# Autonomy through SLAM for an Underwater Robot

John Folkesson and John Leonard

**Abstract.** An autonomous underwater vehicle (AUV) is achieved that integrates state of the art simultaneous localization and mapping (SLAM) into the decision processes. This autonomy is used to carry out undersea target reacquisition missions that would otherwise be impossible with a low-cost platform. The AUV requires only simple sensors and operates without navigation equipment such as Doppler Velocity Log, inertial navigation or acoustic beacons. Demonstrations of the capability show that the vehicle can carry out the task in an ocean environment. The system includes a forward looking sonar and a set of simple vehicle sensors. The functionality includes feature tracking using a graphical square root smoothing SLAM algorithm, global localization using multiple EKF estimators, and knowledge adaptive mission execution. The global localization incorporates a unique robust matching criteria which utilizes both positive and negative information. Separate match hypotheses are maintained by each EKF estimator allowing all matching decisions to be reversible.

## 1 Introduction

The underwater ocean environment poses tremendous challenges to an autonomous underwater vehicle (AUV). Some of these are common to those faced by a remotely operated vehicle (ROV). Hull integrity under pressure, vehicle locomotion, power source, control, and stable vehicle attitude are all special constraints for underwater systems. Operating without physical, radio, or visual contact to the vehicle implies an increased degree of autonomy. To achieve autonomy further challenges must be

John Folkesson
Massachusetts Institute of Technology
e-mail: johnfolk@mit.edu

John Leonard
Massachusetts Institute of Technology
e-mail: jleonard@mit.edu

C. Pradalier, R. Siegwart, and G. Hirzinger (Eds.): Robotics Research, STAR 70, pp. 55–70.
springerlink.com 

addressed that relate to the machine's decision making. These challenges are mainly in the areas of perception, localization, and reasoning. All three of these areas must accurately deal with uncertainties. Perception is limited by the physical properties of water; localization is made difficult by the presence of tides, wind driven currents, and wave surge, which can easily disorient a robot while underwater.

Underwater machine proprioception uses a vehicle model and the actuation signals, often implemented as the prediction step of an extended Kalman filter (EKF). Additional proprioception can be provided by an inertial sensor. Available exteroceptive sensors are depth, altitude, gravitation, and magnetic field. These sensors give absolute measurements of four of the six degrees of freedom (DOF) for the vehicle motion. The other two DOF are the position in the horizontal plane or local grid. When the motion of the vehicle in this plane is estimated solely using proprioceptive localization, the errors in this so called dead-reckoning estimate grow without bound.

There are sensors to give absolute measurements of these horizontal plane motions. One can augment the natural environment with acoustic beacons to set up an underwater system analogous to the satellite global positioning system (GPS) for open air systems. Recent state-of-the-art work with these long baseline (LBL) navigation systems is provided by Yoerger *et al.* [1]. Another common technique is to obtain Doppler velocity measurements using a Doppler Velocity Log (DVL), as illustrate by Whitcomb *et al.* [2]. These techniques are the solutions of choice for many types of missions in which low cost of the AUV is not a primary concern.

Errors in localization will lead to decision mistakes and failed missions. In some cases absolute localizations is less important than localization relative to some underwater features such as pipelines, cables, moored objects, or bottom objects. These cases require correct interpretation of exteroceptive sensors capable of detecting the features. The most useful underwater exteroceptive sensor for this process is sonar. Sound travels well under all conditions while light is useful only in clear water at short range. The sonar data can be formed into an image that can be processed using some of the same tools used to process camera images. The projective geometry and the interpretation of images is different so, for example, shadows are formed on the far side of objects relative to a sound source. These shadows give a great deal of information to a skilled human analyst.

There are a number of different types of sonar systems that produce images comparable to camera images. Side-scan sonar can form narrow sonar beams restricted to a plane perpendicular to the sonar array. The returns from these beams can be pieced together over the trajectory of the vehicle to give a detailed image of the sea floor. Forward looking sonars can generally see in a pie slice shaped wedge in front of the sonar. They usually have good angular resolution over a span of angles about an axis and good range resolution. The resolution in beam angles to the axis is relatively wide forming the wedge as the sound propagates.

Information from one sonar image is of limited use to the AUV. It is the repeated observation of the same objects that helps to remove disturbances from the motion estimate. It is also by tracking objects over many images that noise can be distinguished from the features of interest which can then be recognized as being

relevant to the mission goals. Thus knowledge acquired from these observations can be compared to knowledge acquired prior to and during the current mission. As such, the state of the vehicle can depend on the observations and the mission execution becomes adaptive to these observations. Missions that would be impossible to carry out even with perfect absolute localization can now be envisioned as feasible with imperfect absolute localization. This is the power of the autonomous intelligent machine.

Our system is designed to carry out target reacquisition missions at low cost and without the use of pre-set acoustic beacons as positioning aids. The use of a DVL to correct for the effects of water currents is not part of the system, for cost considerations. The current state of the art for these types of reacquisition missions is to use human divers; this is impossible or undesirable for many situations, and an AUV-based system that could attach itself to a target offers many advantages. The system will find a specified moored target using an *a priori* map of the area's features. These features can be various moored or bottom objects that happen to be in the area. The *a priori* map is made using an AUV equipped with sophisticated sensors and a side-scan sonar. The mission critical decision centers on the position of the vehicle relative to this *a priori* map. The method to make this decision is to use SLAM to build a map of the sonar observations [3, 4, 5, 6, 7] and match this map to the *a priori* map in an EKF [8, 9, 10, 11]. Several EKF estimators are run in parallel allowing multiple match hypotheses to be considered. All matching decisions are reversible in that one of the hypotheses (known as the null hypothesis) makes no matches between the features; this null hypothesis can then be copied to any of the other estimators essentially resetting them and allowing them to be matched differently.

The local map building is done in a tracking filter. This computation is both accurately and efficiently carried out using a Square Root Smoothing SLAM algorithm. The tracking filter implements the algorithm as a graph over the vehicle states and feature positions. The information tracked over a number of frames is consolidated into composite measurements and used to update the bank of EKF estimators at a low frequency. Thus problems of complexity as the size of the slam graph grows are avoided by cutting off sections of the graph periodically and feeding them as single measurements of both features and AUV pose to the EKF estimators. One gets the accuracy of the graphical square root slam model that re-linearizes all measurements about the current solution without any explosion of complexity. At the same time one can use several EKF estimators updated with these accurate measurements at low frequency to allow multiple hypotheses and reversible decisions. As the matching is also done at this low frequency, the matching algorithm can be of higher complexity.

## 2 Mission Scenario

The mission starts with a survey of an undersea region containing moored and bottom features by an AUV equipped with sophisticated navigation and side-scan sonar.

This data is analyzed by an expert and an *a priori* map of features is produced. One of the moored features is identified as the target of interest to be reacquired. The *a priori* map and chosen target are input to our AUV system and then the vehicle is released from a distance of between 100 to 1,500 m from the field (Fig. 1). The AUV travels to a point about 70 m out and dives on the field. It surveys the field and builds a SLAM map of the features. When the AUV has achieved good localization relative to the *a priori* map, it will attempt to 'capture' the target. That is, still using the features for navigation, it will maneuver to drive its V shaped grippers into the mooring line of the target. At that point it will close the grippers tightly on the line and become attached.

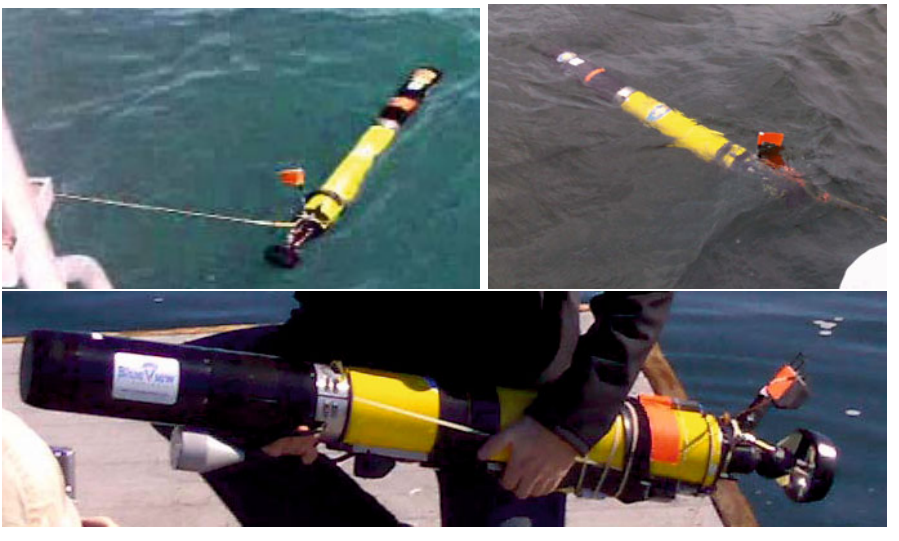

**Fig. 1** The iRobot Ranger AUV is equipped with a Blueview Blazed Array sonar in the nose section. It is launched by hand.

## 3 System Components

The AUV is a Ranger manufactured by iRobot and equipped with depth sensor, altimeter, GPS, a 3D compass, and a Blueview Blazed Array forward looking sonar (FLS) as shown in Fig. 1.

The software system consists of 7 functional modules. A robust and fast detector finds virtually all the interesting features in the sonar images with a few false positives. These are fed to a tracking filter that is able to filter out the false positives from the real objects and form 'composite measurements' out of the sonar and motion measurements over a section of the path. These composite measurements are then input to the bank of EKF SLAM estimators. These try to match the *a priori* map to the SLAM map using a special matching criteria described in a later section.

The output of these estimators as well as information on the currently tracked features is passed to the interpreter which extracts information relative to the current mission objectives. This information is requested by the mission executor and used to make the high level decisions leading to a control output sent to the actuators. Final capture of the target is done using a sonar servo PID controller.

In addition, there is a dead reckoning service which provides motion estimates to the tracking filter, the EKF estimators, the mission executor, and the sonar servo control. This dead-reckoning service uses a detailed model of the robot and environment along with two separate EKF estimators to provide estimates of the motion with and without GPS. (GPS is of no use underwater but is used to initialize the position before dives.)

## 4 Sonar Feature Detector

The Blazed Array FLS has two sonar heads, a vertical and a horizontal, giving a position in 3D for features detected in both. The field of view of each head is 45 degrees by 30 degrees. The returns are given as range and angle in the plane of the 45 degree direction. Thus measurements have a large uncertainty perpendicular to this plane, extending outward in a 30 degree wedge shape. The effective range of the sonar is 40 m for good sonar reflectors and about 20 m for normal bottom objects such as rocks. Thus to get enough detections of normal bottom objects to be of real use the robot must pass within about 5 m of the object. This makes finding the objects of the *a priori* map a challenge as currents can easily throw the AUV off by more than this amount. The cruising speed of the Ranger is about 1.2 knot. The currents encountered during our validation testing were typically more than 0.5 knot and vary with time, depth, and in the horizontal plane.

The sonar horizontal head has been modified to tilt downward at 5 degrees to give a better view of the bottom targets. The formation of vertical and horizontal images out of the raw sonar data using the Blueview SDK library takes about 180 ms of processor time per ping. The feature detection algorithm then adds about 18 ms or 10% to this time. The steps used to process the images are as follows:

1. Form a 200 x 629 image of the vertical head sonar data, (in Fig. 2 the images are rotated 90 degrees).
2. Down-sample this to 50 columns (rows in Fig. 2).
3. Find sea bottom, altitude, and relative slope using line extraction.
4. If slope (pitch) is too far from level flight stop.
5. Form a 200 x 629 image of the horizontal head sonar data.
6. Down-sample this to 50 columns (rows in Fig. 2).
7. Use the altitude from step 3 to select one out of a number of averaged background images each for a specific altitude range.
8. Low pass filter the current horizontal image into the background image.
9. Subtract the background image (noise) from the horizontal image to form two separate images, one the normalized image and the other the shadow image.
10. Segment vertical image into bottom and open water.

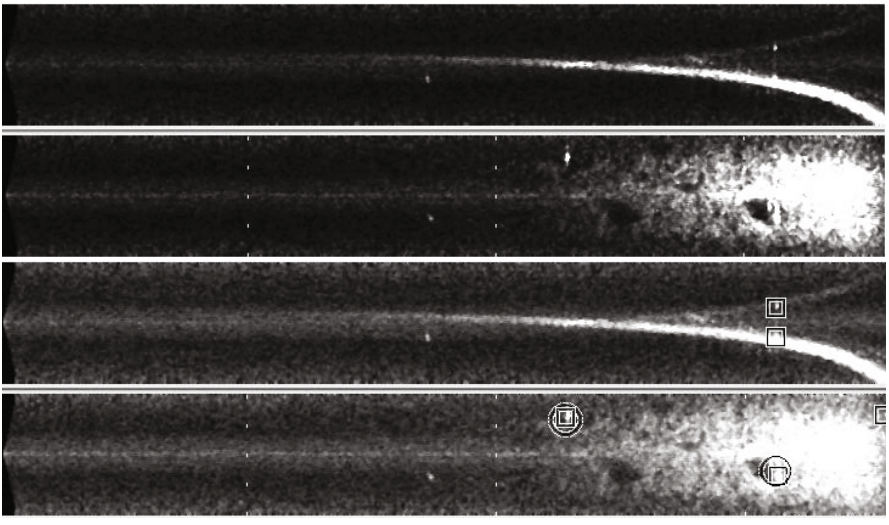

**Fig. 2** Here is an example of the vertical (upper) and horizontal (lower) sonar head images. The top pair are the input to the detection module and the bottom pair show the output feature detections highlighted as circles for shadows and squares as edges. The images are rotated to show the range increasing from right to left. The white hash marks are spaced at 10, 20, and 30 m in the horizontal images for scale reference. The polar angle is shown in the as increasing downward in the images and has a total range of 45 degrees. The bright arc in the upper images is the sea bed and would appear as a straight line if plotted in Cartesian space instead of polar coordinates as shown here. The near-field region of open water is barely seen in these images due to the AUV pitch. It would normally be a wider darker region along the right edge of the horizontal image, here seen very narrow.

11. Segment horizontal image into three regions near field open water, ranges where the bottom is sufficiently bright to see shadows, and the remaining range out to 40 m.
12. Low pass filter the image segments pixels along each bearing (column) to smooth them eliminating very small features and noise.
13. Search for edges as gradients in each segment with thresholds based on the average background image level. (This gives some adaption to different bottom types.)
14. Search for shadows (below noise floor) in bottom segment coupled with an adjacent bright area closer to the sonar from the object that created the shadow.
15. Then select for output any feature found in both heads and the strongest remaining detections in each segment.

Step 3 allows the algorithm to process differently based on the altitude and pitch of the vehicle which is important as the esonification of the bottom will be dependent on these. Thus the image can be segmented into regions of open water and sea floor. The regions of open water are much less noisy and can therefore have lower detection thresholds. The down-sampling steps, 2 and 6, are crucial to fast processing

without the loss of information. The sonar has an angular resolution of 1 degree so that 50 columns will be sufficiently fine for the 45 degree field of view. The averaged background images at each altitude are important for normalizing the image (step 9) and for detecting shadows (step 14). In step 9 the normalized pixels will be set to the difference when positive or 0 while the the shadow pixels will be set to minus the difference when this is positive or 0.

## 5 Mapping

The mapping approach utilizes the strengths of two SLAM methods each focused on separate aspects of the mapping problem. One aspect involves the tracking of feature detections and using them to improve the dead-reckoning over local map areas of 10-50 m in size. The other aspect involves piecing these local map estimates together to give an estimate of the map over the entire field and to then provide information on uncertainties to the map matching algorithm. Fig. 3 provides an aid to understanding the workings of the various parts of the mapping approach.

The feature tracking uses an accurate Square Root Smoother [12, 13] as previously described in [14, 15]. The inputs to this incremental Gaussian estimator are the sonar detections of features and the dead-reckoning estimates between the sonar measurements. The filter estimates the Gaussian Maximum Likelihood state along with a representation of its covariance. The state vector consists of the locations of the observed features and the all the poses of the AUV at these observation times. This representation allows us to delay initialization of features until we have gathered enough information on them. The initialization can then add all the previous measurements. Thus information is neither lost nor approximated. At any point the entire estimator can be re-linearized around the current state which improves consistency of the estimate. Thus the tracking filter avoids two of the problems encountered in SLAM estimation, consistency and loss of information. The trade off is increased computational complexity. This complexity is limited by the relatively small size of the local maps.

Periodically a section of the path along with all its measurements is cut out and formed into an independent estimator for just that subset of measurements. Then all variables for the intermediate pose states are eliminated by marginalizing them out forming a composite measurement with a state vector consisting of the starting and ending AUV poses and the feature locations. This mean and covariance estimate can then be used in an update of the bank of EKF estimators. The composite measurements are local maps of the features seen along a section of the robot path. These local maps are then joined using the bank of EKF estimators.

The EKF estimators have a state that consists of the AUV pose at the time of the last update and the locations of all features, both from the *a priori* map and the composite measurements. Thus the EKF SLAM filters first augment the state with the new pose and features, then do an EKF update step on the features locations and the two poses, then marginalize out the earlier pose and merge any features that were tracked between two adjacent local maps. There is no explicit prediction step.

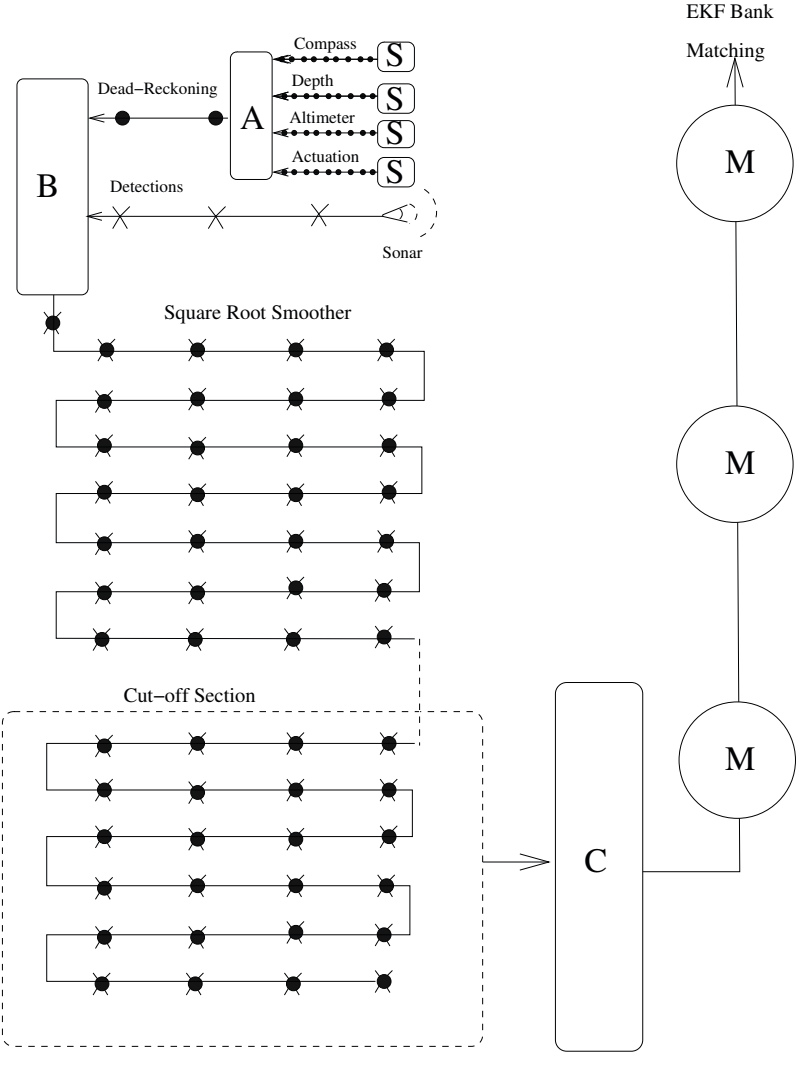


**Fig. 3** We illustrate the compression of information in the various stages of mapping. We use an EKF in block (A) to combine the motion measurements of depth, altitude, compass, and actuation from sensors (S) into dead-reckoning estimates synchronized with the sonar feature detections. These are then used in block (B) to add nodes to the square root smoother graph. This smoother performs the feature tracking function eliminating false detections and forming maximum likelihood Gaussian estimates of the remaining detections. As this graph grows, sections are periodically cut-off reducing the number of nodes in the remaining graph. The cut-off section becomes an independent smoother graph. In block (C) we form composite measurements (M) out of the cut-off section of the graph and feed it to the bank of EKF estimators for matching to the *a priori* map. These composite measurements (M) of features and poses are local maps. The GPS measurements also update the EKF Bank. The EKF Bank also perform the current estimation. The current estimate from the most aggressive EKF is used by (A) to provide predictions to (B).

Each EKF estimator of the bank, numbered 0 to n-1, can match the features observed on the local maps to the features of the *a priori* map. These matches can be different for each estimator and the EKF will compute the maximum likelihood mean given the chosen match. The EKFs can also compute the Mahalanobis distance corresponding to the match. Thus the bank of estimators allows parallel match hypotheses to be continuously evaluated and updated. Any estimator's state can be copied into another estimator. Thus incorrect matches can be copied over once a better match is found. Estimator number 0, the most conservative, allows no matching at all and on each iteration is used to reset estimator number 1 to this state. All the estimators are then matched. The matching criteria are increasingly aggressive as the estimator number increases. If this then produces a match better (more likely) than any higher numbered estimator it is copied to those estimators. The likelihood is measured using the Unseen Match Scale described in Sect. 6.

## 6 Matching

Matching correctly to the *a priori* map is the most critical decision of each mission. We base our matching algorithm on a quantity we call the Unseen Match Scale (UMS). The goal is to capture the likelihood of a given match hypothesis $h$. The use of negative information is illustrated in Fig. 4. We see that only hypothesis (c) can explain the negative information of not having seen both previously mapped features (the x's) on the recent local map. We match one prior map feature to the recently observed feature (the dot) while the other prior map feature has not yet been scanned by the sonar. The UMS quantifies this information as follows:

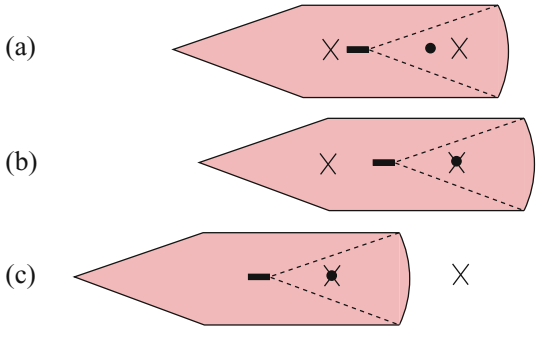


**Fig. 4** Illustration of the Unseen Match Scale. The figure shows three match hypotheses. The shaded area shows the part of the environment scanned by the sonar on a recent local map with the dotted line showing the area of the last image. The small rectangle is the AUV at the last pose, the two x's are features from the *a priori* or earlier local map, and the dot is a feature on the most recent local map. Hypothesis (a) illustrates the unmatched situation (the null hypothesis). Hypothesis (b) cannot explain why one x was not seen. Hypothesis (c) explains why we have only one dot.

$$UMS(h) = -\Lambda N + D_h + U_h \tag{1}$$

The *UMS* has three terms. The first term is proportional to the number of matched feature pairs[1], $N$, where $\Lambda$ is a parameter. The second is the Mahalanobis distance of the match as computed using the EKF covariance matrix. The third is the unseen feature energy defined as:

$$U_h = \ln(P(unseen|null)) - \ln(P(unseen|h)) \tag{2}$$

This expression is the negative of the log of the probability of all the unseen features normalized relative to the null hypothesis of making no new matches. Unseen features are features that were predicted to be in the robot's sensor field of view but were not initialized in the map. That is, these features are seen on one local map but are not seen on an overlapping local map. In Fig. 4, hypothesis (a) has three unseen features, (b) has one, and (c) has none.

We must compute the probability of the unseen features. For that we need several concepts. We form coarse grids over the area of each local map. Thus the same chain of local maps described in the last section will now be linked to a chain of grids summarizing the search for features by the sonar over the sections of path corresponding to the local maps. As the robot moves scanning the sea and sea floor with its sonar, we increment a counter in each scanned grid cell for each scan. This gives us a table containing the number of times each cell was scanned. We denote these numbers as $s_{ig}$, where $i$ is the cell index and $g$ is the index of the local grid.

We refer to the probability that the feature will be detected when the cell containing it is scanned as the feature's visibility, $v$. The feature will be initialized on the map if it is detected a number of times. We call this the detection threshold, $n_d$.

We define $Q_{fg}(h)$ as the probability of not initializing feature $f$ predicted to lie in grid cell $i$ of grid $g$. It can be computed from a binomial distribution.

$$Q_{fg}(h) = \sum_{j=0}^{n_d-1} \binom{j}{s_{ig}} v^{(s_{ig}-j)}(1-v)^j. \tag{3}$$

The sum is over the number of times the feature may have been detected without being initialized.[2] We can form these sums for every feature and every local map on which it was not initialized. We can then compute the unseen feature energy of eq. (2) as the sum of these $Q_{fg}$ over all the unseen features.

[1] This is the criteria used in the standard Joint Compatibility Branch and Bound algorithm[16] along with a threshold on the Mahalanobis distance.

[2] The cells $i$ of eq. (3) are inferred based on the match of $h$. This is done by computing the new state generated by making the match. The new feature states will then imply a new transform to the local grid frames based on the features seen on the grids. This then is used to transform the unseen features to the grids and finally find the cell $i$ that the feature falls in. See the appendix for other details.

$$-\ln(P(unseen|h)) = -\sum_{fg \varepsilon unseen} \ln Q_{fg}(h) \tag{4}$$

Notice we get a contribution here that can be directly traced to a particular feature on a particular local map. High values for the unseen feature energy contribution, $-\ln Q_{fg}$, indicate a prime candidate feature $f$ for matching to features on the corresponding local grid $g$. This allows directed searches of the match hypothesis space.

Matched features are no longer unseen. Thus, the sum over unseen features will not include terms from the grids that had any feature matched to the feature we are considering. That gives a gain for matching that is not arbitrary or heuristic, but rather is precisely computed based on the simple statistics accumulated on the grids. The only parameter selected by hand is $\Lambda$. A probabilistic interpretation for $\Lambda$ is presented in [17]. For the results presented here, good results have been obtained with $\Lambda$ either set to zero or given a small value.

The complexity of calculating the $Q_{fg}$ will be less than the number of features times the number of grids. Only features that fall on the local grid need to have $Q_{fg}$ calculated. Thus the complexity is dependent on the path of the robot and the success of the matching. If the complexity were to become a problem, one could address this by considering only matches between features on the most recent grids and the rest of the grids. Thus we would give up trying to make matches that we failed to make on earlier iterations. That would then result in constant complexity as the map grew. Typically the complexity is not a problem so long as the matching is working properly. In that case the features are merged and no longer considered as unseen. Thus the number of $Q_{fg}$ to be computed is reduced.

We must estimate the feature's visibility, $v$. We do this by examining the grids on which the feature was seen. We divide the number of detections by the sum of the $s_{ig}$ for the feature.[3] We sum the $s_{ig}$ using a Gaussian weighting over a window around the feature's predicted cell.

While the unseen feature energy, $U_h$ of eq. (2), normally decreases as matches are added to the hypothesis, it does not *always* decrease, because the new state for the features given the match hypothesis $h$ may imply a transformation which causes more overlap of local grids and a higher value for some $Q_{fg}$. Hence, some matches result in a decrease in the value of unseen feature energy.

All matches result in a decrease in the first term of the UMS eq. (1). The decrease in value may more than offset the inevitable increase from the second term of the UMS. The match with the lowest UMS is tested to see if it is ambiguous. If not, the match is made. In order to test ambiguity, the difference between the UMS for all matches, including null, and the UMS for the best match is checked. We threshold this difference with a parameter which we call the ambiguity parameter. We will make the match if this ambiguity parameter is less than the difference for all matches that conflict with the candidate match. Matches that agree with the candidate match, that is they give the same match pairs for all features in common, are

[3] In [18] they use the negative information of not observing a feature to remove spurious measurements from the map. This is another example of the usefulness of negative information. We could, in the same spirit, remove features with very low values of $v$.

not conflicting.[4] The null hypothesis is considered conflicting as are hypotheses that for some feature in common give a matched feature that is both different from and local to the matched feature of the best hypothesis. By local we mean that the two features are on the same local grid. Local features are always considered impossible to match to one another. If they were the same the local map builder should have matched them.

Using this measure we can say that the conflicting hypotheses have an energy or minus log likelihood that is higher than the one chosen by more than the ambiguity parameter.

By adjusting this ambiguity parameter we are able to set levels of aggressiveness in matching. We utilize this when running multiple hypotheses in the bank of parallel EKF SLAM estimators. Normally the more aggressive hypothesis will have a lower total energy as calculated by the accumulated total of the UMS for all matches merged. Occasionally the aggressive match will be wrong and then a more conservative hypothesis may eventually be able to find the true match lowering its energy below the aggressive one. When this happens we can copy the lower energy solution to the aggressive matcher's estimator and continue. In that way the aggressive matcher always has the most likely solution given the data and the implied search of all the parallel run estimators.

## 7 Field Testing

The system has been extensively tested and refined over 11 sea trials starting in 2006 each lasting from 2 to 3 weeks. We started in benign environments with unrealistic highly reflective moored targets that were easily detectable in the FLS from 60 m and progressed to less ideal sea conditions, normal moored targets, and bottom features including natural features not detectable past 20 m range. Finally control and attachment to the target were added.

The UMS matching algorithm was first validated in an A/B comparison test in St. Andrews Bay, Florida in June 2007 in which the UMS and Joint Compatibility Branch and Bound (JCBB) criteria were alternately used over 18 trials on a field of strong reflective moored targets. The bay has a tidal current that gave the AUV significant dead-reckoning errors on most trials. The AUV was released about 100 m from the field and from 8 directions. The AUV made a single pass of the field and had to decide on the match before finishing the pass. The trials were paired so as to give nearly equal starting conditions for the two algorithms. A success was defined as achieving a correct match to the field followed by the vehicle's successfully exiting the field and then re-approaching it again in a movement of mock capture toward the correct target. No actual attachment was done. The results of the live test are summarized in table 1. The difference in the frequencies is positive by 1.41

[4] The reason we need to introduce the concept of not conflicting is that a hypothesis that is correct but not complete might have a UMS very close to the correct and complete hypothesis. This is true if the left out pair(s) of the incomplete hypothesis do not change the energy by very much.

standard deviations. This gives a 91% significance to the difference and indicates that the UMS did outperform the simpler JCBB matching criteria.

To give an unambiguous comparison in the live test, we only used one match hypothesis. We later ran the data off line with 4 hypotheses and got the improved result shown in the table[5]. Of the four missions it could not match, two were UMS runs and two were JCBB runs. This gives some measure of the fairness of the random elements of each run.

**Table 1** In the 2007 tests the Joint Compatibility criteria, JCBB, uses a threshold on the Mahalanobis distance of the multiple pair match and chooses the most pairs. This was compared to the UMS criteria and the difference in rates was considered as proof that the UMS performed better. In the 2008 tests we used both moored and bottom targets which were detectable from between 20 and 40 m. In the March 2009 test there was no matching done as we tested the control to attachment on a single moored target in a test pond. In the June 2009 tests we added better modeling and estimation of currents along with better feature modeling.

Selected Test Results

| Match Criteria | Runs $n$ | Successes | Frequency | $\sqrt{s_n^2/n}$ |
|---|---|---|---|---|
| | | | | |
| Bright Targets - June 2007: | | | | |
| UMS | 9 | 6 | 67% | 17% |
| JCBB | 9 | 3 | 33% | 17% |
| UMS - JCBB | | | 33% | 24% |
| UMS Multi-hypothesis | 18 | 14 | 78% | 10% |
| | | | | |
| Normal Targets - June 2008: | | | | |
| UMS Multi-hypothesis | 9 | 3 | 33% | 17% |
| | | | | |
| No Current - March 2009: | | | | |
| One Feature | 17 | 17 | 100% | |
| | | | | |
| Normal Targets - June 2009: | | | | |
| UMS Multi-hypothesis (all) | 26 | 17 | 65% | 9% |

Next the detection of features in the sonar images was substantially improved from a simple intensity detector in the horizontal image to the detection algorithm described in Sect. 4. We added a PID control and grippers to the robot to enable actual capture of the mooring lines of the target. We tested in Narragansett Bay, Rhode Island in June 2008, on a field consisting of various man-made and naturally occurring objects on the sea bottom. Again the bay had a significant tidal current which gave us substantial dead-reckoning errors. We did nine capture runs. Two of the

[5] On one of the failed multi-hypothesis runs the solution was switched to the correct match after the robot had returned to the surface, too late to be considered a success.

runs resulted in the AUV hitting the mooring line and breaking off the gripper arm, which we considered a success for the software system. We captured the mooring line on the ninth run of the day for an overall success rate of 33%.

We then improved the final target capture with the addition of sonar servoing of heading and altitude for the PID control. In March 2009, we performed tests in a man made test pond to evaluate the terminal homing behavior with one target and no current. In 17 missions in which the SLAM algorithm locked on to the target, the vehicle successfully latched onto the line all 17 trials. This provides validation for the final capture phase.

The overall robustness of the system was further improved via two further refinements to the matching algorithm that are described in the appendix. We also improved the model and estimation of current variation with depth. In June of 2009 the entire system was tested in the Gulf of Mexico. 15 objects were placed with a nominal 12-15 m spacing and forming three parallel lines. Three of the objects were moored while the others were bottom objects. Over a two week period the system was evaluated by running 26 missions of which 17 resulted in attaching to the chosen target. We found that the success rate depended strongly on the initial approach to the field. By having the robot travel in a straight heading towards the center of the field along the direction of the current we achieved 16 of 18 successes with the other 8 trials using other strategies. Surface currents ranged from 0.4 to 1.0 knots.

Of the 9 failed missions 4 had no match and 5 miss-matched. No mission managed to make a successful second attempt after getting a new GPS fix but 2 timed out after making a correct match. The miss-matches typically occurred after initially missing the field and accumulating large dead-reckoning errors. Five of the successful runs had initial miss-matches that were corrected before capture was attempted. In all failures there were too large dead-reckoning errors on reaching the field. The dead-reckoning errors were attributed to disturbances in the water which can not be estimated and compensated for. We have succeeded in developing a model of the variation of ocean currents with depth but this only can remove the disturbances that matches the model. No variation in the horizontal plane can be estimated. The model parameters are estimated on the fly during GPS pop-ups during the initial approach and prior to the final dive into the field.

## 8 Conclusions

This paper has presented a summary of a field deployed AUV navigation system that achieves a high level of autonomy to perform a challenging real-world mission. We have worked methodically towards creating a robust system to reliably reacquire underwater targets, reducing the danger to the manned personnel that typically perform these missions. We first broke the problem into specialized functional modules and optimized each separately. By adding new functionality and refinements based on repeated testing at sea, incremental improvements have accumulated to give the AUV a robust autonomy. The value of the system has been comprehensively demonstrated in numerous field trials over a three-year period. Continued

testing is in progress to improve the target capture success rate of the overall system for increasingly difficult ocean conditions.

**Acknowledgements.** This work was funded by the Office of Naval Research under grant N00014-05-1-0244, which we gratefully acknowledge. We would also like to thank the other team members, iRobot, Blueview, Naval Postgraduate School, SeeByte, and NSWC-PC for their assistance.

## References

1. Yoerger, D., Jakuba, M., Bradley, A., Bingham, B.: Techniques for deep sea near bottom survey using an autonomous underwater vehicle. International Journal of Robotics Research 26(1), 41–54 (2007)
2. Whitcomb, L., Yoerger, D., Singh, H., Howland, J.: Advances in Underwater Robot Vehicles for Deep Ocean Exploration: Navigation, Control and Survery Operations. In: The Ninth International Symposium on Robotics Research. Springer, London (2000) (to appear)
3. Williams, S., Dissanayake, G., Durrant-Whyte, H.: Towards terrain-aided navigation for underwater robotics. Advanced Robotics-Utrecht 15, 533–550 (2001)
4. Ribas, D., Ridao, P., Neira, J., Tardos, J.: Slam using an imaging sonar for partially structured underwater environments. In: Proc. of the IEEE International Conference on Intelligent Robots and Systems (IROS 2006). IEEE, Los Alamitos (2006)
5. Leonard, J.J., Carpenter, R., Feder, H.J.S.: Stochastic mapping using forward look sonar. Robotica 19, 467–480 (2001)
6. Tena, I., de Raucourt, S., Petillot, Y., Lane, D.: Concurrent mapping and localization using sidescan sonar. IEEE Journal of Ocean Engineering 29(2), 442–456 (2004)
7. Fairfield, N., Kantor, G.A., Wettergreen, D.: Towards particle filter SLAM with three dimensional evidence grids in a flooded subterranean environment. In: Proceedings of ICRA 2006, May 2006, pp. 3575–3580 (2006)
8. Dissanayake, M.G., Newman, P., Clark, S., Durrant-Whyte, H., Corba, M.: A solution to the simultaneous localization and map building (SLAM) problem. IEEE Transactions on Robotics and Automation 17(3), 229–241 (2001)
9. Newman, P., Leonard, J., Rikoski, R.: Towards constant-time SLAM on an autonomous underwater vehicle using synthetic aperture sonar. In: Proc. of the International Symposium on Robotics Research, ISRR 2003 (2003)
10. Leonard, J., Rikoski, R., Newman, P., Bosse, M.: Mapping partially observable features from multiple uncertain vantage points. IJRR International Journal on Robotics Research 7(3), 943–975 (2002)
11. Hahnel, D., Burgard, W., Wegbreit, B., Thrun, S.: Towards lazy data association in SLAM. In: The 11th International Symposium of Robotics, vol. 15, pp. 421–431. Springer, Heidelberg (2005)
12. Dellaert, F.: Square root SAM: Simultaneous location and mapping via square root information smoothing. In: Robotics: Science and Systems (2005)
13. Dellaert, F., Kaess, M.: Square root SAM: Simultaneous location and mapping via square root information smoothing. International Journal of Robotics Reasearch 25(12), 1181–1203 (2006)
14. Folkesson, J., Leonard, J., Leederkerken, J., Williams, R.: Feature tracking for underwater navigation using sonar. In: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2007), pp. 3678–3684 (2007)

15. Folkesson, J., Leederkerken, J., Williams, R., Patrikalakis, A., Leonard, J.: A Feature Based Navigation System for an Autonomous Underwater Robot. In: Field and Service Robots, pp. 105–114. Springer, Heidelberg (2008)
16. Neira, J., Tardós, J.D.: Data association in stocastic mapping using the joint compatibility test. IEEE Transaction on Robotics and Automation 17(6), 890–897 (2001)
17. Folkesson, J., Christensen, H.I.: Closing the loop with graphical SLAM. IEEE Transactions on Robotics and Automation, 731–741 (2007)
18. Montemerlo, M., Thrun, S.: Simultaneous localization and mapping with unknown data association using FastSLAM. In: Proc. of the IEEE International Conference on Robotics and Automation (ICRA 2003), vol. 1, pp. 1985–1991 (2003)
19. Singh, H., Roman, C., Pizarro, O., Eustice, R.: Advances in high-resolution imaging from underwater vehicles. In: International Symposium of Robotics Research, San Francisco, CA, USA (October 2005)

## Appendix

A few refinements must be made to the simple formula of eq. (3) in the actual implementation. First, the prediction of the feature location and the grid number $s_{ig}$ are uncertain. We therefore use a Gaussian window over the grid cells and sum the contributions from adjacent cells near the predicted feature location. This gives a more continuous scale. Second, the details of the local map formation are such that features can be initialized by being detected $n_d$ times in total on adjacent local maps. So we need to sum over the local maps before and after as well.[6]

Third, a minor adjustment needs to be made to avoid numerical problems such as extremely small probabilities. The $Q_{fg}$ are limited to avoid their becoming too small. This is done by performing the following transformation on the values computed as described above and in the main text:

$$Q_{fg} \leftarrow p * Q_{fg} + (1 - p) \tag{5}$$

We call $p$ the model probability as it is the probability that our model of the features is correct. We used $p = 0.99$.

Two additional refinements were added to the algorithm this year. A fourth refinement is to accumulate grids at a number of depths and then use the grid at the observed feature's depth of the computation. A fifth refinement accumulated grids at three ranges from the robot: 0-8 m, 8-20 m, and 20-40 m. This adjustment allowed us to model the fact that some features are not visible from the longer distances.

[6] This is straightforward but the resulting formulas are too complex to include here due to space limitations.