Sonar-based Obstacle Avoidance for a Large, Non-point, Omni-directional Mobile Robot

Hong Yang[1], Johann Borenstein[1], David Wehe[2]
[1] Department of Mechanical Engineering and Applied Mechanics
[2] Department of Nuclear Engineering and Radiological Sciences
The University of Michigan
Ann Arbor, MI 48109-1101,USA
yanghong@umich.edu, johannb@umich.edu, dkw@umich.edu

ABSTRACT

This paper presents a method for reliable, real-time obstacle avoidance for an omnidirectional mobile robot, called "OmniMate." The OmniMate is a large, rectangular-footprint platform with a flat loading deck measuring 184×92 cm (72×36").

To protect this vehicle from collisions we equipped it with a ring of 32 ultrasonic sensors and we developed a modified version of our earlier Vector Field Histogram (VFH) method, called *Double-VFH* (DVFH). The DVFH method accounts for the rectangular footprint of the OmniMate by calculates simultaneously two *vector field histograms* centered at two points along the longitudinal axis of OmniMate.

Experimental results of the OmniMate traversing densely cluttered obstacle courses at an average speed of 0.2 - 0.3 m/sec and a maximum speed of 0.4 m/sec are also presented.

1. INTRODUCTION

The OmniMate, shown in Figure 1, is a rectangular-shaped mobile platform measuring 184×92 cm (72×36") with three Degrees-of-Freedom of motion. The OmniMate is made from two "trucks" that are connected by a so-called compliant linkage and a flat loading deck. Each truck can rotate relative to the loading deck and the compliant linkage allows linear relative motion between the two trucks. The complete description of the kinematic design and control of the OmniMate is given in [Borenstein and Evans, 1997].

Because of its elongated, rectangular shape it is quite difficult to provide the vehicle with adequate all-around protection from collisions with unexpected obstacles. The easiest method for guiding a mobile robot of any shape around obstacles is to "shrink" the robot conceptually to a single point, while the obstacle perimeter is enlarged by half of the robot's largest dimension [Lozano-Perez, 1987]. This method, known as "configuration space approach" works well with relatively small, circular-footprint mobile robots. However, with larger, non-circular platforms this approach is too conservative as it doesn't take into account the robot's orientation.

Figure 1: OmniMate with ultrasonic sensors for fully omnidirectional obstacle avoidance

Some researchers apply global path planning techniques to optimize the orientation of an irregular shaped mobile robot relative to the obstacles in the environment, for example by using Voronoi Diagram [Barraquand, 1992]. However, this method is not quite suitable for real-time application in dynamic environments.

The solution presented in this paper is based on the *vector field histogram* (VFH) method, which was originally developed by Borenstein and Koren [1991] for circular-footprint robots. A review of that method and some other related methods is given in Section 2. Then, in Section 3, we introduce the new Double-VFH (DVHF) method, which is designed to work with rectangular-footprint platforms with omnidirectional motion capabilities, like the OmniMate. The complete experimental system is described in Section 4 and experimental results are presented in Section 5.

2. EARLIER WORK

In earlier research at the University of Michigan several related obstacle avoidance methods were developed. In this section we will briefly review these earlier methods in order to document the evolution toward the DVFH method, which is the focus of this paper.

2.1 The Virtual Force Field (VFF) Method

Borenstein and Koren [1989] developed one of the earliest obstacle avoidance methods for mobile robots, called the *virtual force field* (VFF) method. The VFF method worked 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/s.

The VFF method is specifically designed to accommodate and compensate for inaccurate range readings from ultrasonic or other sensors. 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. Figure **2** 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****),

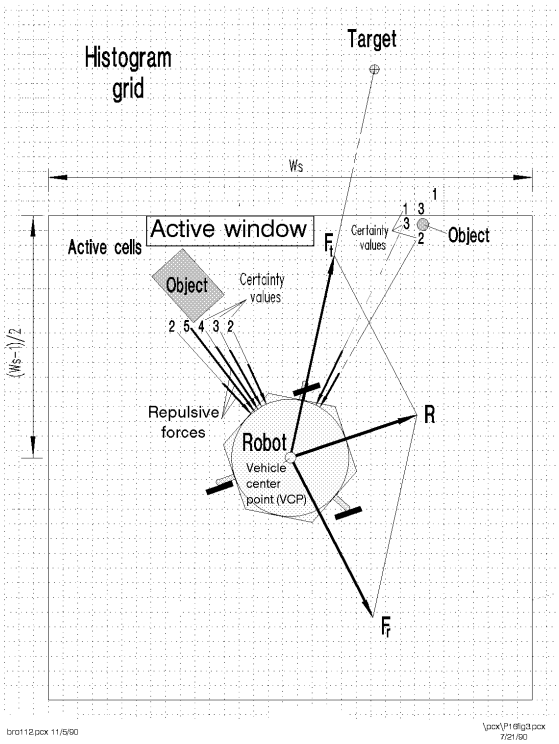

Figure 2: The Virtual Force Field (VFF) concept: Occupied cells exert repulsive forces onto the robot, while the target applies an attractive force.

and cells that momentarily belong to the *active region* are called "*active cells"* (denoted as $c^*_{i,j}$). In the original 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 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* $\boldsymbol{R}$. The direction of $\boldsymbol{R}$ is used as the reference for the robot's steering command.

In the course of our experimental work with the VFF algorithm we found that the potential field approach caused oscillations in the presence of obstacles, especially in narrow passages.

2.2 The Vector Field Histogram (VFH)

To overcome these problems, Borenstein and Koren [1991] developed the *vector field histogram* (VFH) method. 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.

2.2.1 Creating the Polar Histogram

The *polar histogram* $\boldsymbol{H}$ 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* $\boldsymbol{C}^*$ is mapped onto $\boldsymbol{H}$ as shown in Figure 3, resulting in 72 values that can be interpreted as the *instantaneous polar obstacle density* around the robot. Figure 4 shows two representations of the same sample polar histogram. In Figure **4**a the polar histogram is plotted as a bar chart, with bars of different heights indicating the *polar obstacle density* in different directions. In Figure 4b is polar histogram is overlaying the *histogram grid,* from which it was created by the mapping process of Figure 3. The size of each blob in Figure 4b represents the certainty value of the corresponding cell in the histogram grid.

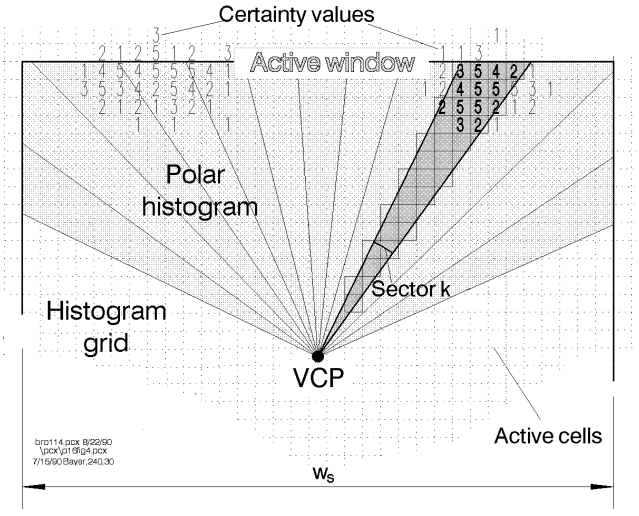

Figure 3: Mapping the histogram grid onto the polar histogram.

2.2.2 Computing the Steering Control

After the *polar histogram* has been constructed, the VFH algorithm computes the required steering direction for the robot. As can be seen in Figure 4b, 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 usu-

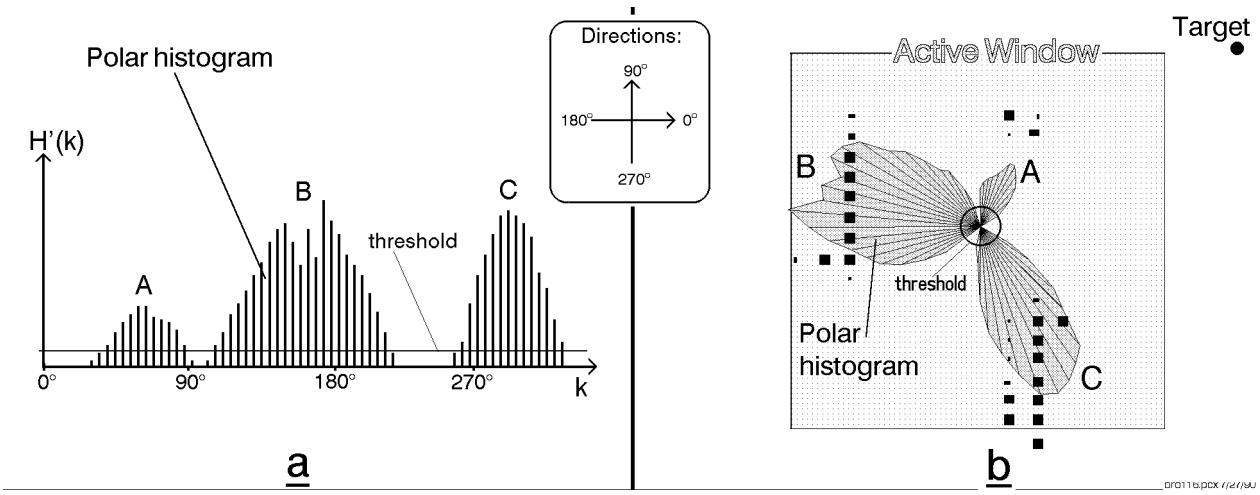

Figure 4: Snapshot of a momentary polar histogram for a sample obstacle course with three obstacles A, B, and C.

ally several candidate-valleys, the algorithm selects the one that most closely matches the direction to the target.

2.3 The Combined Vector Field (CVF) Method

As a first attempt at avoiding obstacles with large, rectangular-shaped mobile robots Borenstein and Raschke [1991] developed the *combined vector field* (CVF) method, which combines VFF and VFH. With this method, the principal steering direction of the non-point robot is determined by the VFH algorithm, while the VFF algorithm applies virtual forces as a corrective measure to account for the robot's dimension. Although the CVF method used only steep force profiles with short-range effects in its VFF components, the oscillation tendency of this potential fields-based was observed in high obstacle density environments.

3. THE DOUBLE-VFH (DVFH) METHOD

This paper introduces the Double-VFH (DVFH) method, a modification of the original VFH method for the elongated, rectangular-shaped OmniMate mobile robot with three DOF. As explained in Section 1, the OmniMate is based on two trucks, which are kinematically simple differential drive vehicles. The DVFH algorithm treats these two trucks identically, that is, either one may lead or follow.

In order to implement the DVFH algorithm a command velocity vector $V_{A/B}$ for each truck must be calculated once during each sampling interval. Nominal command velocity vectors (i.e., those that implement a desired path without consideration of obstacles) are easily derived from the desired trajectory of the center point of the robot.

In the DVFH method, separate polar histograms are built for truck A and truck B, centered at the two points O_A and O_B located at the center-points of each truck as shown in Figure 5.

Using the original VFH method as described in Section 2.2.2 candidate directions for travel are calculated separately for each polar histogram and the arbitration scheme described next is applied to resolve conflicts.

3.1 Cost-based Arbitration

When separate directions for travel are computed for each of the two trucks, then it is quite likely that these directions conflict. The desired behavior for resolving such a conflict is described first verbally in the following example, followed by a formal discussion of the cost function strategy used to implement this behavior.

As an example for the desired behavior, consider a situation where truck A functions as the leader. Truck A should have high priority and it should determine the principal steering direction for the whole robot, while imposing certain constraints on truck B.

Truck B, which is forced to be the follower when truck A leads, should act with a tendency to align itself with the orientation of the robot. This minimizes the conflict between A and B and helps recover from any deviation from the nominal extension of the compliant linkage.

The *cost function* $W(\Delta a_1, \Delta a_2, \Delta a_3)$ is described as:

$$W = k_1 \Delta a_1 + k_2 \Delta a_2 + k_3 \Delta a_3 \qquad (1)$$

where:

Δa_1 — the difference between the candidate steering direction of the leading truck and the current orientation of robot.

Δa_2 — the difference between the candidate steering direction of the following truck and the current orientation of robot.

Δa_3 — the difference between the candidate steering direction of the robot center point and the previous one.

k_1, k_2, k_3 — weighting coefficients.

This cost function is used by the DVFH algorithm to determine for each truck its motion mode: *leader* or *follower*. The larger the first term in the cost function W, the larger the tendency for this truck to switch from leader mode to follower mode; The larger the second term in the cost function, the larger the tendency to maintain the current motion mode unchanged; The third term determines the tendency for the robot to deviate from the original path when an obstacle is encountered.

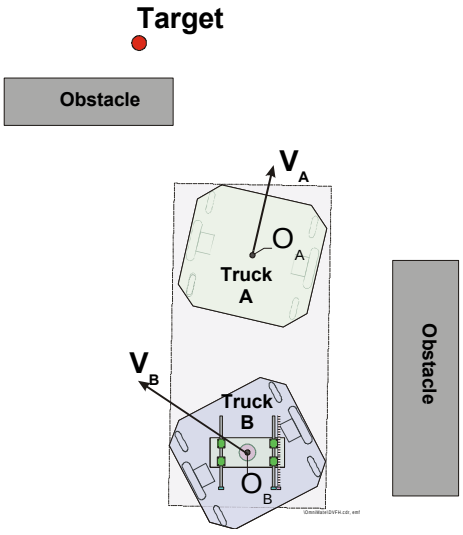

Figure 5: Selecting directions for both trucks.

The motion mode decision-making procedure is shown in Figure 6. Based on the odometry and ultrasonic sensor information, the direction to the target and the orientation of the vehicle are determined. Also, the candidate steering directions of truck A and truck B are precalculated. W_A, the cost value if truck A serves as leader and W_B, the cost value if truck B serves as leader are calculated by the cost function. By comparing W_A and W_B, the motion mode and priority levels are determined for both truck A and B.

When traveling among densely cluttered obstacles there are situations where Trucks A and B might switch roles of leader and follower rapidly and repeatedly. This problem is overcome by implementing a hysteresis function based on two thresholds, P_A and P_B (with $P_A < P_B$):

The resulting decision strategies are summarized below:

- If $W_A < W_B$ and $|W_A - W_B| < P_A$, then truck A will change to or remain as leader.
- If $W_A > W_B$ and $|W_A - W_B| > P_B$, then truck B will change to or remain as leader.
- In any other case both trucks remain in their current motion mode.
- Whenever one truck changes its motion mode, the other will make the change according to it.

After the motion mode is decided, the leader can select a candidate steering direction autonomously (as explained in Section 2.2.2). At the same time the follower must submit to restrictions on its choice of steering directions and speeds. This approach is visualized in Figure 7.

3.2 Special considerations

One special case exists in so-called trap situations that arise when the leader's forward direction is blocked by obstacles or when a sharp direction change is needed for the follower in order to clear a nearby obstacle. In this case the cost function W is calculated again and compared with the threshold, and the robot will have the tendency to switch the motion mode and drive backward to find a new candidate direction toward the target under the control of the newly decided leader. In this case a more efficient high-level path planner is needed to solve the local minimum problem.

If the robot is used in tele-autonomous mode (i.e., a mode in which a human operator prescribes a direction of travel and the OmniMate modifies the direction autonomously to avoid collisions), it may be desired to let the robot stand still to wait for a new command input. The new command input will often also result in a change in motion mode.

Another special case is an environment with densely cluttered obstacles. In such environments it is often necessary to reduce the overall speed of the platform. To address this problem a speed reduction function is defined as follows.

Let θ be the rotation angle of the truck relative

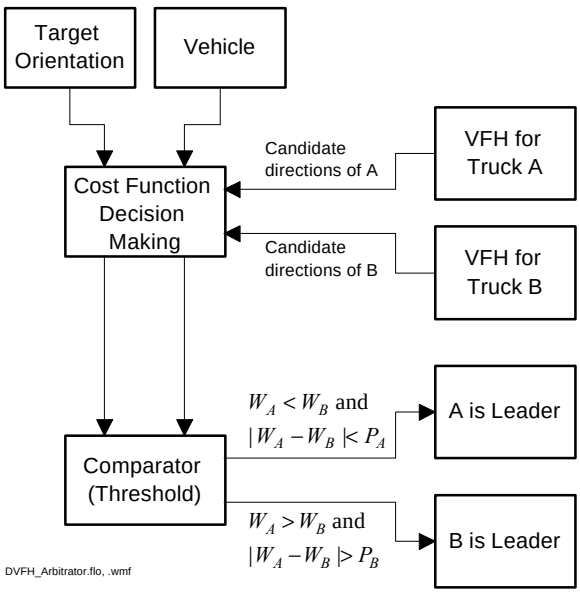

Figure 6: Motion mode decision-making strategy

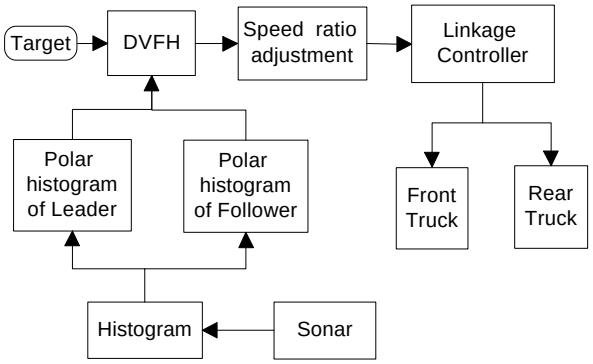

Figure 7: Block diagram of the DVFH obstacle avoidance system for the OmniMate.

to the axis of the robot, and let ω be the rate of this rotation. Then the reduced translation speed V' is:

$$V = V'(1 - K_1 \omega/\omega_{max}) \times (1 - K_2 \theta/\theta_{max}) + V_{min} \quad (2)$$

Where:

V' : Reference translational speed.

K_1, K_2 : Empirically determined constants for angular speed ratio and rotation angle, which cause a sufficient reduction in translational speed.

ω_{max} : Maximum rotation speed allowed.

θ_{max} : Maximum rotation speed allowed.

V_{min} : A lower speed limit to compensate for actuators dead zone. Currently V_{min} = 0.5 cm/sec.

4. EXPERIMENTAL RESULTS

The obstacle avoidance system of the OmniMate comprises a ring of 32 Polaroid ultrasonic sensors, an electronic interface board equipped with four 68HC11 micro-controllers, and software that runs on the OmniMate onboard PC. The sonar ring is mounted underneath the loading deck and along the periphery of the OmniMate (see Fig. 1). The electronic interface board implements the earlier developed method for Error Eliminating Rapid Ultrasonic Firing (EERUF) [Borenstein and Koren, 1995]. The EERUF method allows each sonar to detect and reject readings caused by crosstalk or other ultrasonic noise in the environment. This feature allows for much faster firing rates than those used in conventional sonar-based obstacle avoidance systems, yet provides more reliable and noise-free data (see [Borenstein and Koren, 1995] for details on the EERUF method).

The *DVFH* obstacle avoidance method proposed in this paper was extensively tested on the OmniMate in the Mobile Robotics Lab of the University of Michigan. Typically the robot was able to traverse even densely cluttered obstacle coursed at a maximum speed of 0.4 m/sec, although the average speed was only 0.2 - 0.3 m/sec because the robot slowed down whenever it came close to obstacles. Higher maximum speeds were possible in less densely cluttered environments.

5. CONCLUSIONS

This paper presented the DVFH obstacle avoidance method for the OmniMate robot. The DVFH method was extensively tested on the OmniMate and found to provide smooth and robust control.

The DVFH method as described in this paper provides only one mode of motion with obstacle avoidance, which, in practice, is similar to a follow-the-leader approach. Yet, the OmniMate is capable of full 3 DOF motion, including sideways crabbing or diagonal motion. We are currently developing other versions of DVHF that will provide full obstacle avoidance for other types of motion.

Acknowledgements

This material is based upon work supported by DOE under Award No. DE-FG04-86NE37969.

6. REFERENCES

1. Barraquand, J., Langloi, B., and Latombe, J.C., 1992, "Numerical Potential Field Techniques for Robot Path Planning." *IEEE Trans. on Systems, Man, and Cybernetics*, Vol.22, No.2, pp. 224-241.

2. 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.

3. 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, pp. 278-288.

4. Borenstein, J. and Koren, Y., 1995, "Error Eliminating Rapid Ultrasonic Firing for Mobile Robot Obstacle Avoidance." *IEEE Trans. on Robotics and Automation*, Feb., Vol. 11, No. 1, pp. 132-138.

5. Borenstein, J. and Raschke, U., 1992, "Real-time Obstacle Avoidance for Non-Point Mobile Robots." *SME Trans. on Robotics Research,* Vol. 2, pp. 2.1-2.10.

6. Borenstein, J. and Evans, J., 1997, "The OmniMate Mobile Robot – Design, Implementation, and Experimental Results." *Proc. of the 1997 IEEE Int. Conference on Robotics and Automation*, Albuquerque, NM, Apr. 21-27, pp. 3505-3510.

7. Borenstein, J. and Koren, Y, 1995, "Error Eliminating Rapid Ultrasonic Firing for Mobile Robot Obstacle Avoidance." *IEEE Trans. on Robotics and Automation, February, Vol. 11, No. 1, pp 132-138.*

8. Lozano-Perez, 1987, "A simple Motion Planning Algorithm for General Robot Manipulators." *IEEE Trans. on Robotics and Automation*. Vol. RA-3, No.3, pp. 224-238

9. Moravec, H. P. and Elfes, A., 1985, "High Resolution Maps from Wide Angle Sonar." *Proc. of the IEEE Conf. on Robotics and Automation*, Washington, D.C., pp. 116-121.