The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

On-line Visual Vocabularies for Robot Navigation and Mapping

Tudor Nicosevici and Rafael Garcia

Abstract— **Detecting already-visited regions in vision-based navigation and mapping helps reduce drift and position uncertainties. Inspired from content-based image retrieval, an efficient approach is the use of *visual vocabularies* for measuring similarities between images. In this way, images corresponding to the same scene region can be associated. The state of the art proposals that address this topic suffer from two main drawbacks: (i) they require heavy user intervention, generally involving trial and error tasks for training and parameter tuning and (ii) they are suitable for batch processing only, where all the data is readily available before data processing.**

We propose a novel method for visual vocabulary navigation and mapping that overcomes these shortcomings. First, the vocabularies are built and updated online, during robot navigation, in order to efficiently represent the visual information present in the scene. Also, the vocabulary building process does not require any user intervention.

I. Introduction

Global positioning from local observations represents a topic with multiple applications in vision-based navigation. 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 for positioning uncertainty reduction, positioning based on previously created maps (kidnapped robot problem), position recovery after occlusions or failures, etc.

In computer vision, these applications became possible with the introduction of affine invariant features [20][17][4]. They allow visual tracking under various geometric transformations: rotation, scale and (to some extent) affine distortions. Furthermore, by using feature descriptors as quantifications of image patches, the latter can be tracked with little a priori knowledge of their position.

Nevertheless, when dealing with vast scenes, visual navigation systems have to deal with thousands of features for loop closure or position recovery. Matching features over such an amount of data using standard methods would be too confusing and computationally expensive. In order to deal with data at this scale efficiently, the solution is to represent visual information at a higher level of abstraction, using visual vocabularies [24], where each image is described as set of visual word occurrences. Visual words represent generalized image patch representations, obtained by clustering (grouping) together similar feature descriptors.

Some proposals of visual vocabulary-based robot navigation have been reported in the literature [8][9][25][1]. Inspired from image database retrieval and object recognition methodologies [24][23][26], they use sets of pre-computed visual vocabularies for online image indexing. Hence, the building of visual vocabularies involves a series of steps prior to navigation. First, visual data is gathered from the area where the navigation will take place. Then visual features are extracted and the visual vocabulary is generated. Unfortunately, this is an inefficient approach since it requires laborious preparations and involves strong a priori knowledge of the navigation area.

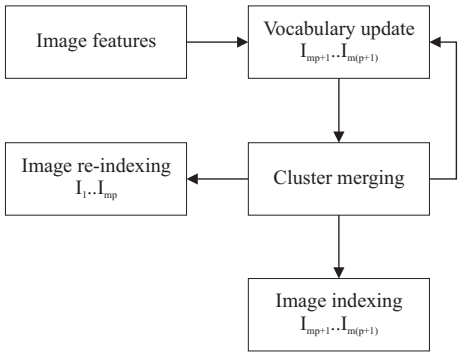

Fig. 1. **Online visual vocabulary and image indexing.** Every m frames, the vocabulary is updated with new visual features extracted from the last m frames. The complete set of features in the vocabulary is then merged until convergence. The obtained vocabulary is used to index the last m images. Also,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

We propose a novel framework for incremental visual vocabulary building. It requires no user intervention and no a priori information about the environment. The system creates a reduced vocabulary as soon as visual information becomes available during navigation. The vocabulary gets updated in order to correctly model the visual information present in the scene. The vocabulary is built using a novel clustering method that takes into account the global distribution of visual data, increasing its efficiency. Also, we present a new method for feature-cluster association and image indexing, suited for online applications.

The proposed method is implemented on a Structure From Motion (SFM) algorithm oriented towards underwater navigation and 3D mapping [22]. Here, the visual vocabularies are used to quantify visual similarities between frames, allowing the detection of loop-closures.

This paper is structured in the following way: first, a detailed presentation of the novel vocabulary building method is presented in Section II, along with online image indexing in Section III. The fourth part discusses some of the experiments that we have conducted, aimed towards testing various aspects of the online visual vocabulary. The paper concludes with some remarks and a proposal for further work.

II. Visual Vocabulary

All the state of the art visual vocabulary-based navigation proposals assume an initial off-line stage [9][1]. This stage involves pre-acquiring visual features from the scene. These features are then used to build the visual vocabulary by means of some clustering method. Typical off-line 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. Finding the adequate parameters for an optimum vocabulary is a tedious task which generally involves a trial and error approach. For example, 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 discriminant.

We propose a novel visual vocabulary building technique that is both scalable (thus suitable for on-line applications) and automatic (see Figure 1). For this, we propose the use of a modified version of agglomerative clustering [5]. Agglomerative 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, the elementary clusters are generated from visual tracking of scene points [21], with each elementary cluster corresponding to one feature track. The visual vocabulary is generated by incrementally merging these clusters. The vocabulary building process can be summarized in two steps (see Figure 2):

- **Vocabulary initialization step.** The vocabulary is initialized with the elementary clusters corresponding to the first m images. The clusters are gradually merged until convergence (the merging criterion is discussed in detail in the last part of Section II).
- **Vocabulary update step.** As the robot moves, more visual information of the scene becomes available that 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 n-dimensional 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[1], we define them through:

[1]Feature tracking provides multiple (noisy) observations of a scene point.

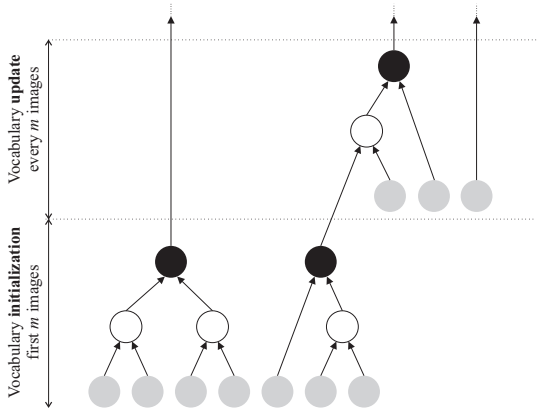

Fig. 2. **Iterative Visual Vocabularies.** In the initialization step (bottom part) the vocabulary is populated with elementary clusters (marked in gray), extracted from the first m images. These clusters are merged until convergence. The final clusters of the initialization step are marked in black.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

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

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

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.

Each cluster merging involves the joining of two clusters (see Figure 2). The parameters of the newly generated cluster are obtained directly from the merging clusters, without the need of recomputing 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 [14]:

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

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

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

C. Cluster merging

Generally, clustering algorithms use some similarity measurement to decide which data should be grouped into clusters. Often, similarity measurements are represented by distances in the n-dimensional data space, including: Euclidean distance, Manhattan distance [15], Chebyshev norm [12], Mahalanobis distance [18], 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. The proposed method, based on Fisher's linear discriminant [11] [19], 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* given by:

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

$$S_W = \frac{1}{N}\sum n_k R_k \tag{7}$$

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 [2].
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 two clusters are merged and Sb, Sw are updated accordingly[2].

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

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

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

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.

D. Convergence criterion

The two steps shown in Section II-C are repeated, gradually merging clusters, until no more merges are possible (that would increase the value of the objective function Q). In this way, the method offers a natural convergence criterion, eliminating the need of any user parameters.

[2]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.

E. Vocabulary update

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[3]:

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

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

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

Taking into account the changes in 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 using:

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

F. Linear Discriminant Analysis (LDA)

Using the cluster information contained in the visual vocabulary, we aim to find a data transformation that would maximize cluster separability and would allow us to reduce the dimensionality of the data, thus increasing the speed of both vocabulary building and image indexing. For this, we consider maximizing the following LDA objective function [11][19][10]:

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

where w is a vector determining 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.

III. Image indexing

Inspired from text document indexing [16], *visual bag of words* techniques use *visual vocabularies* to represent the images by associating the features present in the image with the clusters (visual words) in the vocabulary [7] [23] [26]. The result is a histogram representing the number of

[3]The number of elements in an elementary cluster corresponds to the number of frames in the feature track.

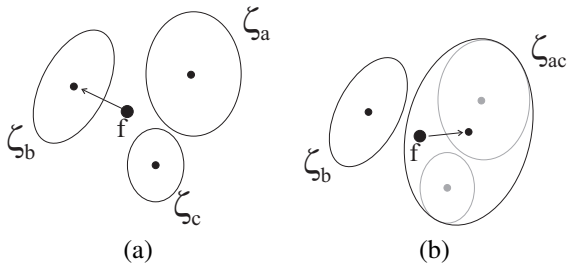

Fig. 3. **Feature-cluster association – classical approach.** In (a) the feature f is associated with cluster ζ_b, using feature-to-cluster centroid distance. After the vocabulary update, clusters ζ_a and ζ_c are merged (b). The centroid of the newly obtained cluster ζ_{ac} is now closer to f, determining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

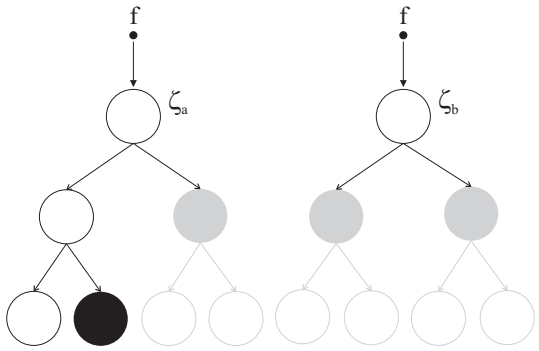

Fig. 4. **Top-down feature-cluster association.** The trees are visited by comparing each node with the feature. If a node is too dissimilar to the feature (marked in light grey), the rest of the tree corresponding to the node is not visited. The feature is associated with ζ_a due to the highest similarity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

occurrences of each visual word in the image. The similarity between images is calculated by comparing these histograms.

Generally, there are two aspects that define the efficiency of a visual vocabulary: (i) similar image features should be associated with the same clusters (*repetitiveness*) and (ii) dissimilar image features have to be associated with different clusters (*discriminative power*). We aim to maximize these two properties by using the objective function presented in Section II-C.

In the context of the on-line vocabulary, we 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.

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-C). This approach is suitable for image indexing in the case of static vocabularies[4] [24].

As we use an on-line approach for vocabulary building, such a feature association method would not be stable. In Figure 3a, 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} (Figure 3b). As the feature f is now closer to the centroid of the new cluster ζ_{ac}, it would be associated to it. In this case, feature f would be associated with different clusters before and after the vocabulary update.

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 leafs of the trees correspond to the elementary clusters (see Figure 2).

During the feature-cluster association, the trees are visited top-down, calculating the similarity (Euclidean distance) between each feature and the tree nodes. 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:

$$D(f, \zeta_k) \leq \tau D_m \tag{15}$$

where $D(f, \zeta_k)$ is 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 ($\tau \geq 1$).

The selected trees are visited in parallel (see Figure 4). For efficiency purposes, a stopping criterion similar to eq. 15 is used, hence avoiding visiting branches that contain nodes that are not close to f. The feature is finally associated to the visual word corresponding to the most similar leaf.

B. Image re-indexing

During the update process, the configuration of the vocabulary changes. Consequently, the similarity between images indexed at different update stages cannot be computed. Also, indexing the images after each vocabulary update is not a viable solution due to its large computational cost.

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

$$\widetilde{H}_I^p = {}^p T_{p-1} H_I^{p-1} \tag{16}$$

where H_I^{p-1} is the indexing of image I at vocabulary update stage $p-1$ and $\widetilde{H}_I^p$ 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

[4]A static vocabulary represents a vocabulary that is calculated before the image indexing stage and does not change throughout it.

contain new visual information. In this case, it is very unlikely that any feature from any image before the update would has been associated to them. Therefore, the bins $\widetilde{H}_I^k$ 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 to 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, ${}^pT_{p-1}$ 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 $[h_1\ h_2\ h_3]^{\mathrm{T}}$ corresponding to the visual vocabulary containing (ζ_1, ζ_2, ζ_3); during the vocabulary update, clusters ζ_1, ζ_2 merge into ζ_{12} and a new cluster ζ_4 is added. In this case, the transformation ${}^pT_{p-1}$ becomes:

$$\begin{bmatrix} h_{12} \\ h_3 \\ h_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} \tag{17}$$

C. Image Similarity

The visual resemblance between images is quantified by measuring the similarity of their corresponding histograms[5]. As the histograms are represented by vectors containing the occurrences of the visual words, we calculate their similarity using the normalized scalar product (cosine of the angle between vectors) [24]:

$$s_{rq} = \frac{H_r^{\mathrm{T}} H_q}{\|H_r\|_2 \cdot \|H_q\|_2} \tag{18}$$

where s_{rq} is the similarity score between images I_r and I_q, H_r and H_q are the histograms of the images; $\|H\|_2 = \sqrt{H^{\mathrm{T}}H}$ is the L_2 norm of vector H.

In eq. 18, the similarity score is highly influenced by histogram elements corresponding to visual words with high occurrence. Generally, these 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 *term frequency-inverse document frequency* [3]:

$$\overline{h}_k = \frac{n_{ki}}{o_i} \log \frac{mp}{O_k} \tag{19}$$

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 mp is the total number of indexed images.

D. Cross-over detection

During on-line navigation and mapping, increased values of s_{rq} between the current image and any previous one indicate a high probability of the two images representing the same scene region (i.e. loop closing). This information can be used for both introducing new constraints in the mapping model and reducing the navigation-related uncertainties.

Noise, low contrast and especially motion blur may sometimes decrease the efficiency of image indexing leading to false positives when detecting cross-overs. Assuming a smooth camera motion, there must be a certain degree of overlap between neighboring frames in the image sequence. In other words, if an image I_q has a high degree of visual similarity with some other image I_r, the neighbors of I_r must also be (at least partially) visually similar to I_q.

Seeing the similarity between image I_q and all the images in the sequence as a time-dependent measurement, we employ individual compatibility test [6] to reject the false positives.

IV. Experimental Results

The proposed online visual vocabulary technique is implemented on top of the Structure From Motion algorithm presented in [21]. Given a sequence of images of a generic scene, the SFM algorithm extracts and tracks visual features, recovering the up-to-scale 3D geometry of the scene. The visual tracker in the SFM algorithm uses state of the art feature extractors and descriptors, including: SIFT [17], SURF [4], MSER [4], Harris [13], etc.

In this section, we present two sets of experiments, aimed at testing and validating the main stages of the online vocabulary and image indexing.

The first experiment was carried out in the laboratory, using a relatively flat scene that contains books, boxes and magazines. The scene composition was chosen to be visually complex, combining uniform (low texture) regions, natural scenes, geometric figures and abstract drawings.

The test sequence consists of 215 images of 640×480 pixels, acquired using a Canon G9 compact camera (see Figure 5 for some snapshots of the sequence). The images contain a certain amount of motion blur and defocusing, allowing us to test the robustness of the visual vocabulary. The camera is moved while in a down-looking orientation, describing a loop trajectory with a partial overlap between the first and the last images. Figure 6 illustrates the resulting scene model and camera trajectory, after applying SFM on the image sequence. Hereafter, we discuss the vocabulary building process, image indexing and cross-over detection.

The detection and extraction of features was carried out using SURF, yielding ∼37,000 tracks corresponding to the 3D vertices. Each image feature is represented using a 64-element normalized vector, describing the Haar wavelet responses in the neighborhood of the feature [4]. The vocabulary was built incrementally, during the scene reconstruction process.

The vocabulary was initialized using the visual information extracted from the first 20 images and updated every 10 frames (with 21 updates in total). Figure 7 illustrates the evolution of the vocabulary size as it gets updated. Towards the end of the sequence, the increase rate lowers, stabilizing

[5]In this paper, the term "histogram" of image I refers to a vector embodying the number of occurrences of each visual word in I.

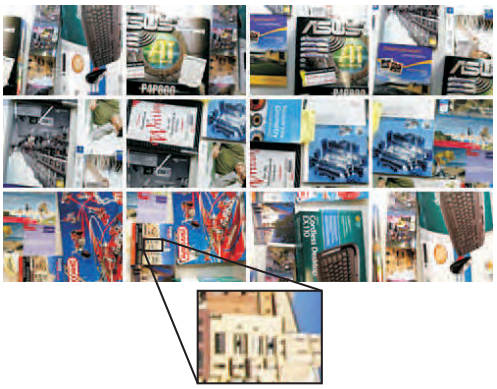

Fig. 5. **Laboratory experiment: Input image sequence.** Sample images from the input sequence. The first and the last image have a partial overlap. The blow-up shows the motion blur and defocusing.

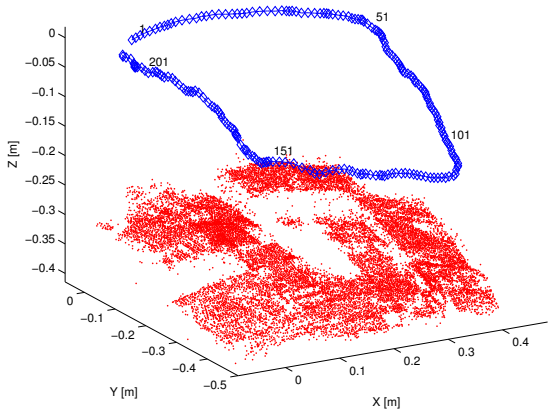

Fig. 6. **Laboratory experiment: The result of the SFM algorithm.** The scene model contains ∼37,000 vertices (marked in green). The camera describes loop a trajectory (marked in blue) with an overlap between the first and last images.

at $\sim 4,000$ words. We test the quality of the data clustering in the vocabulary and the efficiency of the proposed indexing method, using a direct data association experiment. For each image feature, we associate an elementary cluster that corresponds to the smallest Euclidean distance in the feature space. The image features are then "sent down" the indexing trees. If the image features end up at the leaf corresponding to the associated elementary cluster, it is considered a hit and a miss otherwise. We conducted the test using various LDA dimensionality reduction stages and different τ values (see eq. 15). Due to space limitations, we omit a detailed presentation of the results. It should be mentioned, however, that reducing the feature dimension from 64 to 24 and using $\tau = 1.4$, we obtain a miss rate of 0.96%. These parameters highly reduce the computational times for both vocabulary building and frame indexing while maintaining the properties of the vocabulary (see Section III). Figure 8 illustrates the execution time evolution for both vocabulary

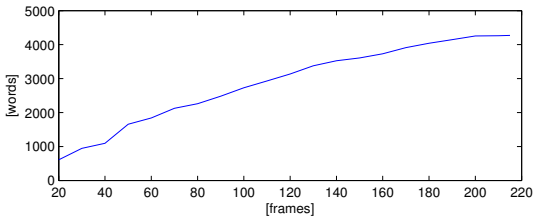

Fig. 7. **Laboratory experiment: Vocabulary size evolution.** The vocabulary is updated every 10 frames. The final size of the vocabulary is 4271 words.

building and frame indexing[6]. For the whole sequence, the average vocabulary update time was 1.36 sec./update and the average frame indexing time was 0.23 sec./frame.

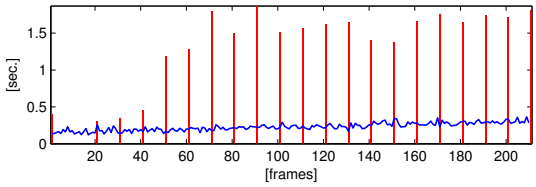

Fig. 8. **Laboratory experiment: Execution times.** The vocabulary building time (red bars) and the frame indexing time (blue line) evolution vs. the number of frames.

The last part of the laboratory experiment consisted in the detection of the loop closure. After image indexing, the similarity matrix shown in Figure 9 clearly illustrates a high degree of visual resemblance between the first images and the last images of the sequence (upper-right corner).

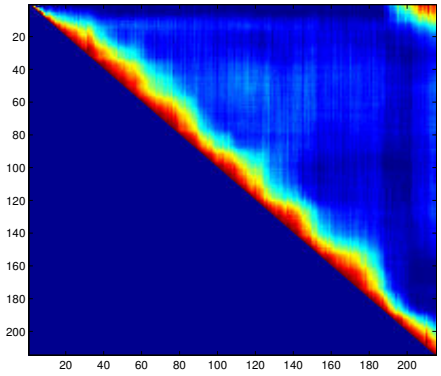

Fig. 9. **Laboratory experiment: Image similarity matrix.** High values close to the main diagonal correspond to the similarity of the images with their close neighbors. The bright region in the upper-right corner of the matrix denotes an overlap between frames in the beginning and the end of the sequence.

Figure 10 illustrates the similarity score between I_{215} and all the images in the sequence. The peak at image I_1 indicates a high visual similarity between frames I_1 and I_{215} of 0.8.

[6]The algorithms were mainly implemented in Matlab, with certain routines implemented in C. All the experiments presented in this paper were carried out on a Intel Pentium Core Duo 2.13GHz machine with 4GB of RAM, running Windows Vista.

In order to see how well the similarity score represents the actual overlap, we calculated the projective homography between the two images shown in Figure 11. Using the homography, we obtained an overlapping ratio of 0.82. This shows that the similarity score accurately represents the overlapping between frames.

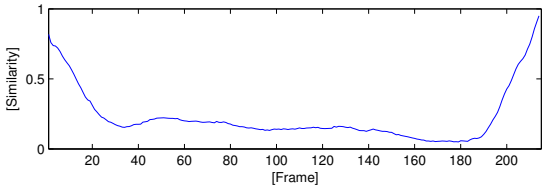

Fig. 10. **Laboratory experiment: Image similarity for query image I_{215}.** The plot shows the similarity between frame I_{215} and all the previous frames. The peak on the far right of the plot correspond to the time-adjacent frames. The peak corresponding to I_1 indicates an overlap.

Fig. 11. **Laboratory experiment: loop detection.** Query frame I_{215} (top) and frame I_1 (bottom) correspond to the loop closure in the camera trajectory.

The second experiment presented here is aimed at testing the efficiency of the online visual vocabulary method in describing natural, unstructured environments for underwater robot navigation and mapping. For this, we have chosen an underwater dataset, acquired using a Remotely Operated Vehicle (ROV) near Bahamas by the University of Miami. The sequence is comprised by 235 frames of 720×530 pixels. Figure 12 illustrates the result of the 3D reconstruction and ROV trajectory estimation. In this experiment, we carried out the online vocabulary building and image indexing using the same parameters as in the previous experiment. The resulting image similarity matrix in Figure 13 successfully points out the cross-overs in the robot trajectory. An exemplification of this is provided in Figure 14, where a query for frame

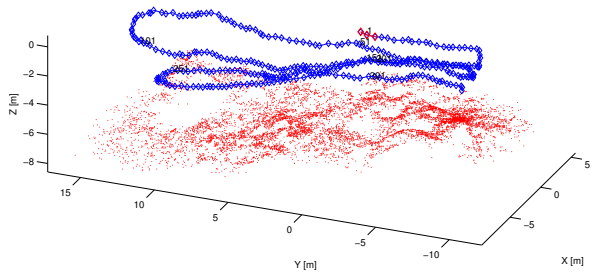

Fig. 12. **Underwater experiment: The result of the SFM algorithm.** The scene model contains 62,000 vertices. The trajectory of the ROV has several cross-overs.

I_{204} shows two peaks at frames I_{52} and I_{155}, with similarity scores of 0.73 and 0.75 respectively. The estimated overlap ratio between I_{204} and frames I_{52} and I_{155} is 0.78 and 0.8 respectively, showing that the similarity scores closely represent the overlap between images. Figure 15 clearly illustrates that the three frames correspond to the same region of the scene.

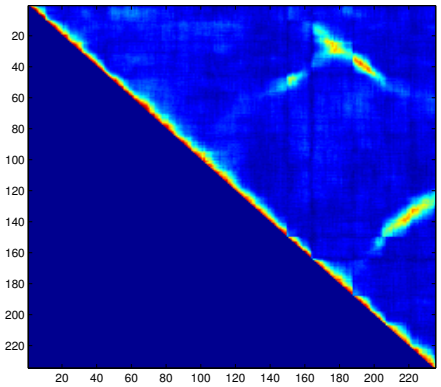

Fig. 13. **Underwater experiment: Image similarity matrix.** The bright regions off the main diagonal correspond to multiple cross-overs.

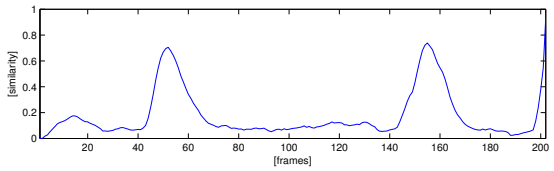

Fig. 14. **Underwater experiment: Image similarity for query image I_{204}.** The plot shows the similarity between frame I_{204} and all the previous frames. The two peaks corresponding to frames I_52 and I_155 indicate that all three frames correspond to the same region of the scene.

V. Conclusions

This paper presents a novel visual vocabulary building method, oriented towards on-line robot navigation. The

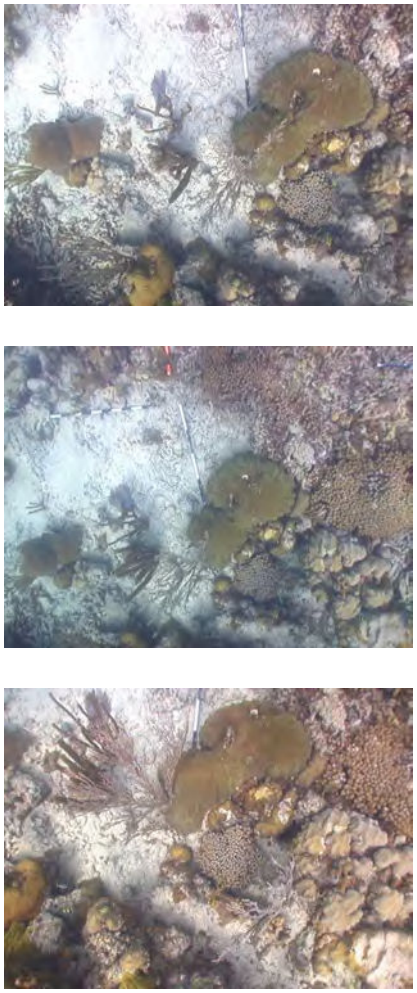

Fig. 15. **Underwater experiment: loop detection.** Query frame I_{204} (top) and frames I_{52} (middle) and I_{155} (bottom) were successfully determined as corresponding to the same region of the scene, defining a loop closure.

proposed method has contributions at different levels. The visual vocabulary in built incrementally, eliminating the need of the off-line training stage and increasing the flexibility of visual-navigation. The feature clustering method uses a global approach, ensuring a more efficient data distribution. Also, a new hierarchical technique increases the feature-cluster association robustness, in the context of a constantly changing vocabulary. We also propose a novel image re-indexing method that eliminates the necessity of repeated indexing as the vocabulary changes.

Consequently, we discuss some experimental results that show the applicability of the method in cross-over detection for robot navigation and mapping.

Ongoing and future work includes testing a method for decreasing the memory usage of the vocabulary (for example by eliminated small clusters at the bottom of the hierarchy), that would allow testing the method for large scale navigation.

VI. ACKNOWLEDGMENTS

This work has been partially supported by the Spanish Ministry of Science and Innovation (grant CTM2007-64751), and the EU Marie Curie FREESUBNet.

REFERENCES

[1] A. Angeli, S. Doncieux, J. A. Meyer, and D. Filliat. Incremental vision-based topological SLAM. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1031–1036, 2008.

[2] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Wu. An optimal algorithm for approximate nearest neighbor searching. *Association for Computing Machinery*, 45:891–923, 1998.

[3] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press,, 1999.

[4] H. Bay, T. Tuytelaars, and L. J. Van Gool. SURF: Speeded Up Robust Features. In *European Conference on Computer Vision*, pages 404–417, Graz, Austria, May 2006.

[5] P. Berkhin. Survey of clustering data mining techniques. Technical report, 2002.

[6] J. Castellanos and J. Tardos. *Mobile Robot Localization and Map Building: A Multisensor Fusion Approach*. Kluwer Academic Publishers, 1999.

[7] G. Csurka, C. Bray, C. Dance, , and L. Fan. Visual categorization with bags of keypoints. In *European Conference on Computer Vision*, pages 1–22, 2004.

[8] M. Cummins and P. Newman. Probabilistic appearance based navigation and loop closing. In *IEEE International Conference on Robotics and Automation*, Rome, April 2007.

[9] M. Cummins and P. Newman. FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. *International Journal of Robotics Research*, 27(6):647–665, 2008.

[10] R. O. Duda, P. E. Harta, and D. H. Stork. *Pattern Classification*. Wiley–IEEE, 2nd edition, 2000.

[11] R. C. Fisher. The use of multiple measurements in Taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.

[12] I. S. Gradshteyn and I. M. Ryzhik. *Tables of Integrals, Series, and Products*. Academic Press, 7th edition, 2007.

[13] C. G. Harris and M. J. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, pages 147–151, Manchester, U.K., 1988.

[14] Patrick M. Kelly. An algorithm for merging hyperellipsoidal clusters. Technical report, 1994.

[15] E. F. Krause. *Taxicab Geometry*. Dover Publications, 2004.

[16] David D. Lewis. Naive (bayes) at forty: The independence assumption in information retrieval. pages 4–15. Springer Verlag, 1998.

[17] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):90–110, 2004.

[18] P. C. Mahalanobis. On the generalised distance in statistics. In *National Institute of Sciences of India*, volume 2, pages 49–55, 1936.

[19] G. J. McLachlan. *Discriminant Analysis and Statistical Pattern Recognition*. Wiley Interscience, 2004.

[20] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1/2):43–72, 2005.

[21] T. Nicosevici and R. Garcia. Online Robust 3D Mapping Using Structure from Motion Cues. In *MTS/IEEE OCEANS Conference*, pages 1–7, 2008.

[22] T. Nicosevici, R. Garcia, S. Negahdaripour, M. Kudzinava, and J. Ferrer. Identification of Suitable Interest Points Using Geometric and Photometric Cues in Motion Video for Efficient 3-D Environmental Modeling. In *IEEE International Conference on Robotics and Automation*, pages 4969–4974, 2007.

[23] A. Opelt, A. Fussenegger, and P. Auer. Weak hypotheses and boosting for generic object detection and recognition. *Lecture Notes in Computer Science*, 3022:71–84, 2004.

[24] J. Sivic. *Efficient visual search of images and videos*. PhD thesis, University of Oxford, 2006.

[25] J. Wang, R. Cipolla, and H. Zha. Vision-based Global Localization Using a Visual Vocabulary. In *IEEE International Conference on Robotics and Automation*, pages 4230–4235, 2005.

[26] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classifcation of texture and object categories: An in-depth study. Technical report, INRIA, 2005.