Sonar Range Data Processing and Enhancement

Riccardo Cassinis, Paolo Venuti

Università di Udine, Dipartimento di Matematica e Informatica,
Via Zanon, 6, I-33100 Udine - Italy

Abstract

This paper deals with the problem of correcting and enhancing range data obtained from sonar sensors on mobile robots. The major sources of errors are examined, and a method is proposed that allows reasonable corrections to be applied, using knowledge of the sensor's characteristics and of the robot's operating environment.

The used algorithm is explained, and experimental results are presented.

1. Introduction

Sonar range data are widely used in mobile robots, both because of the low cost of related sensors, and because of their relative independence from the environmental situation (lighting, smoke, etc.). Depending on the structure and on the needs of the robot's control system, these data can be used as an indication of obstacles distance, to build environmental maps, or to determine the robot's position inside an already known environment.

Sonar range data are however affected by a number of errors, that can severely impair the quality of measurements. Besides those errors that depend on variations of the speed of sound (temperature, humidity and pressure of the transmitting medium) [6,8], and that can be quite easily corrected using appropriate auxiliary sensors, this paper deals with two other errors, that derive from the finite (and usually quite wide) opening of the sonar beam, and from the fact that under some conditions the sonar beam is mirror-reflected from the surfaces it hits, instead of being reflected back to the sensor.

The first of these errors yields as a consequence that, when using the sonar as an horizon scanner (and this applies both to rotating sonars and to sonar belts), all openings look smaller than they actually are, while obstacles look larger.

The second error causes "false openings" to appear in sonar scans: when the beam hits smooth surfaces under some critical angles, it is not reflected back to the sensor, but bounces away, thus showing openings where none are to be found, or, at least, giving erroneous distance measurements.

The paper deals with a method for processing sonar range readings taken from a rotating sonar mounted on top of a mobile robot, that eliminates or at least reduces the errors just discussed. The method uses both heuristic knowledge about the sensor and the common environment structure, and algebraic calculations for correcting data.

2. The problem.

Sonar range detectors are based on the time-of-flight principle. The time required for a burst of ultrasonic waves to travel from an emitter to an obstacle and for the reflected burst to travel from the obstacle to a receiver is proportional to the overall length of the burst's path, and inversely proportional to the speed of sound in the transmitting medium. In robot applications, the emitter and the receiver are usually co-planar and placed close to each other, or, as it happens in the system being described, they are integrated into the same transducer. Thus, the time elapsed from the burst emission to the echo reception can be used as an indication of the distance from the transducer to the nearest obstacle. Very sophisticated sensors have been made commercially available, which take into account differences in the reflecting surfaces, the attenuation in the transmitting media, etc. However, all these sensors suffer from the fact that the transducer irradiates waves in a solid volume, as it is shown in Fig. 1. The shape of the radiation diagram (and of the sensitivity diagram for the microphone) is due to wave physics considerations [8].

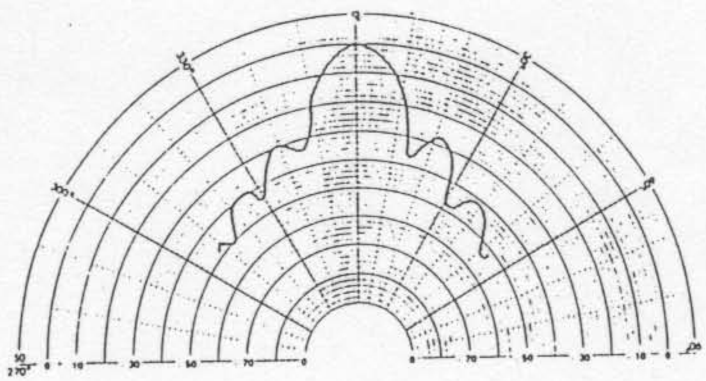


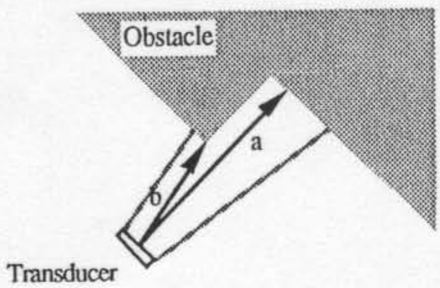
Fig. 1. - Typical sensitivity diagram for the transducer.

The main consequence is that, since electronic circuits are designed to compute the distance measuring the time it takes for the first echo to come back, the device, when pointing in a given direction, will actually measure the distance of the closest object within a solid angle, whose opening depends on the transducer's characteristics (Fig. 2). The obvious solution of narrowing the ultrasonic beam by using appropriate means (increasing the size of the transducer, for instance, or using acoustic lenses in front of it) is often impractical, because it would significantly increase the size and the weight of the transducer, which, specially for rotating devices, have to be kept to a minimum.

Other solutions, based for instance on wave interference techniques, can be used in some cases, but are quite complex and do not completely solve the problem.

The method proposed here, on the other hand, is based on a standard transducer, whose beam is a cone with an opening of about 30°.

(DSSIA 15



a: desired measurement
b: actual reading
Dotted lines indicate actual beam opening

Fig. 2. - Errors due to beam width.

The second mentioned problem is that, when the beam hits a solid surface, it is partly mirror reflected, and partly scattered. The ratio between reflected and scattered energy varies according to the smoothness of the surface being hit. The practical result is that, when the beam hits the surface under a "critical" angle (Fig. 3a), no echo will be detected, or a false echo will come back to the sensor after several meaningless reflections (Fig. 3b).

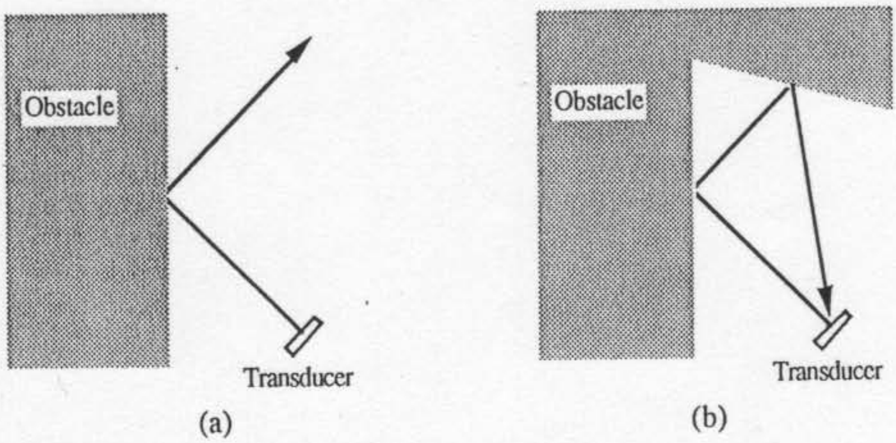


Fig. 3. - Errors due to false reflections.

3. The proposed solution.

The two mentioned problems require different processing steps in order to be solved. Thus, two separate algorithms have been developed. Since it is quite difficult, if not impossible, to completely remove the effects of the width of the sonar beam, attention was focused on the problem of locating openings and correcting their dimensions, since, for a mobile robot, the problem of understanding if it will be able to pass through a door is the most common one related to sonar data. It must be noted here that heuristic knowledge of the sensor and of the operating environment have been widely used. The proposed algorithm was specially tailored for indoor use in offices, houses, etc.; in order to make it suitable for use in other environments, some modifications to the rules or to some parameters should be applied.

The first algorithm presented here deals with the identification of openings and the correction of their apparent size; the second one is related to the elimination of false reflections.

3.1 - Detecting openings.

The algorithm's purpose is to provide as an output position and dimension of each detected opening, using as an input the polar diagram obtained from the sonar sensor. This diagram is provided as an array of range readings taken while rotating the sonar around its vertical axis. Although the number of readings per revolution can be varied according to the needs of the user, an array of 200 readings is normally used. This means that the angular distance between two consecutive readings is 1.8°.

The chosen approach is to locate the edges of openings where abrupt changes in range readings are detected. General ideas about this procedure can be found in [5].

Fig. 4 shows a practical polar diagram of sonar readings, which will be used in all following examples. Distances are in cm. A deep opening in the wall facing the robot can be seen.

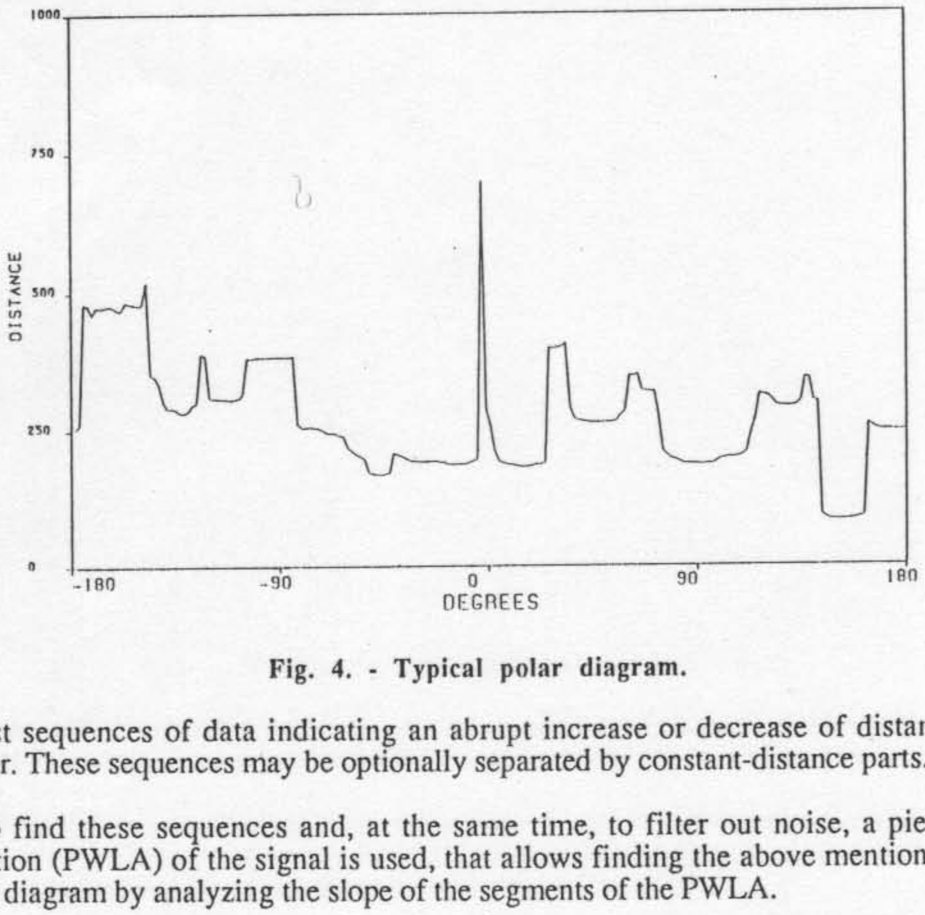


Fig. 4. - Typical polar diagram.

The longest sequences of data indicating an abrupt increase or decrease of distance should be searched for. These sequences may be optionally separated by constant-distance parts.

In order to find these sequences and, at the same time, to filter out noise, a piecewise linear approximation (PWLA) of the signal is used, that allows finding the above mentioned significant parts of the diagram by analyzing the slope of the segments of the PWLA.

The split-and-merge algorithm presented in [9,10,11] has been used. It provides an approximation of the kind

$$\sum_{i=1}^{m} a_i \phi_i (x)$$

where $\phi_i(x)$ is a set of linearly independent functions.

The general problem is: given a set of points $S=\{(x_i, y_i) \mid i=1, 2, ..., N\}$, determine the minimum number n such that S is partitioned in n subsets $S_1, S_2, ..., S_n$, in each one the given points being approximated by a polynomial of degree $\leq$m-1, with an error smaller than a preset amount.

In our case the approximation is done using a piecewise linear function, thus in the above formula $m=2$, $\phi_1(t)=1$, $\phi_2(t)=t$.

The problem then requires that on each subset the given points should be approximated by a straight line using the least squares method.

Details of the algorithm can be found in [12].

The output of the PWLA is a set of segments, each one defined as that part of the line $y=m_i x+q_i$, that ranges from $x1_i$ to $x2_i$. Thus, the algorithm's output is a list of records, each one containing m_i, q_i, $x1_i$ and $x2_i$ for segment i.

Slopes greater than a preset amount (STOL) will indicate abrupt increases in distance, while negative slopes whose absolute value is greater than STOL will indicate abrupt decreases.

The next step is to build a string (SLOPE) of characters, based on the previously mentioned list, which is suitable for further processing. In order to build such string, the values of m_i in the list are examined. For each segment, a character is added to SLOPE (which is initially empty) according to the following rule:

$SLOPE_i$="/" if $m_i > STOL$;
$SLOPE_i$="\" if $m_i < -STOL$;
$SLOPE_i$="0" if $|m_i| \leq STOL$.

A terminator ("$") is then added at the end of SLOPE.

This string is fed to a finite-state automaton, whose state table is the following:

	"0"	"/"	"\"	"$"
0	0,*nop*	1,*left*	0,*nop*	5,*nop*
1	2,*nop*	1,*nop*	3,*right*	5,*nop*
2	0,*nop*	1,*nop*	3,*right*	5,*nop*
3	4,*nop*	1,*do2*	3,*right*	5,*do1*
4	0,*do1*	1,*do1*	3,*right*	5,*do1*
5	-,*halt*	-,*halt*	-,*halt*	-,*halt*

The machine is initialized with i=1.

Actions associated with each state transition are as follows:

nop: $i \leftarrow i+1$;

left: $wl \leftarrow ABSC_i$;
$vl \leftarrow DIST_i$;
$i \leftarrow i+1$;

right: $wr \leftarrow ABSC_i$;
$vr \leftarrow DIST_i$;
$i \leftarrow i+1$;

do1: **if** $|vr - vl| < ETOL$ **then** output $\leftarrow$ wl, wr;
$i \leftarrow i+1$;

do2: **if** |vr - vl| < ETOL **then** output ← wl, wr;
wl ← $ABSC_i$;
vl ← $DIST_i$;
i ← i+1;

halt: the algorithm stops.

ETOL is the maximum allowable difference of distance between the edges of an opening. For instance, with reference to Fig. 5, |d1-d2| must be ≤ ETOL in order to consider the shadowed area as an opening.

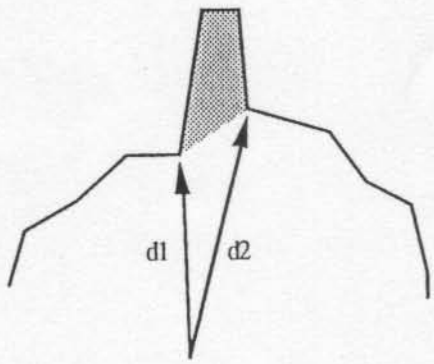


Fig. 5. - Explanation of the meaning of ETOL.

$ABSC_i$ contains the value of the abscissa of one of the edges of the segment being processed: the left edge if the slope is positive, and the right edge if the slope is negative. If the slope of the segment is below STOL, the value of $ABSC_i$ is undefined.

$DIST_i$ contains the distance value associated to $ABSC_i$.

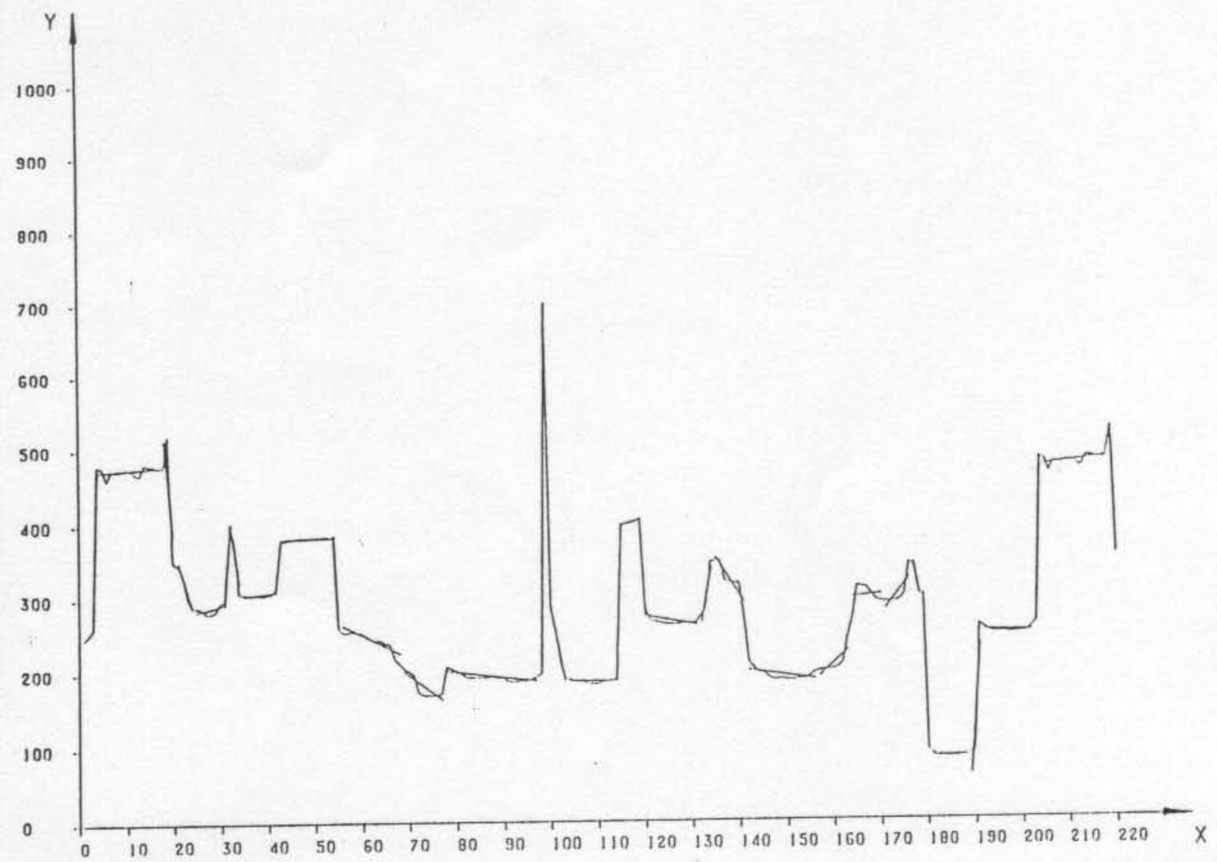
wl, wr, vl and vr are temporary storage variables.

output is a record containing the extremes of the detected opening.

Fig. 6 shows the output of the algorithm described so far, when the input is the one shown in Fig. 4.

The last step is a very simple one: once the extremes of openings have been found, their position is recomputed according to the opening of the sonar beam. This phase involves only shifting the edges on the polar diagram, according to the sensor's beam width.

The automaton's state table was designed so as to take into account complex situations: e.g., doors behind other doors.



string: 0/0\00/\00\000/\\0/0\0/00000\\0/0/0\

	X	Y		X	Y
Begin:	3	260	Begin:	114	190
Peak:	19	519	Peak:	119	405
End:	20	350	End:	120	289
Begin:	31	298			
Peak:	32	387			
End:	34	307			
Begin:	98	198			
Peak:	99	697			
End:	103	193			

Fig. 6. - Output of the openings detection algorithm.

3.2 - Detecting erroneous reflections.

The algorithm just described does not solve the second problem mentioned in Par. 2, i.e., the apparent openings that are detected when, due to mirror reflection, the ultrasonic beam "bounces away" instead of going back to the sensor.

It is important to note that there is no way of safely detecting such "false openings" with a set of readings taken from a single location, unless a map of the environment is already available. The only thing that can be done is to detect those portions of the signal where false reflections are likely to occur. In other words, it is possible to mark openings as "maybe openings" when some conditions are met, but in order to decide if these "maybe openings" are real ones, readings taken from several different positions in the environment are required.

The criterion that has been used is to measure the incidence angle of the sonar beam on detected walls close to detected openings. If this angle is smaller than a preset threshold, the detected opening is "suspicious", since it could be due to mirror reflection. The problem here is that the threshold depends from the material hit by the sonar beam: smooth materials, such as glass, have larger thresholds than rough ones.

The first step of the algorithm uses the list of openings provided by the previously described procedure. Each opening is examined to determine wether it could be a false one or not, computing the incidence angle. This can obviously be done only if the opening is surrounded by a long enough straight wall. If not, the opening is anyway considered to be questionable. All suspect openings are inserted in a list to be further processed.

The robot is then moved over a limited distance, and openings visible from the new position are searched for.

Since the robot's movements are only approximately known, the new position of the machine must be computed. In order to do this, a matching is performed among the previous and the actual horizon map. Since it is very difficult to make such matching directly using polar diagrams from the sensor, because coarse errors can be contained in readings, a more sophisticated procedure has to be used. The chosen procedure is described in [4, 7], and is based on probability maps. These maps are generated using probability distributions that allow identification of empty, occupied or unknown regions. The output is in the form of two matrices, whose elements represent portions of the environment surrounding the robot. For each element, the probability that it is empty (first matrix), or that it is occupied (second matrix) are given (Fig.7). The two matrices are computed using two different algorithms. It should also be noted that these maps are suitable for other purposes, such as navigation planning and environment recognition.

Fig. 7. An example of probability maps.

Once the maps pertaining to the starting and to the final position of the robot are obtained, they are matched to determine the new position of the machine. The used algorithm is still the one described in [4, 7], but, since, during the last movement, the robot cannot have moved outside a known area, the algorithm has been modified in such a way that only this area is checked for matching.

Proper coordinate transformations are then applied to the list of suspicious openings previously described. This list is then checked against the new openings list. Suspicious openings which have disappeared in the second list were obviously due to false reflections, and are therefore deleted.

A new environment map is then built combining the two previous maps, and will be used for further iterations of the algorithm.

4. Experimental results

As it was already said, the proposed method uses heuristic knowledge about the environment in which the robot operates. Due to this fact, it is difficult to evaluate the performance of the method, since it is difficult do decide what a "standard" environment is. Experiments carried on so far are related to the environment available in our laboratory, which is a quite normal office, with standard furniture (tables, chairs, etc.) and with a traditional structure (doors, corridors, usually straight angles between walls, etc.).

The program was first written in Pascal on a VAX computer, and used streams of range data coming from the sonar sensor of RISK robot [2, 3]. The robot has a modified Polaroid sensor [1, 6] mounted on its top (about 120 cm high). The sensor can be rotated using a stepper motor, that allows 200 steps/revolution, and yields data with a resolution of 1 cm over a range extending from 30 to 750 cm.

It was seen that the combined effect of the transducer's radiation diagram and of electronic circuits (filters and gain controls), limits the opening of the ultrasonic beam to an angle of 30°. The boundary of this beam is very steep. Therefore, it was assumed that the sensor can "see" every object inside the beam, and is completely "blind" to anything which is outside it, regardless of the distance.

As far as openings detection and correction is concerned, the algorithm works well (up to 100% correct recognitions), provided that the right thresholds are set. It must be noted however that the accuracy of the algorithm is inversely proportional to the distance: therefore, iterating the algorithm while the robot moves ensures that any mistakes will disappear as the robot approaches the opening.

The first part of the algorithm has O(n) complexity: with 200 readings/revolution the computing time is .16 seconds on a VAX 780. Obviously, transferring the algorithm on a dedicated microcomputer would significantly decrease computing times.

As far as the second part of the algorithm is concerned, a compromise must be found between the precision required in determining the new robot's position and the computing time.

In fact, the algorithm for generating maps has $O(n^3)$ complexity, and the matching algorithm has O(n log n) complexity, where n is the dimension of the matrix containing the map.

Actually, computing time is around 13 seconds for 32x32 maps, and 45 seconds for 64x64 maps. Although these times are too long for practical use, we believe that implementation on a dedicated computer would reduce them dramatically, because of some problems that were encountered in the VAX implementation; furthermore, it must be noted that the algorithm can be made much faster if parallel processing is available, or if it is integrated with other means for determining the robot's position. Once the position of the robot is known, the time required for comparing the openings lists and for deleting false openings is negligible.

5. Conclusions

A method has been presented that allows using inexpensive sonar sensors in a much more sophisticated way than they were originally intended. From a philosophical point of view, the importance of including knowledge about the operating conditions of sensors, instead of developing inefficient general-purpose algorithms was stressed. From a practical point of view, an algorithm was developed that allows simple use of sonars as an aid to mobile robots operation, removing inconvenients due to their characteristics, with no additional hardware requirements.

Bibliography

[1] Biber, C., Ellin, S., Shenk, E., and Stempeck, J.: *The Polaroid Ultrasonic Ranging System*, in *Polaroid Ultrasonic Ranging System Handbook,* Polaroid Corp., Cambridge, Mass., 1980.

[2] Cassinis, R., Biroli, E., Meregalli, A., and Scalise, F.: *Behavioral Model Architectures: a New Way of Doing Real-time Planning in Intelligent Robots,* in *Proc. SPIE's Cambridge '87 Symposium on Advances in Intelligent Robotics Systems,* Cambridge, Mass., 1987.

[3] Cassinis, R.: *BARCS: A New Way of Building Robots,* Internal Report 87/036, Department of Electronics, Politecnico di Milano, Milano, 1987.

[4] Elfes, A.: *Sonar-based Real-world Mapping and Navigation,* in *IEEE Journal of Robotics and Automation,* RA-3, 3, pag. 249-265, 1987.

[5] Horowitz, S.L.: *A Syntactic Algorithm for Peak Detection in Waveforms with Applications to Cardiography,* in *Communications of the ACM*, Vol. 18, 5, pag. 281-285, 1975.

[6] Maslin, G.D.: *A Simple Ultrasonic Ranging System,* in *Polaroid Ultrasonic Ranging System Handbook,* Polaroid Corp., Cambridge, Mass., 1980.

[7] Moravec, H.P., and Elfes, A.: *High Resolution Maps from Wide Angle Sonar,* in *Proc. IEEE Int. Conf. on Robotics and Automation*, Saint Louis, Missouri, pag. 128-135, 1985.

[8] Olson, H.F.: *Elements of Acoustical Engineering,* D. Van Nostrand Co., Inc., New York, 1947.

[9] Pavlidis, T., and Horowitz, S.L.: *Segmentation of Plane Curves,* in *IEEE Transactions on Computers*, C-23, 8, pag. 860-870, 1974.

[10] Pavlidis, T.: *Structural Pattern Recognition,* Springer-Verlag, Berlin, Heidelberg, New York, 1977.

[11] Pavlidis, T.: *Waveform Segmentation through Functional Approximation,* in *IEEE Transactions on Computers*, C-22, 7, pag. 689-697, 1973.

[12] Venuti, P.: *Metodi per il trattamento di segnali provenienti da sensori ultrasonici impiegati su robot mobili*, Master thesis, University of Udine, Udine, 1988 (in Italian).