Three Dimensional Evidence Grids for SLAM in Complex Underwater Environments

Nathaniel Fairfield * George Kantor David Wettergreen
Carnegie Mellon University

Abstract

We describe the initial results in constructing the simultaneous localization and mapping system for exploration of flooded subterranean environments like mines and caves for an autonomous hovering vehicle which will explore flooded cenotes (sinkholes) in Mexico. In May 2005, a preliminary data set was collected in the Zacatón cenote using a drop sonde equipped with a circular array of 32 pencil beam sonar transducers. This paper presents the results of applying an evidence grid and particle filter localization system to the Zacatón data, as well as some simulations that investigate the effectiveness of different sonar geometry configurations.

1 Introduction

Underwater vehicle navigation where GPS is not easily accessible is still dominated by two basic approaches: direct position measurement with a surveyed long baseline (LBL) network of acoustic transponders, and dead-reckoning with a combination of inertial sensors and doppler velocity sensors. The first is unsatisfactory because of the effort and infrastructure involved in deploying and calibrating the acoustic baseline system, and the second is unsatisfactory because of the gradual accumulation of error. Both compromise the autonomy of the underwater vehicle. For vehicles without access to GPS, simultaneous localization and mapping (SLAM) provides a promising alternative.

*Email: than@cmu.edu

Most SLAM strategies depend on the extraction of features from sensor data, however in the underwater domain there are not many sensors which are capable of providing the resolution necessary to resolve and recognize features, though there has been work with using tunnel cross-sections, or slide images, as features [2]. In the case where there are free floating artificial features, scanning sonars have been shown to have high enough resolution to support feature-based SLAM [12]. Alternatively, in clear water with good lighting, vision-based feature detection and sonar have been combined [11]. However, many underwater environments are characterized by large monotonous feature-poor regions, where the most promising work has been with Synthetic Aperture Sonar (SAS) to support range-and-bearing SLAM [6]. Finally, there is the mine-mapping work by [9] done in unstructured (but not underwater) environments, using scan matching with a laser range finder to recover the 2D vehicle pose from which the full 3D map is reconstructed.

DEPTHX is an autonomous hovering vehicle which will explore flooded cenotes (for a full description of the DEPTHX project, see [7]). In the DEPTHX application, the array of pencil-beam sonars provides a constellation of ranges around the vehicle, but lacks the resolution and point density of a laser scanner, which makes scan matching or traditional corner and edge detection a difficult proposition. Furthermore, the deeper portions of the cenotes are completely unexplored and may have unexpected geometries, which makes it even more difficult to design feature detectors. For these reasons, a data-driven approach is appropriate, such as evidence grids [5].

While 2D evidence grid-based SLAM is well established in the indoor mobile robot domain, it has been found to have limited applicability in truly 3D environments – largely because the 2D map simplification is only suitable in "two and a half" dimensional environments. The difficulty in generalizing the evidence grid approach to full 3D centers on the multiplication of the computational cost of accessing, modifying, and storing the map due to the third dimension. This computational cost figures highly in the choice of model for the sonar beams – simpler is cheaper. Of course, the sonar beam model need not be the same when evidence is inserted into the map as when a query is made, and in fact [5] found that using different insertion and query beam patterns can improve performance.

In this paper, we present a 3D evidence grid representation, in which space is divided into a grid of cubic volume elements, or voxels, which contain the occupancy evidence gathered from sensors. We describe an implementation that supports the use of 3D evidence grids in a large (cubic kilometer) environment, fast insertion of sonar information into the map, as well as very fast querying.

Once a map of Zacatón is created, it is necessary to demonstrate that this map can be combined with measurements from the sonar array to provide an accurate estimate of vehicle position. Due to the uncertainties in sonar measurements and vehicle motion models, we choose to address the localization problem using a Bayesian framework.

Particle filtering, also known as Bayesian bootstrap, condensation, and the Sequential Monte-Carlo method, provides a proven implementation of Bayesian filtering for systems whose belief state, process noise, and sensor noise are modelled by arbitrary probability density functions (PDFs). One common variant is the Sampling/Importance Resampling (SIR) filter of [3], which will be used here.

In Section 2 we describe the May 2005 field expedition to Zacatón, as well as some of its salient features. Next, in Section 3 we turn to a brief description of 3D evidence grids and our implementation. Similarly, Section 4 provides a description of our implementation of an SIR particle filter for localization. The next two sections describe the tests of the localization system on Zacatón sonar data (Section 5), and the performance of various simulated sonar configurations (Section 6). Finally, we close with future work and conclusions.

2 Exploring Zacatón

The Zacatón cenote (sinkhole) in Tamaulipas, Mexico, is roughly a truncated cone 110m wide, and at least 300m deep. Prior to the May 2005 DEPTHX expedition it was unmapped except for wire drops to establish the depth. Zacatón is the deepest of a series of hydrothermal cenotes in the region, but it alone contains floating reed islands, called zacates. Zacatón also produces microbial mats in the photic zone and geochemical features which make it an excellent testing ground for the DEPTHX vehicle.

In order to "get our toes wet" and learn a little more about Zacatón, the DEPTHX team built a sonar test platform, called the DropSonde (see Figure 3). In addition to a data logging computer, the onboard sensors include 32 spirally arranged sonars, a Honeywell HG2001AC RLG IMU, and three depth sensors. The sonars were fired at about 2Hz, and have a 10cm precision. The IMU has about a 1m/s accelerometer drift rate, which precludes its use as a dead-reckoning navigation system, but only a 0.1 degree/hour gyro drift rate, giving use excellent heading and attitude information. The depth sensors had individual accuracies of about 10cm, and were recalibrated to zero at the beginning of each drop. Due to the quality of the IMU and depth sensors, only the x and y coordinates of the vehicle were unknown. This situation will hold for the final DEPTHX vehicle, so our SLAM implementation concentrates on trying to estimate the vehicle's x and y position, but in a fully 3D environment.

During a weeklong field expedition in May 2005, the DEPTHX team dropped the DropSonde progressively down to 200m, the maximum depth rating for several of the components. The DropSonde was lowered on a winch from a stabilized barge, and the locations of the seven "drops" were recorded with surveying equipment, which together with the data from the onboard sensors allowed the sonar data to be reg-

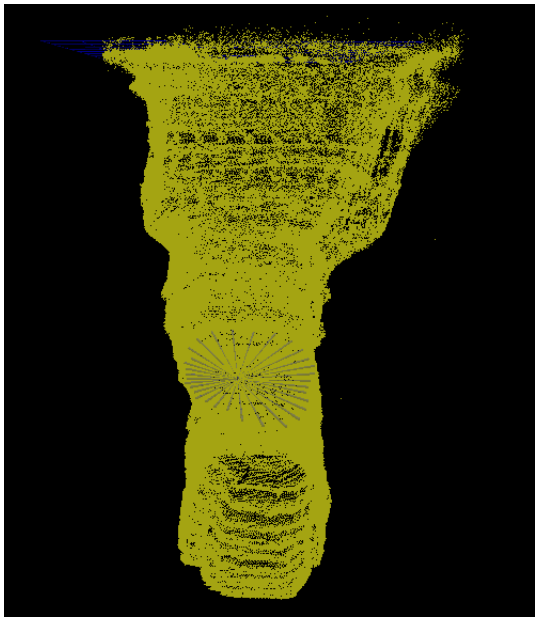

Figure 1: A north-facing pointcloud view of the first 200m of Zacatón – basically a vertical cylinder.

istered. By spinning the DropSonde during its slow ascent and descent, we acquired excellent sonar coverage of the walls. For the first 250m, the cenote is remarkably smooth and cylindrical, although there are several shelves (see Figure 1). In order to see deeper into the cenote, we rotated the DropSonde by 90 degrees so that the sonar ring was oriented vertically, and thus were able to collect datapoints at a depth of about 280m when the DropSonde was at 200m. These deepest datapoints are sparse and hard to interpret, but there seems to be a large shelf sloping down from the south-east at 270m. This could be the first indication of a horizontal tunnel, which is a common feature in this type of geological formation.

3 Evidence Grids

In principle, a 3D evidence grid is based on the same set of concepts and Bayesian update equations as the classic 2D evidence grids which have been so well described in the literature (see [5] [8]). The general idea is that each value in the 3D evidence grid

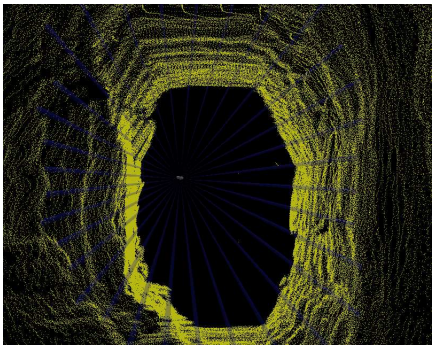

Figure 2: A downward-facing pointcloud view (north is up) Zacatón at about 200m depth. The vehicle model is to scale, and shows the sonar beams.

represents the probability that the corresponding volume in the real world is occupied. Usually, this probability is stored as the log likelihood for compactness, numerical stability, and computational efficiency. For an example voxel map customized for visualization purposes, see Figure 4.

3D evidence grids also must support the operations for the insertion and querying of evidence. The basic operation on the 3D evidence grid is to use transformation matrices to calculate the coordinates of a point in space relative to the vehicle's current position. However, using the full matrix operations is too computationally expensive for operations such as filling in evidence cones or simulating ranges by tracing lines. Many of these tasks can be efficiently decomposed into raster operations, which can be performed by a 3D variant of the classic 2D Bresenham line drawing algorithm. Bresenham 3D is extremely fast and uses integer arithmetic, so it is well suited for use with the discrete evidence grid.

3.1 Dual-resolution Maps

Memory usage is one of the major challenges of using a 3D evidence grid. Using a single byte to represent the evidence in voxels half a meter on side, a half-kilometer cube would consume a gigabyte of memory (1024^3 bytes). In many environments, and particularly in our underwater domain, much of the volume is wasted – either it is unreachable and will

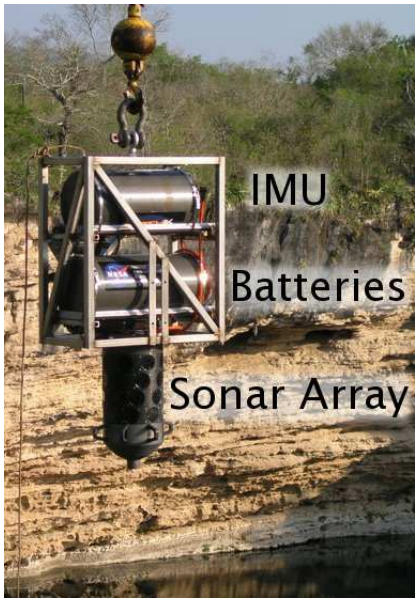

Figure 3: The Dropsonde as it is being lowered into Zacatón.

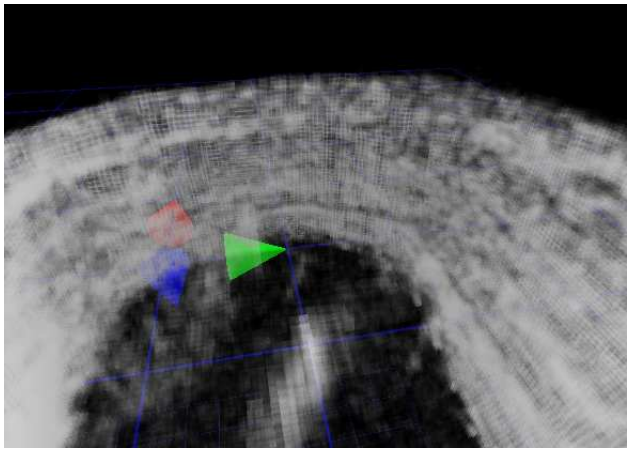

Figure 4: A 3D evidence grid. For visualization purposes, only positive evidence is represented as an increase both in brightness and opacity. This map was constructed with the data from four drops and is about fifty meters deep. The blue grid spacing is 10 meters.

never be explored, or it is very well explored and completely empty (such as the central water column in Zacatón, see Figure 2).

In order to address these issues, we have implemented a dual-resolution map data structure. There are two levels of resolution: coarse (2m) and fine (0.25m). The coarse map is completely instantiated as a normal evidence grid, but the fine map is only instantiated in chunks as needed. Each fine chunk (or submap) represents the same spatial region as a single coarse voxel. The sensor information is inserted into the map using the following method:

- First, the negative evidence of the beam is inserted. As long as no submaps have been instantiated, the negative evidence is inserted into the coarse maps. If a submap has been instantiated, the negative evidence is inserted into the submap according to the beam model.
- Next, the evidence at the end of the beam is inserted. In this case submaps will be created if they do not already exist, and the evidence is inserted into them according to the beam model.
- Any time a submap is updated, the corresponding coarse voxel is added to a "dirty" list. During the cleanup pass, all dirty coarse voxels are set to be the average value of their submap – if this value is below a certain threshold, the submap is deemed to be empty and is discarded.

The dual-resolution map data structure usually converges to large regions of unambigiously empty or unexplored space (represented in the coarse map), that is surrounded by a thin crust of high-resolution submaps that provide excellent surface detail. In a loose sense, instead of representing the entire volume in high resolution, we are just representing the surface area – the resulting savings in memory footprint are significant.

This is clearly a simplified and ad-hoc implementation of an octree-like data structure. The difficulty of implementing a full octree is that there is a large amount of computational overhead involved in the transformation between resolutions (almost all of the linear algebra must be redone), as well as the overhead in maintaining consistency between resolutions (the cleanup phase described above).

4 Particle Filter Localization

The goal of our particle filter is to estimate x_j, in this case a vector which contains the X and Y coordinates of the vehicle, from the measurements r^j_{meas}. The evolution of the state can be described by the (possibly non-linear) function

$$x_j = f_j(x_{j-1}, u_{j-1}, v_{j-1})$$

where u_j is the control input and v_j is identically and independently distributed (iid) process noise. Of course, we don't actually know f_j, but we can approximate it with $\hat{f}_j$. Likewise, the measurements r^j_{meas} are described by

$$r^i_{meas} = h_j(x_j, n_j)$$

where n_j is iid measurement noise, which is approximated by our measurement model as $\hat{h_j}$.

The standard Sampling Importance Resampling (SIR) filter includes the following steps:

Sampling This is also known as the prediction stage. During the prediction stage, we use our approximation of the process $\hat{f}_j$ to predict the new position x_j. In practice, this means applying $\hat{f}_j$ (every time with a different noise variable v_j) to each particle. The distribution of particles is now an approximation of the prior PDF of the vehicle position.

Importance Weighting During this stage, also known at the update stage, we use the actual sonar measurements r_{meas} and one of the weighting metrics given above to calculate weights w_j for each particle. The weights of the particles now approximate the importance density of the vehicle position.

Resampling We then merge these two distributions to yield the posterior PDF of vehicle position. Using the $O(n)$ algorithm described in [1], we sample the particles according to their normalized weights. For example, if a single particle has as much weight as all the others combined, it will make up 50% of the new set of particles.

We experimented with three simple methods for deriving a position estimate from the particles. The first was to use the position of the particle with the maximum weight – this can be thought of as one way of approximating the mode of the probability distribution represented by the particles. The second method was to use the mode of the positions of the particle cloud, which was approximated by constructing histograms of the particle positions in the X and Y directions, and then selecting the bin with the largest number of particles. This method was sensitive to the number of histogram bins. The last method was to take the mean of the positions of all the particles. All three approaches worked reasonably well, but the mean appeared to consistently produce the lowest position estimate error (see the results in Section 5.1).

4.1 Sonar Beam Model

The sonar beam model is the method by which a single range measurement, which is what we get from a sonar sensor, is inserted into the evidence grid. There are several methods which can be used to construct a beam model. First, we could attempt to derive the model from physical first principles, such as the sonar transducer beam model, detection threshold, and the sonar equations [10]. Alternatively, if we had a perfect map of the environment, we could use it to *learn* the beam model that resulted in the map that closest approximated the true map [5]. Finally, we could go with the simplest reasonable approximation – a cone with a cap (see Figure 5). The cone itself would be negative evidence, since any obstacle within the sonar beam would have returned an echo. At the end of the cone, there is probably an obstacle, unless the range was spurious, in which case positive evidence should be entered in the disc-shaped cap at the end of the cone.

Ultimately, the success of a beam model can only be evaluated based on the quality of the localization solution resulting from the system as a whole. Since the entire system must share computational resources, this means that a beam model which is simple and fast can yield overall better performance, because it allows the system to spend more resources

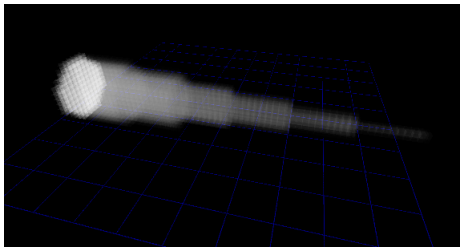

Figure 5: A low-resolution sample of the cone beam model, with a 6 degree beamwidth.

on localization (for example, more particles in the particle filter). In the test results described in Section 5.1, we simply used the cone beam model (see Figure 5), but varied the beamWidth, positiveEvidence, and negativeEvidence parameters in an attempt to find the optimal values for minimizing localization error.

4.2 Range Simulation

There is a range of physical fidelity which we could use to approximate sonar ranging. The most ambitious would be to simulate the propogation of the sonar pulse through the water, including echoes off of hard surfaces, in order to generate a time versus acoustic energy plot, similar to the signal processed by the actual sonar electronics. Simplifying this model a bit by ignoring the echoes, a similar time versus energy plot could be constructed simply by tabulating the evidence at a particular range within the sonar cone. Both of these methods are expensive, and unecessarily accurate given the level of approximation in other areas of the system. Furthermore, range simulation, or querying is *the* core operation for particle filter-based SLAM, and should be as fast as possible.

4.2.1 Ray Tracing

A simpler method for querying a sonar range from the 3D evidence grid is to trace a single ray from the origin of the sonar query until some threshold or other terminating condition is met. A more sophisticated terminating condition might be desirable if the evidence grid is sufficiently noisy that there is a significant risk of false returns.

We need to find a method for evaluating each particle in order to compute the weights $w_j (j = 1...N_{particles})$ for the SIR filter. For each particle x_j and range measurement $r^i_{meas} (i = 1...N_{sonar})$, we want to compute the probability $p(x_j|r^i_{meas})$, which can be rewritten as:

$$p(x_j|r^i_{meas}) = \frac{p(x_j)p(r^i_{meas}|x_j)}{p(r^i_{meas})}$$

using Bayes rule. On the RHS, all the particles are (initially at least) equally likely, so for the purpose of assigning relative weights, we can ignore the $p(x_j)$, and we can also discard $p(r^i_{meas})$ by normalizing, we are left with:

$$p(x_j|r^i_{meas}) \propto p(r^i_{meas}|x_j)$$

We are now left with the task of finding $p(r^i_{meas}|x_j)$ from the values we actually have, namely r^i_{sim} conditioned on x_j. If we assume a Gaussian error model for our sonar range-finders (see Figure 6), then we see that:

$$p(r^i_{meas}) = (2\pi\sigma^2)^{-\frac{1}{2}} e^{\frac{-(r^i_{sim}-r^i_{meas})^2}{2\sigma^2}}$$

We then compute the weights w_j by normalizing the product over i of $p(r^i_{meas}|x_j)$:

$$w_j = \frac{\prod_{i=1}^{N_{sonar}} p(r^i_{meas}|x_j)}{\sum_{j=1}^{N_{particles}} w_j}$$

Taking the logarithm of both sides shows that maximizing this weight metric is very close to minimizing the intuitive sum squared error metric:

$$\log w_j = \frac{N_{sonar}}{\sqrt{2\pi\sigma^2}} + \sum_{i=1}^{N_{sonar}} \frac{-(r^i_{sim} - r^i_{meas})^2}{2\sigma^2}$$

.

4.2.2 Point Correlation

Another method, described in [4], generates an error metric for each particle without actually simulating any ranges whatsoever. Instead, we use the actual sonar ranges r^i_{meas} and find the endpoints of the

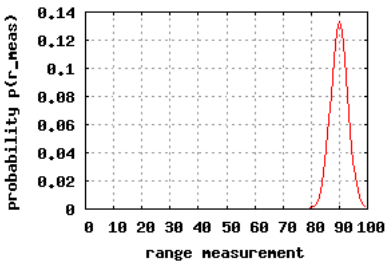

Figure 6: An example sonar range finder error model used in the ray tracing method.

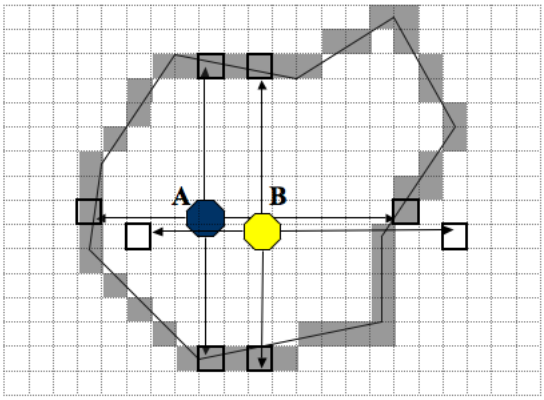

Figure 7: According to the point correlation metric, particle A has twice as much correlation with the evidence grid (positive evidence is show as gray) as particle B.

sonar beams $c_{i,j}$ by projecting the range from each of the particle poses x_j (see Figure 7). In our case, we have excellent attitude from the IMU and depth information from the depth sensors , so only x and y are unknown, and we can actually store the deltas for each sonar beam, and reuse them for each particle, resulting in an additional speedup. We then compute the correlation of the endpoints $c_{i,j}$ by examining the map value at that point. This approach ignores visibility constraints (nearer obstacles do not interfere), and relies on the sensor error being properly accounted for in the map (rather than with a sensor model as in the ray tracing approach), but is much less computationally intensive.

In order to come up with weights for the particles using the Point Correlation method, we follow the same line of reasoning as given before. In this case, however, we are trying to find $p(r^i_{meas}|x_j)$ from our correlation value at the point $c_{i,j}$, which is found by projecting r^i_{meas} from x_j. Since the map value at $c_{i,j}$ stores the log likelihood ($\log \frac{p(c_{occupied})}{1-p(c_{occupied})}$) that that cell is occupied,

$$p(r^i_{meas}|x_j) = \frac{e^{Map[c_{i,j}]}}{e^{Map[c_{i,j}]}+1}$$

and our desired probability $p(x_j|r)$ is simply the product of the probabilities, or the exponent of the sum of the map values at each $c_{i,j}$, and once again we normalize these sums in order to use them as weights.

5 Localization Tests

As described above, we collected sonar range data from Zacatón by lowering the DropSonde on a line. Since all the drop locations were surveyed in, all of the data from the multiple drops could be registered and merged. While it is possible and even likely that there is error in the registration between drops, there is very little error within a drop. This is because the DropSonde IMU gyro drift rate is negligible, and sources of depth error, such as varying atmospheric pressure, were minimal during the approximately two hours required to perform a drop. Due to the very low error in the IMU heading and attitude, as well as in the depth sensors, we were able to limit ourselves to distributing the particles in the only unknown dimensions, x and y.

In order to test the performance of the localization system on the Zacatón data, we first constructed maps using a subset of the drops, namely the first 50 meters of four different drops. We then used an excluded drop as ground truth, running the localization algorithm and comparing its estimated position with the known vehicle position. During the map construction phase, we varied the parameters of the sonar beam model, including the values of PositiveEvidence, NegativeEvidence, and BeamWidth.

Furthermore, we also experimented with various map resolutions, ranging from 2m to 0.25m. During the localization phase we also tried several variations in the number of particles.

In an attempt to verify an iterative exploration strategy, in which the vehicle will need to rely on the map constructed from the previous drop in order to localize in preparation for new exploration, we also constructed the map using a single drop (two passes), instead of using four drops (eight passes) to construct the initial map.

5.1 Results

Out of all the parameters and metrics, the solution with the lowest mean position error was just 0.33 meters, which is scarcely above the 0.25 meter map resolution for that test. In fact, using maps with 1 meter resolution, there were many solutions with mean position errors around 0.5 meters.

Encouragingly, the results from the single drop map were very close to those using the map constructed from four drops (see Figure 10). This is probably largely due to two factors. First, the drop used to construct the map was relatively close to the drop used to test the map (≈12m). Since Zacatón is largely convex, this means that there are not many sonar "shadows" in the map. Secondly, as mentioned above, there are significant registration errors between the drops, which inevitably degraded the quality of the map. This source of error is practically eliminated when a single drop is used to construct the map.

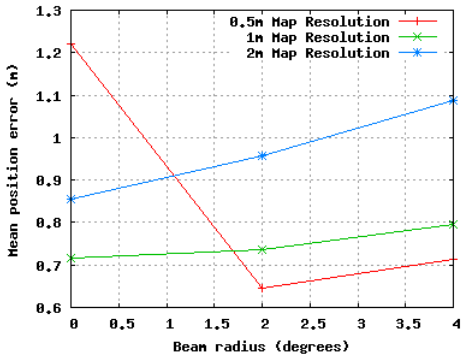

Figure 8: Map resolution versus sonar insertion beam model beam radius.

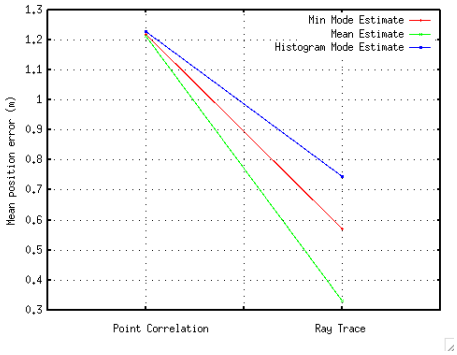

Figure 9: Query method (ray trace and point correlation) versus position estimate method (min mode, mean, histogram mode).

6 Sonar Configuration Tests

Since the real-world datasets were collected with the DropSonde sonar unit, they share the same basic ring sonar array geometry. The primary weakness of this geometry is that when the vehicle is entering unexplored territory, there are no sonars pointing back into the mapped region–which would clearly give crucial information about the vehicle's current pose. In order to compare the performance of several different possible sonar geometries, it was necessary to construct known environments. The primary topic of investigation in this experiment was to compare the performance of the different geometries as the vehicle passed from a mapped region out into an unmapped region. Judging from what we know of the shape of the Zacatón sinkhole, and from other similar formations which have been fully explored, we invisaged three basic scenarios: first, that the vehicle would be exploring down a vertical shaft; second, that the shaft would be sloping at 45 degrees; and finally, that the shaft would be horizontal. While many variations and combinations of these scenarios are possible, we expect that the performance of a particular sonar geometry in these three cases will be sufficiently indicative of its performance in the gen-

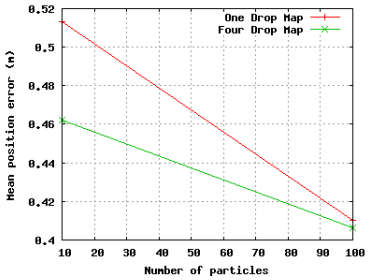

Figure 10: Number of drops used to construct the evidence grid versus the number of particles used the the localization solution.

eral case.

The final DEPTHX vehicle will likely have a more sophisticated sonar array geometry, closely approximating three perpendicular great circles (see Figure 11).

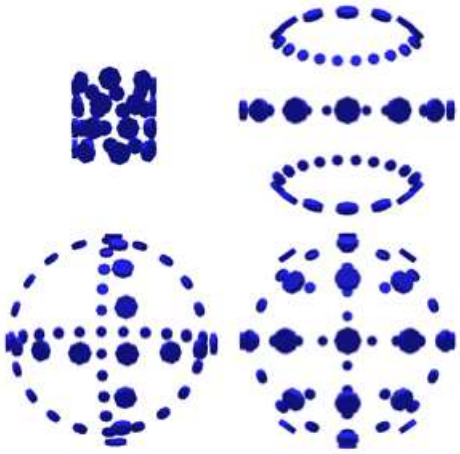

Figure 11: Sonar configurations: Dropsonde(DS), Three Rings (3R), Three Great Circles (3GC), and Three Great Circles Plus (3GC+).

6.1 Results

DropSonde Ring geometry is weak, particularly in the vertical tunnel because it has no sonars to provide "context" (see Figures 13 14 15). Three Great Circles Plus also seems weak, except for the horizontal tunnel test. Three Great Circles seems to be the most versatile, and is the leading candidate for use on the DEPTHX vehicle.

7 Future Work

From our obserservation of the tests on the Zacatón data, we have reason to believe that an Extended Kalman Filter (EKF) might work, at least in the main cylindrical body of Zacatón. One possibility we would like to explore is a hybrid approach where we use the particle filter whenever the vehicle becomes uncertain about its position, and the EKF is used when the particles converge sufficiently .

In the real scenario, it is necessary that the system run in real-time. At the moment it seems safe to say that we should be able to support about 100 particles. However, these are just localization particles, which are much lighter-weight than full SLAM particles, such as are used in a Rao-Blackwellized particle filter. In the final DEPTHX experiment, the vehicle will need to explore at least some distance beyond the mapped region, and during this period it will need to be running a real-time SLAM solution. The difficulty with full SLAM is that the particles must also contain a version of the map which corresponds to the particle's trajectory. How to efficiently represent and manipulate these maps remains is an active research area.

It may be possible to use a hybrid map representation together with an interative exploration strategy, whereby each dive is merged offline into a master map, which is then used on successive dives to localize the robot until is reaches the boundary of exploration, at which point the robot uses a reduced number of particles to allow insertion and querying in real-time.

8 Conclusions

Down to 280 meters, the underwater environment of Zacatón is less complex than we anticipated, however there does appear to be the possibility for something exciting at around 300 meters. This would be

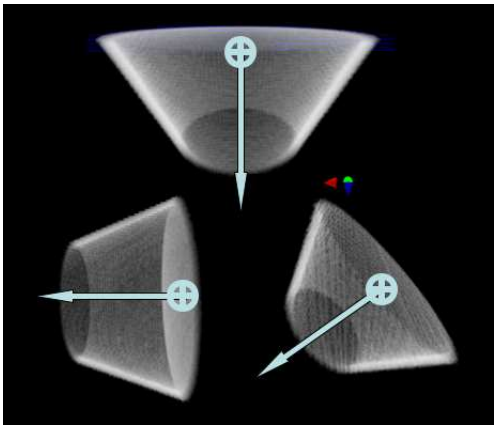

Figure 12: Sonar configuration test tunnels – the upper end of the tunnel is capped, whereas the bottom end of the tunnel is open, so no sonar ranges can be simulated. During the simulation for each tunnel, the vehicle is started at the ⊕, about 1 meter away from the end cap, and proceeds through the 50 meter tunnel in the direction of the arrow.

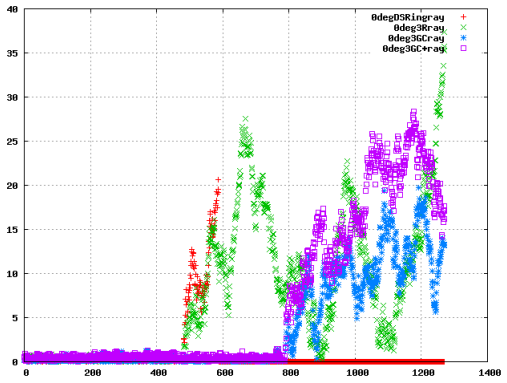

Figure 13: Position estimate error in meters (Y axis) for the different sonar geometries while traveling in 0.1 meter increments (X axis) through the vertical test tunnel.

consistent with other similar geological formations, which seem to follow a pattern of vertical chimneys linked together by horizontal tunnels.

We have demonstrated a functional localization system based on 3D evidence grids and a SIR particle filter. Much work remains to be done to refine the parameters of the system, and to extend it to the full 3D underwater SLAM problem.

References

[1] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online non-linear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, February 2002.

[2] David Bradley, David Silver, and Scott Thayer. A regional point descriptor for global localization in subterranean environments. In *IEEE conference on Robotics Automation and Mechatronics (RAM 2005)*, December 2004.

[3] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. In *Proc. Inst. Elect. Eng. F*, volume 140, pages 107–113, April 1993.

[4] Kurt Konolige and Ken Chou. Markov localization using correlation. In *IJCAI '99: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 1154–1159, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[5] Martin C. Martin and Hans Moravec. Robot evidence grids. Technical Report CMU-RI-TR-96-06, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, March 1996.

[6] P. M. Newman, J. J. Leonard, and R. J. Rikoski. Towards constant-time slam on an autonomous underwater vehicle using synthetic aperture sonar. In D. Paolo and R. Chatila, editors, *Eleventh International Symposium of Robotics Research (ISRR)*, volume 15 of *Springer Tracts in Advanced Robotics (STAR)*, pages 409–420. Springer Verlag, Siena, Italy, 2003.

[7] W. Stone, D. Wettergreen, G. Kantor, M. Stevens, E. Hui, E. Franke, and B. Hogan.

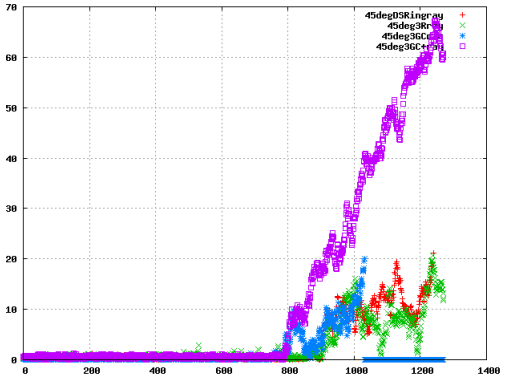

Figure 14: Position estimate error in meters (Y axis) for the different sonar geometries while traveling in 0.1 meter increments (X axis) through the slanted test tunnel.

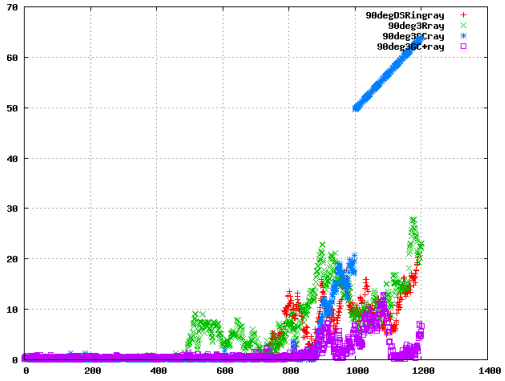

Figure 15: Position estimate error in meters (Y axis) for the different sonar geometries while traveling in 0.1 meter increments (X axis) through the horizontal test tunnel.

Depthx (deep phreatic thermal explorer). In *Proceedings of the 14th International Symposium on Unmanned Untethered Submersible Technology (UUST)*, 2005.

[8] S. Thrun. Robotic mapping: A survey. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, 2002. to appear.

[9] Sebastian Thrun, Dirk Haehnel, David Ferguson, Michael Montemerlo, Rudolph Triebel, Wolfram Burgard, Christopher Baker, Zachary Omohundro, Scott Thayer, and William Red L. Whittaker. A system for volumetric robotic mapping of abandoned mines. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation (ICRA '03)*, May 2003.

[10] Robert J. Urick. *Principles of Underwater Sound*. McGraw-Hill, New York, NY, 3rd edition, 1983.

[11] S. B. Williams and I. Mahon. Simultaneous localisation and mapping on the great barrier reef. In *IEEE International Conference on Robotics and Automation*, New Orleans, USA, April 26-May 1 2004.

[12] S. B. Williams, P. Newman, M. W. M. G. Dissanayake, and H.F. Durrant-Whyte. Autonomous underwater simultaneous localisation and map building. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 22–28, San Francisco CA, USA, April 2000.