Underwater SLAM in Man-Made Structured Environments

David Ribas and Pere Ridao
Grup de Visió per Computador i Robòtica
Universitat de Girona
Girona 17071, Spain
e-mail: dribas@eia.udg.es, pere@eia.udg.es

Juan Domingo Tardós and José Neira
Grupo de Robótica y Tiempo Real
Universidad de Zaragoza
Zaragoza 50018, Spain
e-mail: tardos@unizar.es, jneira@unizar.es

Received 9 January 2008; accepted 30 May 2008

This paper describes a navigation system for autonomous underwater vehicles (AUVs) in partially structured environments, such as dams, harbors, marinas, and marine platforms. A mechanically scanned imaging sonar is used to obtain information about the location of vertical planar structures present in such environments. A robust voting algorithm has been developed to extract line features, together with their uncertainty, from the continuous sonar data flow. The obtained information is incorporated into a feature-based simultaneous localization and mapping (SLAM) algorithm running an extended Kalman filter. Simultaneously, the AUV's position estimate is provided to the feature extraction algorithm to correct the distortions that the vehicle motion produces in the acoustic images. Moreover, a procedure to build and maintain a sequence of local maps and to posteriorly recover the full global map has been adapted for the application presented. Experiments carried out in a marina located in the Costa Brava (Spain) with the Ictineu AUV show the viability of the proposed approach.

1. INTRODUCTION

Simultaneous localization and mapping (SLAM) is one of the fundamental problems that needs to be solved before achieving truly autonomous vehicles. For this reason, in recent years it has been the focus of a great deal of attention (see Bailey & Durrant-Whyte, 2006; Durrant-Whyte & Bailey, 2006, and references therein). Multiple techniques have shown promising results in a variety of different applications and scenarios. Some of them perform SLAM indoors (Castellanos, Montiel, Neira, & Tardós, 1999), outdoors (Guivant, Nebot, & Durrant-Whyte, 2000), and even on air (Kim & Sukkarieh, 2003). However, the underwater environment is still one of the most challenging scenarios for SLAM because of the reduced sensorial possibilities and the difficulty in finding reliable features. Acoustic devices are the most common choice. Many approaches extract features from acoustic data produced by imaging sonars (Leonard, Carpenter, & Feder, 2001; Tena, Petillot, Lane, & Salson, 2001) or sidescan sonars (Tena, Reed, Petillot,

Journal of Field Robotics, 1–24 (2008) © 2008 Wiley Periodicals, Inc.
Published online in Wiley InterScience (www.interscience.wiley.com). • DOI: 10.1002/rob.20249

Bell, & Lane, 2003). Some introduce artificial beacons to deal with complex environments (Newman & Leonard, 2003; Williams, Newman, Rosenblatt, Dissanayake, & Durrant-Whyte, 2001). An alternative approach is analyzing the three-dimensional (3D) structure of the environment to perform SLAM, either using bathymetric scans (Roman & Singh, 2005) or producing a tessellated representation of the scenario (Fairfield, Jonak, Kantor, & Wettergreen, 2007). The use of cameras is limited to applications in which the vehicle navigates in clear water and very near to the seafloor (Eustice, Singh, Leonard, Walter, & Ballard, 2005). On the other hand, the visual information can also be combined with acoustic data to improve the overall reliability of the SLAM system (Williams & Mahon, 2004).

This article focuses on underwater SLAM applied to man-made environments (harbors, marinas, marine platforms, dams, etc.) where structures composed of vertical planes are present and produce reliable features in acoustic images. Although most of the previous work done in this field focuses on open sea and coastal applications, obtaining an accurate positioning in such scenarios would notably increase autonomous underwater vehicle (AUV) capabilities. For instance, an AUV could use a harbor as an outpost for oceanography research if it is able to localize itself and navigate through it with enough accuracy to safely perform the leaving and returning operations (Griffiths, McPhail, Rogers, & Meldrum, 1998). Maintenance and inspection of underwater structures (Martins, Matos, Cruz, & Pereira, 1999) and even surveillance of marine installations are examples of other applications that can benefit from such a system.

We have chosen an extended Kalman filter (EKF)–based implementation of the stochastic map to perform SLAM. The algorithm relies on a mechanically scanned imaging sonar (MSIS) for feature extraction. Although these mechanically actuated devices usually have a low scanning rate, they are quite popular because of their reduced cost. When working with MSIS, it is not uncommon to assume that the robot remains static or moves slowly enough to neglect the induced acoustic image distortion. Here, the static assumption is removed. Vehicle position estimates from the filter are introduced to reduce the effects of motion-induced distortion in the resulting acoustic data. Typically, approaches using imaging sonars have focused on the use of point features (Tena et al., 2001; Williams et al., 2001). This work proposes the use of line features in underwater environments as a representation of the cross sections produced when a sonar scan intersects with existing planar structures. As a result, a two-dimensional (2D) map of the environment is produced by the SLAM algorithm. Most of the applications at hand involve only trajectories performed at a constant depth, and therefore, a 2D map is sufficient for navigation. Moreover, it is generally viable to apply the present method when 3D motion is also required because the predominance of vertical structures in the target scenarios makes the 2D map valid for different operating depths.

In recent years, many different authors have proposed methods to carry out SLAM by building sequences of local maps (Bosse et al., 2003; Bosse, Newman, Leonard, & Teller, 2004; Clemente, Davison, Reid, Neira, & Tardós, 2007; Estrada, Neira, & Tardós, 2005; Leonard & Feder, 2001; Leonard & Newman, 2003; Newman, Leonard, & Rikoski, 2003; Ni, Steedly, & Dellaert, 2007; Tardós, Neira, Newman, & Leonard, 2002; Williams, Dissanayake, & Durrant-Whyte, 2002). The main objective of such techniques is limiting the cost of updating the full covariance matrix, which has $O(n^2)$ complexity (Guivant & Nebot, 2001). By working on local maps of limited size, the cost of these algorithms remains constant most of the time. In this work we consider conditionally independent local maps (Piniés & Tardós, 2007) because they allow sharing vital information between consecutive maps (in this case the underwater vehicle state). Strong empirical evidence also suggests that the use of local maps also improves the consistency of EKF-based SLAM algorithms (Castellanos, Neira, & Tardós, 2004; Huang & Dissanayake, 2007).

A data set obtained during an experiment performed with the Ictineu AUV serves as a test for the proposed SLAM algorithm. The vehicle, a low-cost research platform of reduced dimensions developed at the Underwater Robotics Laboratory of the University of Girona (Ribas, Palomer, Ridao, Carreras, & Hernàndez, 2007), performed a 600-m trajectory in an abandoned marina situated in the Spanish Costa Brava, near St. Pere Pescador (see Figure 1). The resulting data set includes measurements from many different sensors. A Tritech Miniking MSIS provides the acoustic imagery that feeds the feature extraction algorithm. The velocity measurements from a SonTek Argonaut Doppler velocity log (DVL) are introduced in the filter to perform dead reckoning, and a compass and a pressure sensor provide absolute

Figure 1. The Ictineu AUV (left) and an abandoned marina used as test bed for the presented SLAM algorithm situated near St. Pere Pescador, Spain (right).

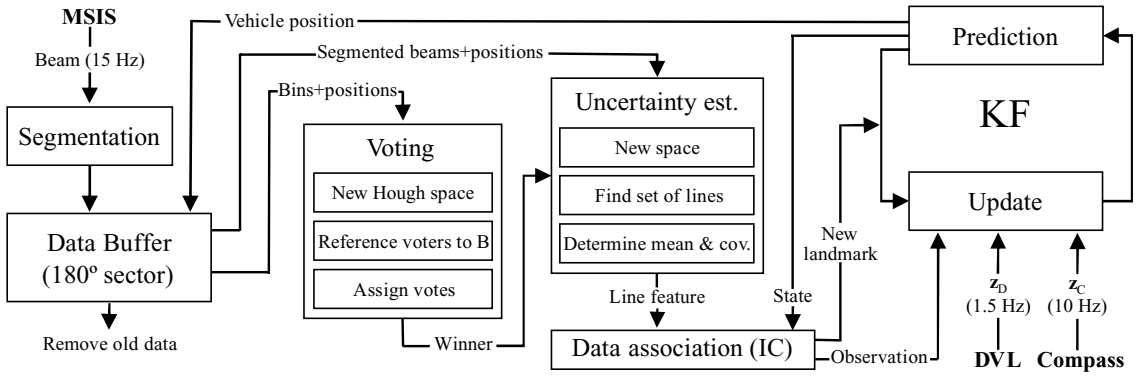

Figure 2. Diagram of the proposed SLAM approach.

measurements for the vehicle heading and depth. A diagram of the SLAM system can be seen in Figure 2.

This article is organized as follows. Sections 2 and 3 present a feature extraction algorithm that searches for line features in the data arriving continuously from the MSIS, removing the effect of motion-induced distortions, and estimates the feature uncertainty from representation in the acoustic image (a preliminary version appeared first in Ribas, Neira, Ridao, & Tardós, 2006). The SLAM algorithm in Section 4 incorporates a method that allows integrating compass measurements independently from the orientation of the map base reference. This has opened the door to implement a method to build local maps for improving scalability and accuracy as described in Section 5. The results obtained with the abandoned marina data set are presented in Section 6. Finally, the conclusions and further work are given in Section 7.

2. WORKING WITH ACOUSTIC IMAGES FROM A MSIS

MSISs perform scans in a 2D plane by rotating a fan-shaped sonar beam through a series of small-angle steps (Figure 3). For each emitted beam, an echo intensity profile is returned from the environment

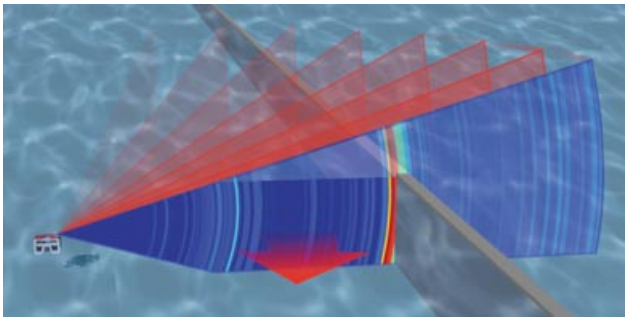

Figure 3. Representation of a mechanically scanned imaging sonar mode of operation. Beams are emitted sequentially at given transducer head angles (fan-shaped silhouettes). For each beam, the echo intensity return profile is discretized in a sequence of bins (arcs at different ranges along the most recently emitted beam).

and discretized into a set of bins (distance vs. echo-amplitude values). Accumulating this information along a complete 360-deg sector produces an acoustic image of the surroundings [see polar representation of a 360-deg scan sector in Figure 4(a) and its corresponding Cartesian representation in Figure 4(c)]. The beam typically has a large vertical beamwidth (for our sensor, about 40 deg), which makes possible the detection of obstacles at different heights, although at the cost of inducing small imprecisions in the range measurements. On the other hand, a narrow horizontal beamwidth (about 3 deg) increments the resolution of the device and improves the sharpness of the acoustic images. It is worth noting that although the sensor can detect the presence of tridimensional objects, it is not able to determine their position in the vertical plane. Therefore, only a 2D representation of the environment is produced.

2.1. Motion-Induced Distortions

Commonly, MSISs have a slow scanning rate. A Tritech Miniking sonar head needs a minimum time of 6 s to complete a 360-deg scan. Depending on the range and resolution settings, the time can be much more (e.g., during the abandoned marina experiment, the required time was 15 s). Therefore, during the acquisition of a scan, the vehicle position can change considerably. Not taking this motion into account may induce important distortions in the resulting acoustic data. Figure 4(c) represents this effect for a set of measurements obtained in the scenario shown in Figure 4(b). Note the misplacement of the walls on the left and top of the scan and how the rotation of the vehicle curves the one on the right. Extracting line features from distorted data is difficult and produces inaccuracies that disturb data association and, as a consequence, yields poor results. Therefore, the first step of the procedure consists of merging the raw sensor data with information regarding the vehicle motion. This information is provided by the SLAM algorithm (see Section 4), which runs simultaneously with the feature extraction algorithm. Incorporating the displacements and rotations of the sensor into the positional information of each sonar measurement leads to an undistorted acoustic image such the one in Figure 4(d). As can be appreciated when comparing with Figure 4(b), this results in a better representation of the real scenario.

2.2. Beam Segmentation

Because objects present in the environment appear as high-echo-amplitude returns in acoustic images, only part of the information stored in each beam is useful for feature extraction [see Figure 5(a)]. Therefore, a segmentation process can be done in order to obtain the most significant information. This process is carried out beam to beam and consists of two steps. First, only those bins with an intensity value over a threshold are selected and stored. This procedure separates the acoustic imprint left by an object in the image from the noisy background data [Figure 5(b)]. The resulting imprint is used to estimate the feature uncertainty, as explained in Section 3.2. The second step is to select among the beam's thresholded data those bins that are local maxima and satisfy a "minimum distance between them" criterion. This means that if two or more of these bins are too close within the beam, they should correspond to the detection of the same object and hence are redundant. Then, the ones with the lowest intensity value are discarded [see the result in Figure 5(c)]. The selected local high-intensity bins are the ones that most likely correspond to objects present in the scene. Thus, they are specially well suited as input to the feature extraction algorithm (Section 3) while, at the same time, the computational efficiency is improved because a small number of bins is involved.

2.3. Dealing with a Stream of Beams

To deal with the stream of measurements produced by the continuous arrival of beams, a data buffer is

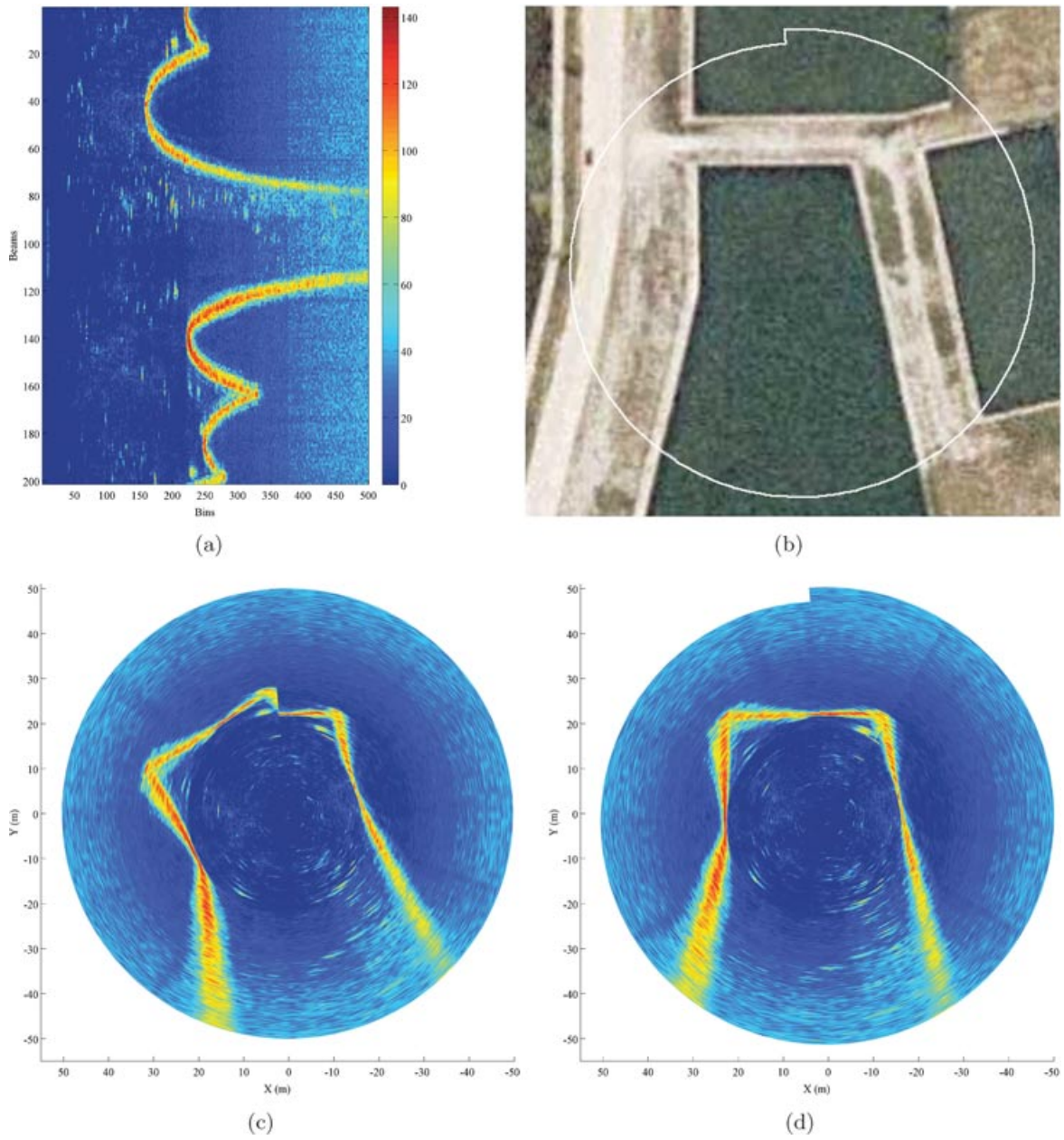

Figure 4. Effect of motion-induced distortion on acoustic images: (a) Raw data represented in polar coordinates. (b) Ortophotomap of the real environment where the sonar data were gathered. (c) The same data set represented in Cartesian coordinates. (d) Undistorted image obtained after taking into account the vehicle motion.

set to store the beams contained within the most recent 180-deg scan sector. Whenever new beams corresponding to an unexplored zone arrive, old beams that fall outside the scan sector are discarded. The choice of a 180-deg sector is not arbitrary because this is the maximum zone that a single line can cover within a sonar scan. Because calculations to search for features are performed with every new beam (Section 3), the buffer should contain the bins that are local maxima [Figure 5(c)] for the line detection process, the segmented beams [Figure 5(b)] for uncertainty estimation, and all its associated positions in

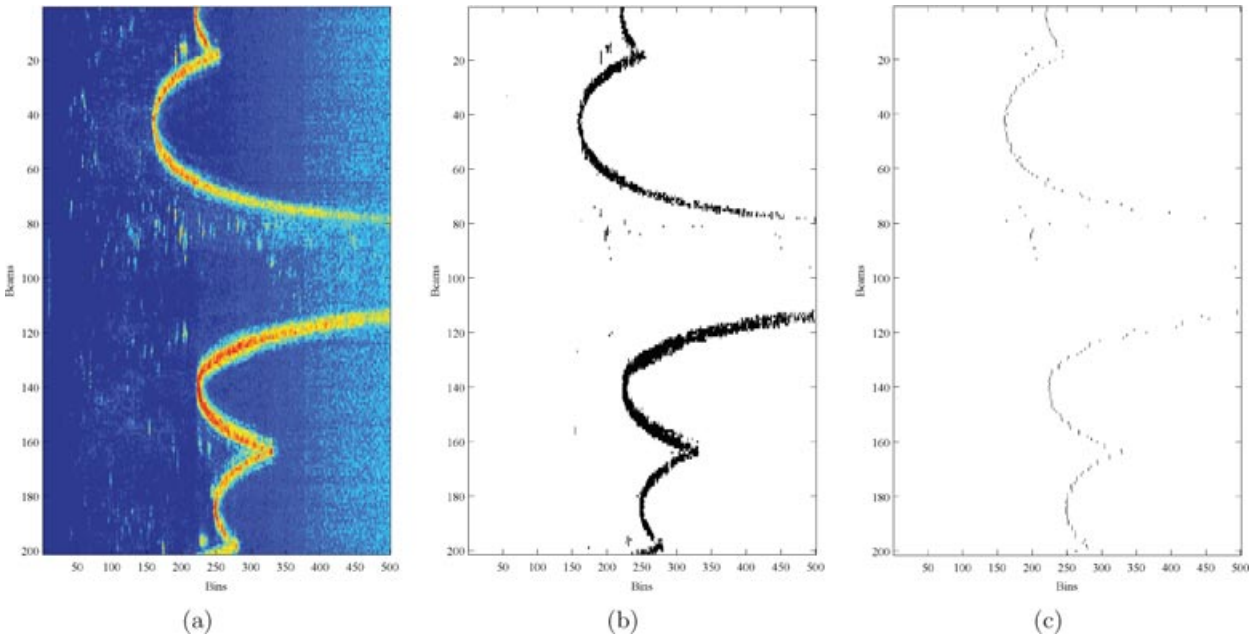

Figure 5. Acoustic data segmentation: (a) Raw sensor data represented in polar coordinates. (b) Data after applying a threshold. (c) Selection of the local maxima bins of each beam.

the world coordinate system to deal with the motion-induced distortions.

3. LINE FEATURE EXTRACTION

The cross section of a sonar scan with walls and other planar structures results in line-shaped features in the acoustic images. The Hough transform (Illingworth & Kittler, 1988) is a feature extraction technique that is specially well suited for this kind of situation. This algorithm accumulates the information from the sensor data into a voting table that is a parameterized representation of all the possible feature locations. Those features that receive a great number of votes are the ones with a relevant set of compatible sensor measurements and thus the ones that most likely correspond to a real object in the environment. In our application, line features are described by two parameters, ρ^B and θ^B (distance and orientation with respect to a base frame B). Hence, the resulting Hough space (HS) is a 2D space where the voting process and the search for maxima can be done efficiently. The base reference frame B can be set arbitrarily. However, our choice for B is the current position of the sensor head at the moment the voting is performed. Because in this implementation the voting is triggered by the arrival of new beams from the sensor, the most recently stored position in the data buffer (the one corresponding to the last beam) defines the position of B. An advantage of choosing this base is that, when a line feature is detected after the voting, its parameters are already represented in the sensor coordinate frame and hence it can be integrated directly into the SLAM framework. It is worth noting that B is not a fixed coordinate frame. As the parameterization in the HS is performed in polar coordinates, setting the reference in a fixed position would produce resolution loss with the increment in range. To avoid this, we need to resituate B according to the vehicle's motion. Unfortunately, this requires recomputing the HS with each change in the position of B. Although it may seem a great deal of computation, in fact the number of bins involved in the voting is not large (fewer than 100 bins during the tests performed) and the calculations can be executed quite rapidly. Moreover, as will be explained in the next section, there are situations in which recalculating the HS can be avoided. Another key issue is the quantization of the HS. In our case, we have observed that selecting the quantization equal to the angular and linear resolutions of our sensor (typically, 1.8 deg and 0.1 m) works fine. A higher resolution does not

necessarily increase the quality of the detection because the sonar resolution limits its precision, On the other hand, a lower resolution would produce a rough observation.

The general execution of the feature extraction process consists of several steps. First, with each beam arrival, the HS is initialized using the current sensor position as the base frame B. Next, all the bins stored in the buffer are referenced to B so that they can be used to vote in the space. It is worth noting that the stored beam positions are taken into account when transforming to B. Hence, the data are undistorted. Then, the votes corresponding to each bin are assigned to the candidate lines by means of a sonar model. Finally, a search for winning candidates is performed.

3.1. Voting Method for Line Detection

Each bin represents the strength of the echo intensity return in a particular place within the insonified area. Owing to the uncertainty produced by the horizontal beamwidth, a measurement cannot be assigned to a single point in the space. A common approach (Leonard & Durrant-Whyte, 1992; Tardós et al., 2002) is to consider the measurement as an arc whose aperture represents the beamwidth uncertainty. Moreover, as a high-intensity return is typically produced when the acoustic wave hits a surface perpendicularly, we can infer that all the surfaces tangent to the arc can explain the high-intensity return. Although this simple model is well suited for air sonar ranging systems, it is not able to explain the acoustic images gathered with a MSIS. A careful analysis of these images reveals that their object detection capability is not limited to the arc-tangent surfaces but that those beams intersecting the surface within the limits defined by a certain maximum incidence angle also produce a discernible return. On the other hand, those beams with a shallower angle are completely reflected and do not perceive the surface. To obtain a better description of this situation, an extended model of the imaging sonar has been adopted (Figure 6). Basically, given a horizontal beamwidth angle α (in our sensor, $\alpha = 3$ deg) and a maximum incidence angle β (generally, more than 60 deg), the set of line features compatible with a particular bin is composed not only of the lines tangent to the arc defined by α but also of all the lines that intersect the arc with an incidence angle smaller than $\pm\beta$. Before performing a voting, this set of lines must be determined for each bin stored in the data buffer. This process will now be described using as reference the illustration in Figure 6. Let the reference frame S define the

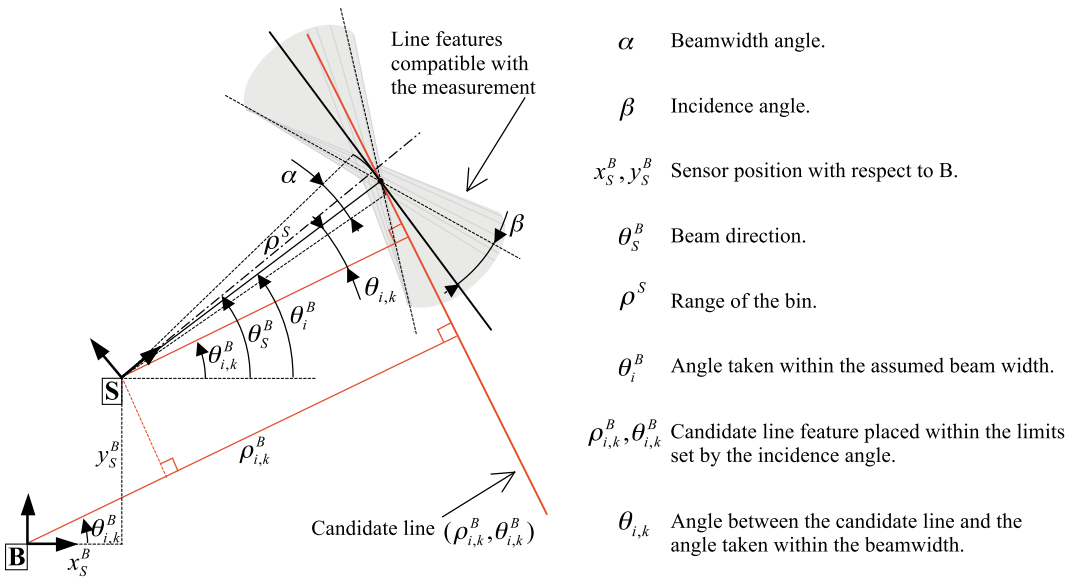

Figure 6. Model of the sonar sensor relating a bin and its compatible candidate lines. B is the base reference frame, and S is a reference frame attached to the sonar transducer head.

position of the transducer head at the moment a particular bin was obtained, with $[x_S^B, y_S^B, \theta_S^B]$ being the transformation that defines the position of S with respect to the chosen base reference B and ρ^S the range at which the bin was measured from the sensor. Both the transformation and the range values can be obtained from the information in the data buffer. To emulate the effect of the horizontal beamwidth, a set of i values are taken at a given resolution within an aperture of $\pm\alpha/2$ around the direction in which the transducer is oriented, also referred as θ_S^B:

$$\theta_S^B - \alpha/2 \leq \theta_i^B \leq \theta_S^B + \alpha/2. \quad (1)$$

Each value of θ_i^B represents the bearing parameter for a line tangent with the arc that models the horizontal beamwidth. As stated earlier, not only are the lines tangent to the arc candidates, but also the ones inside the maximum incidence angle limits of $\pm\beta$. For this reason, k values are taken at a given resolution for each value of θ_i^B and within an aperture of $\pm\beta$:

$$\theta_i^B - \beta \leq \theta_{i,k}^B \leq \theta_i^B + \beta\,. \quad (2)$$

The result of this operation is $i \times k$ different values of $\theta_{i,k}^B$. These are the bearings for a set of lines that are a representation of all the possible candidates compatible with the bin. The final step is to determine the range parameter $\rho_{i,k}^B$ corresponding to each one of the $\theta_{i,k}^B$ bearings obtained. Given the geometry of the problem, they are calculated as

$$\rho_{i,k}^B = x_S^B \cos\left(\theta_{i,k}^B\right) + y_S^B \sin\left(\theta_{i,k}^B\right) + \rho^S \cos(\theta_{i,k}). \quad (3)$$

This set of lines can now be used to determine the cells in the voting space that should receive a single vote from this particular bin. It is assumed that the resolutions chosen during the generation of the $i \times k$ lines are sufficient to ensure a correct exploration of the grid cells and hence that the zone in the discretized space corresponding to the compatible candidates is correctly determined. This process is repeated for all the bins stored in the data buffer. In Figure 7 it is shown how the set of voters looks when assigned to the HS. Note that each selected cell of the HS can receive only one vote from any particular bin and that those cells containing multiple votes therefore represent lines compatible with different individual bins.

Coinciding with the arrival of a new beam, a new voting space is generated to look for winning line candidates. A winning line must be detected only once it has been completely observed (i.e., further beams cannot provide more votes to the candidate). In the voting space, the zone in which the winning lines can exist is completely determined by the subset of all the candidate lines contained in the most recent 180-deg scan sector that do not intersect with the last beam (shaded zones in Figure 7). Any line candidate with a sufficient number of votes found within this zone is declared a winner. This value is chosen as a compromise between avoiding false detections and maintaining the capacity of sensing short walls. (A count of 18 votes has shown good results in different tests.) Performing the detection in this way can ensure that the algorithm detects the lines as soon as they are completely visible. After a line detection, all the bins involved in the election of the selected candidate are removed from the buffer so that they do not interfere with the detection of further features.

It is worth mentioning that, in order to reduce the computational cost of the process, some votings can be skipped. After each voting, it is possible to determine the cell with the largest number of votes and therefore to calculate the number of supplementary votes required to produce a winner. Because additional votes can be obtained only from newly measured bins, it is not necessary to perform more votings before the minimum required number of bins has been measured and introduced in the buffer.

3.2. Uncertainty Estimation

The process to estimate a feature's uncertainty is based on relating the probability of an object existing in a particular place with the measured intensities in the acoustic image representing the same location. There is a high probability that there will be an object in a zone where large intensity values have been measured (e.g., the light shapes in Fig. 4), whereas the probability in the zones with lower intensity measurements gradually decreases to zero (the dark zones in the figure). Given this, the process of applying a threshold to segment the acoustic data can be considered analogous to defining a particular confidence interval for a probability distribution. In other words, a line feature will fall inside the thresholded zone in the acoustic image with a particular confidence level. To make the problem tractable, the probability distribution of a line feature represented in the acoustic image will be approximated to a bivariate Gaussian distribution

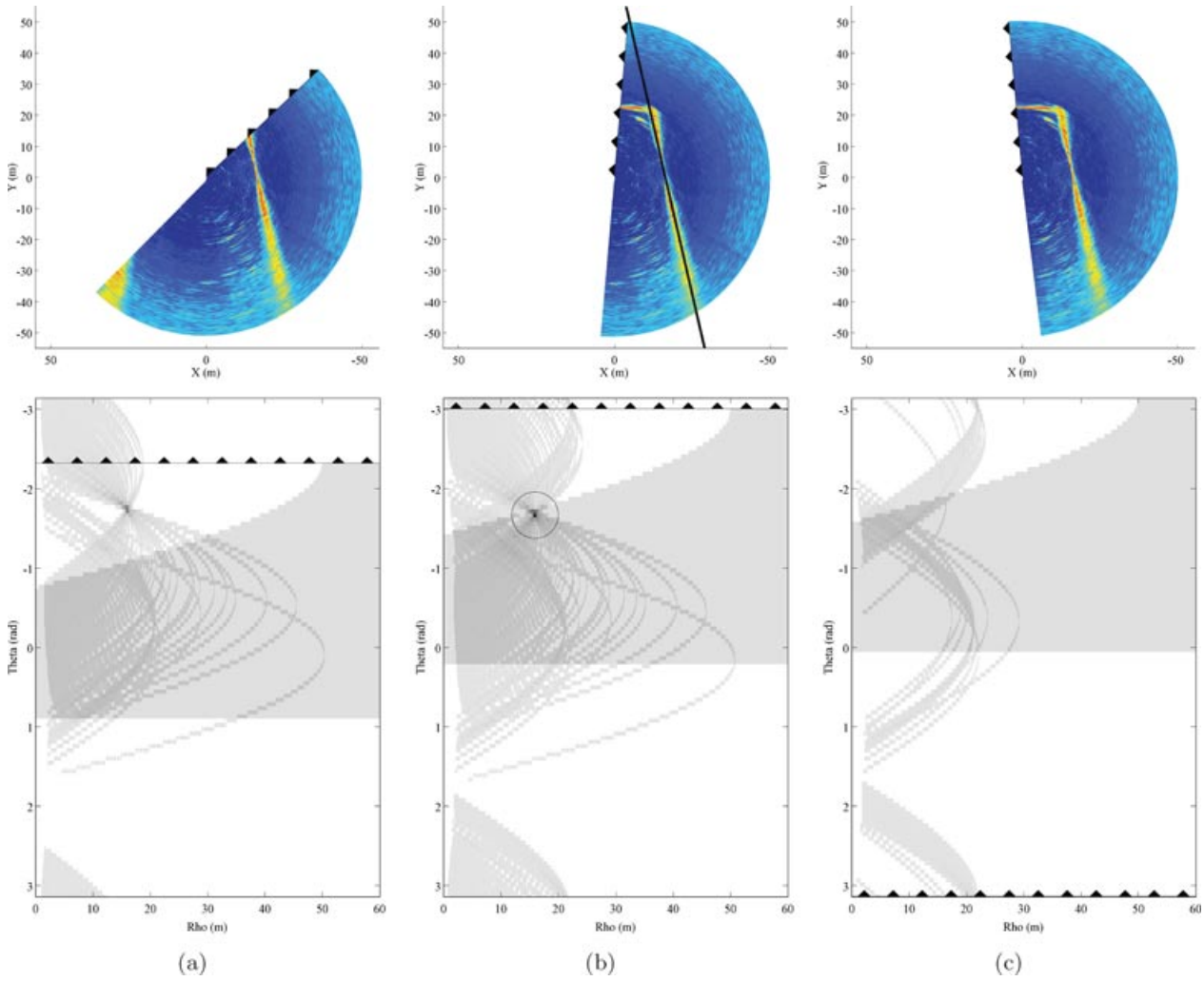

Figure 7. Sequence representing the voting process. The scan sector stored in the buffer is represented together with its corresponding voting space. The line with triangular shapes marks the position of the most recent beam. The shaded zone represents where the candidates have received all the possible votes. (a) Part of the target line is still outside the sector scan and can receive more votes in the future. (b) The line can now be detected because it has been fully observed and more votes cannot be added. (c) Those votes corresponding to the detected line, as well as the old ones that fall outside the 180-deg scan sector, are removed from the HS so they cannot interfere with future line detections.

on its ρ and θ parameters. (An example justifying that this approximation is suitable can be found in Section 3.3.) Therefore, the process to estimate the feature uncertainty consists of determining the Gaussian that best fits the segmented data representing a probability distribution for a given confidence level. A simple description of this process is shown in Algorithm 1. After the detection of a line feature with the voting algorithm, the uncertainty estimation process begins with the assignment of a feasible confidence coefficient to the imprint left after the segmentation. [For instance, it is realistic to assume that the segmented data in Figure 5(b) will contain the real feature in 95% of the cases.] Because the winning candidate line has received a considerable number of votes, it must be one of the lines contained within the confidence interval defined by the segmented imprint. The next step of the process consists of finding a number of compatible lines belonging to the neighborhood of the winning candidate that overlap the segmented data in the same way. The objective of this is to obtain a set of line realizations

Algorithm 1
get_measurements([ρ_c, θ_c], scan, confidence_level)

/* Initialization of the polar grid space that will contain the segmented sonar data */
boolean last180scan [$\rho_{resolution}, \theta_{resolution}$];
[last180scan] = *init_scan*(scan);

/* Set the paradigm with the candidate line from the voting */
[η_c] = *get_overlap_ratio*([ρ_c, θ_c], last180scan)

/* Search for compatible lines */
lines2check = {[ρ_c, θ_c]};
accepted = $\emptyset$;
rejected = $\emptyset$;
While lines2check $\neq \emptyset$ **do**
  [ρ_i, θ_i] = *get_candidate*(lines2check);
  [η_i] = *get_overlap_ratio*([ρ_i, θ_i], last180scan);
  if *accept_line*(η_c, η_i) **then**
    accepted = accepted $\cup$ {[ρ_i, θ_i]};
    lines2check = lines2check\{[ρ_i, θ_i]},
    lines2check = lines2check $\cup$ {*neighbour8connectivity* ([ρ_i, θ_i]) $\cap$ $\overline{\{\text{rejected} \bigcup \text{accepted}\}}$})
  else
    rejected = rejected $\cup$ {[ρ_i, θ_i]};
    lines2check = lines2check\{[ρ_i, θ_i]};

/* Given the set of lines, determine the ellipse that contains the area where they exist */
[*major_axis*, *minor_axis*, $\rho_{mean}, \theta_{mean}, \alpha$] = *get_ellipse*(*accepted*);
/* Given the ellipse and the confidence level related to the segmentation, find the mean and covariance
$\mathbf{z}^V$ = [$\rho_{mean}, \theta_{mean}$]; */
$\mathbf{R}$ = *get_covariance*(*major_axis*, *minor_axis*, $\rho_{mean}, \theta_{mean}$, *confidence_level*) return [$\mathbf{z}^V$, $\mathbf{R}$];

representative of the population contained within the defined confidence interval (i.e., a set of lines that "fill" the segmented area). Estimating the Gaussian distribution from a set of lines is not straightforward; however, it is worth noting that lines described by its ρ and θ parameters can also be represented as points in a polar ρ–θ space. Representing the set of lines in such a space will result in a cloud of points (the lines are similar) with an elliptic form. This particular elliptic disposition of the ρ–θ points suggests that the approximation of the line feature to a Gaussian distribution is correct. Although the space has changed, the set still represents a population of lines within the previously defined confidence interval. This fact is used to estimate the uncertainty of the line feature. This is achieved by approximating the area occupied by the set of points to the area enclosed in the ellipse that a bivariate Gaussian distribution would generate at the same given confidence. By knowing the confidence coefficient and the major and minor axes of the ellipse and their orientation, it is possible to recover the covariance matrix. Moreover, the mean value of a ρ–θ pair defining the line feature can also be obtained from the center of the ellipse. Figure 8 illustrates the different steps involved in the process of estimating the feature uncertainty. The image in Figure 8(a) reproduces a voting space that has just obtained a winning candidate (marked with the small box). The corresponding sonar measurements appear in Figure 8(b) and are represented in the same B-based discrete polar space as the HS. Because the data are represented in polar coordinates, the line feature appears as an arch whose thickness is related to its uncertainty. Note that the ρ–θ pair, representing the winning candidate line in the HS, can also be represented in this space. In fact, to parameterize the line, we use its point with the smallest distance to the origin (again, represented with the same small box in the figure). Applying a threshold and assigning a confidence coefficient to the segmented data results in the space represented in Figure 8(c). At this point, and using the winning candidate line as a paradigm, the search for lines contained within the segmented imprint is performed. The resulting set of lines is contained inside the bounds represented as black arcs, and the representation of the place occupied by their ρ and θ pairs is represented as a black shape at the apex of the arc. The final step of the procedure consists of finding the ellipse containing this area and extracting the covariance matrix given the predefined confidence coefficient. Finally, Figure 8(d) represents the estimated feature over a Cartesian representation of the scan sector. The line in the center corresponds to the ρ–θ mean value, and the lines at the sides represent the uncertainty bounds at 95% confidence.

3.3. Validation of the Feature Extraction Algorithm

To validate the feature extraction algorithm, several tests with both synthetic and real data were carried out. With generating synthetic data, we seek two objectives. The first one is to justify the use of a bivariate ρ–θ Gaussian distribution to represent the uncertain features present in the acoustic images. The second is to have a way of comparing the output from the algorithm with the paradigm, which makes it possible

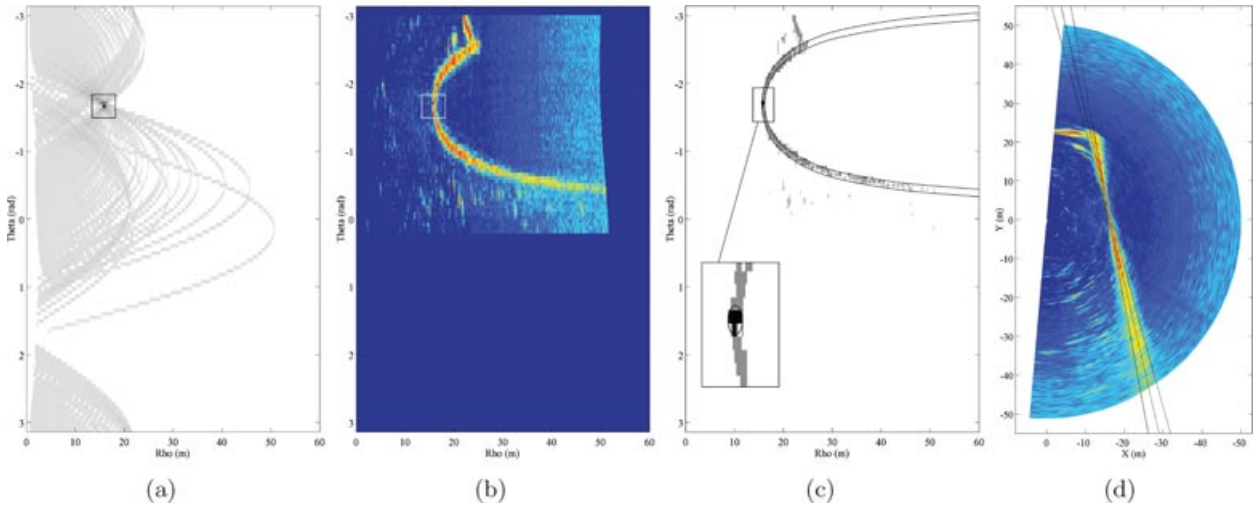

Figure 8. Process for uncertainty estimation. (a) Winning candidate in the HS. (b) Polar representation of the sonar data. (c) Segmented data with the zone occupied by line features inside the confidence level. (d) Estimated feature represented over the scan sector.

to confirm the correctness of the estimation. To obtain the synthetic data, a large population of ρ–θ pairs was generated following a given probability distribution. Then, the lines represented by each pair were projected into a polar space analogous to those produced by the measurements from a MSIS. Each cell from this space represents a bin, and its echo intensity value is assigned according to the number of lines that cross its area. The resulting synthetic data set is represented in polar and Cartesian coordinates in Figures 9(a) and 9(d). In spite of the large uncertainty assigned with the goal of improving the visualization of the uncertainty estimation process, the example has sufficient points in common with the real acoustic images to consider this model as valid. It can be observed how the high-intensity zone in the center corresponds with the major concentration of lines, whereas the dispersion on the sides, caused by the angular uncertainty, produces an effect similar to the loss of intensity and precision affecting the beams with large incidence angles. Figures 9(b) and 9(c) illustrate the voting and the uncertainty estimation process. The elliptic-shaped zone representing the population of compatible lines reflects the Gaussianity of the estimated feature. As can be observed in Figure 9(d), the estimated line feature is a good representation of what appears in the synthetic data. Additional verification of the method can be seen in Figure 10, where the cloud of ρ–θ pairs initially used to generate the synthetic data is plotted together with an ellipse representing the original Gaussian distribution (dashed line) and another one representing the one estimated with the proposed method (solid line). When comparing the two ellipses, it can be appreciated that they are almost coincident except for a small angular misalignment. It is important to note that correlated data, such as those in this example, have turned out to be the most difficult scenario for the proposed uncertainty estimation method, and therefore one could expect even better estimates when working with less-correlated data.

A second set of tests was carried out with real data acquired with the Ictineu AUV. Under real working conditions, it is not possible to obtain reliable references to test the performance of the method. Therefore, only the direct visualization of the estimated line feature represented over the acoustic images can be used as an indicator. The first example in Figure 11(a) shows the features extracted from a data set obtained in a real application scenario; in particular, in the same marina environment that served as the test bed for the SLAM algorithm. The second example is represented in Figure 11(b). It corresponds to an experiment performed in the water tank of the Underwater Robotics Research Center at the University of Girona. This confined environment with highly reflective concrete walls produces noisy data with many reflections and phantoms. The results in both cases are consistent with the representation of the walls in the acoustic images and, moreover, show

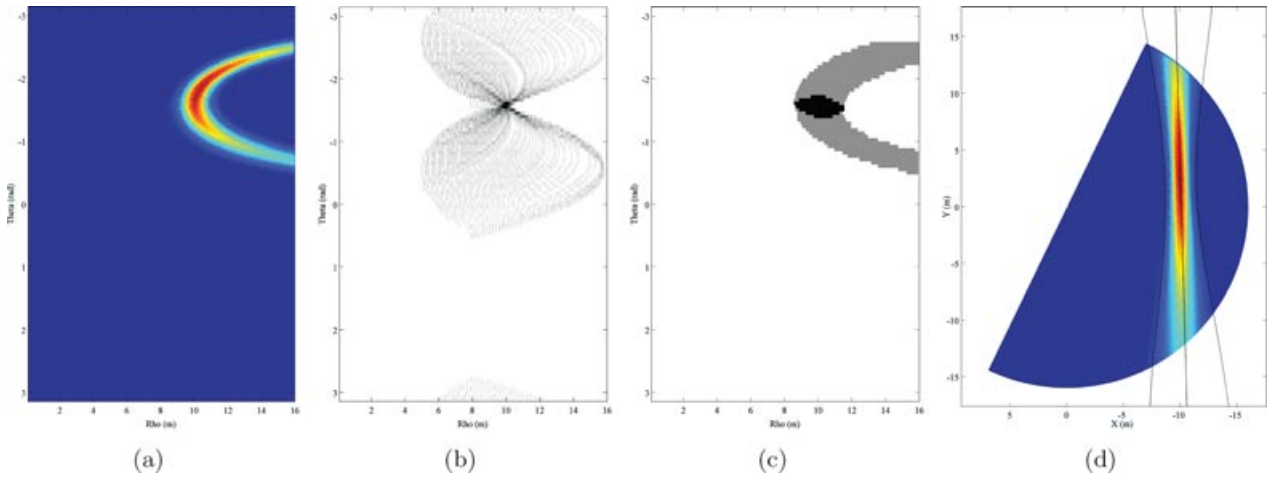

Figure 9. Testing the algorithm with synthetic data. (a) Raw sensor data generated from ρ and θ given a normally distributed uncertainty. Some correlation affects the two variables to increase the difficulty of the test. (b) The voting space clearly identifies the line. (c) Uncertainty estimation using the segmented data. The black elliptic shape corresponds to the lines with compatible overlapping and represents the uncertainty of ρ and θ. (d) The estimated line feature fits almost perfectly with the synthetic one.

a reliable behavior when working with noisy data, filtering linear features from shapeless phantoms.

4. THE SLAM ALGORITHM

An EKF integrates the vehicle's navigation sensors to provide an estimate of its position and retain the estimates of the previously observed features in order to build a map. We favor the use of the EKF over the potentially more efficient information filter (Thrun et al., 2004) because the availability of covariances without any additional computations or approximation is very important for data association. This filter is an implementation of the stochastic map (Smith, Self, & Cheeseman, 1990) in which the estimate of the position of both the vehicle $\hat{\mathbf{x}}_V$ and the set of map features

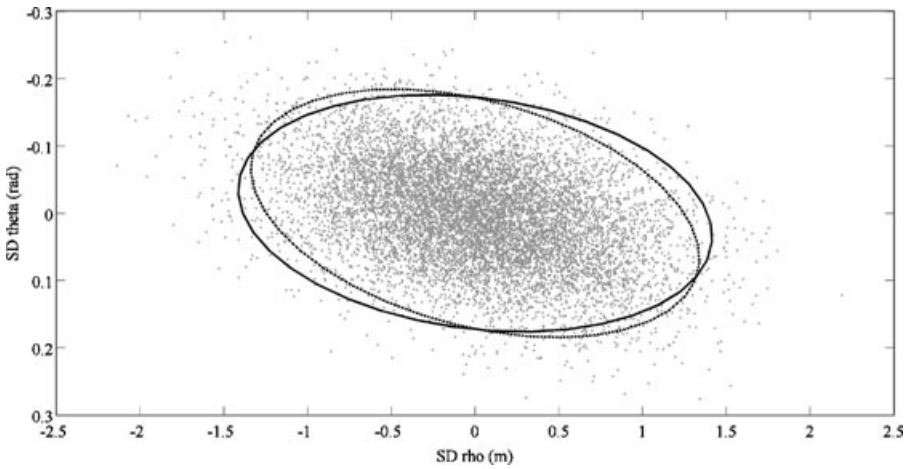

Figure 10. Comparison between the bivariate Gaussian distribution used to produce the synthetic data and the output from the algorithm. The ellipse with dashed line represents the Gaussian distribution at 95% confidence, and the solid one is the output of the algorithm at the same confidence level.

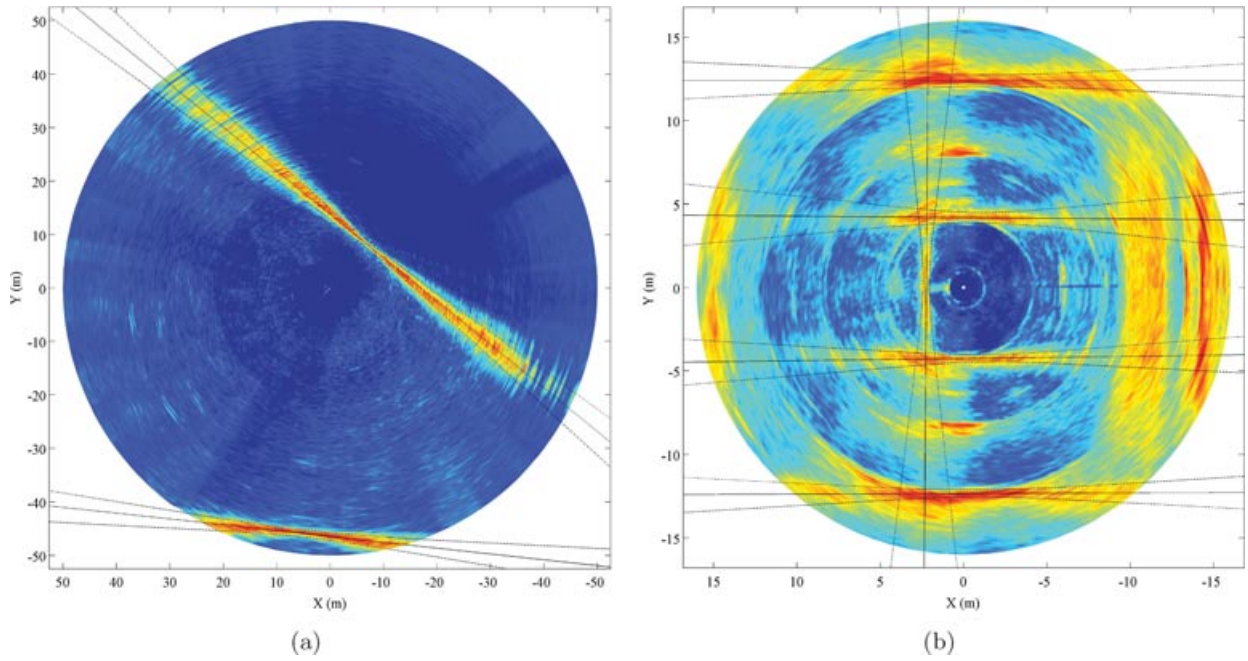

Figure 11. Testing the algorithm with real data. (a) Line features extracted from acoustic data gathered in a marina environment. (b) Line features obtained from acoustic data gathered in a small water tank. The lines on the right-hand side are not estimated as they are split between the start and the end of the scan.

$F = \{\hat{\mathbf{x}}_1 \dots \hat{\mathbf{x}}_n\}$ are stored in the state vector $\hat{\mathbf{x}}$:

$$\hat{\mathbf{x}}(k) = [\hat{\mathbf{x}}_V(k), \qquad \hat{\mathbf{x}}_1(k) \dots \hat{\mathbf{x}}_n(k)]^T. \tag{4}$$

The covariance matrix $\mathbf{P}$ describes the covariance of the vehicle and the features as well as their respective cross correlations:

$$\mathbf{P}(k) = E([\mathbf{x}(k) - \hat{\mathbf{x}}(k)][\mathbf{x}(k) - \hat{\mathbf{x}}(k)]^T | Z(k)). \tag{5}$$

The vehicle's state $\hat{\mathbf{x}}_V$ has dimension 9, which defines the minimum size of the state vector $\hat{\mathbf{x}}$ at the beginning of the execution. The features are represented in polar coordinates, and therefore the state will be increased by 2 with each new incorporation in the map.

4.1. Map Initialization

When creating a new stochastic map at step 0, a base local reference frame L must be selected (Figure 12). In this approach, the initial vehicle position is chosen to set this base location and thus is initialized with perfect knowledge. The vehicle's state $\mathbf{x}_V$ is represented as

$$\mathbf{x}_V = \left[x \; y \; z \; \psi \; u \; v \; w \; r \; \psi_{L_0}\right]^T, \tag{6}$$

where, as defined in Fossen (2002), $[x \; y \; z \; \psi]$ represent the position and heading of the vehicle in the local reference frame L and $[u \; v \; w \; r]$ are their corresponding linear and angular velocities on the vehicle's coordinate frame V. The term ψ_{L_0} represents the angle between the initial vehicle heading at step 0

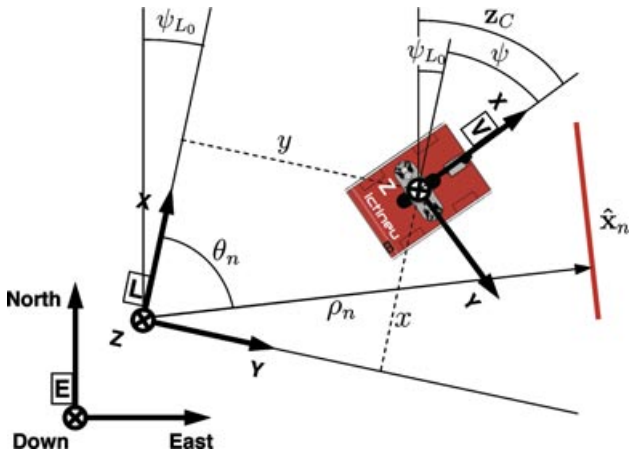

Figure 12. Representation of the different reference coordinate frames.

(orientation of L) and magnetic north in the earth global frame E. This term works as a sensor bias and allows us to initialize the vehicle heading ψ in the local frame L, making it possible to use compass measurements (angle to the north in the E frame) for its estimation as shown in Section 4.4. Assuming that the vehicle is not moving at step 0, the state is initialized as

$$\hat{\mathbf{x}}(0) = \hat{\mathbf{x}}_V(0) = \begin{bmatrix} 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ \hat{\psi}_{L_0} \end{bmatrix}^T;$$
$$\mathbf{P}(0) = \mathbf{P}_V(0) = \begin{bmatrix} \mathbf{0}_{8\times 8} & \mathbf{0}_{8\times 1} \\ \mathbf{0}_{1\times 8} & \sigma^2_{\psi_{L_0}} \end{bmatrix}, \tag{7}$$

where $\hat{\psi}_{L_0}$ takes its value from the first available compass measurement and $\sigma^2_{\psi_{L_0}}$ is initialized accordingly with the sensor's precision. It is worth noting that at the beginning of the execution, the map does not contain any feature and hence the state $\hat{\mathbf{x}}$ contains only the vehicle's state $\hat{\mathbf{x}}_V$.

4.2. Prediction

A simple four-degree-of-freedom, constant-velocity kinematics model is used to predict the state of the vehicle. Because AUVs are commonly operated describing rectilinear transects at constant speed during survey missions, we believe that such a model is a simple but realistic way to describe the motion:

$$\mathbf{x}_V(k) = f[\mathbf{x}_V(k-1), \mathbf{s}_V], \tag{8}$$

$$\begin{bmatrix} x \\ y \\ z \\ \psi \\ u \\ v \\ w \\ r \\ \psi_{L_0} \end{bmatrix}_{(k)} = \begin{bmatrix} x + \left(uT + s_u \frac{T^2}{2}\right)\cos(\psi) \\ \quad -\left(vT + s_v \frac{T^2}{2}\right)\sin(\psi) \\ y + \left(uT + s_u \frac{T^2}{2}\right)\sin(\psi) \\ \quad +\left(vT + s_v \frac{T^2}{2}\right)\cos(\psi) \\ z + wT + s_w \frac{T^2}{2} \\ \psi + rT + s_r \frac{T^2}{2} \\ u + s_u T \\ v + s_v T \\ w + s_w T \\ r + s_r T \\ \psi_{L_0} \end{bmatrix}_{(k-1)}, \tag{9}$$

where $\mathbf{s}_V = [s_u\ s_v\ s_w\ s_r]$ represents an acceleration white noise additive in the velocity with a zero mean and covariance $\mathbf{Q}_V$, which is propagated through integration to the position. On the other hand, as features correspond to fixed objects from the environment, we can assume that they are stationary. Hence, the whole state can be predicted as

$$\hat{\mathbf{x}}(k|k-1) = \{f[\hat{\mathbf{x}}_V(k-1)] \quad \hat{\mathbf{x}}_1(k-1)\ \ldots\ \hat{\mathbf{x}}_n(k-1)\}^T \tag{10}$$

and its covariance matrix updated as

$$\mathbf{P}(k|k-1) = \begin{bmatrix} \mathbf{F}_V(k) & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \mathbf{P}(k-1) \begin{bmatrix} \mathbf{F}_V(k) & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}^T + \begin{bmatrix} \mathbf{G}_V(k) \\ \mathbf{0} \end{bmatrix} \mathbf{Q}_V \begin{bmatrix} \mathbf{G}_V(k) \\ \mathbf{0} \end{bmatrix}^T, \tag{11}$$

where $\mathbf{F}_V$ and $\mathbf{G}_V$ are the Jacobian matrices of partial derivatives of the nonlinear model function f with respect to the state $\mathbf{x}_V$ and the noise s_V, respectively.

4.3. DVL Update

A SonTek Argonaut DVL unit provides bottom tracking and water velocity measurements at a frequency of 1.5 Hz. The unit also includes a pressure sensor allowing depth estimation. The model prediction is updated by the standard KF equations with each new DVL measurement:

$$\mathbf{z}_D = [u_b\ v_b\ w_b\ u_w\ v_w\ w_w\ z_{\text{depth}}]^T, \tag{12}$$

where index b stands for bottom tracking velocity and w for through water velocity. The measurement model is

$$\mathbf{z}_D(k) = \mathbf{H}_D(k)\mathbf{x}(k) + \mathbf{s}_D, \tag{13}$$

$$\mathbf{H}_D(k) = \begin{bmatrix} \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 1} & \mathbf{I}_{3\times 3} & \mathbf{0}_{3\times 2} & \mathbf{0}_{3\times 2n} \\ \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 1} & \mathbf{I}_{3\times 3} & \mathbf{0}_{3\times 2} & \mathbf{0}_{3\times 2n} \\ \mathbf{0}\ 0\ 1 & 0 & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 2} & \mathbf{0}_{1\times 2n} \end{bmatrix}, \tag{14}$$

where $\mathbf{s}_D$ (measurement noise) is a zero-mean white noise with covariance $\mathbf{R}_D$. In some situations, for instance, while operating close to the bottom or in confined spaces, the DVL is unable to produce correct velocity measurements. The bottom tracking measurements are more prone to fail than those with respect to the water. For this reason, and under the condition that no water currents are present, the velocities with respect to the water are introduced in the estimation process. The DVL determines automatically the quality of the received signals and provides a status measurement for the velocities. Then,

different versions of the $\mathbf{H}_D$ matrix are used to fuse one (removing row 2), the other (removing row 1), or both (using the full matrix) readings. In the particular case of the abandoned marina experiment, only 3% of the bottom tracking measurements received a bad status indicator. Therefore, it was not necessary to rely on water velocities during the execution.

4.4. Compass Update

A compass provides measurements at a frequency of 10 Hz. This allows updating the attitude estimate from the model prediction with the standard KF equations. As can be observed in Figure 12, the compass measurement $\mathbf{z}_C$ corresponds to the addition of the heading of the vehicle ψ with respect to the local reference frame L and the orientation of this frame ψ_{L_0}. The resulting measurement model is

$$\mathbf{z}_C(k) = \mathbf{H}_C(k)\mathbf{x}(k) + \mathbf{s}_C, \tag{15}$$

$$\mathbf{H}_C(k) = [0\,0\,0\,1\,0\,0\,0\,0\,1\,\mathbf{0}_{1\times 2n}], \tag{16}$$

where $\mathbf{s}_C$ (measurement noise) is an additive zero-mean white noise with covariance $\mathbf{R}_C$. Working with compass data can be a difficult task in some situations. The effect of electromagnetic fields, such as those produced by the thrusters, and the presence of large structures with ferromagnetic materials can considerably distort measurements and render them unusable. For this reason, it is important to avoid operating close to walls (generally, 1–2 m is sufficient) and to perform a calibration before each mission. On the other hand, a compass is an especially useful sensor for SLAM because it provides *absolute* orientation measurements, unlike the dead-reckoning sensors normally used in SLAM such as wheel encoders, gyros, or, in our case, the DVL. The effect of using a compass is threefold:

1. The error in vehicle orientation will not increase during the SLAM process.
2. Vehicle orientation introduces nonlinearity in the SLAM problem, so that loss of precision because of linearization effects will also be limited.
3. Vehicle orientation errors in a certain step become position errors in future steps. Bounding the errors in orientation will also result in a reduction in the rate of increase of vehicle position errors.

Figure 13 shows the evolution of a vehicle's position and orientation using the DVL velocity data together with the rate of turn measurements from gyros (solid line) and using *absolute* attitude information from the compass (dashed line). We can see that the error in orientation remains constant. There is also a reduction in the rate of increase of the error in the direction transverse to the vehicle's direction of motion.

4.5. Imaging Sonar Beam Arrival

The Tritech Miniking imaging sonar produces beams at a 10–30-Hz rate, depending on the settings of the sensor. Each new beam is stored together with the current vehicle position estimate in a data buffer and fed to the feature extraction algorithm as shown in Sections 2 and 3. Eventually, the information added by a new beam arrival is sufficient to produce a line feature detection. In this case, the ρ–θ pair obtained is represented in a B frame that is placed in the sonar head. For the sake of simplicity, let us assume that the transformation between B and the vehicle's coordinate system is known. Hence, we could represent a new measurement i with respect to the vehicle's frame V as $\mathbf{z}_i^V = [\rho_i^V \;\; \theta_i^V]^T$. Of course, the same transformation should be applied to the covariance matrix obtained from the uncertainty estimation method. This transformation will result in the covariance matrix $\mathbf{R}_i$. The next step is to solve the data association problem; that is, to determine whether the measured line $\mathbf{z}_i^V$ corresponds to any of the features F_j, $j = 1 \ldots n$ already existing in the map and should be used to update the system or, on the contrary, is new and has to be incorporated into the map. The result of the data association process is a hypothesis $\mathcal{H} = j_i$ associating the measurement $\mathbf{z}_i^V$ with one of the map features F_j ($j_i = 0$ indicates that $\mathbf{z}_i^V$ has no correspondence with the existing features). Finding the correct hypothesis is a process involving the analysis of the discrepancy between the actual line measurement and its prediction. This prediction is obtained from the nonlinear measurement function h_j, which relates the i measurement with the state vector $\mathbf{x}(k)$ containing the locations of the vehicle and the j feature:

$$\mathbf{z}_i^V(k) = h_j[\mathbf{x}(k)] + \mathbf{s}_i, \tag{17}$$

$$\begin{bmatrix} \rho_i^V \\ \theta_i^V \end{bmatrix} = \begin{bmatrix} \rho_j - x\cos\theta_j - y\sin\theta_j \\ \theta_j - \psi \end{bmatrix} + \mathbf{s}_i, \tag{18}$$

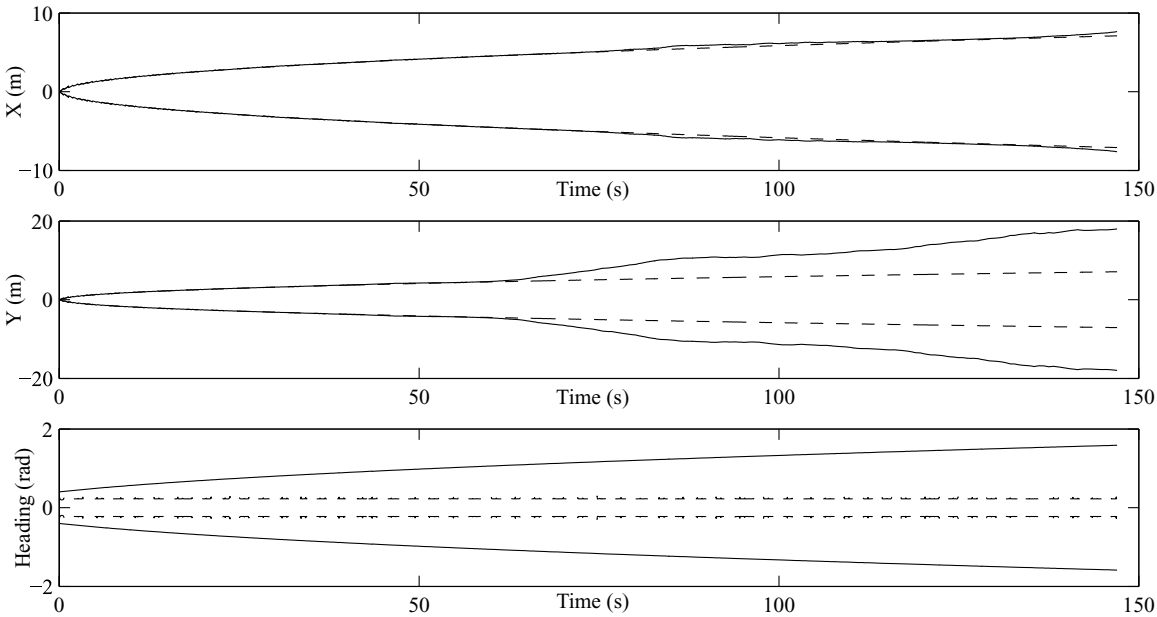

Figure 13. Estimated position covariance plots (2σ bounds). The data correspond to the first minutes of the abandoned marina experiment executing the EKF with the updates from dead-reckoning sensors. The results using a cheap inaccurate gyro sensor are represented with a solid line. The ones using absolute data from a compass are indicated with a dashed line.

where $\mathbf{s}_i$, the noise affecting the line feature observation, is a zero-mean white noise with covariance $\mathbf{R}_i$. To calculate the discrepancy between the measurement and its prediction, the innovation term v_{ij} and its associate covariance matrix $\mathbf{S}_{ij}$ are obtained as

$$\boldsymbol{\nu}_{ij}(k) = \mathbf{z}_i^V(k) - h_j[\hat{\mathbf{x}}(k|k-1)], \tag{19}$$

$$\mathbf{S}_{ij}(k) = \mathbf{H}_j(k)\mathbf{P}(k|k-1)\mathbf{H}(k)_j^T + \mathbf{R}_i, \tag{20}$$

where $\mathbf{H}_j$ represents the Jacobian matrix that linearizes the nonlinear measurement function h_j around the best available estimation of the state $\hat{\mathbf{x}}(k|k-1)$. To determine whether the correspondence is valid, an individual compatibility (IC) test using the Mahalanobis distance is carried out:

$$D_{ij}^2 = \boldsymbol{\nu}_{ij}(k)^T \mathbf{S}_{ij}(k)^{-1} \boldsymbol{\nu}_{ij}(k) < \chi_{d,\alpha}^2, \tag{21}$$

where $d = \dim(\mathbf{h}_j)$ and α is the desired confidence level. It is possible for multiple hypotheses relating the measurement with different map features to satisfy the IC test. Then, in order to select the best candidate, the nearest neighbor criterion is applied. Finally, after the correspondence has been decided, it is used to update the state estimate by means of the EKF update equations:

$$\mathbf{K}_{ij}(k) = \mathbf{P}(k|k-1)\mathbf{H}_j(k)^T \mathbf{S}_{ij}(k)^{-1}, \tag{22}$$

$$\hat{\mathbf{x}}(k) = \hat{\mathbf{x}}(k|k-1) + \mathbf{K}_{ij}(k)\boldsymbol{\nu}_{ij}(k), \tag{23}$$

$$\mathbf{P}(k) = [\mathbf{I} - \mathbf{K}_{ij}(k)\mathbf{H}_j(k)]\mathbf{P}(k). \tag{24}$$

In case there is no valid hypothesis relating the measured line with any of the features from the map (i.e., $\mathcal{H} = 0$), this measurement can be added to the current state vector as a new feature. However, this cannot be done directly because this new feature needs to be represented in the map reference frame. The change of reference is done by compounding (Smith et al., 1990; Tardós et al., 2002) the line feature with the current vehicle position as follows:

$$\hat{\mathbf{x}}(k) = \begin{bmatrix} \hat{\mathbf{x}}_V(k) \\ \hat{\mathbf{x}}_1(k) \\ \vdots \\ \hat{\mathbf{x}}_n(k) \end{bmatrix} \quad \Rightarrow \quad \hat{\mathbf{x}}(k)^+ = \begin{bmatrix} \hat{\mathbf{x}}_V(k) \\ \hat{\mathbf{x}}_1(k) \\ \vdots \\ \hat{\mathbf{x}}_n(k) \\ \hat{\mathbf{x}}_V(k) \oplus \mathbf{z}_i^V(k) \end{bmatrix}. \tag{25}$$

Augmenting the state vector also requires updating the estimated error covariance matrix:

$$\mathbf{P}(k) = \mathbf{F}(k)\mathbf{P}(k)\mathbf{F}(k)^T + \mathbf{G}(k)\mathbf{R}_i\mathbf{G}(k)^T, \quad (26)$$

$$\mathbf{F}(k) = \begin{bmatrix} \mathbf{I} & 0 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & \mathbf{I} \\ \mathbf{J}_{1\oplus} & 0 & \dots & 0 \end{bmatrix}, \quad \mathbf{G}(k) = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \mathbf{J}_{2\oplus} \end{bmatrix}, \quad (27)$$

where $\mathbf{J}_{1\oplus}$ and $\mathbf{J}_{2\oplus}$ are the Jacobian matrices of the compounding transformation.

5. SLAM WITH LOCAL MAPS

The main advantages of building sequences of local maps are the limitation of the cost associated with the update of a full covariance matrix and the improvement of the system's consistency. In the present case, an additional advantage is obtained with using local maps. The parameterization of line features using polar coordinates is the most adequate approach for our type of (polar) sensor. However, it is not the best choice for referencing the features in a large map. Some issues appear when an observation of a new feature is translated from the sensor frame to the map base frame, particularly in those situations in which the map base and the sensor base are far from each other, because a small variation in the θ parameter of a feature with a large ρ value translates in large changes in Cartesian coordinates. Using local maps overcomes this issue as their area is smaller and the reference changes are less critical.

An important restriction of most local mapping methods is that the local maps must be statistically independent (no information can be shared between them) to avoid introducing inconsistency when recovering the global map. As a consequence, vehicle states such as velocities or estimated sensor biases cannot be transferred between maps. Recently, Piniés and Tardós (2007) presented a technique that overcomes this limitation and makes sharing information between local maps possible, while remaining conditionally independent. This is especially useful in our case because it allows us to keep information about the vehicle's state. This method was chosen to implement the local map sequencing in the present work. Although this section summarizes the main characteristics of our particular implementation of the algorithm, a more detailed presentation of the method can be found in the bibliographic reference mentioned.

5.1. Local Map Building

The local map-building process relies on defining a set of state variables that are common to two consecutive maps. This commonality serves as a link to transmit the information from one map to the other while maintaining their conditional independence. In the application at hand, this link makes it possible to use the estimates of the vehicle's velocities and the compass bias obtained at the end of a map to initialize the next local map. Moreover, after new measurements modify the estimate of these terms, it is also possible to update their estimated values in the previous map through back-propagation.

The procedure to build the local maps begins by initializing the filter presented in Section 4. Then, the vehicle moves through the scenario, acquiring sensor information regarding its own state and the position of existing features. After a certain time period, the state vector $\hat{\mathbf{x}}$ will contain the current estimate of the states of the vehicle $\hat{\mathbf{x}}_V$ as well as the position of several map features $F = \{\mathbf{x}_1 \dots \mathbf{x}_n\}$. At a given instant k, the current local map is finished and a new one is initialized by defining a new state $\hat{\mathbf{x}}$ containing only the current vehicle state $\hat{\mathbf{x}}_V$ as follows:

$$\hat{\mathbf{x}}(k) = [\hat{\mathbf{x}}_V(k) \;\; \mathbf{T}\hat{\mathbf{x}}_V(k)]^T, \quad (28)$$

where the first term is a clone of the vehicle's state that will serve as a link between the two local maps and the second term represents the initialization of the vehicle's state in the new map after performing a change of the base reference defined by the linear transformation function $\mathbf{T}$:

$$\mathbf{T}\hat{\mathbf{x}}_V(k) = \begin{bmatrix} 0 & 0 & 0 & 0 & u & v & w & r & \psi + \psi_{L_0} \end{bmatrix}^T, \quad (29)$$

$$\mathbf{T} = \begin{bmatrix} \mathbf{0}_{4\times4} & \mathbf{0}_{4\times4} & \mathbf{0}_{4\times1} \\ \mathbf{0}_{4\times4} & \mathbf{I}_{4\times4} & \mathbf{0}_{4\times1} \\ 0\,0\,0\,1 & \mathbf{0}_{1\times4} & 1 \end{bmatrix}. \quad (30)$$

This transformation sets the current vehicle location as the base reference of the new local map, while its velocity estimates (represented by the vehicle's frame) are preserved. It is important to note that the term of the compass bias is also updated to make integrating compass measurements with respect to the new base possible. The resulting state vector has a dimension of 18. To complete the initialization process, the state covariance matrix P has to be set

accordingly:

$$\mathbf{P}(k) = \begin{bmatrix} \mathbf{P}_V(k) & \mathbf{P}_V(k)\mathbf{T}^T \\ \mathbf{T}\mathbf{P}_V(k) & \mathbf{T}\mathbf{P}_V(k)\mathbf{T}^T \end{bmatrix}, \tag{31}$$

where $\mathbf{P}_V$ is the submatrix corresponding to the vehicle's state from the full covariance matrix of the first map. At this point, the filter is ready to begin the estimation of the new local map using the equations presented in Section 4. Of course, those equations should be adapted to the presence of the common state variables representing the link between the maps.

5.2. Local Map Joining

The map-building procedure will result in a sequence of local maps with the form

$$\mathcal{M}_i = (\hat{\mathbf{x}}^i, \mathbf{P}^i); \quad \text{with} \quad \hat{\mathbf{x}}^i = \begin{bmatrix} \hat{\mathbf{x}}_V^{i-1} & \hat{\mathbf{x}}_1^i \dots \hat{\mathbf{x}}_n^i & \hat{\mathbf{x}}_V^i \end{bmatrix}^T. \tag{32}$$

Each local map $\mathcal{M}_i$ contains the term $\hat{\mathbf{x}}_V^{i-1}$, a copy of the vehicle's state at the end of the previous map $\mathcal{M}_{i-1}$, which represents the common part connecting the two maps. It also contains a set of features $\{\hat{\mathbf{x}}_1^i \dots \hat{\mathbf{x}}_n^i\}$, which have been added to the state vector during the generation of the map, and, finally, the term $\hat{\mathbf{x}}_V^i$, which represents the estimate of the vehicle's state throughout the creation of the map and whose final value will serve to initialize the $\mathcal{M}_{i+1}$ local map.

The process of joining local maps into a single global map is described here using a notation similar to the one presented in Piniés and Tardós (2007). Consider two consecutive local maps defined as

$$\mathcal{M}_A = \left(\begin{bmatrix} \hat{\mathbf{x}}_A \\ \hat{\mathbf{x}}_{Ca} \end{bmatrix}, \begin{bmatrix} \mathbf{P}_A & \mathbf{P}_{ACa} \\ \mathbf{P}_{CaA} & \mathbf{P}_{Ca} \end{bmatrix} \right), \tag{33}$$

$$\mathcal{M}_B = \left(\begin{bmatrix} \hat{\mathbf{x}}_{Cb} \\ \hat{\mathbf{x}}_B \end{bmatrix}, \begin{bmatrix} \mathbf{P}_{Cb} & \mathbf{P}_{CbB} \\ \mathbf{P}_{BCb} & \mathbf{P}_B \end{bmatrix} \right). \tag{34}$$

The part common to both maps is represented by $\hat{\mathbf{x}}_{Ca}$, which corresponds to the state of the vehicle at the end of $\mathcal{M}_A$, and $\hat{\mathbf{x}}_{Cb}$, which is initialized as an exact clone of $\hat{x}_{Ca}$ during the creation of the $\mathcal{M}_B$ map but evolves because of the updates propagated through the correlation terms during the generation of $\mathcal{M}_B$. The rest of the information stored in the maps is represented by $\hat{\mathbf{x}}_A$ and $\hat{\mathbf{x}}_B$. According to the general form described in Eq. (32), $\hat{\mathbf{x}}_A$ will contain the common term representing the link with a previous map and all the features in $\mathcal{M}_A$ and $\hat{\mathbf{x}}_B$ will contain the features in $\mathcal{M}_B$ and the estimate of the vehicle's state at the end of the map.

The objective of the map-joining process is to obtain a single global map containing the information from all the local maps. In this example, the global map is represented by

$$\mathcal{M}_{AB} = \left(\begin{bmatrix} \hat{\mathbf{x}}'_A \\ \hat{\mathbf{x}}_{Cb} \\ \hat{\mathbf{x}}_B \end{bmatrix}, \begin{bmatrix} \mathbf{P}'_A & \mathbf{P}'_{ACb} & \mathbf{P}'_{AB} \\ \mathbf{P}'_{CbA} & \mathbf{P}_{Cb} & \mathbf{P}_{CbB} \\ \mathbf{P}'_{BA} & \mathbf{P}_{BCb} & \mathbf{P}_B \end{bmatrix} \right). \tag{35}$$

The last two blocks of the global map coincide exactly with $\mathcal{M}_B$. (They are up to date.) Therefore, only the terms related to $\mathbf{x}_A$ need to be updated (a tilde is used to denote those terms). This is because the first map has been updated only with its own measurements but does not contain any information obtained during the generation of the second map. To transmit the effect of those measurements to the estimates in the $\mathcal{M}_A$ map, a back-propagation procedure is carried out:

$$\mathbf{K} = \mathbf{P}_{ACa}\mathbf{P}_{Ca}^{-1}, \tag{36}$$

$$\mathbf{P}'_{ACb} = \mathbf{K}\mathbf{P}_{Cb}, \tag{37}$$

$$\mathbf{P}'_A = \mathbf{P}_A + \mathbf{K}(\mathbf{P}'_{CbA} - \mathbf{P}_{CaA}), \tag{38}$$

$$\hat{\mathbf{x}}'_A = \hat{\mathbf{x}}_A + \mathbf{K}(\hat{\mathbf{x}}_{Cb} - \hat{\mathbf{x}}_{Ca}). \tag{39}$$

Moreover, in order to recover the full covariance matrix of the global map, it is necessary to calculate the correlation term relating the two local maps:

$$\mathbf{P}'_{AB} = \mathbf{P}'_{ACb}\mathbf{P}_{Cb}^{-1}\mathbf{P}_{CbB}, \tag{40}$$

$$= \mathbf{K}\mathbf{P}_{CbB}. \tag{41}$$

At this point, all the elements in $\mathcal{M}_{AB}$ have been determined. It is important to note that this map-joining procedure is applicable to sequences of more than two local maps. After each union, the resulting map still contains the common elements that serve as a link with the adjacent ones; therefore, the same procedure can be applied.

Each element from the resulting global map is still represented in the base frame of the respective local map. Moreover, it is possible that some features could have been observed from different local maps and therefore they are repeated. The final part of this procedure consists of transforming all the features to

a common coordinate frame. Data association can be carried out and, after obtaining the correspondences, the global map can be updated to produce a better estimate. In the context of this work, the Joint Compatibility Branch and Bound (JCBB) data association algorithm has been used (Neira & Tardós, 2001) to obtain the hypothesis relating features from different local maps. Then, an implicit measurement equation representing the equivalence between paired features is used to perform the update (Castellanos & Tardós, 1999).

6. EXPERIMENTAL RESULTS

To test the reliability of the proposed algorithm, we carried out an extensive experiment in an abandoned marina on the Costa Brava (Spain). The Ictineu AUV gathered a data set along a 600-m operated trajectory that included a small loop around the principal water tank and a 200-m straight path through an outgoing canal. The vehicle moved at about 0.2 m/s, and the experiment lasted 50 min. The data set included measurements from the DVL, the compass, and the imaging sonar, which was set to a range of 50 m, with a resolution of 0.1 m and 1.8 deg. To minimize the perturbations in the compass, a calibration was performed before starting the experiment. Because the bottom was natural and a minimum distance to the walls was always maintained, the calibration was not affected by important distortions. However, to deal with punctual effects, a pessimistic approach was adopted for the compass uncertainty model. For validation purposes, the vehicle was operated close to the surface attached to a global positioning system (GPS)–equipped buoy used for registering the trajectory. Figure 14 represents the trajectory obtained during the generation of the different submaps (solid black line), which is a good approximation to the one measured with the GPS (dashed line). In our implementation, a new local map is started when the

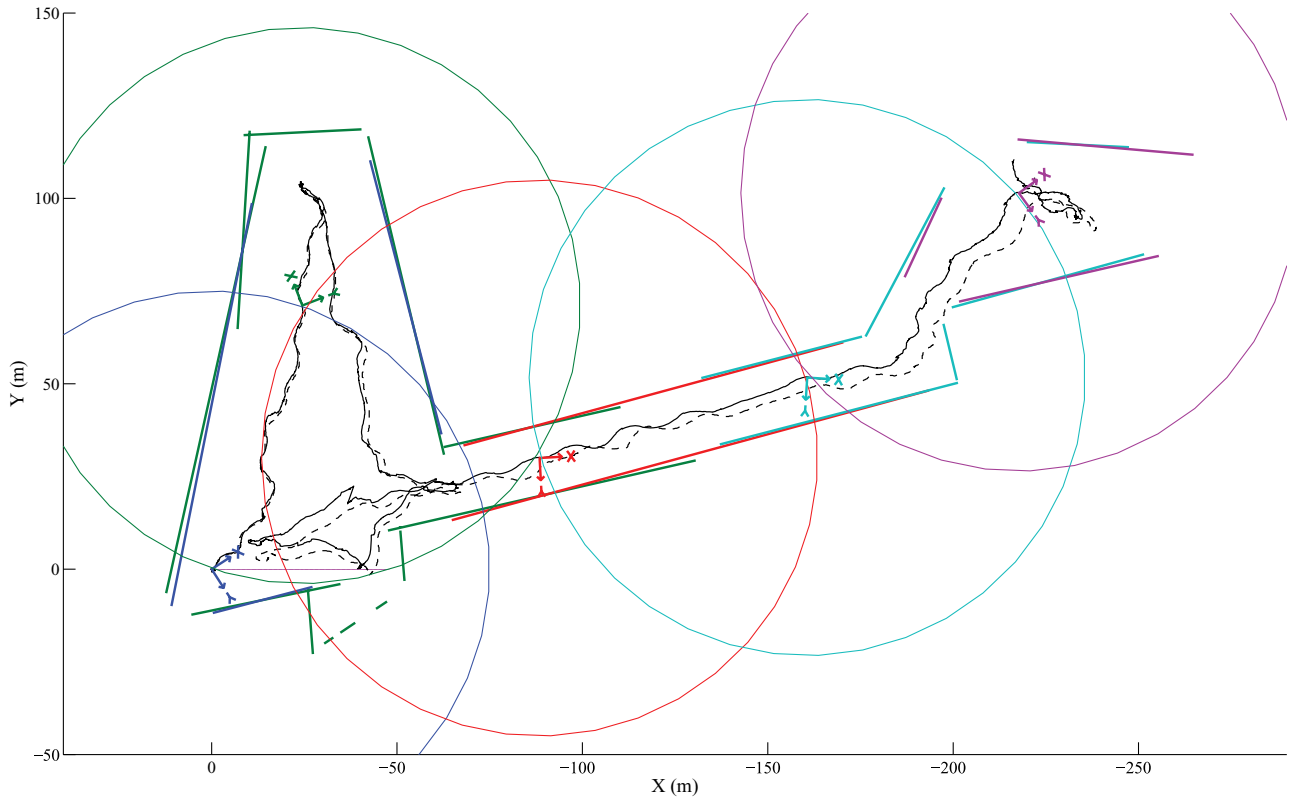

Figure 14. Sequence of local maps. The SLAM trajectory is represented with a solid line and the GPS with a dashed line. The circles represent the limits of each local map, and their centers coincide with the corresponding base frames.

Figure 15. The resulting global map together with the dead-reckoning (dash–dotted line), GPS (dashed line), and SLAM (solid line) trajectories represented over a satellite image of the scenario.

vehicle is at a distance superior to 75 m from the map base frame. (These limits appear as circles in the figure.) This resulted in the creation of five local maps whose dimensions made possible a correct representation of the features in polar coordinates. The line segments corresponding to the walls were also obtained by analyzing the overlapping of the measured line features with their corresponding imprints in the thresholded acoustic data. It is important to clarify that only the lines are included in the state vector and hence are estimated with the filter. The endpoints of the segment should, therefore, be taken only as an approximation. A remarkable resemblance can be observed between the SLAM trajectory estimated during the creation of the different maps and the reference data from the GPS. At approximately [−40, 25], a sudden position change appears as a consequence of reobserving, in the second local map and after performing a small loop, the features at the beginning of the long canal. Given the shape and dimensions of the scenario and the range setting of the sonar, the few places where a loop closure could happen are limited to the principal tank. The path followed toward the top part of this tank is split between the two first local maps. Therefore, the only place where a loop closure could happen is in the lower part of the tank, when the features at each side go out of sight. In these loop-closing situations, a discontinuity is introduced in the trajectory stored in the sonar data buffer. It is, however, uncommon for such strong position corrections to affect the feature extraction process. The update that produces this discontinuity generally takes place just after the complete observation of a feature and during the initial moments of the next one. Therefore, the major part of the new bins introduced into the buffer will usually be obtained on the already corrected track. It can also be observed how the discrepancy with the GPS data increments when the vehicle moves through the canal. This is mainly caused

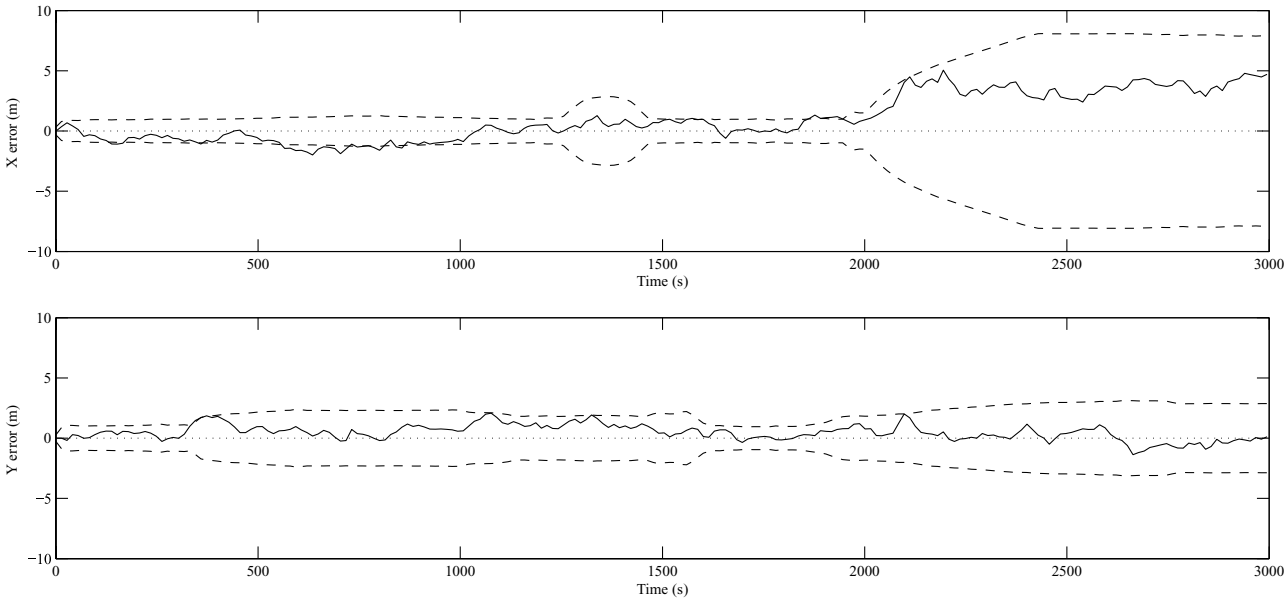

Figure 16. Error plots (2σ bounds) for the resulting estimated trajectory after the local map joining. The GPS data were used as ground truth.

by the absence of features placed perpendicular to the direction of the canal axis, which makes the correction of the errors accumulating in this direction difficult. The global map and the estimated trajectory (solid line) obtained after the joining are plotted in Figure 15 layered over a satellite image. For comparison, the GPS trajectory (dashed line) and a dead-reckoning trajectory (dot–dashed line) obtained by executing the filter with only the measurements from the DVL and the compass are also represented. As can be observed, the dead-reckoning data suffer from an appreciable drift (even causing the trajectory to go outside the canal), whereas the SLAM estimated trajectory follows the GPS track with considerable precision. The resulting map is also a good approximation, matching almost perfectly with the real position of the marina's boundaries. A problem with the position of a feature is observed in the upper-left-hand part of the map. This effect is due to the similarity between the two intersecting lines. The small intersection angle makes it difficult for the feature extraction to discern between the two lines and, eventually, they are interpreted as a single (slightly distorted) one. Of course, this also affects the measurement of the segment endpoints, as it makes it difficult to determine the overlapping with the thresholded data and tends to make longer segments. Some minor problems with the measurement of the segment endpoints are also observed in the small channel entrance in the lower-left-hand part of the map. They mainly appear because of the polar parameterization used in the line features, which, in some particular situations, produces a misplacement of the segment endpoints. Figure 16 represents the error plots for the resulting estimated trajectory obtained after producing the local map. The GPS data were used as the ground truth. As can be seen, the error is contained within the 2σ limits, confirming the correct operation of the SLAM. Additional results validating the algorithm are shown in Figure 17, which reproduces two acoustic images generated by placing the sonar measurements from the complete data set according to the dead-reckoning and the SLAM estimated trajectories. An averaged representation of all the overlapping scans was used; therefore, one can expect the diffuse appearance shown on the dead-reckoning image as a result of the dispersion induced by the erroneous trajectory. On the other hand, using the SLAM trajectory provides a more accurate placement of the measurements that results in a sharper image. Only the acquisition of the sensor data was performed in real time by the computers onboard the Ictineu. This

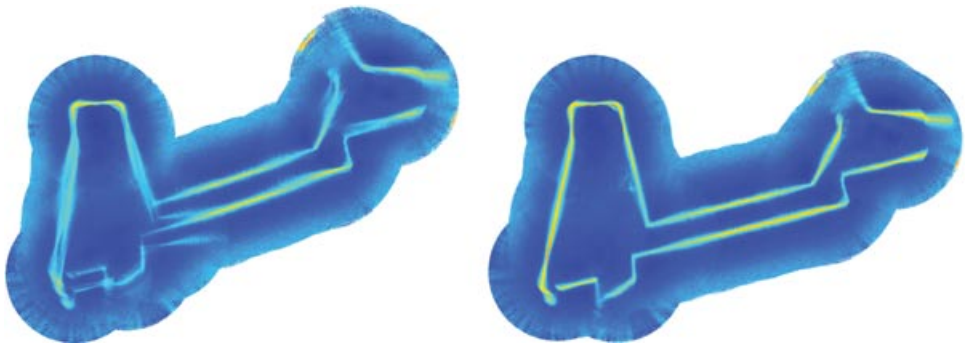

Figure 17. Acoustic map obtained after an averaged composition of the sonar readings along the dead-reckoning estimated trajectory (left) and the SLAM estimated trajectory (right).

SLAM approach was implemented on Matlab and executed offline on an ordinary desktop computer. The execution time is shorter than the duration of the real experiment. We assume that a more optimized implementation should be able to operate onboard.

7. CONCLUSIONS AND FURTHER WORK

An algorithm to perform SLAM in partially structured underwater environments has been presented. It takes advantage of walls and other planar structures typically present in some scenarios of interest, in which reliable line features from acoustic images can be extracted. The main contributions of this work include a feature extraction method capable of working with the continuous stream of data from the MSIS while dealing with the distortions induced by the vehicle movement in the acoustic images, a new method for estimating their uncertainty, and the application domain with experimental results supporting the viability of the proposal. Moreover, the EKF–SLAM algorithm has been complemented with the implementation of a local map-building method that makes possible sharing information between maps while maintaining their independence. An additional contribution that complements this work is the publication of the marina data set that includes logs from the MSIS, DVL, compass, and GPS sensors. The data set can be downloaded from Ribas (2006).

Further work can be done to improve the feature representation in the state vector by substituting the current ρ–θ parameterization for a more reliable approach such as the SP-map (Castellanos & Tardós, 1999). This SLAM approach will also benefit from the integration of additional types of features. The presented algorithm can be adapted to extract any other kind of features susceptible to parameterization, for instance, the detection of point features representing corners in the intersection of two walls. Of course, the possibility of building local maps also opens the door to mapping larger environments and also further improvements such as multirobot SLAM. All these improvements should conclude with the testing of the algorithm in more challenging environments.

ACKNOWLEDGMENTS

The authors thank Marc Carreras, Andres El-Fakdi, Narcís Palomeras, and Emili Hernàndez for their help with the experimental setup and the acquisition of the marina data set. This work has been funded in part by projects DPI2005-09001-C03-01 and DPI2006-13578 of the Dirección General de Investigación of Spain.

REFERENCES

Bailey, T., & Durrant-Whyte, H. F. (2006). Simultaneous localization and mapping (SLAM): Part II, State of the art. IEEE Robotics and Automation Magazine, 13(3), 108–117.

Bosse, M., Newman, P., Leonard, J., Soika, M., Feiten, W., & Teller, S. (2003). An Atlas framework for scalable mapping. In Proceedings of the IEEE International Conference on Robotics and Automation, Taipei, Taiwan (pp. 1899–1906). Thousand Oaks, CA: Sage.

Bosse, M., Newman, P., Leonard, J. J., & Teller, S. (2004). SLAM in large-scale cyclic environments using the Atlas framework. International Journal of Robotics Research, 23(12), 1113–1139.

Castellanos, J., Montiel, J., Neira, J., & Tardós, J. (1999). The SPmap: A probabilistic framework for simultaneous localization and map building. IEEE Transactions on Robotics and Automation, 15(5), 948–953.

Castellanos, J. A., Neira, J., & Tardós, J. D. (2004). Limits to the consistency of EKF-based SLAM. In Proceedings of the 5th IFAC Symposium on Intelligent Autonomous Vehicles, Lisbon, Portugal (pp. 1244–1249).

Castellanos, J. A., & Tardós, J. D. (1999). Mobile robot localization and map building: A multisensor fusion approach. Boston, MA: Kluwer Academic Publishers.

Clemente, L., Davison, A. J., Reid, I. D., Neira, J., & Tardós, J. D. (2007). Mapping large loops with a single handheld camera. In Proceedings of Robotics Science and Systems, Atlanta, GA.

Durrant-Whyte, H. F., & Bailey, T. (2006). Simultaneous localization and mapping (SLAM): Part I, The essential algorithms. IEEE Robotics and Automation Magazine, 13(2), 99–108.

Estrada, C., Neira, J., & Tardós, J. D. (2005). Hierarchical SLAM: Real-time accurate mapping of large environments. IEEE Transactions on Robotics, 21(4), 588–596.

Eustice, R., Singh, H., Leonard, J., Walter, M., & Ballard, R. (2005). Visually navigating the RMS Titanic with SLAM information filters. In Proceedings of Robotics Science and Systems, Cambridge, MA (pp. 57–64). Cambridge, MA: MIT Press.

Fairfield, N., Jonak, D., Kantor, G. A., & Wettergreen, D. (2007). Field results of the control, navigation, and mapping systems of a hovering AUV. In Proceedings of the 15th International Symposium on Unmanned Untethered Submersible Technology, Durham, NH.

Fossen, T. I. (2002). Marine control systems: Guidance, navigation & control of ships, rigs and underwater vehicles (1st Ed.). Trondheim, Norway: Marine Cybernetics AS.

Griffiths, G., McPhail, S., Rogers, R., & Meldrum, D. (1998). Leaving and returning to harbour with an autonomous underwater vehicle. In Proceedings of the Oceanology International, Brighton, UK (volume 3, pp. 75–87). New Malden, UK: Spearhead Exhibitions Ltd.

Guivant, J., Nebot, E., & Durrant-Whyte, H. F. (2000). Simultaneous localization and map building using natural features in outdoor environments. In Proceedings of the International Conference on Intelligent Autonomous Systems, Venice, Italy (pp. 581–588).

Guivant, J. E., & Nebot, E. M. (2001). Optimization of the simultaneous localization and map-building algorithm for real-time implementation. IEEE Transactions on Robotics and Automation, 17(3), 242–257.

Huang, S., & Dissanayake, G. (2007). Convergence and consistency analysis for extended Kalman filter based SLAM. IEEE Transactions on Robotics, 23(5), 1036–1049.

Illingworth, J., & Kittler, J. (1988). A survey of the Hough transform. Computer Vision, Graphics, and Image Processing, 44(1), 87–116.

Kim, J. H., & Sukkarieh, S. (2003). Airborne simultaneous localisation and map building. In Proceedings of the IEEE International Conference on Robotics and Automation, Taipei, Taiwan (pp. 406–411).

Leonard, J. J., Carpenter, R. N., & Feder, H. J. S. (2001). Stochastic mapping using forward look sonar. Robotica, 19(5), 467–480.

Leonard, J. J., & Durrant-Whyte, H. F. (1992). Directed sonar sensing for mobile robot navigation. Norwell, MA: Kluwer Academic Publishers.

Leonard, J. J., & Feder, H. (2001). Decoupled stochastic mapping. IEEE Journal of Oceanic Engineering, 26(4), 561–571.

Leonard, J. J., & Newman, P. M. (2003). Consistent, convergent and constant-time SLAM. In Proceedings of the International Joint Conference on Artificial Intelligence, Acapulco, Mexico (pp. 1143–1150). Morgan Kaufmann.

Martins, A., Matos, A., Cruz, N., & Pereira, F. L. (1999). IES an open system for underwater inspection. In Proceedings of the Oceans MTS/IEEE, Seattle, WA (volume 2, pp. 549–554).

Neira, J., & Tardós, J. D. (2001). Data association in stochastic mapping using the joint compatibility test. IEEE Transactions on Robotics and Automation, 17(6), 890–897.

Newman, P. M., & Leonard, J. J. (2003). Pure range-only sub-sea SLAM. In Proceedings of the IEEE International Conference on Robotics and Automation, Taipei, Taiwan (volume 2, pp. 1921–1926).

Newman, P. M., Leonard, J. J., & Rikoski, R. J. (2003). Towards constant-time SLAM on an autonomous underwater vehicle using synthetic aperture sonar. In Proceedings of the 11th International Symposium on Robotics Research, Sienna, Italy (volume 15, pp. 409–420). Berlin: Springer.

Ni, K., Steedly, D., & Dellaert, F. (2007). Tectonic SAM: Exact, out-of-core, submap-based SLAM. In Proceedings of the IEEE International Conference on Robotics and Automation, Rome, Italy (pp. 1678–1685).

Piniés, P., & Tardós, J. (2007). Scalable SLAM building conditionally independent local maps. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA (pp. 3466–3471).

Ribas, D. (2006). Dataset obtained in an abandoned marina, St. Pere Pescador (Spain). http://eia.udg.es/%7Edribas (online; accessed July 12, 2007).

Ribas, D., Neira, J., Ridao, P., & Tardós, J. D. (2006). SLAM using an imaging sonar for partially structured environments. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China (pp. 5040–5045).

Ribas, D., Palomer, N., Ridao, P., Carreras, M., & Hernàndez, E. (2007). Ictineu AUV wins the first SAUC-E competition. In Proceedings of the IEEE International Conference on Robotics and Automation, Rome, Italy (pp. 151–156).

Roman, C., & Singh, H. (2005). Improved vehicle based multibeam bathymetry using sub-maps and SLAM. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, Canada (pp. 3662–3669).

Smith, R., Self, M., & Cheeseman, P. (1990). Estimating uncertain spatial relationships in robotics. In Autonomous robot vehicles (pp. 167–193). Springer-Verlag: New York.

Tardós, J. D., Neira, J., Newman, P., & Leonard, J. (2002). Robust mapping and localization in indoor environments using sonar data. International Journal of Robotics Research, 21(4), 311–330.

Tena, I., Petillot, Y., Lane, D. M., & Salson, C. (2001). Feature extraction and data association for AUV concurrent mapping and localisation. In Proceedings of the IEEE International Conference on Robotics and Automation, Seoul, Korea (pp. 2785–2790).

Tena, I., Reed, S., Petillot, Y., Bell, J., & Lane, D. M. (2003). Concurrent mapping and localisation using side-scan sonar for autonomous navigation. In Proceedings of the 13th International Symposium on Unmanned Untethered Submersible Technology, Durham, NH.

Thrun, S., Liu, Y., Koller, D., Ng, A. Y., Ghahramani, Z., & Durrant-Whyte, H. F. (2004). Simultaneous localization and mapping with sparse extended information filters. International Journal of Robotics Research, 23(7–8), 693–716.

Williams, S., & Mahon, I. (2004). Simultaneous localisation and mapping on the Great Barrier Reef. In Proceedings of the IEEE International Conference on Robotics and Automation, New Orleans, LA (volume 2, pp. 1771–1776).

Williams, S. B., Dissanayake, G., & Durrant-Whyte, H. F. (2002). An efficient approach to the simultaneous localisation and mapping problem. In Proceedings of the IEEE International Conference on Robotics and Automation, Washington, DC (pp. 406–411).

Williams, S. B., Newman, P. M., Rosenblatt, J., Dissanayake, G., & Durrant-Whyte, H. (2001). Autonomous underwater navigation and control. Robotica, 19(5), 481–496.