{\boldsymbol{x}} \cdot \bar{\boldsymbol{\theta}} < 0$ for the ground-truth LC training sample, while the second one quantifies the cost of $\bar{\boldsymbol{x}} \cdot \bar{\boldsymbol{\theta}} \geq 0$ for the N-LC ground-truth cases. Finally, the hypothesis vector $\bar{\boldsymbol{\theta}}$ can be achieved by minimizing the total cost (eq. 7) through gradient-descent, with the training samples being already normalized into the interval $[0, 1]$. The selection of logistic-regression as a classification technique is justified due to its high tolerance over unbalanced training samples. As King and Zeng (2001) and Crone and Finlay (2012) pointed out, this effect can be accounted when the training and testing data contain approximately the same amount of LC and N-LC events, as to be further considered during the learning phase in Section 4.1.3.

Moreover, in order to select the window size w_F, we formulate a cross-validation step using another standing-apart set of feature vectors, $\bar{\boldsymbol{x}}_{cv}^{(i)}$. We assess multiple filter size scenarios ($w_F = 2, 3, 4, 5, 6, 7$) corresponding to multiple feature vector's lengths $n = w_F^2$ and we create a $\bar{\boldsymbol{\theta}}_h$ hypothesis for each one of them ($h \in [0, 5]$). Next, the cross-validation error for every $\bar{\boldsymbol{\theta}}_h$ is evaluated using:

$$J_{cv}(\bar{\boldsymbol{\theta}}_h) = \frac{1}{2l_{cv}} \sum_{i=1}^{l_{cv}} \left(h_\theta(\bar{\boldsymbol{x}}_{cv}^{(i)}, \bar{\boldsymbol{\theta}}_h) - y_{cv}^{(i)} \right)^2 \tag{10}$$

while the hypothesis producing the smaller J_{cv} $(\bar{\boldsymbol{\theta}}_h)$ is going to be adopted for the final filter's kernel. Similar attempts to influence the values of $\mathbf{M}_S$ matrix can be found in other techniques as well (Newman et al. 2006; Milford and Wyeth 2012), yet in our case, the filtering is interpreted as a classification approach. It should also be noted that during the on-line execution of our algorithm, the only $\mathbf{m}_S$ sub-matrices that we need to formulate and filter are those indicated by the inverse indexing lists. Thus, the $\mathbf{M}_S$ and $\mathbf{M}_S^F$ matrices retain a sparse representation.

Filtered matching scores overpassing $r_s = -\theta_0$ (or equivalently, matching scores with $\bar{\boldsymbol{x}} \cdot \bar{\boldsymbol{\theta}}_h \geq 0$) are considered to contain loop closure frame candidates and the next step of our method refers to their individual image-members associations. The sequence distinction technique described in Section 3.2 does not ensure that the produced trajectory intervals are going to be aligned between multiple traversals of the same area. Thus, some image-members of the query S_q may actually need to be matched with the members of different neighboring database sequences. For this reason, we allow each S_q to be associated with multiple S_d, as long as they are subsequent.

3.4 Image-to-Image Matching

In order to provide a typical LCD technique, our method should provide image-to-image pairs as a final output. Although we find our sequence matches sufficient enough, so as to detect revisited regions of the trajectory, it is possible for some camera poses to be associated without necessarily observing the same content. One can consider the example of two trajectory tracks that, even though remain parallel and spatially close to each other for the majority of their length, their respective courses slowly deviate until they observe significantly different views. The corresponding two sequences, assigned to those tracks (S_{m_1} and S_{m_2}), are naturally going to be matched despite that their last camera measurements may not correspond to loop closure events. In such a scenario, during the slow deviation of the trajectories, the visual content from both sequences does not drastically change, preventing the activation of our visual word variance constraint and the further segmentation of the sequences. In those cases, a simple "one-to-one" pairing would fail, since the last image-members of S_{m_1} and S_{m_2} should not be considered as loop closures. To address those cases, the individual I-VWVs need to be considered. For a highly accurate SLAM system, it would be sufficient to detect a single pair of loop closing camera poses per sequence match using the highest $L2$-scoring I-VWV pair. Yet, as a general rule (assuming an odometry with low accuracy), we need to seek for as many detections as possible. More specifically, for every image-member of S_{m_1} we seek in its paired sequence S_{m_2} (or paired sequences, if more than one associations were produced by the previous step) for the image that produces the maximum I-VWV $L2$-score. Subsequently, in order to reject image-pairs that cannot be visually associated, a loop closure event is identified if the measured similarity is greater than a threshold r_i. A common practice for many LCD systems (e.g. (Gálvez-López and Tardós 2012; Mur-Artal et al. 2015; Lynen et al. 2014; Bampis et al. 2016)) is to apply a final geometrical-verification test in order to accept a loop closing pair of images. Since such tests are based on the computationally expensive estimation of a valid camera transformation matrix, the r_i threshold must be decided so as to reduce the geometrical-verification steps to the minimum required by the SLAM and the pose-graph optimization technique (Latif et al. 2013).

4 Results

In this section we evaluate the individual components of our system and we compare the achieved overall performance against other state-of-the-art methods. In order to measure

Table 1. Properties of the used datasets.

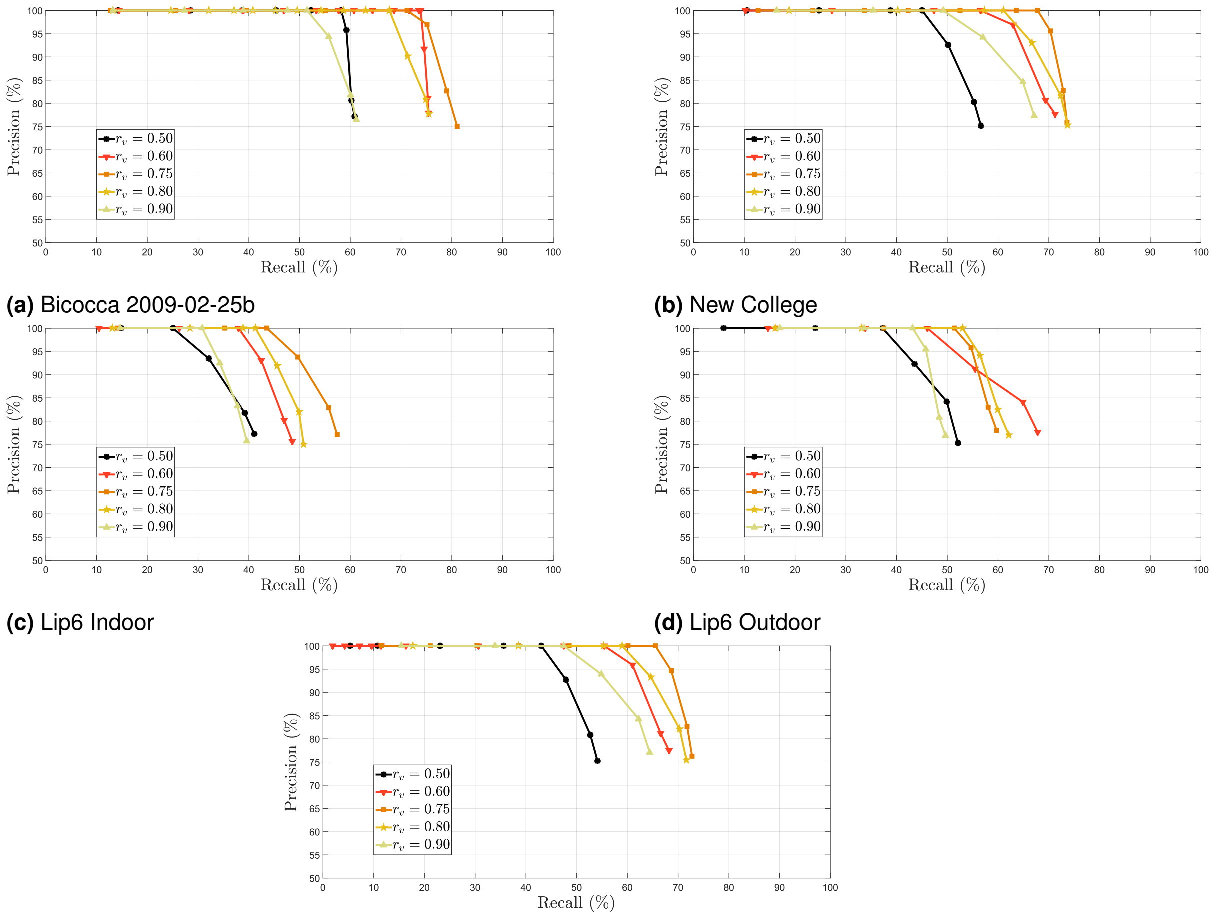
	Dataset	Enviroment	Conditions	Camera position	Image size
Training / Cross-validation	Bovisa 2008-09-01	Indoors & Outdoors	Static	Frontal	320×240
	Bicocca 2009-02-25b	Indoors	Static	Frontal	640×480
	New College	Outdoors	Dynamic	Frontal	512×384
	Lip6 Indoor	Indoors	Static	Frontal	240×192
	Lip6 Outdoor	Outdoors, Urban	Highly dynamic	Frontal	240×192
Testing	Malaga 2009 Parking 6L	Outdoors	Slightly dynamic	Frontal	1024×768
	City Centre	Outdoors, Urban	Dynamic	Lateral	640×480
	KITTI Seq. 00	Outdoors, Urban	Dynamic	Frontal	1241×376
	KITTI Seq. 05	Outdoors, Urban	Dynamic	Frontal	1241×376

the accuracy of an implementation we utilize the Precision-Recall metrics. As a reminder, Precision is defined as the ratio between accurately detected loop closing frames (true-positive) and the total number of detections returned by the method (true-positive plus false-positive). Additionally, Recall is defined as the true-positive detections found, over the total number of loop closing frames that exist in the used dataset (true-positive plus false-negative). For our experiments, we consider a sequence match as true-positive if at least one loop closing camera pose is contained. Nine different datasets (indoors and outdoors) were used for our experiments, namely Bovisa 2008-09-01 (RAWSEEDS 2007-2009) (BV), Bicocca 2009-02-25b (RAWSEEDS 2007-2009) (BC), New College[1] (Smith et al. 2009) (NC), Lip6 Indoor (Angeli et al. 2008) (L6I), Lip6 Outdoor (Angeli et al. 2008) (L6O), Malaga 2009 Parking 6L Blanco et al. (2009) (MG6L), City Centre (Cummins and Newman 2008) (CC), KITTI sequence 00 (Geiger et al. 2013) (KITTI00) and KITTI sequence 05 (Geiger et al. 2013) (KITTI05). Regarding the KITTI dataset we considered only sequences 00 and 05 since, among the rest, they provide the most meaningful loop closure events in urban and long-term operational conditions. Table 1 contains a brief description for every case. Datasets BC through L60 were used as training and cross-validation sets for our method's parameters, while the remaining ones (MG6L through KITTI05) were treated as testing cases measuring the performance of our final system. In such way, the achieved detection accuracy is not directly influenced by the algorithm's optimization, thus offering a fair evaluation. Note that the loop closure ground-truth information for the cases of BC, NC, MG6L and CC was manually-created within the work of Gálvez-López and Tardós (2012). L6I and L6O datasets contain their own ground-truth information as provided by Angeli et al. (2008), while for the KITTI sequences, this information was obtained through the corresponding odometry data.

4.1 Off-line Training and Performance Evaluation

4.1.1 Vocabulary training: Using a vocabulary training set corresponding to a specific environment with limited visual variations, inevitably biases the system's performance to the respective operational conditions. Within the scope of this work, we aim to create a vocabulary able to perform on a variety of indoors and outdoors conditions. In accordance to those terms, the BV dataset was selected as a standing-apart training sample in order to offer an objective evaluation. Using $10K$ frames, a set of $9M$ ORB descriptors was extracted and used as an input to our hierarchical clustering. Thus, a binary vocabulary tree was produced retaining a total of 10^6 discrete visual words as leaf nodes.

4.1.2 Trajectory segmentation: As described in Section 3.2, our algorithm dynamically separates the input image stream into sequences based on the observed visual words' variance. Considering the system's overall performance as a final objective, a validation test based on Precision-Recall metrics was formulated in order to measure the effect of different r_v. Multiple values were selected and assessed on the four training datasets. In this step, the production of Precision-Recall measurements is not straightforward since our system does not create any loop closure output during the sequences' partition. To that end, we temporarily fixed the kernel of our consistency filter on having all its α_i elements equal to zero except from $\alpha_4 = 1$, canceling its effect on the detection and promoting r_s to a means of alternating the Precision-Recall measurements. Figures 4a through 4d show the most informative curves we obtained by considering sequence matches for every training dataset, respectively. The curves shown in Fig. 4e were created accordingly by treating all the datasets as a unified one (just as the same robot traveled through every dataset, one after another). Note that for a range of r_v values



(a) Bicocca 2009-02-25b

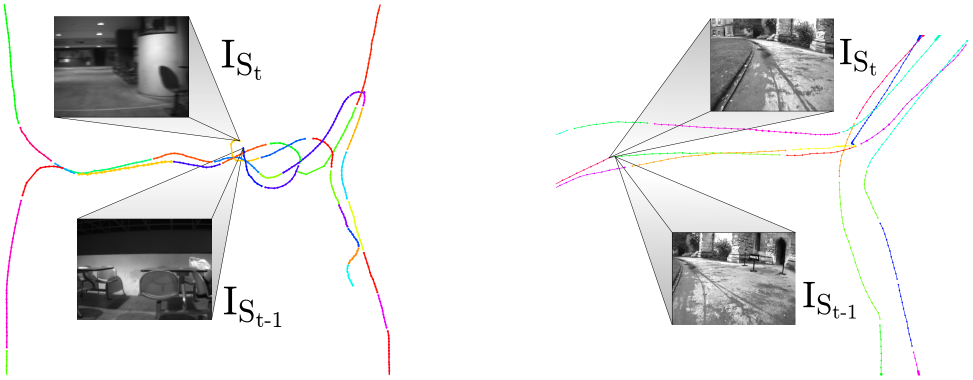
(b) New College

(c) Lip6 Indoor

(d) Lip6 Outdoor

(e) Unified

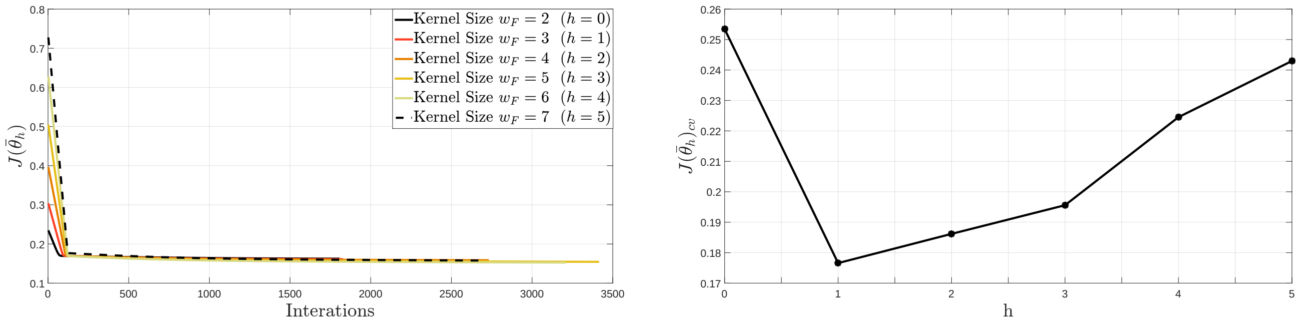
Figure 4. Precision-Recall curves for different r_v values tested on various training datasets. All instances from datasets (a) through (b) are considered to comprise a unified one (e). The best performance is obtained within the range of $[0.6, 0.8]$. Considering the unified dataset, $r_v = 0.75$ corresponds to the highest achieved Recall rate.



(a) Bicocca 2009-02-25b

(b) New College

Figure 5. Resulting sequences (represented with different colors) for the selected value of $r_v = 0.75$. The trajectory is segmented when the input frame does not contain a sufficient number of common visual words with the rest of the sequence. Representative examples of frames signaling the beginning of a new sequence (I_{S_t}), together with the last image-member of the previous sequence ($I_{S_{t-1}}$), are highlighted for each dataset.



(a) Rates of convergence for each one of the tested filter size hypothesis.

(b) Cross-validation error obtained for each one of the tested filter size hypothesis.

Figure 6. Consistency filter training results.

Table 2. Tested sequence distinction approaches.

Method	Recall (%)	Average Execution Time (ms)
Progressive $L2$-score	68.12	0.14
Visual Word Variance	67.54	0.01
Windowed Progressive $L2$-score	69.85	0.77
Windowed Visual Word Variance	68.01	0.12
Image Islands	74.36	8.84

between 0.6 and 0.8 the achieved performance remains relatively stable. Considering the BC dataset, it appears that the best performance is achieved using a visual word variance threshold of $r_v = 0.6$ while for the case of NC the most beneficial case was $r_v = 0.75$. This is owned to the fact that BC corresponds to an indoor environment, therefore visual changes tend to be more severe than NC. Given a specified application scenario (indoors/outdoors, dynamic/non-dynamic, frontal/lateral camera view, etc.), the most appropriate r_v value can be selected accordingly. Though, in this paper we aim for a generic setup and thus the value of $r_v = 0.75$ was adopted. The resulting sequences for the most representative regions of BC and NC (containing turning points and sight changes) are shown in Fig. 5. Note that for the L6I and L6O datasets no odometry ground-truth is provided by Angeli et al. (2008).

The proposed visual word variance metric is not the only kind of measurement considered for our methodology. Other approaches, capable of running in real-time and on-line (while the robot is moving) without the requirement of the whole database beforehand, were also examined. Table 2 presents some of the evaluated techniques together with their respective best-case Recall rates (for 100% Precision) and average execution time, tested on the aforementioned unified training dataset. Method "Progressive $L2$-score" marks the completion of a sequence each time the $L2$-score between the current and the previously acquired input frame becomes smaller than a certain level, while method "Visual Word Variance" refers to the proposed approach that we previously evaluated. Methods "Windowed Progressive $L2$-score" and "Windowed Visual Word Variance" apply an additional averaging sliding window over the two aforementioned metrics. According to them, the mean values of $L2$-score and σ_v are calculated respectively, between the current and the last p input frames. Thus, the most recent sequence is finalized each time the average $L2$-score becomes lower than a certain level ("Windowed Progressive $L2$-score"), or when the average σ_v becomes higher than one ("Windowed Visual Word Variance"). These approaches were selected with the aim to prevent the unnecessary partition of the input stream when an instantaneous change of the view occurs (e.g. instantaneously looking sideways) while the robot is still located in the same scene. Interestingly, the two methods did not provide any considerable advantage to the system's performance. This is owed to the fact that even if a continuous trajectory region is segmented without a semantic meaning, the produced additional sequences can all still be matched to a potentially loop closing non-segmented database entry. The only disadvantage of such a division is the additional processing steps induced by the unnecessary segmentation of the searching space. Yet, as it can be seen in Table 2, the continuous calculations of the extra $L2$-score or σ_v measurements render the window-based approaches as unfavorable, compared to their corresponding straightforward ones. Finally, method "Image Islands" is inspired by the techniques described by Gálvez-López and Tardós (2012) and Milford and Wyeth (2012). The procedure starts with the calculation of the $L2$-scores between the I-VWVs.

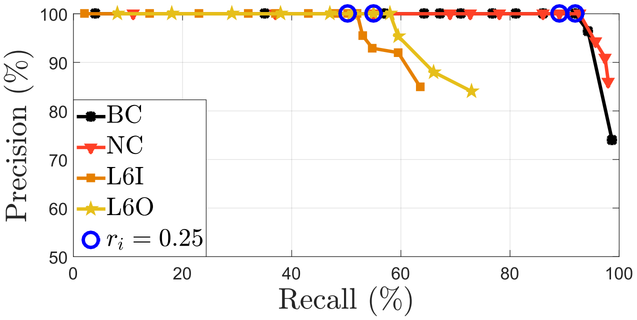


Figure 7. Precision-Recall curves measuring the final performance of the proposed system on every training dataset.

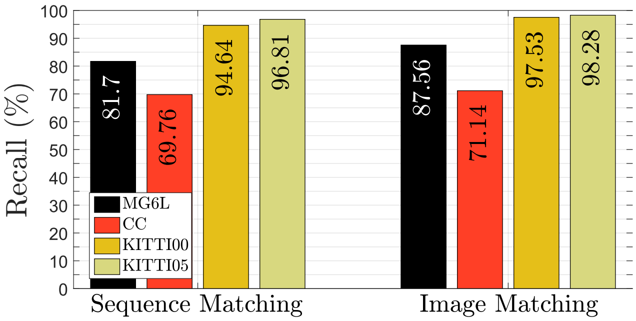


Figure 8. Achieved Recall rates corresponding to 100% Precision for each one of the main matching procedures. The evaluation was performed on the four testing datasets using a fixed parameter set presented in Table 3.

Then, close-in-time sets of images that present considerable similarity scores with the database are grouped in order to create the required sequences. The remaining frames are not employed. The rest of the proposed steps (S-VWV formulations/comparisons, etc.) remain the same. As seen, although the last approach presents higher Recall rates for 100% Precision accuracy, it is still the most costly in terms of execution time. Considering a case of a powerful processing architecture, the "Image Islands" approach would be the most beneficial approach for distinguishing the required sequences. Yet, within the scope of this paper, time efficiency is crucial and thus the proposed visual word variance is adopted since it achieved a nice trade-off between performance and operational frequency.

4.1.3 Temporal consistency filter: The following set of parameters that we need to assess is the coefficients of the proposed temporal consistency filter. Once more, the same four training datasets were selected and the corresponding M_S matrices were formulated using the procedure described in Section 3.3. The concatenation of all four M_S resulted into a training sample that contained a representative ratio between LC (Fig. 3b) and N-LC (Fig. 3c) events allowing the classifier to address analogous cases during its on-line execution. Yet, given a specified operational environment, the estimated hypothesis vector can be accordingly adjusted to fit each particular sample distribution without the need of retraining the whole system, as described by King and Zeng (2001). The sample was further separated into two subsets, namely training ($\bar{\boldsymbol{x}}_{tr}^{(i)}$) and cross-validation ($\bar{\boldsymbol{x}}_{cv}^{(i)}$), containing 70% and 30% of the total loop-closing and non-loop-closing entries, respectively. Using eq. 7 through 9 for every filter size scenario, we obtained six different hypothesis vectors $\bar{\boldsymbol{\theta}}_h$. The rates of convergence for each one of them are shown in Fig. 6a, while Fig. 6b contains their respective cross-validation error. The winning hypothesis $\bar{\boldsymbol{\theta}}_1^w = [\theta_0^w, \theta_1^w, ..., \theta_9^w]^T$ $(h = 1)$ corresponds to a filter size of $w_F = 3$. Thus, the filter's kernel was found using the values of $\theta_1^w, ..., \theta_9^w$ as:

Table 3. Parameter setup.

Tree branches (K)	10
Tree levels (L)	6
ORBs/frame (N_f)	300
Visual Var. (r_v)	0.75
Filter Kernel (F)	$\begin{bmatrix} 2.31 & -0.57 & -1.88 \\ -0.41 & 2.19 & -0.75 \\ -1.83 & -0.34 & 2.15 \end{bmatrix}$
Seq. Matching (r_s)	3.5
Image Matching (r_i)	0.25

$$F = \begin{bmatrix} 2.3088 & -0.5663 & -1.8762 \\ -0.4084 & 2.1938 & -0.7538 \\ -1.8333 & -0.3420 & 2.1512 \end{bmatrix} \tag{11}$$

Additionally, considering that θ_0^w converged into the value of -3.5, we found r_s from $r_s = -\theta_0^w = 3.5$. The filtered similarity matrix $\mathbf{M}_S^F$ in Fig. 3d was produced by applying the kernel of eq. 11 over the $\mathbf{M}_S$ entries of BC dataset. Here, we need to point out that the feature vector converged into a kernel with the same structure as the one of eq. 6, proving that our temporal consistency objective is admissible and achieved.

4.1.4 Overall performance: Using the aforementioned trained filtering, the overall performance of our pipeline was evaluated for each training dataset. By varying threshold r_i and considering image-to-image matches, we acquired the Precision-Recall curves presented in Fig. 7. In order to test our whole methodology on some standing-apart cases, we used the MG6L, CC, KITTI00 and KITTI05 datasets. Figure 8 shows the Recall rates corresponding to 100% Precision for each proposed matching step. The used parameter setup is summarized in Table 3. The

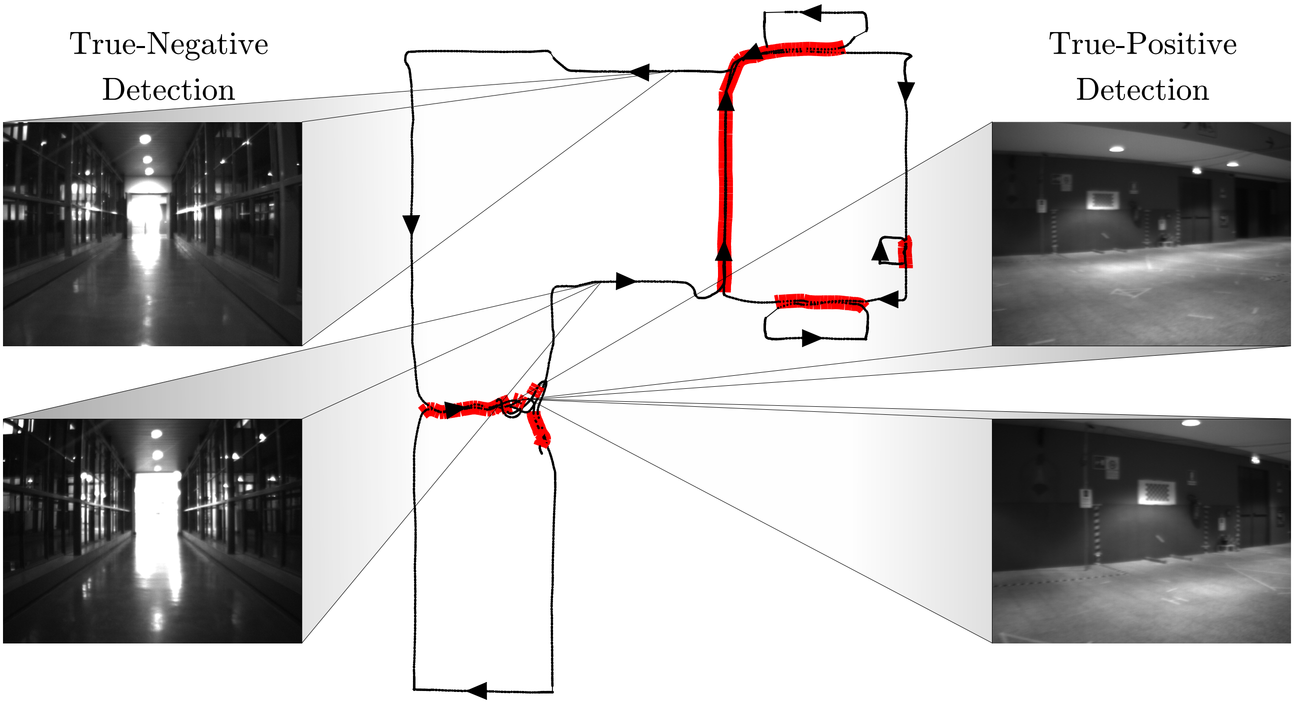


Figure 9. Loop closure detection results on the Bicocca 2009-02-25b dataset. The respective camera poses are marked with red. Representative true-negative and true-positive examples are highlighted.

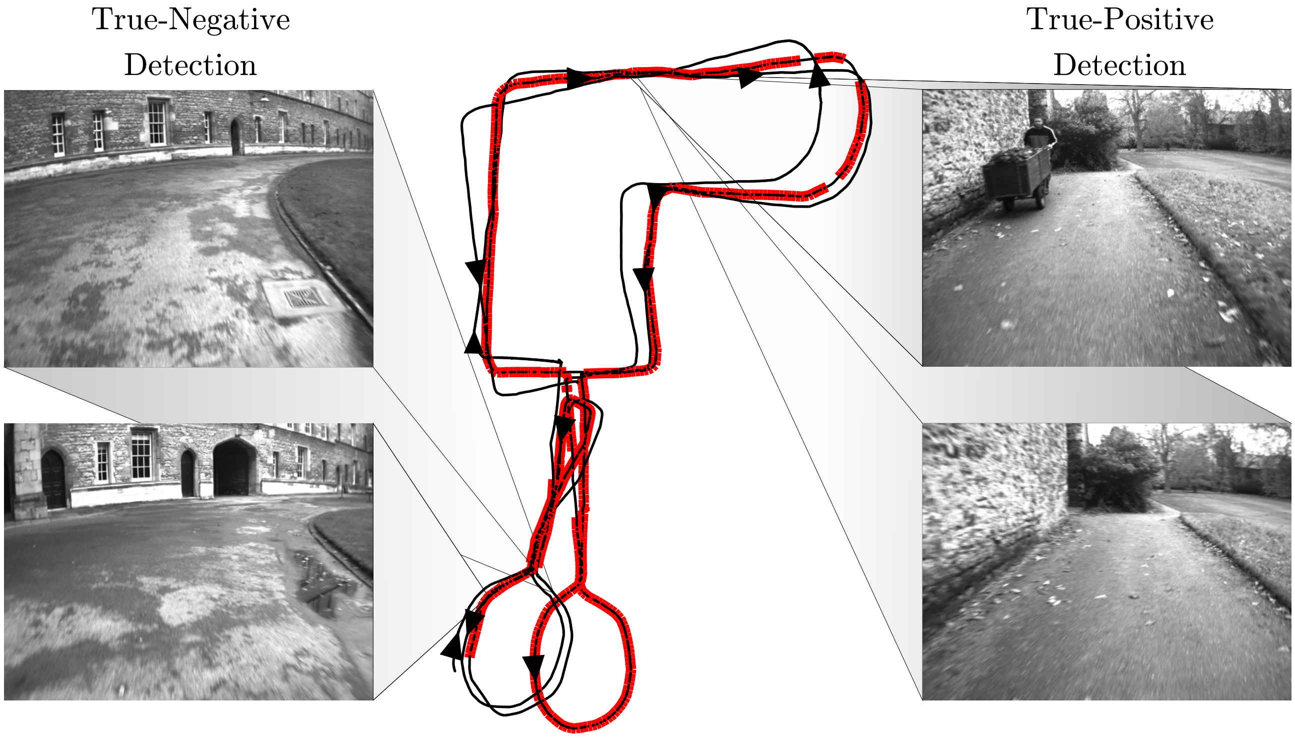


Figure 10. Loop closure detection results on the New College dataset. The respective camera poses are marked with red. Representative true-negative and true-positive examples are highlighted.

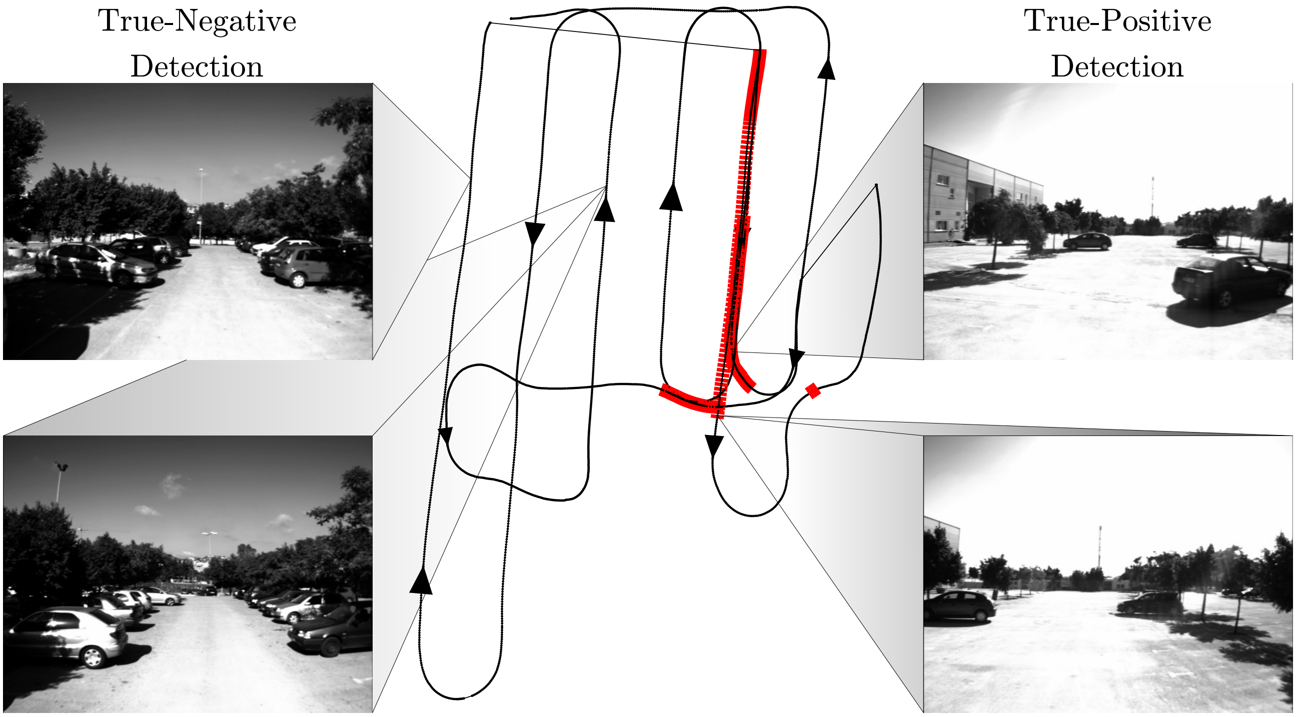


Figure 11. Loop closure detection results on the Malaga 2009 Parking 6L dataset. The respective camera poses are marked with red. Representative true-negative and true-positive examples are highlighted.

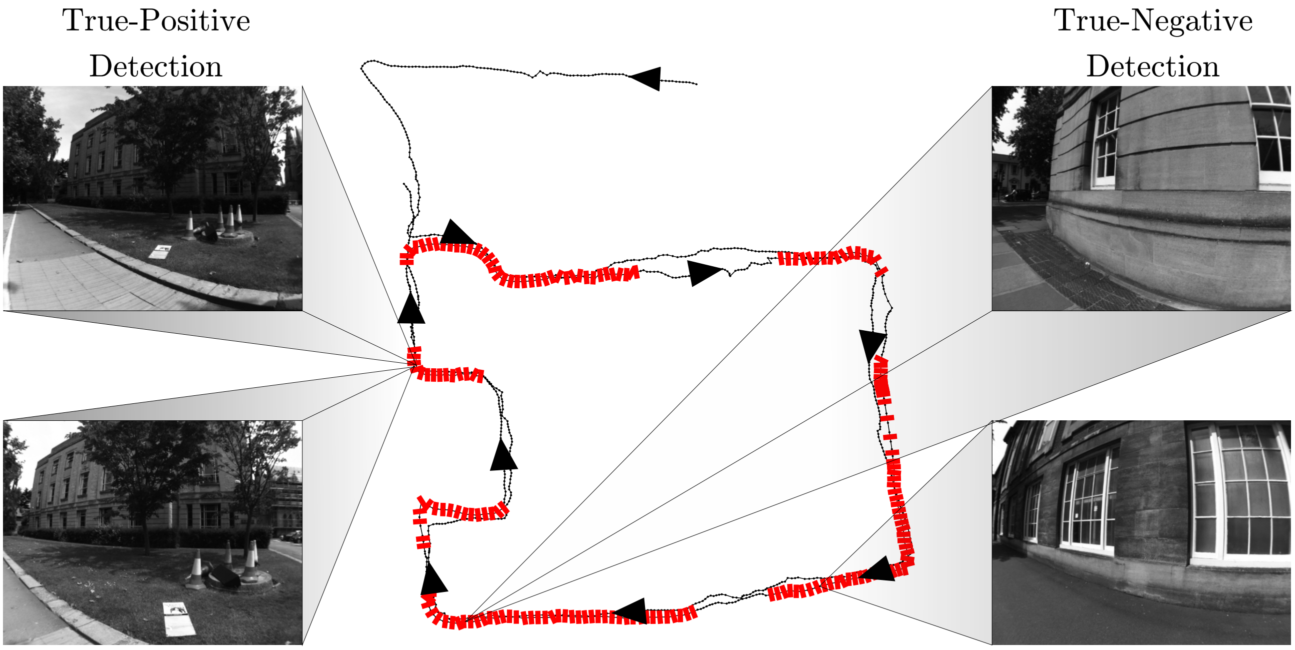


Figure 12. Loop closure detection results on the City Centre dataset. The respective camera poses are marked with red. Representative true-positive and true-negative examples are highlighted.

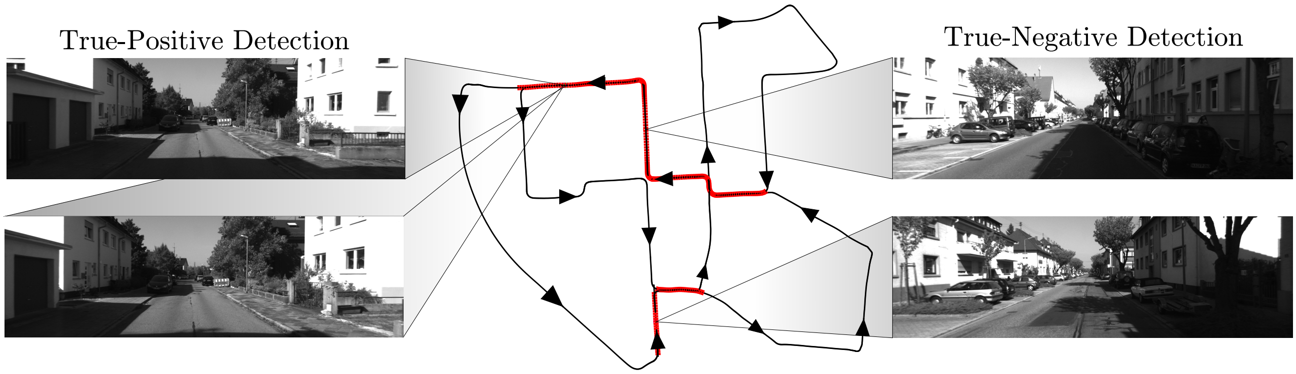


Figure 13. Loop closure detection results on the KITTI sequence 00 dataset. The respective camera poses are marked with red. Representative true-positive and true-negative examples are highlighted.

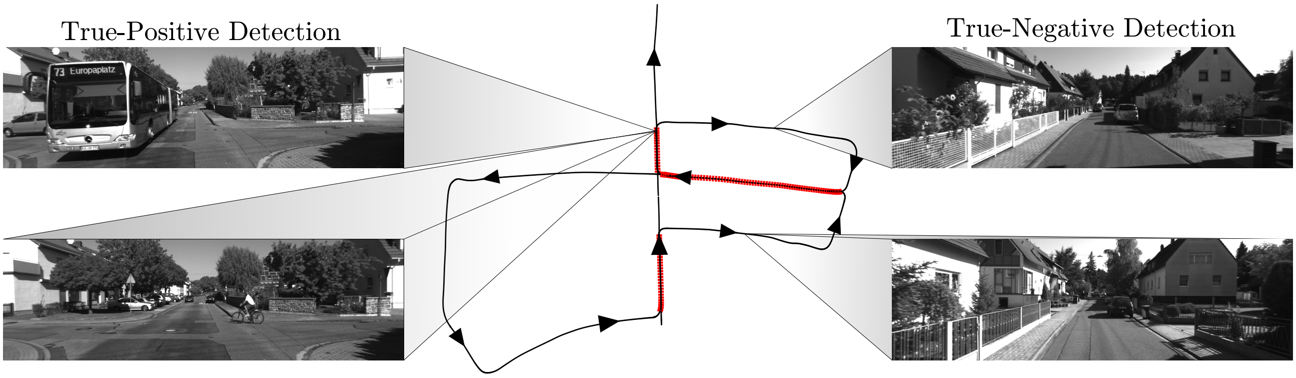


Figure 14. Loop closure detection results on the KITTI sequence 05 dataset. The respective camera poses are marked with red. Representative true-positive and true-negative examples are highlighted.

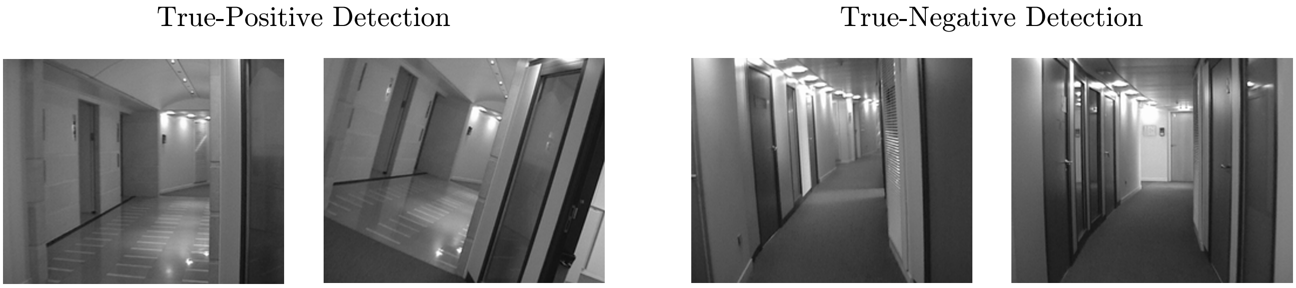


Figure 15. Representative true-positive and true-negative detections on Lip6 Indoor. The dataset does not provide any odometry ground-truth.

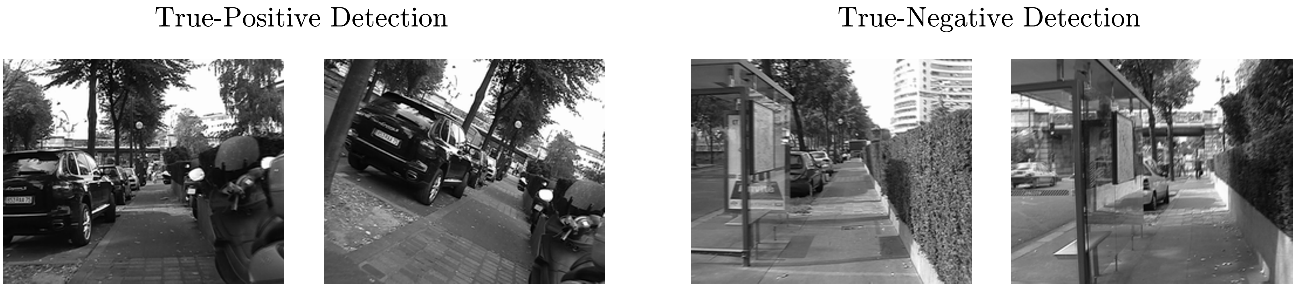


Figure 16. Representative true-positive and true-negative detections on Lip6 Outdoor. The dataset does not provide any odometry ground-truth.

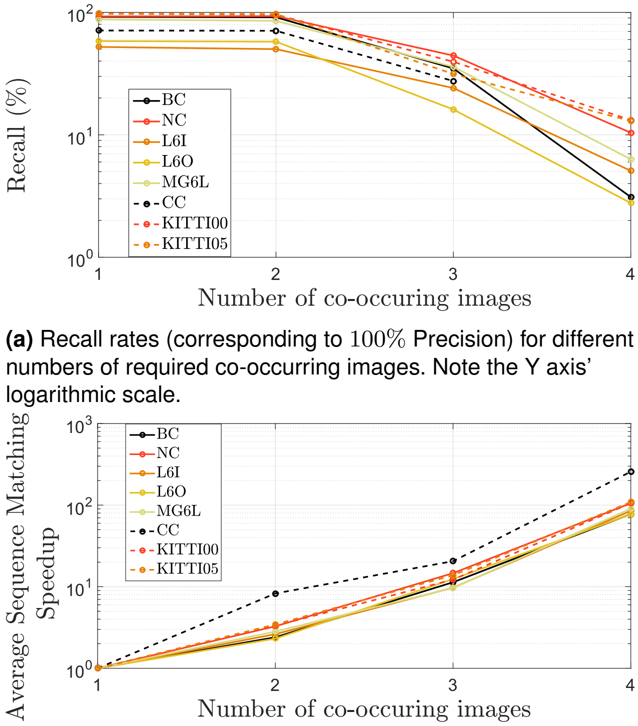
Table 4. Comparative results showing the achieved Recall rates (%) for 100% Precision accuracy. Entries highlighted with bold indicate the best performing approach for each dataset.

	BC	NC	L6I	L6O	MG6L	CC	KITTI00	KITTI05
Cummins and Newman (2011) FAB-MAP 2.0	–	–	–	–	68.52	38.77	–	–
Angeli et al. (2008) SIFT+COLOR	–	–	36.86	23.59	–	–	–	–
Gálvez-López and Tardós (2012) DBoW2	81.20	55.92	–	–	74.75	31.61	–	–
Mur-Artal and Tardós (2014) DBoW2-ORB	76.60	70.29	–	–	81.51	43.03	–	–
Khan and Wollherr (2015) IBuILD	–	–	41.74	25.58	78.13	38.92	–	–
Milford and Wyeth (2012) SeqSLAM	13.90	55.12	20.91	5.63	15.09	42.80	83.54	87.95
Based on Milford and Wyeth (2012) SeqSLAM-BoVW	73.15	85.97	**54.43**	39.95	80.48	60.27	89.29	91.65
Bampis et al. (2016) S-VWV Baseline	78.10	77.55	45.69	51.92	76.78	68.49	81.54	84.80
Proposed	**91.90**	**92.74**	52.22	**58.32**	**87.56**	**71.14**	**96.53**	**97.28**

presented results reveal that our algorithm provides nice scaling capabilities over the selected testing sets. Note that the sequence matching Recall rates correspond to the testing evaluation of the trained logistic-regression classifier. As expected, the highest Recall improvement, obtained through the image matching step, occurred in the MG6L dataset since in that case, sequence tracks with small loop closing overlap are more profound (Fig. 1). Finally, operational examples for each one of the used datasets are presented in Fig. 9 through 16. At this point we need to state once again that L6I and L6O do not offer any odometry ground-truth and that BV does not contain considerable loop closure events.

4.2 Comparative Results

In this subsection, the overall performance of our system is compared against other state-of-the-art techniques. Within the scope of this work, we aim for a solution capable of achieving high-quality LCD results while still retaining a real-time performance for key-frame SLAM applications ($\sim$100-200ms per frame (Strasdat et al. 2010; Mei et al. 2009; Davison et al. 2007)). For this reason, the methods described by Cummins and Newman (2011); Angeli et al. (2008); Milford and Wyeth (2012); Gálvez-López and Tardós (2012); Mur-Artal and Tardós (2014) and Khan and Wollherr (2015), as well as our previous preliminary version (Bampis et al. 2016), were selected for assessment. As it can be seen in Table 4, our approach achieves higher Recall rates than any other tested algorithm tangibly proving the capabilities of the proposed S-VWV to S-VWV matching. Table entries marked with "–" correspond to evaluations not available in the literature, while all the included performance metrics were obtained using a common loop closure ground-truth. Once again, the BV dataset was not tested as it does not present sufficient loop closure events. Among the selected approaches, FAB-MAP 2.0 (Cummins and Newman 2011) and SIFT+COLOR (Angeli et al. 2008) are both considered as golden standards for LCD tasks. Additionally, since for the case of FAB-MAP 2.0 no actual Precision-Recall measurements are provided by Cummins and Newman (2011) regarding the used datasets, the presented performance is obtained from the setup described in work of Gálvez-López and Tardós (2012). In order to evaluate the performance of SeqSLAM algorithm (Milford and Wyeth 2012), one of the most representative sequence-based LCD techniques that groups camera measurements based on their matching similarities, we made use of the OpenSeqSLAM[2] implementation. Since the original version was optimized for addressing the vPR task under changing illumination conditions, rather than identifying revisited places under different viewpoints (MacTavish and Barfoot 2014), it is reasonable that the



(a) Recall rates (corresponding to 100% Precision) for different numbers of required co-occurring images. Note the Y axis' logarithmic scale.

(b) Achieved speedup for different numbers of required co-occurring images. Note the Y axis' logarithmic scale.

Figure 17. Reducing the visual words' multitude during the sequence matching procedure. The X axis refers to the number of frames a visual word needs to co-occur in order to be included in the respective S-VWV. By considering only the visual words co-occurring in more than one frames, we can double the computational frequency of the sequence matching functionality without compromising the system's performance.

SeqSLAM performance is not competitive in many of the tested datasets. To confirm the arguments presented in Section 2.2 and to offer fair comparisons, we additionally implemented a BoVW-based version of SeqSLAM using the same visual vocabulary with our method and computing I-VWV to I-VWV $L2$-norms. This version utilizes the same elements with the proposed algorithm, though differs in its key characteristic by adopting a sequence matching, rather than a sequence description approach. The best performing parameters for each assessed case were identified according to the literature. As it can be seen, the proposed S-VWV description outperforms BoVW-based SeqSLAM for almost every case, with the L6I being the only dataset slightly failing. This is due to the fact that the presented algorithm exploits its highest potentials, as compared to other sequence based techniques, in dynamical environments where the visual words may not be constantly detected by each camera measurement as illustrated in Fig. 2. The case of L6I dataset, though, corresponds to a static environment, with a camera facing forward and traveling through well lighted corridors. Therefore, the proposed unified description is unavailing and thus it performs analogously to the BoVW-based SeqSLAM. Methods DBoW2 (Gálvez-López and Tardós 2012) and DBoW2-ORB (Mur-Artal and Tardós 2014) also accumulate similarity metrics between single frames to form a sequence matching score, though in those cases, image groups are formulated based on their time intervals. Finally, the IBuILD method (Khan and Wollherr 2015) corresponds to dynamical vocabulary creation approach, capable of running in real-time, while the S-VWV Baseline method (Bampis et al. 2016) also incorporates the presented S-VWV based description, yet, lacks an adoptive parameterization since every sequence was distinguished based on a fixed traversed distance.

5 Algorithm Efficiency and Mobile Device Implementation

In addition to effective similarity properties, our first level of sequence-to-sequence matching allows for a more efficient implementation in terms of computational complexity. More specifically, one can consider the proposed S-VWV comparisons as a means of rejecting large trajectory regions that are different in the general view, followed by a supplementary examination of the individual image-members. Taking into account an example of a long traversed route with n_c camera poses, we can evaluate the number of comparisons that a generic image-to-image based method and the proposed one need to perform. In the first case, each individually generated I-VWV have to be compared with the whole database in order to find the most similar match, leading to a total of up to ${n_c}^2/2$ similarity measurements. On the contrary, our method is capable of detecting revisited trajectory regions using only $\left(n_c/\bar{M}\right)^2/2$ comparisons, where $\bar{M}$ denotes the average size of the produced sequences. Subsequently, only for the identified revisited scenes, the individual image associations corresponding to loop closures are produced. Even for the extreme case in which a match has been found for all the sequences, the additional computational steps for producing image pairs are in the order of $n_c * \bar{M}$, which accumulated to $\left(n_c/\bar{M}\right)^2/2$, are still fewer than the ${n_c}^2/2$ required by the brute-forcing image-to-image matching for prolonged datasets. Naturally, the inverse indexes are going to reduce the execution time of both approaches in the same manner. Thus, their effect is omitted from this example.

The nature of the presented sequence description algorithm provides an additional means for further reducing the computational complexity of the S-VWV matching functionality. During the formulation of each sequence, it is natural to expect a set of visual words to be observed by multiple image-members. Such visual words

Table 5. Time-profiling for $15K$ images of New College dataset.

		Time (ms/frame)			
		Mean	Std	Min	Max
Feature Extraction	oFAST detection	11.38	5.05	4.59	30.08
	ORB description	7.21	2.45	3.63	21.21
	Total	18.58	4.84	8.29	52.53
VWVs Calculation	Tree Traversal	3.84	0.76	1.43	17.49
	Form I-VWV	0.05	0.02	0.03	0.61
	Form S-VWV	0.07	0.05	0.001	0.78
	Total	3.96	0.77	1.43	17.49
Matching	Inverse Indexing	0.09	0.34	0.83	3.55
	Seq. Matching	4.87	7.72	0.66	12.47
	Image Matching	0.02	0.26	0.20	8.54
	Total	4.99	8.04	2.81	12.56
Whole application		27.44	8.74	13.66	58.01

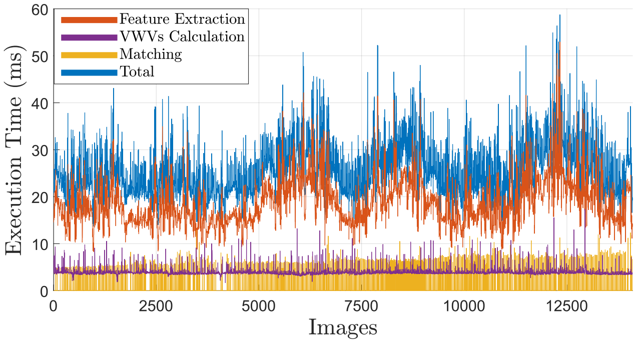


Figure 18. Execution time per image for each one of the main processing steps of the proposed algorithm, measured for $15K$ images of the New College dataset.

typically correspond to better localized ORB descriptors in the vocabulary clustering space and they are originated from more representative sequence landmarks. Thus, a significant speedup can be achieved, during the sequence matching procedure, by only considering visual words observed from more than one image-members. As it can be seen in Fig. 17, the exclusion of visual words occurring in only one image has minor effect on the achieved Recall rates, while at the same time increases the computational frequency of the sequence matching functionality by more than 100%.

Since our system is capable of detecting loop closure events using only a monocular camera while retaining a low computational complexity, an application for mobile devices was developed in order to provide a complete and fully functional system. Using the Android Development Kit provided by Google's Project Tango (Google 2017) we implemented a C++ based algorithm[3] utilizing the parallelization capabilities of the ARM-NEON co-processor. The application was specifically designed so as to respect the limitations in terms of available RAM and processing power that a mobile device induces. In particular, we used a sparse representation for each of the description vectors (S-VWVs and I-VWVs) while the ARM-NEON co-processor, built upon the SIMD architecture, undertook the parallelizable procedures, e.g. the ORB features detection/description and the Hamming distance calculation for the vocabulary tree traversal. In addition, we assigned the sequence matching procedure on a dedicated thread running concurrently with the rest of our algorithm (but not on a different core). Since the sequence matching is triggered each time a new sequence is completed and not for every input frame, it is natural to burden the execution time unevenly. During the formulation of a new sequence, no similarity score calculations are performed allowing the implementation to run in less than 25ms. Every time a new sequence is completed, an instantaneous overhead appears, preventing our application from running in constant time. Thus, using two individual threads in a pipeline manner, the most recently created sequence is compared to the database and the loop closures –if any– are detected while the formulation of a new S-VWV occurs. In this way, even though we do not achieve any speedup over the total execution time, the necessary calculations are evenly spread along the acquisition of every input image. Extension 1 shows an instance of our application running on the Tango device in a real-world scenario.

Using the implementation described above, we formulated a time-profiling experiment on the biggest tested dataset. Table 5 presents the execution time obtained by each of the processing stages for $15K$ images of the NC dataset (padding the available RAM of the device). As one can observe, the most demanding procedure of our algorithm is the Feature Extraction, which can be considered as pre-computed for many SLAM architectures. Although the Matching procedure may require a maximum of 12.56ms, only its average cost is perceptible by the whole application due to the used pipelining. In addition, by forcing a serialized execution of the two pipelining threads, we obtained the timing measurements presented in Fig. 18 for each one of the main algorithm's procedures.

6 Discussion

In this paper, a novel sequence description technique is proposed which allows us for the first time to combine the entire visual information of a place into a single descriptor, while still retaining a feature based approach. The newly introduced module of S-VWV allows for a two-layered LCD system that firstly recognizes revisited scenes and later associates the individual loop closing camera poses. Instead of adopting a spatiotemporal approach so as to distinguish the individual sequences, an efficient visual word variance metric was selected, separating the input stream with respect to the visual content's alterations. In addition, taking into account the temporal consistency constraint that successively recognized camera measurements need to obey, a novel similarity filtering was proposed. By considering the filter's kernel as a binary classifier, its coefficients were learned using a cost function minimization scheme. Finally, an implementation of the presented algorithm was developed and tested on a mobile device, utilizing the parallelization properties of the SIMD architecture and proving the computational efficiency of our method.

By successfully describing an image sequence, rather than matching single instances and accumulating their similarity scores, our method provides a unified solution. Yet, its true potentials are fully exploited when the executed trajectory includes long loop closing tracks. In contrast with the rest of the evaluated datasets, BC contains many sharp and rapid turning movements (especially during the end of the traversed route as shown in Fig. 5a) causing the used sequence distinction function to over-segment the trajectory into scenes that actually correspond to the same place. An analogous example can be considered with a camera rotating around its yaw axis and causing the formulation of multiple sequences, while still remaining into the same physical place. In those cases, the presented method descends into a simple BoVW-based approach, with the formulated S-VWVs retaining a similar structure to the corresponding I-VWVs. Even though this aimless segmentation does not affect the system's performance (see Section 4.1.2), it may unnecessarily increase the computational complexity of the sequence matching procedure by expanding the searching database space. Possible ways to avoid such extreme cases are the tested "Windowed Progressive $L2$-score" and "Windowed Visual Word Variance" approaches. A Scale-Space filtering (Witkin 1984) over the above measurements can also be useful in order to achieve a dynamical window size. Having as principal objective the identification of image groups that share a sufficient number of common visual words, further co-visibility and graph-connectivity measurements can be applied and evaluated (Chandrasekhar et al. 2014b; Erkent and Bozma 2015), especially in cases of off-line map optimization or RL scenarios, where the traversed map is fully formulated beforehand (Chandrasekhar et al. 2014a; Johns and Yang 2011; Moon et al. 2016).

Given our choice of ORB local features, a scale and rotation invariant description is achieved. Such a mechanism, though, does not promote a direction invariant LCD system, especially when the respective robot is equipped with a monocular frontal-oriented camera. This is owned to the fact that the detected patches' appearance, although originated from the same place, changes significantly when observed from two opposite directions, leading many vPR systems to exclude or fail in identifying those revisited paths (Lynen et al. 2014; Fraundorfer et al. 2007). Even though such cases are not commonly encountered, thus not handled by the proposed and other sequence-based vPR techniques (Newman et al. 2006), possible solutions include the utilization of lateral-oriented/panoramic cameras (Lynen et al. 2014; Agarwal et al. 2015), or the estimation of the detected patches' orientation (Davison et al. 2007) in order to predict their appearance changes. Using the proposed system on such events would additional imply an appropriate temporal consistency filter structure. Thus, our filter needs to be retrained accordingly (probably converging into a higher window size) or applied subsequently twice, once with the structure of eq. 11 and once with the same kernel flipped on both axes, in order to additionally promote $\mathbf{m}_S$ sub-matrices with high antidiagonal similarity metrics.

An important characteristic of the introduced S-VWV based LCD approach is its fundamental generality. In contrast with the sequence matching approaches (that accumulate similarity metrics between multiple images), the proposed descriptor can efficiently adapt to a distributed architecture, such as the recent work of Cieslewski and Scaramuzza (2017), and allow for a decentralized vPR system of multiple agents. Additionally, since such a description vector can be combined with any kind of vocabulary, an extension of our work would be to utilize visual words that present invariance over illumination or other environmental changes (Linegar et al. 2016; McManus et al. 2015; Yue-Hei Ng et al. 2015; Lee et al. 2013). As a final thought, the notion of a sequence description could serve as a basis for introducing new variables in the LCD procedure. Approaches capable of quantizing time depended measurements into the S-VWVs, such as the optical flow or the robot's ego-motion, can be investigated with the view to further assist the matching functionality.

Appendix A: Index to multimedia Extensions

Table 6. Multimedia Extensions

Extension	Media type	Description
1	Video	Operational Example on a Mobile Device

Notes

1. An improved odometry, found at Smith et al. (2009) website, was assigned to NC dataset for visualization purposes.

2. The OpenSeqSLAM implementation can be found at http://openslam.org/openseqslam.html.
3. The reader can visit https://github.com/loukbabi/PREVIeW to download and review two C++ versions of our algorithm, viz. desktop-based and mobile-based, with the acronym “PREVIeW - Place Recognition with unifiEd sequence VIsual Words”.

Acknowledgements

The ORB feature extraction ARM-NEON implementation was developed at MARS Lab UMN and funded by Google’s Project Tango. The authors would also like to thank Dorian Gálvez-López for providing their manually-created loop closure ground-truth regarding many of the used datasets.

References

Agarwal P, Burgard W and Spinello L (2015) Metric localization using google street view. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 3111–3118.

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

Arroyo R, Alcantarilla PF, Bergasa LM and Romera E (2015) Towards Life-Long Visual Localization using an Efficient Matching of Binary Sequences from Images. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. pp. 6328–6335.

Arroyo R, Alcantarilla PF, Bergasa LM and Romera E (2016) Fusion and binarization of CNN features for robust topological localization across seasons. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 4656–4663.

Arthur D and Vassilvitskii S (2007) k-means++: The advantages of careful seeding. In: *Proceedings of the ACM-SIAM symposium on Discrete algorithms*. pp. 1027–1035.

Bampis L, Amanatiadis A and Gasteratos A (2016) Encoding the description of image sequences: A two-layered pipeline for loop closure detection. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 4530–4536.

Bay H, Tuytelaars T and Van Gool L (2006) SURF: Speeded Up Robust Features. In: *Proceedings of the European Conference on Computer Vision*. pp. 404–417.

Blanco JL, Moreno FA and Gonzalez J (2009) A collection of outdoor robotic datasets with centimeter-accuracy ground truth. *Autonomous Robots* 27(4): 327–351.

Chandrasekhar V, Min W, Li X, Tan C, Mandal B, Li L and Hwee Lim J (2014a) Efficient retrieval from large-scale egocentric visual data using a sparse graph representation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. pp. 527–534.

Chandrasekhar V, Tan C, Min W, Liyuan L, Xiaoli L and Hwee LJ (2014b) Incremental graph clustering for efficient retrieval from streaming egocentric video data. In: *Proceedings of the IEEE International Conference on Pattern Recognition*. pp. 2631–2636.

Cieslewski T and Scaramuzza D (2017) Efficient decentralized visual place recognition using a distributed inverted index. *IEEE Robotics and Automation Letters* 2(2): 640–647.

Crone SF and Finlay S (2012) Instance sampling in credit scoring: An empirical study of sample size and balancing. *International Journal of Forecasting* 28(1): 224–238.

Cummins M and Newman P (2008) FAB-MAP: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research* 27(6): 647–665.

Cummins M and Newman P (2011) Appearance-only SLAM at large scale with FAB-MAP 2.0. *The International Journal of Robotics Research* 30(9): 1100–1123.

Davison AJ, Reid ID, Molton ND and Stasse O (2007) MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29(6): 1052–1067.

Erkent Ö and Bozma HI (2015) Long-term topological place learning. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. pp. 5462–5467.

Fei X, Tsotsos K and Soatto S (2016) A Simple Hierarchical Pooling Data Structure for Loop Closure. In: *Proceedings of the European Conference Computer Vision*. pp. 321–337.

Folkesson J and Christensen H (2004) Graphical SLAM-a self-correcting map. In: *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1. pp. 383–390.

Fraundorfer F, Engels C and Nistér D (2007) Topological mapping, localization and navigation using image collections. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 3872–3877.

Gálvez-López D and Tardós JD (2012) Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics* 28(5): 1188–1197.

Geiger A, Lenz P, Stiller C and Urtasun R (2013) Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research* 32(11): 1231–1237.

Google (2017) Project tango. URL `https://developers.google.com/tango/hardware/tablet`.

Grisetti G, Kümmerle R, Stachniss C and Burgard W (2010) A tutorial on graph-based SLAM. *Intelligent Transportation Systems Magazine* 2(4): 31–43.

Jegou H, Douze M and Schmid C (2008) Hamming embedding and weak geometric consistency for large scale image search. In: *Proceedings of the European Conference on Computer Vision*. pp. 304–317.

Johns E and Yang GZ (2011) From images to scenes: Compressing an image cluster into a single scene model for place recognition. In: *Proceedings of the IEEE International Conference on Computer Vision*. pp. 874–881.

Khan S and Wollherr D (2015) IBuILD: Incremental bag of binary words for appearance based loop closure detection. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. pp. 5441–5447.

King G and Zeng L (2001) Logistic regression in rare events data. *Political Analysis* : 137–163.

Klein G and Murray D (2007) Parallel tracking and mapping for small AR workspaces. In: *IEEE and ACM International Symposium on Mixed and Augmented Reality*. pp. 225–234.

Konolige K, Bowman J, Chen J, Mihelich P, Calonder M, Lepetit V and Fua P (2010) View-based maps. *The International Journal of Robotics Research* 29(8): 941–957.

Labbe M and Michaud F (2013) Appearance-based loop closure detection for online large-scale and long-term operation. *IEEE Transactions on Robotics* 29(3): 734–745.

Latif Y, Cadena C and Neira J (2013) Robust loop closing over time for pose graph slam. *The International Journal of Robotics Research* 32(14): 1611–1626.

Lee JH, Zhang G, Lim J and Suh IH (2013) Place recognition using straight lines for vision-based slam. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. pp. 3799–3806.

Lim H, Lim J and Kim HJ (2014) Real-time 6-dof monocular visual slam in a large-scale environment. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. pp. 1532–1539.

Linegar C, Churchill W and Newman P (2016) Made to measure: Bespoke landmarks for 24-hour, all-weather localisation with a camera. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. pp. 787–794.

Lowe DG (2004) Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60(2): 91–110.

Lowry S, Sünderhauf N, Newman P, Leonard JJ, Cox D, Corke P and Milford MJ (2016) Visual Place Recognition: A Survey. *IEEE Transactions on Robotics* 32(1): 1–19.

Lynen S, Bosse M, Furgale P and Siegwart R (2014) Placeless place-recognition. In: *Proceedings of the IEEE International Conference on 3D Vision*, volume 1. pp. 303–310.

MacTavish K and Barfoot TD (2014) Towards hierarchical place recognition for long-term autonomy. In: *Proceedings of the IEEE International Conference on Robotics and Automation, Visual Place Recognition in Changing Environments Workshop*. pp. 1–6.

Maddern W, Stewart A, McManus C, Upcroft B, Churchill W and Newman P (2014) Illumination invariant imaging: Applications in robust vision-based localisation, mapping and classification for autonomous vehicles. In: *Proceedings of the IEEE International Conference on Robotics and Automation, Visual Place Recognition Changing Environments Workshop*.

McManus C, Upcroft B and Newman P (2015) Learning place-dependant features for long-term vision-based localisation. *Autonomous Robots* 39(3): 363–387.

Mei C, Sibley G, Cummins M, Newman PM and Reid ID (2009) A constant-time efficient stereo SLAM system. In: *Proceedings of the British Machine Vision Conference*. pp. 1–11.

Milford MJ and Wyeth GF (2012) SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. pp. 1643–1649.

Moon H, Lee JH, Lee S and Suh IH (2016) Effective place scene clustering using straight lines. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 487–494.

Mur-Artal R, Montiel JMM and Tardos JD (2015) ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics* 31(5): 1147–1163.

Mur-Artal R and Tardós JD (2014) Fast relocalisation and loop closing in keyframe-based slam. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. pp. 846–853.

Newman P, Cole D and Ho K (2006) Outdoor SLAM using visual appearance and laser ranging. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. pp. 1180–1187.

Nister D and Stewenius H (2006) Scalable recognition with a vocabulary tree. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2. pp. 2161–2168.

Piniés P, Paz LM, Gálvez-López D and Tardós JD (2010) CI-Graph simultaneous localization and mapping for three-dimensional reconstruction of large and complex environments using a multicamera system. *Journal of Field Robotics* 27(5): 561–586.

RAWSEEDS (2007-2009) Robotics Advancement through Web-publishing of Sensorial and Elaborated Extensive Data Sets (Project FP6-IST-045144). URL http://www.rawseeds.org/rs/datasets.

Rosten E and Drummond T (2006) Machine learning for high-speed corner detection. In: *Proceedings of the European Conference on Computer Vision*. pp. 430–443.

Rublee E, Rabaud V, Konolige K and Bradski G (2011) ORB: an efficient alternative to SIFT or SURF. In: *Proceedings of the IEEE International Conference on Computer Vision*. pp. 2564–2571.

Schindler G, Brown M and Szeliski R (2007) City-scale location recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 1–7.

Shakeri M and Zhang H (2016) Illumination invariant representation of natural images for visual place recognition. In:

Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 466–472.

Sivic J and Zisserman A (2003) Video Google: A text retrieval approach to object matching in videos. In: *Proceedings of the IEEE International Conference on Computer Vision*. pp. 1470–1477.

Sizikova E, Singh VK, Georgescu B, Halber M, Ma K and Chen T (2016) Enhancing place recognition using joint intensity-depth analysis and synthetic data. In: *Proceedings of the European Conference Computer Vision, Workshop*. pp. 901–908.

Smith M, Baldwin I, Churchill W, Paul R and Newman P (2009) The new college vision and laser data set. *The International Journal of Robotics Research* 28(5): 595–599.

Strasdat H, Montiel J and Davison AJ (2010) Scale Drift-Aware Large Scale Monocular SLAM. In: *Proceedings of the Robotics: Science and Systems*. p. 5.

Sünderhauf N, Shirazi S, Dayoub F, Upcroft B and Milford MJ (2015a) On the performance of ConvNet features for place recognition. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 4297–4304.

Sünderhauf N, Shirazi S, Jacobson A, Dayoub F, Pepperell E, Upcroft B and Milford M (2015b) Place recognition with ConvNet landmarks: Viewpoint-robust, condition-robust, training-free. *Proceedings of Robotics: Science and Systems* .

Thrun S and Montemerlo M (2006) The graph SLAM algorithm with applications to large-scale mapping of urban structures. *The International Journal of Robotics Research* 25(5-6): 403–429.

Valgren C and Lilienthal AJ (2010) SIFT, SURF & seasons: Appearance-based long-term localization in outdoor environments. *Robotics and Autonomous Systems* 58(2): 149–156.

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

Witkin A (1984) Scale-space filtering: A new approach to multi-scale description. In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 9. pp. 150–153.

Wolf J, Burgard W and Burkhardt H (2005) Robust vision-based localization by combining an image-retrieval system with Monte Carlo localization. *IEEE Transactions on Robotics* 21(2): 208–216.

Yue-Hei Ng J, Yang F and Davis LS (2015) Exploiting local features from deep networks for image retrieval. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. pp. 53–61.