

Real-time map-building for fast mobile robot obstacle avoidance

Johann Borenstein and Yoram Koren

The University of Michigan, Department of Mechanical Engineering and Applied Mechanics
Advanced Technology Laboratories, 1101 Beal Avenue, Ann Arbor, MI 48109

ABSTRACT

This paper introduces HIMM (*histogramic in-motion mapping*), a new method for real-time map building with a mobile robot in motion. HIMM represents data in a two-dimensional array (called a *histogram grid*) that is updated through rapid continuous sampling of the onboard range sensors during motion. Rapid in-motion sampling results in a statistical map representation that is well-suited to modeling inaccurate and noisy range-sensor data. HIMM is integral part of an obstacle avoidance algorithm and allows the robot to immediately use the mapped information in real-time obstacle-avoidance. The benefits of this integrated approach are twofold: (1) quick, accurate mapping; and (2) safe navigation of the robot toward a given target.

HIMM has been implemented and tested on a mobile robot. Its dual functionality was demonstrated through numerous tests in which maps of unknown obstacle courses were created, while the robot simultaneously performed real-time obstacle avoidance maneuvers at speeds of up to 0.78m/sec.

1. INTRODUCTION

Map-building and obstacle avoidance are usually treated as separate, sequential functions of a mobile robot navigation system. This paper, by contrast, introduces a new *integrated* mapping and obstacle avoidance system. The map-building function, called *histogramic in-motion mapping* (HIMM), is optimized for *simultaneous* use with the obstacle avoidance function. Integration benefits both functions in unique ways:

1. Continuous and rapid in-motion sampling generates a statistical world-model, called *histogram grid*, that models the *level of evidence* for the existence of an obstacle. Collection of range-samples from *continuously varying locations* (during motion) yields more accurate maps than sampling from a few (stationary) locations does⁸.
2. The intensity of an obstacle avoidance maneuver is proportional to the *level of evidence* for the existence of an obstacle. In other words, the system reacts to weak evidence with a moderate steering maneuver, while stronger evidence causes a more drastic avoidance maneuver. This progressive-response approach renders the system insensitive to noisy or erroneous sensory data while maintaining a level of "alertness" that allows the robot to avoid suddenly appearing obstacles, even when running at high speeds.

This paper focuses on the map building aspect of our system. A comprehensive account of our *vector field histogram* (VFH) obstacle avoidance method is given in previous publications^{2,3}.

The mobile robot used in this research is CARMEL (Computer-Aided Robotics for Maintenance, Emergency, and Life support). CARMEL is based on a commercially available mobile platform with a unique three-wheel drive (synchro-drive) that permits omnidirectional steering⁴. The Cybermation platform has a maximum travel speed of $V_{max}=0.78\text{m/sec}$ and weighs about 125 kg. We equipped this vehicle with a ring of 24 ultrasonic sensors⁷. In our system, the sonars are set up to measure distances between 27-200cm, and a complete panoramic scan (readings from all 24 sensors) takes 160 msec. A special sonar-firing algorithm reduces crosstalk dramatically (a rough estimate is one erroneous reading due to crosstalk per 1500 range readings).

2. BACKGROUND

Ultrasonic sensors are used in many mobile robots because of their low price and ease of operation. Unlike visual images, ultrasonic data is easily sampled and processed. However, any system based on ultrasonic sensors must cope with their inherent limitations, namely: a) poor directionality; b) frequent misreadings due to ultrasonic noise from external sources or stray reflections from neighboring sensors (i.e., crosstalk); and c) specular reflections.

A powerful method to deal with the problems of ultrasonic sensors is the *probabilistic representation* of obstacles in a grid-type world model. One such method, called the *certainty grid*, was developed at Carnegie-Mellon University^{5,6} (CMU).

With the *certainty grid* world model, the robot's work area is represented by a two-dimensional array of square elements denoted as cells. Each cell contains a *certainty value* (CV) that indicates the measure of confidence that an obstacle exists within the cell area. Cvs are updated by a heuristic probability function that takes into account the characteristics of a given sensor. For example, ultrasonic sensors have a conical field of view. A typical ultrasonic sensor returns a radial measure of the distance to the nearest object within the cone, yet does not specify the *angular* location of the object. Thus, a distance measurement d results from an object located anywhere within the area A (see Fig. 1). However, an object located near the acoustic axis (the center of the cone) is more likely to produce an echo than an object further away from the acoustic axis¹.

3. REAL-TIME MAP BUILDING WITH HIMM

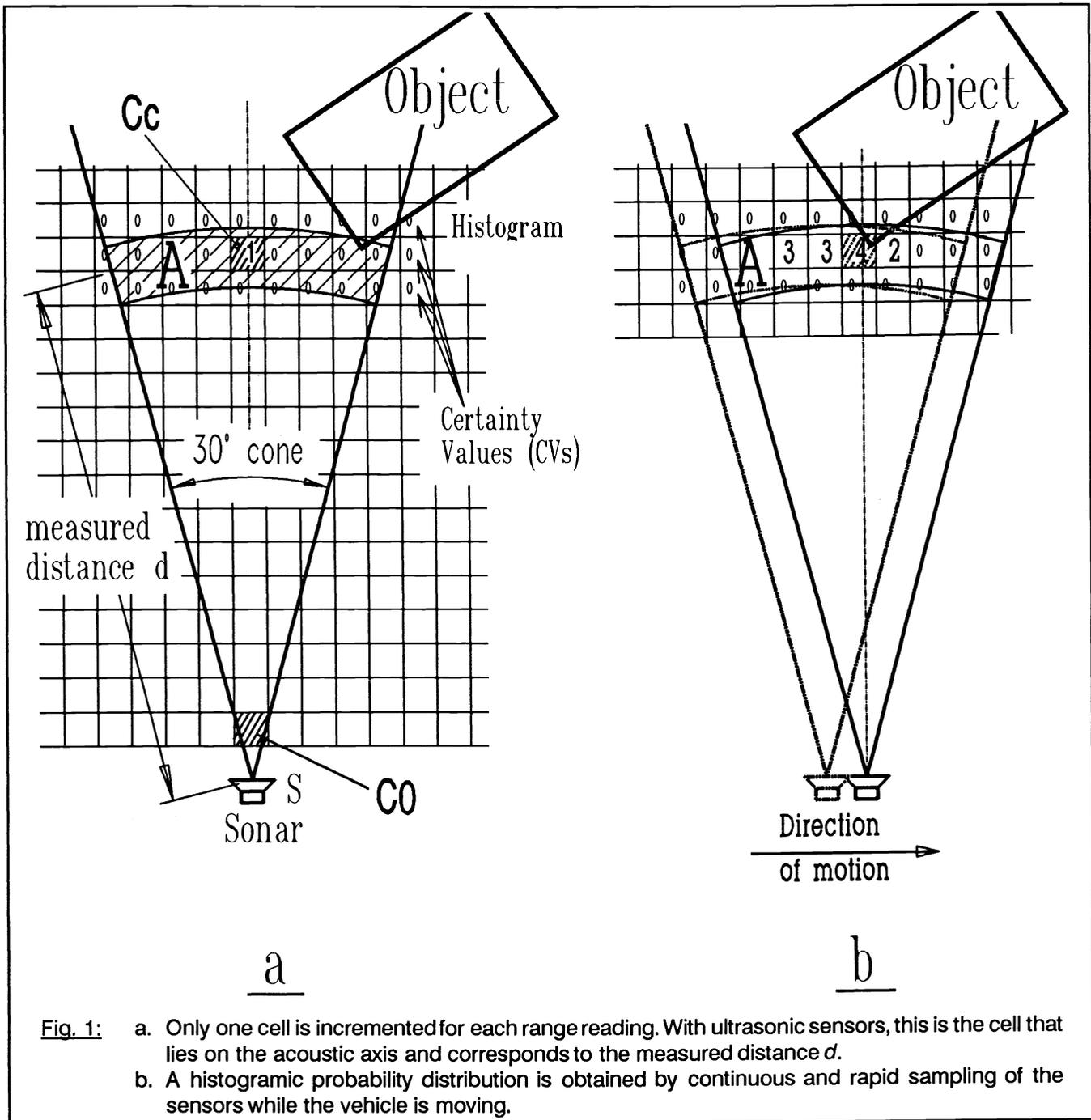
Map building requires the definition of a data structure and a method to update information in this structure. In this Section we discuss features of our data structure and the special requirements for real-time map-building.

3.1 The *histogram grid*

HIMM uses a two-dimensional Cartesian grid for obstacle representation, called the *histogram grid*. 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 all those cells affected by a range reading. This procedure is computationally intensive and would impose a heavy time-penalty on real-time execution by an onboard computer. Our method, on the other hand, increments only one cell in the *histogram grid* for each range reading, creating a "*pseudo-probability*" distribution* with only small computational overhead. 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 oversimplified, a *pseudo-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 pseudo-probability distribution*, in which high *certainty values* are obtained in cells close to the actual location of the obstacle.

In order to evaluate the accuracy of this method, we have developed an *index of performance* (IOP) for grid-type map-building⁸. This IOP is a measure of the weighted average distance between a filled cell in the grid and the closest *actual* obstacle boundary. The mathematical formulation of the IOP as well as more implementation details are given in a previous publication⁸. For comparison, we have implemented one version of the *certainty grid* method⁶. In experimental runs through numerous obstacle courses we found that HIMM consistently produced significantly more accurate maps⁸. We accredit this result to the fact that rapid, in-motion sampling produces many more useful readings than the *certainty grid* method, which assembles readings from only a few (stationary) positions. HIMM, on the other hand, shows objects from constantly changing angles and positions, since one panoramic "snapshot" (i.e., 24 readings) is assembled every 8-13cm of travel.

* We use the term *pseudo-probability* in the literal sense of "*probability*" or "*likelihood*", in contrast to the strict mathematical sense of "*probability*".



When a range measurement d is returned from an ultrasonic sensor, it is reasonable to assume (although not always true) that the sector bounded by S and A (see Fig. 1a) is free of obstacles. One may thus *decrement* cells in that area. This function is desirable, as it "clears" the area from erroneous readings. In our implementation, only those cells that are located on the line connecting center cell C_c and origin cell C_o (i.e., the acoustic axis) are decremented.

A final note concerns the actual implementation of HIMM: Whenever a cell is incremented, the increment (denoted I^+) is actually 3 (not 1, as may be expected) and the maximum CV of a cell is limited to $CV_{max}=15$. Decrements (denoted I^-), however, take place in steps of -1 and the minimum value is $CV_{min}=0$. Increments are larger than decrements because

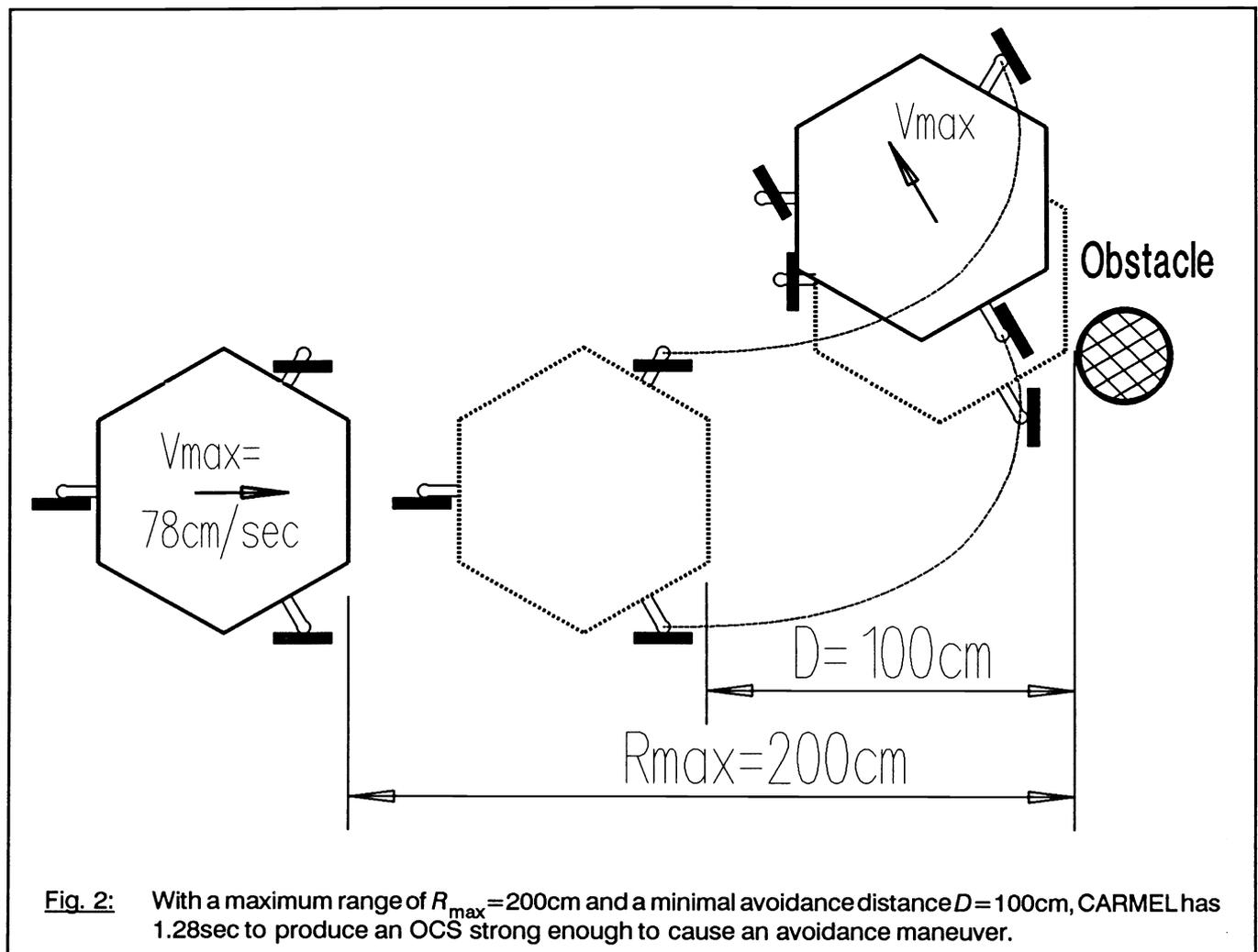
only one cell is incremented for each reading, whereas multiple cells are decremented (i.e., all cells between C_c and C_o , in Fig. 1a).

3.2 Fast mapping for real-time obstacle avoidance

In addition to producing maps, HIMM also provides instantaneous environmental information for use by the integrated obstacle avoidance algorithm. To understand how this function is supported, it is necessary to mention some characteristics of our *vector field histogram* (VFH) method for real-time obstacle avoidance.

The response of our VFH algorithm is proportional to the *square* of a CV. We will call this the *squared certainty value* (SCV). For example, if five readings have incremented a particular cell (i,j) , then $CV_{i,j}=5 \times 3=15$, and $SCV_{i,j}=(15)^2=225$. We introduce the SCV to express our confidence that recurring range readings represent actual obstacles (as opposed to single readings, which may be caused by noise or crosstalk).

Furthermore, the VFH obstacle avoidance response is stronger when a *cluster* of SCVs is encountered, whereas single, unclustered cells provoke only a mild response. For this reason, we will define the term *obstacle cluster strength* (OCS) as the sum of all SCVs in a certain *cluster* (i.e., a grouping of neighboring cells with $CV > 0$).

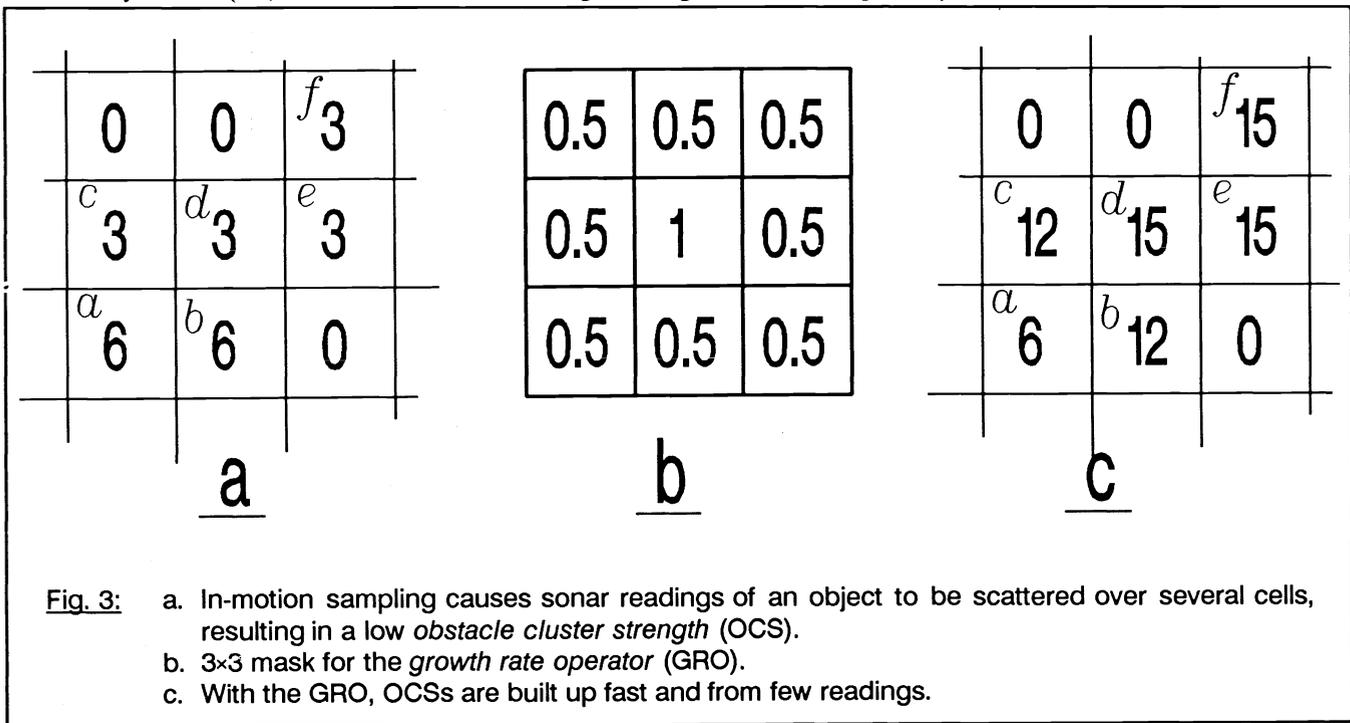


HIMM, as explained so far, works very well for slow-moving vehicles; yet when a vehicle is traveling at relatively high speeds (e.g., $V_{max} > 0.5\text{m/sec}$), matters are more complicated. The following example explains one of these problems.

Suppose CARMEL approaches a thin vertical pole while traveling at its maximum speed $V_{max} = 0.78\text{m/sec}$. To avoid a collision at this speed, CARMEL must begin an avoidance maneuver at a distance of approximately $D = 100\text{cm}$ from the obstacle, as shown in Fig. 2. The obstacle is initially detected by the robot at $R_{max} = 200\text{cm}$. Thus, the HIMM algorithm has at most $t_c = (R_{max} - D) / V_{max} = 1.28\text{sec}$ to produce an OCS strong enough to cause an avoidance maneuver. Since each sensor is sampled once every $T_p = 160\text{msec}$, the robot can sample at most $n_c = t_c / T_p = 8$ readings from the same sensor (for simplicity, we assume here that only one sensor can "see" the object, as is often the case with thin vertical poles or pipes). n_c is the *critical number of readings* needed to provoke an avoidance maneuver.

A map building algorithm for simultaneous real-time obstacle avoidance must thus build a significant OCS *quickly* and from few readings, while maintaining high contrast with erroneous readings. This task is further complicated by *in-motion sampling*, as the following example shows:

When a *stationary* ultrasonic sensor is repeatedly sampled, it will always increment the same cell for an obstacle, even if that cell does not accurately correspond to the obstacle. After n_c readings, the CV of that cell will reach the maximum value, $CV = 15$, with an $OCS = 15^2 = 225$. *In-motion sampling*, on the other hand, will usually cause the same n_c sonar readings to be scattered over several neighboring cells, even when the obstacle is a thin pole. This might result in a *cluster* such as the one shown in Fig. 3a. In this example, eight range readings were taken and projected onto the *histogram grid* in the following order: two readings - cell a; two readings - cell b; one reading each - cells c,d,e, and f. This cluster yields $OCS = 6^2 + 6^2 + 3^2 + 3^2 + 3^2 + 3^2 = 108$, which is less than the OCS that would result from the same number of readings by a stationary sensor (i.e., $OCS = 225$ since all readings are registered in a single cell).



To compensate for this adverse scattering effect (caused by *in-motion sampling*), we introduce a method to increase the OCS *growth rate*. This method uses a *growth rate operator* (GRO) to increment a cell (i,j) faster when the immediate neighbors of the cell hold high CVs. This function is implemented in real-time by convolving $CV_{i,j}$ with the 3x3 mask given in Fig. 3b, and adding the increment $I^* = 3$, yielding

$$CV'_{ij} = CV_{ij} + I^+ + \sum_{\substack{p,q=1 \\ p,q=-1}} (w_{p,q} CV_{i+p,j+q}) \quad (1)$$

with $w_{p,q}=0.5$ for $p = \pm 1$ and $q = \pm 1$, and $w_{p,q}=1$ for $p=q=0$

where

CV_{ij} previous *certainty value* of cell (i,j) ,

CV'_{ij} updated *certainty value* of cell (i,j) , after incrementing with GRO mask,

I^+ constant increment ($I^+=3$),

w weighing factor.

Unlike mask operators in computer vision algorithms (where the operator is applied to every pixel after an image has been sampled), the GRO is applied to each individual range reading as it is projected onto the *histogram grid*.

Reading	Cell	CV'
1	$a = a + I^+$	$= 0 + 3 = (3)$
2	$a = a + I^+$	$= 3 + 3 = 6$
3	$b = b + I^+ + \frac{1}{2}(a)$	$= 0 + 3 + 3 = (6)$
4	$b = b + I^+ + \frac{1}{2}(a)$	$= 6 + 3 + 3 = 12$
5	$c = c + I^+ + \frac{1}{2}(a+b)$	$= 0 + 3 + 9 = 12$
6	$d = d + I^+ + \frac{1}{2}(a+b+c)$	$= 0 + 3 + 15 = 15^*$
7	$e = e + I^+ + \frac{1}{2}(b+d)$	$= 0 + 3 + 13.5 = 15^*$
8	$f = f + I^+ + \frac{1}{2}(e+f)$	$= 0 + 3 + 15 = 15^*$

* Note that CVs are limited to $CV_{\max} = 15$;
 () denote temporary values.

Table I: Example computation of an OCS using the *growth rate operator* (GRO) in Fig. 3b.

Using Eq. (1) to update the *histogram grid* in the example of Fig. 3a (assuming cells are read and updated in the same order), we obtain the *cluster* shown in Fig. 3c. The OCS of this *cluster* is computed as shown in Table 1, resulting in an $OCS = 6^2 + 12^2 + 12^2 + 15^2 + 15^2 + 15^2 = 999$.

It should be clear that the GRO has little or no effect at all on erroneous readings. Erroneous readings (due to noise or crosstalk) appear randomly and usually affect single, unclustered cells.

4. MOVING OBJECTS

Integrated map-building and obstacle avoidance are particularly challenging with moving objects. On the one hand, the object must be noted, in order to be avoided. On the other hand, the resulting map should not show a "phantom" object that has moved on.

HIMM provides a partial solution to this problem. While HIMM does not explicitly identify a moving object as such, the changing location of the object is temporarily noted in the *histogram grid* and the robot can avoid the object. As the object

moves on, subsequent range readings detect the empty space and, accordingly, decrement the temporarily high CVs. The resulting map usually shows only a faint trace of the path of the object, which does not cause the robot to avoid a "phantom" object. Subsequent robot travel through the path of the object erases the trail completely.

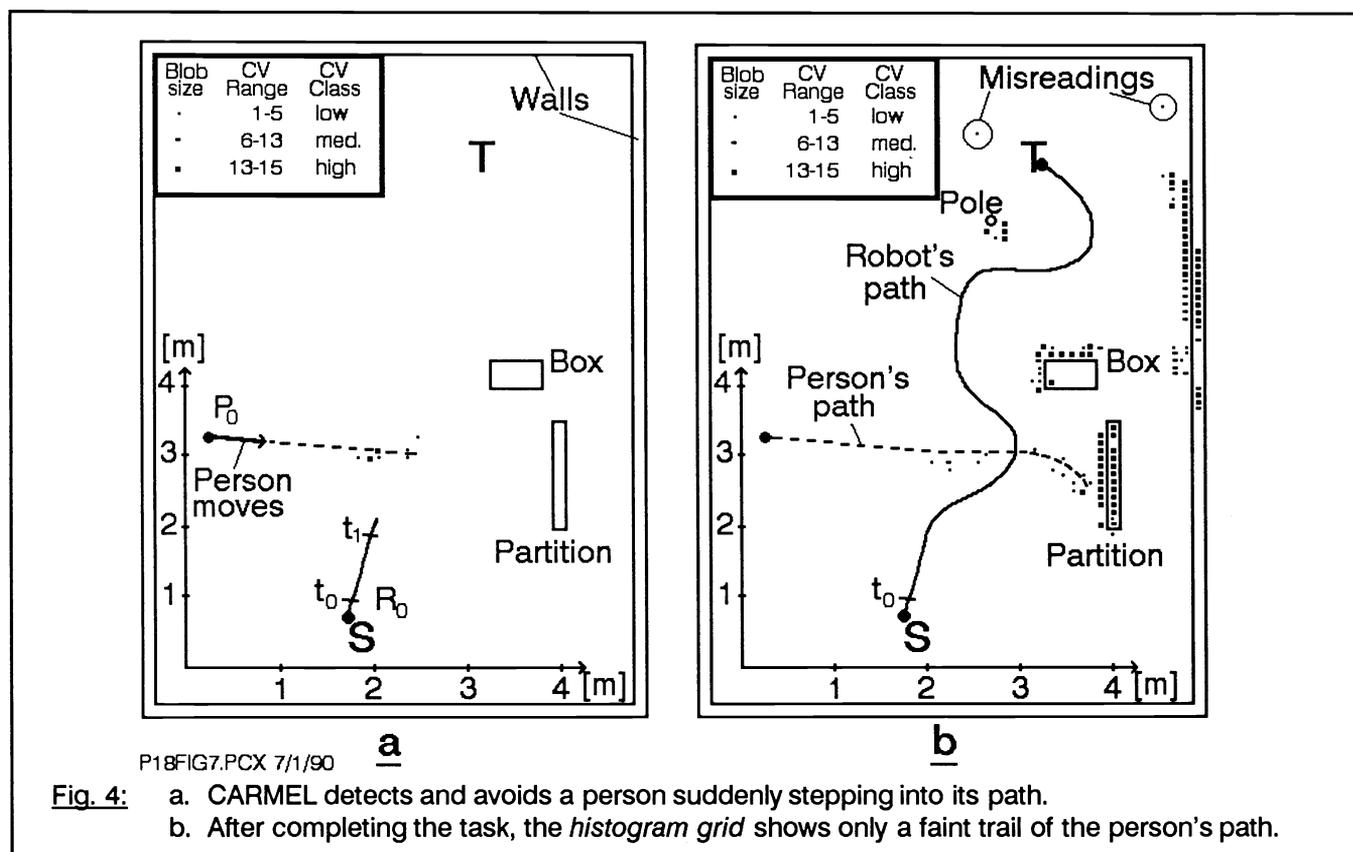


Fig. 4 shows an experiment with moving objects. Three layers of information are depicted in this figure:

1. The robot's path, starting at 'S' and ending at 'T', is plotted as the curved line. This path resulted from an actual run of our mobile robot, with real-time obstacle avoidance by the VFH method.
2. Obstacles and walls are outlined by solid lines. The objects labeled "Partition" is a two-inch-thick styrofoam sheets. The object labeled "Pole" is a 3/4 inch cardboard pipe, and the object labeled "Box" is a cardboard box.
3. The *histogram grid* is also shown in Fig. 4. Empty cells are not indicated, while small black rectangles (blobs) represent filled cells. Each cell represents a real-world square of size 10cm×10cm. On the computer screen, CVs are coded with a different color for each one of the 16 possible CVs (ranging from 0 to 15). This effect cannot be reproduced in the screen-dump of Fig. 4a, but classes of low, medium, and high CVs can be distinguished by different blob sizes, as described in Fig. 4a.

Fig. 4 shows CARMEL's path after accelerating and running at its maximum speed, 0.78m/sec. At t_0 (with CARMEL at R_0 in Fig. 4a), a person starts to walk into the robot's path. The person's starting position is P_0 and he walks in the direction indicated by the arrow. At t_1 , the OCS (due to the moving person) grows strong enough to provoke a significant obstacle avoidance response, and the robot takes a sharp turn to the right. Note the CVs representing the walking person.

During the avoidance maneuver (Fig. 4b), the robot slows down considerably, while the person continues walking at a steady pace, leaving a trail of medium-to-high CVs. This trail, however, is eroded by the decrementing function of free

space implemented in HIMM. Within one or two seconds, CVs in the trail are decremented to values small enough to allow the robot to pass through the trail, *behind* the walking person. The diagram in Fig. 4b shows the screen after the robot successfully reaches its original target T. Note how CVs along the person's trail have been reduced to low values. Application of a reasonable threshold (e.g., $CV_{\text{thres}} = 12$) would result in a permanent map that does not show any of the temporary path of the moving person.

5. CONCLUSIONS

HIMM, a new method for combined real-time map building and obstacle avoidance has been introduced and tested. In this method, inaccurate ultrasonic sensor data is *pseudo-probabilistic* modeled in a two-dimensional *histogram grid*. A statistical representation is obtained through rapid, continuous sampling of the sensors during motion. With HIMM, any range reading is immediately represented in the map and has immediate influence on the concurrent obstacle avoidance algorithm.

Further optimization, by means of the *growth rate operator* (GRO) allows the HIMM method to build high-contrast representations based on only a few range readings. This feature allows the robot to react quickly to obstacles that appear suddenly, even when traveling at high speeds.

Moving objects in the robot's path are treated as obstacles and can be avoided by the obstacle avoidance algorithm. However, CVs in the trail are decremented to values that allow the robot to pass through the trail, once the object has moved on. The resulting *permanent map* is free of traces of the moving objects and can be used for subsequent global path planning.

6. ACKNOWLEDGMENTS

This work was sponsored by the Department of Energy Grant DE-FG02-86NE37969.

7. REFERENCES

1. Borenstein, J. and Koren, Y., "Obstacle Avoidance With Ultrasonic Sensors." *IEEE Journal of Robotics and Automation*, Vol. RA-4, No. 2, 1988, pp. 213-218.
2. Borenstein, J. and Koren, Y., "Real-time Obstacle Avoidance for Fast Mobile Robots in Cluttered Environments." *1990 IEEE International Conference on Robotics and Automation*, Cincinnati, Ohio, May 13-18, 1990.
3. Borenstein, J. and Koren, Y., "The Vector Field Histogram – Fast Obstacle Avoidance for Mobile Robots." Submitted for publication in the *IEEE Journal of Robotics and Automation*, July 1989.
4. Cybermation, "K2A Mobile Platform." *Commercial Offer*, 5457 JAE Valley Road, Roanoke, Virginia 24014.
5. Elfes, A., "Using Occupancy Grids for Mobile Robot Perception and Navigation." *Computer Magazine*, June 1989, pp. 46-57.
6. Moravec, H. P. and Elfes, A., "High Resolution Maps from Wide Angle Sonar." *Proceedings of the IEEE Conference on Robotics and Automation*, Washington, D.C., 1985, pp. 116-121.
7. POLAROID Corporation, Ultrasonic Components Group, 119 Windsor Street, Cambridge, Massachusetts 02139.
8. Raschke, U. and Borenstein, J., "A Measure of Performance for Grid-type Map-building Techniques." *1990 IEEE International Conference on Robotics and Automation*, Cincinnati, Ohio, May 13-18, 1990.