Automatic Visual Bag-of-Words for Online Robot Navigation and Mapping

Tudor Nicosevici and Rafael Garcia, *Member, IEEE*

***Abstract*—Detecting already-visited regions based on their visual appearance helps reduce drift and position uncertainties in robot navigation and mapping. Inspired from content-based image retrieval, an efficient approach is the use of visual vocabularies to measure similarities between images. This way, images corresponding to the same scene region can be associated. State-of-the-art proposals that address this topic use prebuilt vocabularies that generally require *a priori* knowledge of the environment. We propose a novel method for appearance-based navigation and mapping where the visual vocabularies are built online, thus eliminating the need for prebuilt data. We also show that the proposed technique allows efficient loop-closure detection, even at small vocabulary sizes, resulting in a higher computational efficiency.**

***Index Terms*—Field robots, localization, loop-closure, mapping, simultaneous localization and mapping (SLAM).**

I. Introduction

GLOBAL positioning from local observations represents a topic with multiple applications in robot navigation and mapping. It allows estimating the location of the robot within a visual map with little or no *a priori* knowledge of its position. This approach leads to multiple applications: loop-closure detection, positioning based on previously created maps (kidnapped robot problem), position recovery after occlusions or failures, etc.

The detection of situations where a robot visits a previously mapped area—loop-closure problem—is of particular interest in the simultaneous localization and mapping (SLAM) community. If correctly detected, such situations can be exploited, allowing correction of drifts accumulated during the navigation and mapping, using either offline or online approaches.

Visual-based loop-closure detection is a well-known inherently complex problem due to the amount of data that need to be analyzed, since typical image feature extractors yield thousands of features per image.

Manuscript received June 16, 2011; revised December 9, 2011; accepted March 20, 2012. Date of publication May 10, 2012; date of current version August 2, 2012. This paper was recommended for publication by Associate Editor D. Kragic and Editor G. Oriolo upon evaluation of the reviewers' comments. This work was supported in part by European Union Project FP7-ICT-2011-288704 and the Spanish Ministry of Science and Innovation under Grant CTM2010-15216.

T. Nicosevici was with the Computer Vision and Robotics Group, University of Girona, 17071 Girona, Spain. He is now with Coronis Computing Ltd., 17003 Girona, Spain (e-mail: tudor@eia.udg.edu).

R. Garcia is with the Computer Vision and Robotics Group, University of Girona, 17071 Girona, Spain (e-mail: rafael.garcia@udg.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2012.2192013

In [1] and [3], a loop-closing detection method is proposed that computes the visual similarity using features. During navigation, they extract key points from each image (e.g., SIFT [4]). These features are matched among images and the visual similarity is proportional to the number of successfully matched features. Generally, such methods are sensitive to occlusions while being computationally expensive, limiting their application over large navigation trajectories.

A robust and computationally efficient solution is to represent entire images as observations rather than individual image features. In this context, loops are detected on the basis of image similarity, drastically decreasing the amount of data that need to be processed. The reduced computational cost related to such approaches enables brute force loop-closure detection, even for large robot trajectories. This allows correct detection of trajectory loops, independent of robot pose and covariance estimation accuracy.

Initial proposals on image similarity loop detection based on use image representations are based on a single global descriptor, embodying visual content, such as color or texture [5]–[7]. Such global descriptors are sensitive to viewpoint and illumination changes, decreasing the robustness of the loop detection.

The emergence of modern viewpoint invariant feature extractors and descriptors, inspired from text document indexing, has led to the development of so called visual bag-of-words (BoW) methods. BoW image representation employs two stages: 1) In the training stage, sets of visual features are grouped or clustered together to generate *visual vocabularies*, i.e., collections of generalized visual features or *visual words*; 2) in the second stage, the images are represented as histograms of visual word occurrences. While discarding the geometric information in images, BoW proved to be very robust methods to detect visual similarities between images, allowing efficient loop-closure detection even in presence of illumination and camera perspective changes, partial occlusions, etc.

Initially, BoW techniques have been developed for object recognition and content-based image retrieval applications. Such methods use a training set of images from which the visual vocabularies are built, mostly using k-means clustering [8]–[10], where the user is required to specify the number of visual words in the vocabulary (for a detailed comparison of different clustering strategies, see [11]). Alternatively, other works have proposed the use of hierarchical k-means or approximated k-means, increasing the efficiency of the vocabulary building process for large training datasets [12], [13].

Schlinder *et al*. [14] propose the use of kd-trees to build a visual vocabulary as proposed by Nister and Stewenius [13]. The vocabulary is then used for SLAM at the level of a city with

1552-3098/$31.00 © 2012 IEEE

good results. Galvez-Lopez and Tardos [15] propose the use of a vocabulary based on binary features for fast image matching.

Konolige *et al*. [16] propose a two-stage method in which visual vocabularies are first used to extract candidate views followed by a feature-based matching.

The main shortcoming of the aforementioned methods is the use of a static vocabulary: The vocabulary is built *a priori* and remains constant during the recognition stage, failing to accurately model objects or scenes not present during training [17]. This shortcoming is particularly critical in the case of mapping and navigation, where a robot should be able to successfully detect loop-closure situations in uncontrolled environments. As a consequence, a group of authors in the SLAM community has proposed alternatives to address this problem. Notably, Filliat [18] and Angeli *et al*. [19], [20] assume an initial vocabulary which is gradually incremented with new image features in an agglomerative fashion, using a user-defined distance threshold as the merging criterion. Alternatively, in [21]–[25], a large-scale loop-detection probabilistic framework is porposed based on BoW. They show good results employing k-means-based static vocabularies built from large sets of visual information, not necessarily acquired in the same areas where the robot navigation takes place. As an alternative, Zhang [3] proposes a workaround to the offline vocabulary building stage by describing images directly using visual features, instead of vector-quantized representation of BoW. Here, the complexity of raw feature matching for loop-closure detection is partially reduced by means of a feature selection method that reduces the number of features extracted from images.

We propose a novel method to build online visual vocabulary (OVV). This paper extends the work presented in [26], offering an improvement in the vocabulary building and feature-cluster association mechanisms. In addition, here, we present a more complete and comprehensive set of experiments that further validates the proposal.

The proposed approach is aimed to increase the efficiency and accuracy of loop detection in the context of online robot navigation and mapping. It requires no user intervention and no *a priori* information about the environment. OVV creates a reduced vocabulary as soon as visual information becomes available during the robot survey. As the robot moves, the vocabulary is constantly updated in order to correctly model the visual information present in the scene.

Current state-of-the-art clustering methods, such as k-means, k-medians, or agglomerative, use local cluster relationships as basis for the merging criterion, resulting in a high probability for these algorithms to get stuck in a local minima. In contrast, we propose a new clustering criterion that takes into account the entire distribution of the clusters, increasing the efficiency of the resulting vocabularies. In addition, we present a novel method for feature-cluster association and image indexing suited for incremental vocabularies.

The remaining of this paper is structured as follows. The following section proposes a novel vocabulary building method, followed by a proposal of a new image indexed method. The OVV process is then validated through a series of experimental results, including a 18.5-km trajectory dataset, along with the application of OVV on large-scale 3-D reconstruction and mapping for land and underwater environments. This paper concludes with some remarks and proposal for further work.

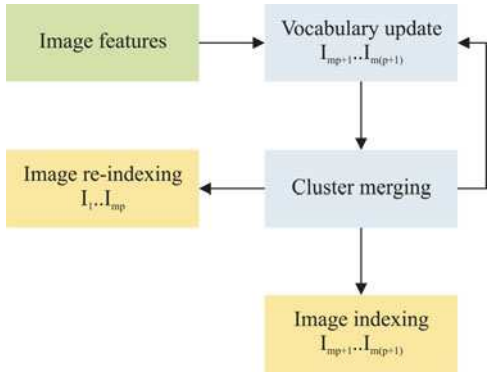


Fig. 1. *Flowchart of OVV and image indexing*. Every m frames, the vocabulary is updated with new visual features extracted from the last m frames. The complete set of features in the vocabulary is then merged until convergence. The obtained vocabulary is used to index the last m images. In addition, the previously indexed frames are reindexed to reflect the changes in the vocabulary.

II. Visual Vocabulary

State-of-the-art visual vocabulary-based loop-closure algorithms assume an initial training stage. This stage involves preacquiring visual features, which are then used to build the visual vocabulary by means of some clustering methods. Typical vocabulary building methods use k-means, k-medians, or fixed-radius clustering algorithms, which require the user to set various parameters such as the number of clusters in the vocabulary, or some distance threshold. Finding the adequate parameters for an optimum vocabulary is a tedious task which generally involves a trial-and-error approach. For instance, a vocabulary with too many words would not have enough abstraction power to detect similarities between images. In contrast, a vocabulary with too few words would be too confusing and generalized to be discriminative.

In this paper, we propose a novel incremental visual vocabulary building technique that is both scalable (thus suitable for online applications) and automatic (see Fig. 1). In order to achieve this goal, we use a modified version of agglomerative clustering. Agglomerative clustering algorithms begin with each element as a separate cluster—called hereafter *elementary clusters*—and merge them using some similarity measurement into successively larger clusters until some criterion is met (e.g., minimum number of clusters, maximum cluster radius, etc.).

A. Vocabulary Building

In our proposal, elementary clusters are generated from visual tracking of scene points, with each elementary cluster corresponding to one feature track. The feature tracks are generated by gathering multiple observations of the same scene point, as the camera moves [27]. While not required by OVV, this step allows us to preselect the number of visual features used in building the vocabulary, decreasing the computational costs.

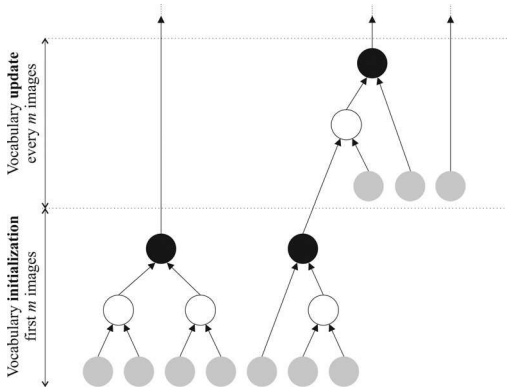


Fig. 2. *Iterative visual vocabularies*. In the initialization step (bottom part), the vocabulary is populated with elementary clusters (marked in gray) extracted from the first m images. These clusters are merged until convergence. The final clusters of the initialization step are marked in black. In the update step (top part), new elementary clusters that are obtained from blocks of m images are added to the vocabulary. The complete set of clusters is then merged until convergence.

The visual vocabulary is built by incrementally merging these clusters. The building process can be summarized in two steps (see Fig. 1).

1) *Vocabulary initialization step:* The vocabulary is initialized with the elementary clusters corresponding to the first m images. Clusters are gradually merged until convergence is achieved (the merging criterion is discussed in detail in Section II-D).
2) *Vocabulary update step:* As the robot moves, more visual information of the scene becomes available, which needs to be contained in the vocabulary. Therefore, from every block of m images, new elementary clusters are extracted. These clusters are added to the vocabulary and the complete set of clusters is gradually merged until convergence. This step is repeated for each block of m new images.

B. Cluster Characterization

Each cluster in the vocabulary is defined by its position in the t-dimensional feature space and its size (radius). This provides complete information about both the cluster distribution and the interaction between clusters. As previously shown, all the input information (for both initialization and update) comes from elementary clusters, such that all the other clusters in the vocabulary are formed by merging these clusters. As the elementary clusters are generated from feature tracking that provide multiple (noisy) observations of a scene point, we define them through

$$C_k = \frac{\sum_{i=1}^{n} f_k^i}{n} \tag{1}$$

$$R_k = \frac{\sum_{i=1}^{n} (f_k^i - C_k)(f_k^i - C_k)^{\mathrm{T}}}{n-1} \tag{2}$$

where C_k is the cluster centroid given by the mean of feature vectors corresponding to scene point k in image i, and R_k is the covariance matrix of the observations of point k.

C. Cluster Updating

Each cluster merging involves the joining of two clusters (see Fig. 2). The parameters of the newly generated cluster are obtained directly from the merged clusters, without the need to recompute them from the original data. This saves both computational time and memory, especially in the case of large clusters. The position and size of the new cluster are given by [28]

$$C_{ab} = \frac{n_a C_a + n_b C_b}{n_a + n_b} \tag{3}$$

$$R_{ab} = \frac{n_a - 1}{n_a + n_b - 1} R_a + \frac{n_b - 1}{n_a + n_b - 1} R_b + \frac{n_b \cdot n_a}{(n_a + n_b)(n_a + n_b - 1)} \cdot [(C_a - C_b)(C_a - C_b)^{\mathrm{T}}] \tag{4}$$

where C_a and C_b are the centroids of the merging clusters, having n_a and n_b elements, respectively.

D. Cluster Merging Criterion

Generally, clustering algorithms use some similarity measurement to decide which data should be grouped into clusters. Similarity measurements are often represented by distances in the t-dimensional data space, including Euclidean distance, Manhattan distance [29], Chebyshev norm [30], Mahalanobis distance [31], vector angle, etc. These clustering criteria analyze the data only locally and can be suboptimal, especially in high-dimensional cluttered spaces such as those used for visual feature representation.

We propose a novel clustering method that takes into account the global distribution of data, increasing both the distance between clusters and their compactness. This is crucial, as the efficiency of visual vocabularies is determined by two properties: 1) *repetitiveness*: similar image features should be associated with the same cluster and 2) *discriminative power*: Dissimilar image features have to be associated with different clusters.

The proposed method, based on Fisher's linear discriminant [32] [33], clusters the data in order to maximize the following objective function:

$$Q = \frac{tr(S_B)}{tr(S_W)} \tag{5}$$

where $tr()$ is the trace operator; S_B represents the *between clusters scatter matrix*, and S_W represents the *within clusters scatter matrix*, which are defined, respectively, by

$$S_B = \frac{1}{N} \sum_{k=1}^{N} n_k (C - C_k)(C - C_k)^{\mathrm{T}} \tag{6}$$

$$S_W = \frac{1}{N} \sum_{k=1}^{N} n_k R_k \tag{7}$$

where C is the global centroid of the data, N represents the total number of data elements, and n_k is the number of data elements contained in cluster k.

Practically, the merging takes place in two steps.

1) For each cluster, we search for merging candidates in its neighborhood (in the Euclidean sense), using a k-dimensional tree (kd-tree) approach [34].
2) For each possible merging pair of clusters, we compute the objective function Q' that would be obtained if the two clusters were merged. If there is an increase in the value of the objective function, then the two clusters are merged, and Sb and Sw are updated accordingly.[1]

Each merging step changes the distribution of data in the vocabulary, requiring the recomputation of both S_B and S_W. As a direct recomputation would be very costly, we propose an incremental update scheme:

$$\begin{aligned} S'_B = S_B &+ \frac{n_a + n_b}{N}(C - C_{ab})(C - C_{ab})^{\mathrm{T}} \\ &- \frac{n_a}{N}(C - C_a)(C - C_a)^{\mathrm{T}} \\ &- \frac{n_b}{N}(C - C_b)(C - C_b)^{\mathrm{T}} \end{aligned} \tag{8}$$

$$\begin{aligned} S'_W = S_W &+ \frac{n_a + n_b}{N}(R_{ab}) \\ &- \frac{n_a}{N}(R_a) - \frac{n_b}{N}(R_b) \end{aligned} \tag{9}$$

where S'_B and S'_W are the updates of S_B and S_W, respectively; C_{ab} and R_{ab} are the centroid and covariance matrix of the merged cluster.

E. Convergence Criterion

The two steps that are shown in Section II-D are repeated, gradually merging clusters, until no more merges are possible (that would increase the value of the objective function Q). This way, the repetitiveness and discriminative power of the resulting vocabulary are maximized. Moreover, using a natural convergence criterion, the process eliminates the need of user-set parameters such as cluster radius or number of clusters, specific to other vocabulary building algorithms.

F. Adding New Clusters

During the vocabulary update step, new elementary clusters are added, containing new visual features. For each newly added elementary cluster ζ_e, S_B and S_W have to be updated accordingly. Similar to the merging step, we avoid recalculating the scatter matrices by proposing a novel update method.

The update of S_W simply involves the covariance matrix R_e of ζ_e, weighted by its number of elements n_e:

$$S'_W = \frac{NS_W + R_e}{N + n_e}. \tag{10}$$

In the case of elementary clusters, n_e corresponds to the number of frames in which a given image feature has been tracked.

Adding any new cluster in the vocabulary affects the global data centroid C. The new centroid C' is incrementally obtained from

$$C' = \frac{CN + C_e n_e}{N + n_e}. \tag{11}$$

Taking into account the changes in the centroid C, S_B is updated using

$$\begin{aligned} S'_B = \frac{N}{N + n_e}&(S_B + \delta_C^{\mathrm{T}}\delta_C - V^{\mathrm{T}}\delta_C - \delta_C^{\mathrm{T}}V) \\ &- \frac{n_e}{N + n_e}(C_e - C')^{\mathrm{T}}(C_e - C') \end{aligned} \tag{12}$$

where $\delta_C = C' - C$, and V is the weighted sum of differences between each newly added cluster centroid and global data centroid. V is obtained incrementally by using

$$V' = \frac{NV + N\delta_C + n_e(C_e - C')}{N + n_e}. \tag{13}$$

G. Linear Discriminant Analysis

Using the cluster information contained in the visual vocabulary, we aim to find a data transformation that would maximize cluster separability and would allow the reduction of the dimensionality of the data, thus increasing the speed of both vocabulary building and image indexing. For this, we consider maximizing the following linear discriminant analysis (LDA) objective function [32], [33], [35]:

$$J(w) = \frac{w^{\mathrm{T}} S_B w}{w^{\mathrm{T}} S_W w} \tag{14}$$

where w is a vector that determines the maximum cluster separability direction. Formulating the maximization of $J(w)$ as a generalized eigenvalue problem, we obtain a data transformation G from the eigenvectors corresponding to w. By selecting m columns of G corresponding to the highest values of w, we reduce the dimensionality of the data to s dimensions.

H. Vocabulary Update Criterion

In Section II-A, for simplicity of explanation, we stated that the vocabulary is updated each m images. In practice, the vocabulary is updated adaptively, rather than at fixed intervals so that it constantly represents an accurate model of the visual content in images.

During image indexing, features are associated with clusters in the vocabulary. For each association of a feature f_l with a cluster ζ_k, we check if the feature falls within the cluster, using

$$|f_l - C_k| \leq 3\sigma_k \tag{15}$$

where σ_k is the standard deviation of cluster ζ_k. In (15), the absolute value $|\cdot|$ and the comparison are to be understood componentwise, i.e., only if the condition is met for all the dimensions, we consider that the feature falls within the cluster.

At each vocabulary update step, we index images until the percentage of features falling within the radius of their associated clusters drops below 90%. At this point, we update the vocabulary.

[1] In practice, we first compute the gain in Q for each possible merging pair, creating a list from the highest to the lowest gain. The clusters are merged following the order in the list, making the merging step independent of the order in which the clusters are analyzed.

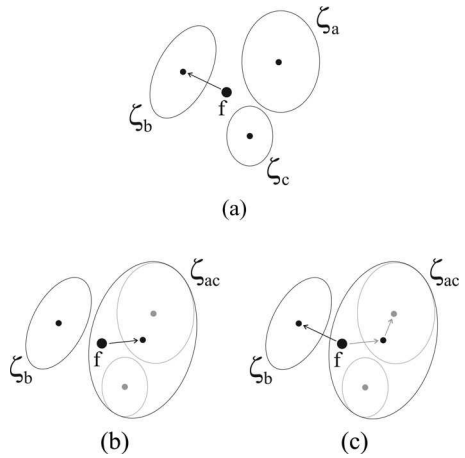


Fig. 3. *Feature-cluster association.* (a) Feature f is associated with cluster ζ_b, using feature-to-cluster centroid distance. After the vocabulary update, clusters ζ_a and ζ_c are merged. The centroid of the newly obtained cluster ζ_{ac} is now closer to f. Using a classical approach, feature f would be associated with ζ_{ac} (b). Using hierarchical trees, feature f is correctly associated with cluster ζ_b (c).

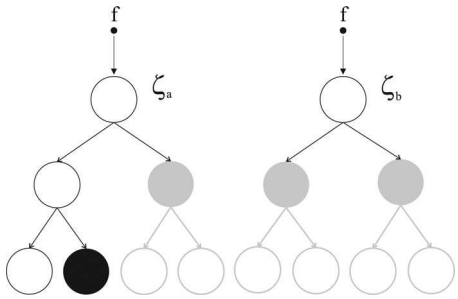


Fig. 4. *Top-down feature-cluster association.* The trees are visited by comparing each node with the feature. If a node is too dissimilar to the feature (marked in light gray), the rest of the tree corresponding to the node is not visited. The feature is associated with ζ_a because of the highest similarity between f and the leaf marked in black.

III. Image Indexing

Inspired from text document indexing [36], BoW techniques use visual vocabularies to represent the images by associating the features present in each of the images with the visual words in the vocabulary [37]–[39]. The result is a histogram that represents the number of occurrences of each visual word in the image. The similarity between images is calculated by comparing these histograms.

When detecting loop closures, it is paramount that image features are correctly associated with clusters, even in the presence of illumination and perspective changes. We partially achieve this by maximizing the repetitiveness and discriminative power of the vocabulary (see Section II-D). However, in the context of the online vocabulary, we need to define a third property: *stability*. As the vocabulary is constantly updated, the aim is to ensure that similar features are associated with the same clusters at different stages of the vocabulary update. We achieve this property through a novel feature-cluster association technique, as described in the following.

A. Cluster Association

The association between features and visual words is performed by comparing each feature with all the clusters in the vocabulary. The feature is then associated with the most similar cluster. Most image-indexing techniques calculate the similarity between features and clusters using distances in the feature space (see Section II-D). This approach is suitable for image indexing in the case of static vocabularies that are calculated before the image indexing and do not change throughout it [8].

Since we use an online approach for vocabulary building, such a feature association method would not be stable. In Fig. 3(a), feature f is associated with the closest cluster ζ_b. After the vocabulary is updated, clusters ζ_a and ζ_c are merged, yielding a new cluster ζ_{ac} [see Fig. 3(b)]. As the feature f is now closer to the centroid of the new cluster ζ_{ac}, it would be associated with it. In this case, feature f would be associated with different clusters before and after the vocabulary update. As a consequence, an image I_k containing feature f indexed at different vocabulary stages would have different representations. The amount of occurrences of such situations increase with each vocabulary update, ultimately leading to inconsistent image indexing.

Alternatively, the proposed feature-cluster association technique uses a tree-based approach. The trees are formed during the vocabulary building process. The nodes of the trees represent the clusters, while the branches define the cluster hierarchy. The roots of the trees correspond to the visual words, while the leaves of the trees correspond to the elementary clusters (see Fig. 2).

During the feature-cluster association, the trees are visited top-down, calculating the similarity (Euclidean distance) between each feature and the tree nodes [see Fig. 3(c)]. In order to speed up the association process, we visit only those trees corresponding to visual words in the vicinity of the feature. For this, we calculate the distance between the feature and the visual words and select the trees where

$$D(f, \zeta_k) \leq \tau D_m \tag{16}$$

with $D(f, \zeta_k)$ being the distance between feature f and ζ_k; D_m is the minimum distance between the feature f and the visual words, and τ is a user-defined constant[2] ($\tau \geq 1$).

The selected trees are visited in parallel (see Fig. 4). For efficiency purposes, we use the same stopping criterion shown in (16), hence avoiding visiting branches that contain nodes that are not close to f. The feature is finally associated with the visual word corresponding to the most similar leaf.

B. Image Reindexing

It should be taken into account that during the update process, the configuration of the vocabulary changes. Consequently, the similarity between images indexed at different update stages cannot be computed. In addition, indexing the images after each vocabulary update is not a viable solution because of its large computational cost.

[2]User parameter τ provides a balance between computational efficiency and accuracy of the image indexing. As shown in Section IV, optimum results are obtained using a typical value of $\tau = 1.4$, which is not data dependent.

We propose a novel solution to this shortcoming by defining a transformation ${}^{p}\Gamma_{p-1}$ that embodies the changes in the vocabulary during the update stage. This transformation allows a fast reindexing of the images (hence eliminating the need for repeated image indexing):

$$\widetilde{W}_I^p = {}^{p}\Gamma_{p-1} W_I^{p-1} \tag{17}$$

where W_I^{p-1} is the indexing of image I at vocabulary update stage $p-1$, and $\widetilde{W}_I^p$ is an approximation of the image indexing I at vocabulary update stage p.

During update, the visual vocabulary undergoes the following changes.

1) *Adding of elementary clusters:* If these new clusters are not absorbed into already existing clusters, they contain new visual information. In this case, it is very unlikely that any feature from any image before the update would have been associated with them. Therefore, the bins $\widetilde{W}_I^k$ are initialized to 0.
2) *Cluster merging:* In the case that two (or more) clusters merge, any feature previously associated with these clusters would be associated with the newly formed cluster. In this case, the number of occurrences associated with the new cluster is the sum of occurrences of the merging clusters.

To reflect these changes, ${}^{p}\Gamma_{p-1}$ has to initialize the histogram elements corresponding to newly added clusters and sum the elements corresponding to merging clusters. For a better understanding, let us consider the following example: At stage $p-1$, the indexing of image I yields $[w_1\ w_2\ w_3]^{\mathrm{T}}$ corresponding to the visual vocabulary containing $(\zeta_1,\ \zeta_2,\ \zeta_3)$; during the vocabulary update, clusters ζ_1 and ζ_2 merge into ζ_{12}, and a new cluster ζ_4 is added. In this case, the transformation ${}^{p}\Gamma_{p-1}$ becomes

$$\begin{bmatrix} w_{12} \\ w_3 \\ w_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}. \tag{18}$$

C. Image Similarity

The visual resemblance between images is quantified by measuring the similarity of their corresponding histograms of visual words. As the histograms are represented by vectors that contain the occurrences of the visual words, we calculate their similarity using the normalized scalar product (cosine of the angle between vectors) [8]:

$$s_{rq} = \frac{W_r^{\mathrm{T}} W_q}{\|W_r\|_2 \cdot \|W_q\|_2} \tag{19}$$

where s_{rq} is the similarity score between images I_r and I_q, and W_r and W_q are the histograms of visual words corresponding to the images; $\|W\|_2 = \sqrt{W^{\mathrm{T}} W}$ is the L_2 norm of vector W.

In (19), the similarity score is highly influenced by histogram elements corresponding to visual words with high occurrence. Generally, these frequent words represent visual features commonly found in the images, thus having low discriminative power. In order to counterbalance this shortcoming, the elements of the histograms are weighted using the *term frequency-inverse document frequency* approach proposed in [40]:

$$\overline{w}_k = \frac{n_{ki}}{o_i} \log \frac{m_p}{O_k} \tag{20}$$

where n_{ki} is the number of occurrences of word k in image I_i, o_i is the total number of words in I_i, O_k is the total number of images containing word k, and m_p is the total number of indexed images.

D. Loop-Closure Detection

Increased values of s_{rq} between the current image and any previous one indicate a high probability of the two images that represent the same scene region (i.e., loop closing). This information can be used to both introduce new constraints in the mapping model and reduce the navigation-related uncertainties.

E. Increasing Vocabulary Efficiency

During online navigation and mapping, new image features are extracted and added to the visual vocabulary. Over long image sequences, this could result in complex vocabulary structure that decreases the efficiency of OVV in terms of computational times. This effect is partially reduced by using approximate nearest neighbor techniques [34] on both vocabulary building and image indexing; however, we further improve the computational efficiency of OVV by pruning branches corresponding to nodes that provide little information, using the following criterion:

$$tr(R_k^i) < p \cdot tr(R_k) \tag{21}$$

where R_k^i is the radius of node i in cluster ζ_k, and p is a user-defined scalar value. In our experiments, we have found that using a value $p = 0.1$ provides a good balance between computational efficiency and accuracy of OVV.

IV. Experimental Results

In practice, the OVV process was implemented on top of an online feature-tracking algorithm. The algorithm extracts and tracks visual features (e.g. SIFT [4], speeded up robust feature (SURF) [41], maximally stable extremal regions (MSER) [42], HarrisAffine [43], etc.), which are used for online vocabulary building and image indexing.

We carried out a series of experiments in order to assess the influence of LDA dimension reduction s and relative threshold τ [see (16)] on the accuracy and computational times of OVV (for details, see [44]). The two parameters are user-set and provide a tradeoff between computational efficiency and accuracy of vocabulary building and image indexing. Experiments show that these two parameters are not data sensitive so that for all experiments, we used $\tau = 1.4$ and $s = 24$, which provide a good balance between speed and accuracy. We consider two images to correspond to a loop-closure situation when their visual similarity $s_{rq} \geq 0.45$.

This section discusses some experiments that are designed to evaluate the efficiency and accuracy of the two contributions of this paper: 1) incremental building of the vocabulary and 2) image indexing based on hierarchical trees. The efficiency and

accuracy of the OVV algorithm is tested using a data association and a comparison with ground truth.

In the first experiment, we provide a detailed analysis of OVV for a large-scale loop-closure problem in a mixed environment. The second part of the experimental results illustrate the application of OVV in 3-D robot navigation and mapping. For this purpose, we have implemented OVV in conjunction with an online structure from motion (SfM) algorithm—DPR-SfM [45]. During navigation and mapping, the visual features extracted by DPR-SfM are used to create a 3-D map of the environment, while they are simultaneously used for vocabulary building and image indexing. When a loop-closure situation is detected, the resulting information is used to correct the accumulated drift.

Essentially, in the second part, we show the use of OVV for 3-D navigation and mapping in the case of two distinct scenarios: 1) an urban environment and 2) and underwater environment. The latter was chosen due to the additional difficulties imposed by the underwater environment: the high rate of light absorption in the water decreases the range of cameras and the contrast in images; moreover, the scattering effect due to floating microscopic particles induces a blurriness effect, further decreasing the contrast of images, also inducing the "marine snow" effect. All these aspects decrease drastically the image quality, resulting in nosier less discriminant image features.

In order to provide an objective assessment of the proposed algorithm, in the first experiment, we carry out a comparison between OVV and a state-of-the art visual SLAM algorithm: FAB-MAP2 [23]. In the last two experiments, we continue the detailed analysis of OVV. However, in these cases, we were not able to obtain consistent results using FAB-MAP2 because of its inability to cope with high overlapping frame sequences, such as those provided by video cameras.



Fig. 5. *Mixed Environment Experiment—Image acquisition setup*. The dual-camera setup was mounted on a car during data acquisition.



Fig. 6. *Mixed Environment Experiment—Car trajectory during data acquisition*. The 18.5-km trajectory (overlayed in yellow, as recorded by the DGPS system) was chosen to include multiple loop closures. The starting point can be seen at the lower right corner.

A. Large-Scale Mixed Environment

In this experiment, we have acquired data corresponding to a large trajectory consisting of two part: 1) an urban area with well-structured and diversified visual characteristics and 2) an area mainly formed by a natural (more repetitive) landscapes depicting trees, grass, etc. Both scenarios are common in land-based robot/vehicle navigation. The data were acquired using a setup consisting of two Canon 50D DSLR cameras equipped with 24-mm Canon fixed lenses, mounted on a car (see Fig. 5). The dual-camera setup was used to increase the field of view of the image acquisition system, increasing the probability of detecting the loop-closure situations. For ground truth, we used a differential global positioning system (DGPS), allowing for accurate positioning, even in the case of low GPS coverage situations, which is often the case in urban areas. The setup, mounted on a car, was used to gather data that represent both urban and natural environments. Fig. 6 illustrates the trajectory of the car during data acquisition. Over the 18.5-km trajectory, the system was set to automatically acquire images every 0.5 s, resulting in a total of 11 500 images. The trajectory was chosen to include a series of smaller loops at the beginning of the sequence, along with two large loops: the first containing almost exclusively urban scenery and the second containing a combination of urban and natural environments.

For ground truth, the data recorded by the DGPS device were interpolated to match the image acquisition rate (the DGPS provides GPS fixes at a frequency of 1 Hz). Any images acquired within a distance of 30 m were considered to depict the same area, hence corresponding to a loop-closure situation (the distance threshold was estimated from the average distance from the cameras to the scene, and validated manually). The car orientation was not taken into account because of the large total field of view of the imaging system ($\sim 164^\circ$ horizontally).

Prior to feature tracking, the images were downsampled to 640×480 pixels resolution to simulate a low-end image acquisition system. The image features were extracted and described using SURF, yielding a total of 40 million feature tracks. The visual vocabulary was built online, during feature tracking. The final size of the vocabulary was $\sim 30\,000$ visual words. Fig. 7 shows the evolution of the vocabulary. At the beginning of the sequence, the vocabulary grows quickly. However, as scene features tend to repeat, the growth rate slows down at the middle of the sequence. The vocabulary grows again at the end of the sequence to model features corresponding to novel sceneries (natural environment).

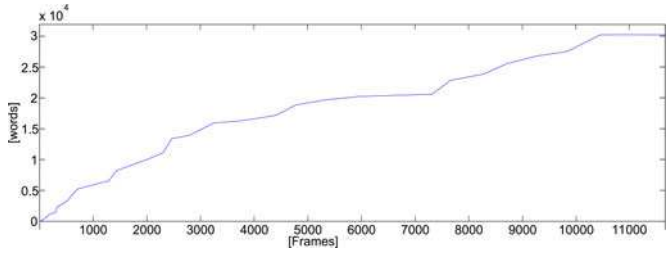


Fig. 7. *Mixed Environment Experiment—Vocabulary size evolution*. The vocabulary growth slows down at the middle of the sequence but increases at the end of the sequence as novel types of sceneries are imaged. There are 35 vocabulary updates in total.

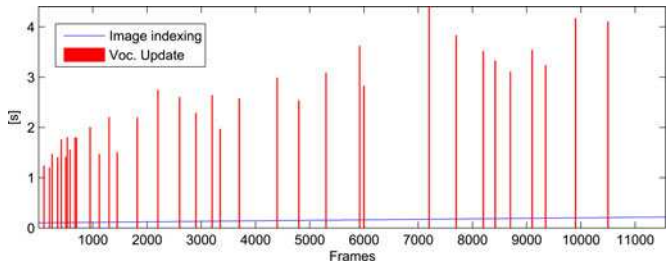


Fig. 8. *Mixed Environment Experiment—Vocabulary building and image-indexing computational times*. The vocabulary update step takes an average of 1.6 s/update, while the image indexing takes an average of 0.11 s/frame.

The entire process was run on an Intel 2 Quad machine running Windows 7. The execution times for vocabulary building and image indexing are shown in Fig. 8, where it can be observed that the vocabulary is being updated at short intervals at the beginning of the sequence, where high amounts of visual information are being learned by the vocabulary. The vocabulary update intervals decrease during the rest of the sequence, only when new visual information becomes available. The image-indexing times are maintained constant throughout the sequence. It should be mentioned here that currently, OVV is mainly implemented in MATLAB with some routines implemented in C++.

The precision of OVV was assessed by comparing extracting visually similar images as measure by OVV and comparing the result with the ground truth. Here, we make a comparison between the results obtained by the proposed algorithm and the results obtained by the FAB-MAP2 algorithm proposed by Cummins and Newman [23]. For this purpose, we ran FAB-MAP2 in two cases: 1) using a generic offline built visual vocabulary of 40 000 visual words, containing mostly urban visual data; and 2) using a generic offline built vocabulary containing 80 000 visual words, embodying data from various environments (indoor/outdoor, natural, etc.).

Fig. 9 shows the results of the precision/recall evaluation. The goal of this analysis is twofold: 1) to evaluate the performance of the incremental image indexing and 2) to compare the accuracy of OVV with respect to FAB-MAP2.

The incremental indexing was compared with the full indexing of the entire set of images using the vocabulary generated by OVV in its final form. Fig. 9 shows that the proposed incremental method closely approximates the full indexing, with

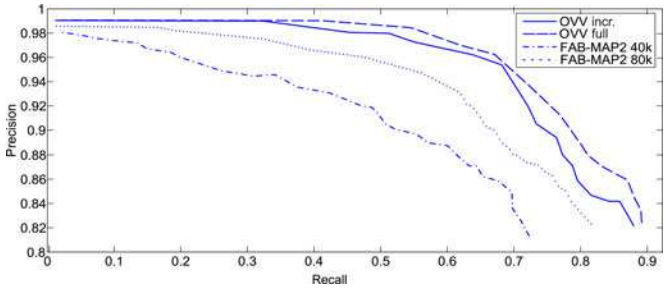


Fig. 9. *Mixed Environment Experiment—Precision/Recall evaluation*. Comparison between OVV in two cases: Using the proposed incremental indexing versus full reindexing using the final form of the vocabulary and FAB-MAP2 for two vocabulary cases.

very little loss in precision $\simeq 0.01$ with a gain in computational cost of $\simeq 30$ times.

On the other hand, it can be observed that OVV outperforms FAB-MAP2 in both cases, while using a smaller size vocabulary. OVV uses a 3-D camera pose model, while FAB-MAP2 uses a camera rotation model to check the geometrical consistency of the detected loop closures. Such stages highly reduce the number of false positives, thus increasing the accuracy of the algorithms. However, in this paper, we focus primarily on the accuracy and efficiency of measuring visual similarities between images; hence, the results presented here are those provided by the algorithms, with no geometrical consistency checks.

The evaluation of the algorithms was carried out using precision/recall analysis where the *precision* represents the ratio between the true detected loop closures and the total detected loop closures, and the *recall* represents the ratio between the true detected loop closures and the true loop closures.

A more detailed analysis of the results shows that OVV can cope with common challenges found in outdoor environments such as illumination and camera viewpoint changes, partial occlusions, moving pedestrians, cars, etc. Furthermore, all the loop-closure situations were successfully detected even at early stages of the navigation, where the vocabulary contained little visual information (see Fig. 10 for a few examples of detected loop-closure situations).

Nevertheless, there are a few cases where the erroneously matched images represent different locations, resulting into false loop-closure detections. As expected, these situations are related to strongly repetitive patterns in images, mostly related to natural sceneries: grass, trees, earth, etc., as illustrated in Fig. 11.

B. Underwater Experiment

This experiment is aimed to test the efficiency of the OVV method in describing natural unstructured environments for underwater robot navigation and mapping, under typical challenges found in this environment. The data were acquired in Tortugas, Florida Keys, using a Phantom Remotely Operated Vehicle of the University of Miami, Coral Gables, FL. The 1000-image sequence has a resolution of 720×530 pixels and depicts a region comprised mainly of rocks and sand. The sequence is



Fig. 10. *Mixed Environment Experiment—Successfully detected loop-closure situations*. Loop closures are successfully detected in the presence of camera viewpoint changes in dynamic environments (moving pedestrians, occlusion due to the presence of cars, etc.)



Fig. 11. *Mixed Environment Experiment—False positives in loop-closure detection*. Some of the wrongly detected loop-closure situations, mostly related to repetitive patterns.

characterized by repetitive textures, allowing the test of OVV algorithms in presence of increased *perceptual aliasing*.[3]

Fig. 12 illustrates the estimated 3-D model that contains 125 850 vertices and the camera trajectory. The online vocabulary was initialized using the feature tracks in the first 20 frames.

[3]The perceptual aliasing problem corresponds to scenes with poor or repetitive textures that are characterized by the fact that different regions of the scene appear similar to the camera.

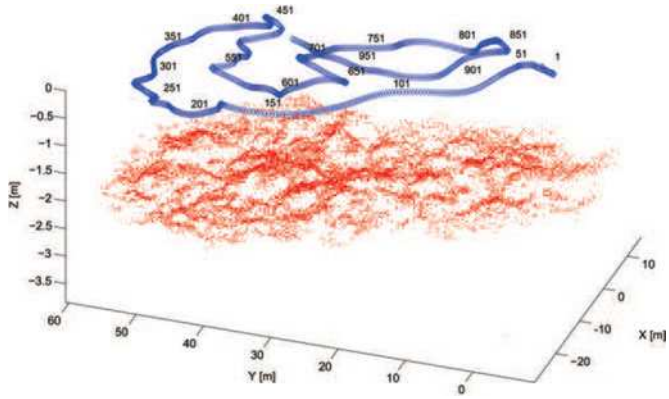


Fig. 12. *Underwater Experiment 2—Estimated 3-D model and camera trajectory*. The scene model shown in red contains $\simeq$ 126 000 vertices. The trajectory of the camera (blue) presents some partial overlaps.

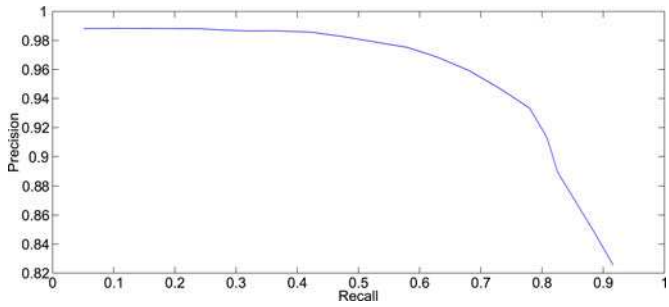


Fig. 13. *Underwater Experiment—Precision/Recall evaluation*. The maximum precision is slightly lower in this experiment mostly due to the perceptual aliasing.

During scene reconstruction, the vocabulary went through 15 updates, containing 6644 visual words, at the end of the sequence.

In the case of this experiment, no ground truth was available from navigation due to the lack of GPS coverage in the underwater environment. As an alternative, we exhaustively matched the feature between each pair of images in the sequence, estimating the overlap between all the possible image pairs using a projective homography model. We consider images with an overlap ratio higher than 0.5 to correspond to loop-closing situations (as the overlap denotes the fact that images correspond to the same region of the scene).

After comparing the results of OVV with the ground truth, the precision/recall curve (as illustrated in Fig. 13) shows a slightly decreased precision, with respect to other environments, due to the perceptual aliasing and the decrease in the image quality. This effect can also be observed in the image similarity matrix [see Fig. 14(a)], which is denoted by the slightly bright background corresponding to nonoverlapping images having a small degree of visual resemblance.

In order to detect the loop-closure situations, the image similarity matrix was binarized using a threshold of 0.45, which provides a good balance between precision and recall (thus minimizing the false positives and false negatives). This value for the binarization threshold was found to be optimum for all the experiments that we have carried out. The resulting loop-closure

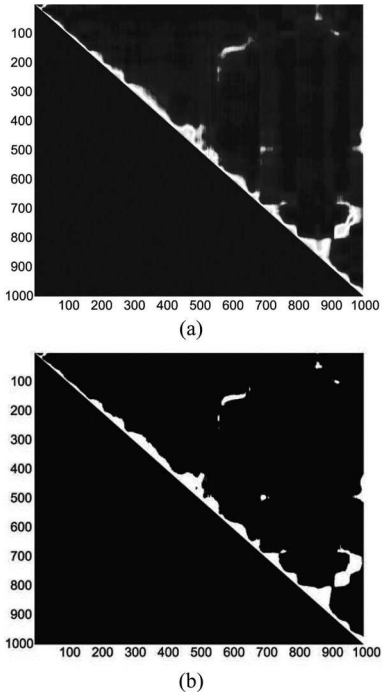


Fig. 14. *Underwater Experiment—Image similarity matrix and loop-closure detection*. (a) *Image similarity matrix*. Highlighted values off the main diagonal correspond to loop-closure situations. (b) Detected loop-closure situations after binarizing the image similarity matrix with a threshold of 0.45.

detection matrix [see Fig. 14(b)] clearly depicts areas where the robot revisits previously mapped areas.

Fig. 15 illustrates some of the pairs of images, corresponding to loop closures in the camera trajectory.

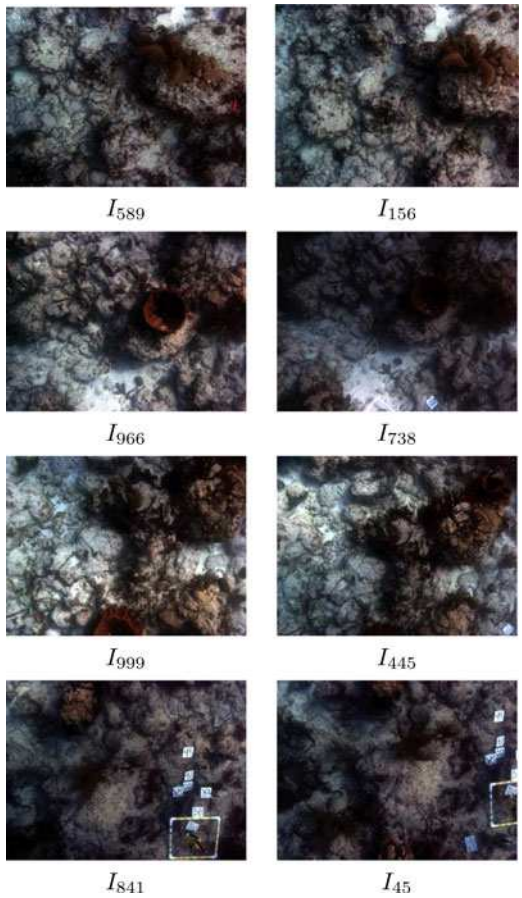


Fig. 15. *Underwater Experiment 2—Loop detection*. Pairs of images corresponding to some of the detected loop closures. Query frames are shown in the left column, and their corresponding most similar frames are shown in the right column.

C. Outdoor Experiment

The third experiment presented here was aimed to test the OVV algorithm for loop-closure detection in large-scale urban modeling applications. For this, we acquired a sequence of Unirii Square in Timisoara, Romania. The square, as illustrated in Fig. 16, has a rectangular shape, measuring $\simeq 155 \times 120$m and is surrounded by historical buildings of various shapes and textures. We used a Pentax Optio A30 compact digital camera for video acquisition, while walking through the square following a loop trajectory. The resulting image sequence contains 961 frames of 640×460 pixels in resolution (see Fig. 17).



Fig. 16. *Outdoor Experiment—Overview of the Unirii square*. Aerial view of the Unirii square.

After applying DPR-SfM on the sequence using SURF, the resulting model that is shown in Fig. 18 contains 240 000 vertices. The drift due to error integration is evident at the loop closure, where the facades of the buildings are repeated [see Fig. 18(b)]. The main reason behind the high drift in this dataset is the decreased precision in feature localization due to the low quality of the images: The camera uses a high compression ratio MPEG2 codec, which results in loss of details in images.

The visual vocabulary was generated and the images were indexed during the scene reconstruction. The final vocabulary contains 7182 words. The matrix shown in Fig. 19, resulting after the binarization of the image similarity matrix, clearly shows a loop closure corresponding to the first and last frames of the sequence. The situation is exemplified in Fig. 20, where frames I_{960} and I_{45} depict the same region of the scene.

After the loop-closure detection (see Section IV-C), the vertices corresponding to the same scene points were merged, and Bundle Adjustment [46] was applied on the whole model, correcting the drift. The result is shown in Fig. 21.

In order to further illustrate the performance of OVV, in the remainder of this section, we present a test that was carried out in order to test the capacity of OVV indexing to be extended to



Fig. 17. *Outdoor Experiment—Input images.* Sample images from the dataset, showing some of the typical challenges such as occlusions, sun flickering, etc. In addition, the partial overlap between the first and last image can be clearly observed.

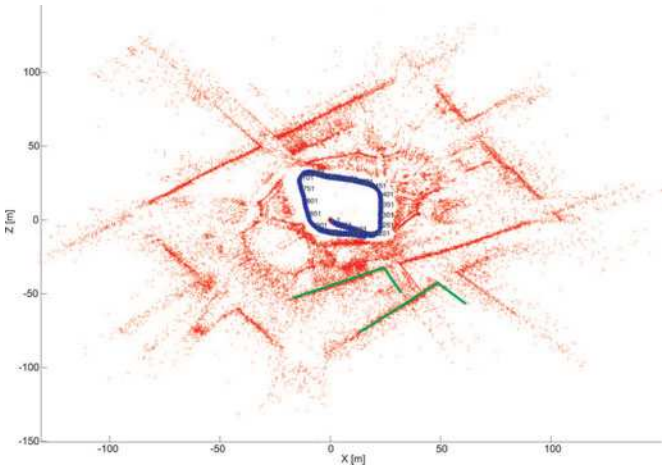


Fig. 18. *Outdoor Experiment—3-D model and camera trajectory.* Top view of the model (red) along with camera trajectory (blue). The drift at the loop closure is marked in green (repeated edges at the bottom marked in green).

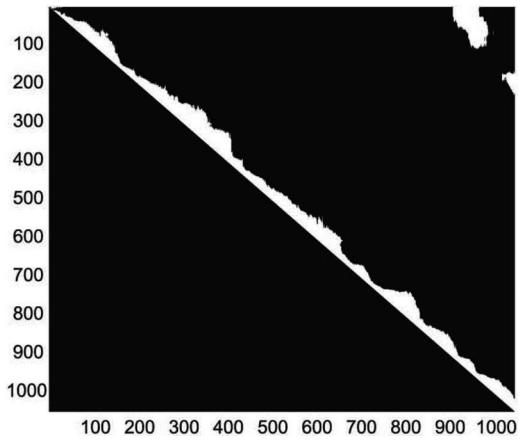


Fig. 19. *Outdoor Experiment—Loop-closure matrix.* The highlighted region in the upper-right corner of the matrix corresponds to an overlap between the first and last frames of the sequence.



Fig. 20. *Outdoor Experiment—Loop-closure detection.* Example of image pair corresponding to the loop closure.

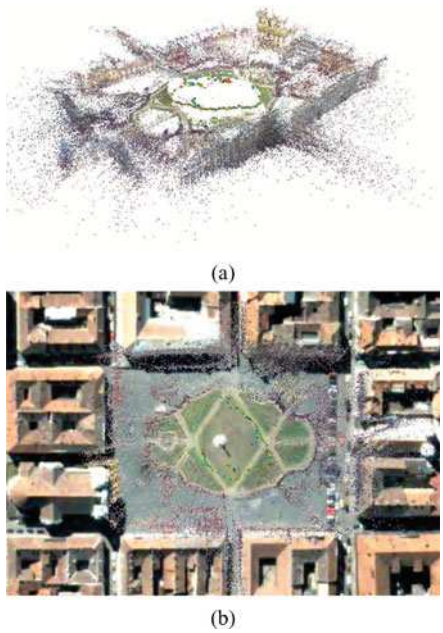


Fig. 21. *Outdoor Experiment—3-D model and camera trajectory after BA.* (a) View of the 3-D model using colored vertices and the camera trajectory. (b) Top view of the 3-D model aligned with an aerial view of Unirii Square from Google Earth—the reconstruction fits accurately with the photo.

other images of the same location. For this, we selected a set of photos from *Google Images* [47] depicting Unirii Square, taken at different times of the day and from various viewpoints. Each photo was then indexed and the most visually similar image from the original dataset was extracted. The aim is to test how well we can identify photos taken under different conditions. Fig. 22 illustrates the results. The majority of photos were correctly associated ($\simeq 90\%$). Generally, the cases in which OVV did not correctly identify the location were the result of 1) extreme zooming, where the query pictures show details of the buildings not modeled in the vocabulary due to the limited resolution of the original dataset; 2) severe obstructions that block most of the visual content modeled in the vocabulary; and 3) extreme lighting changes—pictures taken early evening or by night, where most of the visual details are lost due to low contrast. Moreover, in the last row in Fig. 22, we illustrate an example of poor localization due to high dynamic range processing of the query image.

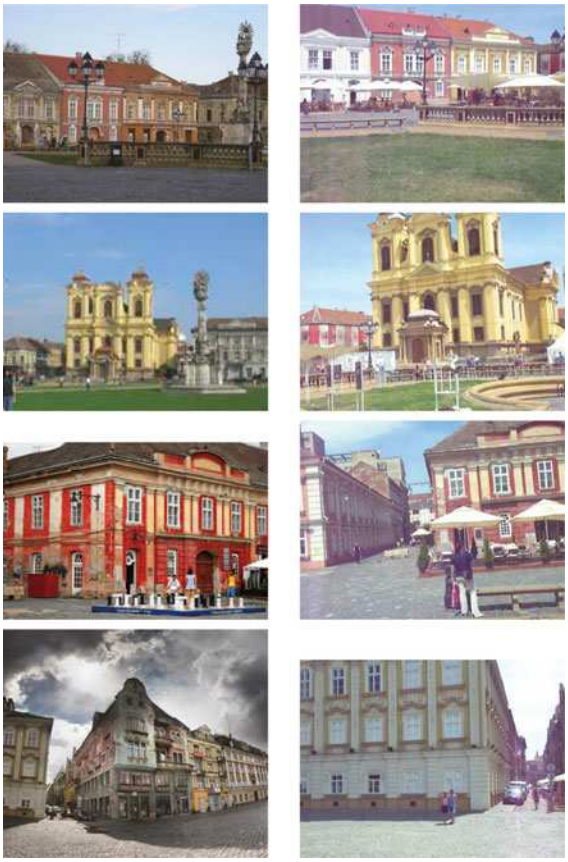


Fig. 22. *Outdoor Experiment—Location identification.* Google Images photos used as query images (left column) and the most visually similar image from the original dataset (right column). The last row shows an example of poor location identification, due to the postprocessing of the query photo.

V. Conclusion

We have developed a new visual BoW method for loop-closure detection, which is oriented toward online navigation and mapping. The method uses a novel incremental vocabulary building process. As the vocabulary is being constantly updated to include new visual information, we propose a novel incremental image-indexing process in conjunction with a tree-based feature-labeling method, which increases the stability of feature-cluster associations at different vocabulary stages.

The proposed method requires no *a priori* knowledge of the environment, as the visual vocabularies are built online, during robot navigation. In addition, while most BoW require the user to set parameters, such as the number of words in the vocabulary, which are generally data dependent, we show in [44] that the default values of parameters used by OVV yield optimum results, regardless of the type of environment, size of the robot trajectory, etc.

In this paper, we have presented a series of experiments, representing various types of environments and a comparison with a state-of-the-art visual SLAM algorithm. We show that using the proposed clustering technique, we obtain more accurate loop-closure detection, even with a smaller vocabulary size, than other SLAM algorithms. This is due to a novel clustering criteria, which takes into account the global distribution of the data, resulting in more compact and discriminant visual words.

In addition, we avoid fully reindexing the images as content of the vocabulary changes, using an incremental image-indexing method. Experimental results show that this approach allows us to highly reduce the computational times with only a small loss in precision.

The scope of this paper is oriented toward the capacity of OVV to detect loop-closure situations. However, the accuracy of the estimation will significantly increase by using OVV in conjunction with geometrical consistency checks.

In the near future, we aim to fully implement the proposed algorithm using a more efficient programming language, such as C++, decreasing the computational loads related to vocabulary building and image indexing. Moreover, we are currently analyzing the possibility to use GPU-based parallel processing in order to further reduce the computational times of OVV.

References

[1] C. Wahlgren and T. Duckett, "Topological mapping for mobile robots using omnidirectional vision," in *Proc. Swedish Workshop Auton. Robot.*, 2005, pp. 22–25.

[2] T. Goedeme, T. Tuytelaars, and L. V. Gool, "Visual topological map building in self-similar environments," in *Proc. Int. Conf. Informat. Control, Autom. Robot.*, 2006, pp. 3–9.

[3] H. Zhang, "BoRF: Loop-closure detection with scale invariant visual features," in *Proc. IEEE Int. Conf. Robot. Automation*, 2011, pp. 3125–3130.

[4] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 90–110, 2004.

[5] P. Lamon, I. Nourbakhsh, B. Jensen, and R. Siegwart, "Deriving and matching image fingerprint sequences for mobile robot localization," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2001, vol. 2, pp. 1609–1614.

[6] B. J. A. Kroese, N. A. Vlassis, R. Bunschoten, and Y. Motomura, "A probabilistic model for appearance-based robot localization," *Image Vis. Comput.*, vol. 6, pp. 381–391, 2001.

[7] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin, "Context-based vision system for place and object recognition," in *Proc. Int. Conf. Comput. Vis.*, 2003, vol. 1, pp. 273–280.

[8] J. Sivic, "Efficient visual search of images and videos" Ph.D. dissertation, Univ. Oxford, Oxford, U.K., 2006.

[9] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman, "Total recall: Automatic query expansion with a generative feature model for object retrieval," in *Proc. Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.

[10] J. Sivic and A. Zisserman, "Efficient visual search of videos cast as text retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 4, pp. 591–606, Apr. 2009.

[11] T. Botterill, S. Mills, and R. Green, "Speeded-up bag-of-words algorithm for robot localisation through scene recognition," in *Proc. Image Vis. Comput.*, Oct. 2008, pp. 1–6.

[12] 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. Vis. Pattern Recognit.*, 2007, pp. 1575–1589.

[13] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2006, pp. 2161–2168.

[14] G. Schindler, M. Brown, and R. Szeliski, "City-scale location recognition," in *Proc. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–7.

[15] D. Galvez-Lopez and J. Tardos, "Real-time loop detection with bags of binary words," in *Proc. Intell. Robots Syst.*, 2011, pp. 51–58.

[16] K. Konolige, J. Bowman, J. Chen, P. Mihelich, M. Calonder, V. Lepetit, and P. Fua, "View-based maps," *Int. J. Robot. Res.*, vol. 29, no. 8, pp. 941–957, 2010.

[17] T. Yeh, J. Lee, and T. Darrell, "Adaptive vocabulary forests br dynamic indexing and category learning," in *Proc. Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.

[18] D. Filliat, "A visual bag of words method for interactive qualitative localization and mapping," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2007, pp. 3921–3926.

[19] A. Angeli, S. Doncieux, J. A. Meyer, and D. Filliat, "Incremental vision-based topological SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2008, pp. 1031–1036.
[20] A. Angeli, S. Doncieux, J. A. Meyer, and D. Filliat, "Real-time visual loop-closure detection," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2008, pp. 1842–1847.
[21] M. Cummins and P. Newman, "Probabilistic appearance based navigation and loop closing," in *Proc. IEEE Int. Conf. Robot. Autom.*, Rome, Italy, Apr. 2007, pp. 2042–2048.
[22] M. Cummins and P. Newman, "FAB-MAP: Probabilistic localization and mapping in the space of appearance," *Int. J. Robot. Res.*, vol. 27, no. 6, pp. 647–665, 2008.
[23] M. Cummins and P. Newman, "Highly scalable appearance-only slam fab-map 2.0," in *Proc. Robot. Sci. Syst.*, 2009, pp. 1–8.
[24] R. Paul and P. Newman, "FAB-MAP 3D: Topological mapping with spatial and visual appearance," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2010, pp. 2649–2656.
[25] A. Glover, W. Maddern, M. Milford, and G. Wyeth, "FAB-MAP + RatSLAM: Appearance-based SLAM for multiple times of day," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 3507–3512.
[26] T. Nicosevici and R. Garcia, "Online visual vocabularies for robot navigation and mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2009, pp. 205–212.
[27] T. Nicosevici and R. Garcia, "Online robust 3D mapping using structure from motion cues," in *Proc. MTS/IEEE Oceans Conf.*, 2008, pp. 1–7.
[28] P. M. Kelly, "An algorithm for merging hyperellipsoidal clusters," Los Alamos Natl. Lab., Los Alamos, NM, Tech. Rep. LA-UR-94-3306, 1994.
[29] E. F. Krause, *Taxicab Geometry*. New York: Dover, 2004.
[30] I. S. Gradshteyn and I. M. Ryzhik, *Tables of Integrals, Series, and Products*, 7th ed. New York: Academic, 2007.
[31] P. C. Mahalanobis, "On the generalised distance in statistics," in *Proc. Nat. Inst. Sci. India*, 1936, vol. 2, pp. 49–55.
[32] R. C. Fisher, "The use of multiple measurements in Taxonomic problems," *Ann. Eugen.*, vol. 7, pp. 179–188, 1936.
[33] G. J. McLachlan, *Discriminant Analysis and Statistical Pattern Recognition*. New York: Wiley-Interscience, 2004.
[34] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Wu, "An optimal algorithm for approximate nearest neighbor searching," *Assoc. Comput. Mach.*, vol. 45, pp. 891–923, 1998.
[35] R. O. Duda, P. E. Harta, and D. H. Stork, *Pattern Classification*, 2nd ed. New York: Wiley–IEEE, 2000.
[36] D. D. Lewis, "Naive (bayes) at forty: The independence assumption in information retrieval," in *Proc. Eur. Conf. Mach. Learn.*, 1998, pp. 4–15.
[37] G. Csurka, C. Bray, C. Dance, and L. Fan, "Visual categorization with bags of keypoints," in *Proc. Eur. Conf. Comput. Vis.*, 2004, pp. 1–22.
[38] A. Opelt, A. Fussenegger, and P. Auer, "Weak hypotheses and boosting for generic object detection and recognition," *Lecture Notes Comput. Sci.*, vol. 3022, pp. 71–84, 2004.
[39] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: An in-depth study," INRIA, Grenoble, France, Tech. Rep. 5737, 2005.
[40] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. New York: ACM, 1999.
[41] H. Bay, T. Tuytelaars, and L. J. V. Gool, "SURF: Speeded up robust features," in *Proc. Eur. Conf. Comput. Vis.*, Graz, Austria, May 2006, pp. 404–417.
[42] J. Matas, O. Chum, M. Urba, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *Proc. Brit. Mach. Vis. Conf.*, 2002, pp. 384–396.
[43] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool, "A comparison of affine region detectors," *Int. J. Comput. Vis.*, vol. 65, no. 1/2, pp. 43–72, 2005.
[44] T. Nicosevici, "Efficient 3D scene modeling and mosaicing," Ph.D. dissertation, Comput. Vis. Robot. Group—Univ. Girona, Girona, Spain, 2009.
[45] T. Nicosevici, N. Gracias, S. Negahdaripour, and R. Garcia, "Efficient 3D modeling and mosaicing," *J. Field Robot.*, vol. 26, no. 10, pp. 759–788, 2009.
[46] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—A modern synthesis," in *Proc. Int. Conf. Comput. Vis.*, Corfu, Greece, Sep.1999, vol. I, pp. 298–372.
[47] "Google Images," (2009). [Online]. Available: http://images.google.com/



Tudor Nicosevici received the B.Sc. degree in telecommunications from the Polytechnic University of Timisoara, Timisoara, Romania, in 2001 and the M.Sc. degree in 2004 and the Ph.D. degree in 2009 from the University of Girona, Girona, Spain, in 3-D modeling and mosaicing.

He was a Postdoctoral Researcher with VICOROB, University of Girona in 2009. Since 2010, he has been an R&D Engineer with Coronis Computing, Girona. His main research interests include composite imaging and mosaicing techniques, 3-D reconstruction techniques, lighting correction and compensation in image analysis, and high-level characterization and classification of scenes.



Rafael Garcia (M'02) received the M.Sc. degree in computer science from the Autonoma University of Barcelona, Barcelona, Spain, in 1994 and the Ph.D. degree in computer engineering from the University of Girona, Girona, Spain, in 2001.

He is currently the Director of the Computer Vision and Robotics Group, University of Girona. He is the author of more than 120 articles in peer-reviewed journals and scientific conferences. His research activity is mainly focused on underwater robotics in topics such as large-scale seafloor mosaicing, sensor fusion, 3-D reconstruction, semantic representation of video imagery, robot navigation, and mapping.