Sonar-Based Real-World Mapping and Navigation

ALBERTO ELFES, MEMBER, IEEE

Abstract—A sonar-based mapping and navigation system developed for an autonomous mobile robot operating in unknown and unstructured environments is described. The system uses sonar range data to build a multileveled description of the robot's surroundings. Sonar readings are interpreted using probability profiles to determine empty and occupied areas. Range measurements from multiple points of view are integrated into a sensor-level sonar map, using a robust method that combines the sensor information in such a way as to cope with uncertainties and errors in the data. The resulting two-dimensional maps are used for path planning and navigation. From these sonar maps, multiple representations are developed for various kinds of problem-solving activities. Several dimensions of representation are defined: the abstraction axis, the geographical axis, and the resolution axis. The sonar mapping procedures have been implemented as part of an autonomous mobile robot navigation system called Dolphin. The major modules of this system are described and related to the various mapping representations used. Results from actual runs are presented, and further research is mentioned. The system is also situated within the wider context of developing an advanced software architecture for autonomous mobile robots.

I. INTRODUCTION

TO WIDEN the range of application of robotic devices, both in industrial and research applications, it is necessary to develop systems with high levels of autonomy and able to operate in unstructured environments with little *a priori* information. To achieve this degree of independence, the robot system must have an understanding of its surroundings, by acquiring and manipulating a rich model of its environment of operation. For that, it needs a variety of sensors to be able to interact with the real world and mechanisms to extract meaningful information from the data being provided. Systems will little or no sensing capability are usually limited to fixed sequence operations in highly structured working areas and cannot provide any substantial degree of autonomy or adaptability.

A central need, both for manipulators and for mobile robots, is the ability to acquire and handle information about the existence and localization of objects and empty spaces in the environment of operation of the device. This is crucial for fundamental operations that envolve spatial and geometric

The author is with the Mobile Robot Laboratory, The Robotics Institute, Carnegie-Mellon University, Pittsburgh, PA 15213, USA.

reasoning. Typically, due to limitations intrinsic to any kind of sensor, it is important to compose information coming from multiple readings, and build a coherent world-model that reflects the information acquired and the hypotheses proposed so far. This world model can then serve as a basis for essential operations such as path planning, obstacle avoidance, landmark identification, position and motion estimation, etc.

Building up such a description involves the complex task of extracting range information from the real world. Several range measurement systems have been proposed in the literature [21]. Of particular interest for mobile robot research [22], [24] are stereo vision systems [29], [31], [25], [39] and active rangefinding devices [14], [15], [19], since these do not require artificial environments or contrived lighting.

In this paper, we explore the use of one specific active rangefinding device, an ultrasonic range transducer, to build a dense two-dimensional map of the robot's surroundings. Each sonar distance reading provides information concerning *empty* and *occupied* volumes in a cone in front of the sensor. The reading is interpreted using probability profiles that are projected onto a rasterized map, where *unknown, occupied,* and *empty* areas are explicitly represented.

Range measurements from multiple points of view (taken from multiple sensors on the robot and from different positions occupied by the robot) are integrated into the sonar map, using a robust method that combines the sensor information in such a way as to cope with uncertainties and errors in the data. Overlapping empty areas reinforce each other, the same happening with occupied areas. Additionally, the empty spaces serve to narrow down the profiles of occupied spaces. The result is that the sonar map grows in coverage and its definition improves as more readings are added. The resulting sonar map shows regions probably occupied, probably empty, and unknown areas. The method works effectively in cluttered environments, and the dense maps resulting can be used for motion planning, position estimation, landmark recognition, and navigation.

These probabilistic sensor-level sonar maps serve as the basis for a multilevel description of the robot's operating environment. These multiple descriptions are developed for various kinds of problem-solving activities. Several dimensions of representation are defined: the *abstraction* axis, the *geographical* axis, and the *resolution* axis.

The sonar mapping method has been implemented as part of an autonomous mobile robot navigation system called Dolphin. This system is intended to provide sonar-based mapping and navigation for an autonomous mobile robot operating in unknown and unstructured environments. The system is completely autonomous in the sense that it has no *a priori*

VIII. Supervisor

- Global Supervision of System Behaviour
- User Interface

VII. Global Planning

- Task-Level Planning to provide sequences of sensory, actuator and processing (software) actions
- Simulation
- Error-Recovery and Replanning in case of failure or unexpected events

VI. Control

- Scheduling of Activities
- Integration of Plan-Driven with Data-Driven Activities

V. Navigation

- Navigation Modules provide services such as Path-Planning and Obstacle Avoidance

IV. Real-World Modelling

- Integration of local pieces of correlated information into a Global Real-World Model that describes the robot's environment of operation
- Matching acquired information against stored maps
- Object Identification
- Landmark Recognition

III. Sensor Integration

- Information provided by different Sensor Modules is correlated and abstracted
- Common representations and compatible frames of reference are used

II. Sensor Interpretation

- Acquisition of Sensor Data (Vision, Sonar, Laser Rangefinder, etc.)
- Interpretation of Sensor Data

I. Robot Control

- Set of Primitives for Robot Operation
- Actuator Control (e.g., Locomotion)
- Sensor Control
- Internal Sensor Monitoring

Fig. 1.   Conceptual processing levels in mobile robot software architecture.

model or knowledge of its surroundings and also carries no user-provided map. It incrementally builds sonar maps that are used to plan safe paths and navigate the vehicle towards a given goal and may be coupled to other systems, such as vision, that would locate landmarks that would serve as long-range destinations. The system has been tested both indoors and outdoors using the Neptune and Terregator mobile robots at CMU.

In the sequence of this paper, we will briefly identify some of the conceptual processing levels needed for mobile robot software and relate the present system to this framework, describe the sonar mapping method, discuss the multiple representations developed for mapping information, present the overall system architecture, and show some results from actual runs. We finish with an outline of further research.

II. Conceptual Processing Levels for an Autonomous Mobile Robot

The sonar mapping and navigation system described in this paper is part of a wider investigation into issues related to the development of a software architecture for an autonomous mobile robot. In this section, we briefly outline a conceptual framework within which the sonar system is situated by characterizing the *conceptual processing levels* into which the various problem-solving activities of a mobile robot software architecture can be classified. The levels include the robot control, sensor interpretation, sensor integration, real-world modeling, navigation, control, global planning, and supervisor levels (Fig. 1), and are briefly described below.

Robot Control: This level takes care of the physical control

of the different sensors and actuators available to the robot. It provides a set of primitives for locomotion, actuator and sensor control, data acquisition, etc., that serve as the robot interface, freeing the higher levels of the system from low-level details. It includes activities such as vehicle-based motion estimation and monitoring of internal sensors. *Internal sensors* provide information on the status of the different physical subsystems of the robot, while *external sensors* are used to acquire data about the outside world.

Sensor Interpretation: On this level the acquisition of sensor data and its interpretation by sensor modules is done. Each sensor module is specialized in one type of sensor or even in extracting a specific kind of information from the sensor data. The modules provide information to the higher levels using a common representation and compatible frames of reference.

Sensor Integration: Due to the intrinsic limitations of any sensory device, it is essential to integrate information coming from qualitatively different sensors, such as stereo vision systems, sonar devices, laser range sensors, etc. Specific assertions provided by the sensor modules are correlated to each other on this level. For example, the geometric boundaries extracted from an obstacle detected by sonar can be used to provide connectivity information concerning a set of scattered three-dimensional (3D) points generated by the stereo vision subsystem. On this level, information is aggregated and assertions about specific portions of the environment can be made.

Real-World Modeling: To achieve any substantial degree of autonomy, a robot system must have an understanding of its surroundings, by acquiring and manipulating a rich model of its environment of operation. This model is based on assertions composed from the various sensors, and reflects the data obtained and the hypotheses proposed so far. On this level, local pieces of information are used in the incremental construction of a coherent global real-world model; this model can then be used for several other activities, such as landmark recognition, matching of newly acquired information against previously stored maps, and generation of expectations and goals.

Navigation: For autonomous locomotion, a variety of problem-solving activities are necessary, such as short-term and long-term path planning, obstacle avoidance, detection of emergencies, etc. These different activities are performed by modules that provide specific services.

Control: This level is responsible for the scheduling of the different activities and for combining plan-driven and data-driven activities in an integrated manner to achieve coherent behavior. In other words, this level tries to execute the task-level plan that was handed to it, while adapting to changing real-world conditions as detected by the sensors.

Global Planning: To achieve a global goal proposed to the robot, this level provides task-level planning for autonomous generation of sequences of actuator, sensor, and processing actions. Other activities needed include simulation, error detection, diagnosis and recovery, and replanning in the case of unexpected situations or failures.

Supervisor: Finally, on this level a supervisory module controls the various activities and provides an interface to a human overseer.

By identifying these areas of activity, we are not implying that communication among processing modules is only possible between adjacent levels. On the contrary, experience with real systems shows that usually there are very complex interconnections and interdependencies between the various subsystems, with multiple flows of control and data. Additionally, a specific module (such as stereo vision or sonar mapping) may be a very complex system in itself, with sophisticated control, planning and problem-solving activities.

Clearly, none of the presently existing mobile robot systems cover all of the levels described. This conceptual structure provides, however, a context within which some of our research is situated [10], [31] and has influenced in particular the design of the Dolphin sonar-based mapping and navigation system, as mentioned in Section V.

III. SONAR-BASED MAPPING

This section describes a sonar-based mapping method developed for mobile robot navigation [32], [12]. We discuss the relative merits of sonar sensors, describe the interpretation of sonar data and the map-building process, and present some experimental results.

A. Range Sensors

Several methods for obtaining range data have been reported in the literature. The survey provided by Jarvis [21] discusses, among others, contrived lighting techniques (including striped lighting and grid coding), depth from occlusion, texture gradient and focusing, some "shape from" methods, range from stereo or motion, and triangulation-based and time-of-flight rangefinders. Of these, contrived lighting techniques are not generally useful to systems operating in unstructured environments, while triangulation-based rangefinders suffer from "gaps" in the data due to occlusions.

Of particular interest for natural unstructured environments are stereo vision systems and active rangefinding devices. In the time-of-flight category, the main two representatives are ultrasonic and laser rangefinders. Sonar systems have lower resolution; on the other hand, they are orders of magnitude less expensive than laser-based sensors. Phase-shift-based laser rangefinders are subject to a 2π uncertainty. In the case of the ERIM sensor [19], this limits the useful range of the sensor to 64 ft, as compared to 35 ft for the Polaroid sonar sensor. Both sensors suffer from absorption and specular reflection problems, while measurement precision is obviously much higher with laser rangefinders.

B. Stereo Vision and Sonar

One of the traditional approaches in mobile robot research has been the use of stereo vision systems to extract range information from pairs of images [29]. One of the difficulties in applying these techniques in real-world navigation is the fact that the intrinsic computational expense of extracting three-dimensional (3D) information from stereo pairs of images limits the number of points that can be tracked [40].

Real-time constraints preclude the obtention of dense 3D descriptions such as those presented in [16], [35], though recent work in the area shows very promising results [34]. Additionally, traditional stereo vision systems relied upon specific features such as high-contrast edges or points that could be easily tracked along several images [30].

As a result, practical real-world stereo vision navigation systems such as the ones described in [29], [40], [25] only build sparse depth maps of their surroundings, selecting points to be matched and tracked using an interest operator. Handling on the order of 30–50 points, the system described in [41] takes 30–60 s to generate a 3D map (on a VAX-11/780 processor).

These limitations led us to explore the use of an alternative kind of sensor that could deliver range information directly. Direct sonar range measurements promised to provide basic navigation and denser maps with considerably less computation. Additionally, we also became interested in exploring the composition of information from qualitatively different sensors, such as a sonar array and a stereo pair of cameras into a more complex and rich description of the robot's environment.

C. Sonar Applications in Robotics

Sonar range sensing is a mature technology but few applications until recently involved detailed map building. Traditionally, active and passive sonar systems have been used for military intelligence applications in marine environments. Active sonar systems are used for communications, navigation, detection, and tracking, while passive sonar systems are used in surveillance [3]. Some of this research is obviously classified. More recently, with the rapid increase of commercial and civilian activity in the oceans, nonmilitary marine sonar systems have spread. Typical applications include navigation, charting, and fishing. Generally speaking, these systems are characterized by detailed knowledge of the physical properties of the marine environment, by sophisticated sonar signal generation and processing devices [3], and by addressing the problem of localization and tracking of objects [17], as opposed to determining their shape.

An important area of application is medical imaging [8]. Ultrasound systems used in medicine are active and build maps for human perusal, but again depend on accurate physical models of the tissues that the sound traverses [20], and work with very small beam widths, about 1–3°. Typical frequencies used span the range from 1 to 20 MHz, and distances on the order of 10–50 cm are measured.

Other applications of sonar sensors include rangefinding devices for industrial control applications, as well as camera autofocus systems [38].

In the robotics area, ultrasonic range transducers have recently attracted increasing attention [1], [2]. This is due in part to their simplicity, low cost, and the fact that distance measurements are provided directly. Some research has focused specifically on the development of more elaborate beam-forming and detection devices (see, for example, [26]), or the utilization of phased array techniques [2], such as are used in an advanced side-looking sonar system for submersibles. Other efforts investigate the application of highly

sophisticated signal processing techniques [3] to complex sonar signals.

Specific applications of sonar sensors in robotics include simple distance measurements [15], localizing object surfaces [4], determining the position of a robot in a given environment, and some *ad hoc* navigation schemes [5].

Miller [27], [28] uses sonar sensors to determine the position of a robot. It assumes that an accurate map of the environment is known and performs a search to determine where the robot would have to be to explain a given set of distance readings. The method does not take into account the errors that occur in actual sonar data. A similar approach is used by Drumheller [9] who also pressuposes an accurate map of the environment, but is able to cope with noisy data.

An independent CMU sonar mapping and navigation effort [6], [7] used a narrower beam, formed by a parabolic reflector, to build a line-based description of the robot's surroundings. The sonar readings are interpreted by fitting line segments to the points detected and matching these to an *a priori* map of the environment of the robot. One difficulty with this approach was that the geometric interpretation is done very early and is difficulted by noise and uncertainty in the data.

D. Sonar-Based Mapping for Mobile Robots

By contrast, our own work has centered on the development of a system for sonar-based mapping and navigation for an autonomous mobile robot operating in unknown and unstructured environments. The system has no *a priori* map of its surroundings. Instead, it acquires data from the real world through a set of sonar sensors and uses the interpreted data to build a sonar map of the robot's operating environment.

In applying sonar range sensors to mobile robot mapping and navigation, we expected to obtain dense maps of the robot's environment with regions classified as *empty*, *occupied*, and *unknown*, with sufficient precision and detail so as to be useful for autonomous navigation. This includes path planning, obstacle avoidance, motion solving, and landmark recognition. Additionally, we planned to develop a hierarchy of representations, from data intensive maps where position details are stored, to symbolic representations suitable for high-level planning.

1) Approach: Our method starts with range measurements obtained from sonar units whose position with respect to the robot is known. Each sonar range measurement is interpreted as providing information about *probably empty* and *somewhere occupied* volumes in the space subtended by the sonar beam (in our case, a 30° cone in front of the sensor). This occupancy information is modelled by probability profiles that are projected onto a rasterized two-dimensional horizontal map, where empty, occupied, and unknown areas are represented. Sets of range measurements taken from multiple sensors on the robot and from different positions occupied by the robot as it travels provide multiple views that are systematically integrated into the sonar map. In this way, the accuracy and extent of the sonar map are incrementally improved and the uncertainty in the positions of objects is reduced. Overlapping empty areas reinforce each other, the
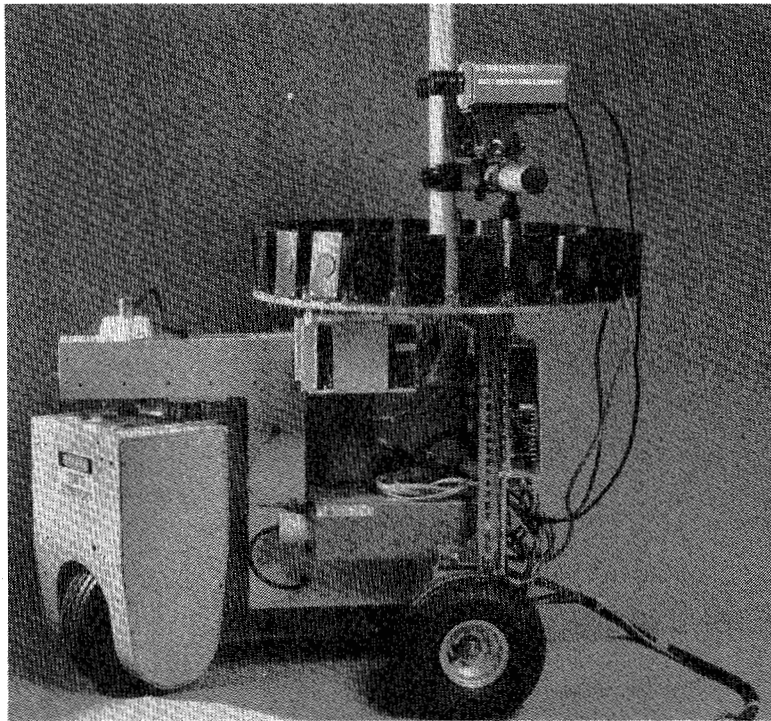
Fig. 2. Neptune mobile robot, with pair of cameras and sonar ring.

same happening with occupied areas. Additionally, empty regions serve to sharpen the boundaries of the occupied regions, so that the map definition improves as more readings are added. The final map shows probably occupied, probably empty, and unknown regions. This method deals effectively with noisy data and with the limitations intrinsic to the sensor.

For positional update as well as recognition of previously mapped areas, we developed a way of matching two sonar maps by convolving them. The method gives the displacement and rotation that best brings one map into registration with the other, along with a measure of the goodness of the match.

E. The Sonar System

1) The Sonar Sensor: The sonar devices being used are Polaroid laboratory grade ultrasonic range transducers [38]. These sensors have a useful measuring range of 0.9–35.0 ft. The main lobe of the sensitivity function is contained within a solid angle Ω of 30° and falls off to -38 dB. The beamwidth ω at -3 dB is approximately 15°. Experimental results showed that the range accuracy of the sensors is on the order of ± 0.1 ft. We are using the control circuitry provided with the unit, which is designed to return the distance to the nearest sound reflector in its field of view. The sensor interface does not provide information on multiple echos nor on the phase shift or the intensity of the detected echo.

2) The Sonar Sensor Array: The sonar sensor array, built at Denning Mobile Robotics, Inc., consists of a ring of 24 Polaroid ultrasonic transducers, spaced 15° apart. A Z80 controlling microprocessor selects and fires the sensors, times the returns and provides the corresponding range value. Over a serial link, this information is sent to a VAX mainframe, where currently the interpretation of the sonar data and the higher level mapping and navigation functions are performed.

3) The Robots: We have conducted several experiments by mounting the sonar sensor array on two currently available robots. The Neptune mobile robot [36] was developed at the Mobile Robot Laboratory of the Robotics Institute, Carnegie-Mellon University (CMU) (see Fig. 2). It has been used successfully in several areas of research, including stereo vision navigation [25], [41] and path planning [42]. Mounted on this vehicle, the sonar sensors are at an height of 31 in above the ground.

The Terregator robot (Fig. 3) is a larger vehicle developed by the Civil Engineering Robotics Construction Laboratory, CMU [23]. It has been used in several outdoor experiments, including road-following [43] and outdoor sonar navigation [12].

A new mobile robot, Uranus, incorporating an innovative omnidirectional design [37], is currently nearing completion and will be used for several current and new projects in the Mobile Robot Lab [31]. We plan to continue the sonar work using this vehicle.

F. Sonar Mapping

The sonar mapping process incorporates various stages. Initially, the sensor data are preprocessed, screened, and annotated with the corresponding sensor position. In sequence, the readings are interpreted using the probability density functions. A set of readings taken from one position of the robot is used to build a *view,* which stores the *empty, occupied,* and *unknown* areas as seen from that position. This view is then combined with the sonar map.

1) Problems with the Sonar Data: A number of problems are inherent to the data obtained from the sonar device and intrinsic to the sensor itself.

• The timing circuitry limits the range precision.

Fig. 3. Terregator outdoors robot.

- The detection sensitivity of the sensor varies with the angle of the reflecting object to the main beam axis.
- Sonar beams can suffer multiple reflections or specular reflections away from the sensor, giving false distance readings.
- Because of the relatively wide angle of the sonar beam, an isolated sonar reading imposes only a loose constraint on the position of the detected object.

These problems preclude a direct interpretation of the sonar readings and led us to consider a probabilistic approach to the interpretation of range data.

2) Preprocessing the Sonar Data: We begin by preprocessing the incoming data to remove easily detectable incorrect readings. This includes rejecting data below the lower range threshold $R_{\min}$ (usually due to faulty sensors) or above the useful range R_u of the device, as well as averaging multiple readings taken from the same sensor at the same robot position. The useful range is defined as $R_u = \min (R_g, R_{\max})$, with R_g being the range distance from the sensor to the ground and $R_{\max}$ the maximum sensor range.

3) Occupancy Probabilities: Due to the wide beamwidth, the data obtained from the sonar sensor provide only indirect information about the location of the detected objects. We combine the constraints from individual readings to reduce this uncertainty. Our inferences are represented as probabilities in a discrete grid.

A range reading is interpreted as making an assertion about two volumes in 3D space: one that is *probably empty* and one that is *somewhere occupied*. We model the sonar beam by two probability density functions, f_E and f_O, defined over these volumes. Informally, these functions measure our confidence concerning a point inside the cone of the beam being empty, and our uncertainty about the location of the surface patch that caused the echo, somewhere on the range surface of the cone. The probability density functions are defined based on the geometry of the beam and the spatial sensitivity pattern of the sonar sensor. They are parameterized by the range reading and the beamwidth.

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

R range measurement returned by the sonar sensor,
ϵ maximum sonar measurement error,
ω sensor beamwidth,
Ω solid angle subtending the main lobe of the sensitivity function,
S $= (x_s, y_s, z_s)$, position of the sonar sensor,
δ distance from P to S,
θ angle between the main axis of the beam and P as seen from S.

We now define two volumes of space in the sonar beam.

Probably empty region: This includes 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.
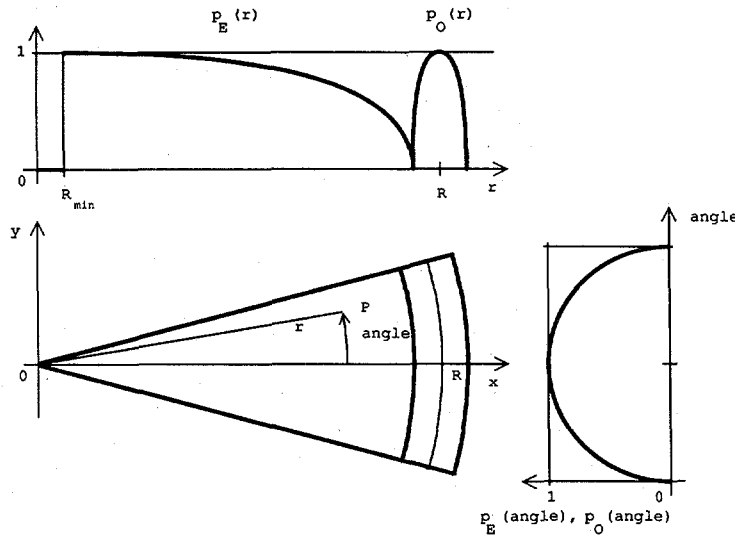
Fig. 4.   Probability profiles corresponding to Probably Empty and Somewhere Occupied regions in sonar beam. Profiles correspond to horizontal cross section of beam.

Somewhere occupied region: This includes 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.

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

$$p_E(x, y, z) = p \text{ [point } (x, y, z) \text{ is empty]}$$

$$= E_r(\delta) \cdot E_a(\theta) \qquad (3.1)$$

where

$$E_r(\delta) = 1 - ((\delta - R_{\min})/(R - \epsilon - R_{\min}))^2,$$

$$\text{for } \delta \in [R_{\min}, R - \epsilon], \quad (3.2)$$

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

and

$$E_a(\theta) = 1 - (2\theta/\Omega)^2 \qquad \text{for } \theta \in [-\Omega/2, \Omega/2]. \quad (3.3)$$

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

$$p_O(x, y, z) = p \text{ [position } (x, y, z) \text{ is occupied]}$$

$$= O_r(\delta) \cdot O_a(\theta) \qquad (3.4)$$

where

$$O_r(\delta) = 1 - ((\delta - R)/\epsilon)^2, \qquad \text{for } \delta \in [R - \epsilon, R + \epsilon], \quad (3.5)$$

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

and

$$O_a(\theta) = 1 - (2\theta/\Omega)^2, \qquad \text{for } \theta \in [-\Omega/2, \Omega/2]. \quad (3.6)$$

Note that, strictly speaking, p_E and p_O are not true probability density functions.

Fig. 4 shows the empty and occupied probability distributions for a sonar beam that returned a range reading R. The profiles shown correspond to a horizontal cross section of the sonar beam ($z = z_s$). For map building, these probability density functions are evaluated for each reading and projected on a horizontal two-dimensional grid.

4) Representing occupancy 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 $\Delta \times \Delta$. Each cell in the final sonar map contains its occupancy status (unknown, empty, or occupied) with an associated certainty factor, using the following convention:

unknown   0
empty   [-1, 0)
occupied   (0, 1].

A cell is considered unknown if no information concerning it is available. Cells can be empty with a certainty factor emp (x_i, y_j) (ranging from 0 to -1) and occupied with a certainty factor occ (x_i, y_j) (ranging from 0 to 1). The final map is computed from two separate arrays derived from the empty and occupied probability distributions introduced above. In the arrays themselves the empty and occupied probabilities are maintained as values ranging from zero to one.

5) Composing information from several readings: The sonar map is built by computing the empty and occupied sonar beam probability distributions for each range readings, projecting these probabilities onto the discrete cells of a view, and combining the view with the sonar map, which already stores the information derived from other readings. The position and orientation of the sonar sensors is used to register the view with the map.

Each sonar reading provides partial evidence about a map cell being occupied or empty. Different readings asserting that a cell is empty will confirm each other, as will readings

implying that the cell is occupied. On the other hand, evidence that the cell is empty will weaken the certainty of it being occupied and vice versa.

The operations performed on the empty and occupied probabilities are not symmetrical. The probability distribution for empty areas represents a space whose totality is probably empty, while the occupied probability distribution for a single reading represents a lack of knowledge concerning the location of the reflecting object, somewhere on the front of the beam. Empty regions are simply combined using a probabilistic addition formula. The occupied probabilities for a single reading, on the other hand, are initially weakened by conflicting data and then normalized to make their sum unity. Only after this narrowing process are the occupied probabilities from each reading combined, using again a probabilistic 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. Possibly occupied regions also increase, while the fuzziness of these regions decreases. The overall effect as more readings are added is a gradually increasing coverage along with an increasing precision in object locations. Correct information is incrementally enhanced and wrong data are progressively canceled out. Typically, after a few hundred readings (and less than a second of computer time) the method is able to "condense out" a comprehensive map covering a thousand square feet with up to 0.1-ft position accuracy in the position of detected objects. 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 feet map may contain only a hundred linear feet of boundary.

Formally, the evidence combination process proceeds along the following steps.

1) Initialization: The sonar map is set to unknown.

2) Superposition of empty areas: The probabilities corresponding to empty areas are combined using the probability addition formula p_E (cell) := p_E (cell) + p_E (reading) − p_E (cell) × p_E (reading).

3) Superposition of occupied areas: The probabilities corresponding to occupied areas are initially weakened by the evidence of the empty certainty factors, using p_O (reading) := p_O (reading)·(1 − p_E (cell)). We then normalize the occupied probabilities over the beam front. Finally, the occupied probabilities are combined using p_O (cell) := p_O (cell) + p_O (reading) − p_O (cell) × p_O (reading).

4) Thresholding: The final occupation value attributed to a cell in the sonar map is given by comparing the relative strengths of the empty and occupied values.

G. Results

Fig. 5 shows a typical sonar map obtained using the method outlined, with the conflicting information still superposed. The corresponding occupied and empty certainty factor distributions are shown in Figs. 6 and 7. These are the maps obtained before the thresholding step. The final maps obtained after thresholding are shown in Figs. 8–10. These maps correspond

to the Mobile Robot Lab at CMU. In Figs. 5 and 8 the outline of the room and the major objects is shown, as well as the positions of the robot from where the readings were taken. The 3D plots correspond to a view from the lower left corner of the room.

The resulting sonar maps are very useful for navigation and landmark recognition. They are much denser than the ones generated by our stereo vision programs and computationally about an order of magnitude faster to produce. We have demonstrated an autonomous navigation system [12], discussed in Section V, that uses an A*-based path planner to obtain routes in these maps. The system was tested in cluttered indoor environments using Neptune, and outdoors in open spaces, operating among trees, using the Terregator.

H. Matching

One useful capability in robot navigation is the ability to match sets of observations against each other. Possible applications include landmark recognition and updating the robot's estimate of its position and orientation.

Towards this end, Moravec developed a method that can match two maps and report the displacement and rotation that best takes one into the other [32]. The sonar maps described in the previous section are used.

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 labeled occupied. A revised matching procedure compares maps A and B through trial transformation T (represented by a 2 × 2 rotation matrix and a two-element displacement vector) by enumerating the occupied cells of A and transforming the coordinates of each such cell in A 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 coordinates 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 coordinates of occupied cells. The cost grows as $O(n^4)$, and the typical VAX running time is down to a few minutes.

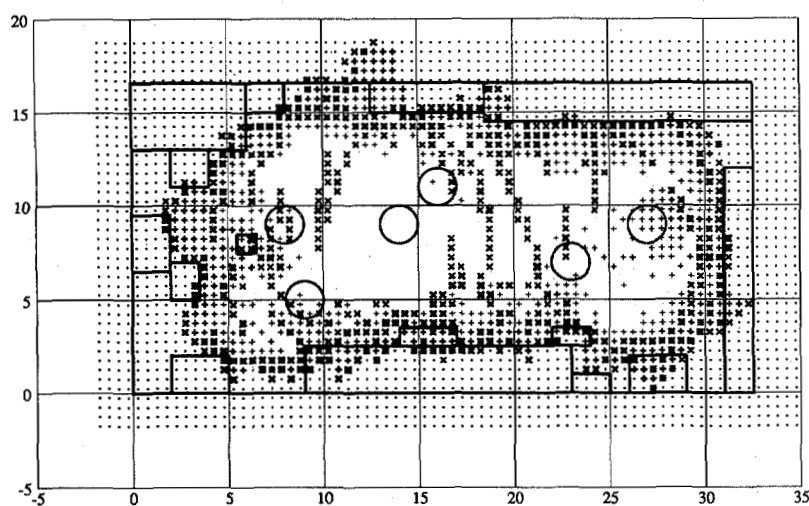A further speedup is achieved by generating a hierarchy of

Fig. 5. Two-dimensional sonar map. Each symbol represents square cell 0.5 ft on side. Empty areas with high certainty factors are represented by white space; lower certainty factors by + symbols of increasing thickness. Occupied areas are represented by × symbols, and unknown areas by ·. This map still shows conflicting information superposed. Robot positions where scans were taken are shown by circles and outline of room and of major objects by solid lines. Experiment was done in Mobile Robot Lab.
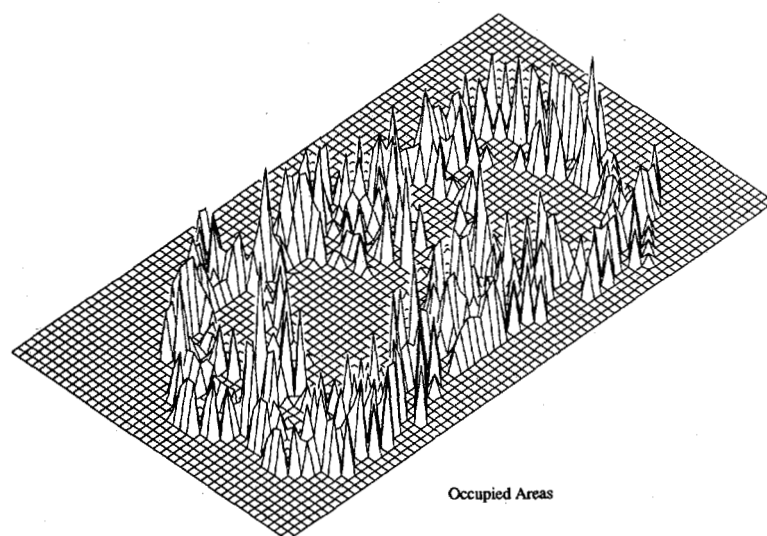

Fig. 6. Occupied areas in sonar map. This 3D view shows certainty factors occ (X, Y).
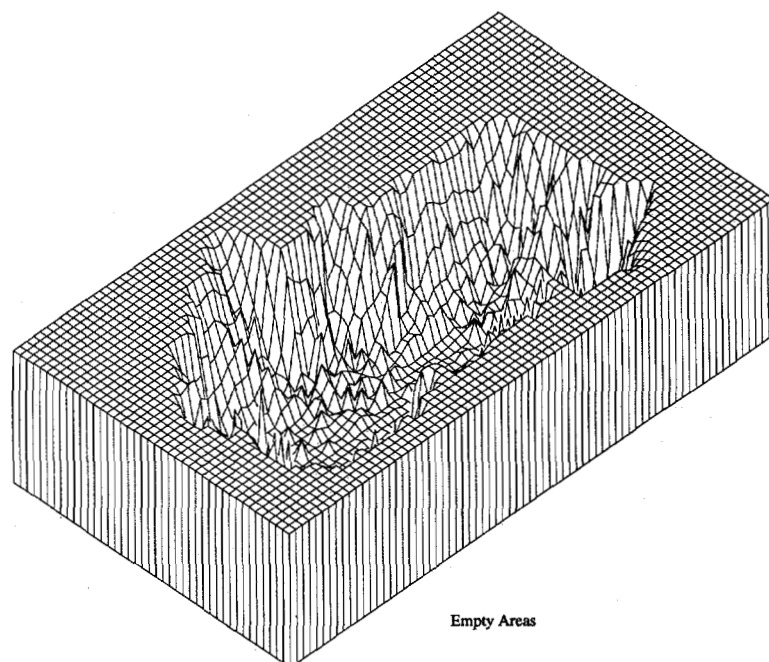

Fig. 7. Empty areas in sonar map. This 3D view shows certainty factors emp (X, Y).
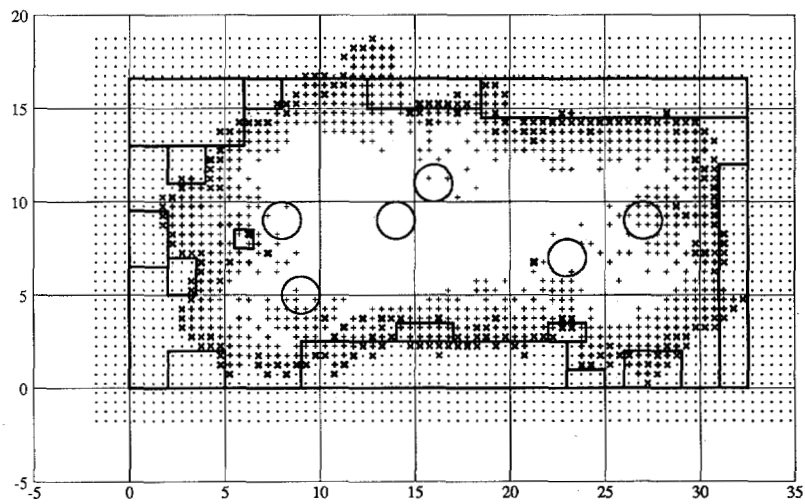
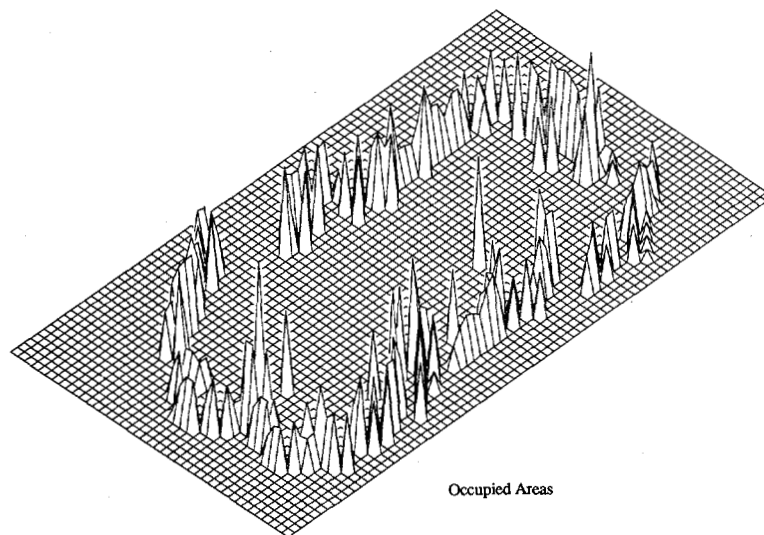Fig. 8.   Two-dimensional sonar map after thresholding.

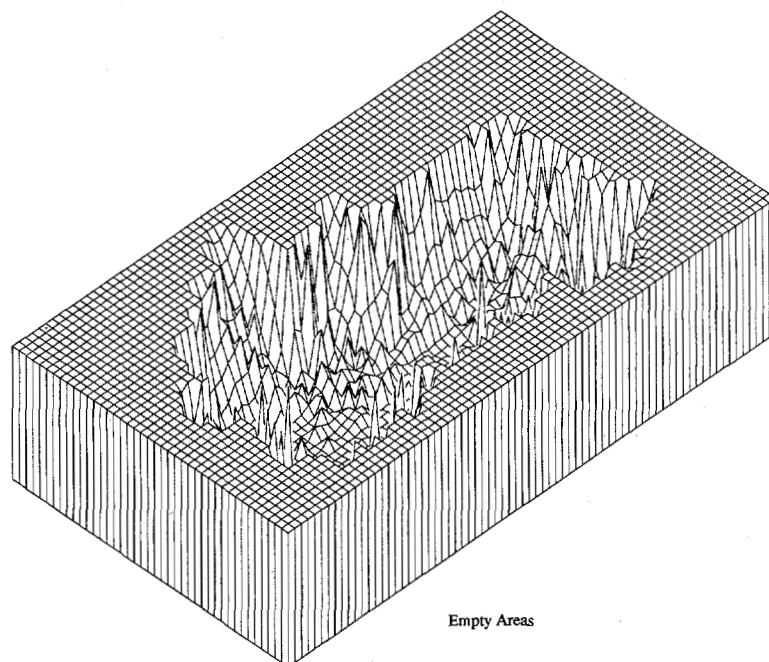Fig. 9.   Occupied areas in sonar map after thresholding.

Fig. 10.   Empty areas in sonar map after thresholding.

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 practical VAX matching times to under a second.

We found one further preprocessing step is required to make the matching process work in practice. Raw maps at typical resolutions (6-in cells) produced from moderate numbers on sonar measurements (about 100) have narrow bands of cells labeled "occupied." In separately generated maps of the same area the relative positions of these narrow bands shift 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 and only the lower frequencies carry information. The problem can be fixed by filtering (blurring) the occupied cells to remove the high-frequency noise. Experiments suggest that a map made from 100 readings should be blurred with a spread of about 2 ft, while for a map made from 200 readings a 1-ft smear is adequate. Blurring increases the number of cells labeled "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 full process outlined here, maps with about 3000 6-in cells made from 200 well-spaced readings of a cluttered 20 by 30-ft room can be matched with an accuracy of about 6 in displacement and 3° rotation in one second of VAX time.

IV. MULTIPLE AXIS OF REPRESENTATION OF SONAR MAPPING INFORMATION

In the previous section we have shown how the range data acquired from the real world through a sonar sensor array are interpreted and used to build a sonar map. These probabilistic local maps, described earlier, are the starting point for building a multileveled and multifaceted description of the robot's operating environment. In this section, we briefly describe these multiple axes of representation of mapping information and mention how they are used in different kinds of navigational activities [11]. We define the following axes of representation (Fig. 11).

1) The Abstraction Axis: Along this axis we move from a sensor-based low-level data-intensive representation to increasingly higher levels of interpretation and abstraction. Three levels are defined as the sensor level, the geometric level, and the symbolic level.

2) The Geographical Axis: Along this axis we define views, local maps, and global maps, depending on the extent and characteristics of the area covered.

3) The Resolution Axis: Sonar maps are generated at different values of grid resolution for different applications. Some computations can be performed satisfactorily at low levels of detail, while others have to be done at high or even multiple degrees of resolution.

A. The Abstraction Axis

The first kind of sonar map built from the sonar range data uses the *probabilistic* representation described earlier. A two-dimensional grid covering a limited area of interest is used. This map is derived directly from the interpretation of the sensor readings and is, in a sense, the description closest to the real world. It serves as the basis from which other kinds of representations are derived. Along the abstraction axis, this data-intensive representation is also called the *sensor level map*.

The second level is called the *geometric level*. It is built by scanning the sensor level map and identifying groups of cells with high occupied confidence factors. These are merged into uniquely labeled objects with explicitly represented polygonal boundaries (see Fig. 14). If needed, the same can be done with empty areas.

The third is the *symbolic level,* where maps of larger areas (typically global maps) are described using a framelike representation. This description bears only a topological equivalence to the real world. Nodes may represent "interesting" areas, where more detailed mapping information is necessary or available, or may correspond to simpler or "uninteresting" areas (navigationally speaking), such as corridors.

Different kinds of problem-solving activities are better performed on different levels of abstraction. For example, global path planning (such as how to get from one building wing to another) would be done on the symbolic level, while navigation through a specific office or lab uses the sensor-level map, where all the detailed information about objects and free space, as well as the associated certainty factors, is stored.

B. The Geographical Axis

To be able to focus on specific geographical areas and to handle portions of as well as complete maps, we define a hierarchy of maps with increasing degrees of coverage. Progressing along the geographical axis, we start with a *view,* which is a map generated from scans taken from the current position and which describes the area visible to the robot from that position. As the vehicle moves, several views are acquired and combined into a *local map.* The latter corresponds to physically delimited spaces such as labs or offices, which define a connected region of visibility. Global Maps are sets of several local maps and cover wider spaces such as a whole wing of a building, with labs, offices, open areas, corridors,
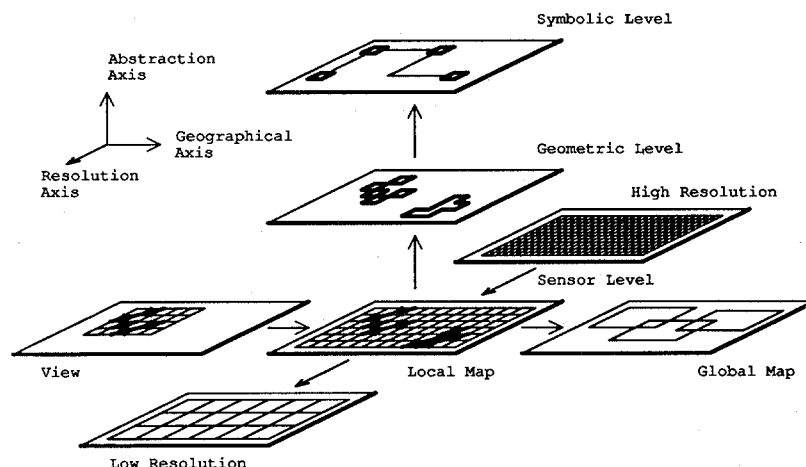
Fig. 11.   Multiple axis of representation of sonar maps.

etc. The global map stores information about the whole known environment of operation of the robot.

C. The Resolution Axis

Finally, along the resolution axis, we again start with the sensor-level local map and generate a progression of maps with increasingly less detail. This allows certain kinds of processing to be performed at lower levels of resolution with correspondingly less computational expense. Alternatively, it enables operations at coarser levels to guide the problem-solving activities at finer levels of resolution.

The most detailed sonar maps that can be obtained from the method outlined in Section III-F (considering the intrinsic limitations of the sensors) have a cell size of 0.1×0.1 ft. For navigation purposes, we have typically been using a 0.5-grid for indoors and a 1.0-ft grid for outdoors. Nevertheless, several operations on the maps are expensive and are done more quickly at even lower levels of resolution. For these cases we reduce higher resolution maps by an averaging process that produces a coarser description, as discussed in Section III-H.

V. SYSTEM ARCHITECTURE

To provide a context for the multiple descriptions introduced above, we present in this Section the overall architecture of the Dolphin sonar-based mapping and navigation system. The functions of the major modules and their interaction with the various sonar map representations are discussed, and the results of an actual run are shown.

The Dolphin system is intended to provide sonar-based mapping and navigation for an autonomous mobile robot operating in unknown and unstructured environments. Conceptually, two modes of operation are possible: in the cruising mode, the system acquires data, builds maps, plans paths, and navigates towards a given goal. In the exploration mode, it can wander around and collect enough information so as to be able to build a good description of its environment. The system is intended for indoor as well as outdoor use; it may be coupled to other systems, such as vision [23], to locate landmarks that would serve as long-range destinations.

A. Sonar-Based Mapping and Navigation System Architecture

The overall architecture of the sonar mapping and navigation system is shown in Fig. 12. The function of the major

modules and their interaction with the different sonar map representations are described below [11]:

|	
sonar control	interfaces to and runs the sonar sensor array, providing range readings;
scanner	preprocesses and filters the sonar data and annotates it with the position and orientation of the corresponding sensor, based on the robot's motion estimate;
mapper	using the information provided by the scanner, generates a view obtained from the current position of the robot (this view is then integrated into a local map);
cartographer	aggregates sets of local maps into global maps and provides map handling and bookkeeping functions;
matcher	matches a newly acquired sonar map against already stored local maps for operations such as landmark identification or update of the robot's position estimate;
object extraction	provides geometric information about obstacles (objects are extracted by merging regions of occupied cells and determining the corresponding polygonal boundaries); a region-coloring approach is used for unique labeling;
graph building	generates a frame-based symbolic description of the environment;
path planning	can occur on three different levels: *symbolic* path planning is done over wider areas (global maps) and at a higher level of abstraction (symbolic maps); *geometric* path planning can be used as an intermediary stage, when the uncertainty in local maps is low, and has the advantage of being faster than finding routes in the sensor map; finally, *sensor map* path planning generates detailed safe paths (the latter performs an A* search [18] over the map cells, with the cost function
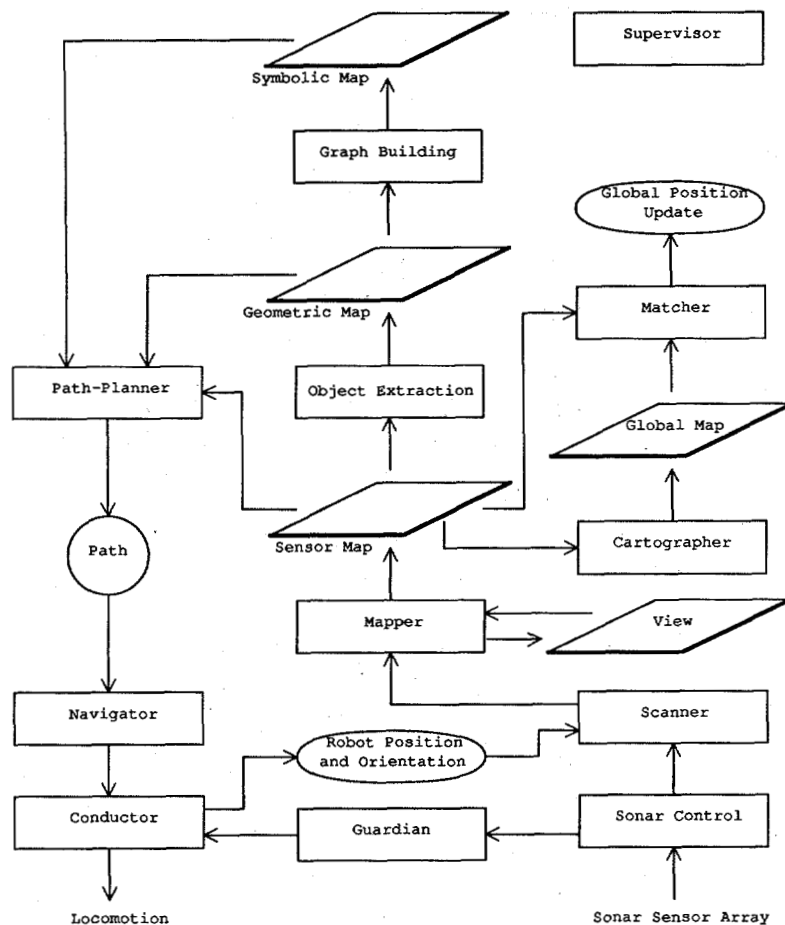
Fig. 12.   Architecture of Dolphin sonar mapping and navigation system.

taking into account the occupied and empty certainty factors, as well as the unknown areas and the distance to the goal; the path found by this module is provided to the navigator);

navigator   takes care of the overall navigation issues for the vehicle (includes examining already planned paths to determine whether they are still usable, invoking the path-planner to provide new paths, setting intermediary goals, overseeing the actual locomotion, etc.);

conductor   controls the physical locomotion of the robot along the planned path (the latter is smoothed and approximated by sequences of line segments, using a line-fitting approach; this module also returns an estimate of the new position and orientation of the robot);

guardian   during actual locomotion, this module continuously checks the incoming sonar readings and signals a stop if the robot is coming too close to a (possibly moving) obstacle not detected previously. It serves as a "sonar bumper";

supervisor   oversees the operation of the various

modules and takes care of the overall control of the system. It also provides a user interface.

We are currently working on a computational definition of the symbolic level and the corresponding mechanisms to extract this description from the sonar maps. All other subsystems mentioned have been implemented.

Comparing this architecture with the conceptual framework outlined in Section II, we can identify an immediate correspondence between the subsystems of the Dolphin system and some of the processing levels described previously: the sonar control and conductor modules belong to level I; scanning and mapping provide functions on level II; the object extraction, graph building, cartographer, and matcher operate on level IV; path planning, navigation, and the guardian are situated in level V; and the supervisor is on level VIII.

B. Tests of the System

The Dolphin system described in this section was tested in several indoor runs in cluttered environments using the Neptune mobile robot. Additionally, it was also tested in outdoor environments, operating among trees, using the Terregator robot, as part of the CMU ALV project. The system operated successfully in both kinds of environments, navigating the robot towards a given destination.

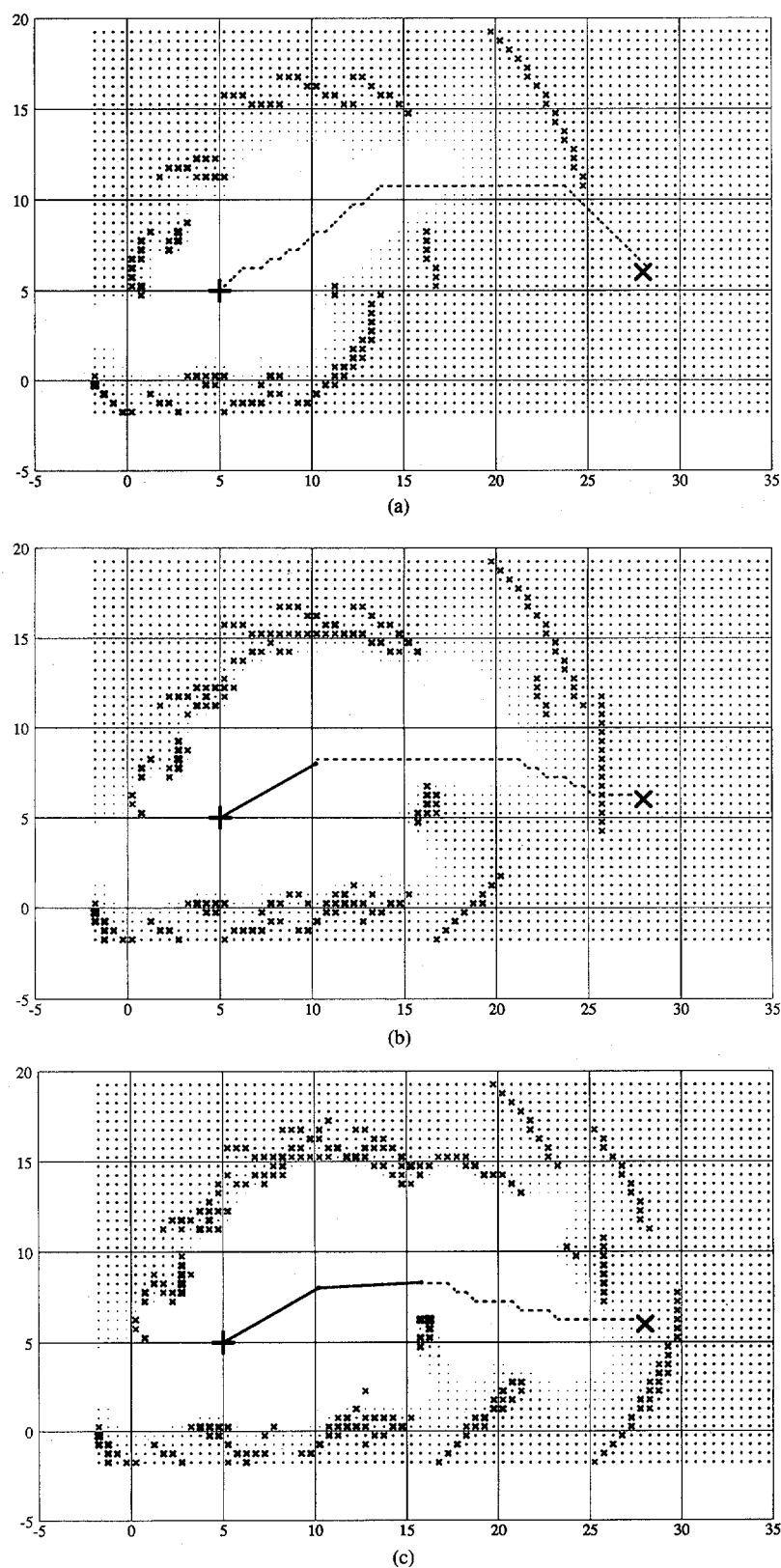In Fig. 13, an example run is given. The sequence of maps

Fig. 13.   Example run. This run was performed indoors, in Mobile Robot Lab. Distances are in ft. Grid size is 0.5 ft. Planned path is shown as dotted line, and route actually followed by robot as solid line segments. Starting point is solid + and goal, solid ×.
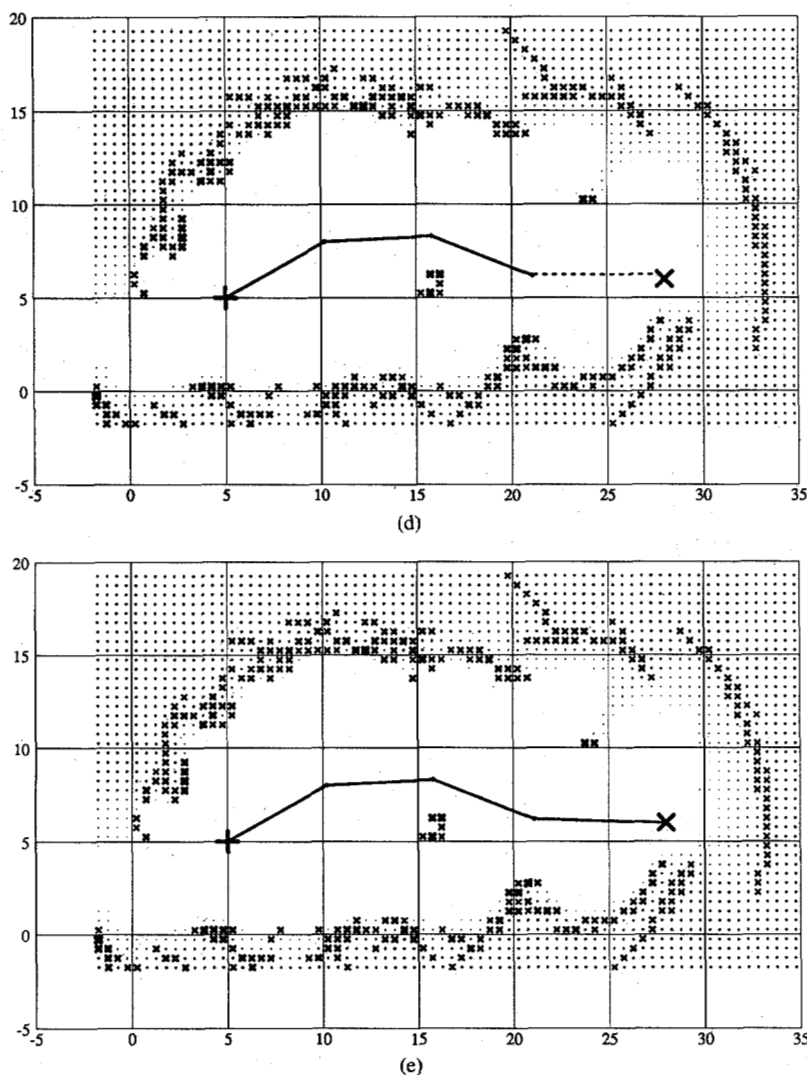
(d)

(e)

Fig. 13. (*Continued*).

shows how the sonar map becomes gradually more detailed and how the path is improved as more information is gathered. This example corresponds to an indoor run carried out in the Mobile Robot Lab. A distance of approximately 25 ft was covered; the grid size is 0.5 ft. Objects present in the lab included chairs, tables, boxes, workstations, filing cabinets, etc.

In Fig. 14, an outdoor run, together with an example of the object extraction module, is shown. The geometric map is build by scanning the sonar map and using a region-coloring technique to extract the obstacles. The objects are uniquely labeled and the polygonal boundaries are determined. The map shown corresponds to an outdoor run in Schenley Park, among trees. A distance of approximately 50 ft was traversed. The grid size was 1.0 ft, which proved adequate for navigation but did not allow a more precise description of the actual boundaries of the detected objects.

VI. FURTHER RESEARCH

We are currently extending the research described in this paper by pursuing several topics. These include

- introducing the robot position uncertainty into the view

registration and map making process;

- performing motion solving by matching a set of readings taken from a new robot position against the sonar map constructed so far;

- investigating issues in sensor integration in the specific context of combining sonar maps with 2D stereo data [39];

- exploring better sonar beam models by using Gaussian distributions in the empty and occupied probability density functions, and by taking into account the dependency of the beamwidth on the range measured;

- implementation of a distributed version of the Dolphin system as an actual test of the distributed control system described in [13].

We also mention that plans currently exist to extend the sonar-mapping method to do three-dimensional modeling. For that, a sonar sensor array that covers part of the surface of a sphere will be constructed [33].

VII. CONCLUSION

Research in mobile autonomous vehicles provides a very rich environment for the development and tests of advanced
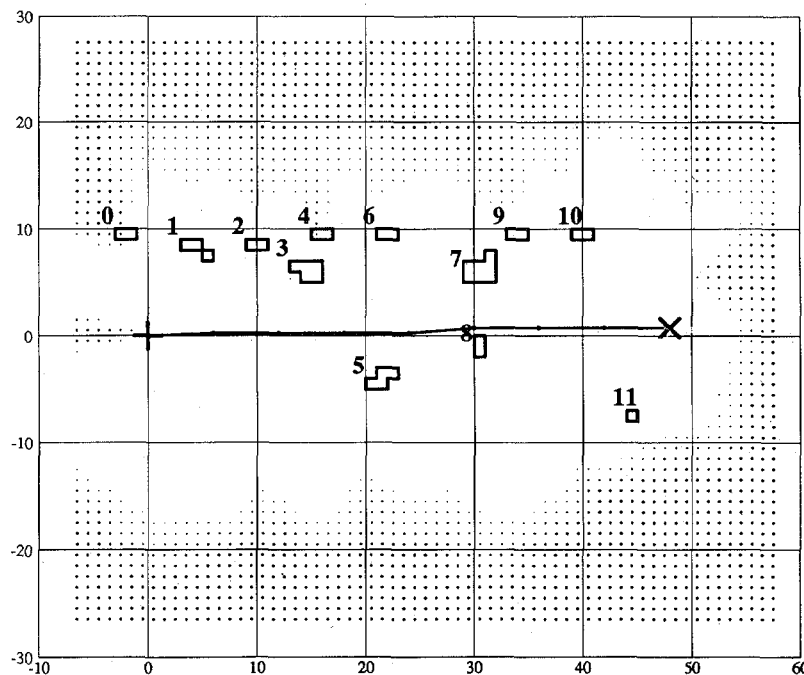
Fig. 14. Objects extracted from sonar map. Objects are uniquely labeled and their polygonal boundaries are shown. This map shows outdoor run, and objects are trees. Distances are in ft. Grid size is 1.0 ft.

concepts in a variety of areas such as robotics, artificial intelligence, sensor understanding and integration, real-world modeling, planning, and high-level control. Much work in mobile robots has either concentrated on very specific subareas, such a path planning, or has been conceptual work that never made it to a real application. This is partially due to the complexity of the overall task, the unavailability of adequate testbeds and the difficulty of doing actual experiments in something close to real time. This paper has presented a sonar-based mapping and navigation system that is able to operate in unknown and unstructured environments. It provides a sufficiently rich description of the robot's environment to call for more complex tasks; additionally, this is done in a sufficiently real-time situation so as to allow actual experiments to be done within a reasonable time frame.

Frequently, the real-world information available through sensors is quite inadequate. Stereo systems used in mobile robot research build a very sparse description of the robot's surroundings. Some of the range-based systems build denser descriptions but do not cope well with measurement errors. The representation chosen for sonar maps provides a way of explicitly describing unknown, empty, and occupied areas and of reasoning about them. This, coupled with a robust method for combining multiple sensor readings, allows us to cope with uncertainties and errors in the data and provides dense sonar maps, useful for navigation.

ACKNOWLEDGMENT

REFERENCES

[1] U. Ahrens, "Möglichkeiten und Probleme der Anwendung von Luft-Ultraschallsensoren in der Montage-und Handhabungstechnik," *Robotersysteme,* vol. 1, 1, 1985.

[2] ——, "Möglichkeiten und Grenzen des Einsatzes von Luft-Ultraschallsensoren in der Montage-und Handhabungstechnik," *Robotersysteme,* vol. 1, 4, 1985.

[3] A. B. Baggeroer, Sonar Signal Processing, in *Applications of Digital Signal Processing,* Signal Processing Series. Englewood Cliff, NJ: Prentice-Hall, 1978.

[4] M. K. Brown, "Locating object surfaces with an ultrasonic range sensor," in *Proc. 1985 IEEE Int. Conf. Robotics and Automation,* St. Louis, MO, Mar. 1985.

[5] A. Chattergy, "Some heuristics for the navigation of a robot," Robotics Res. Lab., Dep. Elec. Eng., Univ. of Hawaii, Honolulu, 1984.

[6] J. L. Crowley, "Position estimation for an intelligent mobile robot," in *1983 Annu. Research Rev.,* Robotics Inst., Carnegie-Mellon Univ., Pittsburgh, PA, 1984.

[7] ——, "Dynamic world modelling for an intelligent mobile robot using a rotating ultra-sonic ranging device," in *Proc. 1985 IEEE Int. Conf. Robotics and Automation,* St. Louis, MO, Mar. 1985.

[8] G. B. Devey and P. N. T. Wells, "Ultrasound in medical diagnosis," *Sci. Amer.,* vol. 238, May 1978.

[9] M. Drumheller, "Mobile robot localization using sonar," Artificial Intelligence Lab., Mass. Inst. Technol., AI-M-826, Jan. 1985.

[10] A. Elfes and S. N. Talukdar, "A distributed control system for the CMU rover," in *Proc. 8th Int. Joint Conf. Artificial Intelligence,* Karlsruhe, Germany, Aug. 1983.

[11] A. Elfes, "Multiple levels of representation and problem-solving using maps from sonar data," in *Proc. DOE/CESAR Workshop on Planning and Sensing for Autonomous Navigation,* Oak Ridge Nat. Lab., Univ. California, Los Angeles, Aug. 18–19, 1985.

[12] ——, "A sonar-based mapping and navigation system," in *1986 IEEE Int. Conf. Robotics and Automation,* San Francisco, CA, Apr. 7–10, 1986.

[13] ——, "A distributed control architecture for an autonomous mobile robot," *Int. J. Artificial Intelligence in Eng.,* vol. 1, Oct. 1986.

[14] O. D. Faugeras, "Object representation, identification, and positioning from range data," presented at the 1st Int. Symp. Robotics Research, Cambridge, MA, 1984.

[15] G. Giralt, R. Chatila, and M. Vaisset, "An integrated navigation and motion control system for autonomous multisensory mobile robots,"

presented at the 1st Int. Symp. Robotics Research, Cambridge, MA, 1984.

[16] W. E. L. Grimson, *From Images to Surfaces: A Computational Study of the Human Early Visual Systems.* Cambridge, MA: MIT Press, 1981.

[17] J. Hallam, "Resolving observer motion by object tracking," in *Proc. 8th Int. Joint Conf. Artificial Intelligence,* Karlsruhe, Germany, Aug. 1983, pp. 792–798.

[18] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst., Sci., Cybern.,* vol. SSC-4, 1968.

[19] M. Herbert and T. Kanade, "Outdoor scene analysis using range data," in *Proc. 1986 IEEE Int. Conf. Robotics and Automation,* San Francisco, CA, Apr. 7–10, 1986.

[20] M. Hussey, *Diagnostic Ultrasound: An Introduction to the Interactions between Ultrasound and Biological Tissues.* London: Blackie, 1975.

[21] R. A. Jarvis, "A perspective on range finding techniques for computer vision," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. PAMI-5, Mar. 1983.

[22] M. Julliere and L. Marce, "Contribution a l'autonomie des robots mobiles," Lab. d'Applications des Techniques Electroniques Avancées, Inst. Nat. des Sciences Appliqués, Rennes, France, 1982.

[23] T. Kanade and C. E. Thorpe, "CMU strategic computing vision project report: 1984 to 1985," Robotics Inst., Carnegie-Mellon Univ., Pittsburgh, PA, CMU-RI-TR-86-2, Nov. 1985.

[24] M. Lionel, "Contribution a l'autonomie des robots mobiles," Ph.D. dissertation, L'Institut National des Sciences Appliquées de Rennes et L'Université de Rennes I, Rennes, France, July 1984.

[25] L. H. Matthies and C. E. Thorpe, "Experience with visual robot navigation," in *Proc. IEEE Oceans 84,* Washington, DC, Aug. 1984.

[26] G. L. Miller, R. A. Boie, and M. J. Sibilia, "Active damping of ultrasonic transducers for robotic applications," in *Proc. Int. Conf. Robotics,* Atlanta, GA, Mar. 1984.

[27] D. Miller, "Two dimensional mobile robot positioning using onboard sonar," in *Pecora IX Remote Sensing Symp. Proc.,* Sioux Falls, SD, Oct. 1984.

[28] ——, "A spatial representation system for mobile robots," in *Proc. 1985 IEEE Int. Conf. Robotics and Automation,* St. Louis, MO, Mar. 1985.

[29] H. P. Moravec, "Obstacle avoidance and navigation in the real world by a seeing robot rover," Ph.D. dissertation, Stanford Univ., Sept. 1980 (also available as Stanford AIM-340, Cs-80-813 and CMU-RI-TR-01-82, 1982; and published as *Robot Rover Visual Navigation.* Ann Arbor, MI: UMI Research Press, 1981.

[30] ——, "The Stanford cart and the CMU rover," *Proc. IEEE,* vol. 71, July 1983.

[31] H. P. Moravec *et al.,* "Towards autonomous vehicles," in *1985 Robotics Research Review,* Robotics Inst., Carnegie-Mellon Univ., Pittsburgh, PA, 1985.

[32] H. P. Moravec and A. Elfes, "High resolution maps from wide angle sonar," presented at the Int. Conf. Robotics and Automation, IEEE, Mar. 1985.

[33] H. P. Moravec, "Three-dimensional imaging with cheap sonar," in *Autonomous Mobile Robots: Annual Report 1985,* Mobile Robot Lab., Pittsburgh, PA, Tech. Rep. CMU-RI-TR-86-4, Feb. 1986.

[34] H. K. Nishihara and T. Poggio, "Stereo vision for robotics," presented at the 1st Int. Symp. Robotics Research, Cambridge, MA, 1984.

[35] Y. Ohta and T. Kanade, "Stereo by intra- and inter-scanline search using dynamic programming," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. PAMI-7, Mar. 1985.

[36] G. W. Podnar, M. K. Blackwell, and K. Dowling, "A functional vehicle for autonomous mobile robot research," CMU Robotics Inst., Apr. 1984.

[37] G. Podnar, "The Uranus mobile robot," Autonomous Mobile Robots: Ann. Rep. 1985, Mobile Robot Lab., Pittsburgh, PA, Tech. Rep. CMU-RI-TR-86-4, Feb. 1986.

[38] *Ultrasonic Range Finders.* Polaroid Corporation, 1982.

[39] B. Serey and L. Mattheis, "Obstacle avoidance using 1-D stereo vision," to be published.

[40] C. E. Thorpe, "The CMU rover and the FIDO vision and navigation system," presented at the Symp. Autonomous Underwater Robots, Univ. New Hampshire, Marine Systems Engineering Lab., May 1983.

[41] C. E. Thorpe, "FIDO: Vision and navigation for a robot rover," Ph.D. dissertation, Dep. of Comput., Sci., Carnegie-Mellon Univ., Pittsburgh, PA, Dec. 1984.

[42] C. E. Thorpe, "Path relaxation: Path planning for a mobile robot," CMU Robotics Inst., CMU-RI-TR-84-5, Apr. 1984; also in *Proc. IEEE Oceans 84,* Washington, DC, Aug. 1984, and *Proc. AAAI-84,* Austin, TX, Aug. 1984.

[43] R. S. Wallace, K. Matsuzaki, Y. Goto, J. Webb, J. Crisman, and T. Kanade, "Progress in robot road following," in *Proc. 1986 IEEE Int. Conf. Robotics and Automation,* San Francisco, CA, Apr. 1986.

Alberto Elfes (S'82–M'86) received the E.Eng. degree in electronics engineering in 1975 and the M.S. degree in computer science in 1980, both from the Instituto Tecnológico de Aeronáutica (ITA), Brazil. He is completing his Ph.D. degree in the Electrical and Computer Engineering Department, Carnegie-Mellon University.

He is an Assistant Professor in the Computer Science Department of ITA, currently on leave. He has been a Research Assistant with the Mobile Robot Lab of the Robotics Institute, CMU, since 1982 and is a Research Associate with the Engineering Design Research Center since 1986. His research interests include applications of Artificial Intelligence in Robotics, Computer Vision, Sensor Integration, Mobile Robots, and the use of AI in the CAD of processors and VLSI systems. His previous research includes work in satellite image processing, pattern recognition, knowledge representation, and description of decision processes in medical diagnosis. At CMU, he has done research in distributed problem-solving, software architectures for mobile robots, planning, range-based mapping and autonomous navigation, sensor integration, real-world modeling, and geometric and spatial reasoning.

Mr. Elfes is a member of the Association for Computing Machinery.