HISTOGRAMIC IN-MOTION MAPPING
FOR MOBILE ROBOT OBSTACLE AVOIDANCE

by
J. Borenstein , Member IEEE and **Y. Koren** , Senior Member IEEE
Department of Mechanical Engineering and Applied Mechanics
The University of Michigan, Ann Arbor, MI 48109

ABSTRACT

This paper introduces *histogramic in-motion mapping* (HIMM), 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 in-motion sampling of onboard range sensors. Rapid in-motion sampling results in a map representation that is well-suited to modeling inaccurate and noisy range-sensor data, such as that produced by ultrasonic sensors, and requires minimal computational overhead. Fast map-building allows the robot to immediately use the mapped information in real-time obstacle-avoidance algorithms. 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 0.78 m/sec.

1. INTRODUCTION

This paper introduces a system in which *mapping* and *obstacle avoidance* are integrated. In this system, range data from ultrasonic sensors is continuously sampled and a map is built and updated immediately while the robot is traveling. Simultaneously, the obstacle avoidance algorithm uses the instantaneous mapping-information to avoid newly detected obstacles.

One advantage of this integrated system is its ability to progressively adapt the strength of an obstacle avoidance reaction 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 of the vehicle. This progressive-response approach renders the system insensitive to low *levels of evidence*, which are likely to be caused by noise or crosstalk. Without this provision, the robot would frequently perform avoidance maneuvers for non-existing obstacles. Evidence for real obstacles, however, grows quickly due to the *growth rate operator* (GRO) introduced in Section 3.2.

Map building methods depend strongly on the characteristics of the sensors that provide the raw data. Ultrasonic sensors, for example, provide good range data but offer only poor directionality (an opening angle of 30° is typical for the Polaroid [19] sensors used in our research). Another problem with ultrasonic sensors are *specular reflections* from smooth surfaces, such as indoor walls or furniture. Systems using ultrasonic sensors must also cope with frequent misreadings due to ultrasonic noise from external sources or stray reflections from neighboring sensors (i.e., crosstalk). A more detailed discussion on relevant characteristics and limitations of ultrasonic sensors can be found in the literature [1],[11],[12],[15]. Our method is designed to overcome these shortcomings of systems based on ultrasonic sensors.
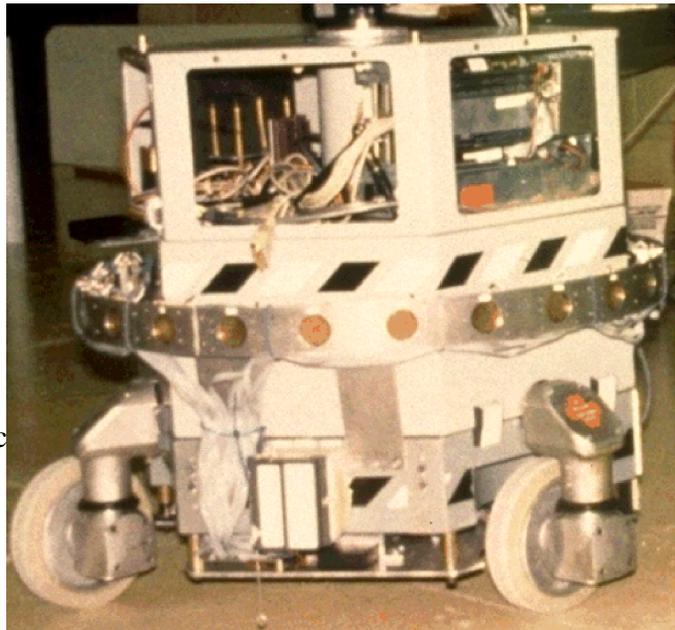

Figure 1: CARMEL, the first of the University of Michigan's mobile robots.

To verify the map building algorithm we tested it on our mobile robot CARMEL (*C*omputer-*A*ided *R*obotics for *M*aintenance, *E*mergency, and *L*ife support). CARMEL is based on a commercially available mobile platform with a unique three-wheel drive (synchro-drive) that permits omnidirectional steering [7]. We equipped this vehicle with a ring of 24 ultrasonic sensors [19], as shown in Fig. 1.

This paper focuses on the map building aspect of our system, rather than on the obstacle avoidance algorithms. A comprehensive discussion of our two obstacle avoidance methods, the *Virtual Force Field*(VFF) and the *Vector Field Histogram*(VFH) method, is given in [2],[3], and in [4],[5], respectively. Section 2 evaluates related work in map building, and Section 3 explains our real-time map building method in detail.

2. MAP BUILDING WITH ULTRASONIC SENSORS

In order to create a map from ultrasonic range measurements, the environment must first be scanned. To do so, many mobile robots are equipped with 24 sensors that are mounted on a horizontal ring around the robot [2], [6], [8], [14], [17]. *Ring scanning* does not require rotating parts and motors, and a full 360° panorama can be acquired rapidly. However, all sensors cannot be fired at once, since this would cause significant crosstalk. Special scanning sequences can be designed that reduce crosstalk, but increase the overall time needed to obtain a full panorama (i.e., all 24 sensors). Typical scan times range from 100 to 500 ms, for a full panorama.

A pioneering method for probabilistic representation of obstacles in a grid-type world model has been developed at Carnegie-Mellon University (CMU) [9],[16],[17]. The resulting world model, called a *certainty grid*, is especially suited to the unified representation of data from different sensors such as ultrasonic, vision, and proximity sensors [17], as well as the accommodation of inaccurate sensor data such as range measurements from ultrasonic sensors. 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 [19] 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. 2). 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 axis [1]. Consequently, with the CMU method, CVs are assigned to all cells in A, but higher values are assigned to cells closer to the acoustic axis, according to a heuristic probability function.

Additional information can be derived from a range reading concerning the sector between S and A (see Fig. 2). If an echo is received from an object at distance d, then this sector must be free of objects. In [9] and [16] this is expressed by applying a probability function with negative values to the cells in the empty area.

In CMU's original *certainty grid* method [9],[16],[17], the mobile robot remains *stationary* while taking a panoramic scan with its ring of 24 ultrasonic sensors. The *certainty grid* is

updated with a probabilistic function C_x that is applied to each one of the 24 range readings. Then, the robot moves to a new location, stops, and the procedure repeats. After the robot traverses a room in this manner, the resulting *certainty grid* represents a fairly accurate map of the room. A global path-planning method is employed iteratively as the map is incrementally updated.

More recent work by Elfes [10] introduces the *occupancy grid*, which is similar to the *certainty grid*. However, the *occupancy grid* models the sonar sensors with *Gaussian* uncertainty and applies a more rigorous mathematical model to recurring readings.
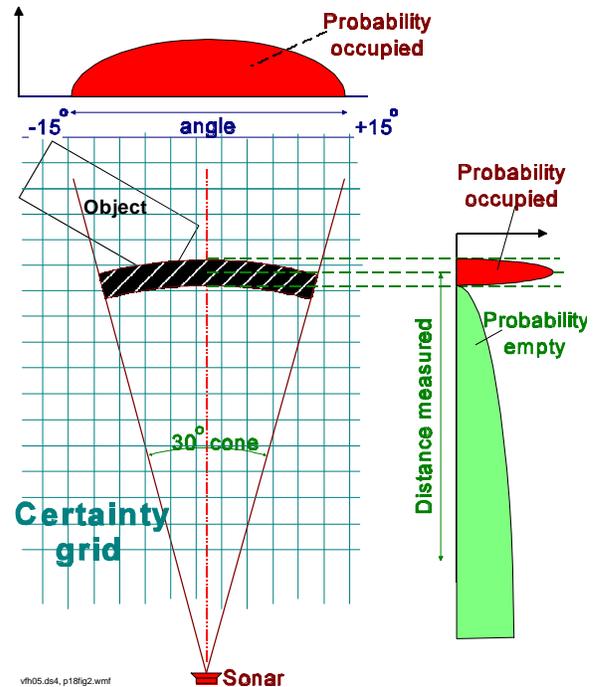

Figure 2: Two-dimensional projection of the conical field of view of an ultrasonic sensor. A range reading *d* indicates the existence of an object somewhere within the shaded region A (Carnegie Mellon's model).

3. REAL-TIME MAP BUILDING WITH HIMM

This section introduces *histogramic in-motion mapping* (HIMM), a new method for real-time map building with a mobile robot in motion. HIMM has been developed and tested on our mobile robot CARMEL. CARMEL's 24 ultrasonic sensors are divided into six groups; each with four perpendicular sensors. The six groups are fired in sequence, while the four sensors of each group are triggered simultaneously. When the six groups have been triggered, a complete 360° panorama is obtained. Our sonars are set up to measure distances between R_{min}=0.27m to R_{max}=2m. Therefore, each sensor awaits its echo for $t=2 \cdot R_{max}/V_{sound}$=11.7ms, where V_{sound}=340m/sec is the speed of sound in air, and the factor 2 accounts for the *roundtrip* of the soundwaves. A complete panoramic scan (readings from all 24 sensors) should therefore take 6t=70.2ms. However, the sensor controller computer runs a special signal enhancement algorithm that reduces crosstalk significantly (approximately by a factor of 20, resulting in roughly 1 false reading per 1500 range readings). This improvement takes a toll on the processing time, so that T_s=160ms is required to process all 24 readings. It should be noted that range readings are processed and asynchronously communicated to the map building computer in groups of 4. This way, any range reading is presented to the map building algorithm at most 160/6=27ms after it has been taken.

3.1 The Histogram Grid

HIMM uses a two-dimensional Cartesian *histogram grid* for obstacle representation. This representation has been derived from the *certainty grid* concept described in Section 2. 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 (i.e., all cells in the area A of Fig. 2). This procedure is computationally intensive and might impose a 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. For ultrasonic sensors, the incremented cell is the one that corresponds to the measured distance d (see Fig. 3a) and lies on the acoustic axis of the sensor. While this approach may seem to be an oversimplification, a probability[1] 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. 3b. This results in a *histogramic probability distribution*, in which high certainty values are obtained in cells close to the actual location of the obstacle. Note that the HIMM method is less accurate when the robot is stationary. A comparative evaluation of the accuracy of our method is given in [20].

HIMM makes use of the "empty sector" between S and A (see Fig. 4), as does CMU's *certainty grid* method [9],[16]. However, instead of computing and projecting a negative probability function for all cells in the sector, we take advantage of our fast sampling approach and decrement only those cells that are located on the line connecting center cell C_c and origin cell C_o (i.e., the acoustic axis, in Fig. 4).

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 $CV_{min}=0$. Note that CV_{max} and CV_{min} have been chosen arbitrarily. I^+ was determined experimentally (in relation to CV_{max}), by observing that too large a value would make the robot react to single, possibly false readings, while a smaller value would not build up CVs in time for an avoidance maneuver. I^- was determined experimentally and in relation to I^+. I^- must be smaller than I^+ because only one cell is incremented for each reading, whereas *multiple* cells might be decremented for one reading (i.e., all cells between C_c and C_o, in Fig. 4). Note that it is exceedingly difficult to formulate a mathematically rigorous relationship between those parameters. The reason is the large number of unknowns that affect a general formulation. For example, an object may be "seen" at a certain instance by one or more sensors simultaneously. Whether or not an echo is received depends on the relative angles, the surface structure, the reflectiveness of the object, and the distance to the

---

[1] We use this term in the literal sense of "*likelihood.*"

object.

HIMM is part of a real-time obstacle avoidance system; it not only produces maps but 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.
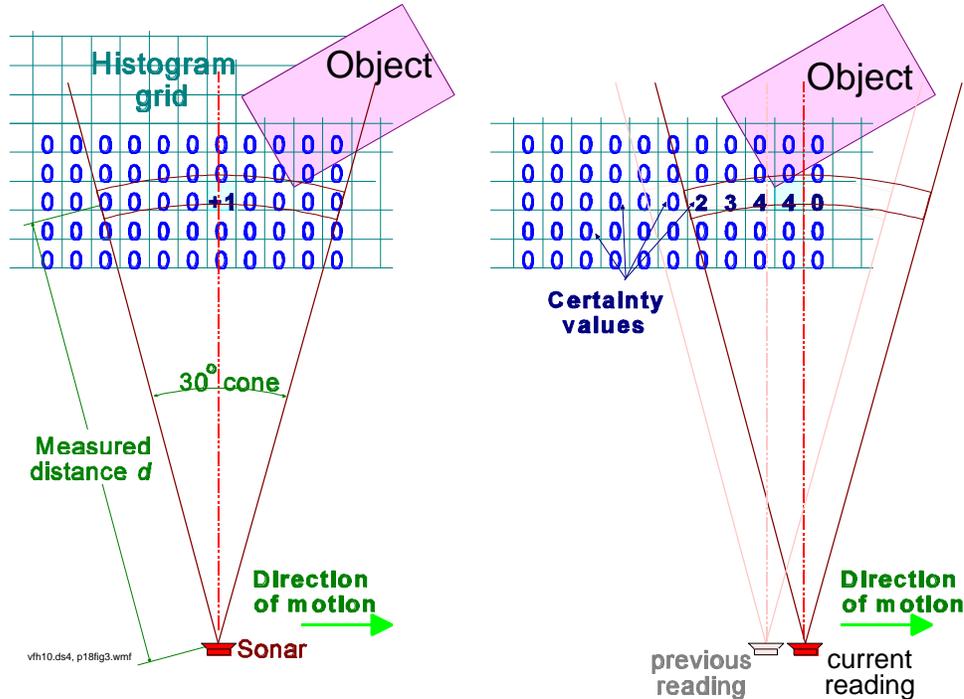

Figure 3:
a. Only one cell is incremented for each range reading. With ultrasonic sensors, this is the cell that lies on the acoustic axis and corresponds to the measured distance *d*.
b. A *histogramic probability distribution* is obtained by continuous and rapid sampling of the sensors while the vehicle is moving.

A detailed discussion of the VFH method is given in [4] and [5].

To increase the signal-to-noise ratio, the obstacle avoidance response of the 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 $CV_{i,j}=5 \cdot 3=15$, 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).

3.2 Fast Mapping for Real-time Obstacle Avoidance

HIMM, as explained so far, serves two functions: a) it produces high CVs for cells that correspond to obstacles, and b) it keeps low CVs for cells that were incremented due to mis-readings (e.g., noise or crosstalk) or moving objects. For slow-moving vehicles, this system works very well; yet when a vehicle is traveling at relatively high speeds (e.g., $V>0.5$m/sec), matters are more complicated. The following example explains one of these problems. Note

that the numerical values in this example correspond to actual specifications of our mobile robot CARMEL. Other systems will have different values, but the principle will be the same.

Suppose CARMEL approaches a thin vertical pole while traveling at its maximum speed V_{max}=0.78 m/sec. To avoid a collision at this speed, CARMEL must begin an avoidance maneuver at a distance of approximately D=100 cm from the obstacle, as shown in Fig. 5. The obstacle is initially detected by the robot at R_{max}=200 cm. Thus, the HIMM algorithm has at most t_c=(R_{max}-D)/V_{max}=1.28 sec to produce an OCS strong enough to cause an avoidance maneuver. Since each sensor is fired once every T_p=160 msec, the robot can sample at most n_c=t_c/T_p=8 readings from the same sensor[2]. Note that n_c is the *critical number of readings* needed to provoke an avoidance maneuver.

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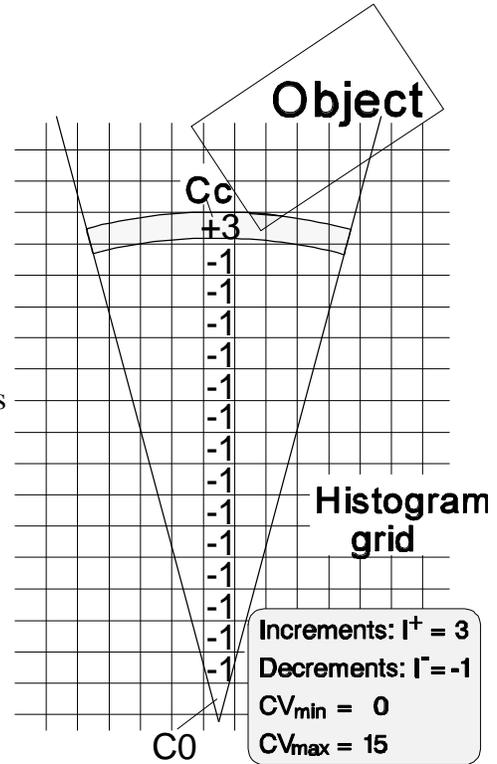
Object

Cc
+3
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1

Histogram grid

Increments: I^+ = 3
Decrements: I^- = -1
CV_{min} = 0
CV_{max} = 15

C0

p18l4bw.drw, p18l4bw.pcx, 3/11/91

Figure 4: Information on the empty sector between *S* and *A* is used to decrement all cells along line $C_0 \rangle C_c$.

When a *stationary* ultrasonic sensor is repeatedly sampled, it will usually increment the same cell for an obstacle, even if that cell does not accurately correspond to the obstacle. Assuming the robot was able to take n_c readings, the CV of that cell will reach the maximum value, CV=15, with an OCS = 15^2 = 225. The resulting *cluster* comprises of one cell only. Actually, only 5 readings were needed to reach CV=15; the remaining 3 readings are "lost" because of the limit CV_{max}. *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. 6a. 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_{scat}= $6^2+6^2+3^2+3^2+3^2+3^2$ = 108, which is less than the OCS that would result from the same number of readings by a stationary sensor (225). In general, if n readings (with $n < CV_{max}/I^+$) are scattered equally over neighboring cells, they will result in $OCS_{scat}=n(I^+)^2$ which is smaller then $OCS_{single}=(nI^+)^2$, which results when all n_c

---

[2] For simplicity, we assume here that only one sensor can "see" the object, as is often the case with thin vertical poles or pipes.

readings are projected into one cell. As was mentioned in Section 3.2, it is imperative for fast, real-time obstacle avoidance that a high OCS is built quickly, in order to cause a strong avoidance maneuver in time.

To compensate for the adverse scattering effect caused by *in-motion* sampling, we introduce a method to significantly increase the *growth rate* of an OCS (when readings are scattered in neighboring cells). 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 3×3 mask given in Fig. 6b, and adding the usual increment $I^+=3$, yielding

$T_p = 160\,\text{msec}$

$t_c = \dfrac{R_{max} - D}{V_{max}} = 1.25\,\text{sec}$

$n_c = \dfrac{t_c}{T_p} = 8$ (n_c = critical number of readings)

Avoidance path

V_{max}

Obstacle

$V_{max} = 80\,\text{cm/sec}$

$D = 100\,\text{cm}$

$R_{max} = 200\,\text{cm}$

p18fig5.pcx
10/20/90

Figure 5: With a maximum range of R_{max}=200cm and a minimal avoidance distance D=100 cm, CARMEL has 1.28 sec to produce an *obstacle cluster strength* (OCS) strong enough to cause an avoidance maneuver.

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

Certainty values

a

Object			
0	0	3	0
3	3	3	0
6	6	0	0
0	0	0	0

b

0.5	0.5	0.5
0.5	1	0.5
0.5	0.5	0.5

GRO Mask

c

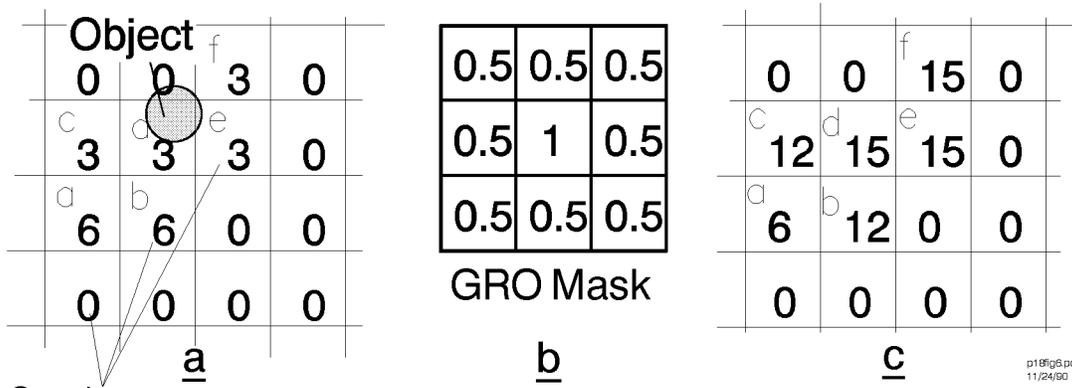| 0	0	15	0
12	15	15	0
6	12	0	0
0	0	0	0

p18fig6.pcx
11/24/90

Figure 6: a. In-motion sampling causes sonar readings of an object to be scattered over several cells, resulting in a low OCS.
b. 3×3 mask for *growth rate operator* (GRO).
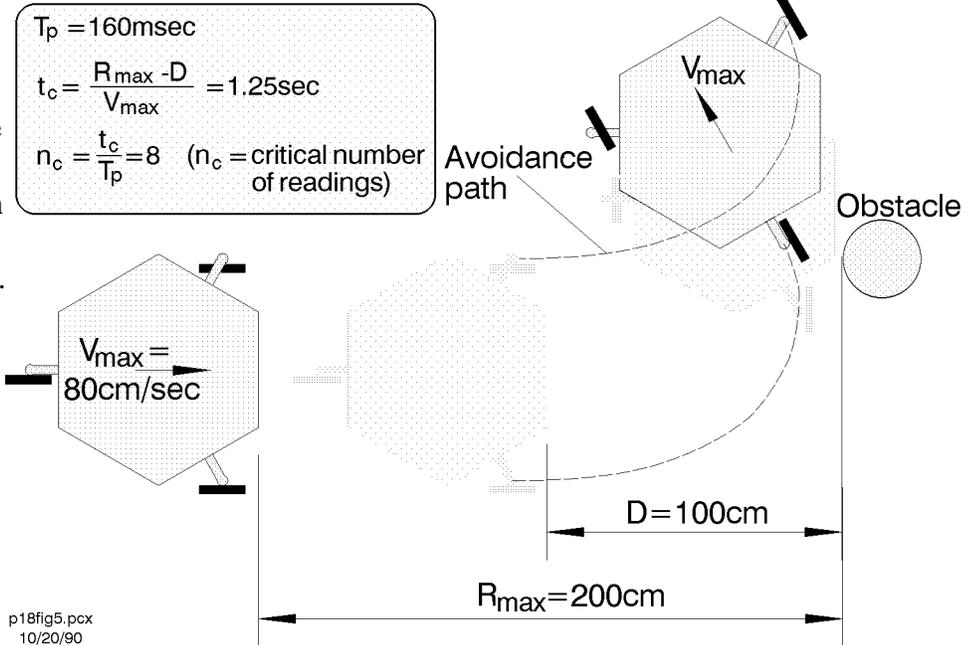c. With the GRO, OCSs are built up fast and from few readings.

where
$CV_{i,j}$ previous certainty value of cell (i,j),
$CV'_{i,j}$ updated certainty value of cell (i,j),
I^+ constant increment ($I^+=3$),
w weighing factor.

with $w_{p,q}=0.5$ for $p=\pm1$ and $q=\pm1$, and $w_{p,q}=1$ for $p=q=0$

Unlike mask operators in computer vision algorithms, where the operator is applied to every pixel *after* an image has been sampled, the GRO (by means of Eq. 1) is applied to each range reading as it is projected onto the *histogram grid.*

The following numeric example demonstrates the function of the GRO. Using Eq. (1) to re-construct the *histogram grid* in the previous example of Fig. 6a (assuming cells are read and updated in the same order), we obtain the *cluster* shown in Fig 6c. The individual steps of this computation are listed in Table I, and the OCS of this *cluster* is $OCS_{GRO}=6^2+12^2+12^2+15^2+15^2+15^2=999$.

Table I: Example computation of an OCS using the *growth rate operator* (GRO) in Fig. 6b.

```
 Reading    Cell                                       CV'
)))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
  1       a = a + I⁺                   = 0 + 3              = (3)
  2       a = a + I⁺                   = 3 + 3              =  6
  3       b = b + I⁺ + ½(a)            = 0 + 3 +   3        = (6)
  4       b = b + I⁺ + ½(a)            = 6 + 3 +   3        = 12
  5       c = c + I⁺ + ½(a+b)          = 0 + 3 +   9        = 12
  6       d = d + I⁺ + ½(a+b+c)        = 0 + 3 + 15         = 15*
  7       e = e + I⁺ + ½(b+d)          = 0 + 3 + 13.5       = 15*
  8       f = f + I⁺ + ½(e+f)          = 0 + 3 + 15         = 15*
)))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
 * Note that CVs are limited to CV max = 15;
   ( ) denote temporary values.
```

It should be clear that the GRO has little or no effect on erroneous readings. Erroneous readings (due to noise or crosstalk) appear randomly and usually effect single, unclustered cells, and are filtered out by the algorithm discussed above.

The only disadvantage of the GRO is the following: Without the GRO, high CVs are usually obtained at the actual location of an object, while lower CVs (due to the inaccuracies of the sensors) are scattered around the borders of the object. With the GRO, however, low-certainty areas adjacent to high-certainty areas build up to high CVs, resulting in a tendency to represent obstacles larger then they really are. This distortion, however, is not very significant; furthermore, we can fine-tune the trade-off between more accurate maps and

faster OCS build-up by adjusting the weighing factor w. Reducing w reduces the effect of the GRO, and completely cancels the effect with $w=0$. It is also possible to set w adaptively to the instantaneous speed of the mobile robot.

4. Experimental Results

The map building algorithms were tested on CARMEL. This platform has a maximum travel speed of $V_{max}=0.78$m/sec and a maximum steering rate of $\Omega=120$deg/sec; it weighs about 125kg. A Z-80 on-board computer serves as the low-level controller of the vehicle. Two computers were added to the platform: a 20Mhz, 80386-based AT-compatible that runs the integrated obstacle-avoidance/map-building algorithm, and a PC-compatible single-board computer to control the sensors.

Fig. 7 shows the result of an experimental run. Three layers of information are depicted in this figure:

a.   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. CARMEL's average speed in this run was 0.54m/sec and the maximum speed was 0.78m/sec.

b.   Obstacles and walls are shown as solid lines. Objects labeled "partitions" are two-inch-thick styrofoam sheets with smooth surfaces. The object labeled "pole" is a 3/4 inch cardboard pipe, and the object labeled "box" is a cardboard box. In general, poles do not cause specular reflections and are thus easy to detect from all directions. The pole in Fig. 7, however, is very slender and produces only a very weak echo, much weaker than a flat wall or a pole of larger diameter.

c.   The *histogram grid* is also reproduced in Fig. 7. Empty cells are not shown, while filled cells are represented by small black rectangles (blobs). Each cell represents a real-world square of size 10cm×10cm. While the certainty values in our system may range from 0 to 15, the screen-dump of Fig. 7a can only show classes of low, medium, and high CVs, distinguished by different blob sizes. Note that the left side of partition <u>a</u> is outside of the range limit of the ultrasonic sensors, and so are most of the walls.

In order to convert the *histogram grid* into a permanent map, we define an arbitrary threshold, e.g., $CV_{thres}=12$. Any $CV<CV_{thres}$ is rejected (i.e., set to zero). This conversion results in the final grid-type map shown in Fig. 7b for the *histogram grid* in Fig. 7a.

During a typical run through a densely cluttered obstacle course, CARMEL's speed may vary considerably: from 0.8 m/sec to sometimes as low as 0.04 m/sec [2]. In a given time interval, more range readings are taken in a certain area when the robot travels at low speeds,

adding undue weight to these readings. However, this undesirable effect is constrained by the upper bound of $CV_{max}=15$, which limits the CV for any given cell. On the other hand, due to the GRO, cells reach CV_{max} quickly and with only a few readings even when the robot is traveling at high speeds.
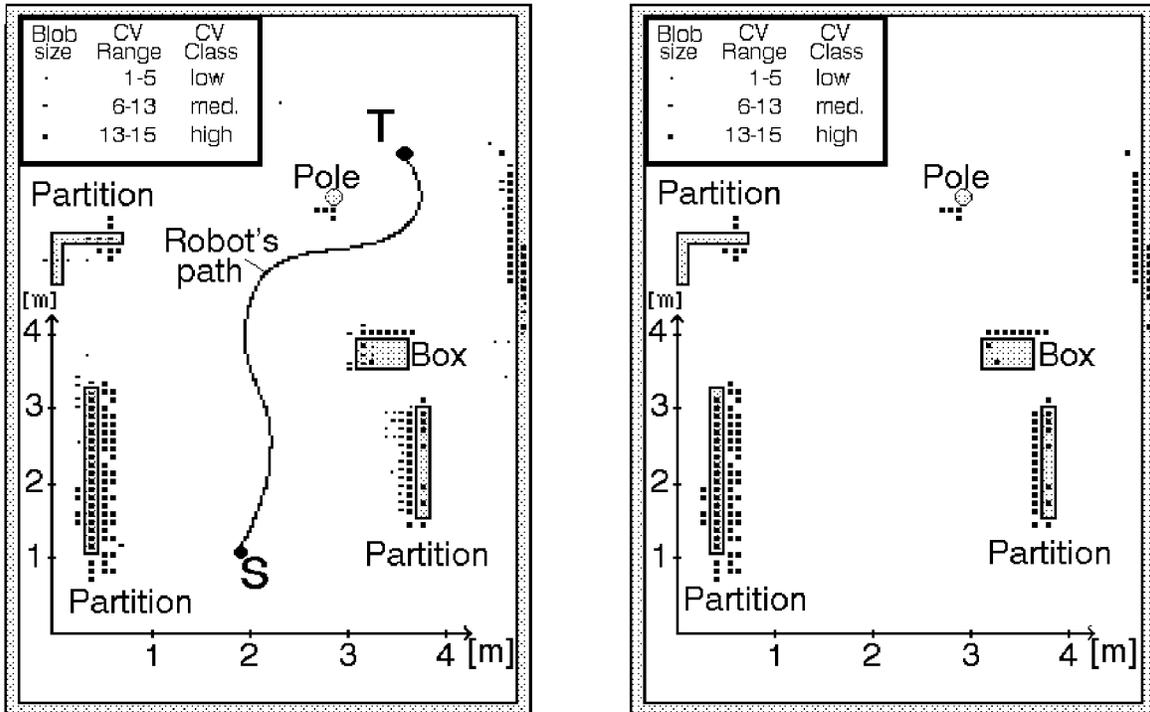

Figure 7: a.   A real-time run with CARMEL, showing the robot's path, the actual location of the unexpected obstacles, and the resulting histogram grid.
b.   The histogram grid after thresholding with $CV_{thres}=12$.

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 statistically modeled in a two-dimensional *histogram grid*. A *histogramic probability* 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 is essential for the robot to react quickly to unexpected obstacles, even when traveling at high speeds.

Acknowledgements

6. 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." *IEEE Transactions on Systems, Man, and Cybernetics,* Vol. 19, No. 5, Sept/Oct 1989, pp. 1179-1187.

3. Borenstein, J. and Koren, Y., "Tele-autonomous Guidance for Mobile Robots."*IEEE Transactions on Systems, Man, and Cybernetics,* special issue on unmanned systems and vehicles, December 1990, pp. 1437-1443.

4. 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, pp. 572-577.

5. Borenstein, J. and Koren, Y., "The Vector Field Histogram) Fast Obstacle Avoidance for Mobile Robots." *IEEE Journal of Robotics and Automation,* Vol. 7, No. 3., June 1991, pp. 278-288..

6. Crowley, J. L., "World Modeling and Position Estimation for a Mobile Robot Using Ultrasonic Ranging." *Proceedings of the 1989 IEEE International Conference on Robotics and Automation,* Scottsdale, Arizona, May 14-19, 1989, pp. 674-680.

7. Cybermation, "K2A Mobile Platform." *Commercial Offer,* 5457 JAE Valley Road, Roanoke, Virginia 24014, 1989.

8. Denning Mobile Robotics, Inc., "Securing the Future."*Commercial Offer,* 21 Cummings Park, Woburn, MA 01801, 1985.

9. Elfes, A., "Sonar-based Real-World Mapping and Navigation."*IEEE Journal of Robotics and Automation,* Vol. RA-3, No 3, 1987, pp. 249-265.

10. Elfes, A, "Using Occupancy Grids for Mobile Robot Perception and Navigation." *Computer Magazine,* June 1989, pp. 46-57.

11. Everett, H. R., "A Multielement Ultrasonic Ranging Array." *Robotics Age*, July 1985, pp. 16-20.

12. Flynn, A. M., "Combining Sonar and Infrared Sensors for Mobile Robot Navigation." *The International Journal of Robotics Research*, Vol. 7, No. 6, December 1988, pp. 5-14.

13. Kadonoff, M. B., et al., "Arbitration of Multiple Control Strategies for Mobile Robots." *Proceedings of the SPIE, Mobile Robots (1986)*, Vol. 727, 1986, pp. 90-98.

14. Korba, L. W., Liscano, R., and Durie, N., "An Intelligent Mobile Platform for Health Care Applications." *Proceedings of the ICAART 88 conference*, Montreal, Canada, 1988, pp. 462-463.

15. Kuc, R. and Barshan, B., "Navigating Vehicles Through an Unstructured Environment With Sonar." *Proceedings of the 1989 IEEE International Conference on Robotics and Automation*, Scottsdale, Arizona, May 14-19, 1989, pp. 1422-1426.

16. 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.

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

18. Pin, F. G. et al., "Autonomous Mobile Robot Research Using the Hermies-III Robot." *IROS International Conference on Intelligent Robot and Systems*, Tsukuba, Japan, Sept. 1989.

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

20. Raschke, U. and Borenstein, J., "A Measure of Performance for Grid-type Map-building Techniques," to be presented at the *1990 IEEE International Conference on Robotics and Automation*, Cincinnati, Ohio, May 13-18, 1990, pp. 1828-1832.

List of Footnotes

[1] We use this term in the literal sense of "*likelihood.*"

[2] For simplicity, we assume here that only one sensor can "see" the object, as is often the case with thin vertical poles or pipes.