

# Selective Submap Joining for Underwater Large Scale 6-DOF SLAM

Josep Aulinas, Xavier Lladó and Joaquim Salvi  
Computer Vision and Robotics Group  
Institute of Informatics and Applications  
University of Girona  
17071 Girona, Spain  
{jaulinas, llado, qsalvi}@eia.udg.edu

Yvan R. Petillot  
Oceans Systems Lab  
School of Engineering and Physical Sciences  
Heriot Watt University  
Edinburgh EH14 4AS, United Kingdom  
Y.R.Petillot@hw.ac.uk

**Abstract**—Autonomous Underwater Vehicles (AUVs) need positioning systems different than the Global Positioning System (GPS), which does not work in underwater scenarios. A possible solution to this lack of GPS signal are the Simultaneous Localization and Mapping (SLAM) algorithms. SLAM algorithms aim to build a map while simultaneously localize the vehicle within it. These algorithms suffer from several limitations in front of large scale scenarios. For instance, they do not perform consistent maps for large areas, mainly because uncertainties increase with the scenario. In addition, the computational cost increases with the map size. It has been demonstrated that the use of local maps reduces computational cost and improves map consistency. Following this idea, in this paper we propose a new SLAM technique based on using independent local maps, combined with a global level stochastic map. The global level contains the relative transformations between local maps. These local maps are updated once a new loop is detected and the amount of overlapping between local maps is high. Thus, maps sharing a high number of features are updated through fusion, maintaining the correlation between landmarks and vehicle. Experimental results on real data obtained from the REMUS-100 AUV show that our approach is able to obtain large map areas consistently.

## I. INTRODUCTION

Remotely Operated underwater Vehicles (ROVs) and Autonomous Underwater Vehicles (AUVs) have been developed in order to explore underwater environments. ROVs are linked to the ship by a tether and operated by a person on board the ship. The tether is a group of cables that carry electrical power, video and data signals back and forth between the operator and the vehicle. AUVs do not have any umbilical cable. Therefore, they require complete autonomy. In order to achieve this autonomy, the AUV is equipped with on board sensors, which provide information about the vehicle, such as speeds, orientations or accelerations, and about the environment, such as the relative location of salient features with respect to the vehicle. This information is used to estimate the approximate position of the vehicle and to build a stochastic map of the area where the vehicle navigates. Computing both, the map and the position, at the same time is known as Simultaneous Localization and Mapping (SLAM) or Concurrent Mapping and Localization (CML) [1]. Some examples of applications where SLAM could be used are underwater cartography, geological mapping, off-shore structures inspection, studies of biodiversity, deep-water archaeology and any sort of underwater activity that might be harmful for humans.

In an underwater environment it is difficult to use cameras due to the lack of visibility and scattering. Laser range finders are imprecise when working in these scenarios, and the GPS does not work in these conditions, since the water attenuates the electromagnetic waves. The most commonly used sensors to measure navigation data on an AUV are the Inertial Measurement Unit (IMU) and the Doppler Velocity Log (DVL), while acoustic cameras or side-scan sonar are used to gather data from the environment. The IMU and the DVL do not give absolute localization, therefore if the vehicle is wrongly localized, nor the IMU neither the DVL will provide useful information to recover the right position. In addition, as positioning is relative to past information, the localization problem is biased and the measurement noise produces drift. Besides, the detection of salient features is a complex task, due to the fact that sonar images are also noisy. The sensor noise together with the lack of other navigation aids makes the mapping and localization of underwater vehicles a more difficult challenge compared to terrestrial or aerial vehicles.

Several SLAM approaches have been proposed aiming to solve the localization and mapping problem. However, most of them suffer from two important drawbacks when dealing with large maps: 1) the computational demand increases with the map size, and 2) the map becomes inconsistent. To overcome these limitations other techniques introduced the idea of submapping, i.e. split the whole scenario into various small local maps [2], [3], [4], [5]. It is important to stress that submapping techniques are more common on terrestrial applications, where a 3-DOF system applies. In this paper we present a new submapping SLAM technique for underwater scenarios based on a Selective Submap Joining. As well as of introducing the vehicle definition and the motion and observation models for a 6-DOF AUV, we also describe our main contribution which relies on the strategy used to decide whether to fuse the submaps. Experimental results on real data obtained from the REMUS-100 AUV show that our Selective Submap Joining SLAM is able to obtain large map areas consistently.

The paper is structured as follows: Section II gives a summary on current SLAM techniques. Section III describes the insights of our approach. Section IV presents the experimental setup, the conducted tests and the results. Finally, conclusions and future work are presented in Section V.

## II. SIMULTANEOUS LOCALIZATION AND MAPPING

Fully autonomous vehicles must be able to localize themselves even when there is no prior information of the environment. Several approaches tackle this localization problem assuming an *a-priori* known scenario. For instance, a vision-based localization technique was proposed in [6], using a coded pattern placed on the bottom of a water tank and an on-board downward looking camera. The main problem with this strategy is that in real environments is difficult to deploy a coded pattern on the seabed. In addition, vision systems have a very limited working range (1-5m) on underwater applications. Other approaches use a GPS-aided localization, but the attenuation of electromagnetic waves through the medium of water limits the application of GPS to near surface activities. For instance, in [7] GPS is used to localize buoys, which send acoustic signals to the AUV, while in [8], the AUV receives GPS signals while floating and then dives to exchange acoustic messages with underwater sensors. A standard xyz positioning system for underwater vehicles is the long-baseline (LBL) acoustic transponder. LBL operates on the principle of time-of-flight and it has been proven to operate up to a range of 10 km [9]. The main drawback of LBL is that it requires two or more acoustic transponder beacons to be deployed to the sea floor. Short-baseline (SBL) systems provide more accurate positioning information, but suffer from the same drawbacks than the LBL. Another set of approaches avoid the use of external devices by using computer algorithms. For instance, the use of particle filters for AUV localization presented in [10]. This approach is shown to work with high performance, although, it only works when the map is known *a-priori*.

In many situations, the map is unknown, and therefore there is the need to simultaneously estimate the map and localize the vehicle inside this map. This problem known as SLAM is one of the fundamental challenges of robotics [1]. The SLAM problem involves finding appropriate representation for both the observation and the motion models, which is generally performed by computing its prior and posterior distributions using probabilistic algorithms, for instance Kalman Filters (KF), Particle Filters (PF) and Expectation Maximization. These probabilistic techniques are very popular in the SLAM context because they tackle the problem by explicitly modelling different sources of noise and their effects on the measurements [11].

### A. The Extended Kalman Filter

The Extended Kalman Filter (EKF) is a well known and widely used filter in the context of SLAM [12]. EKF-SLAM represents the vehicle's pose and the location of a set of environment features in a joint state vector. This vector is estimated and updated by the EKF. EKF provides a suboptimal solution due to several approximations and assumptions. EKF complexity grows with the number of landmarks, because each landmark is correlated to all the other landmarks. This means that EKF memory complexity is  $O(n^2)$  and a time complexity of  $O(n^2)$  per step, where  $n$

is the total number of features stored in the map. Thus, EKF becomes dramatically expensive when dealing with large areas.

Several researchers propose solutions to solve the limitations of the EKF-SLAM in large areas. In terms of computational complexity, in [13] the authors propose to delay the global update stage after several observations reducing significantly the cost. Regarding map consistency, the Unscented Kalman Filter (UKF) [14] achieves better consistency addressing the approximation and assumption issues of the EKF, but with a higher computational complexity. Other approaches reduce the computational cost taking advantage of the sparsity structure of the covariance matrix inverse [15]. These are the so called Information Filter (IF) base techniques, which have problems with the data association since no covariance matrix is involved in the process. Overall, EKF is still one of the most popular and reliable filters in SLAM.

### B. Submapping Techniques

Several submapping techniques have been proposed in order to address the problems of consistency and computational complexity of a standard EKF-SLAM. An early example is the Decoupled Stochastic Map (DSM) approach [16]. However, with this approach the maps tend to be inconsistent because the correlations are broken. The Constrained Local Submap Filter (CLSF) [17] and the Local Map Joining (MJS) [2] produce efficient global maps by consistently combining completely independent local maps. The Divide and Conquer SLAM (DCS) [3] is capable to recover the global map in approximately  $O(n)$  time. The Constant Time SLAM (CTS) [18], the Atlas approach [19], and the Hierarchical SLAM (HS) [4] store the link between local maps by means of an adjacency graph. The HS imposes loop constraints on the adjacency graph, producing a better estimation of the global level map. The Conditionally Independent Local Maps (CILM) [5] is based on sharing information between consecutive submaps. This way, a new local map is initialized considering the *a-priori* knowledge.

None of them, however, has been tested on underwater scenarios, where some extra constraints apply. Firstly, the map sensing is limited to either acoustics or short range vision. Secondly, underwater scenarios are in general unstructured and require 3D navigation (6-DOF motion), while most current SLAM solutions are used on man-made (geometrically simple) indoor spaces, where a 2D map representation is sufficient. Therefore, this paper focuses on the use of SLAM on AUV navigation, which requires further testing and improvements.

## III. MAPPING LARGE AREAS - SELECTIVE SUBMAP JOINING SLAM

### A. Overview of Our Approach

The main idea of our approach is to use the EKF based SLAM to build local maps. The size of these local maps is bounded by the total number of features and by the level of uncertainty. The relative topological relationship between

---

**Algorithm I:** Selective Submap Joining SLAM

---

```
begin mission
while navigating do
   $\hat{\mathbf{x}}_i, \hat{\mathbf{P}}_i = \text{EKF SLAM}() \leftarrow (\text{Build submap } \mathcal{M}_i)$ 
   $\hat{\mathbf{x}}_G, \hat{\mathbf{P}}_G = \text{build global map}(\hat{\mathbf{x}}_i, \hat{\mathbf{P}}_i)$ 
   $\mathcal{H}_{Loop} = \text{check possible loops}(\hat{\mathbf{x}}_G, \hat{\mathbf{P}}_G)$ 
  for  $j = \mathcal{H}_{Loop}$  do
    refer  $\mathcal{M}_i$  and  $\mathcal{M}_j$  to a common base reference
     $\mathcal{H}_{ij} = \text{data association}(\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_j, \hat{\mathbf{P}}_i, \hat{\mathbf{P}}_j)$ 
    if  $\mathcal{H}_{ij} > \text{threshold}$  then
       $\hat{\mathbf{x}}_{ij}, \hat{\mathbf{P}}_{ij} = \text{map fusion}(\hat{\mathbf{x}}_i, \hat{\mathbf{P}}_i, \hat{\mathbf{x}}_j, \hat{\mathbf{P}}_j, \mathcal{H}_{ij})$ 
       $\hat{\mathbf{x}}_G, \hat{\mathbf{P}}_G = \text{update global map}(\hat{\mathbf{x}}_{ij}, \hat{\mathbf{P}}_{ij})$ 
    endif
  endfor
endwhile
```

---

consecutive local maps is stored in a global level map. The global level is used to search for loop closure, i.e. the vehicle is revisiting a region. Once a loop is detected, the data association between those maps that are closing the loop is computed. The loop closing strategy involves a decision on whether to fuse local maps depending on the amount of found correspondences. The whole process is presented in *Algorithm I*. In the following sections we present the details of all the steps.

### B. Global Map Building

Local maps  $\mathcal{M}_i \mathcal{M}_{i+1} \dots \mathcal{M}_j$  are built sequentially using the EKF-SLAM algorithm (see Fig. 1). Every local map has its reference frame w.r.t. the vehicle's starting point. The reference frame of a local map  $\mathcal{M}_{i+1}$  coincides with the last vehicle's position in the previous map  $\mathcal{M}_i$ . Therefore, the relative transformation between two consecutive map's  ${}^{\mathcal{M}_i}_{\mathcal{M}_{i+1}}\mathcal{T}$  is equivalent to the vehicle's pose at the last position of  $\mathcal{M}_i$ . This link is stored in a global level map  $X_G$  together with its uncertainty  $P_G$  (1). A new link is stored every time a local map reaches its end. The information contained in this global level is very important to detect loops, because each local map can be back referred to any other map and its information can be then used by data association algorithms to confirm loop closure.

### C. Loop Closing

This global level can be understood as an adjacency graph, where each local map is a node, and the transformations between local maps are the links. The global level is used to check the possibility of being in front of loops. A loop closure is accepted when the vehicle is revisiting a region. In order to know the size of the revisited region, the data

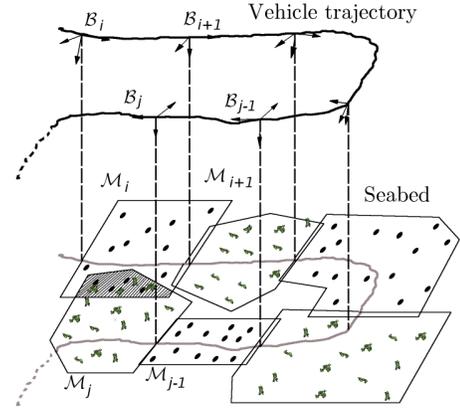


Fig. 1. Schematic representation of a sequence of submaps. The dashed area emulates loop closing, where two submaps share several landmarks.

association between those maps that are closing the loop is computed, using the Joint Compatibility Branch and Bound algorithm [20]. The loop closing strategy involves a decision on whether to fuse maps or keep them independent. This fact inspired the name of our SLAM approach, the Selective Submap Joining SLAM (SSJS). If the region being revisited is small we keep the maps independent. Otherwise, if the region being revisited is large, the maps that are closing the loop are joined and fused into a single map in a similar way to the Map Joining algorithm [2]. The size of the region being revisited is defined by the number of features contained in this region. This number of features is predefined as a *threshold*, and its effect on the overall performance of the SSJS approach has been analysed in section IV, where we show experimental results.

### D. Local Maps Joining

Given two submaps  $\mathcal{M}_i$  and  $\mathcal{M}_j$  w.r.t. a common reference  $\mathcal{B}$ , they are first stored into a joint state vector, as in (2).  $J_1$  are the partial derivatives of the transformation that maps  $\mathcal{M}_j$  to  $\mathcal{M}_i$  with respect to the  $\mathcal{M}_i$ , while  $J_2$  is with respect to  $\mathcal{M}_j$ .

$$\mathbf{B}_{\mathbf{x}_{i+j}} = \begin{bmatrix} \mathbf{B}_{\mathbf{x}_i} \\ \mathbf{B}_{\mathbf{x}_j} \end{bmatrix} \quad \mathbf{P}_{i+j} = \begin{bmatrix} \mathbf{P}_i & \mathbf{P}_i \mathbf{J}_1^t \\ \mathbf{J}_1 \mathbf{P}_i & \mathbf{J}_1 \mathbf{P}_i \mathbf{J}_1^t + \mathbf{J}_2 \mathbf{P}_j \mathbf{J}_2^t \end{bmatrix} \quad (2)$$

The common landmarks from  $\mathcal{M}_i$  are the predictions (as in a standard EKF) and the common landmarks from  $\mathcal{M}_j$  are understood as new observations. Afterwards, the innovation vector and matrix are computed, followed by the EKF update stage. Together with the map fusion, the corresponding link in the global level is corrected. This correction is obtained directly from the map fusion since the links within the fused maps are correlated and updated with all the information. After deciding whether to fuse or not the maps, a new submap is built and the whole process is repeated again. Notice that if the decision is to fuse two maps, they become a single one.

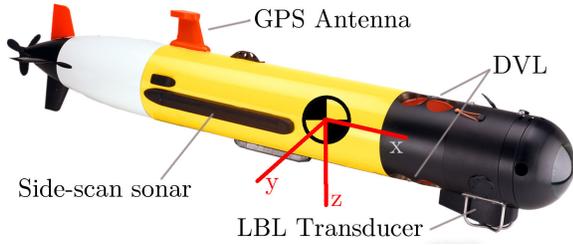


Fig. 2. REMUS 100 and some of the on board sensors. The vehicle coordinate frame is also illustrated in the image.

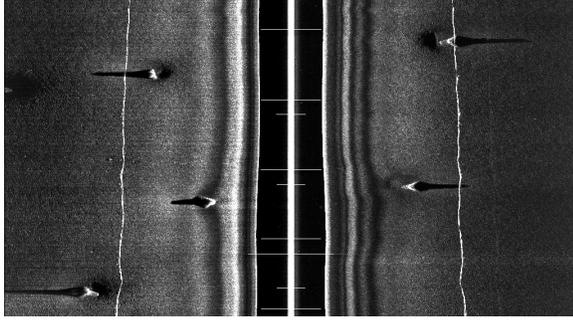


Fig. 3. Example of a side-scan sonar image with five salient features.

## IV. EXPERIMENTS AND RESULTS

### A. Vehicle Platform

The AUV REMUS-100 was used to gather the experimental data (see Fig. 2). The vehicle was equipped with a DVL and IMU, giving navigation data relative to the vehicle reference frame such as velocities, orientations and depth. In addition, the vehicle was carrying a side-scan sonar pointing both ways, starboard and port. From the navigation information provided by the sensors, the vehicle state can be defined by a 9-vector, composed of the 6-DOF vehicle's pose  $(x \ y \ z \ \phi \ \theta \ \psi)^t$  and the vehicle frame linear velocities  $(v_x \ v_y \ v_z)^t$ .

The map is composed of objects, rocks and other detectable features. The state of these features is defined as 3D points  $(x_{l_i} \ y_{l_i} \ z_{l_i})^t$ . Notice that the 3D point of an object represents the gravity centre of the object. These features are extracted from side-scan images (see Fig. 3). In addition to feature information, the side-scan sonar provides a measure of the altitude, i.e. the distance from the sensor to the seabed. Therefore, the joint state vector estimate  $\hat{\mathbf{x}}$  for our problem (3) contains both vehicle state and map information.

$$\hat{\mathbf{x}} = (x_V \ y_V \ z_V \ \phi_V \ \theta_V \ \psi_V \ v_x \ v_y \ v_z \dots \dots x_{l_1} \ y_{l_1} \ z_{l_1} \dots x_{l_n} \ y_{l_n} \ z_{l_n})^t \quad (3)$$

As mention in the introduction, current submapping techniques are used for terrestrial applications where a 3-DOF model applies. However, AUVs require a 6-DOF model, therefore we describe a 6-DOF motion and observation models in the following paragraphs. The motion model used here is a 6-DOF constant velocity kinematics model (4).

Where  ${}_{\mathbf{x}_k}^{\mathbf{x}_{k-1}}\mathcal{R}$  is the rotation matrix necessary to go from instant  $k-1$  to instant  $k$ .

$$\begin{bmatrix} x_k \\ y_k \\ z_k \\ \phi_k \\ \theta_k \\ \psi_k \\ v_{x_k} \\ v_{y_k} \\ v_{z_k} \end{bmatrix} = \begin{bmatrix} {}_{k-1}^k\mathcal{R} \begin{bmatrix} v_{x_{k-1}} dt \\ v_{y_{k-1}} dt \\ v_{z_{k-1}} dt \end{bmatrix} + \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ z_{k-1} \end{bmatrix} \\ \phi_{k-1} \\ \theta_{k-1} \\ \psi_{k-1} \\ v_{x_{k-1}} \\ v_{y_{k-1}} \\ v_{z_{k-1}} \end{bmatrix} \quad (4)$$

The motion model is expressed as a non-linear function  $\mathbf{x}_k = f(\mathbf{x}_{k-1})$ . The Jacobian of  $f$  is taken in order to linearise the model. This linearised function is used to predict the changes of the covariance matrix from time  $k-1$  to  $k$ . The observation model gives the predicted sensor pose from the last known position, and is represented by the function  $\hat{z}_k = h(\mathbf{x}_k)$ . This model can be expressed using matrices  $\mathbf{H}_k$  (5).

$$\mathbf{H}_k = \begin{bmatrix} \mathbf{H}_{k,o} \\ \mathbf{H}_{k,v} \\ \mathbf{H}_{k,d} \end{bmatrix} \quad (5)$$

Through the sensors of our system, we obtain measurements for the vehicle's orientation, linear speeds and the depth, the altitude (seabed's depth), and salient feature positions. The DVL and the IMU are not exactly at the origin of the body-fixed vehicle frame, but the software of the vehicle provides their measurements *w.r.t.* the vehicle's frame. The observation models are the ones shown in (6) for the orientations, (7) for the velocities, and (8) for the depth.

$$\mathbf{H}_{k,o} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (6)$$

$$\mathbf{H}_{k,v} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

$$\mathbf{H}_{k,d} = [0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0] \quad (8)$$

### B. Problem Definition

The experiments were conducted in a real environment. The AUV was sent underwater to perform a recognition mission. During the mission the vehicle navigated a large area of about 300m x 400m (see Fig. 4). The whole navigation consisted in a large number of loops, i.e. revisiting the same area several times. The vehicle's depth was almost constant around 12 meters, while the sea floor with respect to the water surface was oscillating around 16 meters. The sea floor was considerably flat, but with several salient objects. The total navigation time was of almost 4 hours. The experiment was conducted to gather the data, but not to run our algorithms on-line with the mission. The gathered data was post processed and run through our algorithms. The main objective of these post-computations were to analyse

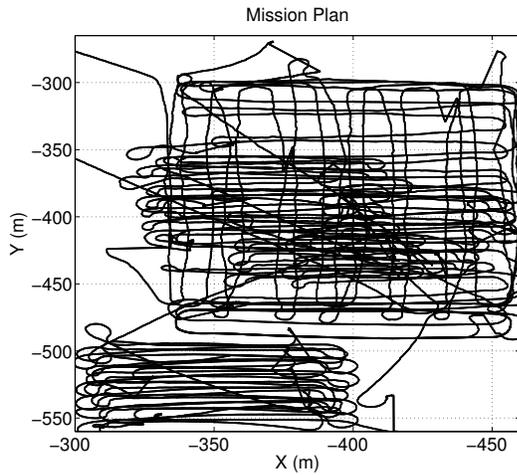


Fig. 4. Trajectory of the vehicle for the whole mission.

the performance of the algorithm, showing the consistency of our approach, and to demonstrate the improvement obtained by using submaps, analysing also the computational cost under different configurations.

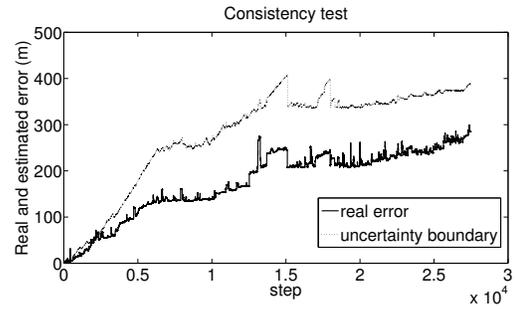
### C. Consistency Analysis

Map consistency is shown in Fig. 5(a). This figure shows that the discrepancy between LBL and SSJS is always kept inside the uncertainty boundaries, which means that the filter will not cause divergences due to overconfidence. Fig. 5(b) shows a clear example of an overconfident estimation of this discrepancy, which will lead to inconsistencies. This overconfidence appeared in simulations with larger submaps. This was an expected result, as with large submaps the approach tends to be a standard EKF. Working with very small submaps produces a similar overconfidence. The estimation of the error does not grow enough as compared to the discrepancy between LBL and SSJS. Therefore, it is necessary to choose a map size inbetween.

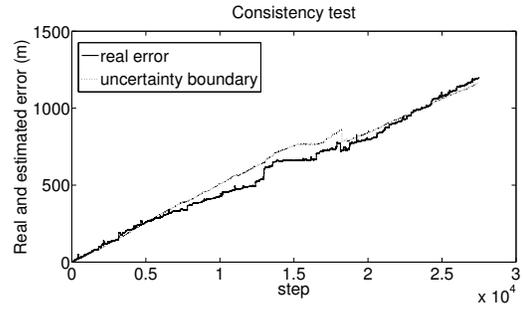
In order to illustrate the performance of our SSJS another interesting plot is shown in Fig. 6. This figure shows the outcome of the map fusion approach. The top part of this figure presents three local maps at a certain step on the experiment. Two maps that share several features can be fused. The bottom part of this figure shows the resulting map after the fusion, where the association between maps is properly solved, improving both maps and also correcting the localization of the vehicle.

### D. Computational Cost Analysis

The time required to complete the whole map was computed for all configurations. The mean, maximum and minimum times for a given number of features per map is plotted in Fig. 7. These three curves show an increase on the computational time on those cases with a number of features per submap higher than fifteen and lower than ten. Therefore, in terms of time demand it is interesting to start a new map once the old map contains between ten and fifteen landmarks.



(a)



(b)

Fig. 5. Example of a consistent configuration for our approach (a), where the discrepancy between LBL and SSJS (real error) is always lower than the uncertainty boundary (estimated error). When this condition is not kept (b), the algorithm is overconfident about its estimates and it may diverge in long term missions.

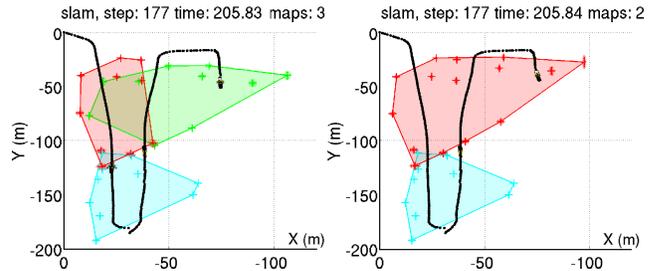


Fig. 6. Example of a map fusion step. In the left, three local maps have been built. The two upper maps share a 40% of their landmarks. In the right, these two maps have been fused.

There is a direct relationship between the ratio (number of correspondences between maps/number of features per map) and the time consumption. Figure 8 demonstrates this fact and shows a clear convergence to 0.5. Therefore, in terms of time consumption, a proper value for the number of correspondences between maps in order to be fused, should be half of the number of features per submap.

## V. CONCLUSIONS AND FUTURE WORKS

This paper has presented a novel approach for localization and mapping of autonomous underwater vehicles in unstructured environments with salient features. More specifically, the paper has presented the Selective Submap Joining SLAM algorithm (SSJS), the mathematical model for the REMUS-100, and a real experimental validation.

In our SSJS SLAM approach, the amount of information shared between maps is taken into account. This decision

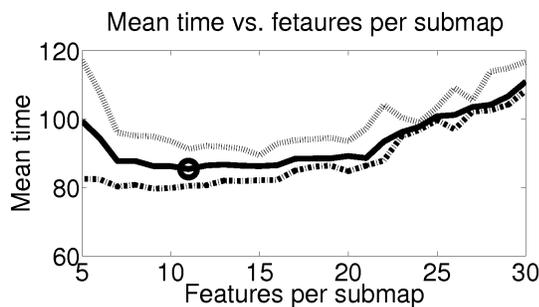


Fig. 7. Computational time required to fulfil the whole mission. The upper curve is the maximum time among all the experiments with a certain number of features per map. The lower curve is the minimum time, and the intermediate is the mean time. The cross shows the lowest time in the mean curve.

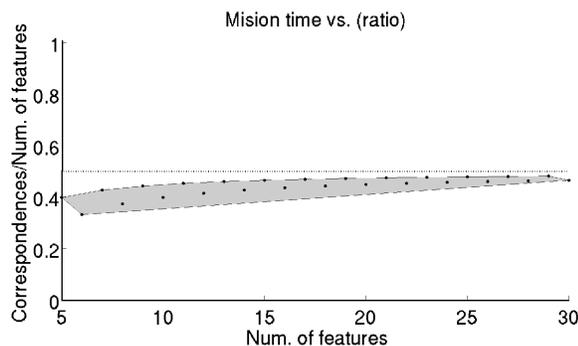


Fig. 8. The lowest mission times occur when the threshold for the number of correspondences is approximately 40% and 50% the number of features per submap. The region in grey corresponds to configurations that produced minimum mission time.

is made on the basis that fusing two maps that share many landmarks will produce a better update than fusing two maps that only share few landmarks. The experiments show a reduction of the effects of the linearisation error and also a more precise reconstruction of the map since the drift suffered in shorter distances is smaller and the data association can be more robustly solved. In addition, different parameters involved with the algorithm have been analysed, driving us to the conclusion that in order to obtain a good compromise between computational cost and map consistency, the best map size should be between ten and fifteen. Furthermore, the best performance was obtained when the threshold used to decide whether to fuse local maps was set near 50% of the total map size.

Future work is intended to integrate this method with vision sensors together with scan matching techniques. The proposed approach will be used as the module to localize the AUV position and to build an accurate 3D map of the seabed.

#### ACKNOWLEDGMENTS

The authors gratefully acknowledge the contribution of the Ocean Systems Lab members for its advices on how to acquire and process side-scan sonar data. The authors also

acknowledge the support of the research project DPI-2007-66796-C03-02 funded by the Spanish Ministry of Science and Innovation. J. Aulinas holds a University of Girona scholarship (BR-07/03).

#### REFERENCES

- [1] H. Durrant-Whyte and T. Bailey, *Simultaneous localization and mapping (SLAM): Part I*, IEEE Robotics and Automation Magazine, vol. 13, num. 2, pp 99–108, 2006.
- [2] J. D. Tardós, J. Neira, P. M. Newman, and J. J. Leonard, *Robust mapping and localization in indoor environments using sonar data*, International Journal on Robotics Research, vol. 21, num. 4, pp. 311–330, 2002.
- [3] L. M. Paz, J. D. Tardós, and J. Neira, *Divide and conquer: EKF SLAM in  $O(n)$* , IEEE Transaction in Robotics, vol. 24, num. 5, pp. 1107–1120, 2008.
- [4] C. Estrada, J. Neira, and J. D. Tardós, *Hierarchical SLAM: real-time accurate mapping of large environments*, IEEE Transaction in Robotics, vol. 21, num. 4, pp. 588–596, 2005.
- [5] P. Piniés and J. D. Tardós, *Scalable SLAM building conditionally independent local maps*, IEEE/RJS International Conference on Intelligent Robots and Systems, pp. 3466–3471, 2007.
- [6] M. Carreras, P. Ridaó, R. Garcia, and T. Nicosevici, *Vision-based localization of an underwater robot in a structured environment*, IEEE International Conference on Robotics and Automation, 2003. ICRA 03, vol. 1, pp. 971–976, 2003.
- [7] A. Caiti, A. Garulli, F. Livide, and D. Prattichizzo, *Localization of autonomous underwater vehicles by floating acoustic buoys: a set membership approach*, IEEE Journal of Oceanic Engineering, vol. 30, no. 1, pp. 140–152, Jan. 2005.
- [8] M. Erol, L. Vieira, and M. Gerla, *AUV-aided localization for underwater sensor networks*, International Conference on Wireless Algorithms, Systems and Applications (WASA 2007), pp. 44–54, Aug. 2007.
- [9] L.L. Whitcomb, D.R. Yoerger, and H. Singh, *Combined Doppler/LBL based navigation of underwater vehicles*, International Symposium on Unmanned Untethered Submersible Technology, Durham, New Hampshire, May 1999.
- [10] F. Maurelli, S. Krupinski, Y. Petillot, J. Salvi, *A Particle Filter Approach for AUV Localization*, MTS/IEEE OCEANS, OCEANS'08, Quebec City (Canada) September 15-18, 2008
- [11] S. Thrun, D. Burgard and W. Fox. *Probabilistic Robotics*, MIT Press, 2005. ISBN-10: 0-262-20162-3.
- [12] R. Smith, M. Self, and P. Cheeseman, *A Stochastic map for uncertain spatial relationships*, 4th International Symposium on Robotics Research, O. Faugeras and G.Giralt (Editors), the MIT Press, pp. 467–474, 1988.
- [13] J. E. Guivant and E. M. Nebot, *Optimization of the simultaneous localization and map-building algorithm for real-time implementation*, IEEE Transaction on Robotics and Automation, vol. 17, num. 3, pp. 242–257, 2001.
- [14] E. Wan and R. van der Merwe, *Kalman Filtering and Neural Networks, Chapter 7: The Unscented Kalman Filter*. Wiley, 2001. ISBN: 978-0-471-36998-1.
- [15] M.R. Walter, R.M. Eustice and J.J. Leonard, *Exactly sparse extended information filters for feature based SLAM*, International Journal of Robotics Research, vol.26, num. 4, pp 335–359, 2007.
- [16] J. J. Leonard and H. J. S. Feder, *A computationally efficient method for large-scale concurrent mapping and localization*, Robotics Research: The 9th International Symposium, D. Koditschek and J. Hollerbach (Editors), Springer Verlag, pp. 169–176, 2000.
- [17] S. B. Williams, G. Dissanayake, and H. Durrant-Whyte, *An efficient approach to the simultaneous localisation and mapping problem*, IEEE International Conference on Robotics and Automation, vol. 1, pp. 406–411, 2002.
- [18] P.M. Newman and J.J. Leonard, *Consistent, convergent, and constant-time SLAM*, International Joint Conference on Artificial Intelligence (IJCAD), pp. 1143–1150, 2003.
- [19] M. Bosse, P. M. Newman, J. J. Leonard, and S. Teller, *SLAM in large scale cyclic environments using the atlas framework*, International Journal in Robotics Research, vol. 23, num. 12, pp. 1113–1139, 2004.
- [20] J. Neira and J.D. Tardós, *Data Association in Stochastic Mapping Using the Joint Compatibility Test*, IEEE Transaction on Robotics and Automation, vol. 17, num. 6, pp. 890–897, Dec 2001.