

# High Resolution Maps from Wide Angle Sonar

Hans P. Moravec

Alberto Elfes

The Robotics Institute

Carnegie-Mellon University

## Abstract

*We describe the use of multiple wide-angle sonar range measurements to map the surroundings of an autonomous mobile robot. A sonar range reading provides information concerning empty and occupied volumes in a cone (subtending 30 degrees in our case) in front of the sensor. The reading is modelled as probability profiles projected onto a rasterized map, where somewhere occupied and everywhere empty areas are represented. Range measurements from multiple points of view (taken from multiple sensors on the robot, and from the same sensors after robot moves) are systematically integrated in the map. Overlapping empty volumes re-inforce each other, and serve to condense the range of occupied volumes. The map definition improves as more readings are added. The final map shows regions probably occupied, probably unoccupied, and unknown areas. The method deals effectively with clutter, and can be used for motion planning and for extended landmark recognition. This system has been tested on the Neptune mobile robot at CMU.*

## 1. Introduction

This paper describes a sonar-based mapping system developed for mobile robot navigation. It was tested in cluttered environments using the Neptune mobile robot [8], developed at the Mobile Robot Laboratory of the Robotics Institute, CMU. The Neptune system has been used successfully in several areas of research, including stereo vision navigation [5, 10] and path planning [11]. Other research in the laboratory includes the investigation of adequate high-level robot control structures, the use of distributed and parallel processing methods to improve the real-time response of the system, navigation in outdoor environments and the design and construction of more advanced robots with higher mobility.

Primarily because of computational expense, practical real-world stereo vision navigation systems [7, 10] build very sparse depth maps of their surroundings. Even with this economy our fastest system, described in [5], takes 30 - 60 seconds per one meter step on a 1 mips machine.

---

This work has been supported in part by Denning Mobile Robotics, Inc., by the Western Pennsylvania Advanced Technology Center and by the Office of Naval Research under contract number N00014-81-K-0503. The second author is supported in part by the Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq, Brazil, under Grant 200.986-80; in part by the Instituto Tecnológico de Aeronáutica - ITA, Brazil; and in part by The Robotics Institute, Carnegie-Mellon University.

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the funding agencies.

Direct sonar range measurements promised to provide basic navigation and denser maps with considerably less computation. The readily available Polaroid ultrasonic range transducer [9] was selected and a ring of 24 of these sensors was mounted on Neptune.

We find sonar sensors interesting also because we would like to investigate how qualitatively different sensors, such as a sonar array and a pair of cameras, could cooperate in building up a more complex and rich description of the robot's environment.

### 1.1. Goals

We expected sonar measurements to provide maps of the robot's environment with regions classified as empty, occupied or unknown, and matches of new maps with old ones for landmark classification and to obtain or correct global position and orientation information.

### 1.2. Approach

Our method starts with a number of range measurements obtained from sonar units whose position with respect to one another is known. Each measurement provides information about empty and possibly occupied volumes in the space subtended by the beam (a thirty degree cone for the present sensors). This occupancy information is projected onto a rasterized two-dimensional horizontal map. Sets of readings taken both from different sensors and from different positions of the robot are progressively incorporated into the sonar map. As more readings are added the area deduced to be empty expands, and the expanding empty area encroaches on and sharpens the possibly occupied region. The map becomes gradually more detailed.

For navigation and recognition we developed a way of convolving two sonar maps, giving the displacement and rotation that best brings one map into registration with the other, along with a measure of the goodness of the match.

The sonar maps happen to be very useful for motion planning. They are denser than those made by our stereo vision programs, and computationally about an order of magnitude faster to produce. We presently use them with the Path Relaxation method [11] to plan local paths for our robot.

### 1.3. Related Work

Sonar is a developed technology but few applications until recently involved detailed map building. Traditional marine applications, camera autofocus systems, and some simple robot navigation schemes [2, 6] rely on sparse proximity measurements to accomplish their narrow goals.

The most advanced sonar systems used in marine intelligence operations locate sound sources passively [1]. Ultrasound systems used in medicine are typically active and build maps for human perusal, but depend on accurate physical models of the environments that the sound traverses [4], and work with very small beam widths, about  $1^\circ - 3^\circ$ . Narrow beam widths, formed by phased array techniques, are also used in advanced side looking mapping sonar system for submersibles. An independent CMU sonar mapping effort [3] also used a narrow beam, formed by a parabolic reflector, in its attempts to build a line-based description.

In contrast the sonar sensors that we choose to employ have a wide beam, with an effective angle of about  $30^\circ$ .

## 2. The Sonar System

### 2.1. The Sensor

The sonar devices being used are Polaroid laboratory grade ultrasonic transducers [9]. These sonar elements have a useful measuring range of 0.9 to 35.0 ft. The main lobe of the sensitivity function corresponds to a beam angle of  $30^\circ$  at  $-38$  dB. Experimental results showed that the range accuracy of the sensors is on the order of 1 %. We are using the control circuitry provided with the unit, which is optimized for giving the range of the nearest sound reflector in its field of view, and works well for this purpose.

### 2.2. The Array

The sonar array, built at Denning Mobile Robotics, and mounted on the *Neptune* mobile robot is composed of:

- A ring of 24 Polaroid sonar elements, spaced  $15^\circ$  apart, and mounted at an height of 31 inches above the ground (see Fig. 1).
- A Z80 controlling microprocessor which selects and fires the sensors, times the returns and provides a range value.
- A serial line over which range information is sent to a VAX mainframe that presently interprets the sonar data and performs the higher level mapping and navigation functions.

## 3. Sonar Mapping

### 3.1. Obtaining Reliable Range Data from the Sonar Sensor

We begin our map building by preprocessing the incoming readings to remove chronic errors. The following steps are used:

- **Thresholding:** Range readings above a certain maximum  $R_u$  are discarded. We observe that sonar readings caused by specular reflections are often near the maximum range of the device ( $R_{max}$ ). With  $R_u$  slightly below  $R_{max}$ , many of these readings are discarded. The system becomes slightly myopic, but the overall quality of the map improves. Very large open spaces are detected by analyzing the set of distance values obtained from the sonar, and in this case the filtering is not done. A similar heuristic is applied for small readings: values below the minimum sensor range  $R_{min}$  are usually glitches and are discarded.

- **Averaging:** Several independent readings from the same sensor at the same position are averaged. The sonar readings are subject to error not only from reflections but also from other causes such as fluctuations in the effective sensitivity of the transducer. As a result readings show a certain dispersion. Averaging narrows the spread.
- **Clustering:** A set of readings from one sensor at a given position sometimes shows a clustering of the data around two different mean values. This happens when different readings are being originated by objects at staggered distances. We apply a simple clustering analysis to the data, and extract a mean value for each cluster for use in subsequent processing.

### 3.2. Representing the Sonar Beam

Because of the wide beam angle the filtered data from the above methods provides only indirect information about the location of the detected objects. We combine the constraints from individual readings to reduce the uncertainty. Our inferences are represented as probabilities in a discrete grid.

A range reading is interpreted as providing information about space volumes that are probably EMPTY and somewhere OCCUPIED. We model the sonar beam by probability distribution functions. Informally, these functions model our confidence that the various points inside the cone of the beam are empty and our uncertainty about the location of the point, somewhere on the range surface of the cone, that caused the echo. The functions are based on the range reading and on the spatial sensitivity pattern of the sonar.

Consider a position  $P = (x, y, z)$  belonging to the volume swept by the sonar beam. Let:

- $R$  be the range measurement returned by the sonar sensor,
- $\epsilon$  be the mean sonar deviation error,
- $\omega$  be the beam aperture,
- $S = (x_s, y_s, z_s)$  be the position of the sonar sensor,
- $\delta$  be the distance from  $P$  to  $S$ ,
- $\theta$  be the angle between the main axis of the beam and  $SP$ .

We now identify two regions in the sonar beam:

- **Empty Region:** Points inside the sonar beam ( $\delta < R - \epsilon$  and  $\theta \leq \omega/2$ ), that have a probability  $p_E = f_E(\delta, \theta)$  of being empty.
- **Somewhere Occupied Region:** Points on the sonar beam front ( $\delta \in [R - \epsilon, R + \epsilon]$  and  $\theta \leq \omega/2$ ), that have a probability  $p_O = f_O(\delta, \theta)$  of being occupied.

Fig. 2 shows the probability profiles for a sonar beam that returned a range reading  $R$ . The horizontal crosssection of the beam is associated with two probability distributions corresponding to the empty and the occupied probabilities.

The empty probability density function for a point  $P$  inside the sonar beam is given by:

$$p_E(x, y, z) = p[\text{position } (x, y, z) \text{ is empty}] = E_r(\delta) \cdot E_a(\theta) \quad (1)$$

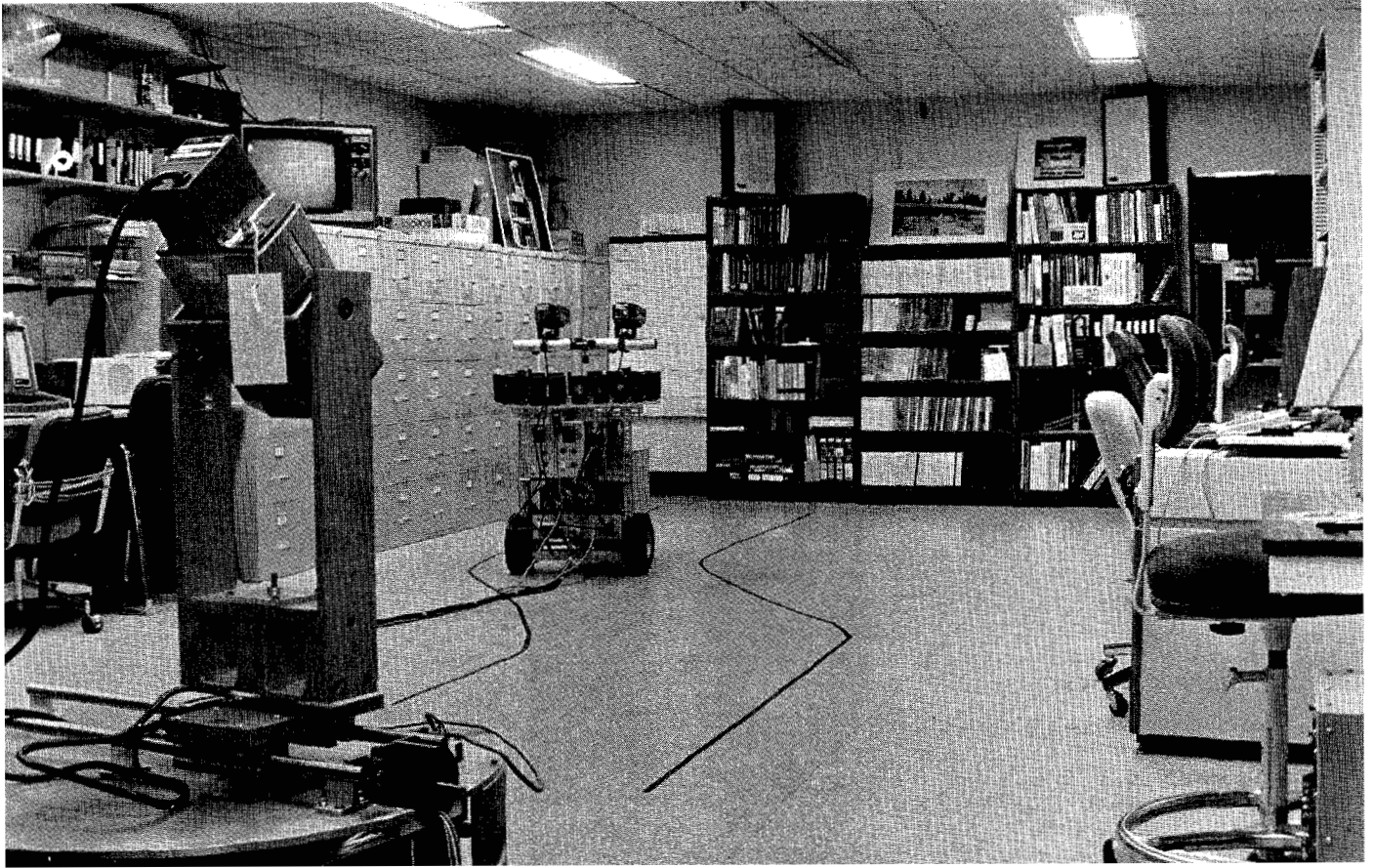


Figure 1: The *Neptune* mobile robot, with a pair of cameras and the sonar ring, in our laboratory. Sonar maps of this lab can be seen in Figures 3 through 8.

where:

$$E_r(\delta) = 1 - ((\delta - R_{min}) / (R - \epsilon - R_{min}))^2 \text{ for } \delta \in [R_{min}, R - \epsilon] \quad (2)$$

$$E_r(\delta) = 0 \text{ otherwise.}$$

And:

$$E_a(\theta) = 1 - (2\theta/\omega)^2 \text{ for } \theta \in [-\omega/2, \omega/2] \quad (3)$$

The occupied probability density function for a point **P** on the beam front is given by:

$$p_O(x,y,z) = p[\text{position } (x,y,z) \text{ is occupied}] = O_r(\delta) \cdot O_a(\theta) \quad (4)$$

where:

$$O_r(\delta) = 1 - ((\delta - R) / \epsilon)^2 \text{ for } \delta \in [R - \epsilon, R + \epsilon] \quad (5)$$

$$O_r(\delta) = 0 \text{ otherwise.}$$

And:

$$O_a(\theta) = 1 - (2\theta/\omega)^2 \text{ for } \theta \in [-\omega/2, \omega/2] \quad (6)$$

These probability density functions are now projected on a horizontal plane to generate map information. We use the profiles that correspond to a horizontal section of the sonar beam ( $z = z_s$ ).

### 3.3. Representing Maps

Sonar Maps are two-dimensional arrays of cells corresponding to a horizontal grid imposed on the area to be mapped. The grid has  $M \times N$  cells, each of size  $\Delta \times \Delta$ . The final map has cell values in the range  $(-1,1)$ , where values less than 0 represent probably empty regions, exactly zero represents unknown occupancy, and greater than 0 implies a probably occupied space. This map is computed in a final step from two separate arrays analogous to the empty and occupied probability distributions introduced above.

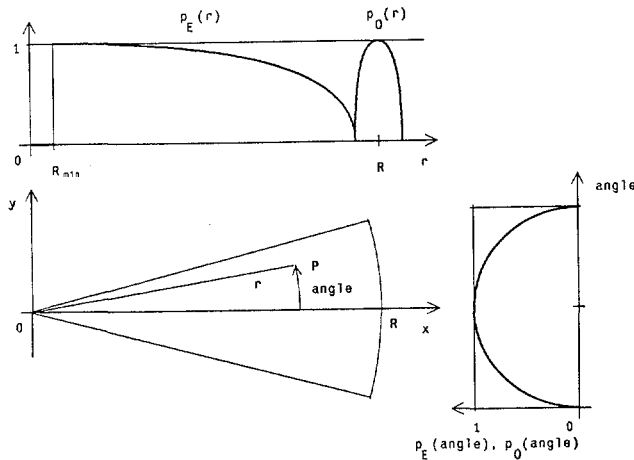
A cell is considered UNKNOWN if no information concerning it is available. Cells can be EMPTY with a degree of certainty or confidence  $Emp(X,Y)$  and OCCUPIED with a degree of certainty  $Occ(X,Y)$  both values ranging from 0 to 1.

The *a priori* empty and occupied certainty values for a given grid cell  $(X,Y)$  and reading are determined by taking the minimum of the reading's  $p_r$  and maximum of  $p_O$ , respectively, over the cell through a horizontal slice through the beam center.

### 3.4. Composing Information from Several Readings

The map is built by projecting the beam probabilities onto the discrete cells of the sonar map and there combining it with information from other beams. The position and the orientation of the sonar sensor are used to register the beam with the map.

Different readings asserting that a cell is EMPTY will enhance each other, as will readings implying that the cell may be OCCUPIED while evidence that the cell is EMPTY will weaken the certainty of it being OCCUPIED and vice-versa.



**Figure 2:** The Probability Profiles corresponding to the probably Empty and somewhere Occupied regions in the sonar beam. The profiles represent a horizontal cross section of the beam.

The operations performed on the empty and occupied probabilities are not symmetrical. The probability distribution for *empty* areas represents a solid volume whose totality is probably empty but the *occupied* probability distribution for a single reading represents a lack of knowledge we have about the location of a single reflecting point somewhere in the range of the distribution. Empty regions are simply added using a probabilistic addition formula. The *occupied* probabilities for a single reading, on the other hand, are reduced in the areas that the other data suggests is empty, then normalized to make their sum unity. Only after this narrowing process are the *occupied* probabilities from each reading combined using the addition formula.

One range measurement contains only a small amount of information. By combining the evidence from many readings as the robot moves in its environment, the area known to be empty is expanded. The number of regions somewhere containing an occupied cell increases, while the range of uncertainty in each such region decreases. The overall effect as more readings are added is a gradually increasing coverage along with an increasing precision in the object locations. Typically after a few hundred readings (and less than a second of computer time) our process is able to "condense out" a comprehensive map covering a thousand square feet with better than one foot position accuracy of the objects detected. Note that such a result does not violate information theoretic or degree of freedom constraints, since the detected boundaries of objects are linear, not quadratic in the dimensions of the map. A thousand square foot map may contain only a hundred linear feet of boundary.

Formally the evidence combination process proceeds along the following steps:

1. RESET: The whole Map is set to UNKNOWN by making  $Emp(X,Y) = 0$  and  $Occ(X,Y) = 0$ .
2. SUPERPOSITION OF EMPTY AREAS: For every sonar reading  $k$  modify the emptiness information over its projection by:

$$\begin{aligned} \text{ENHANCE: } Emp(X,Y) &:= \\ Emp(X,Y) + Emp_k(X,Y) - Emp(X,Y) \times Emp_k(X,Y) \end{aligned}$$

3. SUPERPOSITION OF OCCUPIED AREAS: For each reading  $k$ , shift the occupied probabilities around in response to the combined emptiness map using:

$$\text{CANCEL: } Occ_k(X,Y) := Occ_k(X,Y) \cdot (1 - Emp(X,Y))$$

$$\text{NORMALIZE: } Occ_k(X,Y) := Occ_k(X,Y) / \sum Occ_k(X,Y)$$

$$\begin{aligned} \text{ENHANCE: } Occ(X,Y) &:= \\ Occ(X,Y) + Occ_k(X,Y) - Occ(X,Y) \times Occ_k(X,Y) \end{aligned}$$

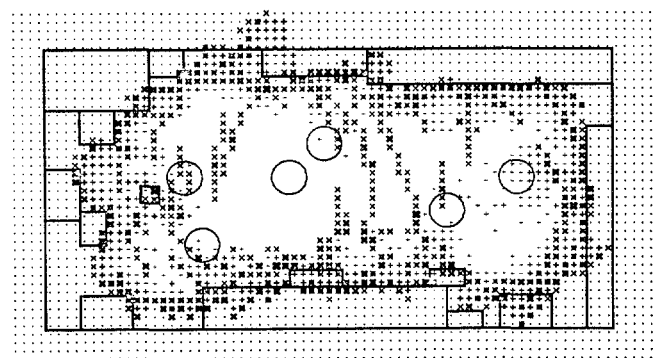
4. THRESHOLDING: The final occupation value attributed to a cell is given by a thresholding method:

$$\begin{aligned} \text{THRESHOLD: } Map(X,Y) &:= \\ \begin{cases} Occ(X,Y) & \text{if } Occ(X,Y) \geq Emp(X,Y) \\ -Emp(X,Y) & \text{if } Occ(X,Y) < Emp(X,Y) \end{cases} \end{aligned}$$

### 3.5. Maps

A typical map obtained through this method is shown in Fig. 3, and the corresponding certainty factor distributions are shown in Figs. 4 and 5. These are the maps obtained before the thresholding step.

The final maps obtained after thresholding are shown in Figs. 6, 7 and 8.



**Figure 3:** A Two-Dimensional Sonar Map. Each symbol represents a square area six inches on a side in the room pictured in Figure 1. The right edge of this diagram corresponds to the far wall in the picture. Empty areas with a high certainty factor are represented by white space; lower certainty factors by "+" symbols of increasing thickness. Occupied areas are represented by "x" symbols, and Unknown areas by ".". The robot positions where scans were taken are shown by circles and the outline of the room and of the major objects by solid lines.

### 4. Matching

Sonar navigation would benefit from a procedure that can match two maps and report the displacement and rotation that best takes one into the other.

Our most successful programs begin with the thresholded maps described above, with cell values that are negative if the cell is empty, positive if occupied and zero if unknown.

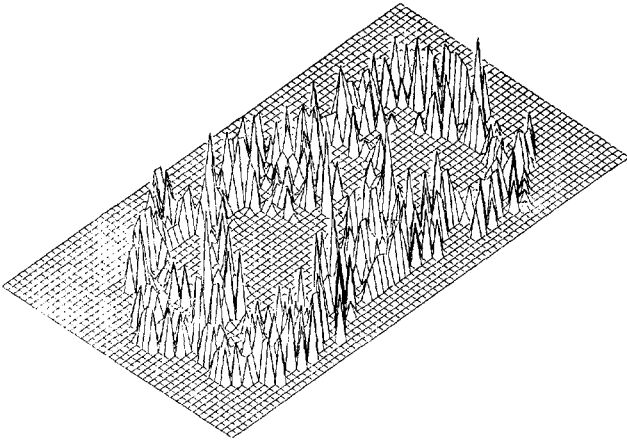


Figure 4: The Occupied Areas in the Sonar Map. This 3-D view shows the Certainty Factors  $Occ(X,Y)$ .

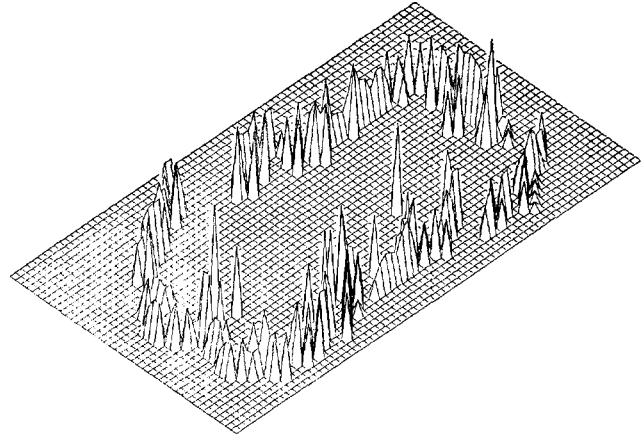


Figure 7: The Occupied Areas in the Sonar Map After Thresholding.

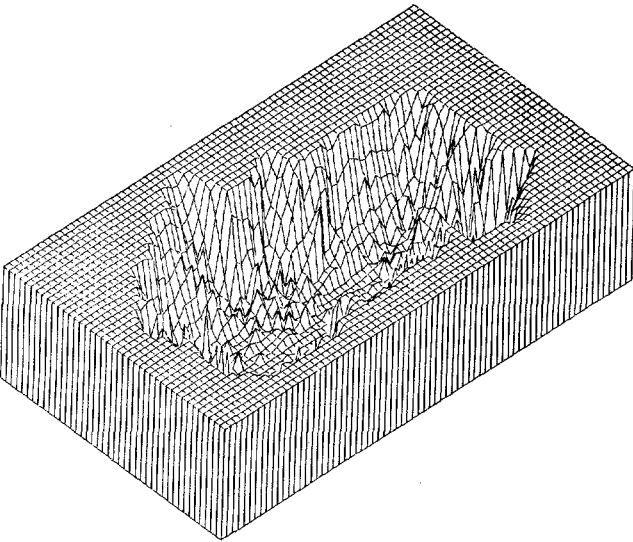


Figure 5: The Empty Areas in the Sonar Map. This 3-D view shows the Certainty Factors  $Emp(X,Y)$ .

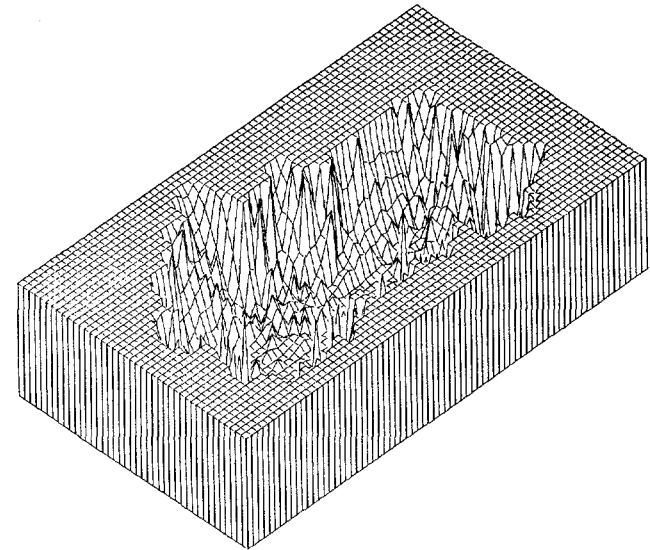


Figure 8: The Empty Areas in the Sonar Map After Thresholding.

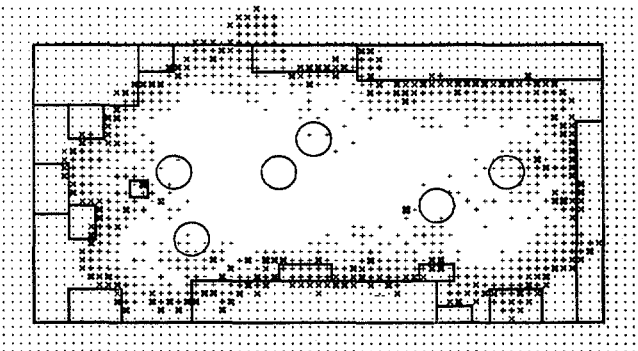


Figure 6: The Two-Dimensional Sonar Map After Thresholding.

A measure of the goodness of the match between two maps at a trial displacement and rotation is found by computing the sum of products of corresponding cells in the two maps. An occupied cell falling on an occupied cell contributes a positive increment to the sum, as does an empty cell falling on an empty cell (the product of two negatives). An empty cell falling on an occupied one reduces the sum, and any comparison involving an unknown value causes neither an increase nor a decrease. This naive approach is very slow. Applied to maps with a linear dimension of  $n$ , each trial position requires  $O(n^2)$  multiplications. Each search dimension (two axes of displacement and one of rotation) requires  $O(n)$  trial positions. The total cost of the approach thus grows as  $O(n^5)$ . With a typical  $n$  of 50 this approach can burn up a good fraction of an hour of Vax time.

Considerable savings come from the observation that most of the information in the maps is in the occupied cells alone. Typically only  $O(n)$  cells in the map, corresponding to wall and object boundaries, are labelled occupied. A revised matching procedure compares maps A and B through trial transformation  $T$  (represented by a  $2 \times 2$  rotation matrix and a 2 element displacement vector) by enumerating the occupied cells of A, transforming the co-ordinates of each such cell through  $T$  to find a

corresponding cell in B. The  $[A, B]$  pairs obtained this way are multiplied and summed, as in the original procedure. The occupied cells in B are enumerated and multiplied with corresponding cells in A, found by transforming the B co-ordinates through  $T^{-1}$  (the inverse function of  $T$ ), and these products are also added to the sum. The result is normalized by dividing by the total number of terms. This procedure is implemented efficiently by preprocessing each sonar map to give both a raster representation and a linear list of the co-ordinates of occupied cells. The cost grows as  $O(n^4)$ , and the typical Vax running time is down to a few minutes.

A further speedup is achieved by generating a hierarchy of reduced resolution versions of each map. A coarser map is produced from a finer one by converting two by two subarrays of cells in the original into single cells of the reduction. Our existing programs assign the maximum value found in the subarray as the value of the result cell, thus preserving occupied cells. If the original array has dimension  $n$ , the first reduction is of size  $n/2$ , the second of  $n/4$  and so on. A list of occupied cell locations is produced for each reduction level so that the matching method of the previous paragraph can be applied. The maximum number of reduction levels is  $\log_2 n$ . A match found at one level can be refined at the next finer level by trying only about three values of each of the two translational and one rotational parameters, in the vicinity of the values found at the coarser level, for a total of 27 trials. With a moderate a-priori constraint on the transformation this amount of search is adequate even at the first (coarsest) level. Since the cost of a trial evaluation is proportional to the dimension of the map, the coarse matches are inexpensive in any case. Applied to its fullest, this method brings the matching cost down to slightly larger than  $O(n)$ , and typical Vax times to under a second.

We found one further preprocessing step is required to make the matching process work in practice. Raw maps at standard resolutions (6 inch cells) produced from moderate numbers (about 100) of sonar measurements have narrow bands of cells labelled *occupied*. In separately generated maps of the same area the relative positions of these narrow bands shifts by as much as several pixels, making good registration of the occupied areas of the two maps impossible. This can be explained by saying that the high spatial frequency component of the position of the bands is noise, only the lower frequencies carry information. The problem is fixed by filtering (blurring) the occupied cells to remove the high frequency noise. Experiment suggests that a map made from 100 readings should be blurred with a spread of about two feet, while for map made from 200 readings a one foot smear is adequate. Blurring increases the number of cells labelled *occupied*. So as not to increase the computational cost from this effect, only the final raster version of the map is blurred. The occupied cell list used in the matching process is still made from the unfiltered raster.

With the process outlined here, maps with about 3000 six inch cells made from 200 well spaced readings can be matched with an accuracy of about six inches displacement and three degrees rotation in one second of Vax time.

## 5. Results

We incorporated the sonar map builder into a system that successfully navigates the Neptune robot through cluttered obstacle courses. The existing program incrementally builds a single sonar map by combining the readings from successive vehicle stops made about one meter apart. Navigation is by dead reckoning - we do not yet use the sonar map matching code. The next move is planned in the most recent version of the map by a path-planning method based on path relaxation [11]. Since this method can cope with a probabilistic

representation of occupied and empty areas and does path-planning in a grid, it fits naturally into our present framework. The system has successfully driven Neptune the length of our cluttered 30 by 15 foot laboratory using less than one minute of computer time.

## 6. Conclusions

We have described a program that builds moderately high resolution spatial maps of a mobile robot's surroundings by combining several hundred range readings from unadorned Polaroid ultrasonic units. The main innovation is an efficient mathematical method that reduces the position uncertainty of objects detected by wide angle sonar beams by combining interlocking constraints in a raster occupation probability map. We have also developed a fast algorithm for relating two maps of the same area to derive relative displacement, angle and goodness of match.

We have used this mapping method in a system that navigates a mobile robot to a desired destination through obstacles and clutter, and are preparing a more elaborate navigation system that depends on matching of the sonar maps to recognize key locations and on higher-level representations to navigate over long distances.

## 7. Acknowledgments

The authors would like to thank Gregg W. Podnar for his help in using the Neptune robot and performing some of the experiments, Michael Fuhrman for precious advice on the use of a 3D drawing package, and Larry Matthies for providing the means to print the corresponding drawings.

## References

1. Bagge-roer, A.B. Sonar Signal Processing. In *Applications of Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, N.J., 1978.
2. Chattergy, A. Some Heuristics for the Navigation of a Robot. Robotics Research Laboratory, Department of Electrical Engineering, University of Hawaii, 1984.
3. Crowley, J.L. Position Estimation for an Intelligent Mobile Robot. 1983 Annual Research Review, The Robotics Institute, Carnegie-Mellon University, Pittsburgh, PA, 1984.
4. Hussey, M., *Diagnostic Ultrasound: An Introduction to the Interactions between Ultrasound and Biological Tissues*, Blackie & Son Limited, London, 1975.
5. Matthies, L.H. and Thorpe, C.E. Experience With Visual Robot Navigation. Proceedings of IEEE Oceans 84, IEEE, Washington, D.C., August, 1984.
6. Miller, D. Two Dimensional Mobile Robot Positioning Using Onboard Sonar. Pecora IX Remote Sensing Symposium Proceedings, IEEE, Sioux Falls, SD, October, 1984.
7. Moravec, H.P. *Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover*. Ph.D. Th., Stanford University, May 1980. Also available as Stanford AIM-340, CS-80-813 and CMU-RI-TR-3
8. Podnar, G.W., Blackwell, M.K. and Dowling, K. A Functional Vehicle for Autonomous Mobile Robot Research. CMU Robotics Institute, April, 1984.
9. Polaroid Corporation. *Ultrasonic Range Finders*. Polaroid Corporation, 1982.
10. Thorpe, C.E. The CMU Rover and the FIDO Vision and Navigation System. Symposium on Autonomous Underwater Robots, University of New Hampshire, Marine Systems Engineering Lab, May, 1983.
11. Thorpe, C.E. Path Relaxation: Path Planning for a Mobile Robot. Technical Report CMU-RI-TR-84-5, CMU Robotics Institute, April, 1984. Also in Proceedings of IEEE Oceans 84, Washington, D.C., August, 1984 and Proceedings of AAAI-84, Austin, Texas, August, 1984