

# Cluster-Based Loop Closing Detection for Underwater SLAM in Feature-Poor Regions\*

Pep Lluís Negre<sup>1</sup>, Francisco Bonin-Font<sup>1</sup> and Gabriel Oliver<sup>1</sup>

**Abstract**—This paper reports on a novel technique to visually detect loop closings in feature-poor underwater environments in order to increase the accuracy of vision-based localization systems. The main problem of the classical visual *Simultaneous Localization and Mapping* (SLAM) for underwater vehicles is the lack of robust, stable and matchable features in certain aquatic environments. The presence of sandbanks, seagrass or other underwater phenomena cause the visual features to concentrate in regions heavily textured, leaving great image areas completely free of visual information. In this situation, the classical loop closing detection algorithms fail, resulting in no corrections for the SLAM system. Our novel method proposes to reinforce the loop closing detection by clustering visual keypoints present in multiple keyframes and to match features of clusters instead of features of keyframes.

This new technique is assessed on the particular application of navigating an *Autonomous Underwater Vehicle* (AUV) in marine environments colonized with seagrass or with the presence of sandbanks. Experiments conducted in several coastal zones on the Balearic Islands show a high degree of success in the visual registration of overlapping areas.

## I. PROBLEM STATEMENT AND RELATED WORK

Underwater visual navigation is a highly challenging task in feature-poor, complex and extremely irregular textured environments. The situation is further complicated in shallow scenarios with rocks, sandbanks, molluscs, anemones or algae/seagrass that slightly swings with the swell. These environments present important difficulties to obtain stable visual keypoints, which results in instability in the calculation of the vehicle visual odometry and a great difficulty to register image pairs that partially overlap, especially if both frames have been taken from different viewpoints, distances or lighting conditions.

Figure 1 illustrates an example image of such environments, where (a) to (d) show, respectively, the ORB, SIFT, SURF and FAST features [8] matched between two consecutive frames, to highlight the stable keypoints. Darker areas correspond to the Mediterranean seagrass *Posidonia Oceanica* (P.O.) and clearer areas correspond to stones. Regardless the type of detector used, areas with seagrass generate very few stable and robust keypoints, being the most of them in areas with rocks. In this kind of environments, only a small portion of the image is useful to generate keypoints that characterize the region.

\*This work is partially supported by Ministry of Economy and Competitiveness under contracts TIN2014-58662-R, DPI2014-57746-C3-2-R and FEDER funds.

<sup>1</sup>Systems, Robotics and Vision group, University of the Balearic Islands (UIB), ctra de Valldemossa, km 7.5, 07122 Palma de Mallorca (Spain) plnegre@uib.cat, francisco.bonin@uib.es and goliver@uib.es.

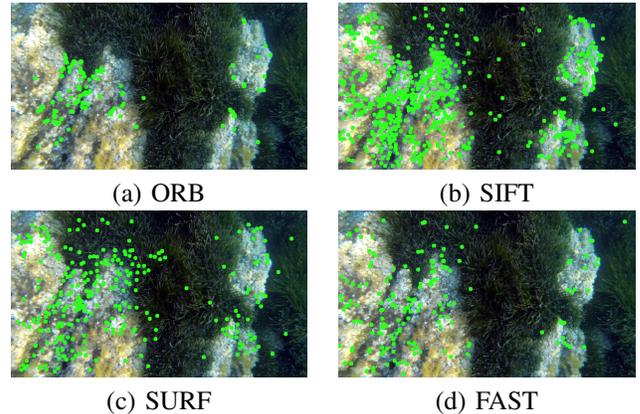


FIG. 1: (a) to (d) feature matchings in two consecutive frames for ORB, SIFT, SURF and FAST keypoints.

Usually, due to the difficulty of the environment, underwater robot localization systems are performed combining data provided by different sensors. Inertial units (accelerometer or gyroscope), a DVL (Doppler Velocity Log), a pressure sensor or other sensory systems are often integrated in a Kalman filter to localize the vehicle [13]. But all these approaches are prone to drift without a periodical correction given by a loop closing detection mechanism based on, for example, acoustic sonars [20] or stereo cameras [10], in the well-known SLAM approaches.

In the classical visual SLAM systems, the loop closing detection is performed between two keyframes [19], [15]. Normally, features of the last keyframe are matched with the features of all previously grabbed keyframes to find candidates to close a loop. The problem of applying these techniques in the underwater scenarios described above is that the feature matchings between pairs of keyframes may be insufficient. It is very difficult to describe its loop closing transformation, since the major part of the image does not have enough salient visual information.

Indeed, many of the underwater visual SLAM algorithms present on the literature [17] require the presence of artificial structures, such as boats, dams or marine platforms to succeed in the task. Very few visual SLAM systems are tested on natural seabeds [1] but, in any case, experiments have been always performed in environments rich in texture [14]. In the field of underwater hull inspection, some authors [7] improve the image registration process by labeling each frame with a saliency score in such a way that only visually salient keyframes are preserved. In [12], a particle filter is used to delimit the candidate images to close a loop with a query, in an sparse and optimized graph-map context.

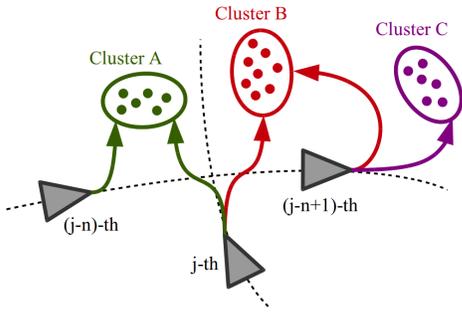


FIG. 2: Sample SLAM trajectory with keypoints clustering and loop closing detection between last inserted keyframe ( $j$ -th) and two past keyframes ( $i$ -th and  $(i+1)$ -th).

The main contribution of this paper is a novel technique that reduces the number of keyframes and increases the effectivity in the loop closing detection in poor textured underwater areas with respect to the classical solutions based on keyframe-to-keyframe matching. To achieve this goal, we designed a loop closing detection mechanism between the last keyframe and multiple past keyframes. We take advantage of the contrast between the poor textured (sand, seagrass) and the well textured (rocks, coral) regions present on the underwater scenarios to cluster the image keypoints. Then, when an area of the environment is imaged several times from different keyframes (i.e. viewpoints), although the overlapping area between image pairs is small, the total imaged area can provide a sufficient number of keypoints to be recognized when revisited. So, in principle, detecting a loop closure would be more effective if, instead of matching features of two overlapping keyframes, the system matches clusters of keypoints that can belong to several keyframes.

The idea is illustrated in figure 2. The dashed lines represent the robot trajectory at two different time instants of the mission. The triangles represent three robot positions where keyframes have been taken. There are two consecutive keyframes:  $(j-n)-th$  and  $(j-n+1)-th$ ; and one isolated:  $j-th$ . The *cluster A* is seen from keyframes  $(j-n)-th$  and  $j-th$ , while *cluster B* is seen from  $(j-n+1)-th$  and  $j-th$ , finally, *cluster C* belongs only to keyframe  $(j-n+1)-th$ . It can happen that the feature matchings between the two possible pairs of keyframes (i.e.  $(j-n)-th$  with  $j-th$  and  $(j-n+1)-th$  with  $j-th$ ) are not enough to close any of both loops or, may be, there are sufficient matches but they are concentrated at a very small region of the image, so the transformation obtained in the registration process is very inaccurate. However, it is clear that clustering features and using the combination of matchings of the keyframe  $j-th$  with the two keyframes  $(j-n)-th$  and  $(j-n+1)-th$ , the probability of closing a loop increases considerably and the final transformation will be more accurate. The example of figure 2 illustrates the case where one keyframe closes a loop with two past keyframes but our algorithm extends this idea to several keyframes present on the region.

However, the cluster-based loop closing scheme can not operate without a complete Graph-SLAM system. There-

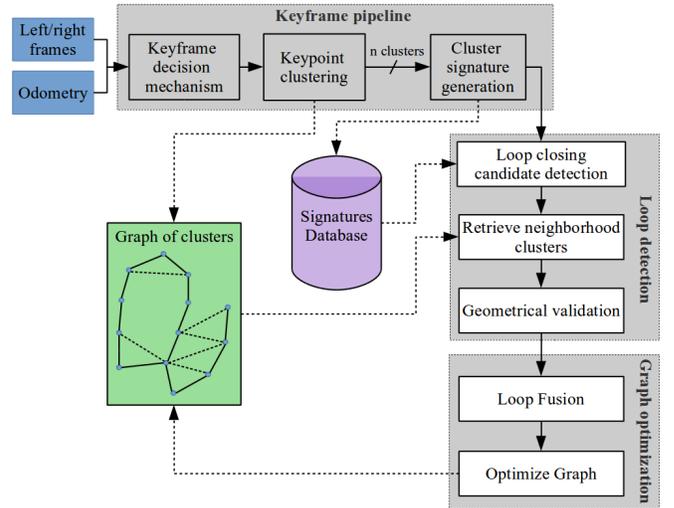


FIG. 3: Flowchart of the overall stereo Graph-SLAM process.

fore we have taken the previous work based on an online stereo Graph-SLAM [10] to integrate the new loop closing mechanism, modifying all those aspects necessary for proper operation.

This work is framed in the context of the ARSEA Spanish national project (TIN2014-58662-R), where marine areas are surveyed with an SPARUS II AUV [4] equipped with a stereo rig looking downwards to detect, map and control extended settlements of P.O.. This seagrass is an endemic specie of the Mediterranean sea that forms large meadows which play an extremely important role in the maintenance of the subsea ecosystems.

The remainder of this paper is organized as follows: Section II outlines the complete localization and mapping pipeline; Section III shows the experimental setup and some significant and successful results of loop closing detection, mapping and localization in feature-poor marine areas with P.O.. Section IV concludes the paper.

## II. THE OVERALL GRAPH-SLAM PIPELINE

Figure 3 shows an overview of the complete SLAM process, which incorporates three threads that run in parallel: the keyframe pipeline, the loop detection and the graph optimization. One of the main differences between the classical SLAM algorithms and our implementation is the absence of a tracking stage. Although we have tried some of the most popular tracking algorithms [15], [19], these fail in complex underwater environments. However, some stereo odometers have proved to work in these scenarios [21]. Therefore, we substitute the tracking stage for the *viso2* [6] stereo odometer. Synchronized images and odometry estimates are the system inputs, in such a way that, an estimated camera position can be assigned to each incoming image-pair. In the following paragraphs every algorithm step is explained separately.

### A. Keyframe Decision Mechanism

Only selected frames (so called keyframes) are passed to the algorithm pipeline to reduce the computation time and improve the localization performance. Most of the popular

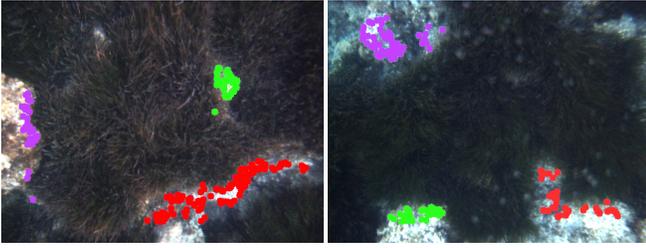


FIG. 4: Two sample images of the DBSCAN keypoint clustering with three clusters per image. Clusters are located on rocks and darker parts of the images correspond to P.O..

SLAM systems use a descriptor matching threshold, together with geometrical viewpoint constrains to decide when to insert a new keyframe [15]. But the environments in which we are working in this paper are poor in features, thus the keyframe decision mechanism takes into account only geometrical information. We use the method presented in [3] to estimate the overlap between the current frame and the last processed keyframe. When this overlap is smaller than a threshold, a new keyframe is passed to the pipeline.

### B. Keypoint Clustering and its Signature Generation

For every keyframe, ORB [18] keypoints are detected in the image and clustered using the popular DBSCAN [5] (see figure 4). Moreover, ORB descriptors are extracted and matched between stereo image pairs to triangulate the 3D coordinates of all clustered keypoints. For every cluster, its centroid relative to the camera frame, named  $X_c$ , is calculated from the 3D keypoint coordinates. ORB is used because its good performance and fast calculation and matching.

When the vehicle carries out long missions, searching loop closing candidates by matching descriptors of the current cluster with all previously stored ones has a high computational cost, non suitable for online applications. Instead, every cluster of keypoints is transformed into a light global signature (named *HALOC* [11]) by projecting the descriptor matrix. The use of *HALOC* instead of common bag-of-words methods for loop closing candidate selection is because its high performance on underwater environments. The ORB keypoints extracted in the previous stage are described with SIFT since *HALOC* needs integer elements. In short, for every cluster of keypoints, its SIFT descriptor matrix  $D_{n \times m}$  is extracted, where  $n$  is the number of visual keypoints included in the current cluster and  $m$  is the length of SIFT descriptors (typically, 128). The signature consists in a discretization of the descriptor space by projecting each column of  $D_{n \times m}$  onto a set of  $k$  orthogonal directions defined by unit vectors of  $n$  dimensions. All projections are computed as the scalar product and concatenated in a vector of  $k \cdot m$  dimensions, transforming the descriptor space size into a fixed size:  $\mathbb{R}^{n \times m} \rightarrow \mathbb{R}^{1 \times k \cdot m}$ .  $k$  is set to 3 following the indications of [11]. Finally, SIFT descriptors are discarded when signature is calculated.

Characterizing images with *HALOC* has shown to decrease considerably the computation time and to improve the performance in the processes of image comparison and loop

closing detection, with respect to the popular FAB-MAP, *Locality Sensitive Hashing* (LSH) and VLAD, in underwater environments [11].

### C. Management of Graph and Signatures Database

The graph (based on the popular g2o framework [9]) consists of nodes and edges representing the arrangement and connection of the clusters in the world coordinate frame. Every new cluster is inserted into the graph as a new node and stores the following information:

- A cluster unique identifier ( $i$ ).
- The corresponding keyframe unique identifier ( $j$ ).
- The estimate of the camera global position, named  $\mathbf{E}_{wc}(j)$ , taken as the odometry input (*viso2*) for the current keyframe.
- The node position ( $x,y,z$ ), relative to camera frame, named  $\mathbf{X}_c(i)$ . This is the cluster centroid.
- The node position ( $x,y,z$ ), relative to the world frame, named  $\mathbf{X}_w(i)$ .

$\mathbf{X}_w(i)$  is computed by:

$$\mathbf{X}_w(i) = \mathbf{E}'_{wc}(j)\mathbf{X}_c(i), \quad (1)$$

where  $\mathbf{E}'_{wc}(j)$  is the first estimate of the corrected camera global position corresponding to the current keyframe,  $j$ . This position ( $\mathbf{E}'_{wc}(j)$ ) uses all the graph optimizations until the insertion of cluster/s of the keyframe  $j - 1$  and the odometry estimates of keyframes  $j - 1$  and  $j$ :

$$\mathbf{E}'_{wc}(j) = \mathbf{E}''_{wc}(j - 1)\mathbf{D}_{j-1}^j, \quad (2)$$

where

$$\mathbf{D}_{j-1}^j = \mathbf{E}_{wc}^{-1}(j - 1)\mathbf{E}_{wc}(j), \quad (3)$$

and  $\mathbf{E}''_{wc}(j - 1)$  is the position of the camera corresponding to keyframe  $j - 1$  extracted from Eq. 1 after the last graph optimization.

The edges between clusters are set following these conventions:

- 1) The clusters corresponding to the same keyframe, named co-framed clusters, are linked by edges weighed with the maximum number of inliers allowed by the loop closing procedure detailed in section II-D.
- 2) The clusters between consecutive keyframes are linked by edges weighed with the number of inliers found applying the same procedure of loop closing detection. In the absence of inliers among all possible combinations of clusters between the current and previous keyframes, the closer two clusters are connected by an edge weighed with the minimum number of inliers configured in the loop closing detection stage.
- 3) When a loop closing is detected between clusters of non-consecutive keyframes (as shown in figure 2), an edge is added between all those cluster-pairs which have a significant number of inliers. These edges are weighed with its corresponding number of inliers.

Labeling edges that link co-framed clusters with a larger value generates a strong nexus among them. If later, a loop closing with these clusters is found, its corresponding link

in the graph will be the real number of inliers, establishing weaker connexions more pliable in each graph optimization.

On the other hand, the system builds incrementally a database (named *Signatures Database*) that contains an invert index, which stores for each unique node identifier ( $i$ ), its cluster signature, the ORB descriptors and the 3D stereo points (relative to camera frame) of all cluster keypoints.

#### D. Loop Detection and Validation

The loop closing detection stage is divided in three steps: 1) the selection of past cluster candidates to close loops with the clusters of the current keyframe, 2) the search for more cluster matches in the neighborhood of the candidates (if any) and 3) the geometrical validation based on a RANSAC scheme.

The search for loop closing candidates is performed by means of two different systems:

- 1) **By proximity.** For each cluster present in the current keyframe, the closest  $p$  clusters are retrieved using the updated positions of the graph.
- 2) **By cluster signature.** For each cluster present in the current keyframe, its signature is compared using  $l_1$ -norm (according to [11]) with all previous signatures stored in the database. One signature-to-signature matching with  $k = 3$  expends about 10ns in a commercial laptop. Thus, for example, a trajectory comprising 10.000 keyframes would spend approximately 0.1ms on the search for loop closing candidates for a given cluster.  $q$  clusters with the best signature matches are taken as candidates to close a loop.

In both cases, the clusters corresponding to the previous 10 keyframes are not included in the search to avoid loop closures with the immediately preceding neighbors.

Proximity and signature methods generate a set of loop closing candidate pairs, e.g. proximity mechanism could indicate that a cluster  $C_i$  of the current keyframe  $K_j$  has the cluster  $C_{i-s}$  (with  $s > 10$ ) as a loop closure candidate. Possible repeated candidate pairs coming from both methods are discarded before proceeding to the next step.

Once all possible loop closing candidates are obtained, the following steps are applied for every pair of candidate cluster and current cluster:

- 1) Retrieve the set  $S_r$  of  $r$  neighbor clusters of the candidate  $C_{i-s}$  by querying the graph.
- 2) From the *Signatures Database*, retrieve the ORB descriptors corresponding to  $C_{i-s}$ , named  $D(C_{i-s})$ , and to  $S_r$ , named  $D(S_r)$ .
- 3) The union  $D(C_{i-s}) \cup D(S_r)$  is matched with all the current keyframe descriptors,  $D(K_j)$ , using brute force matching with a hamming distance.
- 4) From the *Signatures Database*, extract the 3D stereo points corresponding to the matched features. Then, translate these 3D points to the world frame applying the following equation to each 3D point:

$$\mathbf{P}_w = \mathbf{E}''_{wc}(x)\mathbf{P}_c \quad (4)$$

where  $\mathbf{P}_c$  is the original 3D point relative to camera frame and  $\mathbf{E}''_{wc}(x)$  is the optimized global pose of the keyframe containing  $\mathbf{P}_c$ .

- 5) A 3D to 2D correspondences table is build using the 3D points calculated with Eq. 4 and the matched 2D keypoints of  $K_j$ .
- 6) 3D to 2D correspondences are then passed to an iterative RANSAC PnP algorithm to find a possible camera pose [2], named  $\mathbf{T}_{wc}(j)$  for the current keyframe  $K_j$ . A new loop closure is considered if a valid camera pose with enough inliers is found. Then the consecutive Loop Fusion stage (Section II-E) is responsible for integrating this new loop closure into the graph.

In the case that a valid loop closure is detected, every inlier relates a point contained in a cluster of the current keyframe  $K_j$  with a point located in a cluster of the region defined by  $C_{i-s} \cup S_r$ .

A table indicating the number of inliers for each pair of those aforementioned related clusters is built. This table, together with the new camera pose,  $\mathbf{T}_{wc}(j)$ , for the keyframe  $K_j$ , will be used in the next stage to determine the edges and its corresponding weights into the graph.

Notice that feature matching is not performed only on a cluster pair, instead all the keypoints of the current keyframe are matched with the keypoints in the vicinity of the candidate cluster. Therefore, this region may contain clusters from several past keyframes increasing considerably the probability of loop closure detection.

#### E. Loop Fusion and Graph Optimization

When a loop closing is successfully detected by the previous stage, the loop fusion mechanism is responsible for inserting the different edges and its weights between clusters (nodes) into the graph.

Loop Detection and Validation stage provides the corrected camera pose for the current keyframe,  $\mathbf{T}_{wc}(j)$ , and a table containing the number of inliers for every cluster pair.

The following operations are applied for every cluster included in the current keyframe  $K_j$ :

- 1) The cluster global position,  $X_w(a)$ , is recalculated according to:

$$\mathbf{X}_w(a) = \mathbf{T}_{wc}(j)\mathbf{X}_c(a), \quad (5)$$

where  $a \in \mathbb{N}$ ,  $0 < a \leq N$ , being  $N$  the number of clusters of the current keyframe.

- 2) The edge between a cluster  $C_a$  of the current keyframe and its corresponding counterpart  $C_b$  included in  $C_{i-s} \cup S_r$  is composed by:

$$\mathbf{Q}_{b,a} = \mathbf{X}_w^{-1}(a)\mathbf{X}_w(b), \quad (6)$$

where  $\mathbf{X}_w(b)$  is the position of  $C_b$ , extracted from the updated graph with Eq. 1, substituting  $i$  for  $b$ .

At the end of this process, we obtain different edges defined by a set of transformations as Eq. 6 and its number of inliers. All of them are inserted into the graph weighed

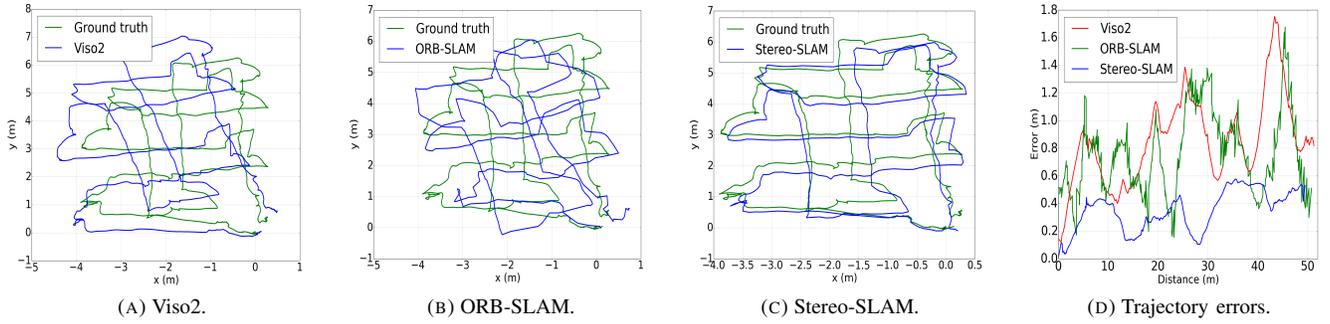


FIG. 5: Indoor Tank results. Plots (A), (B) and (C) illustrate the trajectories of *viso2*, *ORB-SLAM* and *Stereo-SLAM* respectively. Plot (D) shows the trajectory errors for every approach with respect to the ground truth.

with the corresponding number of inliers as explained in Section II-C. Finally a pose graph optimization is performed to achieve global consistency.

### III. EXPERIMENTS

We have performed an extensive experimental validation of our system in two different environments: an indoor water tank and an outdoor coast region of the Balearic Islands.

For each experiment, the trajectories estimated by a visual odometer *viso2* [6], *ORB-SLAM* [15] and our system, named, from now on, *Stereo-SLAM* are computed. The choice of *ORB-SLAM* as benchmark to be compared with our *Stereo-SLAM* is because it is one of the most modern, popular and versatile keyframe-to-keyframe registration SLAM systems that we have proven to work on underwater environments. However, *ORB-SLAM* is a monocular algorithm and, before it can be directly compared with *Stereo-SLAM*, its pose estimates must be conveniently scaled. To this end, the AUVs used for the experiments have been equipped with a pressure sensor that provides a depth measure in meters. This measure, contrasted with estimates of the *ORB-SLAM*, is used to compute a factor to scale the vehicle trajectory.

#### A. Indoor Tank

These experiments used data gathered during the TRIDENT project<sup>1</sup>, and were performed in a water tank located in the University of Girona moving the AUV Girona500 [16] equipped with a stereo camera pointing downwards, in several programmed trajectories. The water tank bottom is covered with a poster that reproduces a marine environment without 3D structure. Ground truth is extracted by matching each image that has been captured online against the known image of the poster printed on the floor of the pool. As the size of the print is known, 6 DOF camera poses can be computed minimizing reprojection errors of matched features. In the experiment of Fig. 5, a survey 50m long is performed passing several times through same locations to close loops.

Figures 5-(a) to 5-(c) illustrate the trajectories of *viso2*, *ORB-SLAM* and *Stereo-SLAM* with respect to the ground

Algorithm	Trans. ME (m)	#LC	#KF
<i>Viso2</i>	0.86	-	-
<i>ORB-SLAM</i>	0.74	4	271
<i>Stereo-SLAM</i>	0.35	5	128

TABLE I: Indoor tank experiments: trajectory mean errors and number of loop closures and keyframes.

truth. Figure 5-(d) shows the trajectory errors concerning to this indoor experiment. Errors are computed by comparing each trajectory position against its corresponding point in the ground truth. Quantitative values are shown in Table I. The column *Algorithm* indicates the localization approach, the column *Trans. ME* shows the translation mean error, that is, the mean of the trajectory error shown in plot 5, and finally, the columns *#LC* and *#KF* indicate the number of loop corrections and the number of keyframes for the SLAM algorithms. *Stereo-SLAM* closes loops between clusters instead of keyframes, this causes that one keyframe can close loop with several past keyframes. Thus, to be fair to *ORB-SLAM*, we consider all the keyframe-to-keyframe loop closures, that is, if one keyframe closes loops with 3 past keyframes the total number of loop closings is increased by 3.

The printed image on the floor of the test pool is rich on textures, thus *ORB-SLAM* and *Stereo-SLAM* close a similar number of loops. Even so, *Stereo-SLAM* closes one more loop, resulting in a more accurate trajectory with respect to the ground truth. Moreover, *Stereo-SLAM* finishes the trajectory using less number of keyframes, resulting in better use of system resources.

#### B. Balearic Coast

A Sparus II AUV [4] equipped with a stereo camera pointing downwards is used to survey areas of the Balearic coast to detect, map and control extended settlements of P.O. seagrass. Two different missions were programmed: one small mission surveying an area of 10m x 12m (Fig. 7-a) and a longer mission which surveys an area of 20m x 22m (Fig. 7-b).

GPS is not available underwater and the vehicle does not have any tracking system that can be used as ground truth. However, in the first experiment, we installed an

<sup>1</sup>See <http://www.irs.uji.es/trident/>

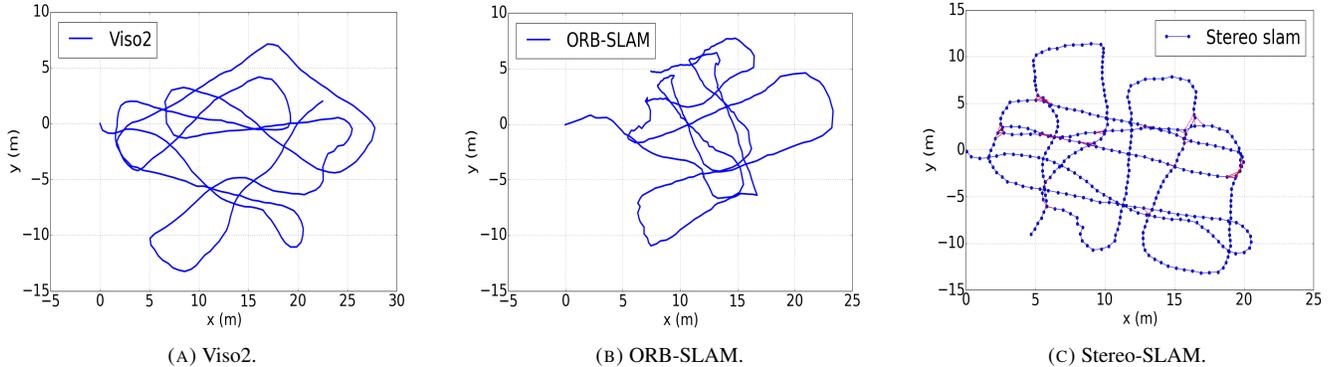


FIG. 6: Balearic coast experiment 2 (large mission of  $20m \times 22m$ ) results. Plots (A), (B) and (C) illustrate the trajectories of *viso2*, *ORB-SLAM* and *Stereo-SLAM* respectively. In (C), keyframes are represented with dots and the edges (corresponding to loop closures) are drawn in red.

Algorithm	Trans. Error (%)	#LC	#KF
<i>Viso2</i>	7.85%	-	-
<i>ORB-SLAM</i>	7.43%	0	512
<i>Stereo-SLAM</i>	2.98%	11	223

TABLE II: Balearic coast experiment 1 (small mission of  $10m \times 12m$ ) results: trajectory errors and number of loop closures and keyframes.

*AR - Marker* on the seabed (see Fig. 7-a), located at a perfectly known position with respect to the origin, to obtain the absolute position of the vehicle when navigating over the marker. There was no ground truth for any point of the trajectory, but calculating the drift of the AUV at the marker was possible.

The mission carried out in the first experiment of  $10m \times 12m$  is a grid over an area of  $120m^2$  colonized with P.O., with steps each  $5m$  and  $6m$  along the axes, resulting in a trajectory  $88m$  long (see Fig. 7-a).

Table II details the experiment results. The column *Algorithm* indicates the localization approach, the column *Trans. Error* shows the translation error at the marker position, and finally, the columns *#LC* and *#KF* indicate the number of loop corrections and the number of keyframes for the SLAM algorithms. *ORB-SLAM* did not close any loop during the mission, thus its error at the marker is similar to the visual odometry since only its tracking stage provided pose estimates.

On the other hand, *Stereo-SLAM* is able to close up to 11 loops across the entire robot mission. *Stereo-SLAM* uses these corrections to significantly improve the position provided by the visual odometer, from an error of 7.85% to 2.98% at the marker position. In this experiment, the total number of keyframes stored by the SLAM algorithms also shows a significant improvement of *Stereo-SLAM* versus *ORB-SLAM*.

For the second experiment<sup>2</sup> of  $20m \times 22m$ , it was not possible to install a marker because of the depth at which the mission was carried out. However, this mission was longer



FIG. 7: Two programmed missions on the AUV control application which integrates Google Maps. (a) Small mission of  $10m \times 12m$  with an *AR-Marker* installed on the seabed at a specific point and (b) large mission of  $20m \times 22m$ .

than the previous and included more programmed trajectory crossings to facilitate the loop closure detections by the SLAM algorithms. Therefore, based on the final shape of the trajectories and the total number of closed loops, the results can be assessed qualitatively.

The mission carried out in this experiment of  $20m \times 22m$  is a grid over an area of  $440m^2$  colonized with P.O., with steps each  $5m$  and  $5.5m$  along the axes, resulting in a trajectory  $274m$  long (see Fig. 7-b).

Figure 6 illustrates the trajectories of *viso2*, *ORB-SLAM* and *Stereo-SLAM*. The internal graph of the *Stereo-SLAM* algorithm represents the clusters and their connections, but in Fig. 6-(c) a graph corresponding to the keyframes is shown. This is possible since the camera position for every cluster is stored, as explained in Section II-C.

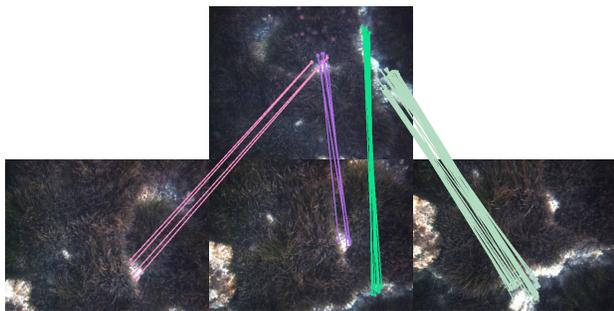
Table III indicates the number of loop closures and keyframes for the *ORB-SLAM* and the *Stereo-SLAM* algorithms. Note that *ORB-SLAM* closes only one loop while *Stereo-SLAM* performs 121 loop corrections. In addition, this large improvement in the number of loop closure detections is achieved with a considerably smaller number of keyframes.

Finally, figure 8 shows two samples of the set of 121 loop detections performed by the *Stereo-SLAM* algorithm. In

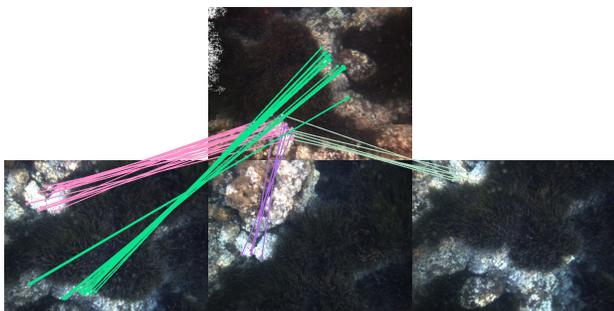
<sup>2</sup>See the video on <https://youtu.be/C4U8eaPzrLg>

Algorithm	#LC	#KF
ORB-SLAM	1	1569
Stereo-SLAM	121	363

TABLE III: Balearic coast experiment 2 (large mission of 20m x 22m) results: number of loop closures and keyframes.



(a) Keyframe 225 closes loop with keyframes 102, 103, 104.



(b) Keyframe 279 closes loop with keyframes 18, 19, 20.

FIG. 8: Two keyframe to multi-keyframe loop closure samples for the Balearic coast experiment 2 (large mission of 20m x 22m). The colored lines correspond to inliers of different clusters for the current keyframe.

both cases, the clusters of the current keyframe (top) close loops with the clusters of the 3 past keyframes (bottom). It can also be seen that these two samples are particularly significant, since the number of inliers between pairs of keyframes would not be enough to validate the loop, but the combination of the inliers of all loop closings provides an accurate transformation.

#### IV. CONCLUSIONS

This work presents a new loop closure detection method which goes one step forward with respect to the current SLAM approaches when applied in poor-feature underwater environments.

The keyframe decision mechanism based on image overlap instead of feature matching threshold ensures that the number of keyframes is appropriate for these scenarios, reducing the computational time and the system resources. Moreover, the search for loop closure candidates using *HALOC* shows again that it is better than *BoW* for underwater environments, as detailed in [11]. Finally, the keyframe to multi-keyframe loop closure mechanism (based on keypoint clusters) closes many loops that would not be detected in a classical keyframe-to-keyframe system. All these issues lead to a considerable

improvement in the AUV localization.

#### REFERENCES

- [1] J. Aulinas, Y. R. Petillot, J. Salvi, and X. Lladó, "The slam problem: a survey." in *CCIA*. Citeseer, 2008, pp. 363–371.
- [2] M. Bujnak, Z. Kukulova, and T. Pajdla, "New efficient solution to the absolute pose problem for camera with unknown focal length and radial distortion," in *Computer Vision—ACCV 2010*. Springer, 2011, pp. 11–24.
- [3] A. Burguera, F. Bonin-Font, and G. Oliver, "Reducing the computational cost of underwater visual slam using dynamic adjustment of overlap detection," in *Emerging Technology and Factory Automation (ETFA), 2014 IEEE*. IEEE, 2014, pp. 1–8.
- [4] M. Carreras, C. Candela, D. Ribas, A. Mallios, L. Magí, E. Vidal, N. Palomeras, and P. Ridao, "SPARUS II, Design of a Lightweight Hovering AUV." in *Proceedings of the Fifth International Workshop in Marine Technology (MARTECH)*, Girona (Spain), 2013.
- [5] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, Portland, USA, 1996, pp. 226–231.
- [6] A. Geiger and J. Ziegler, "Stereoscan: Dense 3d reconstruction in real-time," in *IEEE Intelligent Vehicles Symposium*, jun 2011.
- [7] A. Kim and R. Eustice, "Real-time visual slam for autonomous underwater hull inspection using visual saliency," *IEEE Transactions on Robotics*, vol. 29, no. 3, pp. 719–733, 2013.
- [8] S. Krig, *Computer Vision Metrics. Survey, Taxonomy, and Analysis*. Apress, Springer Link, 2014, ch. Interest Point Detector and Feature Descriptor Survey, pp. 217–282.
- [9] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 3607–3613.
- [10] P. L. Negre, F. Bonin-Font, and G. Oliver, "Stereo Graph SLAM for Autonomous Underwater Vehicles," in *Proc. of International Conference on Intelligent Autonomous Systems (IAS)*, 2014.
- [11] P. L. Negre, F. Bonin-Font, and G. Oliver-Codina, "Global image signature for visual loop-closure detection," *Autonomous Robots*, pp. 1–15, 2015. [Online]. Available: <http://dx.doi.org/10.1007/s10514-015-9522-4>
- [12] P. Ozog and R. Eustice, "Toward long-term, automated ship hull inspection with visual slam, explicit surface optimization, and generic graph-sparsification," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, 2014, pp. 3832–3839.
- [13] N. Palomeras, A. El-Fakdi, M. Carreras, and P. Ridao, "Cola2: A control architecture for auvs," *Oceanic Engineering, IEEE Journal of*, vol. 37, no. 4, pp. 695–716, 2012.
- [14] O. Pizarro, R. Eustice, and H. Singh, "Large area 3d reconstructions from underwater surveys," in *OCEANS'04. MTS/IEEE TECHNO-OCEAN'04*, vol. 2. IEEE, 2004, pp. 678–687.
- [15] R. Mur-Artal, J. M. M. Montiel and J. D. Tardós, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," *IEEE Transactions on Robotics (Accepted)*, 2015.
- [16] D. Ribas, N. Palomeras, P. Ridao, M. Carreras, and A. Mallios, "Girona 500 auv: From survey to intervention," *Mechatronics, IEEE/ASME Transactions on*, vol. 17, no. 1, pp. 46–53, 2012.
- [17] D. Ribas, P. Ridao, J. Tardós, and J. Neira, "Underwater slam in a marina environment," *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pp. 1455–1460, 29 2007–Nov. 2 2007.
- [18] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: an efficient alternative to sift or surf," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2564–2571.
- [19] H. Strasdat, A. Davison, J. Montiel, and K. Konolige, "Double window optimisation for constant time visual slam," in *IEEE International Conference on Computer Vision (ICCV)*, Barcelona (Spain), 2011.
- [20] S. B. Williams, P. Newman, G. Dissanayake, and H. Durrant-Whyte, "Autonomous underwater simultaneous localisation and map building," in *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, vol. 2. IEEE, 2000, pp. 1793–1798.
- [21] S. Wirth, P. L. Negre, and G. Oliver, "Visual odometry for autonomous underwater vehicles," in *Proceedings of the IEEE Oceans*, Bergen, Norway, 2013.