Bathymetry-Based SLAM with Difference of Normals Point-Cloud Subsampling and Probabilistic ICP Registration*

Albert Palomer[1], Pere Ridao[1], David Ribas[1], Angelos Mallios[1], Nuno Gracias[1] and Guillem Vallicrosa[1]

***Abstract*—This paper describes a probabilistic surface matching method for pose-based bathymetry SLAM using a multibeam sonar profiler. The proposed algorithm compounds swath profiles of the seafloor with dead reckoning localization to build surface patches. Then, a probabilistic implementation of the ICP is used to deal with the uncertainty of the robot pose as well as the measured points in a two-stage process including point-to-point and point-to-plane metrics. A surface adaptation using octrees and difference of normals is proposed to have ICP-derived methods working in feature-poor or highly unstructured areas typical of bathymetric scenarios. Moreover, a heuristic based on the uncertainties of the surface points is used to improve the basic algorithm, decreasing the ICP complexity to $O(n)$. The performance of the method is demonstrated with real data from a bathymetric survey with Girona 500 AUV.**

I. INTRODUCTION

Navigation and mapping are two fundamental problems to achieve fully operational Autonomous Underwater Vehicles (AUVs). On-board navigation systems, based on dead reckoning, are known to suffer an unbounded drift even when sophisticated high-end inertial navigation systems are used [1]. Hence, it is common to aid these systems using absolute position fixes from GPS on surface, or coming from acoustic beacon/transponder networks when submerged [2]. State of the art methods include, for instance, Long Base Line (LBL) systems [3], where an on-board transceiver is used to interrogate a fixed number of transponders which are located at a priori known positions. The calibration cost may be reduced if single beacon/transponder methods are used instead [4]. At the cost of decreasing the accuracy, the calibration step may be even avoided if an inverted LBL [5] or Ultra Short Base Line (USBL) [6] is used.

The above mentioned methods have the problem of confining the robot to the limited area of coverage of the acoustic network. Terrain-Based Navigation (TBN) methods [7] have the potential to overcome such a limitation when an a priori digital terrain map of the area is available. On the other hand, a truly autonomous robot should be able to localize itself within the environment using its own sensors without any external infrastructure or previous knowledge of the area. This is fully summarized in the concept of Simultaneous Localization and Mapping (SLAM). After many years of research, there exist a number of algorithms proposed to solve the SLAM problem, with notable achievements mostly in land mobile robotics [8]. Comparatively, there is still limited research done underwater, mostly because of the physical characteristics of the water and the environment complexity.

From the sensors vantage point, we can classify underwater SLAM in two main categories: sonar and vision based SLAM. Although cameras provide a large amount of high-resolution information with fast refresh rate, their use underwater is limited by the visibility conditions of the medium. Recent years have witnessed a number of successful, real world underwater vision implementations of the SLAM problem [9], [10], [11], [12]. In comparison with vision, sonar sensors may work in bad visibility conditions, being able to penetrate deeper ($\sim$150 m) because of the low attenuation of sound in water. However, they provide limited information and medium to low resolution and refresh rate. Although there are still few works in sonar-based underwater SLAM, they are promising. There have been reported several feature-based methods [13], [14], [15], [16], [17], [18]. However, in a natural underwater environment, the appearance of the features change dramatically depending on the point of view which makes it extremely difficult to extract them robustly. For this reason, some researchers have focused their efforts on using featureless methods like occupancy grids and scan matching. In [19], a SLAM algorithm based on a particle filter (PF) was used to build an occupancy grid of a sinkhole with the range measurements from multiple pencil-beam sonars. The method uses a smart representation of the environment based on octree structures, saving space and reducing the computation time. While bathymetry (elevation) 2.5D maps have been extensively used in the context of TBN [7], there exist few works reporting successful SLAM implementations using a multibeam sonar profiler and bathymetric maps. The pioneering work corresponds to [20] who used cross correlation and Iterative Closest Point (ICP) for coarse and fine registration of bathymetric surface patches respectively. Recently, [21] has reported the BPSLAM, a method inspired in the DPSLAM from [22], which uses a PF as in [19] but representing the environment as an elevation map distributed across the ancestry of a given particle. In [23] it has been reported MBpIC-SLAM (based on [20]). This uses a probabilistic ICP for matching, an octree surface adaptation for dealing with areas with poor information, and a heuristic based on the uncertainties of the points in the patches to reduce the complexity of the ICP to $O(n)$.

The method used here for the SLAM is an evolution of the one in [23], which is an extension to 2.5D of our previously reported 2D SLAM framework [24]. The framework presented here employs the same point-to-point and point-to-plane metrics for coarse and fine registration respectively, but proposes a new surface adaptation method based on [25] and octrees which is used to deal with the featureless areas typically found in bathymetric maps. As said before, the method also uses a heuristic based on the uncertainties of the surface points to

*This work was supported by the Spanish project DPI2011-27977-C03-02 (COMAROB) and three European Commission's Seventh Framework Program 2011-2015 projects: FP7-ICT-2011-7-288704 (MORPH), FP7-INFRASTRUCTURES-2012-312762 (EUROFLEETS2) and Marie Curie PERG-GA-2010-276778 (Surf3DSLAM).

[1]All authors are with the Department of Computer Engineering, Universitat de Girona, 17071 Girona, Spain apalomer|pere|dribas|mallios|ngracias at eia.udg.edu, guillem.vallicrosa at udg.edu

978-1-4799-0002-2/13/$31.00 ©2013 IEEE

improve the performance of the algorithm by reducing the ICP complexity to $O(n)$.

The rest of the paper is organized as follows; first the registration algorithm is outlined in Section II, then the SLAM algorithm is presented in Section III followed by the supporting experimental results and conclusions in Sections IV and V.

II. Registration Algorithm

The proposed method is a probabilistic ICP algorithm using point-to-point/point-to-plane association for registering surface patches. The algorithm receives as input two surface patches (a reference surface $\mathbf{S}_{ref}$ and a new gathered surface $\mathbf{S}_{new}$) as well as an initial guess $\mathbf{q}_0 = \mathbf{N}(\hat{\mathbf{q}}_0, \mathbf{P}_{q_0})$ of the 6 degrees of freedom (DOF) displacement between them. In this section it is explained how the stochastic point cloud representations of the patches are extracted as well as the method used for their registration providing an estimate of the the real displacement $\mathbf{q}_{min}$ which minimizes the Mahalanobis distance of the association error.

A. Forming a Surface Patch

Because multibeam sonar profilers can only produce 2D swaths transversal to the vehicle motion, it is necessary to compound them with an estimate of the vehicle trajectory to build a 2.5D surface patch. In our approach, a 6 DOF dead-reckoning estimate of the trajectory is obtained by means of an Extended Kalman Filter (EKF) which merges the predictions from a simple constant velocity kinematic model with the updates from the onboard navigation sensors: a Doppler Velocity Log (DVL), a pressure sensor and an Attitude and Heading Reference System (AHRS). Given that, the process of compounding the measurements with the trajectory is straightforward: Let $\boldsymbol{\delta}_i = \mathbf{N}(\hat{\boldsymbol{\delta}}_{\boldsymbol{i}}, \mathbf{P}_{\boldsymbol{\delta}_\mathbf{i}})$ be one of the polar range measurements that compose a particular sonar swath, and $\mathbf{x}_i = \mathbf{N}(\hat{\mathbf{x}}_i, \mathbf{P}_{\mathbf{x}_i})$ be the vehicle position at the time which the swath was acquired. Then, the position of one point $\mathbf{p_i} = \mathbf{N}(\hat{\mathbf{p}}_\mathbf{i}, \mathbf{P}_{\mathbf{p_i}})$ of the surface patch can be obtained as:

$$\hat{\mathbf{p}}_i = \hat{\mathbf{x}}_i \oplus g(\hat{\boldsymbol{\delta}}_{\boldsymbol{i}}) \quad (1)$$

$$\mathbf{P}_{\mathbf{p_i}} = \mathbf{J}_{1\oplus}\mathbf{P}_{\mathbf{x}_i}\mathbf{J}_{1\oplus}^T + \mathbf{J}_{2\oplus}(\mathbf{J}_g\mathbf{P}_{\delta_i}\mathbf{J}_g^T)\mathbf{J}_{2\oplus}^T \quad (2)$$

where $g(.)$ is the polar to Cartesian conversion function, $\mathbf{J}_g$ is its corresponding Jacobian and $\mathbf{J}_{1\oplus}$ and $\mathbf{J}_{2\oplus}$ are the Jacobians of the compounding function $\oplus$ as defined in [26]. It is worth mentioning that after finishing a surface patch all the $\mathbf{p_i}$ points are referenced to a new frame $\{I\}$ chosen to be coincident with the vehicle pose in the middle of the sequence previously used to build the patch. Choosing this reference frame instead of the initial pose, produces a more convenient uncertainty distribution among the points that form the patch [27].

The different patches are created according to the size and the variance of the 2.5D surfaces. Whenever a new patch is being generated, its variance in depth is checked. If the patch grows in variance, it means that a rich non-flat terrain is being observed. After observing a growing variance during a successive number of data additions (e.g. 200 times), and after ensuring that a minimum patch size has been reached (30m in length), the resulting cloud of points is stored and a new patch is started. Unfortunately, the presence of areas with lack of distinctive features will result in large patches with an important accumulation of internal error. For this reason, a second condition has been established to close the patch when it reaches a maximum size ($\sim$ 80m in length).

B. Patch Registration

To make the registration problem tractable from the computational point of view, only a subset of the surface points are used. As a consequence, the points are more scattered and traditional point-to-point association may produce inaccurate results. On the other hand, we have observed that although point-to-plane registration is often cheated by noisy data and it is not able to correct large displacements, it can improve the registration results when applied after an initial correction done with the point-to-point approach.

1) Point to Point Association: Given a certain point $\mathbf{n_i} = \mathbf{N}(\hat{\mathbf{n}}_\mathbf{i}, \mathbf{P}_{\mathbf{n_i}})$ of a new patch S_{new}, a matching candidate point $\mathbf{r_k} = \mathbf{N}(\hat{\mathbf{r}}_\mathbf{k}, \mathbf{P}_{\mathbf{r_k}})$ of a previously existing reference surface S_{ref} (see Section II-C2 for insight on the selection of candidades), and an initial guess for the displacement between the two patches $\mathbf{q_0} = (\hat{\mathbf{q}}_\mathbf{0}, \mathbf{P}_{\mathbf{q_0}})$ obtained from the estimated dead-reckoning trajectory, the association error $\mathbf{e_{ik}} = \mathbf{N}(\hat{\mathbf{e}}_\mathbf{ik}, \mathbf{P}_{\mathbf{e_{ik}}})$ can be obtained as:

$$\hat{\mathbf{e}}_\mathbf{ik} = \hat{\mathbf{r}}_\mathbf{k} - \hat{\mathbf{q}}_\mathbf{0} \oplus \hat{\mathbf{n}}_\mathbf{i} \quad (3)$$

$$\mathbf{P}_\mathbf{ik} = \mathbf{P}_{\mathbf{r_k}} + \mathbf{J}_{1\oplus}\mathbf{P}_{\mathbf{q_0}}\mathbf{J}_{1\oplus}^T + \mathbf{J}_{2\oplus}\mathbf{P}_{\mathbf{n_i}}\mathbf{J}_{2\oplus}^T \quad (4)$$

The matching candidate is accepted if it passes an individual compatibility test using the Mahalanobis distance:

$$D^2 = \hat{\mathbf{e}}_\mathbf{ik}^T\mathbf{P}_\mathbf{ik}^{-\mathbf{1}}\hat{\mathbf{e}}_\mathbf{ik} < \chi_{d,\alpha}^2 \quad (5)$$

where the acceptance threshold is set by a chi-square distribution with $d = dim(\mathbf{e_{ik}})$ and the confidence level α. Given a point in the S_{ref}, all the possible candidates in S_{new} are tested with the objective of building a set A of compatible points. The point in A with the minimum Mahalanobis distance is chosen for the point-to-point association, while the full set will be later used in the point-to-plane approach.

2) Point to Plane Association: On this stage, a set A_i of $\mathbf{r_k}$ points in S_{ref} which are compatible with a particular point $\mathbf{n_i}$ in S_{new} is used to define a local plane $\Pi(\boldsymbol{\nu}_\mathbf{i}, \mathbf{d_i})$, with $\boldsymbol{\nu}_\mathbf{i}$ being its normal vector and $\mathbf{d_i}$ its distance to the origin. Because of the probabilistic nature of the proposed algorithm, it is necessary to estimate the plane parameters but also its uncertainty. In [28], the authors present a method for estimating planes where the error of a set of samples is minimized using the uncertainty related to the sensor range by means of a weighted Principal Component Analysis (PCA). In our implementation, the samples are the $\mathbf{r_k}$ points represented by their mean and covariance. Minimizing the plane error taking into consideration such uncertainty representation resulted in a problem too complex to be solved analytically. To be able to solve the problem as efficiently as in [28], each point uncertainty was approximated by the largest eigenvalue of its covariance

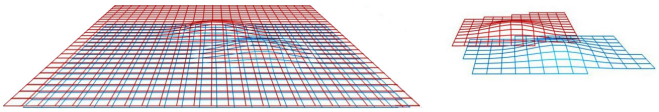


Fig. 1. A 3D visual concept of the idea behind octree use: two surfaces to be matched (left); the same surfaces resampled (right).

matrix. In this way, the error ellipsoid is approximated to a sphere, making it possible to solve the method analytically. The minimization uses Lagrange multiplier to fit the best plane. For a more detailed derivation, please refer to the original work.

The outcome of this plane estimation process are the parameters $\hat{\boldsymbol{\nu}}_\mathbf{i}$ and $\hat{\mathbf{d}}_i$ and their associated covariance matrices $\mathbf{P}_{\nu_i}$ and $\mathbf{P}_{d_i}$. Given this, a new point $\mathbf{a_i}$ which is the orthogonal projection of $\mathbf{n_i}$ over the plane can be defined as:

$$\hat{\mathbf{a}}_\mathbf{i} = \hat{\mathbf{q}}_\mathbf{0} \oplus \hat{\mathbf{n}}_\mathbf{i} - ((\hat{\mathbf{q}}_\mathbf{0} \oplus \hat{\mathbf{n}}_\mathbf{i})^T \hat{\boldsymbol{\nu}}_\mathbf{i} - \hat{\mathbf{d}}_\mathbf{i})\hat{\boldsymbol{\nu}}_\mathbf{i} \quad (6)$$

$$\mathbf{P}_{\mathbf{a_i}} = \frac{\partial \hat{\mathbf{a}}_\mathbf{i}}{\partial \hat{\mathbf{q}}_\mathbf{0}} \mathbf{P}_{q_0} \frac{\partial \hat{\mathbf{a}}_\mathbf{i}}{\partial \hat{\mathbf{q}}_\mathbf{0}}^T + \frac{\partial \hat{\mathbf{a}}_\mathbf{i}}{\partial \hat{\mathbf{n}}_\mathbf{i}} \mathbf{P}_{n_i} \frac{\partial \hat{\mathbf{a}}_\mathbf{i}}{\partial \hat{\mathbf{n}}_\mathbf{i}}^T + \frac{\partial \hat{\mathbf{a}}_\mathbf{i}}{\partial \hat{\boldsymbol{\nu}}_\mathbf{i}} \mathbf{P}_{\nu_{ik}} \frac{\partial \hat{\mathbf{a}}_\mathbf{i}}{\partial \hat{\boldsymbol{\nu}}_\mathbf{i}}^T + \frac{\partial \hat{\mathbf{a}}_\mathbf{i}}{\partial \hat{\mathbf{d}}_\mathbf{i}} \mathbf{P}_{d_{ik}} \frac{\partial \hat{\mathbf{a}}_\mathbf{i}}{\partial \hat{\mathbf{d}}_\mathbf{i}}^T \quad (7)$$

This new point $\mathbf{a_i}$, which is virtual and not existing in S_{ref}, makes possible to calculate the point-to-plane error with the same point-to-point procedure described in eq. 3 to 4 but using $\mathbf{a_i}$ instead of $\mathbf{r_k}$.

3) Minimization: After each association stage, a minimization process is carried out to estimate a displacement $\mathbf{q_{min}}$ which minimizes the addition of the Mahalanobis distance of the association error:

$$\mathbf{q_{min}} = argmin_q \sum (\boldsymbol{\epsilon}^T \mathbf{P}_{\boldsymbol{\epsilon}}^{-1} \boldsymbol{\epsilon}) \quad (8)$$

Being $\boldsymbol{\epsilon}$ a vector composed of all the $\hat{\mathbf{e}}_\mathbf{ik}$ error vectors calculated after associating all the points and $\mathbf{P}_{\boldsymbol{\epsilon}}$ the block diagonal matrix with their corresponding covariances. This minimization is done using a stochastic least squares:

$$\mathbf{q_{min}} = [\mathbf{J}^T \mathbf{P}_{\boldsymbol{\epsilon}}^{-1} \mathbf{J}]^{-1} \mathbf{J}^T \mathbf{P}_{\boldsymbol{\epsilon}}^{-1} \boldsymbol{\epsilon} \quad (9)$$

being $\mathbf{J}$ the Jacobian matrix of the error function at the previous estimation evaluated in all the points.

C. Scan Adaptation

This section discusses some issues related with the use of ICP-based algorithms and proposes two methods which improve the performance of our approach.

1) Adaptive Point Sampling: In a scenario like the one in Fig. 1 (left), where two almost flat surfaces share a poorly visible feature, traditional ICP-based methods may encounter some problems. First, ICP tends to associate each point with its closest neighbor according to a particular metric. For that reason it may be difficult to correctly associate the feature areas when the displacement is large (i.e. they are far from each other, and the proximity of flat areas may lead to a local minimum). This particular issue may benefit from the probabilistic ICP approach, since a more accurate description of the error will indicate that the two features are candidates for the matching.

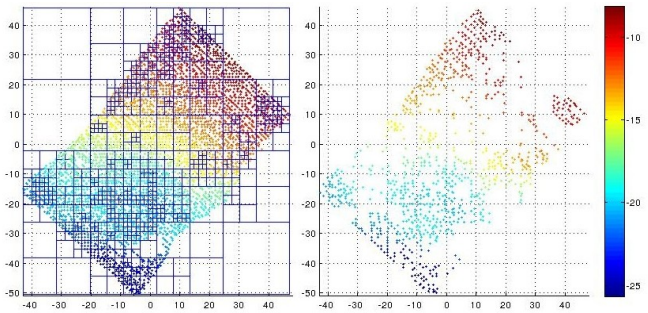


Fig. 2. Topview of the octree (left) and resampled points (right) using the DoN resampling method.

A second issue is that flat areas have a weak contribution to the success of the registration (they all look similar and there are many matching possibilities). Moreover, in scenarios with few features, flat areas may prevail and lead to poor matchings. We believe that ICP algorithms have higher potential when the used points are significant. For that reason, we propose to use a smart sampling over the surface which reduces the size of the point cloud by removing those points which are less informative (see Fig. 1 (right)). Since the surface distribution is not available, the points need to be sampled in a discrete way. An octree structure is used to sample the scan in its most significant areas. An area is considered significant when it has a rich relief.

In previous works [23], the criteria used for resampling a set of points was related with their vertical distance to the average depth of the patch. This criteria works well for patches obtained in a mainly planar seabed, where areas with significant height differences correspond with relevant features. However, in the presence of inclined terrains, the method may wrongly classify smooth areas that are far from the average plane just because of the slope and not because a rich relief.

The weak point of the original approach was using as a reference the average depth of the patch, which actually defines a horizontal plane, to analyze the relief of the patches that may or may not be horizontal. For this reason we propose a new solution using difference of normals (DoN) [25] which has been proved to work properly in the segmentation of unstructured point clouds. The criteria does a principal component analysis (PCA) of two set of points: the one corresponding to the whole patch and the one of the points contained in a particular cell of the octree. Then, the PCA returns 3 directions and their 3 variances. The two directions with the largest variances determine the plane that fits the point cloud, while the remaining direction corresponds to the normal of that plane. Once both normals are computed (the one for all the patch $\overrightarrow{n_p}$ and the one for the cell $\overrightarrow{n_c}$), DoN can be computed as:

$$\overrightarrow{\triangle n} = \overrightarrow{n_p} - \overrightarrow{n_c} \quad (10)$$

if the module of $\overrightarrow{\triangle n}$ is grater than a threshold, which means

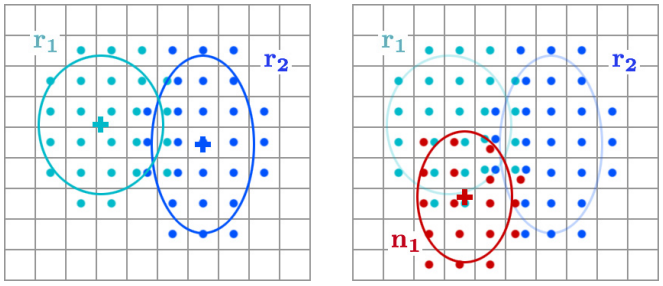


Fig. 3. Use of a support grid during the association time: ellipsoids for the points $\mathbf{r}_1$ and $\mathbf{r}_2$ in $\mathbf{S}_{ref}$ (left); both $\mathbf{r}_1$ and $\mathbf{r}_2$ are candidates for a point $\mathbf{n}_1$ in $\mathbf{S}_{new}$ (right).

that $\overrightarrow{n_p}$ and $\overrightarrow{n_c}$ are significantly different, the cell of the octree is considered that contains relevant data and, therefore, will be divided to resample more points from that region.

Fig 2 (left) shows an example of the octree subdivision applied to a test patch from real data where the size of each cell is inversely proportional to the richness of the area. During the resampling phase a single point is taken for each cell. In this way, less significant areas (bigger cells) produce less points, while significant ones (smaller cells) produce more (see Fig. 2 (right)). This resampling not only improves the chances for a successful matching, even in the presence of large displacements, but also decreases drastically the number of points involved in the registration which greatly increases the performance of the algorithm. In Fig. 2 the number of points has been reduced to 918 from an original patch of 4031.

2) Association in Linear Time: Determining correspondences between the points in two surfaces has a high computational cost for ICP-based methods. This is mainly caused because of the large number of different matrix operations, including inversions, which leads to an $O(n^2)$ algorithm. Hereafter, a method for reducing complexity by considering the uncertainty estimates of the points, which are already available, is proposed.

The method basically takes advantage of the fact that a Gaussian random point $\mathbf{p}_i$ can be graphically represented as an ellipsoid for a desired confidence level α, and that there will be overlapping between the ellipsoids of two points deemed compatible after a Mahalanobis distance test. Therefore, instead of checking correspondences among all the points in a patch, the search can be limited to those points whose ellipsoids overlap. This reduction on the space of exploration decreases the complexity of the algorithm to $O(n)$. The process can be easily understood by following the schematic in Fig. 3. First, a 3D grid is generated for the $\mathbf{S}_{ref}$ patch. For each point $\mathbf{r_k}$ in $\mathbf{S}_{ref}$, the principal axes of the $\mathbf{P}_{r_k}$ uncertainty ellipsoid are obtained for a given confidence level and then, employed to determine the occupied volume within the grid by placing tokens inside the overlapping cells (see Fig. 3 (left)). Then, a similar procedure is repeated for each point $\mathbf{n_i}$ in the second patch $\mathbf{S}_{new}$ (see Fig. 3 (right)). As it can be seen, when a new ellipsoid is laid, it may happen that some of the cells already contain markers from points in the $\mathbf{S}_{ref}$. This situation determines the overlapping of the ellipsoids and therefore, those points will be the ones considered for the Mahalanobis compatibility test.

III. SLAM ALGORITHM

The proposed framework is a pose-based SLAM which uses an EKF as the core of the stochastic process. The state vector $\mathbf{x}$ stores the global 6 DOF vehicle positions $\{\mathbf{x}_1 \dots \mathbf{x}_n^B\}$ corresponding with the positions previously chosen as local reference frames $\{I\}$ during the generation of the patches:

$$\hat{\mathbf{x}}_k = \left[\hat{\mathbf{x}}_n^T \ \dots \ \hat{\mathbf{x}}_1^T\right]_k^T \quad (11)$$

$$\mathbf{P}_{x_k} = E([\mathbf{x}_k - \hat{\mathbf{x}}_k][\mathbf{x}_k - \hat{\mathbf{x}}_k]^T) \quad (12)$$

A. Prediction Step

Each time a new patch is completed, a new pose is introduced in the state vector $\mathbf{x}$. The new pose is obtained by compounding the previous vehicle pose $\mathbf{x}_n$ with the dead-reckoning displacement $\mathbf{q}_k$ estimated during the generation of the current patch:

$$\hat{\mathbf{x}}_{k+1}^+ = \left[\left(\hat{\mathbf{x}}_n \oplus \hat{\mathbf{q}}_k\right)^T \quad \hat{\mathbf{x}}_n^T \ \dots \ \hat{\mathbf{x}}_1^T\right]^T \quad (13)$$

$$\mathbf{P}_{x_{k+1}}^+ = \mathbf{F}_{k+1}\mathbf{P}_{x_k}\mathbf{F}_{k+1}^T + \mathbf{G}_{k+1}\mathbf{P}_{q_k}\mathbf{G}_{k+1}^T \quad (14)$$

with

$$\mathbf{F}_{k+1} = \begin{bmatrix} \mathbf{J}_{1\oplus} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}; \qquad \mathbf{G}_k = \begin{bmatrix} \mathbf{J}_{2\oplus} \\ \mathbf{0} \end{bmatrix} \quad (15)$$

B. Matching Strategy

Whenever a newly created patch is available, it is compared with all the previously created patches looking for matchings. This search is accomplished by determining intersections between the polygons delimiting the coverage areas of the patches (the polygons are calculated after the completion of each patch). A new patch may potentially overlap with several patches which may, or may not be consecutive in time (see for instance patch number 13 in Fig. 4). As it can be also observed, consecutive patches (such as number 1 and 2, or 8 and 9) may have a small overlapping with the new patch. For this reason, we propose to join consecutive patches in order to maximize the overlapping. It is worth mentioning that this is only recommendable for patches which are consecutive in time, since contiguous non-consecutive patches may suffer large drift errors (e.g. patches number 1 and 6). Therefore, the process proposed here has three steps: First, a search for overlapping patches given a newly introduced patch. Second, a search for consecutive patches among the previously selected and finally, the construction of joint surface patches with those which are found to be consecutive. The joint surface is the result of combining all the points in the two patches and representing them in a common $\{I\}$ reference frame. In the proposed approach, the reference frame of the earliest created patch is chosen as the new frame for the joint patch.

C. Scan Matching

Given two overlapping scans S_i and S_n with related poses $\mathbf{x}_i$ and $\mathbf{x}_n$, an initial guess of their relative displacement $\mathbf{q}_{in_0}$ is necessary. This can be easily extracted from the state vector

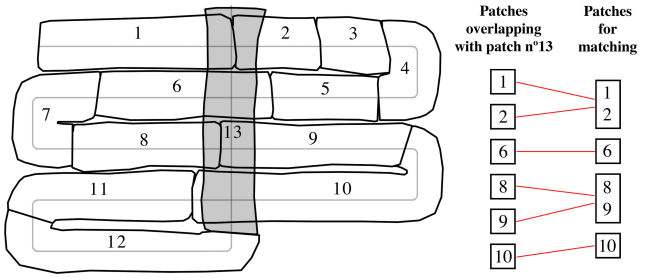


Fig. 4. The surface patch number 13 is overlapping with the previously existing patches numbered as 1, 2, 6, 8, 9 and 10. To improve the matching process, consecutive patches (1 and 2 but also 8 and 9) are joined together. As a result, four patches will take part in the matching.

using the tail-to-tail transformation:

$$\hat{\mathbf{q}}_{in_0} = \ominus\hat{\mathbf{x}}_i \oplus \hat{\mathbf{x}}_n \tag{16}$$

Since the tail-to-tail transformation is actually a nonlinear function of the full state vector $\mathbf{x}$, the uncertainty of the initial guess can be computed by means of its Jacobian:

$$\mathbf{P}_{\mathbf{q}_{in_0}} = \mathbf{J}\mathbf{P}\mathbf{J}^T \tag{17}$$

where

$$\mathbf{J} = \left.\frac{\partial(\ominus\hat{\mathbf{x}}_i \oplus \hat{\mathbf{x}}_n)}{\partial\,\mathbf{x}}\right|_{(\mathbf{x}=\hat{\mathbf{x}})} = [\,\mathbf{J}_{2\oplus}\cdots\mathbf{0}\cdots\mathbf{J}_{1\oplus}\mathbf{J}_{\ominus}\cdots\mathbf{0}\,] \tag{18}$$

After determining the initial guess of the relative displacement $\mathbf{q}_{in_0}$, it is feed to the registration algorithm together with the two patches (S_i and S_n) to produce a refined estimate of the displacement $\mathbf{q}_{in}$.

D. State Update

The displacement described in (16) defines a constraint between two vehicle poses in the state vector which can be expressed by means of the following measurement model:

$$\mathbf{z}_k = h(\mathbf{x}_k, \mathbf{v}_k) = \ominus\mathbf{x}_{i_k} \oplus \mathbf{x}_{n_k} + \mathbf{v}_k; \tag{19}$$

where the measurement $\mathbf{z}_k$ corresponds to the estimated displacement $\hat{\mathbf{q}}_{in}$ and $\mathbf{v}_k$ is a zero-mean white Gaussian noise with covariance $\mathbf{P}_{q_{in}}$ which accounts for the errors in the scan matching process. Given this, the state can be updated with the standard EKF update equations.

IV. Experimental Results

For testing the new scan adaptation algorithm, a real experimental dataset acquired in an area with a sloped seafloor has been used. The experiment was performed with the Girona 500 AUV [29] (Fig. 5) in a sunken volcano at the Santorini island during the Caldera 2012 Eurofleets cruise. The mission consisted on the execution of a bathymetric survey over a lava tongue of geologic interest found at depths between 280 and 330 m and covering an area of approximately 230 x 300 m. The vehicle navigated at an altitude of 15 m over the seabed and, in addition to the measurements from a multibeam echosounder, the dataset also included navigation data from the on-board DVL, AHRS and pressure sensor. Since the purpose of the experiment was to cover a large area in the minimum time, the resulting survey does not have much overlapping between consecutive transects and only the beginning transversal track patch allows for loop closures.

Fig. 6 shows the final elevation map of the area, while Fig. 7 shows the different patches generated to serve as input for the scan-matching algorithm. As it can be seen, the scenario presents a slope of approximately 14%, with some depressions along the tongue of lava. The patches have a very small lateral overlapping and only those corresponding to the transversal transect will allow the crossovers to clearly cover previously visited areas. Because of that limitation in the dataset, we observe small changes in the resulting map after applying the SLAM framework. For that reason, and to demonstrate the consistency of our method, we have proposed a second test in which noise is artificially introduced in the initial patch positions to slightly increase the misalignments in the original map.

The maps in Fig. 8 were generated by rendering the resulting point clouds over a grid of 1m resolution and measuring the standard deviation for each cell. Fig. 8(a) and 8(b) show the initial maps obtained after compounding the multibeam data over the original dead reckoning trajectory and after the addition of noise respectively. As it can be seen, the bigger errors are present in the top part of the map (red spots), coinciding with the last part of the experiment where the robot accumulated more uncertainty, and on the overlapping parts of due to roll error. Given the predominance of errors with a standard deviation around and below 1 m (green to blue areas), it is difficult to assess improvements after the application of the SLAM frameworks by just observing the red spots. For that reason, we have included numerical results to better evaluate the performance of the methods.

Table I shows the numerical results for the two experiments. In the original dataset, after the SLAM is executed, not much improvement can be appreciated, indeed, Fig. 8(a) and Fig. 8(d) are practically the same. This improvement is, in terms of the mean of the standard deviation in each grid cell, 1.20% for the method using the distance to average patch depth and 1.59% for the proposed method using DoN. In terms of the total addition of standard deviations in all the cells, the improvement is 1.20% and 1.68% respectively. In both metrics

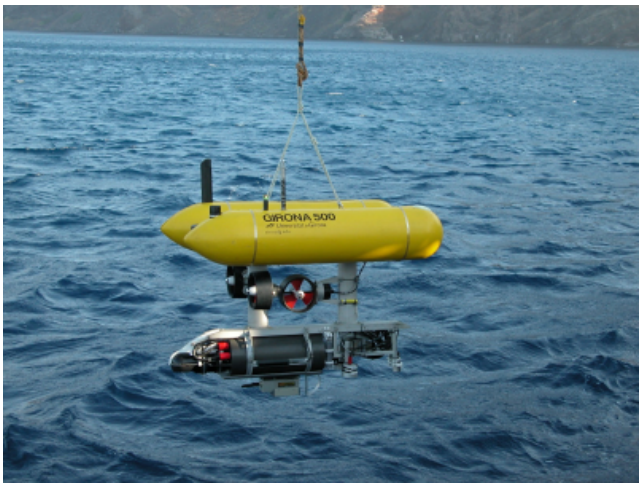
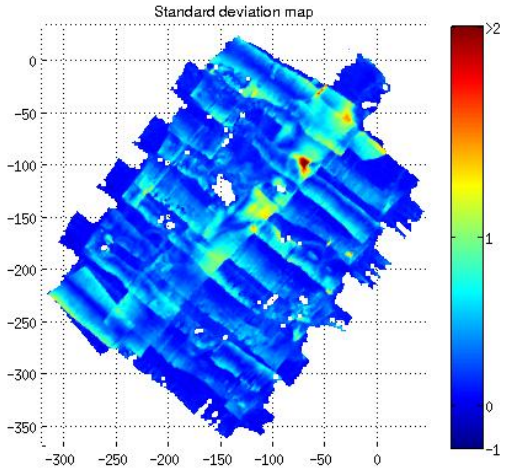
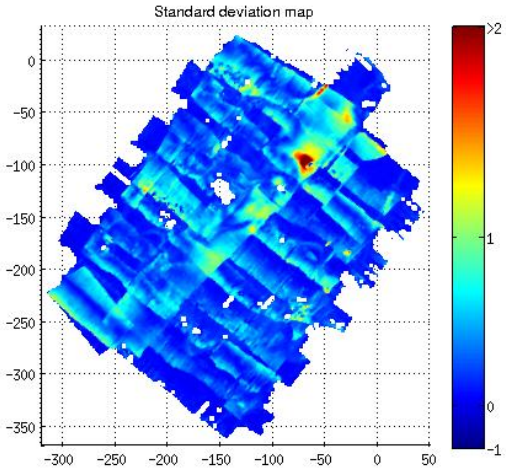


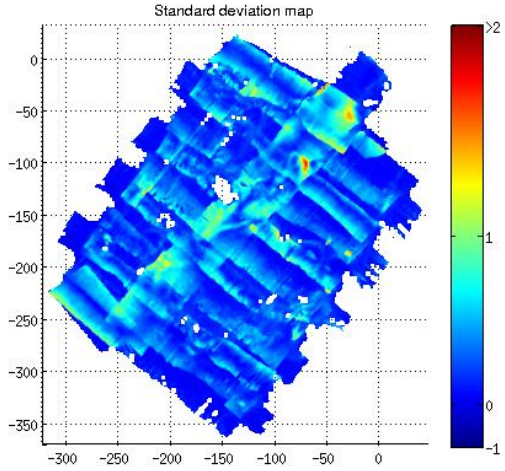
Fig. 5. GIRONA500 AUV being deployed in Greek waters



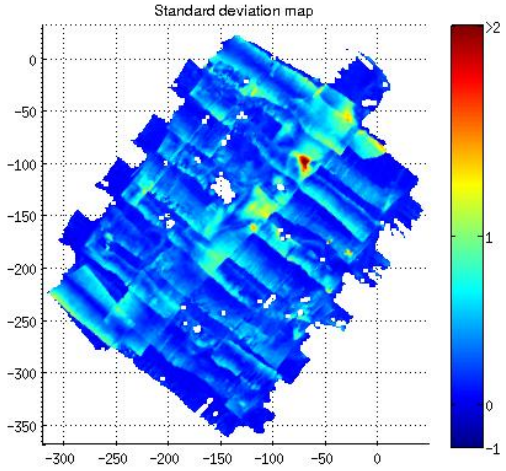
(a) Dead reckoning



(b) Dead reckoning with noise to patch positions



(c) SLAM with distance to average depth



(d) SLAM with DoN

Fig. 8. Standard deviation map (in meters) of the scanned area.

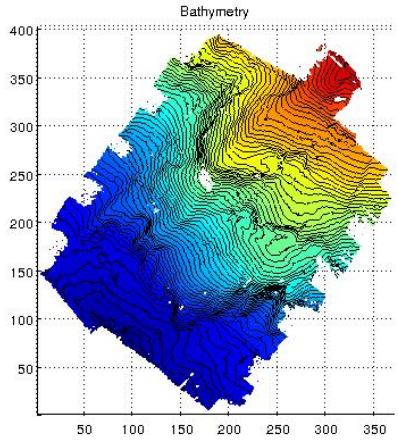


Fig. 6. Corrected bathymetry rendered in 1m cells.

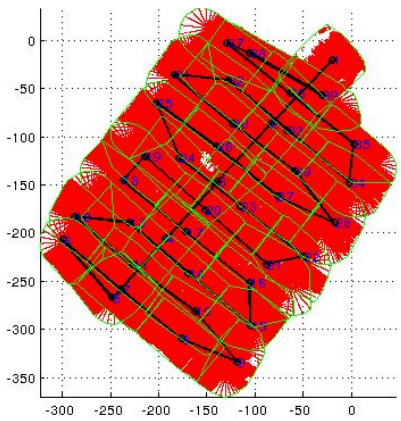


Fig. 7. Patches subdivision over the points. In black the robot trajectory, in red the multibeam points and in green the patches division.

Surface adaptation		Experiment	
		Original	Noise
Dead reckoning	Sum of std	19589.368478	20343.946626
	Mean of std	0,247097	0,257173
Distance to average depth	Sum of std	19354.091382	19354.091382
	improvment	1.20%	4.87%
	Mean of std	0.244129	0.244129
	improvment	1.20%	5.07%
DoN	Sum of std	19260.870833	19260.870833
	improvment	1.68%	5.32%
	Mean of std	0.243168	0.243168
	improvment	1.59%	5.45%

TABLE I. RESULTS STATISTICS SUMMARY

the improvement is bigger with the new approach than the one using the criteria from previous work [23].

The results for the second experiment with the added noise can be seen on the right column of table I. Both surface adaptation methods show exactly the same results already observed for the first experiment. Since the resampling of the surfaces is not subject to random decisions, and noise is only added at the patch positions and not internally, this results are basically the consequence of the ICP matching process converging to the same solution. Again, the results obtained using DoN show an improvement when compared with the previous method.

V. CONCLUSIONS AND FUTURE WORK

This paper presented a SLAM algorithm which takes advantage of a probabilistic framework for dealing with the subdivision of the mapped surface into smaller patches while taking into consideration the motion uncertainty during their formation. The main contribution of this work is a new method for the adaptive subsampling of the sensor data that improves the quality of the results obtained with a previously proposed method [23].

Preliminary tests have been presented to evaluate the proposed method with a dataset obtained in a submerged lava tongue. The small overlapping between the different transects and the relatively good dead-reckoning estimate in the experiment makes it difficult to observe large corrections in the map. However, the results demonstrate some improvement with the new approach, and given the intrinsic operation of the method, we believe that this approach could be better suited for dealing with different typologies of seafloor than the original method.

Future work will include intensive testing and comparison of the different sampling strategies in order to assess their true potential. The analysis will include the use of synthetic data which will make possible to establish a ground truth against which the methods will be compared. Also, future works will need to deal with the internal errors in the patches. Whenever a patch is matched against another, only a rigid transformation that minimizes the matching error is found, but no work has been done to refine the patch internally to remove the errors introduced by the dead-reckoning process.

REFERENCES

[1] R. Panish and M. Taylor, "Achieving high navigation accuracy using inertial navigation systems in autonomous underwater vehicles," *OCEANS 2011 IEEE - Spain*, pp. 1–7, Jun. 2011. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6003517

[2] K. Gade, "Navlab, a generic simulation and postprocessing tool for navigation," *Modeling, identification and control*, pp. 135–150, 2005.

[3] J. C. Kinsey and L. L. Whitcomb, "Preliminary field experience with the DVLNAV integrated navigation system for oceanographic submersibles," *Control Engineering Practice*, vol. 12, no. 12, pp. 1541–1549, Dec. 2004. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0967066103002843

[4] P. Batista, C. Silvestre, and P. Oliveira, "Single Beacon Navigation: Observability Analysis and Filter Design," *American Control Conference (ACC)*, pp. 6191–6196, 2010.

[5] H. G. Thomas, "GIB Buoys: An Interface Between Space and Depths of the Oceans," *Autonomous underwater vehicles, 1998. AUV'98. Proceedings of the 1998 workshop on1*, pp. 181–184, 1998.

[6] M. Mandt, K. Gade, and B. Jalving, "Integrateing DGPS-USBL position measurements with inertial navigation in the HUGIN 3000 AUV," *Proceedings from the 8th Saint Petersburg international Conference on Integrated Navigation Systems*, 2001.

[7] S. Carreno, P. Wilson, P. Ridao, and Y. Petillot, "A survey on Terrain Based Navigation for AUVs," *Oceans 2010 Mts/Ieee Seattle*, pp. 1–7, Sep. 2010. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5664372

[8] T. Bailey and H. Durrant-whyte, "Simultaneous Localization and Mapping (SLAM): Part II," *Robotics Automation Magazine, IEEE*, vol. 13, pp. 108–117, 2006.

[9] R. Eustice, O. Pizarro, and H. Singh, "Visually Augmented Navigation in an Unstructured Environment Using a Delayed State History," *Proceedings of the 2004 IEEE. International Conference on Robotics and Automation*, no. April, pp. 25–32, 2004.

[10] S. Williams and I. Mahon, "Simultaneous Localisation and Mapping on the Great Barrier Reef," *Robotics and Automation, 2004, Proceedings. ICRA'04. 2004 IEEE International Converence on*, vol. 2, pp. 1771–1776, 2004.

[11] R. Eustice, H. Singh, J. Leonard, M. Walter, and R. Ballard, "Visually navigating the RMS Titanic with SLAM information filters," *Proceedings of the Robotics Science and Systems*, 2005.

[12] M. Johnson-roberson, O. Pizarro, S. B. Williams, and I. Mahon, "Generation and Visualization of Large-Scale Three-Dimensional Reconstructions from Underwater Robotic Surveys," *Journal of Field Robotics*, vol. 27, no. 1, pp. 21–51, 2010.

[13] R. N. Carpenter, "Concurrent Mapping and Localization with FLS," *Autonomous underwater vehicles, 1998. AUV'98. Proceedings of the 1998 workshop on1*, vol. 02841, no. 401, pp. 133–148, 1998.

[14] J. Leonard, R. N. Carpenter, and H. Feder, "Stochastic mapping using forward look sonar," *Robotica*, vol. 19, pp. 467–480, 2001.

[15] I. Ruiz, "Enhaced concurrent mapping and localisation mapping algorithm," 2001.

[16] S. Williams, P. Newman, J. Rosenbaltt, G. Dissanayake, and H. Durrant-whyte, "Autonomous underwater navigation and control," *Robotica2*, vol. 19, pp. 481–496, 2001.

[17] P. Newman and J. Leonard, "Pure range-only sub-sea SLAM," *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, pp. 1921–1926, 2003. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1241875

[18] D. Ribas, P. Ridao, J. D. Domingo, and J. Neira, "Underwater SLAM in Man-Made Structured Environments," *Journal of Field Robotics*, vol. 25, pp. 898–921, 2008.

[19] N. Fairfield, D. Jonak, G. Kantor, and D. Wettergreen, "Field results of the control, navigation, and mapping systems of a hovering AUV," *Proceedings of the 15th International Symposium on Unmanned Untethered Submersible Technology*, 2007.

[20] C. Roman and H. Singh, "A Self-Consistent Bathymetric Mapping Algorithm," *Journal of Field Robotics*, vol. 24, pp. 23–50, 2007.

[21] S. Barkby, S. Williams, O. Pizarro, and M. Jakuba, "An Efficient

Approach to Bathymetric SLAM," *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International conference on*, pp. 219–224, 2009.

[22] A. Eliazar and R. Parr, "DP-SLAM: Fast, robust simultaneous localization and mapping without predetermined landmarks," *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 18, pp. 1135–1142, 2003.

[23] S. Zandara, P. Ridao, D. Ribas, A. Mallios, and A. Palomer, "Probabilistic Surface Matching for Bathymetry Based SLAM," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2013.

[24] A. Mallios, P. Ridao, D. Ribas, F. Maurelli, and Y. Petillot, "EKF-SLAM for AUV navigation under probabilistic sonar scan-matching," *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4404–4411, Oct. 2010. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5649246

[25] Y. Ioannou, B. Taati, R. Harrap, and M. Greenspan, "Difference of Normals as a Multi-Scale Operator in Unorganized Point Clouds," *ArXiv e-prints*, 2012.

[26] R. Smith, M. Self, and P. Cheeseman, "Estmating uncertain spatial relationships in robotics," *Autonomous robot vehicles*, vol. 1, pp. 167–193, 1990.

[27] A. Burguera, Y. González, and G. Oliver, "A Probabilistic Framework for Sonar Scan Matching Localization," *Advanced Robotics*, vol. 22, pp. 1223–1241, 2008.

[28] K. Pathak, N. Vaskevicius, and A. Birk, "Revisiting uncertainty analysis for optimum planes extracted from 3D range sensor point-clouds," *2009 IEEE International Conference on Robotics and Automation*, no. 3, pp. 1631–1636, May 2009. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5152502

[29] D. Ribas, N. Palomeras, P. Ridao, M. Carreras, and A. Mallios, "Girona 500 AUV: From Survey to Intervention," *IEEE/ASME Transactions on Mechatronics*, vol. 17, no. 1, pp. 46–53, Feb. 2012. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6093749