High Order Visual Words for Structure-Aware and Viewpoint-Invariant Loop Closure Detection

Loukas Bampis, Angelos Amanatiadis and Antonios Gasteratos

***Abstract*— In the field of loop closure detection, the most conventional approach is based on the Bag-of-Visual-Words (BoVW) image representation. Although well-established, this model rejects the spatial information regarding the local feature points' layout and performs the associations based only on their similarities. In this paper we propose a novel BoVW-based technique which additionally incorporates the operational environment's structure into the description, treating bunches of visual words with similar optical flow measurements as single similarity votes. The presented experimental results prove that our method offers superior loop closure detection accuracy while still ensuring real-time performance, even in the case of a low power consuming mobile device.**

I. Introduction

A conditio sine qua non for a modern autonomous robotic system is the functionality of Simultaneous Localization and Mapping (SLAM). A standard SLAM algorithm usually entails a localization engine, capable of estimating the position of a given robot, and a mapping engine responsible for maintaining the environment's representation. These two engines are interdependent, with their outputs formulating a pose-graph representation of the explored world. In order to produce more accurate results, a loop closure detection mechanism is usually incorporated into the system that creates new edge constraints between revisited pose nodes [1], [2], [3]. These additional constraints can be used in an on-line [4] or post-processing [5], [6] manner, in order to further improve the overall estimation using a cost function minimization technique, e.g. Bundle Adjustment (BA) [7].

Due to loop closure's effectiveness over the SLAM procedure, a variety of techniques has been introduced in recent literature, each of which addresses the task in a different approach. According to the work described in [8], all the proposed methods can be classified into three main categories based on the associated elements' nature. The techniques laying into the first category seek for similarities between the local sub-maps created from each pose in order to create additional edge-constrains. On the contrary, methods that fall into the second category attempt to create links between poses by associating the corresponding images to the overall generated map. Finally, the last category includes appearance-based loop closure detection techniques that aim to create edges based on the image similarities themselves, providing better scaling for long trajectories cases [8]. As part of the last category, the Bag of Visual Words (BoVWs) model has been successfully applied in a wide variety of robotic vision applications, offering excel accuracy, as well as computational and memory management performances. The BoVW model addresses the description as a bottom-up procedure by quantizing the detected local features' descriptor space and forming a histogram of occurrences. While this description is very appealing for real-time applications, it rejects the spatial arrangement information of the detected local feature points and considers only their individual classes.

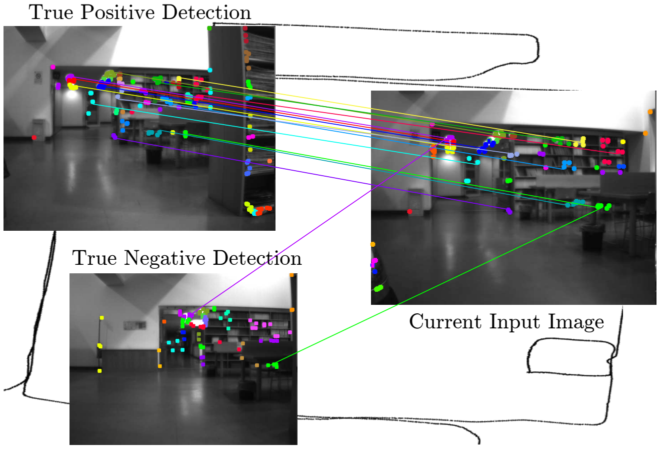


Fig. 1: Loop closure detection using SVHVs on the Bicocca 2009-02-25b [9] dataset. The proposed matching scheme produces more consistent VW associations due to the additional information of the observed environment's structure.

In the typical case, each image is transformed into a BoVW histogram representation of reduced dimensionality, with images that correspond to a common map location tending to produce high similarity metrics (e.g. L2/L1-scores). Many loop closure detection techniques [10], [4], [11], [12] integrate a geometrical verification step as a back-end procedure for rejecting potential false-positive matches that the BoVW model may produce. Such geometrical verification tests are mainly based on the calculation of a computationally expensive image or camera transformation between the associated poses (e.g. 8-point RANSAC), increasing the computational burden and limiting the real-time performance.

The proposed research realizes a novel image description and matching scheme that advances the VW's matches with consistent spatial displacement. Using additional information derived from the normal sequential acquisition of the image stream in an autonomous robotic system, we are

Authors are with the Department of Production and Management Engineering, Democritus University of Thrace, 12 Vas. Sophias, GR-67132, Xanthi, Greece lbampis@pme.duth.gr, aamanat@ee.duth.gr, agaster@pme.duth.gr

able to treat each camera measurement as an aggregation of Structure-Aware Viewpoint-Invariant High-Order Visual-Words (SVHVs) rather than individual Visual-Words (VWs). By SVHV we denote a bunch of VWs that produce similar displacement (optical flow) vectors, in terms of magnitude and orientation, when observed by two consecutively acquired images, and hence they are typically originated from the same surface in a particular depth. The nature of our approach incorporates the environment's structure into the description while still preserving the essential rotation and scale invariance properties. Thus, the computationally expensive geometrical verification steps are avoided, allowing for a real-time loop closure detection system to be developed even for the case of a low-power mobile device. An operational example of the proposed SVHV-based technique is illustrated in Fig. 1.

The rest of this paper is organized with the following structure: Section II discusses the recent literature on loop closure detection. In Section III, the proposed methodology for describing and matching the input images is analytically explained. The implementation details for achieving a real-time mobile application are outlined in section IV. The proposed system's performance, in terms of accuracy and execution time, is presented in Section V, while subsequently, comparative results against other state-of-the-art approaches are shown. Finally, Section VI provides the author's conclusions and future work regarding further extensions and applications of the proposed technique.

II. Related Work

One of the first techniques that introduced the BoVW model into the image recognition problem was described by Sivic and Zisserman [13]. According to them, a visual vocabulary of SIFT-derived words was created using k-means clustering. Then, each image was converted into a description vector/histogram containing each term's frequency of occurrence, through the "Term Frequency – Inverse Document Frequency" (TF–IDF) weighting scheme. Consequently, similar frames were identified by using a cosine distance metric. In a later work, Nister and Stewenius [14] proposed an alternative storage representation of the visual vocabulary based on a tree structure offering a more computationally efficient feature-to-visual-word conversion.

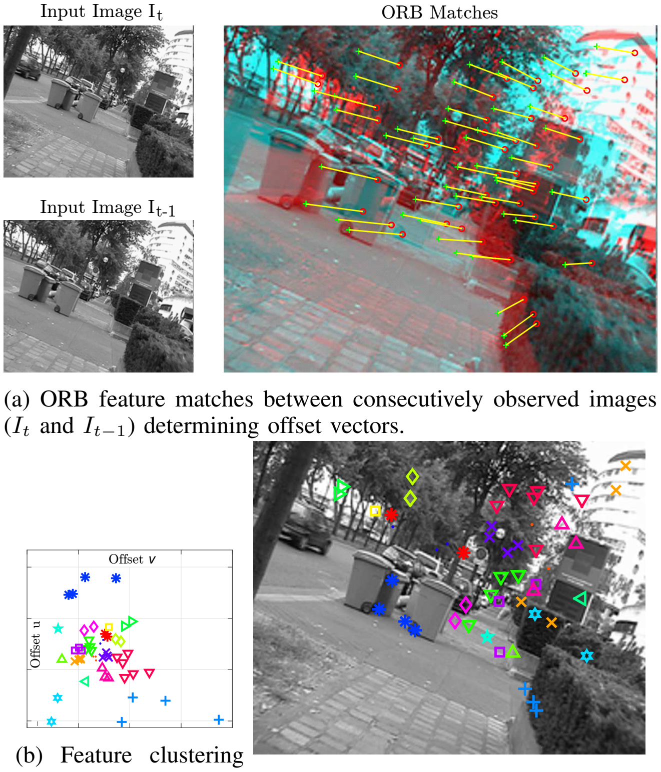
In order to take advantage of the overall operational environment of a loop closure detection system, the FAB-MAP algorithm [15] and its sparsely approximated extension FAB-MAP 2.0 [16] were based on a Chow Liu tree that captured the dependencies between multiple VWs' appearances. Although both techniques stimulated interest for a plethora of later methodologies, it has been reported [17] that their achieved performance may be reduced in cases of trajectories with repetitive visual patterns since no geometrical information was retained between the feature points. In another representative work, Angeli et al. [18] encoded the image description with two distinct visual vocabularies (one based on SIFT descriptors [19] and one on color histograms), while the loop closure detection performance was enhanced by taking into account the matching probability of the previously obtained frames in a Bayesian filtering scheme.

With the aim to provide efficient calculations, more recent techniques have deviated from the above probabilistic loop closure detection frameworks. More precisely, the image matching information and the information derived from the acquired map are exploited by two discrete and subsequent steps. Gálvez-López and Tardós in [10] proposed a loop closure detection system based on the binary description of BRIEF [20] features. To enhance image matches that persist over time, pairs of images were considered as loop closing camera measurements only if they were supported by groups of temporally-consistent and highly-similar frames. In a later work, Mur-Artal and Tardós [21] added further rotation and scale invariance to the above algorithm by using the description of ORB [22] features in a real-time key-frame SLAM system. Aiming to cope with the absence of the local features' spatial arrangement, both of the aforementioned techniques applied a geometrical verification test, based on the evaluation of a valid camera transformation, as a post-processing step that burdened the computational frequency in the false positive cases. On the contrary, our method incorporates a quantitative interpretation of the aforementioned geometrical test into the VWs matching procedure, allowing for the calculation of camera transformations only for the purposes of SLAM in true loop closing cases.

Recently, a wide variety of techniques has been reported in the related literature that aims to address the task of visual place recognition under extreme lighting and/or environmental differences (different periods during the day/year). Many of those methods deviate from the BoVW model due to the local features' inability to be detected and matched under such different conditions [23]. Instead, methods like the ones presented by Milford and Wyeth [24] or Arroyo et al. [25] concluded into a global image description, sacrificing some of the viewpoint invariance, to gain robustness over the potential appearance changes. Another family of techniques that offer robustness over the environmental changes is the one based on the classification power of Convolution Neural Networks (CNNs) [26], [27], [28], [29]. Although CNN-based techniques are considered as state-of-the-art in retrieval and place recognition tasks, they are still disconnected from the overall SLAM and loop closure detection problems. As pointed out by Fei et al. [30] and Sizikova et al. [31], the CNNs' compositionality property, the lack of topological information at the higher networks levels, as well as their reliance over viewpoint-independent surface appearances make them suboptimal for loop closure detection approaches. To cope with the environmental changes' effect, the proposed BoVW-based method can be efficiently combined with an illumination invariant image representation technique, e.g. [32], [33], though such an application is beyond the scope of this paper and thus it is not further discussed.

III. Proposed Method

In this section our loop closure detection approach is described. The proposed system can be divided into off-line



(a) ORB feature matches between consecutively observed images (I_t and I_{t-1}) determining offset vectors.

(b) Feature clustering in the offset space.

(c) Produced SVHV groups.

Fig. 2: The proposed procedure of clustering VWs with common offset vectors and producing SVHV groups. Different clusters are notated with different coloring and markers.

procedures (required for training) and on-line ones operating while the robot's trajectory escalates. Our first step is to train a visual vocabulary in order to quantize the descriptor space and reduce the matching procedure's computational complexity. During the on-line algorithm execution, we take advantage of the sequential acquisition of the input image stream and we calculate the spatial displacement of every detected local feature point. Using an agglomerative clustering technique, the obtained VWs are grouped into clusters based on their displacement domain forming the SVHV groups. In such way, image matches are detected when their corresponding VWs form a sufficient number of co-occurring SVHVs. Finally, a temporal consistency filter kernel, the coefficients of which were trained off-line, is applied to the similarity metrics of neighboring camera measurements, advancing the scores of images that persist over time.

A. Off-line Visual Vocabulary Training

Aiming at calculation efficiency, a tree structured visual vocabulary is adopted based on the scale and rotation invariance of ORB features. To that end, a sample of training descriptors from the Bovisa 2008-09-01 [9] dataset was created and used as input to a k-medians hierarchical clustering based on Hamming distance and k-means++ [34] seeding. Thus, the descriptor space was quantized into a total of $W = 10^6$ VWs (w_i, $i \in [1, W]$) with a vocabulary tree of $L = 6$ levels and $B = 10$ branches per level. In order to provide further robustness into the description, we adopt the TF–IDF model and additionally retain the values N_i^D and N^D corresponding to the number of the i-th VW occurrences and the total number of words occurrences in the whole training dataset, respectively. Thus, an IDF weight can be obtained for every w_i using the formula:

$$idf\,(w_i) = \log\left(logN^D / N_i^D\right). \qquad (1)$$

B. Creating SVHVs with Rotation and Scale Invariance

Given a pre-trained visual vocabulary, the first on-line step of the proposed approach refers to the SVHV-based image description. An SVHV of order O refers to an aggregation of O VWs that are typically derived from the same entity in the observed scene. A possible match between SVHVs implies that the scene does not only contain common features but also common objects, since different VWs and different VWs' layouts both produce distinguishable SVHVs. Figure 2 graphically illustrates the overall procedure for producing SVHVs groups.

Probably the most intuitive approach for grouping the local feature points into SVHVs would be to try to identify neighboring VWs from each input frame. Although straightforward, this approach fails to associate the feature points that are originated from a common entity since it only considers a 2D projection (image plane) of the actual 3D environment. A simple example can be considered where two objects are captured as close to each other, thus producing neighboring feature points, but are actually located in different depths. Another common technique for identifying high-order features is based on calculating the spatial displacement between the query (I_q) and the database (I_d) images' feature points [35], [36], [37], [38]. According to that, the local features detected on I_q and each I_d are converted into their BoVW representations and matched with each other creating point-to-point associations. By quantizing the degrees of freedom of a particular linear image transformation, the produced point-to-point displacements are grouped based on the quantization bin they belong to. Those groups formulate the high-order features for each image pair, though the clustering results can significantly vary according to the observations' viewpoint changes. This is owed to the fact that a scene's projection on two different camera planes can only be precisely described by a perspective transformation of known camera's ego-motion. As an example, observing the same scene, of varying depth, by different camera-angles may result into feature points assigned to various quantization bins of the linear transformation, even though they were originated from the same object.

In contrast with the aforementioned technique, that firstly matches the feature points between query and database images and then creates the groups of VWs, here we take advantage of the sequential frame acquisition in a loop closure detection system and we follow an inverted approach. Our goal is to produce a structure-aware and viewpoint-invariant description by creating the required SVHVs through the means of optical flow. More particularly, we assume that between consecutive frames any rotation effect can be considered negligible, while the only aspect affecting the feature points' displacement is the structure of the observed world.

At a time instance t, the most prominent ORB features are extracted from the current image I_t, converted into VWs and matched with the ones of the previously grabbed I_{t-1}, as shown in Fig. 2a. In order to limit the local feature point detection in the center of each image, typically corresponding to small and noisy parallax information, we divide each I_t into blocks and restrict the detector to preserve a uniform feature point distribution [4]. Then, the spatial offset between each pair of matched features is calculated forming a set of displacement vectors $<u, v>$, with u and v referring to the displacement over the vertical and horizontal image axis, respectively. Then, the I_t SVHV groups are formulated based on a statistical clustering technique to provide further tolerance over possible feature layouts. Toward this end, the dynamical nature of the agglomerative hierarchical clustering "Weighted Pair Group Method with Arithmetic mean" (WPGMA) [39] is exploited. The WPGMA algorithm continuously clusters pairs of data that present the smallest distance $d_{min} = (d_a + d_b)/2$ (with d_a and d_b being the distance of cluster a and b from a previous hierarchical level, respectively) until the point where one cluster is left, forming a tree structure. At each tree node (clustering level) n, we calculate an inconsistency coefficient:

$$c_n = \frac{d_n - m_n}{\sigma_n}, \tag{2}$$

where m_n and σ_n denote the mean and standard deviation, respectively, between distances d_n, d_{n1} and d_{n2}, with $n1$ and $n2$ referring to the children-nodes of clustering level n. Thus, the $n1$ and $n2$ nodes of each parent-node with $c_n > th_c$ are considered to contain two discrete clusters of VWs each corresponding to a group of SVHVs, with th_c being the inconsistency threshold value. The interpretation behind the above inconsistency check is to separate the clusters whose parent-node deviation is greater than th_c times the branch's standard deviation. The calculated spatial offset can not be further used as a descriptive information for each cluster between different traversals of the same area, since it is subject to the camera's ego-motion. On the contrary, it is only computed to separate the VWs of a single image into bunches, based on the scene's structure. Figures 2b and 2c depict the results of a representative offset space clustering together with the corresponding SVHV groups.

Finally, in order to efficiently match the SVHVs in the following steps, an alternative description representation is adopted. For each SVHV group in image I_t, we retain the multiset of its VW-members' indexes as $\mathbb{G}_k = \{w_{i1},\ w_{i2},\ w_{i3},\ ...\}$. A list of the occurring $\mathbb{G}_k$ multisets is then assigned to image I_t with the form of $L_t = <\mathbb{G}_{k1},\ \mathbb{G}_{k2},\ \mathbb{G}_{k3},\ ... >$.

C. Computing the Similarity

As mentioned before, a typical approach for matching two images I_1 and I_2, under the TF–IDF model, is to produce their corresponding description histograms ($V(I_1)$ and $V(I_2)$) and then measure their similarity. The ranking technique utilized in this paper is based on the *cosine similarity*, which is defined as:

$$C_{sim}(I_1,\ I_2) = \frac{V(I_1) \cdot V(I_2)}{\|V(I_1)\| \|V(I_2)\|}. \tag{3}$$

The *cosine similarity* produces the same ranking results with the Euclidean distance when applied to L2-normalized description histograms.

The clustering output of the previously described procedure can significantly vary according to the detected features' number, layout, as well as their ability to be matched between consecutive frames. Thus, matching between SVHVs is not straightforward. To that end, we make use of the already formulated clusters from the database images and reverse-engineer the VW groups at query time. More specifically, for each newly observed query image I_q, we check against every database element I_d, $d \in [0, t - t_w]$[1] for possible loop closing pairs. The procedure for calculating the C_{sim} using the SVHV-based description can be achieved by the following steps:

1) For each VW w_x in I_q and $\mathbb{G}_y$ in L_d, we find the co-existing SVHVs by computing the multiset:

$$\mathbb{Q}_y = \bigcup_{w_x \in \{I_q | c_x < th_c\}} (w_x \cap \mathbb{G}_y). \tag{4}$$

In order to compute the above union an iterative procedure is followed, at each step of which an inconsistency coefficient c_x is computed between the displacement vectors of the current union's VW-members, as measured by image I_q. Any w_x producing $c_x \geq th_c$ is omitted from the multiset as inconsistent with the rest of the group. The resulting $\mathbb{Q}_y$ multiset determines a co-occurring SVHV group.

2) The TF–IDF term for each one of the SVHVs is calculated through [35]:

$$C_y = \binom{Q_y - 1}{O - 1} \sum_{w_i \in \mathbb{Q}_y} idf(w_i), \tag{5}$$

with Q_y being the cardinality of multiset $\mathbb{Q}_y$. Multisets with $Q_y < O$ are going to offer zero contribution to the overall similarity (through the above binomial coefficient term) due to insufficient number of consistent VW matches. Note that the TF[2] normalization term is omitted from Eq. 5 since its effect will be incorporated to our final step.

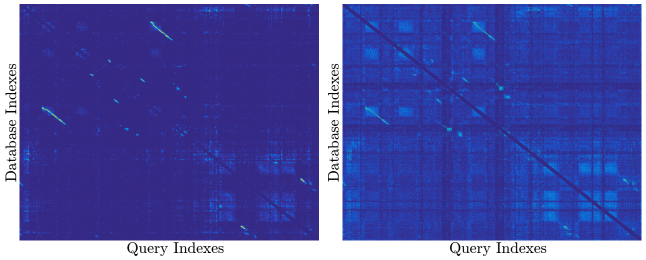
3) A similarity score is produced between I_q and I_d using:

$$C_{scr}(I_q,\ I_d) = \sum_{Q_y > o} C_y. \tag{6}$$

4) Finally, the *cosine similarity* $C_{sim}(I_q,\ I_d)$ is obtained by normalizing the above score with

[1] t_w is a time window preventing image matches that cannot be characterized as loop closures since they were obtained immediately before I_q.

[2] The TF term of a VW w_i corresponds to the number of this word's occurrences normalized by the total number of detected VWs.



(a) SVHV-derived matrix M. (b) VW-derived matrix M.

Fig. 3: Qualitative difference between SVHV-based and VW-based image matching.

its L2-norm $\|V(I_q)\|\|V(I_d)\|$. Note that since $\|V(I)\| = \sqrt{V(I) \cdot V(I)}$, the L2-norm of any image I can be efficiently computed by accumulating the IDF terms of every self-occurring SVHV.

Although the above computations may seem computationally exhaustive, one needs to realize that the majority of $\mathbb{Q}_y$ will result into empty sets. Therefore, they can be omitted from the calculations since they offer zero contribution to the final *cosine similarity*. Note that in Section IV further implementation-related improvements are also discussed.

Through the means of the above procedure a similarity matrix M (with $M(q,d) = C_{sim}(I_q,\ I_d)$) can be incrementally formulated while the trajectory escalates. As it can be seen in Fig. 3a, the structure of a fully constructed SVHV-based similarity matrix is more robust than a simple VW-based one (Fig. 3b), due to the additional information of the overall scene's structure.

D. Temporal Consistency Filter

Even though our approach reduces the possible false positive loop closing matches that differ in terms of geometry, it can also result into false negative cases. As in any BoVW-based approach, there is always the possibility of two images, actually corresponding to the same scene, to produce different VWs due to dynamic changes (e.g. moving objects), aliasing effects, noise, etc. Thus, scores between temporal consistent sets of images are combined, advancing the cases of similarity incoherences. Considering an instance of: a highly similar pair I_{q-1}–I_{d-1}, a pair I_q–I_d of small similarity and a highly similar I_{q+1}–I_{d+1}, our objective is to characterize the I_q–I_d pair as a loop closuring one, by considering that the set is temporally consistent. Instead of simply accumulating the similarities between close-in-time matches [10], a single convolutional filtering kernel K of size h is applied over the entries of similarity matrix M:

$$K = \begin{bmatrix} \kappa_{1,1} & \kappa_{1,2} & \cdots & \kappa_{1,h} \\ \kappa_{2,1} & \kappa_{2,2} & & \\ \vdots & & \ddots & \\ \kappa_{h,1} & & & \kappa_{h,h} \end{bmatrix}. \tag{7}$$

In our previous work [40], a similar filtering technique was applied in order to advance sequence-based similarity scores. In this case though, the consistency filtering refers to single instances, thus a bigger kernel size is preferable.

Avoiding to manually selecting the $\kappa_{i,j}$ values, a training scheme based on cost function minimization is formulated. Eventually, we aim to identify the values of K for which the entries of matrix $\widehat{M} = M * K$ (with $*$ denoting the convolutional operation) would separate the loop-closing from the non-loop-closing pairs with the most effective way, when thresholded by a value κ_0. With the above formulation in mind, it is easy to describe our filtering approach as a linear logistic regression classifier, with a hypothesis vector of $[\kappa_{i,j}, -\kappa_0]$, operating on the similarity entries of matrix M. Through the means of a generic set of training images that contains a sufficient number of loop closure events, together with their corresponding ground truth, the coefficients of this linear classifier can be evaluated using gradient descent. The selection of logistic regression is justified due to its high tolerance when provided with unbalanced training samples. This effect can only be accounted when the training and testing data contain approximately the same amount of loop-closing and non-loop-closing events [41], [42], as to be considered during its learning phase (Section V-A.2). Finally, in order to identify the kernel size h, a series of cross-validation tests are carried out, comparing the achieved performance between different sized convolutional filters.

During the on-line algorithm's execution, kernel K is applied to the similarity sub-matrices m –centered around the corresponding (q,d) entry of M– and produce a filtered similarity metric $\widehat{M}(q,d)$. Subsequently, a loop closure event is identified in the cases where $\widehat{M}(q,d) > \kappa_0$. Here, we need to highlight that the overall $\widehat{M}$ matrix is not required to be fully formulated nor of known size since the operations are performed specifically over a small window of h recently acquired input frames.

IV. Mobile Device Implementation

Aiming to provide a fully integrated system, a C++ based version of the proposed algorithm was developed capable of running in real-time on a mobile device. To achieve this, a series of computational improvements was performed referring to both methodical and parallelization techniques.

With the view to avoid the computationally expensive nearest-neighbor descriptors' search between consecutively acquired images the direct indexing approach [10] was adopted. According to that, feature matches are achieved by associating points with common parent nodes of the vocabulary tree. Since this procedure may result in some false-positive matches, we additionally assigned the subsequent hierarchical clustering to reject feature pairs presenting inconsistent displacement with the majority of clusters [4]. In addition, an efficient voting scheme to calculate the *cosine similarities* was also incorporated through the means of inverse indexing [35]. This way, the C_y terms are only computed between images that contain common VWs, further reducing the filter's K convolutional operations.

Finally, taking advantage of the ARM-NEON co-processor, embedded into the majority of modern mobile

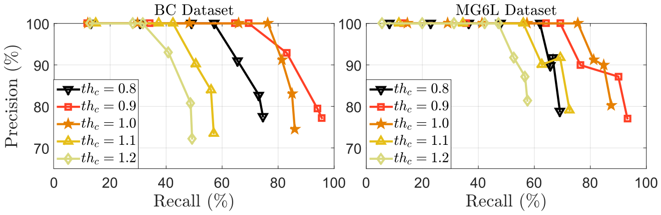


Fig. 4: Precision-Recall curves measuring the effect of th_c threshold on the *training* datasets.

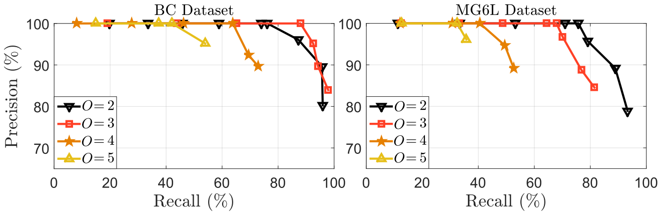


Fig. 5: Precision-Recall curves measuring the effect of SVHVs' order O on the *training* datasets.

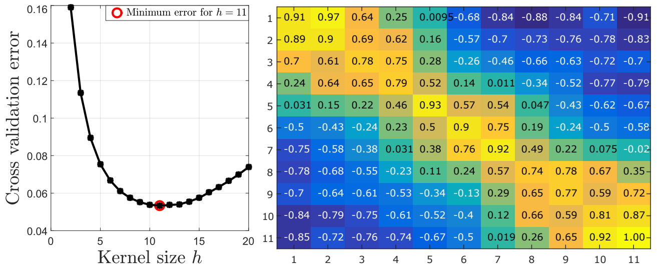
devices, we increased the computational frequency of many procedures through parallelization. More specifically, we implemented SIMD-based (Single Instruction on Multiple Data) versions of the ORB feature detector and descriptor, as well as the Hamming distance required for traversing the vocabulary tree.

V. Experimental Results

In this section an evaluation of the proposed SVHV-based loop closure detection approach is presented and its achieved performance is compared against other state-of-the-art techniques. Throughout our experiments, we made use of the Precision-Recall curves as a means of measuring the recognition accuracy of each implementation. As described, our algorithm is specifically designed for addressing the loop closure detection task under a freely moving hand-held mobile device. Thus, it can be applied to challenging operational environments, in terms of viewpoint changes, and of confined traversed distance due to memory limitations. A variaty of publicly available datasets with the above characteristics was chosen, namely Bicocca 2009-02-25b [9] (BC), Malaga 2009 Parking 6L [43] (MG6L), New College [44] (NC), City Centre [15] (CC), Lip6 Indoor [18] (L6I) and Lip6 Outdoor [18] (L6O). The datasets were distinguished into two sets, i.e. *training* and *testing*. The *training* set contains the first two datasets and it was used for learning the parameters introduced by our method, while the *testing* one contains the rest and it is considered to contain evaluation cases for measuring the method's overall performance.

A. System Evaluation

1) SVHV Formulation: We start by evaluating the effect of th_c threshold to the overall system's performance. To that end, a series of experiments was conducted using our *training* datasets. In general, higher values for this threshold would result in clusters with bigger VW groups and consequently SVHVs of higher order, while lower values would produce the opposite effect. We selected a series of th_c values and produced the corresponding Precision-Recall curves by eliminating kernel's K effect and varying threshold k_0. For this experiment the most general, while still meaningful, case of $O = 2$ was selected since the exact effect of SVHVs' order will be further evaluated. The most representative curves are presented in Fig. 4. As it can be seen, the value of $th_c = 1.0$ performs better for each one of the tested cases and thus, adopted by the algorithm. The examples presented in Fig. 1 and Fig. 2c were obtained by using this threshold. At a first glance, the dominance of $th_c = 1.0$ over every *testing* dataset may seem unintuitive. Though, one needs to consider that key role here plays the production of a sufficient number of clusters and not their exact semantic consistency. In other words, possible groups of VWs that do not correspond to the exact same object are not particularly harmful to the system as long as enough SVHVs are created.



	1	2	3	4	5	6	7	8	9	10	11
1	0.91	0.97	0.64	0.25	0.0095	-0.68	-0.84	-0.88	-0.84	-0.71	-0.91
2	0.89	0.9	0.69	0.62	0.16	-0.57	-0.7	-0.73	-0.76	-0.78	-0.83
3	0.7	0.61	0.78	0.75	0.28	-0.26	-0.46	-0.66	-0.63	-0.72	-0.7
4	0.24	0.64	0.65	0.79	0.52	0.14	0.011	-0.34	-0.52	-0.77	-0.79
5	0.031	0.15	0.22	0.46	0.93	0.57	0.54	0.047	-0.43	-0.62	-0.67
6	-0.5	-0.43	-0.24	0.23	0.5	0.9	0.75	0.19	-0.24	-0.5	-0.58
7	-0.75	-0.58	-0.38	0.031	0.38	0.76	0.92	0.49	0.22	0.075	-0.02
8	-0.78	-0.68	-0.55	-0.23	0.11	0.24	0.57	0.74	0.78	0.67	0.35
9	-0.7	-0.64	-0.61	-0.53	-0.34	-0.13	0.29	0.65	0.77	0.59	0.72
10	-0.84	-0.79	-0.75	-0.61	-0.52	-0.4	0.12	0.66	0.59	0.81	0.87
11	-0.85	-0.72	-0.76	-0.74	-0.67	-0.5	0.019	0.26	0.65	0.92	1.00

(a) Cross validation error between different sized K kernels.

(b) Filter kernel K of size $h = 11$ corresponding to the lowest cross validation error.

Fig. 6: Temporal consistency filter training results.

Next, we assess the effect of SVHVs' order to the system's performance. Once more, several O values were selected and evaluated through the means of Precision-Recall curves. Kernel K was omitted from this experiment as well, while the inconsistency threshold was fixed to $th_c = 1.0$. By varying the value of κ_0, we obtained the results presented in Fig. 5 for each one of our *training* datasets. As expected, higher order O values resulted into more rigorous detections, restricting every SVHV match to be supported by many displacement-consistent VW-members. Due to the fact that BC dataset contains many highly similar but different locations[3], it can be seen that the best performing O value is greater than the case of MG6L. For the rest of this paper, the value of $O = 2$ is adopted since it ensures a 100% Precision accuracy for every tested case.

2) Temporal Consistency Filter: As mentioned, the final step of the proposed approach refers to the application of a consistency filtering over the obtained similarity metrics. To learn the values of kernel K, as the hypothesis vector of a logistic regression classifier, we made use all datasets in the

[3] As shown in Fig. 1, BC dataset refers to an indoors environment (library) with many repeatable visual patterns (bookshelves, studying rooms, etc).

training set. Note that the corresponding loop closure ground truth, required by the learning procedure, was provided by the authors of [10]. Firstly, the similarity matrices M were formulated and concatenated creating a generic sample. We further distinguished this sample into two subsets, namely training and cross-validation, containing 70% and 30% of the data, respectively. The first one was used for estimating the hypothesis vector for each evaluated kernel size h, while the second for measuring their cross validation error. We tested each h value into the interval $[2, 20]$ and show the evaluation results in Fig. 6a. As it can be seen, a kernel size of $h = 11$ corresponds to the lowest classification error and thus selected for our final parameterization. The corresponding kernel K is shown in Fig. 6b. As a final note, the used *training* set was considered to contain a representative ratio between loop-closing and non-loop-closing events. Though, considering a specified operational environment, the estimated hypothesis vector can be accordingly adjusted to fit each particular sample distribution, as described in [41].

3) Overall Performance: We evaluate our system's overall performance both in terms of effectiveness and computational time. Using the parameterization described above, we obtained the Precision-Recall curves presented in Fig. 7 by varying the value of k_0 for each *testing* dataset. Here, it is crucial to establish that within the scope of a loop closure detection system, a false-positive match can lead to catastrophic failure of the SLAM estimation and thus the Precision accuracy must always be retained at 100%, even with the cost of losing some of the Recall rates. Nevertheless, the employed classification technique does not share the same principle and estimates the value of k_0 that produces the smallest overall error between the two classes. For this reason, the value of $k_0 = 1.25$ is more suitable for a loop closure detection system since it offers a 100% Precision accuracy while still retaining the highest possible Recall rates. The results regarding L6I and L6O datasets are particularly descriptive since they both contain loop closure events that occur under a camera's $45°$ rotation in the roll axis. Thus, each one of these datasets was further distinguished into two subsets, one containing the rotated and one the non-rotated loops, so as to assess the algorithm's robustness over possible viewpoint changes.

As a final test, we assessed our algorithm's operational frequency on the Google's Project Tango developing tablet [45]. Using the longest used datasets, i.e. $15K$ frames from the NC, the algorithm was able to process each input frame in $48.7ms$ on average, offering a real-time loop closure detection system (in term of processing the input faster or in equal time with the operational frequency of a key-frame SLAM algorithm). Note that the minimum execution time was measured at $30.4ms$ and the maximum at $57.1ms$.

B. Comparative Results

The achieved Recall performance (for 100% Precision) of our method is compared against other well-established BoVW-based loop closure detection techniques. We restrict the comparisons between approaches that offer real-time performances for a key-frame SLAM system ($\sim$ 100-200ms per frame [4], [5], [6]). Table I summarizes the results for each used dataset, obtained straightforwardly from the respective papers. As it can be observed, our technique presents higher Recall rates for the most of the evaluated cases, leading to a more holistic solution. This is owed to the fact that other BoVW-based methods reject the geometry information from the description and only assess the number of commonly observed local features from a given scene.

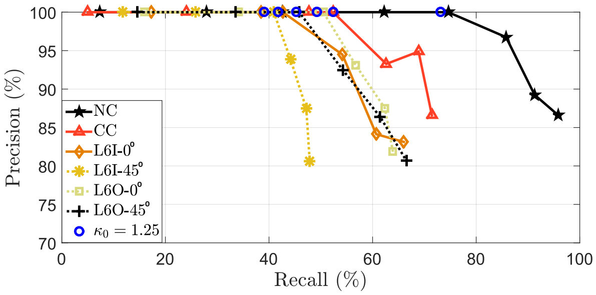


Fig. 7: Precision-Recall curves measuring the algorithm's performance over every *testing* dataset.

TABLE I: Comparative results showing the achieved Recall rates (%) for 100% Precision accuracy.

	BC	MG6L	NC	CC	L6I	L6O
Gálvez-López [10]	81.20	74.75	55.92	31.61	N/A	N/A
Mur-Artal [21]	76.60	**83.94**	70.29	43.03	N/A	N/A
FAB-MAP 2.0 [16]	N/A	68.52	N/A	38.77	N/A	N/A
Angeli [18]	N/A	N/A	N/A	N/A	36.86	23.59
Proposed	**86.37**	80.97	**74.60**	**52.36**	**42.32**	**49.55**

VI. Conclusions and Future Work

In this paper we presented a high-order VW based technique for identifying loop closure events on a freely moving hand-held mobile device. The proposed method clusters VWs producing similar optical flow measurements, as observed by two consecutive viewing-points, offering rotation and scale invariance to the description. Aiming to further improve the detection performance, a temporal consistency filter was applied incrementally over the similarity scores, the coefficients of which were learned on an off-line training step. Although wrapped into a complete pipeline, the main contribution of the paper in hand, i.e. the SVHV-based matching, can be efficiently adapted by the majority of BoVW-based techniques and incorporate the environment's structure into the description.

The authors' plans for future work include the extension of the proposed SVHV-based description into an image sequence architecture as the one proposed in [40]. Additionally, further research can be made in order to incorporate the presented algorithm into a full SLAM system and take advantage of the formulated SVHV groups for the visual odometry estimation.

ACKNOWLEDGMENT

The loop closure ground truth data regarding the BC, MG6L, NC and CC datasets were provided by the authors of [10]. The ORB ARM-NEON implementation was funded by Google's Project Tango and developed at MARS Lab UMN.

REFERENCES

[1] J. Folkesson and H. Christensen, "Graphical SLAM – A self-correcting map," in *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 1, 2004, pp. 383–390.

[2] S. Thrun and M. Montemerlo, "The graph SLAM algorithm with applications to large-scale mapping of urban structures," *Int. J. Robotics Research*, vol. 25, no. 5-6, pp. 403–429, 2006.

[3] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based SLAM," *Intelligent Transp. Syst. Magazine*, vol. 2, no. 4, pp. 31–43, 2010.

[4] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Trans. Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[5] H. Strasdat, J. Montiel, and A. J. Davison, "Scale Drift-Aware Large Scale Monocular SLAM," in *Proc. Robotics: Science and Syst.*, vol. 2, no. 3, 2010, p. 5.

[6] C. Mei, G. Sibley, M. Cummins, P. M. Newman, and I. D. Reid, "A Constant-Time Efficient Stereo SLAM System," in *Proc. British Mach. Vision Conf.*, 2009, pp. 1–11.

[7] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—a modern synthesis," in *Proc. Int. Workshop Vision Algorithms*, 1999, pp. 298–372.

[8] 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 Syst.*, vol. 57, no. 12, pp. 1188–1197, 2009.

[9] RAWSEEDS. (2007-2009) Robotics Advancement through Webpublishing of Sensorial and Elaborated Extensive Data Sets (Project FP6-IST-045144). [Online]. Available: http://www.rawseeds.org/rs/datasets

[10] D. Gálvez-López and J. D. Tardós, "Bags of binary words for fast place recognition in image sequences," *IEEE Trans. Robotics*, vol. 28, no. 5, pp. 1188–1197, 2012.

[11] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, 2007, pp. 1–8.

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

[13] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. IEEE Int. Conf. Comput. Vision*, 2003, pp. 1470–1477.

[14] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, vol. 2, 2006, pp. 2161–2168.

[15] M. Cummins and P. Newman, "FAB-MAP: Probabilistic localization and mapping in the space of appearance," *Int. J. Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008.

[16] ——, "Appearance-only SLAM at large scale with FAB-MAP 2.0," *Int. J. Robotics Research*, vol. 30, no. 9, pp. 1100–1123, 2011.

[17] P. Piniés, L. M. Paz, D. Gálvez-López, and J. D. Tardós, "CI-Graph simultaneous localization and mapping for three-dimensional reconstruction of large and complex environments using a multicamera system," *J. Field Robotics*, vol. 27, no. 5, pp. 561–586, 2010.

[18] A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer, "Fast and incremental method for loop-closure detection using bags of visual words," *IEEE Trans. Robotics*, vol. 24, no. 5, pp. 1027–1037, 2008.

[19] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[20] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary Robust Independent Elementary Features," in *Proc. European Conf. Comput. Vision*, 2010, pp. 778–792.

[21] R. Mur-Artal and J. D. Tardós, "Fast relocalisation and loop closing in keyframe-based SLAM," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2014, pp. 846–853.

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

[23] C. Valgren and A. J. Lilienthal, "SIFT, SURF & seasons: Appearance-based long-term localization in outdoor environments," *Robotics and Autonomous Syst.*, vol. 58, no. 2, pp. 149–156, 2010.

[24] M. J. Milford and G. F. Wyeth, "SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2012, pp. 1643–1649.

[25] R. Arroyo, P. F. Alcantarilla, L. M. Bergasa, and E. Romera, "Towards Life-Long Visual Localization using an Efficient Matching of Binary Sequences from Images," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2015, pp. 6328–6335.

[26] N. Sünderhauf, S. Shirazi, F. Dayoub, B. Upcroft, and M. Milford, "On the performance of ConvNet features for place recognition," in *Proc. IEEE Int. Conf. Intelligent Robots and Syst.*, 2015, pp. 4297–4304.

[27] N. Sünderhauf, S. Shirazi, A. Jacobson, F. Dayoub, E. Pepperell, B. Upcroft, and M. Milford, "Place recognition with ConvNet landmarks: Viewpoint-robust, condition-robust, training-free," *Proceedings of Robotics: Science and Syst.*, 2015.

[28] R. Arroyo, P. F. Alcantarilla, L. M. Bergasa, and E. Romera, "Fusion and binarization of CNN features for robust topological localization across seasons," in *Proc. IEEE Int. Conf. Intelligent Robots and Syst.*, 2016, pp. 4656–4663.

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

[30] X. Fei, K. Tsotsos, and S. Soatto, "A Simple Hierarchical Pooling Data Structure for Loop Closure," in *Proc. European Conf. Comput. Vision*, 2016, pp. 321–337.

[31] E. Sizikova, V. K. Singh, B. Georgescu, M. Halber, K. Ma, and T. Chen, "Enhancing place recognition using joint intensity-depth analysis and synthetic data," in *Proc. European Conf. Comput. Vision Workshop*, 2016, pp. 901–908.

[32] M. Shakeri and H. Zhang, "Illumination invariant representation of natural images for visual place recognition," in *Proc. IEEE Int. Conf. Intelligent Robots and Syst.*, 2016, pp. 466–472.

[33] W. Maddern, A. Stewart, C. McManus, B. Upcroft, W. Churchill, and P. Newman, "Illumination invariant imaging: Applications in robust vision-based localisation, mapping and classification for autonomous vehicles," in *Proc. IEEE Int. Conf. Robotics and Automation, Visual Place Recognition Changing Environments Workshop*, vol. 2, 2014, p. 3.

[34] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proc. ACM-SIAM symposium on Discrete algorithms*, 2007, pp. 1027–1035.

[35] Y. Zhang, Z. Jia, and T. Chen, "Image retrieval with geometry-preserving visual phrases," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, 2011, pp. 809–816.

[36] Y. Zhang and T. Chen, "Efficient kernels for identifying unbounded-order spatial features," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, 2009, pp. 1762–1769.

[37] ——, "Weakly supervised object recognition and localization with invariant high order features." in *Proc. British Mach. Vision Conf.*, 2010, pp. 1–11.

[38] D. Sezganov and M. Porat, "Improving large-scale image retrieval using geometric weighting," in *Proc. Int. Conf. on Machine Learning and Comput. Science*, 2012, pp. 162–166.

[39] R. R. Sokal, "A statistical method for evaluating systematic relationships," *Univ Kans Sci Bull*, vol. 38, pp. 1409–1438, 1958.

[40] L. Bampis, A. Amanatiadis, and A. Gasteratos, "Encoding the description of image sequences: A two-layered pipeline for loop closure detection," in *Proc. IEEE Int. Conf. Intelligent Robots and Syst.*, 2016, pp. 4530–4536.

[41] G. King and L. Zeng, "Logistic regression in rare events data," *Political Analysis*, pp. 137–163, 2001.

[42] S. F. Crone and S. Finlay, "Instance sampling in credit scoring: An empirical study of sample size and balancing," *Int. J. Forecasting*, vol. 28, no. 1, pp. 224–238, 2012.

[43] J.-L. Blanco, F.-A. Moreno, and J. Gonzalez, "A collection of outdoor robotic datasets with centimeter-accuracy ground truth," *Autonomous Robots*, vol. 27, no. 4, pp. 327–351, 2009.

[44] M. Smith, I. Baldwin, W. Churchill, R. Paul, and P. Newman, "The new college vision and laser data set," *Int. J. Robotics Research*, vol. 28, no. 5, pp. 595–599, 2009.

[45] Google Project Tango. [Online]. Available: https://developers.google.com/tango/hardware/tablet