Encoding the Description of Image Sequences: A Two-Layered Pipeline for Loop Closure Detection

Loukas Bampis, Angelos Amanatiadis and Antonios Gasteratos

Abstract— **In this paper we propose a novel technique for detecting loop closures on a trajectory by matching sequences of images instead of single instances. We build upon well established techniques for creating a bag of visual words with a tree structure and we introduce a significant novelty by extending these notions to describe the visual information of entire regions using Visual-Word-Vectors. The fact that the proposed approach does not rely on a single image to recognize a site allows for a more robust place recognition, and consequently loop closure detection, while reduces the computational complexity for long trajectory cases. We present evaluation results for multiple publicly available indoor and outdoor datasets using Precision-Recall curves, which reveal that our method outperforms other state of the art algorithms.**

I. Introduction

The subject of visual Simultaneous Localization and Mapping (SLAM) refers to the task of a robot to localize itself in the world while maintaining a representation of the environment by primarily employing visual sensing systems. Due to its demanding nature and highly received attention, visual SLAM provoked thought for many individual challenges. As part of the graph-based SLAM, the loop closure detection engine is responsible for finding revisited regions of the executed trajectory and creating edge constraints between the present and previously visited pose nodes [1], [2], [3]. Those additional edges can later be used to refine the estimated SLAM output and produce more accurate results [4], [5].

During the last decade, many novel methodologies were presented in the literature aiming to address the loop closure task. These methods can be divided into three main categories: image-to-image, map-to-map and image-to-map matching, with the first one proven to scale better in long trajectories [6]. A well known method adopting a technique based on the first category –also known as appearance-based place recognition– is the FAB-MAP [7], according to which, a Bag of Visual Words (BVW) is created and used to measure the similarity between the acquired images. On the other hand, recently introduced methods have separated the loop closure mechanism from the rest of the SLAM functionality. More specifically, loops are detected by comparing the distances between Visual-Word-Vectors (VWVs), each of which is associated with exactly one image. This approach was initially inspired for solving image retrieval problems [8], yet for the given application time proximity between sequential images is also exploited. Furthermore, maintaining a visual vocabulary with a tree structure (vocabulary tree) [9], [10], [11], is proven to be very efficient both in terms of memory usage and execution time.

Authors are with the Department of Production and Management Engineering, Democritus University of Thrace, 12 Vas. Sophias, GR-671 32, Xanthi, Greece lbampis@pme.duth.gr, aamanat@ee.duth.gr, agaster@pme.duth.gr

The novelty of the presented method (as illustrated by the paper's accompanying video material as well) lies upon the partition of the matching procedure into two stages: initially between long segments of the trajectory and later between single frames, while still retaining a feature-based approach using vocabulary trees. The main advantages the proposed architecture introduces are the following:

- The multitude of loop closure candidates is reduced since a normal sequence consists of multiple images. This characteristic is of significant importance, especially in the case of very long trajectories.
- The ability to reject matches which are different in the general view is provided regardless the fact that some individual frames may be similar.

In this paper, we build upon BVW based techniques and we extend the used VWVs from describing just one image (I-VWV), to an image sequence descriptor (S-VWV). Although, the concept of sequence matching has been already introduced in the literature [12], [13], this is only based on accumulating matching scores between I-VWVs. On the contrary, our method reformulates the VWVs in order to describe image sequences by combining all the visual words found in every image-member into one single vector. In fact, the summation of individual image similarity scores would provide the same results with the proposed method only under the false assumption that the used similarity metrics $f(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$ between two image VWVs ($\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{y}}$) preserve the additive property of linear mapping. Yet, in the real case it may be just an approximation. Following the first level of S-VWV matching, we proceed with individual image-to-image (*i2i*) associations and provide the necessary for SLAM, fundamental matrix. Note that henceforth, the term "*sequences*" will refer to "*sequences of images*" for shortening reasons.

The rest of the paper is organized with the following structure: Section II briefly discusses the related work in the area of appearance-based place recognition. In Section III the proposed algorithm is described in detail, while Section IV provides experimental results on four different datasets (indoor and outdoor) and presents the system's parameter optimization. The computational advantage of sequence-to-sequence (*s2s*) matching is analyzed in Section V and finally, Section VI serves as an epilogue for this paper where our final conclusions are drawn.

II. Related Work

Creating a visual vocabulary to describe and match images is a well established technique and it has been widely used in the past decade to address many place recognition problems [11], [12], [14], [15]. FAB-MAP [7] is a great example that introduces the effectiveness of appearance-based loop detection using visual words. Yet, the extraction of SURF features [16] along with the fact that repetitive patterns may reduce its performance, make it less appealing for long trajectory scenarios [17]. The same issue arises in Schindler's et al. [11] method, where despite the usage of a tree hierarchy for their vocabulary, SIFT feature [18] extraction remains a disadvantage for the execution time.

One of the most representative and acknowledged techniques is described by Gálvez-López and Tardós in [9] with the DBoW2 [19] algorithm. In this work, a vocabulary tree structure of binary words is proposed in order to create I-VWVs and close loops by measuring additive normalized distances between those vectors. Mur-Artal and Tardós used this method in a later work [10] to recognize places, relocalize and detect loops in a real-time implementation of keyframe-based SLAM.

Recently, the notion of matching sequences of images, instead of individual frames, has been reported in the literature. Many techniques adopt a *s2s* matching scheme, especially to recognize places between different lighting conditions (day and night) or year seasons, such as [13], [20], [21], [22]. Despite the variance of approaches, the usage of local feature descriptors is avoided due to their matching incapability under such dramatical environmental and lighting changes. In this paper though, we are interested in maximizing the performance of the loop closure detection task under environment conditions exhibiting insignificant variations. Thus, a feature-based approach is more appropriate since it typically copes better with rotation and scaling changes induced by a freely moving camera. Probably the method that can be characterized as the closest to ours is [12]. This work, although intended for outdoors SLAM applications, pointed out the importance of matching sequences of images. The main difference between [12] and our method is that their version uses cumulative similarity matrices derived from the matching scores between individual images. On the contrary, our method reformulates the approach and utilizes sequence descriptors for the first level of loop closure detection achieving robust results.

III. Proposed Methodology

In this paper, we detect loops using two levels of matching. In the first level, correlations between sequences of images are found, while in the second one their individual frames are matched. To achieve that, we introduce in this paper for the first time a novel pipeline for place recognition that firstly segments the image dataset into groups based on their spatiotemporal proximity and then uses Sequence-Visual-Word-Vectors as a means to find matches between them. Subsequently, it exploits this prior knowledge to find distinct associations of the images themselves.

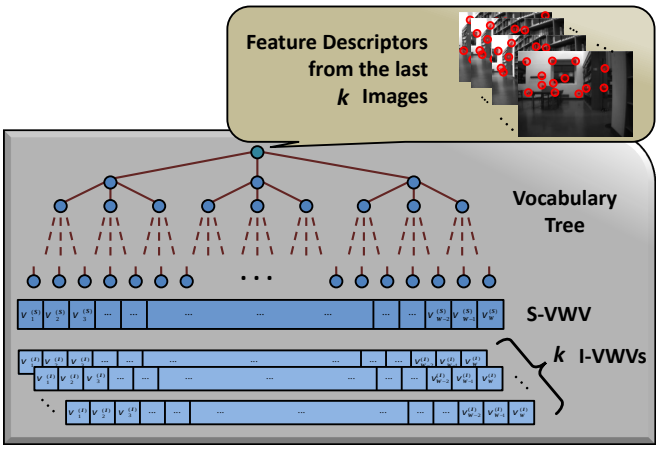


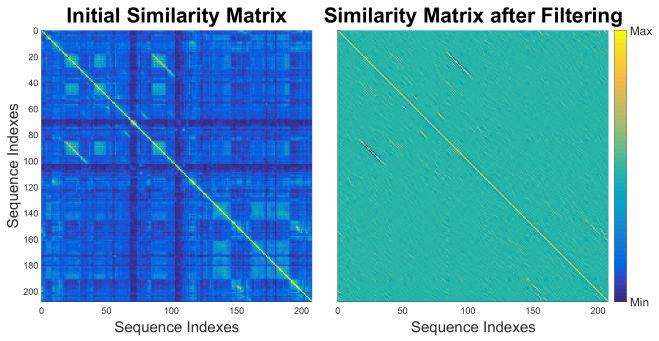
Fig. 1. The main pipeline of the proposed method. Feature descriptors from groups of images traverse the vocabulary tree in order to produce sequence (S-VWVs) and image (I-VWVs) descriptors.

A. Creating Visual-Word-Vectors for Images and Sequences

The first step of the proposed method includes the offline creation of a visual vocabulary in order to quantize the image feature space. A typical approach for creating a BVW with a tree structure [8], [9], [11] was implemented using binary descriptors since they provide low computational complexity. A set of $3M$ BRIEF [23] descriptors were extracted from $10K$ indoor and outdoor images of the Bovisa 2008-09-01 [24] dataset forming a generic training sample D. Using this sample on a k-median hierarchical clustering, with k-mean++ seeding [25] and Hamming distance, we were able to create a tree of $L = 6$ levels and $K = 10$ branches per level with the $W = K^L$ leaf nodes representing the final vocabulary. Two different kinds of multisets need to be introduced here, namely $\mathcal{N}^D$ and $\mathcal{N}_i^D$: $\mathcal{N}^D$ corresponding to the multiset of total visual words found in D and $\mathcal{N}_i^D$ to the multiset of the i-th word occurrences in D.

In order to separate the acquired images into sequences we chose to utilize information provided by the robot's odometry, which is running concurrently with the loop closure detection pipeline, as mentioned before. As the robot moves, a feedback from the odometry thread is provided and the executed trajectory is separated into intervals of μ meters with no overlap between them. Each interval contains a group of acquired image-members and thus a new sequence S is formed. The size of each group, referred to as k, varies and depends on the robot's speed during a sequence and the selected value of μ. In those cases where the robot's odometry is not available, an approximation of the traversed distance can be obtained assuming a relatively steady speed and frame acquisition frequency. Although the first approach is selected for this work, we found that both techniques end up with equivalent sequence sizes and thus performances.

Figure 1 visually depicts the procedure of creating Visual-Word-Vectors for a sequence S and its respective k image-members, $I_m \in S$ $(m \in [1, ..., k])$. We make use of the created vocabulary along with the "term frequency-inverse document frequency" (tf-idf) [14] and formulate two differ-



(a) (b)

Fig. 2. Sequence similarity matrix (a) before and (b) after filtering.

ent kinds of descriptors: the I-VWVs obtained from each individual I_m and the extended version of the proposed sequence description vector, S-VWV, capable of characterizing a physical place as a total. Using the FAST algorithm [26], the most prominent 300 BRIEF feature descriptors are extracted from every image in S. Those descriptors are then quantized into visual words by traversing the created tree and format the following multisets. The first pair of multisets, $\mathcal{N}_i^{(I_m)}$ and $\mathcal{N}_i^{(S)}$, refers to the i-th visual word occurrences in image I_m and sequence S, respectively. The second one consists of $\mathcal{N}^{(I_m)}$ and $\mathcal{N}^{(S)}$ which include the total of visual words found in I_m and S, respectively. The following properties stand for the defined multisets:

$$\mathcal{N}_i^{(S)} = \bigcup_{m=1}^{k} \mathcal{N}_i^{(I_m)} \tag{1}$$

$$\mathcal{N}^{(S)} = \bigcup_{m=1}^{k} \mathcal{N}^{(I_m)} \tag{2}$$

The corresponding k I-VWVs description vectors, analyzed as $\bar{\boldsymbol{v}}^{(I_m)} = (v_1^{(I_m)}, v_2^{(I_m)}, ..., v_i^{(I_m)}, ..., v_W^{(I_m)})$ and a S-VWV one, analyzed as $\bar{\boldsymbol{v}}^{(S)} = (v_1^{(S)}, v_2^{(S)}, ..., v_i^{(S)}, ..., v_W^{(S)})$, are then formulated for every sequence using the tf-idf with:

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

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

In the above equations, $N_i^{(I_m)}$ equals the cardinality of $\mathcal{N}_i^{(I_m)}$ multiset, while $N^{(I_m)}$ the cardinality of $\mathcal{N}^{(I_m)}$. Equivalently, $N_i^{(S)}$ equals the cardinality of $\mathcal{N}_i^{(S)}$ and $N^{(S)}$ the cardinality of $\mathcal{N}^{(S)}$. The terms $N^{(D)}$ and $N_i^{(D)}$ are common for both Eqs. 3 and 4 and correspond to the cardinality of multisets $\mathcal{N}^{(D)}$ and $\mathcal{N}_i^{(D)}$, respectively. Note that the extra computational burden for producing two kinds of VWVs is negligible since the most time consuming part of the procedure, i.e. the tree traversal, is executed only once for each feature descriptor.

Next, we make use of inverse indexing and keep track of the entries that share some common visual words. More specifically, each leaf node, w_i, is assigned with two lists: one holding the containing image indexes and another the sequence indexes. As new I-VWVs and S-VWVs are created, the inverse indexing lists are updated and associate images/sequences with common visual words.

B. Sequence to Sequence Matching

Starting with our first level of matching, each time a new sequence is formed, a similarity measurement between the most recent (query) and all the previously obtained (database) S-VWVs needs to be evaluated. We adopt the L_1-score as a similarity metric to find loop closure candidates between the query (S_q) and every sequence (S_d) in the database set $\mathscr{D}$ that the sequence inverse indexing indicates:

$$L_1\left(\bar{\boldsymbol{v}}_q^{(S)}, \bar{\boldsymbol{v}}_d^{(S)}\right) = 1 - 0.5 \left| \frac{\bar{\boldsymbol{v}}_q^{(S)}}{\left|\bar{\boldsymbol{v}}_q^{(S)}\right|} - \frac{\bar{\boldsymbol{v}}_d^{(S)}}{\left|\bar{\boldsymbol{v}}_d^{(S)}\right|} \right| \tag{5}$$

This metric ranges in $[0, 1]$, with similar sequences achieving higher scores. It is essential to point out here the importance of the proposed S-VWV to S-VWV matching scheme compared to the existed notion of summing similarity scores between I-VWVs. Some existing techniques, e.g. [12], are based on grouping the images into sets based on their time proximity and accumulating the scores between their image-members to detect loop closures. Although those approaches offer legitimate results, they fail to offer a global similarity measurement since the obtained visual words are distributed over multiple description vectors. Thus, they are subjected to a per camera perception of the scene and may produce misleading conclusions. On the contrary, our method formulate a global description vector treating the whole sequence as a single aggregation of observed visual words.

The produced by Eq. 5 sequence matching scores form a similarity matrix, which is obtained incrementally during the acquisition of new sequences, as the one presented in Fig. 2(a). This matrix is symmetric since each one of its (i, j) elements contains a corresponding $L_1\left(\bar{\boldsymbol{v}}_i^{(S)}, \bar{\boldsymbol{v}}_j^{(S)}\right)$ value. In addition to L_1-score, matching scores that tend to remain high between sequences that jointly escalate in time receive a bonus. As an example, for a time window of $h = 3$ sequences, if the similarities of S_{q-1}-to-S_{d-1}, S_q-to-S_d and S_{q+1}-to-S_{d+1} are high, we desire to advance the score of S_q-to-S_d by some factor. On the contrary, the score of a sequence that tends to present high similarity with more than one members of $\mathscr{D}$ is considered ambiguous and receives a penalty. Figure 3 contains examples for both cases. Those two notions reflect a quantitative interpretation of temporal consistency [27] and can be combined by applying a 2-dimensional filter with the following kernel on the similarity matrix's entries:

$$H = \begin{bmatrix} \alpha & -\beta & 0 \\ -\beta & 1 & -\beta \\ 0 & -\beta & \alpha \end{bmatrix} \tag{6}$$

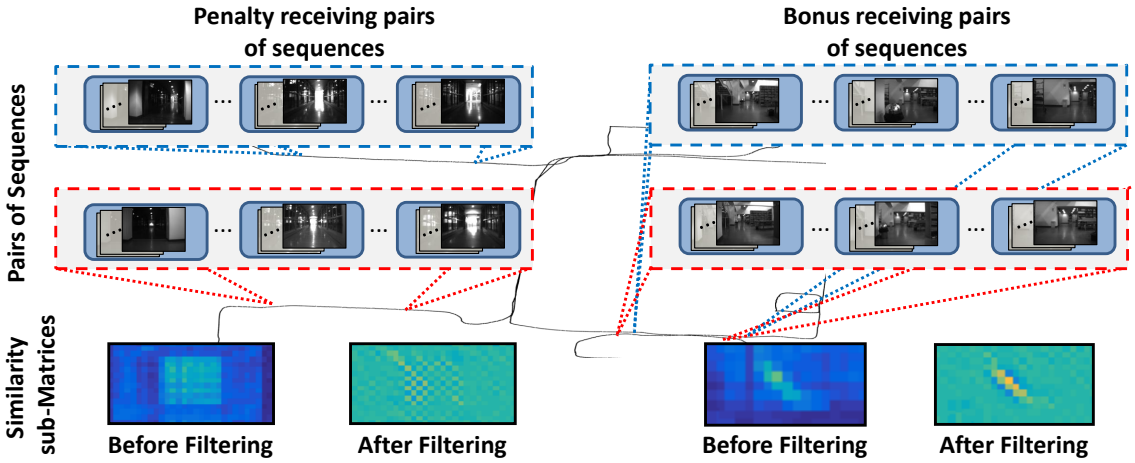


Fig. 3. Penalty (left) and bonus (right) receiving pairs of sequences. This particular example illustrates some actual cases of sequences from the *Bicocca25b* dataset together with their corresponding similarity sub-matrices.

TABLE I
PROPERTIES OF THE USED DATASETS

Dataset Name	Description	Avg. Speed (m/s)	Measurements Frequency (Hz)
Bicocca 2009-02-25b	Indoors Frontal camera Static	0.52	3.75
New College	Outdoors Frontal camera Dynamic	0.94[1]	20
City Centre	Outdoors, Urban Lateral camera Dynamic	N/A	N/A
Malaga 2009 Parking 6L	Outdoors Frontal camera Slightly dynamic	2.75	7.14

where $\alpha, \beta > 0$. This kernel refers to a time window of size $h = 3$, but it can be expanded to support bigger filter sizes. Analogous attempts for influencing the similarity matrix can be found in [12], [13], yet in our case we attempt to topologically interpret the manipulation, yielding a more intelligible matrix as the one seen in Fig. 2(b). It is worth noting here that, since inverse indexing is applied, only the needed similarity sub-matrices are calculated and filtered incrementally, allowing for an online formulation of the whole matrix. Pairs of sequences (S_{q_i} and S_{d_j}) above a threshold a are considered to contain loop closing image candidates; the next step of the proposed algorithm is their subset *i2i* association.

C. Image to Image Matching

In order to find the best *i2i* matches we utilize the individual I-VWVs. Each frame $I_i^{(S_{q_i})} \in S_{q_i}$ is compared only with the ones in sequence S_{d_j} that the image inverse indexing list indicates and it is associated to its best match $I_j^{(S_{d_j})} \in S_{d_j}$. A temporal consistency check is also applied here, which retains a sequential increment of the matched image indexes. This essentially means that since the first image, $I_1^{(S_{q_i})}$, finds its best matching $I_x^{(S_{d_j})}$, the next one, $I_2^{(S_{q_i})}$, can only be matched with one of $I_y^{(S_{d_j})}$ $\left(y \in [x, ..., k_{d_j}]\right)$ and so forth. Note that the inverted matching orientation (i.e. $y < x$) is less likely, since the respective scenes typically seem very different when observed from completely opposite directions, and thus it is excluded [9], [12]. Frames $I_i^{(S_{q_i})}$ which do not associate with any of the S_{d_j} members through the inverse indexes, are ignored.

As a final step, our approach computes the fundamental matrix between the related images in order to geometrically verify the results. A RANSAC based scheme provides our last resource for avoiding false positive loop closures. Image matches are rejected if they fail to provide a fundamental matrix or if the number of the feature point inliers is less than a constant f. A reduction of the feature-to-feature matching candidates can be achieved here by means of the direct indexing [9], [19], which associate the descriptors' corresponding visual words to the tree's parent nodes. This way, only the features that share the same parent, at a certain level of the tree (l), are going to be checked.

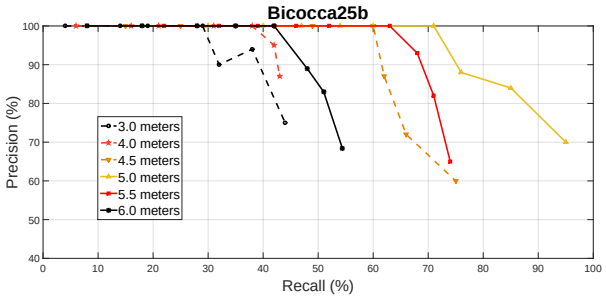
IV. EXPERIMENTAL EVALUATION

In this section, we present the parameter tuning and optimization of our system and finally compare our results with other state of the art algorithms for loop closure detection.

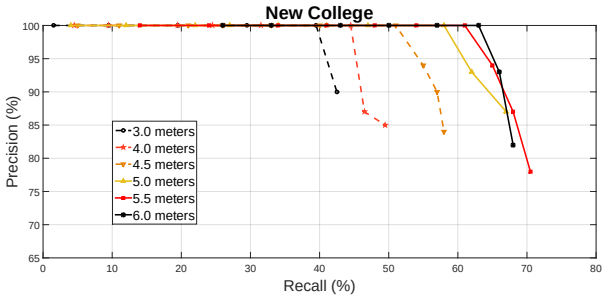
A. Experimental Protocol

With a view to measuring the accuracy of our system, we have chosen to use Precision-Recall curves. Precision is defined as the ratio between the true and the total number of detected loop closures. On the other hand, Recall is equal to the ratio of the correctly detected loop closures, to the total number of loops the dataset contains. Four datasets were used to evaluate the proposed method, namely Bicocca 2009-02-25b [24], New College [28][1], Malaga 2009 Parking 6L [29] and City Centre [7], with Table I containing a brief description

[1]This dataset was used with its improved visual odometry found at the authors' website.

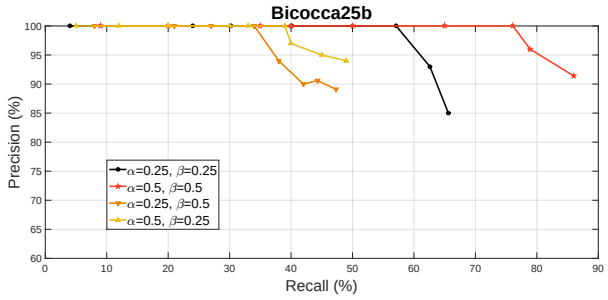


(a)

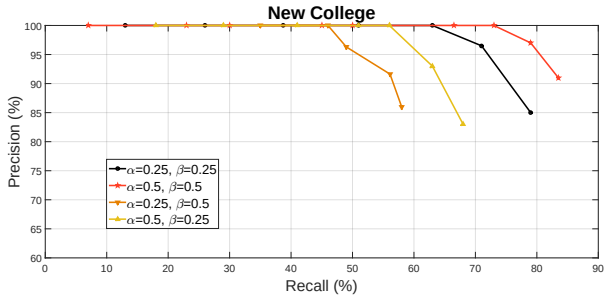


(b)

Fig. 4. Precision-Recall curves, for (a) *Bicocca25b* and (b) *New College* datasets, measuring the effect of length (μ) on the system's performance.



(a)



(b)

Fig. 5. Precision-Recall curves, for (a) *Bicocca25b* and (b) *New College* datasets, measuring the effect of Kernel H on the system's performance.

of them. The first two of these datasets were used to tune and select the unknown variables (thresholds, coefficients, etc) we introduced in previous sections, while the last two were chosen to measure our system's final performance. Thus, the achieved efficiency is not directly influenced by the algorithm's calibration and a robust evaluation is provided.

B. Algorithm Calibration

To start with, we need to determine the number of meters μ forming a sequence. We selected various testing cases for μ and varied threshold a without using the similarity matrix filtering, nor the geometrical verification, in order to produce Precision-Recall curves. Figures 4 (a) and (b) present some of the test cases that produced the most appealing results for the *Bicocca25b* and *New College* datasets respectively. Considering the first dataset, it appears that exceeding a limit of meters results in a reduction of the performance. This is owing to the fact that *Bicocca25b* is an indoor dataset, therefore visual changes tend to be more severe. A similar limit was found for the *New College*, but in this case, the performance was reduced for sequence sizes higher than 15 m. In a given real case scenario, the number of meters composing a sequence can be adjusted accordingly to enhance the performance of a particular environment, but within the scope of this paper a general setting is considered and the value $\mu = 5$ m was selected.

The next set of variables that we need to determine is the coefficients of kernel H. Following the same procedure, we kept μ fixed to 5 m and we assessed a variety of α and β combinations using Precision-Recall diagrams. It is worth to point out here that H is a $rank$ 3 matrix with the respective basis vectors forming a 3-dimensional space. Thus, the filter's effect is invariant to the basis vectors' scaling and the only factor affecting our results is their orientation. The most beneficial cases for both *Bicocca25b* and *New College* datasets are presented in Fig. 5 (a) and (b), respectively. Once again, the experiments were conducted without using the geometrical verification. As seen, the pair $[\alpha, \beta] = [0.5, 0.5]$ performs better and it is adopted as kernel's H coefficients. We also tried different filter sizes h without any improvement in the performance.

For the rest of the algorithm we preserved the implementation parameters proposed by the authors of DBoW2. In particular, we kept their selected setup for the geometrical verification and direct indexing, since we confirmed their superior performance for the *i2i* matching.

C. Overall Performance

Figure 6 illustrates the Precision-Recall curves for the two testing datasets, *Malaga6L* and *City Centre*. We obtained those curves by varying threshold a and considering *i2i* matches. As it can be seen, our method maintains high Recall percentage for 100% Precision due to the advantages of adopting a VWV-based scheme on a sequence descriptor. Even for the case of *City Centre*, one of the most demanding dataset in literature, the proposed methodology retains 100% Precision for Recall magnitudes as high as 68%.

Comparative results are presented in Table II with some of the most well established techniques available in the literature for the same datasets, viz. [9], [10], [30]. The achieved performance of the aforementioned algorithms was obtained straightforwardly from their respective papers, while the

TABLE II
COMPARATIVE RESULTS

Dataset	Approachs	Precision (%)	Recall (%)
Bicocca25b	**DBoW2** [9]	**100**	**81.20**
	Mur-Artal [10]	100	76.60
	FAB-MAP 2 [30]	100	N/A
	Proposed	100	78.10
New College	DBoW2 [9]	100	55.92
	Mur-Artal [10]	100	70.29
	FAB-MAP 2 [30]	100	N/A
	Proposed	**100**	**77.55**
Malaga6L	DBoW2 [9]	100	74.75
	Mur-Artal [10]	**100**	**81.51**
	FAB-MAP 2 [30]	100	68.52
	Proposed	100	76.78
City Centre	DBoW2 [9]	100	31.61
	Mur-Artal [10]	100	43.03
	FAB-MAP 2 [30]	100	38.77
	Proposed	**100**	**68.49**

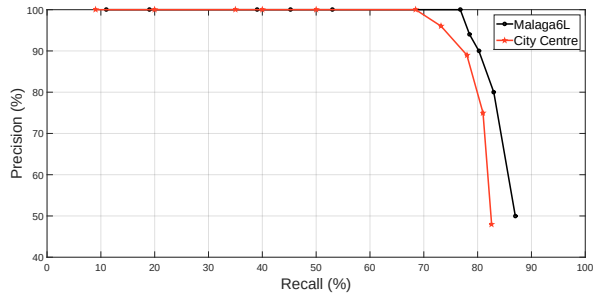


Fig. 6. Precision-Recall curves, for *Malaga6L* and *City Centre* datasets, illustrating the final system's performance.

proposed system's set of used variables is summarized in Table III. As one can observe, our method achieves better and more consistent results in most cases, due to the rich descriptiveness that *s2s* matching offers. In the cases of *Bicocca25b* and *Malaga6L* datasets the achieved performance of our approach is not proportional to the rest of the datasets' due to some sharp turns in their trajectories. We observed that in those cases, some loop closing images happen to fall in the same sequence or the sequences themselves overlap for small regions, thus failing to match with each other. Those issues are originated from the fact that sequences are arranged based on one fixed route distance (μ), rather than substantial changes in the scene appearance. In such cases, a dynamically chosen μ, based on common visual information, can be applied, as to be discussed in the final section. Finally in Fig. 7, the actual loops that our algorithm was able to detect are presented for all the used datasets.

V. ALGORITHM COMPLEXITY

Along with the algorithm's effectiveness, we also examine its computational complexity. The experimental results in [9], [10] reveal that the execution time of the loop closure detection grows proportionally to the length of the executed robot's trajectory. This effect is due to the growth of the database set of images that the query image is going to be compared with. In this section we prove that our first level of *s2s* matching decreases the loop closure candidates, thus ensures a reduction in the computational cost.

TABLE III
SELECTED PARAMETERS

Vocabulary Tree's Branching Factor (K)	10
Vocabulary Tree's Levels (L)	6
Number of Meters per Sequence (μ)	5
Filter's kernel factors [α, β]	[0.5, 0.5]
Sequence Matching Threshold (a)	0.32
Direct Indexing Level (l)	2
Min Feature Matches after RANSAC (f)	12

The matching procedure exhibits a quadratic complexity, which is inevitable to dominate the execution time as the course escalates, especially for cases of long trajectories. Considering an example of a system without the first level of *s2s* matching and a long trajectory with n acquired frames, a total of $n^2/2$ comparisons needs to be performed to associate images with high similarity. In any case, inverse indexing reduces the number of comparisons, but for simplicity we will assume that there exists at least one visual word which is common for every obtained image. In fact, inverse indexing is applied in the first level of the proposed pipeline as well. Thus, it causes the same effect on the multitude of sequences comparisons and it is omitted from this section.

On the contrary, our method groups the images into sets with an average size of $\overline{k}$ and initially uses their corresponding sequence descriptors to recognize places containing loop closing frames. Thereby, the number of comparisons is reduced to approximately $(n/\overline{k})^2/2$ on a first level and later, only for the sequences that overcome threshold a, finds the image associations that corresponds to loop closures. It is worth noting that, using threshold $\mu = 5$ on the four assessed datasets and without considering the inverse indexing effect, the average number of comparisons was reduced from about $10200^2/2$ (*i2i*) to $270^2/2$ (*s2s*).

Finally, in order to evaluate the actual influence of the aforementioned reduction of matching candidates, we formed a timing experiment utilizing the biggest of our tested datasets, *New College*. Using an Intel-i7@2.4GHz CPU and 4GB of memory, we were able to run our C++ based algorithm on $25K$ images obtaining a total execution time of 17.8ms per input frame on average, with 9.1ms being the minimum and 43.2ms the maximum.

VI. CONCLUSIONS AND FUTURE WORK

In this work a novel pipeline for loop closure detection has been proposed, where spatially arranged batches of images are grouped together to form descriptors of places. Instead of accumulating the matching scores from single frames obtained close in time, local features from each of them are combined to create a single VWV, as if they were originated from one super-frame. Thus, the general view/content of a scene can be used to detect revisited trajectory regions, while the loop closing pairs of images are recovered only for

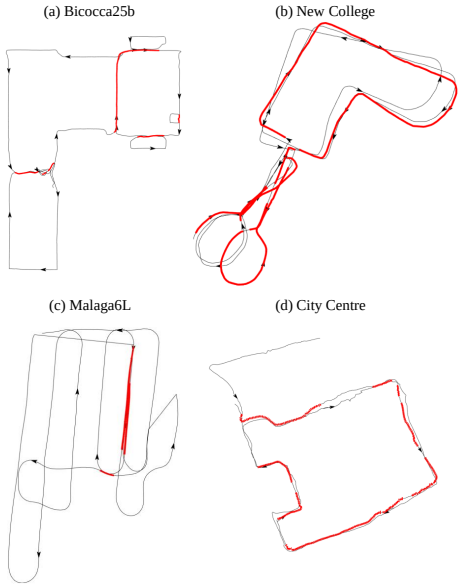


Fig. 7. Loop detection results (highlighted with red) for each tested dataset.

them. Experimental results show that the proposed technique achieves higher performance than other state of art methods while reducing the execution time of the matching function for long trajectories.

With a view to enhancing our proposal, a dynamic sequence length based on the variance of the visual information needs be tested in the future. More specifically, the starting and ending points of a sequence can be selected considering a ratio between new and previously obtained visual words. Thus, the image sequences would be distinguished according only to their content, potentially improving the system's performance. Additionally, even though the main objective of this paper is the introduction of the S-VWVs to the existing literature, robust rotation and scale invariance of the description needs to be further exploited by evaluating more sophisticated types of binary features.

ACKNOWLEDGMENT

Special thanks to Dorian Gálvez-López for kindly providing loop closure ground truth for the used datasets.

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] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

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

[10] 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.

[11] G. Schindler, M. Brown, and R. Szeliski, "City-scale location recognition," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, 2007, pp. 1–7.

[12] P. Newman, D. Cole, and K. Ho, "Outdoor SLAM using visual appearance and laser ranging," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2006, pp. 1180–1187.

[13] 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.

[14] 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.

[15] I. Kostavelis and A. Gasteratos, "Learning spatially semantic representations for cognitive robot navigation," *Robotics and Autonomous Syst.*, vol. 61, no. 12, pp. 1460–1475, 2013.

[16] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded Up Robust Features," in *Proc. European Conf. Comput. Vision*, 2006, pp. 404–417.

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

[19] D. Gálvez-López and J. D. Tardós. (2012) DBoW2: Enhanced hierarchical bag-of-word library for C++. [Online]. Available: http://doriangalvez.com/software

[20] 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.

[21] E. Pepperell, P. I. Corke, and M. J. Milford, "All-environment visual place recognition with SMART," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2014, pp. 1612–1618.

[22] R. Arroyo, P. F. Alcantarilla, L. M. Bergasa, J. J. Yebes, and S. Bronte, "Fast and effective visual place recognition using binary codes and disparity information," in *Proc. IEEE Int. Conf. Intelligent Robots and Syst.*, 2014, pp. 3089–3094.

[23] 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.

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

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

[26] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proc. European Conf. Comput. Vision*, 2006, pp. 430–443.

[27] 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. Robot.*, vol. 24, no. 5, pp. 1027–1037, 2008.

[28] 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.

[29] 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.

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