EVENT: Fourth World Conference on Robotics Research **RETURN TO SME**

AUTHOR'S FACT SHEET
THIS FORM MUST ACCOMPANY YOUR MANUSCRIPT

(Please print or type)

TITLE OF PAPER:	Real-time Obstacle Avoidance for Non-Point Mobile Robots
AUTHOR(S') NAME(S):	**Johann Borenstein** and **Ulrich Raschke**
JOB TITLE:	(Borenstein:) Research Scientist, (Raschke:) Graduate Student
COMPANY:	The University of Michigan, Ann Arbor
ADDRESS:	Advanced Technology Lab, 1101 Beal Avenue
CITY:	Ann Arbor STATE/ PROVINCE: Michigan
ZIP:	48109 COUNTRY: USA
PHONE NUMBER OFFICE:	(313) 763-1560 PHONE NUMBER (HOME): 971-0609

ABSTRACT:
(100 words or less)

This paper presents a method for real-time obstacle avoidance for *non-point* mobile robots. With this method, large, oddly shaped (e.g., disproportionally long or wide vehicles), or mobile robots carrying a overhanging payload can safely and reliably navigate in the close proximity of obstacles and in confined space, while using inaccurate range sensors for real-time obstacle detection.

This method is based on the combination of two obstacle avoidance methods developed previously at the University of Michigan. The previous methods are the *virtual force field* (VFF) method, which is based on potential fields, and the *vector field histogram* (VFH) method, which is based on spatial interpretation of sensory data. The combined method is called the *combined vector field* (CVF). With this method, the *principal steering direction* is determined by the VFH algorithm, while the VFF algorithm applies virtual forces as a *corrective measure*, to account for the vehicle's dimensions. CVF allows safe navigation for *arbitrarily* shaped vehicles and can be set up for different kinematic options (e.g., rectangular vehicles with or without change of orientation).

INDEX TERMS: (refer to page 15 in the Author's Guide)

Robotics, Robots, Mobile Robots, Potential Fields, Obstacle Avoidance

RETURN TO: Society of Manufacturing Engineers

1. Introduction

Most commercially available mobile robots used in research labs have a circular, hexagonal, or polygonal footprint that can be circumscribed by a circle of 60-80 cm in diameter. Common to these robots is the convenient feature that they can be modeled as a point in the plane. To do so, some algorithms artificially enlarge the contour of obstacles, by an amount equal or slightly larger than the robot's radius. This method, called the *configuration space* approach, has been exhaustively described in the literature [Lozano-Perez, 1981].

Other methods simply provide a sufficiently large "safety distance" between the traveling "point-size" robot and the obstacles. Most importantly, the orientation of the robot has no influence on the trajectory, since at any orientation the robot is fully contained in its circumscribing circle.

Safe obstacle avoidance for *non-point* mobile robots is more difficult to accomplish. In this paper, we use the term "*non-point*" to indicate that the robot's shape cannot be approximated by a circle and that the robot's orientation must be taken into account. The algorithm described here is applicable to vehicles of arbitrary shape, but will be explained in terms of a rectangular-shape vehicle.

Some recent work has addressed this issue as a problem of *global path planning* (i.e., where an optimal or near optimal path is sought). One approach taken by several independent researchers [Barraquand and Latombe, 1989; Hague et al., 1989; Lin and Chang, 1990] first constructs a *Voronoi diagram* and then applies potential fields to multiple *act-on* locations on the *non-point* robot. While all three of the above works introduce brilliant new ideas there are some important differences to our work, as listed below.

a. Our method provides *local obstacle avoidance*) it does not solve a global path planning problem. Consequently, the robot is not guaranteed to reach the target on a near-optimal path. In fact, the robot may not even reach the target at all, if it gets trapped locally.

b. On the other hand, our method works in real-time, during motion at high speed (typically, 0.5-0.8 m/sec) and with actual run-time data gathered by onboard sensors. Furthermore, we have extensively tested and verified our method on an actual mobile robot. These are some of the important difference between most simulation-based work and ours:

   - Complete knowledge about the obstacles is assumed in the beginning of the planning process;
   - Computation times are too long for real-time application.
   - The intricate dynamic interaction between robot motion and *gradual* accumulation of sensory data during motion are not addressed.

During the past few years, *potential field methods* (PFMs) for obstacle avoidance have gained increased attention from researchers in the field of robots and mobile robots [Tilove, 1990]. The idea of imaginary forces acting on a robot has been suggested by Andrews and Hogan [1983] and Khatib [1985]. In these approaches obstacles exert repulsive forces onto the robot, while the target applies an attractive force. The sum of all forces, the resultant force $\boldsymbol{R}$, determines the subsequent direction and speed of travel.

In our own previous research we have developed a PFM, called the *virtual force field* (VFF) method [Borenstein and Koren, 1989]. The VFF method works in real-time and with actual sensory data, allowing a mobile robot to traverse a simple obstacle course at average speeds of 0.4-0.6 m/sec. However, this method, as well as *inherently* all PFMs, suffer from a number of severe drawbacks, as discussed in Section 2. To remedy these shortcomings, we have developed a new obstacle avoidance method, which eliminates *all* of the limitations of PFMs. This method, called the *vector field histogram* (VFH), is explained in Section 3. Section 4 explains how a combination of these two methods yields efficient and safe real-time obstacle avoidance for fast mobile robots. Finally, Section 5 briefly describes our mobile robot CARMEL, which was used in all the experiments described in this paper.

2. The Virtual Force Field (VFF) Method

The VFF method is a unique method for real-time obstacle avoidance for fast mobile robots. This method is specifically designed to accommodate and compensate for inaccurate range readings from ultrasonic or other sensors [Borenstein and Koren, 1989]. To do so, the VFF method uses a two-dimensional Cartesian grid, called the *histogram grid* ***C***, to represent data from ultrasonic (or other) range sensors. Each cell *(i,j)* in the *histogram grid* holds a *certainty value* (CV) $c_{i,j}$ that represents the confidence of the algorithm in the existence of an obstacle at that location. This representation was derived from the *certainty grid* concept that was originally developed by Moravec and Elfes, [1985]. In the *histogram grid*, CVs are incremented when the range reading from an ultrasonic sensor indicates the presence of an object at that cell.

Combining the *histogram grid* (as the world model) with the potential field concept, the VFF method allows to immediately use real-time sensor information to generate repulsive force fields. Fig. 1 illustrates this approach: As the vehicle moves, a square "*window*" accompanies it, overlying a region of ***C***. We call this region the "*active region*" (denoted as ***C****), and cells that momentarily belong to the *active region* are called "*active cells*" (denoted as $c^*_{i,j}$). In our current implementation, the size of the window is 33×33 cells (with a cell size of 10×10 cm), and the window is always centered about the robot's position.

Each *active cell* exerts a virtual repulsive force $\boldsymbol{F}_{i,j}$ toward the robot. The magnitude of this force is proportional to $c^*_{i,j}$ and inversely proportional to d^n, where d is the distance between the cell and the center of

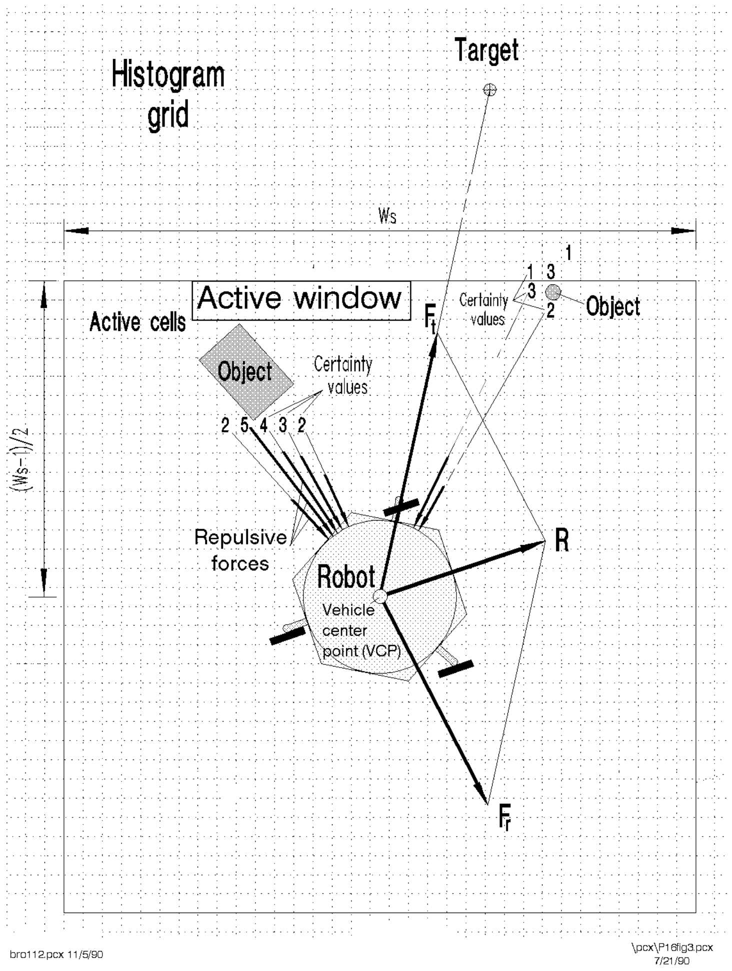

Figure 1: The Virtual Force Field (VFF) concept: Occupied cells exert repulsive forces onto the robot, while the target applies an attractive force.

the vehicle, and *n* is a positive number (usually, *n*=2). All virtual repulsive forces add up to yield the resultant repulsive force $\boldsymbol{F}_r$.

Simultaneously, a virtual attractive force $\boldsymbol{F}_t$ of constant magnitude is applied to the vehicle, "pulling" it toward the target.

Summation of $\boldsymbol{F}_r$ and $\boldsymbol{F}_t$ yields the *resultant force vector* ***R***. The direction of ***R*** is used as the reference for the robot's steering command.

In the course of our experimental work with the VFF algorithm, we identified the following four severe problems that are inherent to PFMs and independent of the particular implementation:

1. Trap situations due to local minima (cyclic behavior).
2. No passage between closely spaced obstacles.
3. Oscillations in the presence of obstacles.
4. Oscillations in narrow passages.

Our paper [Koren and Borenstein, 1991] introduced a mathematical analysis of these inherent problems. As the mathematical analysis shows, one main contributing factor to instability of motion with PFMs is the non-linear force function of the repulsive forces. Usually, this function must be strong enough to be effective at a certain distance form the obstacle, so that the robot can commence an avoidance maneuver in time.

3. The Vector Field Histogram (VFH)

To overcome these problems, we have developed a new obstacle avoidance method called the *vector field histogram* (VFH). The VFH method builds the *histogram grid* the same way the VFF method does. However, the VFH method then introduces an *intermediate data-representation*, called the *polar histogram.* The *polar histogram* retains the statistical information of the *histogram grid* (to compensate for the inaccuracies of the ultrasonic sensors), but reduces the amount of data that needs to be handled in real-time. This way, the VFH algorithm produces a sufficiently detailed spatial representation of the robot's environment for travel among densely cluttered obstacles, without compromising the system's real-time performance. The VFH method was introduced in [Borenstein and Koren, 1990] and a detailed discussion is presented in [Borenstein and Koren, 1991].

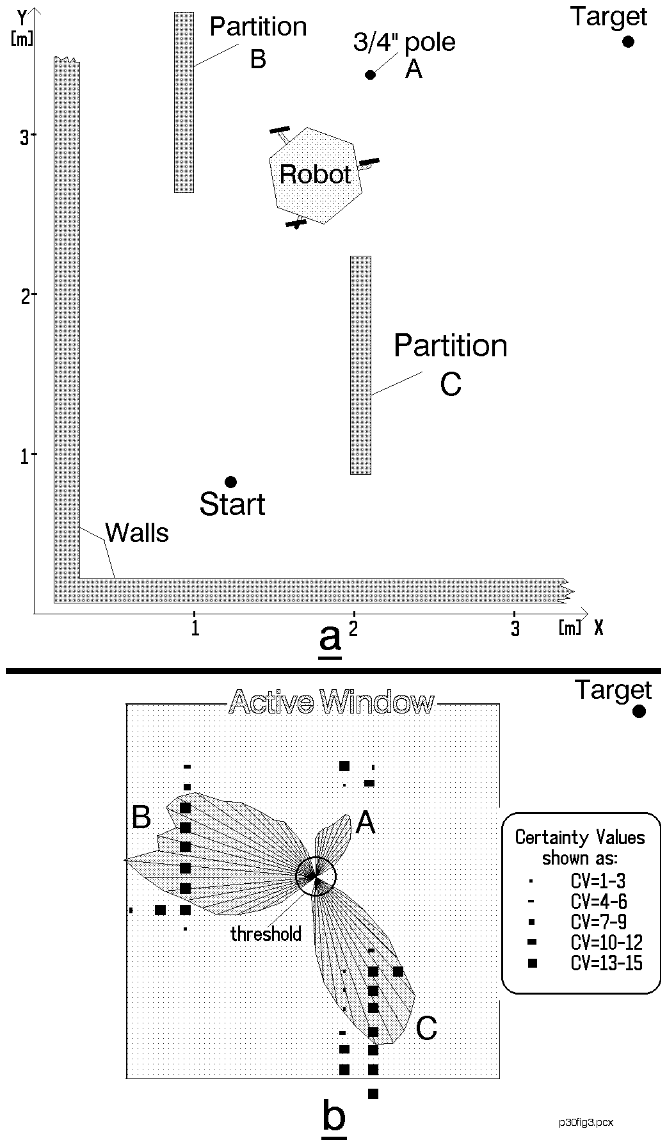

Figure 2:
a. A typical obstacle setup in our lab.
b. The *polar histogram* overlaying part of the *histogram grid* (black dots).

3.1 Creating the Polar Histogram

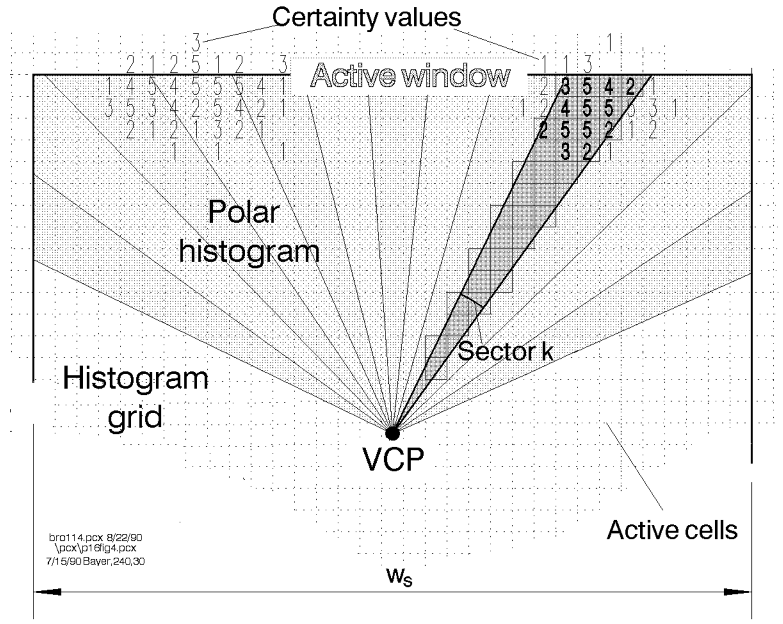

Figure 3: Mapping the *histogram grid* onto the *polar histogram.*

The *polar histogram* ***H*** is an array comprising 72 elements; each element represents a 5°-sector of the robot's surroundings. During each sampling interval, the *active region* of the *histogram grid* ***C**** is mapped onto ***H*** as shown in Fig. 2, resulting in 72 values that can be interpreted as the *instantaneous polar obstacle density* around the robot. Fig. 3 is an example of the different stages through which the *polar histogram* is built: Fig. 3a shows a typical obstacle course in our laboratory. Fig. 3b depicts part of the *histogram grid*, as it was built by the ultrasonic sensors during the robot's travel up to momentary position. The big black blobs in Fig. 3b indicate filled cells in the *histogram grid*. Different *certainty values* (CVs) are shown as blobs of different sizes) as explained in the Figure. Overlaying the *histogram grid* is a graphic representation of the *polar histogram*, created by the mapping process of Fig. 2. The size of each 5°-sector indicated the *polar obstacle density* in the corresponding direction.

3.2 Computing the Steering Control

After the *polar histogram* has been constructed, the VFH algorithm computes the required steering direction for the robot, θ. As can be seen in Fig. 3b, 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 selects the one that most closely matches the direction to the target. Note that the width of a valley, W_V, can be measured in terms of the number of consecutive sectors that are below threshold. A small W_V indicates a narrow passageway or corridor.

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, θ is chosen to be in the center of the valley, in order to maintain equal clearance on each side of the robot.

4. The Combined Vector Field (CVF) for Non-point Mobile Robots

While both VFF and VFH methods were originally designed for *point-sized* robots, a combination of these methods, called the *combined vector field* (CVF), can be used to efficiently guide *non-point* mobile robots through densely cluttered obstacle courses. The combination of the two methods allows us to use each one to its advantage, while avoiding the disadvantages. For example, since stable motion and better spatial resolution are the strength of the VFH method, it is used to determine the *principal steering direction*. The VFF algorithm, on the other hand, can provide local *corrective measures* to account for the shape of the vehicle. Limiting the potential fields-based VFF method to a corrective function, we can use *steep* force profiles with only *short-range* effects. This way we reduce the oscillatory tendency of PFM-based obstacle avoidance. The following discussion explains in more detail how the *combined vector field* works. Note that paragraph enumerations a) through f) correspond to Fig. 4.

a) Computing the *principle steering direction* θ with the VFH method

The task of the VFH component is to protect the vehicle from *frontal collisions* while traveling at high speed. To do so, The VFH algorithm is applied at a point CP_1 to determined the *principal steering direction* θ. A vector $\boldsymbol{F}_{VFH}$ is computed and applied in that direction.

CP_1 is located on the longitudinal axes of the vehicle, but its optimal location (in terms of distance l from the front of the vehicle) differs for different vehicles. l depends on the width of the vehicle and certain parameters in the VFH algorithm, which in turn depend on the type, location, and performance characteristics of the range sensors. Consequently, it is difficult to determine l analytically; however, a *good* location can be found by a few simple experiments. A detailed discussion is beyond the scope of this paper, but a general rule of thumb is that l should be larger for wide vehicles, and smaller for narrow vehicles.

b) Local protection with the VFF method

The VFF method is applied to each one of the *n act-on* points A_n on the periphery of the robot. Technically, this is done for each *act-on* point A_n by adding up all individual repulsive forces $\boldsymbol{F}_{i,j}$ from filled cells in the *histogram grid*, yielding n repulsive forces $\boldsymbol{F}_k$ (note that Fig. 4b does not show the *histogram grid* explicitly). The force fields that act on each *act-on* point are very steep; in our application they are generated by $\boldsymbol{F}_{i,j} \propto 1/d^4$. This way, the effective range is very short; about 40-50 cm in our system. Consequently, $\boldsymbol{F}_k \neq 0$ only when an *act-on* point is very close to an obstacle.

Since the vehicle is protected from frontal collisions by the VFH method, we are only interested in the lateral components of $\boldsymbol{F}_k$, as indicated in Fig. 4b by the dotted vectors. The lateral components $\boldsymbol{F}_k'$ provide *local protection* to all areas of the vehicle that have an associated *act-on* point. In our experimental system, we "placed" five *act-on* points on each side of the vehicle.

c) Extracting the relevant information

Applying the principle of *free-body diagrams*, all lateral forces $\boldsymbol{F}'_k$ can be replaced by a single force $\boldsymbol{F}$ and a *moment* $\boldsymbol{M}$, acting on the vehicle at the *center point CP*.

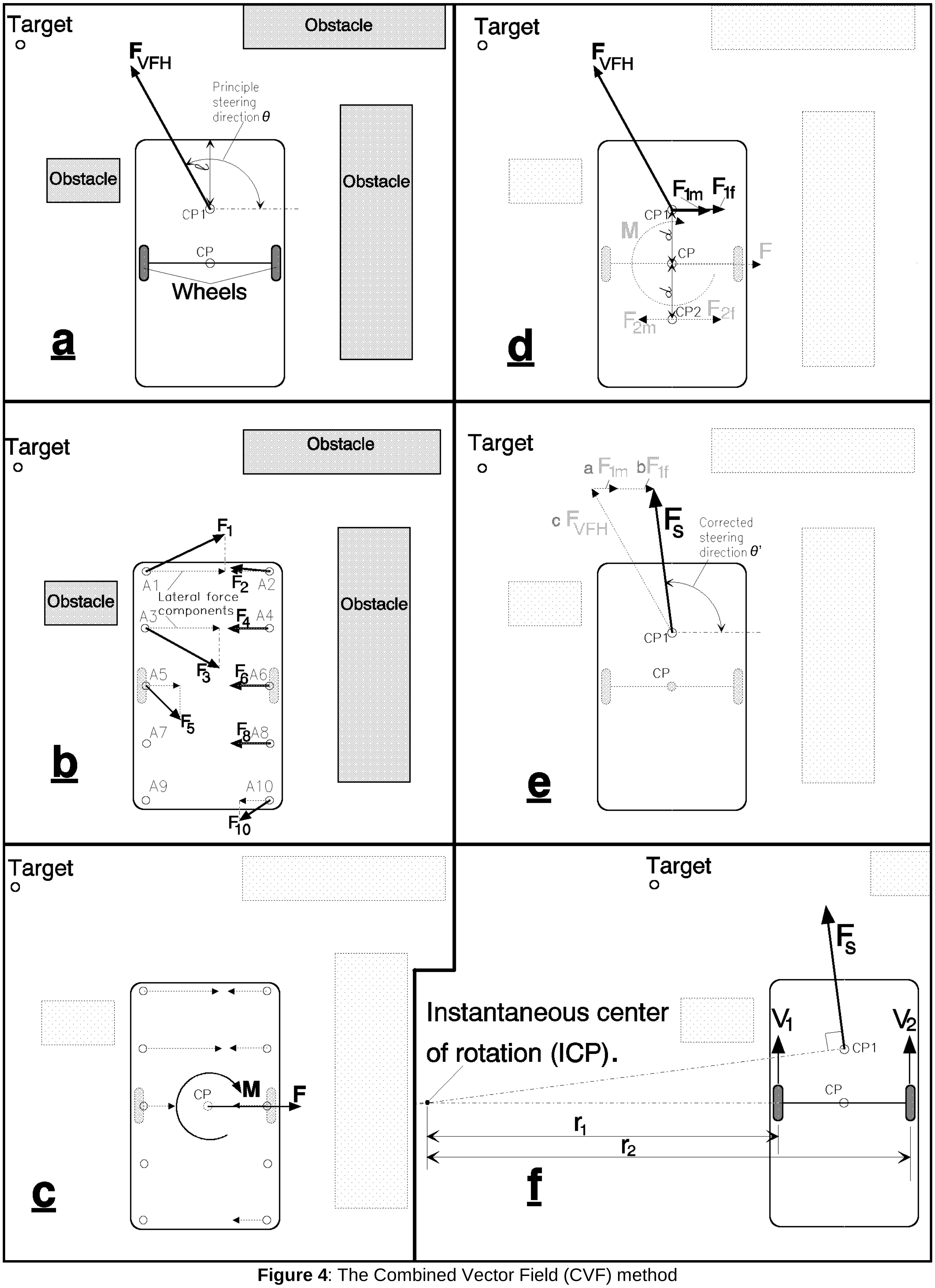

Figure 4: The Combined Vector Field (CVF) method

d) Manipulating the correction information

Next, the moment $\boldsymbol{M}$ is decomposed into a *force couple* $\boldsymbol{F}_{1m}$ and $\boldsymbol{F}_{2m}$. $\boldsymbol{F}_{1m}$ acts on CP_1 and $\boldsymbol{F}_{2m}$ acts on a symmetrically located point CP_2 in the rear part of the vehicle. Note that the *force couple* is statically equivalent to the moment $\boldsymbol{M}$ if it is computed as $\boldsymbol{F}_{1m} = \boldsymbol{F}_{2m} = \boldsymbol{M}/d$, where d is the distance between CP and CP_1 (or CP_2). In another statically equivalent operation $\boldsymbol{F}$ is replaced by two forces $\boldsymbol{F}_{1f} = \boldsymbol{F}_{2f} = \boldsymbol{F}/2$. These forces act on CP_1 and CP_2, respectively. $\boldsymbol{F}_{2m}$ and $\boldsymbol{F}_{2f}$ can be discarded for the remaining discussion (although vehicles with more than two degrees-of-freedom can make use of this information).

e) Computing the corrected steering vector

All three vectors acting on CP_1 can now be combined to compute the *corrected steering vector* $\boldsymbol{F}_s = a\boldsymbol{F}_{1m}+b\boldsymbol{F}_{1f}+c\boldsymbol{F}_{VFH}$. The influence of the *translational* correction component $\boldsymbol{F}_{1f}$ and the *rotational* correction component $\boldsymbol{F}_{1m}$ is controlled by the weighing coefficients a and b. The coefficient c is determined as $c = 1/W_V$, i.e., the inverse of the width of the *candidate-valley* selected by the VFH method (see Sect. 3.2). The effect of this latter scaling factor is as follows: when the robot circumnavigates a single obstacle, the *candidate-valley* will be wide and c will be small. Consequently, the correctional effect of $\boldsymbol{F}_{1f}$ and $\boldsymbol{F}_{1m}$ will be quite significant. In a narrow corridor, on the other hand, the robot is *forced* to be close to the walls and very large repulsive forces develop (because of the non-linear relation $\boldsymbol{F}_{i,j} \propto 4^n$). Furthermore, small diversions of the robot from the centerline result in dramatic fluctuations of the forces and consequently in oscillatory motion. However, c will be relatively large and will dominate the resultant vector $\boldsymbol{F}_s$. This way, the oscillatory behavior usually associated with potential field control is avoided.

f) Controlling the Drive Motors

Once the desired steering direction is determined, the appropriate command must be issued to the motor controllers. For different vehicle kinematics different methods are required. However, one almost universal method (with slight modifications) makes use of the concept of *instantaneous center of rotation* (ICR). The ICR is a well-known concept in kinematics; it allows to describe the motion of a rigid body in terms of a single rotation of the body around one point in space. Of course, for any motion other than pure rotation, the ICR will change from instance to instance. In a fast sampled system smooth motion can be approximated by recomputing an ICR during each sampling interval. In the case here, the ICR can be determined as shown in Fig. 4f. Since every point on the rigid body rotates around the ICR, and points at equal distances from the ICR have the same speed, the *desired* ICR is the intersection of lines perpendicular to $\boldsymbol{F}_s$ (through CP_1) and to the current direction of motion (the longitudinal axis of the vehicle, through CP).

Once the ICP has been determined, the desired wheel velocities V_1 and V_2 can be computed from the ratio

$$\frac{r_1}{V_1} = \frac{r_2}{V_2} \qquad (1)$$

and

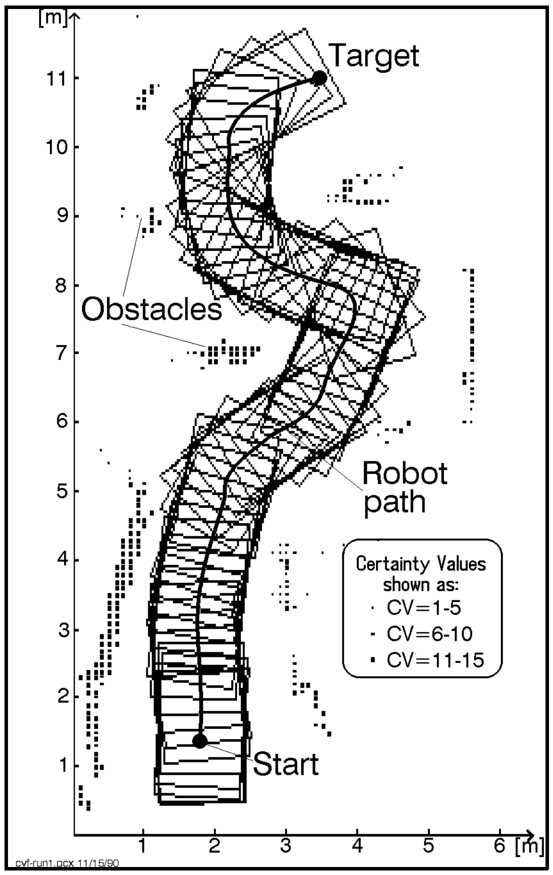

Figure 5: Experimental run with the non-point mobile robot.

$$V_t = \tfrac{1}{2}(V_1 + V_2) \qquad (2)$$

where V_t is the required travel speed.

5. The Experimental System

The experiments described in this paper were run 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 [Cybermation, 1990]. This platform has a maximum travel speed of 0.8 m/sec and a maximum steering rate of 120 deg/sec; it weighs about 125 kg. A Z-80 on-board computer serves as the low-level controller of the vehicle. We equipped this vehicle with a ring of 24 ultrasonic sensors. Two computers were added to the platform: a 20Mhz, 80386-based AT-compatible that runs the CVF obstacle avoidance algorithm, and a PC-compatible single-board computer to control the sensors.

Although the shape and kinematics of CARMEL differ from those of the vehicle described in Fig. 4, CARMEL's behavior can be altered (in the control software) to imitate the behavior of a 2-wheel drive vehicle. Also, the rectangular shape of the vehicle in Fig. 4 was simulated by overlaying the robot's position on the operator's screen with a rectangular wire-frame of size 1.9×1.2 m. A typical experimental run is shown in Fig. 5. Black dots in Fig. 5 show filled cells in the *histogram grid.* The maximum speed in this run was 0.8 m/sec, but the average speed was only 0.5 m/sec, because the robot slows down whenever it heads toward an obstacle) a situation that arose quite often in the densely cluttered obstacle setup of Fig. 5. The iteration time for each complete computation of the CFV algorithm was less than 50 msec on the 20 MHz-386 computer, including the overhead for communication with the onboard sensor and motor control computers.

It should be noted that our method does not explicitly address the dynamics of the vehicle, except for slowing down when an obstacle is ahead of the vehicle. This works satisfactory because the ultrasonic sensor system "looks ahead" more than 2 meters, giving ample time to decelerate. A more detailed description of the slow-down function is given in [Borenstein and Koren, 1991].

6. Conclusions

We have introduced the *combined vector field* (CVF) a new method for controlling and guiding a large, non-point mobile robot among densely cluttered obstacles. This method uses the combination of two different, previously developed obstacle avoidance methods (VFF and VFH) in such a way that the advantages of each method are retained. The CVF method was extensively tested on a mobile robot and yielded smooth, non-oscillatory control at typical travel speeds of up to 0.8 m/sec.

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

7. References

1. Andrews, J. R. and Hogan, N., 1983, "Impedance Control as a Framework for Implementing Obstacle Avoidance in a Manipulator." In *Control of Manufacturing Processes and Robotic Systems*, Eds. Hardt, D. E. and Book, W. J., ASME, Boston, pp. 243-251.
2. Barraquand, J., and Latombe, J. C., 1990, "A Monte-Carlo Algorithm for Path Planning With Many Degrees of Freedom." *The 1990 IEEE International Conference on Robotics and Automation*, Cincinnati, Ohio, May 13-18, pp. 1712-1716.
3. Borenstein, J. and Koren, Y., 1989, "Real-time Obstacle Avoidance for Fast Mobile Robots." *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 19, No. 5, Sept/Oct, pp. 1179-1187.
4. Borenstein, J. and Koren, Y., 1990, "Real-time Obstacle Avoidance for Fast Mobile Robots in Cluttered Environments." *Proceedings of the 1990 IEEE International Conference on Robotics and Automation*, Cincinnati, Ohio, May 13-18., pp. 572-577.
5. Borenstein, J. and Koren, Y., 1991, "The Vector Field Histogram) Fast Obstacle-Avoidance for Mobile Robots." *IEEE Journal of Robotics and Automation*, Vol. 7, No. 3, June, pp. 278-288. .
6. Cybermation, 1990, "K2A Mobile Platform." *Commercial Offer*, 5457 JAE Valley Road, Roanoke, VA 24014.
7. Hague, T., Brady, M., and Cameron, S., 1990, "Using Moments to Plan Paths for the Oxford AGV." *Proceedings of the 1990 IEEE International Conference on Robotics and Automation*, Cincinnati, Ohio, May 13-18, pp. 210-215.
8. Khatib, O., 1985, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots." *1985 IEEE International Conference on Robotics and Automation*, St. Louis, Missouri, March 25-28, pp. 500-505.
9. Koren, Y. and Borenstein, J., 1991, "Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation." Proceedings of the *IEEE Conference on Robotics and Automation* Sacramento, California, April 7-12, , pp. 1398-1404.
10. Lin, P. and S. Chang, S., 1990, "Motion Planning for a Rod Amidst Obstacles of Arbitrary Shapes." Submitted to the *IEEE Journal of Robotics and Automation*.
11. Lozano-Perez, T., 1981, "Automatic Planning of Manipulator Transfer Movements." *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-11, No. 10, October, pp. 681-698.
12. Moravec, H. P. and Elfes, A., 1985, "High Resolution Maps from Wide Angle Sonar." *IEEE Conference on Robotics and Automation*, Washington D.C., pp. 116-121.
13. Tilove, R. B., 1990, "Local Obstacle Avoidance for Mobile Robots Based on the Method of Artificial Potentials." *1990 IEEE International Conference on Robotics and Automation*, Cincinnati, Ohio, May 13-18, pp. 566-571.