Reprint of Proceedings of the 1990 IEEE International Conference on Robotics and Automation,
Cincinnati, Ohio, May 13-18, 1990, pp. 572-577.

Real-time Obstacle Avoidance for Fast Mobile Robots in Cluttered Environments[1]

by
J. Borenstein and **Y. Koren**

Department of Mechanical Engineering and Applied Mechanics
The University of Michigan, Ann Arbor

ABSTRACT

A new real-time obstacle avoidance method for mobile robots has been developed and implemented. This method, named the *Vector Field Histogram* (VFH), permits the detection of unknown obstacles and avoids collisions while simultaneously steering the mobile robot toward the target. A VFH-controlled mobile robot maneuvers quickly and without stopping among densely cluttered obstacles.

The VFH method uses a two-dimensional Cartesian *Histogram Grid* as a world model. This world model is updated continuously and in real-time with range data sampled by the onboard ultrasonic range sensors. Based on the accumulated environmental data, the VFH method then computes a one-dimensional *Polar Histogram* that is constructed around the robot's momentary location. Each sector in the *Polar Histogram* holds the *polar obstacle density* in that direction. Finally, the algorithm selects the most suitable sector from among all *Polar Histogram* sectors with low obstacle density, and the steering of the robot is aligned with that direction. Experimental results from a mobile robot traversing a densely cluttered obstacle course at an average speed of 0.7 m/sec demonstrate the power of the VFH method.

1. Introduction

In our previous research, we developed a real-time obstacle-avoidance algorithm for fast mobile robots, entitled the *Virtual Force Field* (VFF) method (Borenstein and Koren, 1988, 1989). This method used a force-field approach in which obstacles applied virtual repulsive forces to the robot, while the target applied a virtual attractive force. Force-field-based obstacle avoidance had been suggested by Khatib (1985), Krogh (1984), and Krogh and Thorpe (1986), although these methods were not used for real-time navigation.

By contrast, Brooks (1986) and Arkin (1989) use force-field methods on experimental mobile robots (equipped with ultrasonic sensors). However, in Brooks' implementation only the *current* set of range readings is used to compute the resultant repulsive force. This way, erroneous and "good" readings receive the same weight, making the algorithm more prone to sensor errors. Arkin's robot employs a similar method; his robot was able to traverse an obstacle course at 0.12 cm/sec (0.4 feet/sec).

We found, however, that the force-field approach has several severe limitations (Borenstein and Koren, 1990). For example, force-based obstacle-avoidance methods do not allow the robot to pass through narrow passages. Another problem is the instability of motion when traveling within narrow corridors; limitations that were also observed by Arkin (1989) and Tilove (1989).

This paper introduces our new *Vector Field Histogram* (VFH) method, a real-time obstacle-avoidance method for fast-running vehicles that constitutes a significant improvement over the *Virtual Force Field* method. VFH control results in smooth motion of the controlled vehicle among densely cluttered and unexpected obstacles. A VFH controlled vehicle can easily enter narrow passages and can travel in narrow corridors at high speeds and without oscillations. These features are made possible through a number of novel ideas that we have implemented as fast computer algorithms.

2. The Histogram Grid for Sensor-Based World Modeling

The VFH method uses a two-dimensional Cartesian *Histogram Grid* for the representation of obstacles. This rep-

[1]This work was sponsored by the Department of Energy Grant DE-FG02-86NE37969

resentation is derived from the certainty grid concept developed by Moravec and Elfes (1985) and Moravec (1988) at Carnegie Mellon University (CMU). Like the certainty grid, each cell in the *Histogram Grid* holds a certainty value, CV, that represents the confidence of the algorithm in the existence of an obstacle at that location. The *Histogram Grid* differs from the certainty grid in the way it is built and updated: CMU's method projects a probability profile onto those cells that are affected by a range reading. This procedure is computationally intensive and would impose a heavy time-penalty if real-time execution on an onboard computer was attempted.

Our method, on the other hand, creates a probability distribution with only little computational overhead. This effect is obtained by incrementing only one cell in the *Histogram Grid* for each range reading. For ultrasonic sensors, this cell corresponds to the measured distance d (see Fig. 1a) and lies on the acoustic axis of the sensor. While this approach may seem to be an oversimplification, a probabilistic distribution is actually obtained by *continuously* and *rapidly* sampling each sensor while the vehicle is moving. Thus, the same cell and its neighboring cells are repeatedly incremented, as shown in Fig. 1b. This results in a *histogramic probability distribution*, in which high CVs are obtained in cells close to the actual location of the obstacle. Our experiments show that *actual* rapid sampling from the moving robot is more accurate than methods using an *assumed* probability function (Raschke and Borenstein, 1990).

3. The Vector Field Histogram Method

The VFH method employs a *two-stage data reduction* technique, in which three levels of data representation can be distinguished.

a. The highest level holds the detailed description of the robot's environment. In this level, the two-dimensional Cartesian *Histogram Grid* $\boldsymbol{C}$ is continuously updated in real-time with range data sampled by the onboard range sensors. The *Histogram Grid* is absolute and does not change with the robot's momentary location. However, along with the vehicle moves a notional window of size $w_s \times w_s$, overlaying a square region of $\boldsymbol{C}$. We will call this region the "*active region*" (denoted $\boldsymbol{C}^*$), and cells that momentarily belong to the *active region* will be called "*active cells*" (denoted $c^*_{i,j}$). In our current implementation, the size of the *active region* is 33×33 cells. As will be discussed in Sect. 3.1, only *active cells* have immediate influence on the robot control.

b. At the intermediate level, a *Polar Histogram* $\boldsymbol{H}$ is constructed around the robot's momentary center. $\boldsymbol{H}$ comprises n angular sectors of width α (see Fig. 2). α may be chosen arbitrarily but must be such that $n=360/\alpha$ is an integer (e.g., $\alpha=5^o$ and $n=72$). Each sector k corresponds to a discrete angle quantized to multiples of α, such that $k = 0, \alpha, 2\alpha \ldots 360-\alpha$. A transformation (described in Sect. 3.1, below) maps $\mathbf{C}^*$ into $\boldsymbol{H}$ resulting in each sector k holding a value h_k which represents the *polar obstacle density* in the direction k.

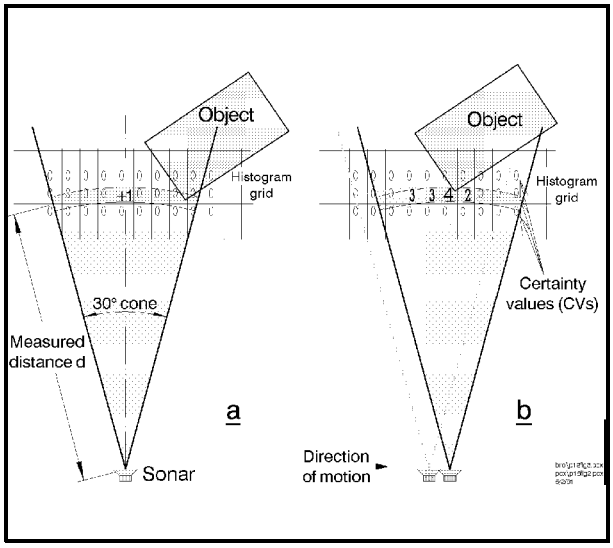

Figure 1:
a. Only one cell is incremented for each range reading.
b. Histogramic probability distribution is obtained by continuous and rapid sampling while the vehicle is moving.

c. The lowest level of data representation is the output of the VFH algorithm: the reference values for the drive and steer controllers of the vehicle.

3.1 First Data Reduction: Creation of the Polar Histogram

The first data reduction stage maps the *active region* of the *Histogram Grid* $\boldsymbol{C}^*$ into the *Polar Histogram* $\boldsymbol{H}$. For this purpose, we will now treat the certainty value of all *active cells* $c^*_{i,j}$ as an *obstacle vector*, the direction of which is determined by the direction $\beta_{i,j}$ from the cell to the *Vehicle Center Point* (VCP)[2].

[2] Note that for our symetrically shaped mobile robot, the VCP is easily defined as the geometric center of the robot. For rectangular shaped mobile robots, it is possible to chose two VCPs, e.g., each one at the center-point of the front and rear axles.

$$\beta_{i,j} = \mathrm{tg}^{-1}\left(\frac{y_i - y_0}{x_i - x_0}\right) \quad (1)$$

and the magnitude is given by

$$m_{i,j} = (c^*_{i,j})^2 \, [a - b\, d_{i,j}] \quad (2)$$

where

a,b	Positive constants.
$d_{i,j}$	Distance between *active cell* (i,j) and the VCP.
$c^*_{i,j}$	Certainty value of *active cell* (i,j).
$m_{i,j}$	Magnitude of the *obstacle vector* at cell (i,j).
x_0, y_0	Present coordinates of the VCP.
x_i, y_j	Coordinates of *active cell* (i,j).
$\beta_{i,j}$	Direction from *active cell* (i,j) to the VCP.

Note that $m_{i,j}$ is proportional to $-d$. Therefore, occupied cells produce large vector magnitudes when they are in the immediate vicinity of the robot, and smaller ones when they are further away. Specifically, a and b are chosen such that $a - bd_{max} = 0$, where $d_{max} = \sqrt{2}\,(w_s-1)/2$ is the distance between the farthest *active cell* and the VCP. This way $m_{i,j}=0$ for the farthest *active cell* and increases linearly for closer cells.

Correspondence between $c^*_{i,j}$ and sector k is established through

$$k = \alpha \cdot \mathrm{INT}(\beta/\alpha) \quad (3)$$

For each sector k, the *polar obstacle density* h_k is calculated by

$$h_k = \sum_{i,j} m_{i,j} \quad (4)$$

Fig. 2 shows graphically the mapping from $\boldsymbol{C}^*$ into $\boldsymbol{H}$. All *active cells* can be related to a sector by means of equations (1) and (3). In Fig. 2 all *active cells* related to sector k have been highlighted. Note that the sector width in Fig. 2 is $\alpha=10^o$ (not $\alpha=5^o$, as in the actual algorithm), to clarify the drawing.

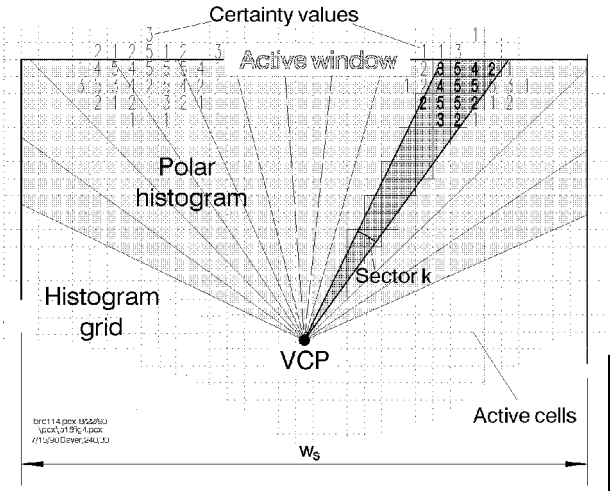

Figure 2: Mapping active cells into sectors of the *Polar Histogram*.

Notice that

a. $m_{i,j}$ in Eq. (2) is proportional to $(c^*_{i,j})^2$. This expresses our confidence that *recurring* range reading represent actual obstacles, as opposed to single occurrences of range readings, which may be caused by noise.

b. $m_{i,j}$ in Eq. (2) is proportional to $-d$. Therefore, occupied cells produce large vector magnitudes when they are in the immediate vicinity of the robot, and smaller ones when they are further away.

Because of the discrete nature of the *histogram grid*, the result of the mapping may appear ragged and cause errors in the selection of the steering direction (as explained in Section 3.2). Therefore, a smoothing function is applied to $\boldsymbol{H}$, which is defined by

$$h'_k = \frac{h_{k-l} + 2h_{k-l+1} + \ldots + lh_k + \ldots + 2h_{k+l-1} + h_{k+l}}{2l+1} \quad (5)$$

In our current implementation, l=5 yields satisfactory smoothing results.

Fig. 3a shows a typical obstacle setup in our lab. Note that the gap between obstacles B and C is only 1.2 m and that A is a thin pole 3/4" in diameter. The actual *Histogram Grid* obtained after partially traversing this obstacle course is shown in Fig. 3b. The *Polar Histogram*, corresponding to the momentary position of the robot O is shown in Fig. 3c. The directions (in degrees) in the *Polar Histogram* correspond to directions measured counterclockwise from the positive x-axis of the *Histogram Grid*. The peaks A, B, and C in the *Polar Histogram* result from obstacle clusters A, B, and C in the *Histogram Grid*.

3.2 Second Data Reduction: Computation of Steering Control

The second data reduction stage computes the required steering direction. We will call this direction Θ_{free}, and the corresponding sector in $\boldsymbol{H}$, k_{free}. This section explains how the required steering direction (in terms of k_{free}) is computed.

As can be seen in Fig. 3c, a *Polar Histogram* typically has peaks (sectors with high obstacle density), and valleys (sectors with low obstacle density). Any valley with obstacle densities below threshold is a candidate for travel. Since there are usually several candidate-valleys, the algorithm

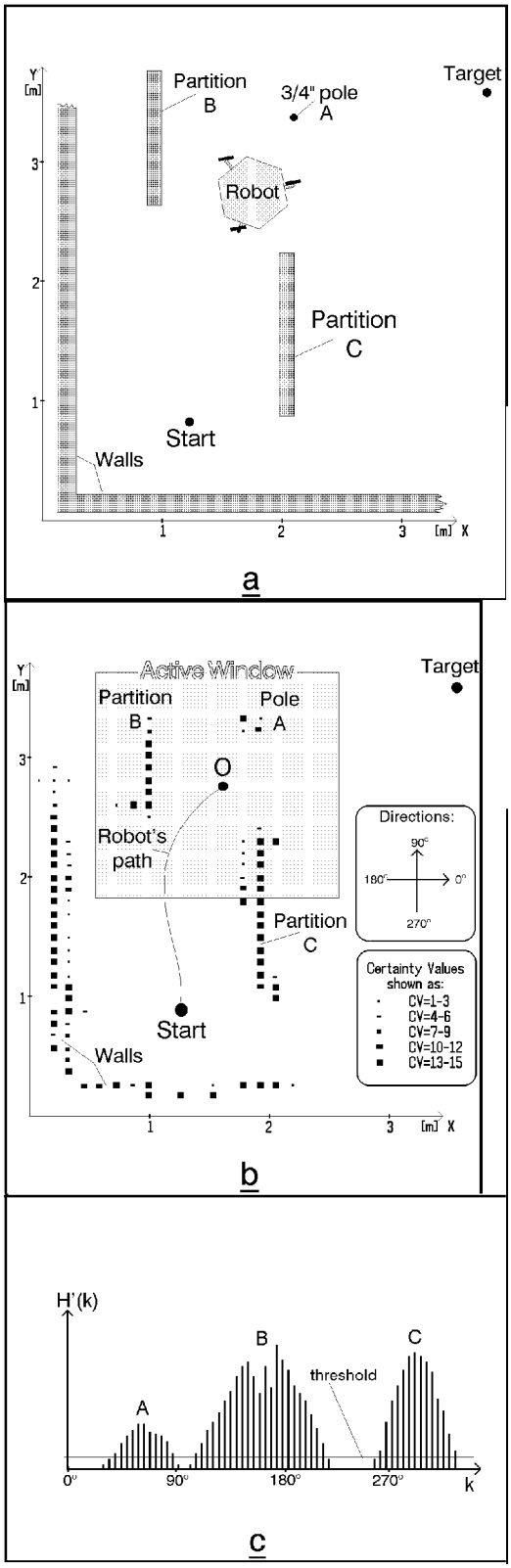

Figure 3:
a. Experimental lab setup
b. Histogram representation of experimental lab setup
c: Polar histogram of obstacles, as seen from position 'O'.

selects the one that most closely matches the direction to the target Θ_{targ} (or k_{targ}).

When the mobile robot approaches or travels between two or more closely spaced obstacles, only a very narrow valley is available for travel. In this case, k_{free} is chosen to be in the center of the valley, in order to maintain equal clearance on each side of the robot. If the selected valley is very wide (e.g., when only one obstacle is close to the robot) the algorithm chooses k_{free} several sectors "deep" into the valley, but not necessarily in its center).

3.3 Speed Control

The robot's maximum speed S_{max}, can be set at the beginning of a run. The robot tries to maintain this speed during the run unless forced by the VFH algorithm to a lower instantaneous speed S. S is determined in each sampling interval as follows:

The *smoothed polar obstacle density* in the *current* direction of travel will be denoted h_c'. $h_c'>0$ indicates that an obstacle lies ahead of the robot, requiring a reduction in speed. Large values of h_c' mean that a large obstacle lies ahead of the robot, or an obstacle is very close to the robot. Either case is likely to require a drastic change in direction and a reduction in speed is necessary to allow for the steering wheels to turn into the new direction. This reduction in speed is implemented by the following function:

$$S' = S_{max} (1 - h_c''/h_m) \quad (6)$$

where

$$h_c'' = \min(h_c', h_m) \quad (7)$$

h_m is an empirically determined constant that causes a sufficient reduction in speed.

Note that Eq. (7) guarantees $S' \geq 0$, since $h_c'' \leq h_m$.

While Eqs. (6) and (7) reduce the speed of the robot in *anticipation* of a steering maneuver, speed can be further reduced proportionally to the actual *steering rate* Ω.

$$S = S'(1-\Omega/\Omega_{max}) \quad (9)$$

where Ω_{max} is the maximal allowable steering rate for the mobile robot.

4. Experimental Results

We implemented and tested the VFH method on a commercially available mobile platform (Cybermation, 1987). This platform has a maximum travel speed S_{max} of 0.78 m/sec and weighs about 125 kg. The Cybermation platform has a unique three-wheel drive (synchro-drive) that permits omnidirectional steering.

We equipped this vehicle with a ring of 24 ultrasonic sensors (Polaroid, 1989). The sensor ring has a diameter of 0.8m, and objects must be at least 0.27m away from the sensors to be detected. Therefore, the theoretical minimum width for safe travel in a narrow corridor is $W_{min} = 0.8 + 2 \cdot 0.27 = 1.34$m.

Two computers were added to the platform: a PC-compatible single-board computer to control the sensors, and a 20Mhz, 80386-based AT-compatible that runs the VFH algorithm.

In extensive tests we run the VFH-controlled platform through difficult obstacle courses. The obstacles were unmarked, commonplace objects such as chairs, partitions, and bookshelves. In most experiments, the vehicle runs at its maximum speed (0.78 m/sec). This speed was only reduced when an obstacle was approached frontally or if required for dynamic reasons.

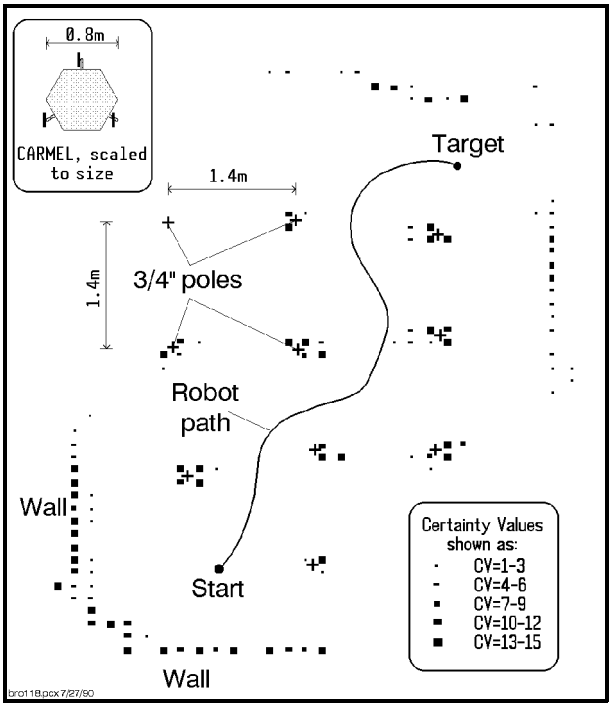

Figure 4: *Histogram Grid* representation of a test run through a field of densely spaced, thin vertical poles.

Fig. 4 shows the *Histogram Grid* after a run through a particularly challenging obstacle course that comprised 3/4" thin vertical poles spaced at a distance of about 1.4m from each other. The approximate original location of the rods is indicated with (+) symbols in Fig. 4. It should be noted that none of the obstacle locations were known to the robot in advance: the obstacle locations in Fig. 4 gradually appeared on the operator's screen while the robot was moving.

Each dot in Fig. 4 represents one cell in the *Histogram Grid*. In our current implementation, certainty values (CVs) range from 0 to 5. CV = 0 means no sensor reading has been projected into the cell during the run (i.e., no dot). CV = 1 (to 4) indicates that one (to four) readings have been projected into the cell; this is shown in Fig. 4 with dots comprising of one pixel (or more). CV = 5 means that five or more readings have been projected into the same cell, and which is represented by a 9-pixel dot in Fig. 4. The robot traversed this obstacle course at an average speed of 0.58 m/sec, without stopping in front of obstacles.

An indication for the real-time performance of the VFH algorithm is the sampling time T_s (i.e., the rate at which the steer and speed commands for the low-level controller are issued). On an Intel 80386-based PC-compatible computer running at 20Mhz, T_s = 27msec. The following steps occur during T_s:

a. Read sonar information,
b. update the *Histogram Grid*,
c. create the *Polar Histogram*,
d. determine the free sector and steering direction,
e. calculate the speed command,
f. communicate with the low-level motion controller (send speed and steer command and receive position update).

5. Conclusions

This paper presents a new obstacle-avoidance method for fast-running vehicles. This method, called the VFH method, has been developed and successfully tested on an experimental mobile robot. The VFH algorithm is computationally efficient, very robust, insensitive to misreadings, and allows continuous and fast motion of the mobile robot without stopping in front of obstacles.

The VFH method is based on the following principles:

a. A two-dimensional Cartesian *Histogram Grid* is updated continuously and in real-time with range data sampled by the onboard range sensors.

b. The data in the *Histogram Grid* is reduced to a one-dimensional *Polar Histogram* that is constructed around

the robot's momentary location. This *Polar Histogram* represents the *polar obstacle density* around the robot.

c. A sector with low obstacle density which is close to the target direction is selected, and the robot's steering is aligned with that direction.

The strength of the VFH method lies in its ability to maintain a statistical obstacle representation at both the world-model level and the intermediate data-level. Therefore, the VFH-controlled vehicle responds to clusters of high likelihood for the existence of an obstacle, while ignoring single (possibly erroneous) data points. Also, since information about narrow passages is still available at the intermediate data level (*free valleys*) the vehicle is able to navigate through narrow passages (e.g., doorways) or negotiate narrow corridors without oscillations.

7. References

Arkin, R. C., "Motor Schema-Based Mobile Robot Navigation." *The International Journal of Robotics Research*, August 1989, pp. 92-112.

Borenstein, J. and Koren, Y., 1988, "High-speed Obstacle Avoidance for Mobile Robots." *Proceedings of the IEEE Symposium on Intelligent Control*, Arlington, Virginia, August 24-26, 1988, pp. 382-384.

Borenstein, J. and Koren, Y., 1989, "Real-time Obstacle Avoidance for Fast Mobile Robots." *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 19, pp. 1179-1187.

Borenstein, J. and Koren, Y., 1990, "Critical Analysis of Potential Field Methods for Mobile Robot Obstacle Avoidance." Submitted for publication in the *IEEE Journal of Robotics and Automation*, February.

Brooks, R. A., 1986, "A Robust Layered Control System for a Mobile Robot." *IEEE Journal of Robotics and Automation*, Vol. RA-2, No. 1, pp. 14-23.

Cybermation, 1987, "K2A Mobile Platform." *Commercial Offer*, 5457 JAE Valley Road, Roanoke, Virginia 24014.

Khatib, O., 1985, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots." *1985 IEEE International Conference on Robotics and Automation*, March 25-28, St. Louis, pp. 500-505.

Koren, Y. and Borenstein, J, 1989, "Analysis of Mobile-Robot/Environment Interaction" *The University of Michigan, Technical Report No. UM-MEAM-89-1*, January 1989.

Krogh, B. H., 1984, "A Generalized Potential Field Approach to Obstacle Avoidance Control." *International Robotics Research Conference*, Bethlehem, Pennsylvania, August.

Krogh, B. H. and Thorpe, C. E., 1986, "Integrated Path Planning and Dynamic Steering Control for Autonomous Vehicles." *Proceedings of the 1986 IEEE International Conference on Robotics and Automation*, San Francisco, California, April 7-10, pp. 1664-1669.

Moravec, H. P. and Elfes, A., 1985, "High Resolution Maps from Wide Angle Sonar." *IEEE Conference on Robotics and Automation*.

Moravec, H. P., 1988, "Sensor Fusion in Certainty Grids for Mobile Robots." *AI Magazine*, Summer 1988, pp. 61-74.

POLAROID Corporation, 1989, Ultrasonic Components Group, 119 Windsor Street, Cambridge, MA, 02139.

Raschke, U. and Borenstein, J., "A Comparisson of Grid-type Map-building Techniques by Index of Performance." *1990 IEEE International Conference on Robotics and Automation*, Cincinnati, Ohio, May 13-18, 1990.

Tilove, R. B., 1990, "Local Obstacle Avoidance for Mobile Robots Based on the Method of Artificial Potentials." To be presented at the *1990 IEEE International Conference on Robotics and Automation*, Cincinnati, Ohio, May 13-18, 1990.