Chapter 4
Online Loop Detection

This chapter proposes a novel loop closure detection framework for visual based navigation and mapping. The proposed approach eliminates the training stage and reduces the user interaction process while increasing both the accuracy and robustness of the loop closure detection.

4.1 Introduction

Vision-based navigation is essentially a *dead reckoning* process. During navigation and map building, vision systems estimate the camera pose relative to either previous poses or an environment map, while they build the map from observations relative to camera poses. All estimations are prone to aliasing, noise, image distortions and numerical errors (see Section 1.3), leading to inaccuracies in both pose and map inferences. While generally small, these inaccuracies build up in time, leading to significant errors over large camera trajectories.

These errors can be reduced by taking advantage of the additional information resulting from *cross-overs*. Cross-overs (or loop-closures) are situations when a camera revisits a region of the scene during a visual survey. If correctly detected, these situations can be exploited in order to establish new constraints, allowing both camera pose and map errors to be decreased (see Figure 4.1), either using offline approaches such as BA [19, 102, 108, 154, 172] or online approaches employing gaussian filters such as the popular Kalman Filter [18, 41, 52, 145] or non-parametric methods such as those using particle filters [99, 112], etc. In this context, the main open issue is the correct and efficient detection of loop closures.

Loop closure detection is an inherently complex problem due to the amount of data that needs to be analysed. As typical image feature extractors yield thousands of features per image, after just a few hundred frames, the resulting map contains tens to hundreds of thousands of features. A brute force loop closure detection, where the current visual observations are compared to the

T. Nicosevici & R. Garcia: *Efficient 3D Scene Modeling and Mosaicing*, STAR 87, pp. 87–125.
DOI: 10.1007/978-3-642-36418-1_4

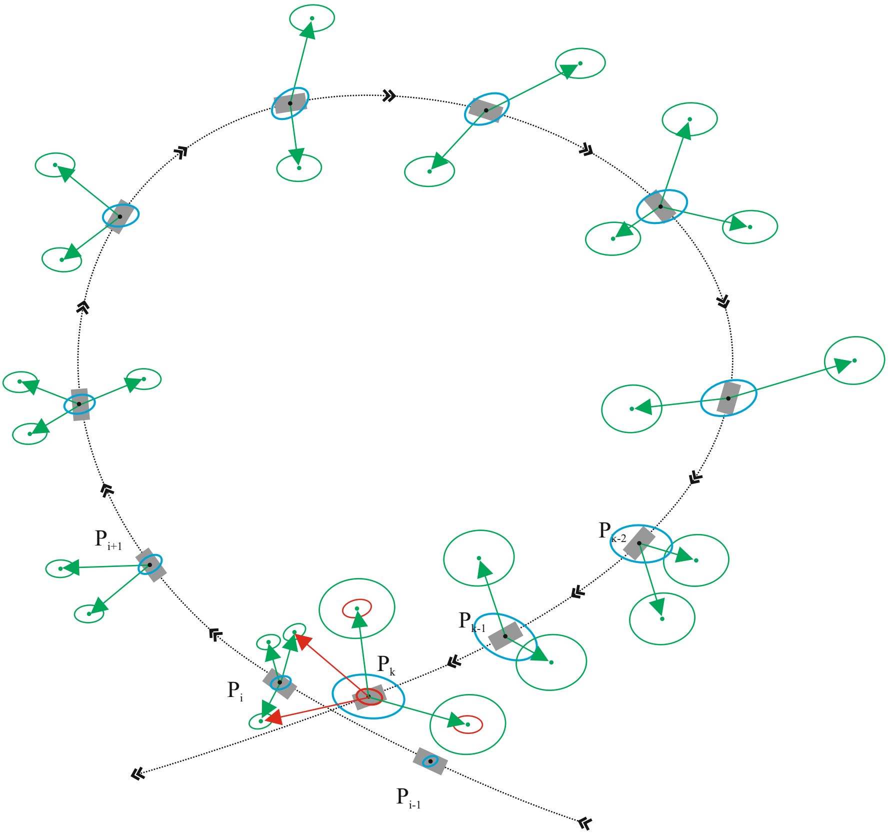


Fig. 4.1 Loop closure detection. As the the camera moves, there is an increasing uncertainty related to both the camera pose and the environment map. At instant t_k, the camera revisits a region of the scene previously visited at instant t_i. If the visual observations between instants t_k and t_i can be associated, the resulting information can be used not only to reduce the pose and map uncertainties at instant t_k, but it also can be propagated, reducing the uncertainties at prior instants.

entire map, would be much too computationally expensive, especially for online applications.

As an alternative, the complexity of the loop closure problem can be reduced by narrowing the search to the vicinity of the current camera pose. This is a widely used approach, mainly in the Simultaneous Localization and Mapping (SLAM) community, where the vision system is modeled as a sensor with a known uncertainty. During navigation, an uncertainty is associated to each vehicle pose and the loop closures are detected by matching current observations with the region of the map corresponding to the current uncertainty space [31, 32, 76, 136]. However, an accurate estimation of the vehicle uncertainty is a complex problem and is generally affected by linearization approximations. To counterbalance this shortcoming, assuring the detection of the cross-over, current observations may be compared with a region of the map corresponding to a higher covariance than the estimated one [80, 106]. Doing so becomes computationally expensive, especially over large trajectory loops, where the covariance of the camera is high. Moreover, the noise model used for covariance estimation does not account for inaccuracies resulting from obstruction, temporary motion blur, sensor failures, etc. These situations lead to poor vehicle pose estimation, not reflected in the uncertainty estimation, in which case the loop closure may not be detected.

In [56, 176, 183], the authors propose a loop-closing detection method that computes the visual similarity using features. During navigation, they extract key points from each image (e.g. SIFT [96]). 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 more robust and computationally efficient alternative is to represent entire images as observations rather than individual image features. In this context, cross-overs are detected on the basis of image similarity, drastically decreasing the amount of data that needs to be processed. The reduced computational cost related to such approaches enable brute force cross-over detection, even for large camera trajectories. This allows correct detection of trajectory loops, independent of camera pose and covariance estimation accuracy.

Initial proposals on image similarity cross-over detection use image representations based on a single global descriptor, embodying visual content such as color or texture[13, 86, 88, 143, 170]. Such global descriptors are sensitive to camera view-point and illumination changes, decreasing the robustness of the cross-over detection.

The emergence of modern feature extractors and descriptors (see Section 2.1.3) has led to the development of new appearance-based cross-over detection techniques that represent visual content in terms of local image descriptors [1, 2, 25, 26, 177]. Inspired from advances in the fields of object recognition and content-based image retrieval [133, 162, 185], recent examples of such approaches describe images using BoW (see Figure 4.2). BoW image representation employs two stages: (*i*) in the training stage, sets of visual features are grouped or clustered together to generate *visual vocabularies* - collections of generalized visual features or *visual words*; (ii) 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 for detecting visual similarities between images, allowing efficient cross-over detection even in presence of illumination and camera perspective changes, partial occlusions, etc.

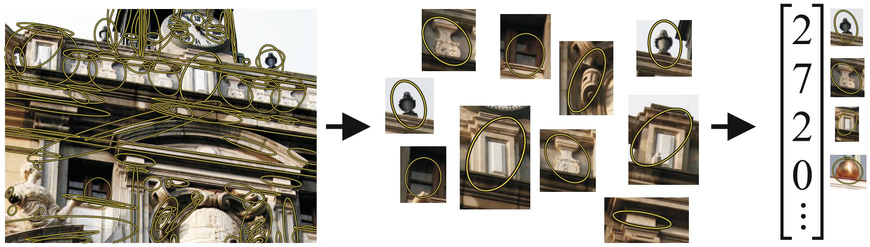


Fig. 4.2 BoW image representation. Images are represented by histograms of generalized visual features.

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 [21, 162, 163], where the user is required to specify the number of visual words in the vocabulary (for a detailed comparison of different clustering strategies, please refer to [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 data sets [137], [131].

In [156] Schlinder *et al.* propose the use of kd-trees to build a visual vocabulary as proposed by Nister and Stewenius in [131]. The vocabulary is then used for SLAM at the level of a city with good results. Galvez *et al.* [45] propose the use of a vocabulary based on binary features for fast image matching.

Konolige *et al.* [84] 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 above-mentioned 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 [182]. 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 series of authors in the SLAM community have proposed alternatives to address this problem. Notably, Filliat [37] and Angeli *et al.* [1, 2], 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, Cummins *et al.* [25–27] and later Paul *et al.* [135] and Glover *et al.* [55] propose a large scale loop detection probabilistic framework 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 [183] proposes a workaround to the off-line 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 for building Online Visual Vocabulary (OVV) [127]. The proposed approach is aimed at increasing the efficiency and accuracy of loop-detection in the context of on-line 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 which takes into account the entire distribution of the clusters, increasing the efficiency of the resulting vocabularies. Also, we present a novel method for feature-cluster association and image indexing, suited for incremental vocabularies.

The remaining of the chapter 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 3D reconstruction and mapping for land and underwater environments. The chapter concludes with some remarks and proposal for further work.

4.2 Visual Vocabulary

State of the art visual vocabulary-based loop-closure algorithms assume an initial training stage. This stage involves pre-acquiring visual features, which are then used to build the visual vocabulary by means of some clustering method. 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 chapter we propose a novel incremental visual vocabulary building technique that is both scalable (thus suitable for online applications) and automatic (see Figure 4.3). 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.).

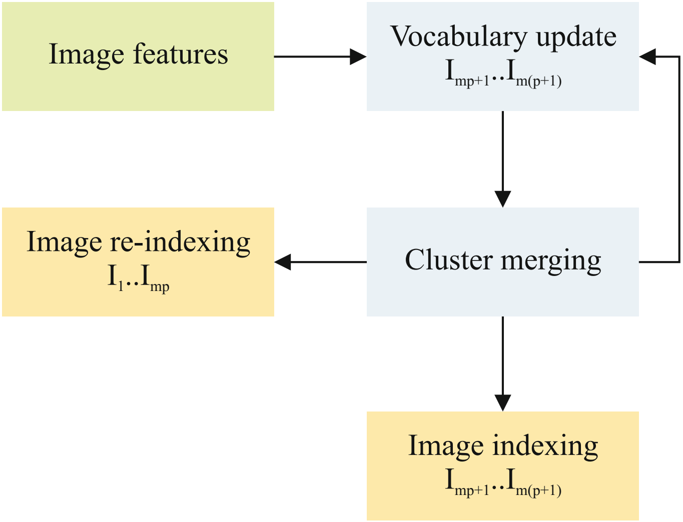


Fig. 4.3 Flowchart of OVV and image indexing. Every m frames, the vocabulary is updated with new visual features extracted from the last m frames. The complete set of features in the vocabulary is then merged until convergence. The obtained vocabulary is used to index the last m images. Also, the previously indexed frames are re-indexed, to reflect the changes in the vocabulary.

4.2.1 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 [126, 129]. While not required by OVV, this step allows us to pre–select the number of visual features used in building the vocabulary, decreasing the computational costs.

The visual vocabulary is built by incrementally merging these clusters. The building process can be summarized in two steps (see Figure 4.3):

- **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 4.2.4).
- **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.

4.2.2 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:

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

$$R_k = \frac{\sum\limits_{i=1}^{n} (f_k^i - C_k)(f_k^i - C_k)^{\mathrm{T}}}{n-1}$$

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.

4.2.3 Cluster Updating

Each cluster merging involves the joining of two clusters (see Figure 4.4). The parameters of the newly generated cluster are obtained directly from the merged 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 [82]:

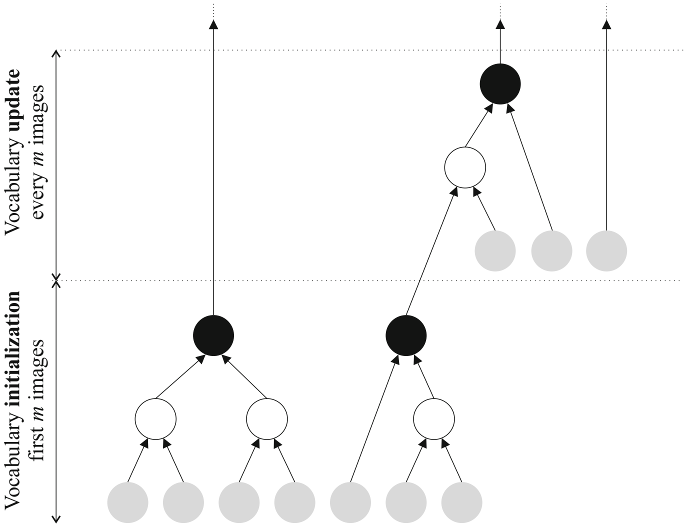


Fig. 4.4 Iterative visual vocabularies. In the initialization step (bottom part) the vocabulary is populated with elementary clusters (marked in gray), extracted from the first m images. These clusters are merged until convergence. The final clusters of the initialization step are marked in black. In 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.

$$\begin{aligned}
C_{ab} &= \frac{n_a C_a + n_b C_b}{n_a + n_b} \\
R_{ab} &= \frac{n_a - 1}{n_a + n_b - 1} R_a + \frac{n_b - 1}{n_a + n_b - 1} R_b \\
&\quad + \frac{n_b \cdot n_a}{(n_a + n_b)(n_a + n_b - 1)} \\
&\quad \cdot \left[(C_a - C_b)(C_a - C_b)^{\mathrm{T}} \right]
\end{aligned}$$

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

4.2.4 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 [85], Chebyshev norm [63], Mahalanobis distance [103], 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: (i) *repetitiveness*: similar image features should be associated to the same cluster and (ii) *discriminative power*: dissimilar image features have to be associated to different clusters.

The proposed method, based on Fisher's linear discriminant [39] [107], clusters the data in order to maximize the following objective function:

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

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:

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

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

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 [4].
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, Sw are updated accordingly[1].

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:

[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.

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

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

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.

4.2.5 Convergence Criterion

The two steps shown in Section 4.2.4 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 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.

4.2.6 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:

$$
S'_W = \frac{N S_W + R_e}{N + n_e}
$$

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:

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

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

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

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

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

4.2.7 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 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 Linear Discriminant Analysis (LDA) objective function [39][107][29]:

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

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.

4.2.8 Vocabulary Update Criterion

In Section 4.2.1, 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:

$$|f_l - C_k| \leq 3\sigma_k$$

where σ_k is the standard deviation of cluster ζ_k. In Eq. 4.2.8, 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.

4.3 Image Indexing

Inspired from text document indexing [89], 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 [24, 133, 185]. The result is a histogram representing 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 presence of illumination and perspective changes. We partially achieve this by maximizing the repetitiveness and discriminative power of the vocabulary (see Section 4.2.4). 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 below.

4.3.1 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 4.2.4). 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 [162].

Since we use an online approach for vocabulary building, such a feature association method would not be stable. In Figure 4.5a, 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 4.5b). 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. 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.

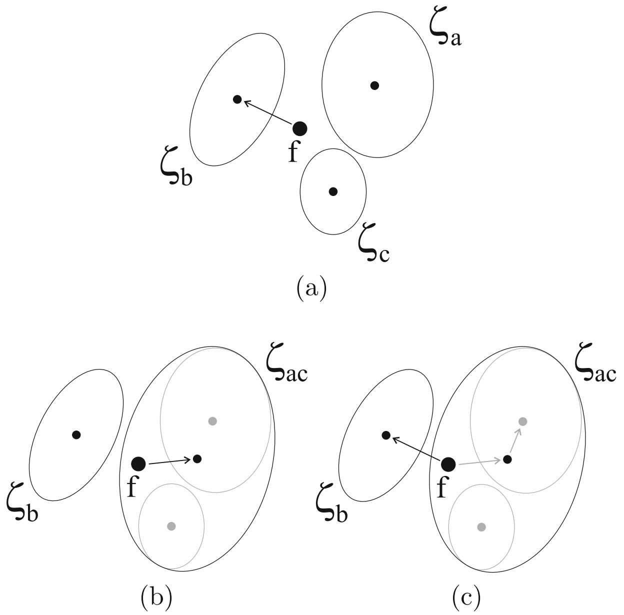


Fig. 4.5 Feature-cluster association. In (a) the feature f is associated with cluster ζ_b, using feature-to-cluster centroid distance. After the vocabulary update, clusters ζ_a and ζ_c are merged. The centroid of the newly obtained cluster ζ_{ac} is now closer to f. Using a classical approach, feature f would be associated with ζ_{ac} (b). Using hierarchical trees, feature f is correctly associated with cluster ζ_b (c).

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 4.4).

During the feature-cluster association, the trees are visited top-down, calculating the similarity (Euclidean distance) between each feature and the tree nodes (see Figure 4.5c). 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:

$$D(f, \zeta_k) \leq \tau D_m$$

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 Figure 4.6). For efficiency purposes, we use the same stopping criterion shown in Eq. 4.3.1, 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.

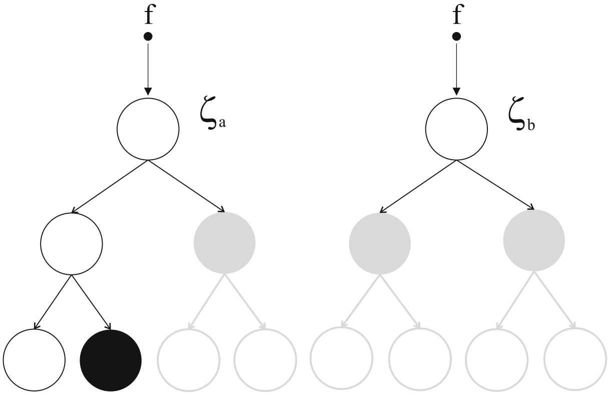


Fig. 4.6 Top-down feature-cluster association. The trees are visited by comparing each node with the feature. If a node is too dissimilar to the feature (marked in light grey), the rest of the tree corresponding to the node is not visited. The feature is associated with ζ_a due to the highest similarity between f and the leaf marked in black.

4.3.2 Image Re-indexing

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. 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 ${}^{p}\Gamma_{p-1}$ 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):

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

[2] User parameter τ provides a balance between computational efficiency and accuracy of the image indexing. As shown in Section 4.4, optimum results are obtained using a typical value of $\tau = 1.4$, which is not data dependent.

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 to 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 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, ${}^{p}\Gamma_{p-1}$ 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, ζ_2 merge into ζ_{12} and a new cluster ζ_4 is added. In this case, the transformation ${}^{p}\Gamma_{p-1}$ becomes:

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

4.3.3 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 containing the occurrences of the visual words, we calculate their similarity using the normalized scalar product (cosine of the angle between vectors) [162]:

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

where s_{rq} is the similarity score between images I_r and I_q, 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 Eq. 4.3.3, 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 [5]:

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

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.

4.3.4 Loop-Closure Detection

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.

4.3.5 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 ANN techniques [4] 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:

$$tr(R_k^i) < p \cdot tr(R_k)$$

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.

4.4 Experimental Results

This section discusses a series of experiments designed to evaluate the efficiency and accuracy of the two contributions of presented in this chapter: (i) incremental building of the vocabulary and (ii) image indexing based on hierarchical trees. The efficiency and accuracy of the online visual vocabulary algorithm is tested using a data association and a comparison with ground truth. In practice, the OVV process was implemented on top of DPR-SfM, which provides extraction and tracking of the image features used by OVV.

In the first part, we assess the influence of LDA dimension reduction s and relative threshold τ (see eq. 4.3.1) on the accuracy and computational times of OVV. 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 $s_{rq} \geq 0.45$.

The second experiment provides a detailed analysis of OVV for a large-scale loop closure problem in a mixed environment. In order to provide an objective assessment of the proposed algorithm, for this experiment we carry out a comparison between OVV and a state-of-the art visual SLAM algorithm, FAB-MAP2 [27].

In the last part of the section, we discuss series of experimental results that illustrate the application of OVV in 3D robot navigation and mapping. During navigation and mapping, the visual features extracted by DPR-SfM are used to create a 3D 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, we show the use of OVV for 3D navigation and mapping in case of two distinct scenarios: (*i*) an urban environment, and (*ii*) and underwater environment. The latter was chosen due to the additional difficulties imposed by the underwater environment: the high rate of light absorbtion 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.

It should be mentioned here that for the urban and underwater experiments, we were not able to obtain consistent results using FAB-MAP2 due to its inability to cope with high overlapping frame sequences, such as those provided by video cameras.

4.4.1 Laboratory Experiment

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 4.7 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 4.8 illustrates the resulting scene model and camera trajectory, after applying DPR-SfM on the image sequence. The detection and extraction of features was carried out using SURF, yielding ∼37,000 tracks corresponding to the

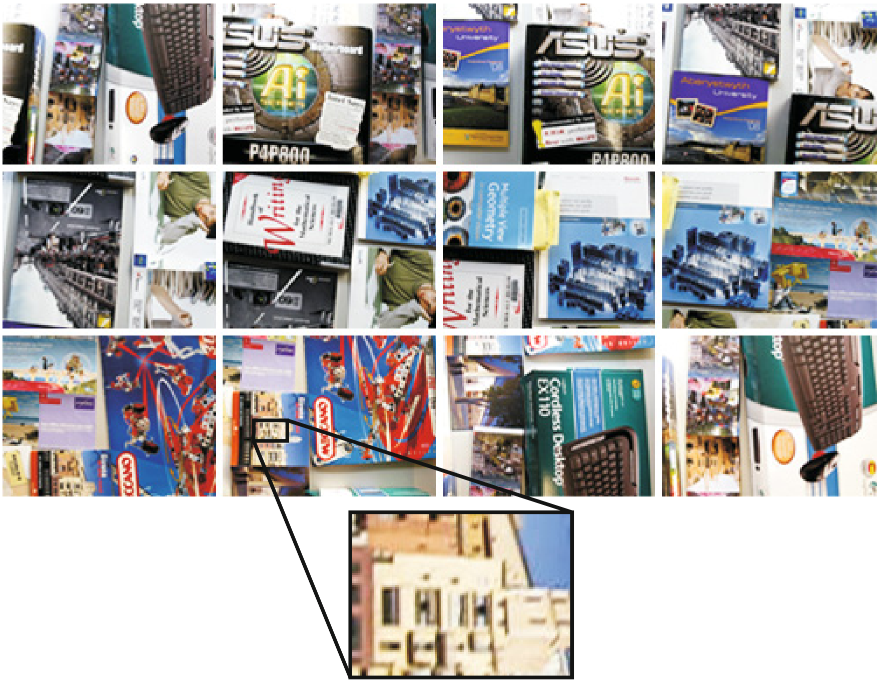


Fig. 4.7 Laboratory Experiment – Input image sequence. Sample images from the input sequence. The first and the last images have a partial overlap. The blow-up shows the motion blur and defocusing.

3D vertices. Each image feature is represented using a 64-element normalized vector as described in Section 2.1.3.

The vocabulary was initialized using the visual information extracted from the first 20 images. During sequence analysis there are 10 vocabulary updates, resulting in a final vocabulary containing 3,485 visual words. Figure 4.9 illustrates the evolution of the vocabulary. Towards the end of the sequence, the growth rate of the vocabulary decreases, as there is little new visual information contained in the last images. The instants when the vocabulary was updated can be better observed in Figure 4.10, along with the computational times of vocabulary building and frame indexing.

OVV can be adjusted using two user-set parameters. Unlike other visual vocabulary algorithms, where various parameters need to be adjusted for each dataset in order to obtain accurate results, the user parameters in OVV are data independent. The first parameter s determines the number of LDA dimensions used for feature clustering and image indexing. A lower number of dimensions decreases both the clustering and frame indexing times, while slightly decreasing the accuracy of the results. The second parameter τ

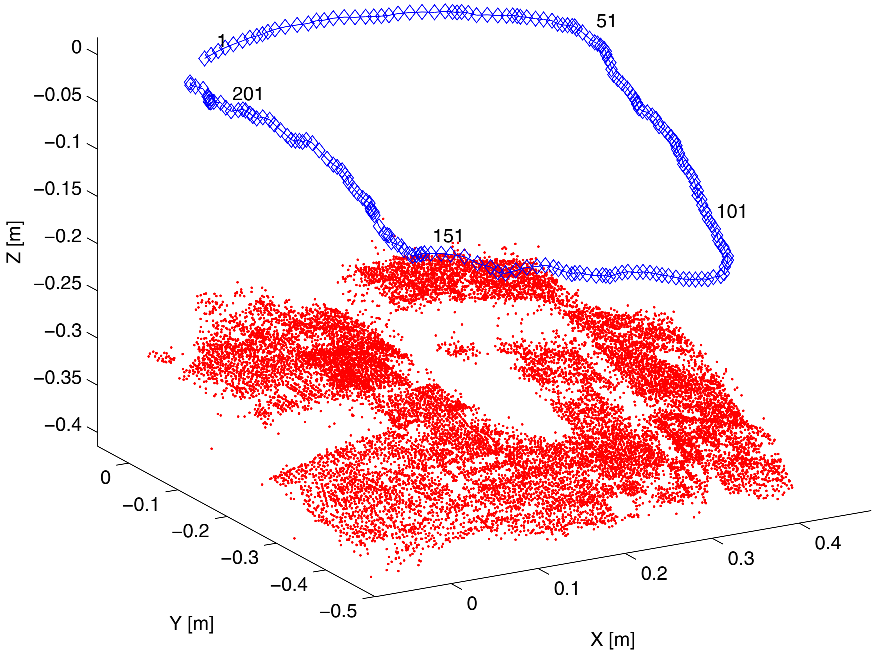


Fig. 4.8 Laboratory Experiment – 3D model and camera trajectory. The scene model contains ∼37,000 vertices (marked in green). The camera describes a loop trajectory (marked in blue) with an overlap between the first and last images.

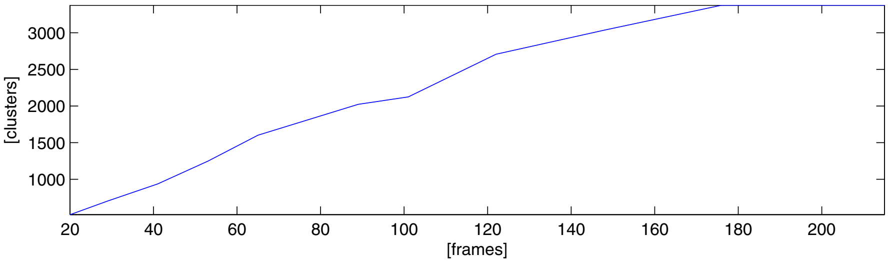


Fig. 4.9 Laboratory Experiment – Vocabulary size evolution. The vocabulary was initialized using the first 20 frames. After 10 updates, the final vocabulary contains $\simeq 3,400$ visual words.

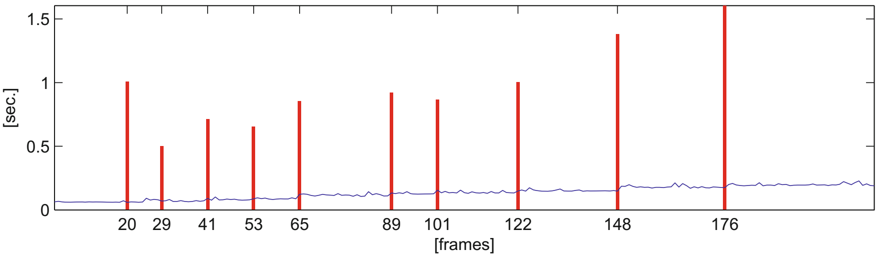


Fig. 4.10 Laboratory Experiment – Computational times. The vocabulary building time (red bars) and the frame indexing time (blue line) evolution vs. the number of frames. A total of 10 vocabulary updates took place with an average of 0.9 sec./update. The average indexing time was 0.13 sec./frame.

determines the amount of tree branches that are simultaneously visited during frame indexing. A lower value of this parameter decreases the computational time related to frame indexing, while slightly decreasing the accuracy of frame indexing.

We designed two tests that assess the efficiency of the OVV and influence of the parameters on the accuracy of the results. In the first test, we use 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. A high ratio of hits denotes a stable vocabulary and feature labeling which is crucial for accurate results, especially in the case of dynamic vocabularies used in OVV, as we show in Section 4.3.1. The second test is aimed at evaluating the accuracy of the visual similarity in representing the actual overlap between images. For this, we compare the similarity matrix (see Figure 4.11) with the overlap ground truth matrix. The overlap matrix was obtained by exhaustively calculating the projective homography between each two images from the sequence. From the homographies, we obtained the overlap ratio between all images in the sequence. We represent the accuracy of the frame similarity matrix by the average of absolute differences between the similarity and the overlap matrices.

The two tests were repeated for different values of s and τ. Table 4.1 shows the accuracy and execution time versus LDA dimensionality reduction. The results clearly show the advantages of LDA. Reducing the dimensionality of data to 24 we obtain more accurate results and greatly increased computational efficiency with respect to full 64 dimensions when no LDA is used. However, decreasing the data dimensionality further diminishes the discriminative power of the vocabulary. This increases the similarity score between

Table 4.1 Laboratory Experiment – OVV accuracy and execution times vs. LDA dim. reduction. As the number of dimensions decreases, total vocabulary building time (2*nd* column) and average frame indexing time (3*rd* column) are reduced, also decreasing the hit percentage (4*th* column) and increasing the visual similarity average error wrt. image overlap (5*th* column). The first row shows the results without using LDA.

LDA Dim. s	Vocab. Time [sec.]	Index. Time [sec./fr.]	Hits [%]	Error
no LDA	11.9	0.24	99.1	0.0714
64	10.9	0.24	99.6	0.0668
48	9.9	0.17	99.5	0.0674
32	8.6	0.13	99.3	0.0682
24	8.3	0.11	99.2	0.0695
16	6.5	0.08	98.8	0.0793
8	5.7	0.05	98.0	0.1216

Table 4.2 Laboratory Experiment – OVV accuracy and execution times vs. τ. Using a higher τ, the average frame indexing time (2*nd* column) increases as more tree branches are visited simultaneously, improving the hit percentage (3*rd* column) and decreasing the visual similarity average error with respect to image overlap (4*th* column).

τ	Index. Time [sec./fr.]	Hits [%]	Error
1.0	0.10	95.0	0.0738
1.1	0.11	97.0	0.0731
1.2	0.11	98.4	0.0715
1.3	0.12	98.9	0.0701
1.4	0.13	99.2	0.0695
1.5	0.15	99.2	0.0693

non-overlapping frames, reducing the overall accuracy of the result. Additional tests on other datasets show that $s = 24$ provides the ideal tradeoff between accuracy and computational efficiency.

Augmenting the value of τ (see Table 4.2), increases the number of tree branches that are simultaneously visited during image indexing. As expected, this results in increased accuracy at the expense of higher computational costs. Using $\tau = 1.4$ offers the ideal trade-off between indexing speed and accuracy, as using higher values increases the related computational cost with no real gain in accuracy. As in the previous case, $\tau = 1.4$ proved to be the ideal value for all the datasets we have tested.

In order to provide the reader with an objective evaluation, we compare the results obtained using OVV with an off the shelf BoW algorithm based on K-means clustering. We have chosen this approach for comparison, due to its popularity in computer vision and visual SLAM community. We set the number of words in the vocabulary to be the same as the number of words in the OVV in its final form – 3,485 words. Due to the random nature of K-means clustering, we ran the clustering algorithm 20 times and chose the vocabulary corresponding to the maximum cluster compactness. The average computational time was 8.9 sec./run. The frames were indexed using minimum Euclidean distance feature-cluster association with an average computational time of 0.3 sec./frame, resulting in an average error between the similarity matrix and frame overlap of 0.0985. This shows that, while incremental, OVV provides better accuracy than offline K-means algorithm.

The last part of the Laboratory Experiment consisted in the detection of the loop closure. For this, we build the image similarity matrix, shown in Figure 4.11. The similarity matrix illustrates a high degree of visual resemblance between the first images and the last images of the sequence (upper-right corner).

Figure 4.12 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}, corresponding to a cross-over (see Figure 4.13). The visual similarity score between the two images is 0.8, accurately representing the ground truth overlapping ratio of 0.82.

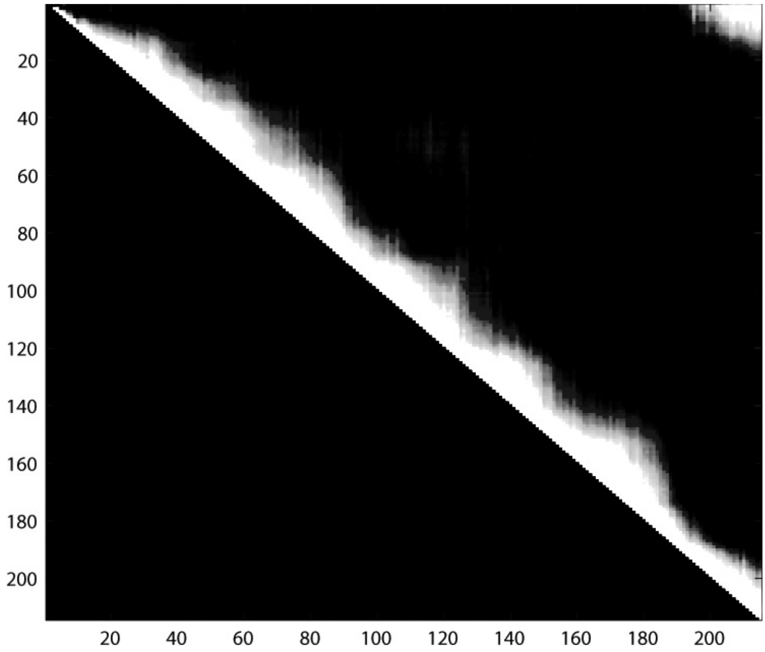


Fig. 4.11 Laboratory Experiment – Image similarity matrix. High values close to the main diagonal correspond to the similarity of the images with their close neighbors. The bright region in the upper-right corner of the matrix denotes an overlap between frames in the beginning and the end of the sequence.

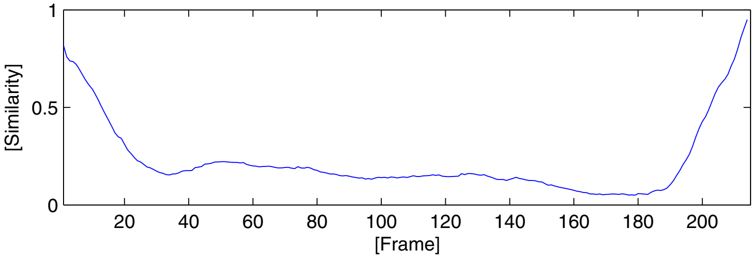


Fig. 4.12 Laboratory Experiment – Image similarity for query image I_{215}. The plot shows the similarity between frame I_{215} and all the previous frames. The peak on the far right of the plot corresponds to time-adjacent frames. The peak corresponding to I_1 indicates an overlap.

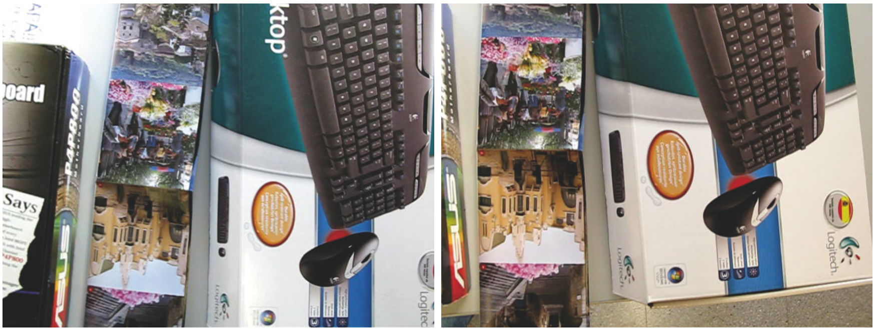


Fig. 4.13 Laboratory Experiment – Loop detection. The first (left) and last (right) images of the sequence correspond to the same region of the scene, determining a loop closure.

4.4.2 Large-Scale Mixed Environment

In this experiment, we have acquired data corresponding to a large trajectory consisting of two part: (*i*) an urban area with well structured and diversified visual characteristics and (*ii*) 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 was acquired using a setup consisting of two Canon 50D DSLR cameras equipped with 24mm Canon fixed lenses, mounted on a car (see Figure 4.14). 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 DGPS system, 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 representing both urban and natural environments. Figure 4.15 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 seconds, 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 was interpolated to match the image acquisition rate (the DGPS provides GPS fixes at a frequency of 1Hz). Any images acquired within a distance of 30m 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 due to the large total field of view of the imaging system ($\sim$ 164 degrees horizontally).

Prior to feature tracking, the images were down-sampled 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 $\sim 30K$ visual words. Figure 4.16 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).

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 Figure 4.17, 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 et al. [27]. For this purpose, we ran FAB-MAP2 in two cases: (*i*) using a generic, off-line built visual vocabulary of 40k visual words, containing mostly urban visual data, and (ii) using a generic, off-line built vocabulary containing 80k visual words, embodying data from various environments (indoor/outdoor, natural, etc.).

Figure 4.18 shows the results of the precision/recall evaluation. The goal of this analysis is two fold: (*i*) evaluate the performance of the incremental image indexing and (*ii*) compare the accuracy of OVV wrt. FAB-MAP2.

The incremental indexing was compared to the full indexing of the entire set of images using the vocabulary generated by OVV in its final form. Figure 4.18 shows that the proposed incremental method closely approximates the full indexing, with 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 3D 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, here 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, without any 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 view-point changes, partial occlusions, moving pedestrians, cars, etc. Furthermore, all the loop-closures situations were successfully detected even at early stages of the navigation, where the vocabulary contained little visual information (see Figure 4.19 for a few examples of detected loop-closure situations).

Nevertheless, there are a few cases where the erroneously matched images representing 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.

4.4.3 Underwater Experiment

This experiment is aimed at testing the efficiency of the online visual vocabulary method in describing natural, unstructured environments for underwater robot navigation and mapping, under typical challenges found in this environment. The data was acquired in Tortugas, Florida Keys using a Phantom ROV of the UoM. The 1,000-image sequence has a resolution of 720×530 pixels and depicts a region comprised mainly by rocks and sand. The sequence

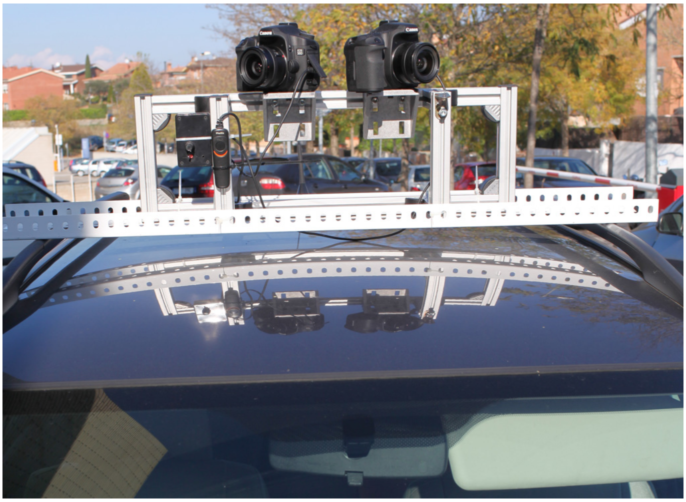


Fig. 4.14 Mixed Environment Experiment – Image acquisition setup. The dual-camera setup was mounted on a car during data acquisition.

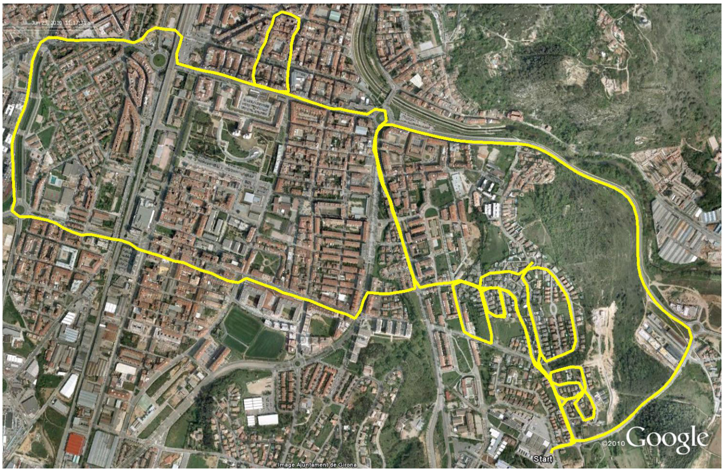


Fig. 4.15 Mixed Environment Experiment – Car trajectory during data acquisition. The 18.5 km trajectory (overlayed in yellow, as recorded by the DGPS system) was chosen to include multiple loop-closures. The starting point can be seen at the lower right corner.

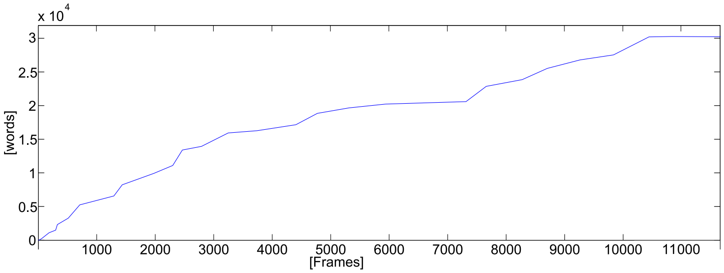


Fig. 4.16 Mixed Environment Experiment – Vocabulary size evolution. The vocabulary growth slows down at the middle of the sequence but increases at the end of the sequence as novel types of sceneries are imaged. There are 35 vocabulary updates in total.

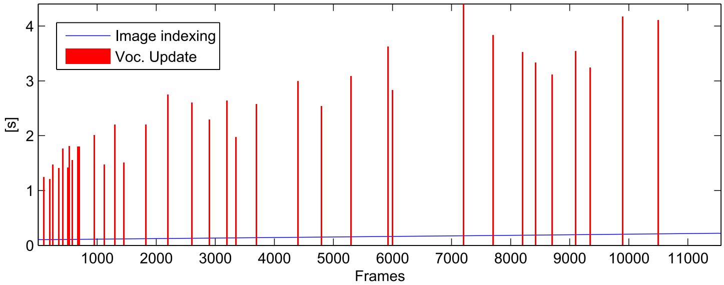


Fig. 4.17 Mixed Environment Experiment – Vocabulary building and image indexing computational times. The vocabulary update step takes an average of 1.6 seconds / update while the image indexing takes an average of 0.11 seconds / frame.

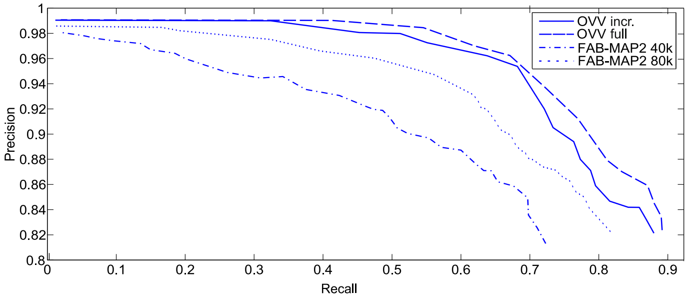


Fig. 4.18 Mixed Environment Experiment – Precision/Recall evaluation. Comparison between OVV in two cases: using the proposed incremental indexing vs. full re-indexing using the final form of the vocabulary; and FAB-MAP2 for two vocabulary cases.

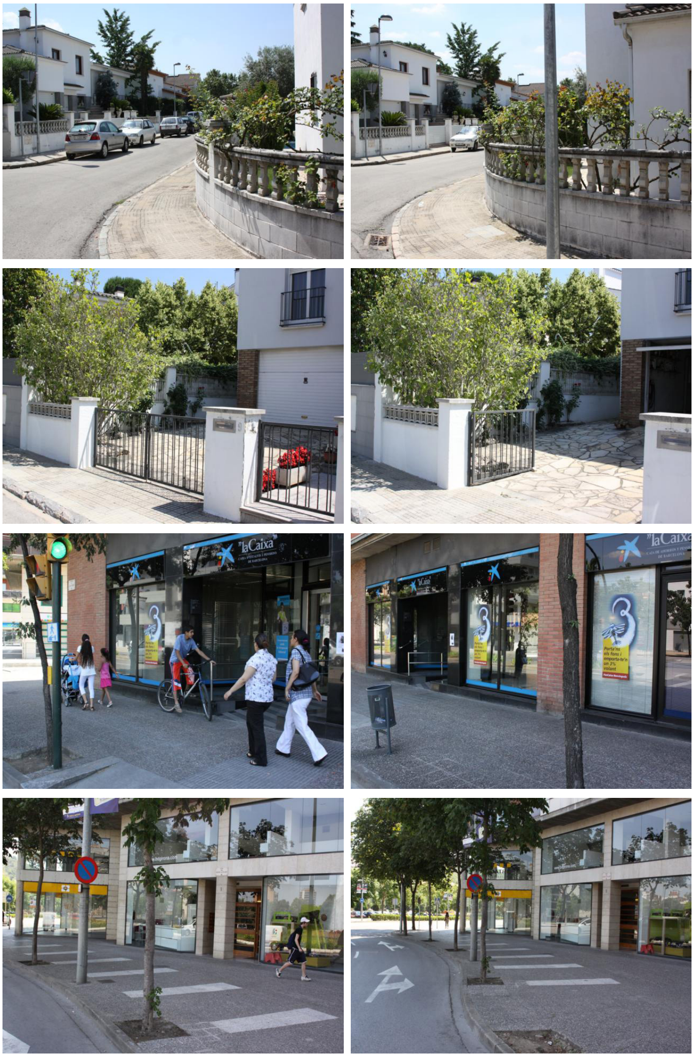


Fig. 4.19 Mixed Environment Experiment – Successfully detected loop-closure situations. Loop closures are successfully detected in the presence of camera view point changes, dynamic environments (moving pedestrians, occlusion due to the presence of cars, etc.)

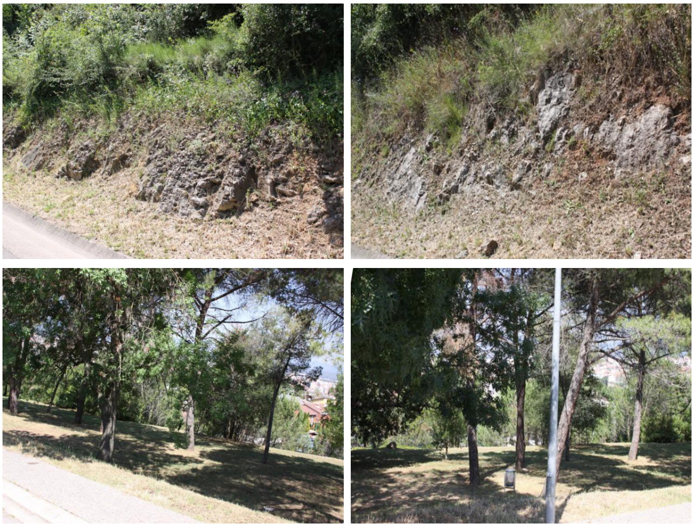


Fig. 4.20 Mixed Environment Experiment – False positives in loop closure detection. Some of the wrongly detected loop-closure situations, mostly related to repetitive patterns.

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

Figure 4.21 illustrates the estimated 3D model containing 125,850 vertices and the camera trajectory. The online vocabulary was initialized using the feature tracks in the first 20 frames. During scene reconstruction, the vocabulary went through 15 updates, containing 6,644 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 (illustrated in Figure 4.22) 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 be also observed in the image similarity matrix (see Figure 4.23a), denoted by the slightly bright

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

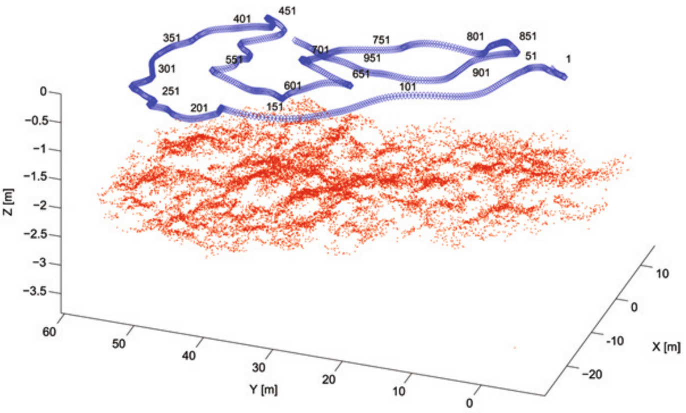


Fig. 4.21 Underwater Experiment – Estimated 3D model and camera trajectory. The scene model shown in red contains $\simeq 126,000$ vertices. The trajectory of the camera (blue) presents some partial overlaps.

background corresponding to non-overlapping 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 we have carried out. The resulting loop-closure detection matrix (see Figure 4.23b), clearly depicts areas where the robot revisits previously mapped areas.

Figure 4.24 illustrates some of the pairs of images, corresponding to loop-closures in the camera trajectory.

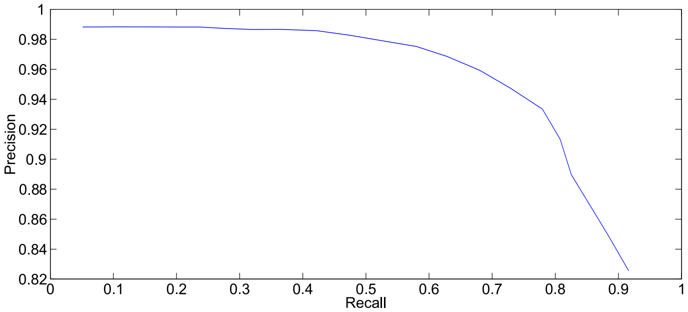
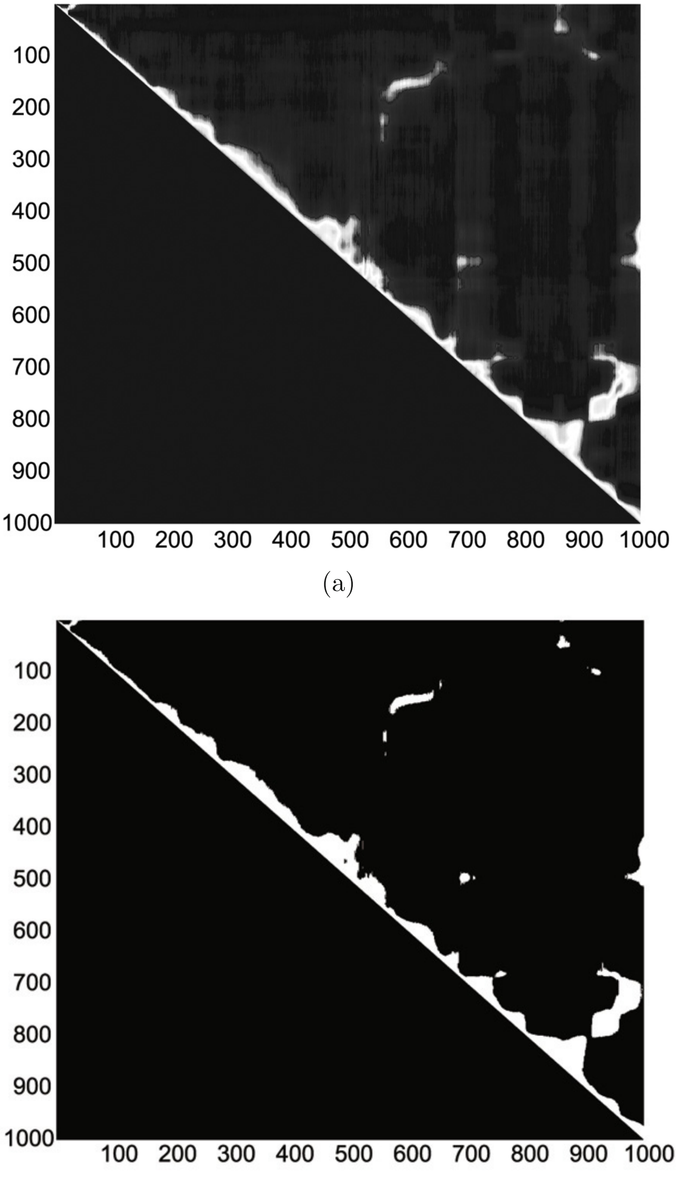


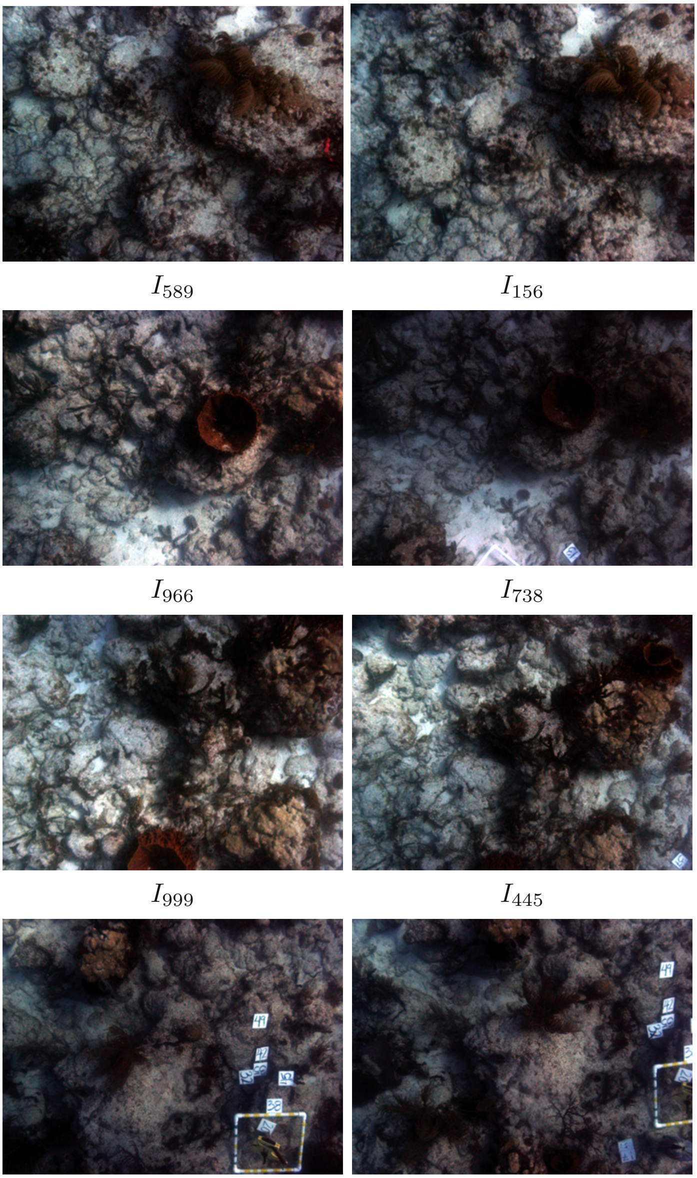
Fig. 4.22 Underwater Experiment – Precision/Recall evaluation. The maximum precision is slightly lower in this experiment mostly due to the perceptual aliasing.



(a)

(b)

Fig. 4.23 Underwater Experiment – Similarity matrix and loop-closures (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.



I_{589} I_{156}

I_{966} I_{738}

I_{999} I_{445}

I_{841} I_{45}

Fig. 4.24 Underwater Experiment – 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.

4.4.4 Coral Reef Experiment

This experiment is aimed at testing the efficiency of the OVV method in describing natural, unstructured environments for underwater robot navigation and mapping. The image sequence, acquired using a ROV near the Bahamas by the UoM, is comprised by 235 frames of 720 × 530 pixels. The surveyed scene contains a coral formation and its surroundings, combining rich texture areas (vegetation and rock formations) and uniform areas (sandy regions).

We applied DPR-SfM on the sequence using SURF features. Figure 4.25 illustrates the 3D reconstruction and the camera trajectory estimation. The resulting $\simeq 62,000$ SURF feature tracks were used to generate the vocabulary as the scene was being reconstructed. The vocabulary was initialized using the first 20 frames and updated 9 times, containing 4,343 in its final form. Analyzing the vocabulary evolution in Figure 4.26, it can be seen that the vocabulary grows fast at the beginning of the sequence. Towards the end,

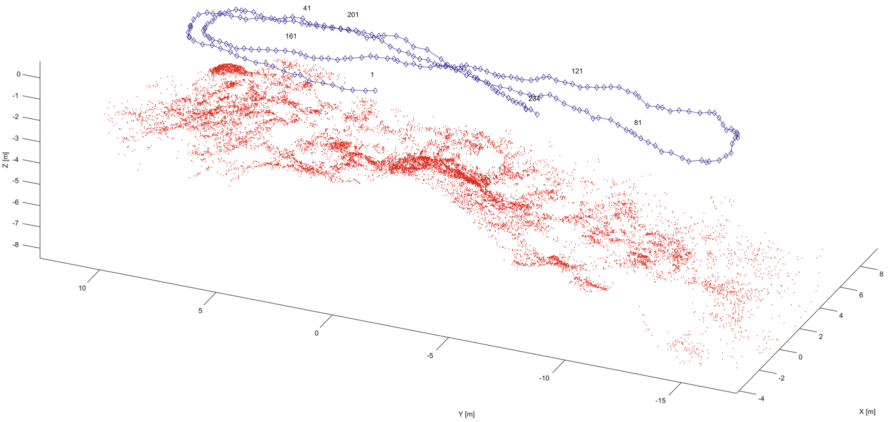


Fig. 4.25 Reef Experiment – 3D model and camera trajectory. The scene model contains $\simeq 62,000$ vertices. The trajectory of the camera has several crossovers.

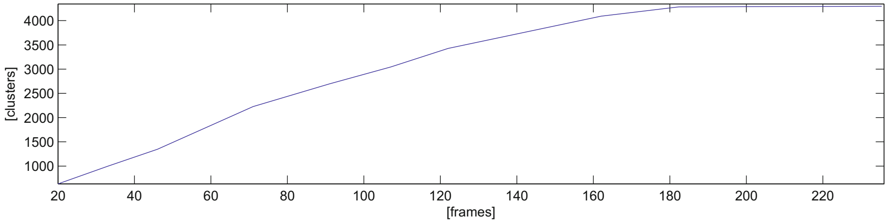


Fig. 4.26 Reef Experiment – Vocabulary size evolution. The vocabulary was initialized using the first 20 frames. After 9 updates, the final vocabulary contains $\simeq 3,400$ visual words.

the vocabulary increase rate slows and the vocabulary update frequency lowers, as there is little unmodeled visual information left in the scene.

After vocabulary building and image indexing, the resulting similarity matrix in Figure 4.27 successfully points out the cross-overs in the

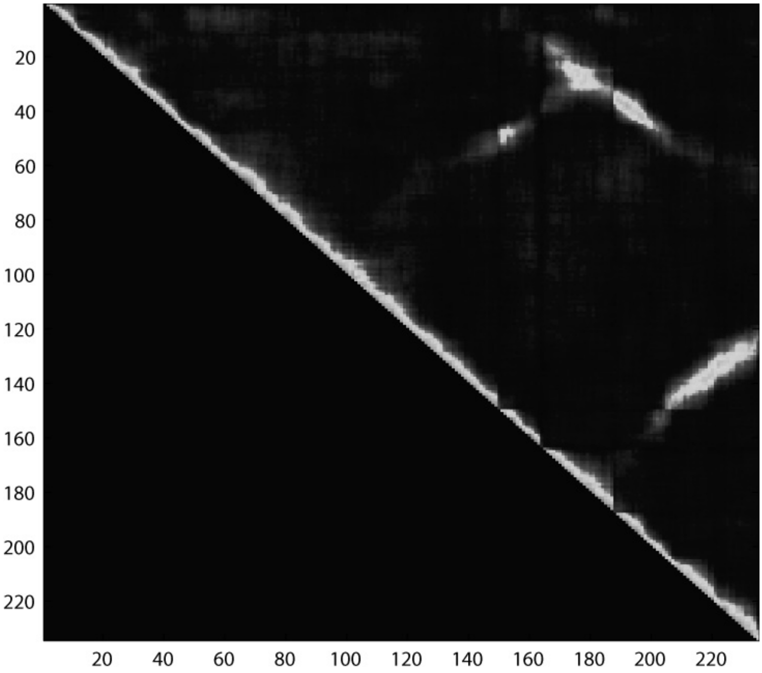


Fig. 4.27 Reef Experiment – Image similarity matrix. The bright regions off the main diagonal correspond to multiple cross-overs.

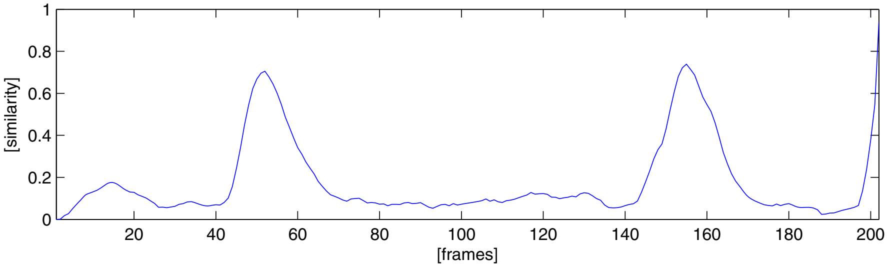
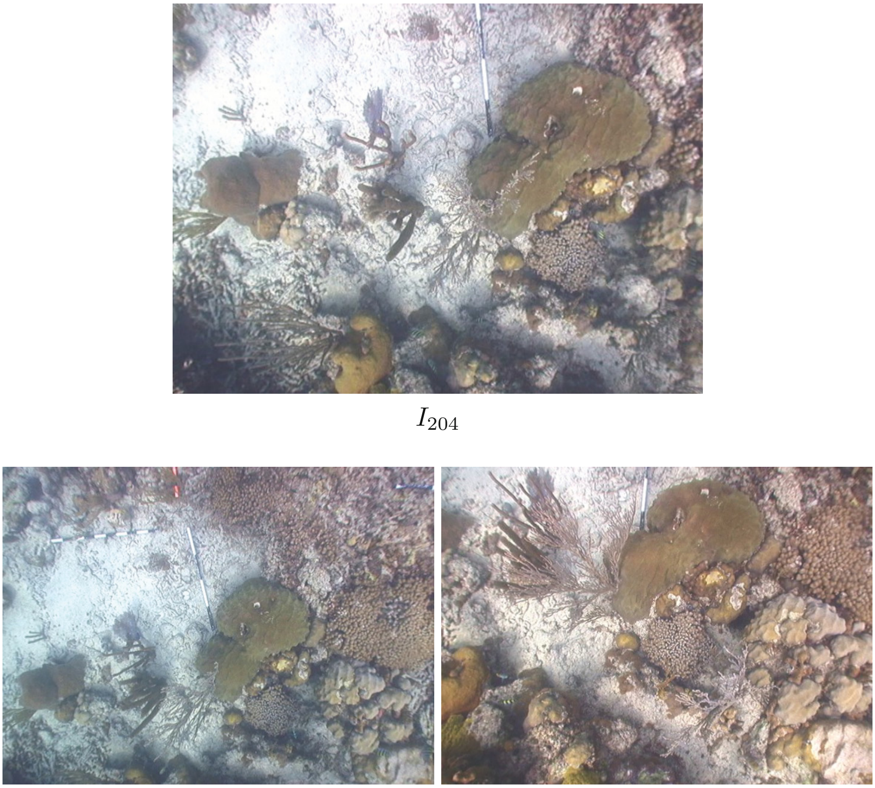


Fig. 4.28 Reef Experiment – Image similarity for query image I_{204}. The plot shows the similarity between frame I_{204} and all the previous frames. The two peaks corresponding to frames I_{52} and I_{155} indicate that all three frames correspond to the same region of the scene.



I_{204}

I_{52}

I_{155}

Fig. 4.29 Reef Experiment – Cross-over. Query frame I_{204} and frames I_{52} and I_{155} were successfully determined as corresponding to the same region of the scene, defining a loop closure.

camera trajectory. An exemplification of this is provided in Figure 4.28, where a query for frame 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 4.29 clearly illustrates that the three frames correspond to the same region of the scene.

To quantify the precision of the similarity matrix in approximating the image overlap, we compared it with the overlap ground truth using the average of absolute differences. The error was 0.095, higher than in the previous experiment. This is expected, since low contrast and high blurriness in underwater imaging decreases the quality of image features.

We compared the result with K-means vocabulary, using the same number of visual words as in the OVV in its final stage. The average error in case of K-means vocabulary is 0.0978, indicating that OVV yields slightly better results in case of underwater imaging.

4.4.5 Outdoor Experiment

Here, we discuss the loop closure detection for the Urban Experiment presented in Section 3.8.7. The visual vocabulary was generated and the images were indexed during the scene reconstruction. The final vocabulary contains 7,182 words. The resulting similarity matrix, shown in Figure 4.30, points out a cross-over between the first and last frames of the sequence. The situation is exemplified in Figure 4.31, where a query for frame I_{960} denotes a visual similarity of 0.8 with frame I_{45}. Figure 4.32 confirms that the two frames correspond to a loop closure.

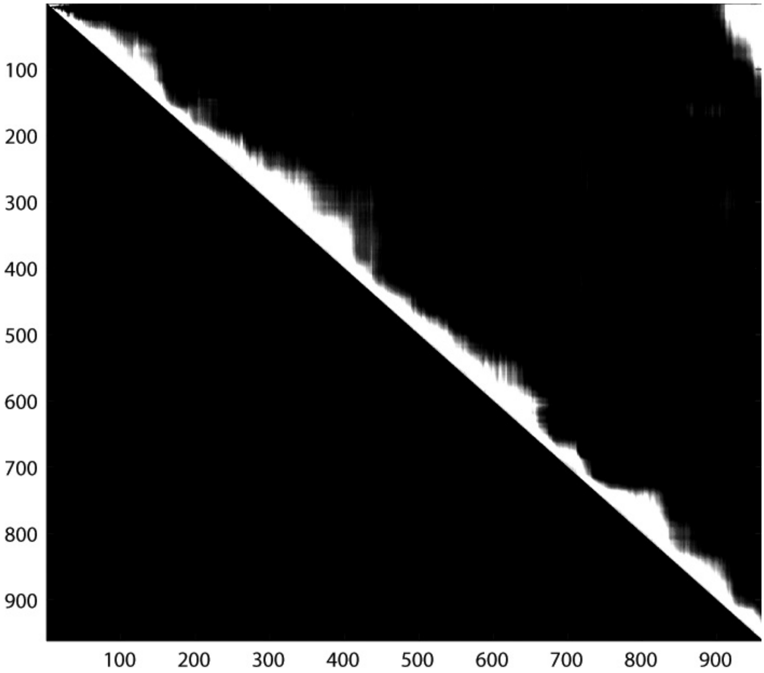


Fig. 4.30 Urban Experiment – Image similarity matrix. The bright region in the upper-right corner of the matrix indicates an overlap between first and last frames of the sequence.

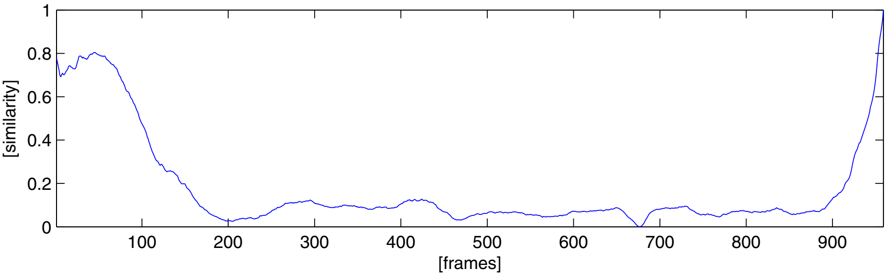


Fig. 4.31 Urban Experiment – Image similarity for query image I_{960}. The plot shows a high degree of visual similarity between frames I_{960} and I_{45}, corresponding to a loop closure.

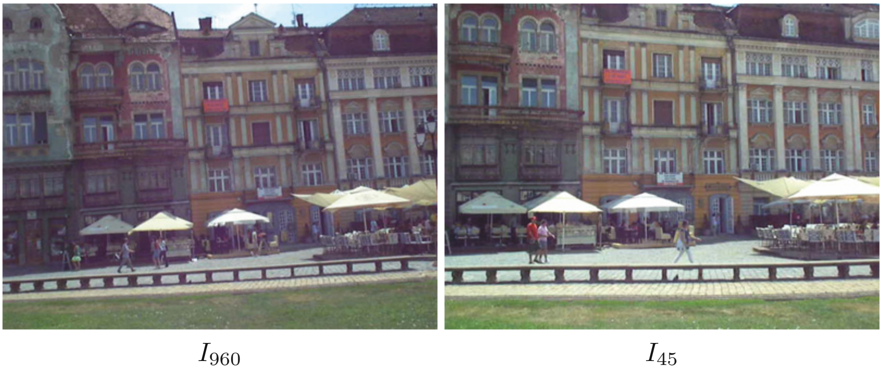


Fig. 4.32 Urban Experiment – Loop detection. Example of image pair corresponding to the loop closure.

In the remainder of this section, we present a test that we have carried out in order to assess the capacity of OVV indexing to be extended to other images of the same location. For this, we selected a set of photos from *Google Images* [59] depicting the Unirii Square, taken at different times of day and from various viewpoints. Each photo was then indexed using the generated vocabulary and the most visually similar image from the original dataset was extracted. Figure 4.33 illustrates the results. The majority of photos were correctly associated ($\simeq$ 90%). Generally, the cases where OVV did not correctly identify the location were the result of: (*i*) 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; (*ii*) severe obstructions that block most of the visual content modeled in the vocabulary; (*iii*) extreme lighting changes – pictures taken in the early evening or at night, where most

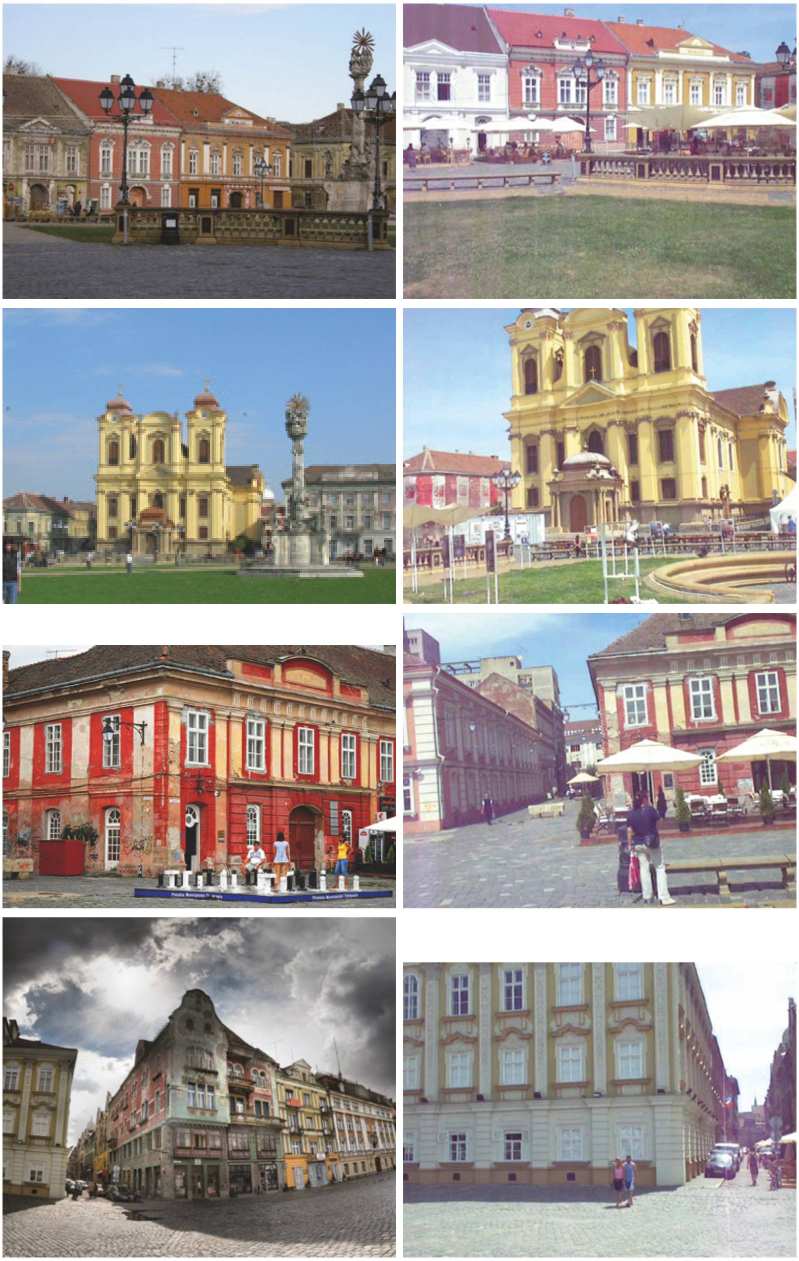


Fig. 4.33 Urban Experiment – Location identification. *Google Images* photos used as query images (left column) and the most visually similar image from the original dataset (right column). The last row shows an example of poor location identification, due to the post-processing of the query photo.

of the visual details are lost due to low contrast. Moreover, in the last row of Figure 4.33 we illustrate an example of poor localization due to HDR processing of the query image.

4.5 Discussion

We have developed a new visual BoW method for loop-closure detection, oriented towards 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 labelling method, that 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. Also, 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 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 chapter, we present 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.

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

The scope of this chapter is oriented towards 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.