A Practical Obstacle Detection and Avoidance System*

Sumit Badal   Srinivas Ravela   Bruce Draper   Allen Hanson

Computer Vision Research Laboratory
University of Massachusetts, Amherst, MA 01002

Abstract

A practical real-time system for passive obstacle detection and avoidance is presented. Range information is obtained from stereo images by first computing a disparity picture from the image pair and extracting points above the ground plane. Then these points are projected onto the ground plane and an *Instantaneous Obstacle Map (IOM)* is obtained. The IOM is transformed into a one dimensional *steering vector* that represents the *hindrance* associated with steering in a particular direction and then a one dimensional search is performed on the steering vector for an angle with least hindrance. The steering direction and hindrance value are used to set the speed of the vehicle. This system has been implemented on the *Mobile Perception Lab (MPL)* at University of Massachusetts at Amherst with considerable success, running at $2Hz$ for 256×240 sized images.

1 Introduction

Obstacle detection systems typically compute the position of obstacles relative to a mobile agent by using range information. Range information may be obtained from ladar (laser ranging) [1, 2], sonar (sound ranging) [3, 4] or vision based techniques [5, 6, 7, 8].

Video technology has the advantages of low cost, low power consumption and a high degree of mechanical reliability. The speed and accuracy of vision algorithms typically scale with faster computing platforms. Further, the passive nature of the camera implies that it has no detectable signature and is relatively free of signal interference in the presence of other sensors. However, the relatively slow speed of stereo-based obstacle detection has limited its application in real-world problems.

Existing implementations of correspondence based algorithms either fail to meet real time requirements or run at coarse resolutions. We believe that the computation of a dense depth map or a detailed 3D reconstruction of the world is difficult, computationally prohibitive and not required in some applications. By making assumptions and compromises to simplify and speed up the task of detection, we have developed a practical stereo based system that is effective for obstacle avoidance at speeds reaching 7MPH (see section 5) on the Mobile Perception Laboratory (MPL) at the University of Massachusetts.

The obstacle detection algorithm trades accuracy and image resolution for speed, effectively limiting the accuracy of obstacle position estimation to within a small area. Further, since the MPL is a large vehicle (a HMMWV[1]), the observable area is navigable only if it has few obstacles. We believe that for sparsely distributed obstacles, a *reflexive* approach, without resort to detailed path planning, is sufficient for effective avoidance. Our approach is reflexive in the sense that the steering and speed are modulated by a fast greedy algorithm based purely on the instantaneous local perception of obstacle positions. The avoidance module runs synchronously with the detection algorithm and does not require the vehicle to stop in the course of avoiding a navigable stretch of sparse obstacles.

The obstacle avoidance algorithm prioritizes obstacles by their distance from the vehicle and generates proportionally larger turns (to avoid them) as the obstacles get nearer (to the vehicle). Therefore the algorithm is relatively insensitive to minor errors in obstacle positions or actuator servos. Is is stable and has been observed to generate desirable paths over time. We have also demonstrated that a reflexive approach can coexist with other behaviors and perform composite tasks, such as maintaining an overall heading while avoiding obstacles in a cross-country scenario. Further, this approach also serves to improve the robustness of the execution of a planned path by reacting to local perturbations of a world state that arise from

*This work was funded in part by ARPA (via TACOM) grant DAAE07-91-C-R035 and NSF grant CDA-8922572.

[1]High Mobility Multi-purpose Wheeled Vehicle

the presence of un-modelled obstacles.

Section 2 summarizes related work. Sections 3 and 4 describe the detection and avoidance algorithms respectively. Section 5 describes the implementation details and demonstrates experimental results. Section 6 presents conclusions and section 7 examines limitations of our approach.

2 Related Work

A majority of work on obstacle detection uses active sensors [3, 4, 2, 1]. [9] uses an omni-directional conic mirror as a sensor.

Optical flow forms the basis of obstacle detection in [5, 6, 7, 8]. Most of the flow based methods assume that the vehicle motion can be modelled as a pure translation, which may not be true in the general scenario.

The concept of 2D affine trackability of shallow structures is exploited in [10, 11] to characterize and hence detect obstacles. The implementation of affine tracking is limited to objects with straight line boundaries and a minimal set of three lines need to be hypothesized as a shallow structure, which are selected manually.

[12] presents three algorithms for obstacle detection, two of which are aimed at qualitative yes/no obstacle detection without indicating which points are obstacles. The third algorithm is based on partial 3D scene reconstruction and continuously estimates the ground plane. Its computational needs limits its applicability in real time applications.

For an excellent survey of the effectiveness of stereo algorithms in the ARPA Unmanned Ground Vehicle program, see [13]. The stereo algorithm developed at INRIA aims at 3D scene reconstruction. The SRI algorithm attempts to produce a set of high quality matches using hierarchal techniques. Although the techniques established correspondences at as much as 87% of the points with only a few spike errors, their implementations on existing hardware fall short of real time speed requirements. The algorithm developed by TELEOS attempts to establish the correspondence of selected points at frame rate. Selection of these points is critical for obstacle detection applications, which we believe is still an open problem.

Trinocular stereo is used to detect obstacles in [14]. The implementation in [14] requires three cameras to simplify matching, yet it falls short of our system by a speed factor of 5-10 for comparable resolutions even though it has custom modifications for its stereo hardware. [15, 14, 16, 9] address real time implementations.

[15] and [17] discuss the application of potential fields for obstacle avoidance. These approaches are reactive since the gradients of the potential field at any point determines the velocity of the manipulator towards a specified goal. [18, 19] discuss the application of harmonic functions to plan smooth paths with no local minima. This technique can also be used to generate robust models of the world incrementally. [16] also discusses real-time obstacle avoidance using harmonic potential functions.

[20, 21] describe reflexive/reactive approaches to autonomous vehicle control. [20] is based on the construction of motor schemas, similar to the work on potential fields to generate robot commands. [21] describes an architecture for Autonomous Land Vehicle Navigation that at the lowest levels contains reflexive behaviors that in a broad sense have a commitment to preserving the constitution of the agent. One such behavior is obstacle avoidance. For a survey of reactive systems, see [22].

[23] describes a generalized approach to reflex control for collision avoidance, introduced in [24]. In this paper the authors construct a reflex controller that examines the nearby C-space and rejects commands that would result in a collision.

3 Obstacle Detection

The obstacle detection algorithm developed here simplify the task of detection. Further, certain justifiable assumptions are made to speed up the detection system. These assumptions make the system amenable to real time, real world situations. In this section we first describe the assumptions, then the system requirements, and finally the detection algorithm.

3.1 Assumptions

There are five basic assumptions that underlie the obstacle detection algorithm:

- Obstacles can be defined as objects protruding sufficiently high from the ground or crevices sufficiently deep in the surface. For the system described here, obstacles are restricted to objects that are at least k feet above the ground-plane (see Figure 1); the system is not designed to detect crevices (as explained at the end of this section).

- **Flat Ground:** It is assumed that the ground can be locally represented by a plane. The assumption is justifiable on the basis that the area where

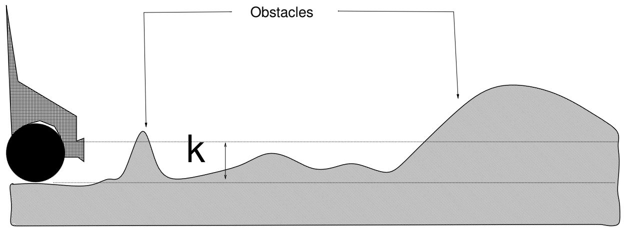


Figure 1: Height **k** defines an obstacle.

a big vehicle can be safely driven is more or less locally flat.

- **Object boundaries form good features:** Obstacles are assumed to be visually distinguishable from the background in the intensity image, since local intensity discontinuities form the basis for matching across stereo image pairs. Note that most correlation based techniques make this assumption.

- **Epipolarity:** Image matching is a two dimensional search that can be reduced to a one dimensional search if constraints imposed by epipolar geometry inherent in an oriented image pair [25] are met. The detection algorithm exploits the epipolarity constraint by employing cameras with identical focal lengths that are aligned up to a scan line.

- **Identical camera/Digitizers:** Identical cameras and digitizers are assumed to simplify the task of processing intensity images and finding correspondences. A difference in focal lengths of the stereo camera lenses introduces a 2D affine transform and obtaining correspondences becomes more complicated. Differences in the dynamic response of the sensors to incident light and scaling or offset of the input signal by the digitizers requires costly and time consuming intensity normalization.

3.2 Calibration

Some of the assumptions require a tedious, but by no means impossible, external stereo camera calibration. Micrometers are used to align the cameras so that the epipolarity constraint is met. Camera and digitizer consistency constraints are met by using cameras, digitizers and auto-irises of identical make and specifications. The camera auto-irises are driven from the video output of a single camera. After the stereo camera pair is aligned, a few points k (as in Figure 1) feet above the ground-plane are marked and the disparity is computed from their right and left camera images. Linear interpolation and extrapolation of these values produces an expected ground plane disparity value (EGP) for every row of the image.

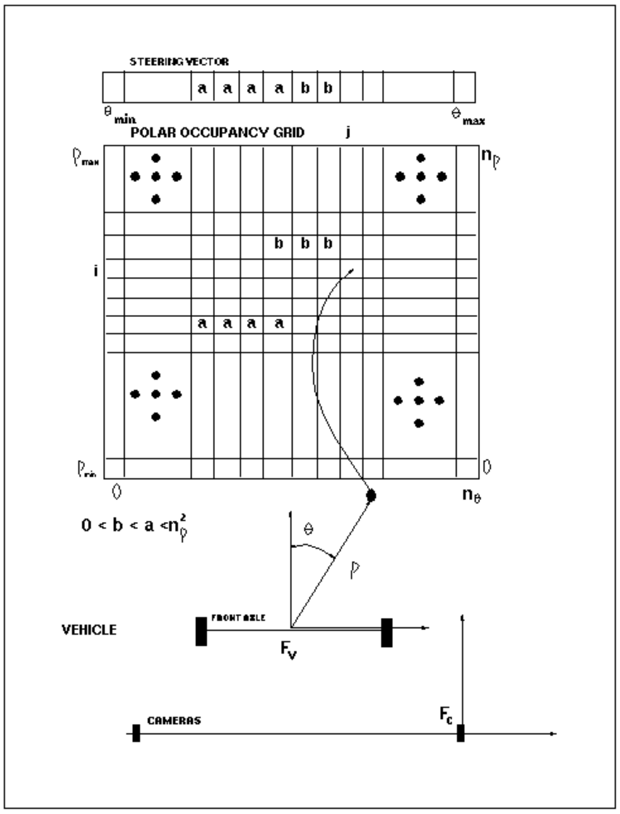


Figure 2: Steps involved in generating the steering vector from IOM frame F_c

3.3 Algorithm

Obstacles are detected and their positions are estimated in a simple four-step process:

1. **Calculation of disparity:** The epipolarity constraint restricts our search for corresponding pixels to one row. To find the pixel $R(i,j)$ in the right image corresponding to $L(i,k)$ in the left image, a simple $n \times n$ correlation mask is used.

2. **Filtering noise:** Because the matching process is so simple, we tend to get quite a few bad

matches which result in incorrect disparities. To eliminate this kind of salt and pepper noise, the popular constant-local-disparity constraint is exploited. This constraint enforces the disparities in a small window to have similar values. In the Marr-Poggio-Grimson approach [26, 27, 28], matching ambiguity is resolved so that the chosen value is close to the majority disparity of unambiguous points in the neighborhood. We make use of the concept in a slightly different way. The disparity computed in the previous step is believed if at least K pixels in its $m \times m$ neighborhood have the same computed disparity; otherwise the disparity is ignored.

3. **Thresholding the disparity:** The disparity $D(i,j)$ at a pixel $R(i,j)$ is compared with the expected disparity of the ground plane $G(i,j)$ and if $D(i,j) > G(i,j)$, $R(i,j)$ is regarded as a pixel corresponding to an obstacle.

4. **Projecting obstacles points onto the ground plane:** Each point that corresponds to an obstacle is projected onto the ground plane using the pixel coordinates (i,j) and the distance computed from the disparity. The output of the projection step is called the Instantaneous Obstacle Map(IOM).

Note that the correlation mask used in step 1 does not always give an unambiguous match, a pixel in left image may match equally well to several pixels in the right image. (This is particularly a problem on road surfaces which have very little texture at our working resolution.) In such cases the largest value of disparity is chosen. This is a conservative choice with regard to positive (protruding) obstacles since it selects the nearest possible position of the obstacle from the vehicle. Unfortunately, this is also the reason the algorithm works poorly on negative obstacles such as crevices.

4 Obstacle Avoidance

Obstacle avoidance is a continuous activity where each iteration of obstacle avoidance transforms the IOM into a suitable representation and then generates motor commands (speed and steering).

The IOM can be economically transformed into a *steering vector* representing a choice of steering directions. The range of the steering vector is the instantaneous turning extent of the vehicle. A value in each entry of this vector is interpreted as a steering *hindrance* in the associated direction. The direction to turn is simply inferred as the steering angle represented by a cell in the steering vector that has the lowest hindrance value, up to a threshold.

4.1 Computing the Steering Vector

The computation of a steering vector from an IOM is illustrated in Figure 2 and is described in the following steps:

1. **Transformation:** IOM points in a 2D cartesian frame $\mathcal{F}_c$ are transformed into a vehicle centered coordinate frame $\mathcal{F}_v$. In our case this frame is the intersection of the front axle and the longitudinal axis of the vehicle. The transformation in general can involve rotation, translation and scale. However for our experiments this matrix has been designed to involve only translation. Obstacle points in $\mathcal{F}_v$ are then expressed in polar coordinates (ρ, θ). If during this step an obstacle is detected that is closer than a minimum distance threshold ρ_{min}^{thrsh} (say) then the vehicle is brought to an immediate halt; otherwise, the next step is applied.

2. **Discretization and Relaxation:** Each obstacle point expressed in a polar form is discretized to represent a location (i,j) in a discrete polar occupancy grid (POG). The POG is similar to a C-space map [29] and coarsely encodes the possible configurations that the vehicle can exist in. An obstacle location $\mathcal{G}(i,j)$ in the POG $\mathcal{G}(0..n_\rho, 0..n_\theta)$ is assigned a value using the formula:
$$\mathcal{G}(i,j) = (n_\rho - i)^2 \quad (1)$$
and all other locations are assigned to 0. $n_\rho + 1$ and $n_\theta + 1$ are the number of rows and columns of $\mathcal{G}$ respectively. Notice that equation 1 assigns smaller values to obstacles that are further away from the vehicle. This assignment scheme serves to represent the hindrance associated with steering in a particular direction and will be discussed shortly. Further, note that the zero entries $\mathcal{G}$ represent the *free space* of a vehicle, albeit for one that has no dimensions. In order to account for the vehicle's dimensions each obstacle point is expanded. This expansion makes the assumption that the vehicle approaches obstacles almost head on and therefore it is sufficient to account only for the vehicle's width by expanding a point appropriately along the axis angle of $\mathcal{G}$.

3. **Steering vector computation:** Conceptually, a one dimensional steering vector $\mathcal{S}(0..n_\theta)$ is com-

(a) **Left Image** (b) **Right Image** (c) **Computed Disparity**

Figure 3: **Disparity Computed from a Stereo Image Pair.**

puted from $\mathcal{G}$ by assigning to an entry $\mathcal{S}(j)$ the value of the smallest non-zero row of $\mathcal{G}$ corresponding to the column j, if one exists, 0 otherwise. A cell j in the steering vector therefore contains the POG value of the closest obstacle along a particular steering direction represented by j.

In practice, $\mathcal{G}$ is never explicitly computed(no space allocation). $\mathcal{S}$ can be computed in $O(n)$ time for n obstacles in an IOM.

4.2 Computation of Steering and Speed

The cell $\mathcal{S}(j)$ is said to encode the *hindrance* associated with steering in the direction j in the sense that, the larger the value of $\mathcal{S}(j)$, the less confident we are of steering the vehicle in the direction represented by j. Then, given $\mathcal{S}$, the choice of steering direction is the corresponding cell j that has the smallest hindrance value below a predetermined threshold τ.

This rule is implemented as a one-dimensional search for a minimum value up to τ starting from the centered wheel position of the vehicle. The threshold τ sets the minimum distance from the vehicle up to which avoidance is effective and the current implementation incrementally lowers the effective avoidance distance, starting at n_ρ and stopping at a distance represented by τ. If no slot is found during this entire process the algorithm raises an exception and the vehicle is brought to an immediate halt.

The above method for computing a steering direction is greedy in the sense that it picks the steering direction as the angle θ_{ref} corresponding to the first satisfactory slot at the farthest possible avoidance horizon.

The second actuator command that the avoidance module modulates is the vehicle's speed. The reference speed v_{ref} sent to the actuator is governed by the following dynamics.

Let v_{max} be the maximum permissible speed and let t be the threshold at which a satisfactory steering angle was found. Then,

$$v_{ref} = w \cdot v_{max} \tag{2}$$

where

$$w = \left(w_1 \cdot \left(\frac{n_\rho - t}{n_\rho} \right)^2 + (1 - w_1) \cdot \left(\frac{|\theta_{ref}| - \theta_d}{\theta_d} \right)^2 \right)$$

and,

$$\theta_d = \begin{cases} |\theta_{max}| & \text{if } \theta_{ref} \geq 0 \\ |\theta_{min}| & \text{if } \theta_{ref} < 0 \end{cases}$$

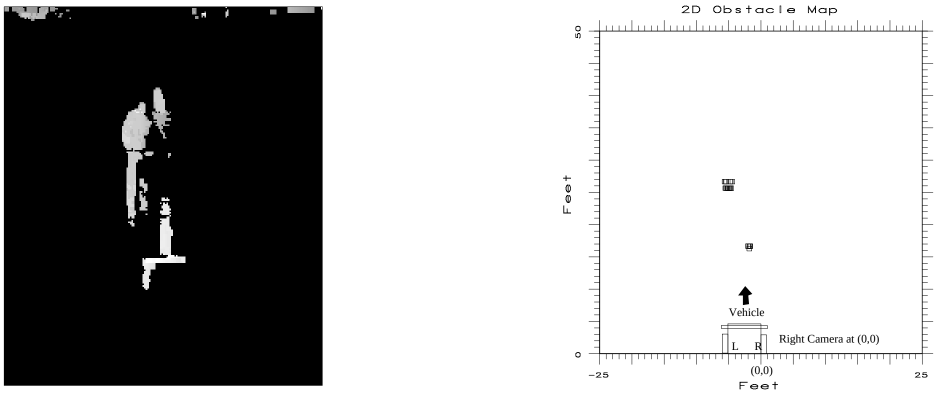
and w_1 is a weight empirically set between 0 and 1.

This metric weighs large turns and close obstacles relatively higher than small turns and distant obstacles and reduces speed when obstacles come close or when sharp turns are taken. Note that in the case when there is an obstacle closer than ρ_{min}^{thrsh} or when t exceeds τ the vehicle comes to an immediate halt due to the corresponding exceptions raised. In this case v_{ref} is never fed to the vehicle. Only legal values of θ_{ref} and v_{ref} are fed to a vehicle controller that operates the actuator commands based on these values.

5 Implementation Details and Experimental Results

5.1 Hardware

The vehicle has an SGI iris4D (used for avoidance) and SUN a Sparc-10 (used for detection) as general



(a) Result of thresholding disparity **(b) Instantaneous Obstacle Map**

Figure 4: **A 2D *Instantaneous Obstacle Map* from thresholded disparity.**

computing resources. DataCube's MV-20, DigiColor and ROI store form the image acquisition, low-level image processing and stereo matching subsystem. An MC68000 based vehicle controller controls steering, brake and throttle.

5.2 Detection

The stereo images are digitized to 256×240 pixels. The constants for the various algorithms (described earlier) are: mask size $N = 5$, threshold neighborhood size $M = 5 \times 5$, of which $K = 9$ must agree. The maximum disparity computed is 50. Due to hardware resource availability and timing constraints, the first and the third steps of the algorithm are implemented on the DataCube MV-20; the SGI host performs step 2 and step 4. The entire process executes at 2Hz, resulting in safe speeds of up to 7MPH. The Instantaneous Obstacle Map produced has the right stereo camera as the origin. Figures 3 and 4 illustrate various intermediate results and the Instantaneous Obstacle Map produced when a typical stereo image pair is processed. The left stereo image 3(a) and right stereo image 3(b) are clipped to the size of the disparity image 3(c). Figure 4(a) is the result of thresholding 3(c) against EGP disparity. Figure 4(b) is the Instantaneous Obstacle Map produced as a result of projecting points corresponding to obstacles on to the ground plane. Note that the "shadows" that appear to the right of the obstacles in Figure 3(c) are the result of correspondence mismatches of ground plane points that are visible from the left camera but occluded by obstacles in the right; these are eliminated after thresholding (see Figure 4(a)).

5.3 Avoidance

The constants and thresholds used in the obstacle avoidance components of the system are: $\rho_{min} = 0$,

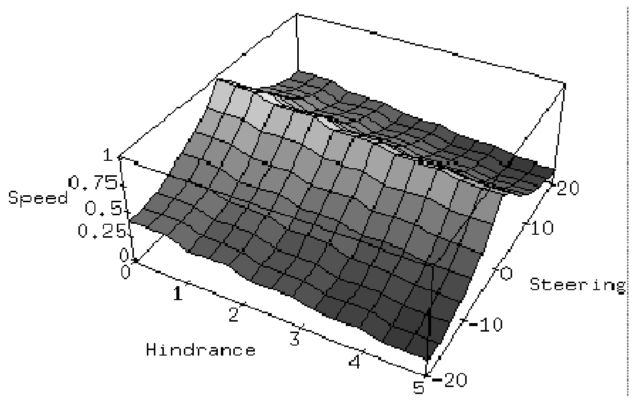


Figure 5: v_{ref} **as a function of steering and hindrance**

$\rho_{max} = 100ft.$ (determined by the effective range of detection), $\theta_{min} = -20^o$, $\theta_{max} = 20^o$ (the steering limits of the vehicle), n_ρ and n_θ were chosen to be 10 and 40 respectively (providing a $10ft$x1^o resolution in $\mathcal{G}$). $v_{max} = 10.0fps$, $\tau = 5$ (representing 50ft.) and $w_1 = 0.6$. Figure 5 illustrates the surface for equation 4.2 with $v_{max} = 1.0$. This surface shows the values of v_{ref} against varying hindrance values $t < \tau$ and steering (in degrees) corresponding to $\theta_{min} < \theta_{ref} < \theta_{max}$. Note that $v_{ref} = 0$ for all other values of θ_{ref} and t.

Figures 6 illustrates a sampled sequence of an obstacle avoidance run. This figure must be read in a left to right. This experiment involves placing both cones and a human obstacle in a configuration clearly observable in the last picture of the figure. The path traced out by the vehicle is also observable as tracks

Figure 6: **An obstacle avoidance sequence**

on the grass in this picture. The distortion observed in the top lines of the images in Figure 6 is an artifact of the video recorder playback (the data given to the algorithm was not distorted).

6 Conclusion

A fast non-hierarchical stereo correspondence algorithm producing sparse disparity information turns out to be effective for obstacle avoidance in a common outdoor setting. The obstacle avoidance module is reflexive in the sense that it is data driven and opportunistic in generating motor commands. The algorithm is relatively insensitive to positional uncertainties of obstacles as well as the inaccuracies of the actuator servos. The algorithm is stable, simple, fast and practical to implement. We realize that our approach may fail, but we also observe that our technique performs very well in most out door scenarios, especially with sparse obstacle sets. A well engineered system has to make appropriate assumptions so that a difficult task becomes amenable in real world, real time settings using available technology.

7 Limitations and Future Work

Relaxation of the flat ground assumption would mean a substantial extension of the obstacle detection system. Automating the representation of the ground plane by computing the equation of the plane on the fly would help in situations when the ground slope changes suddenly. It would also help in situations where the stance of the stereo system is not fixed relative to the ground plane but can be actively controlled by the agent. Other improvements can be expected by relaxing the Epipolarity and the Identical camera/Digitizers assumptions.

Experiments with tracking obstacles over successive IOMs have been promising; the system is being extended to generate local maps by merging successive IOMs.

The obstacle avoidance system developed in this paper cannot by itself perform a task such as to move from point A to point B in an outdoor scenario. However, we have conducted experiments in which the obstacle avoidance system coexists with other modules including a compass-based heading generator and a road follower [30]. Desirable behavior has been observed to emerge from a composition of these modules (for example, move along a particular heading while avoiding obstacles or follow a road while avoiding obstacles). Our obstacle avoidance system has a limitation in that it does not account for the non-holonomic constraints introduced by the vehicle. It would be interesting to incorporate non-holonomic constraints when computing the speed and steering values.

Acknowledgements

We are grateful to Edward Riseman for his guidance and Richard Weiss for his ideas and feedback during the design and implementation of the system. We gratefully acknowledge the help of Bob Heller and Jonathan Lim for answering many of our questions about system support utilities, Gokhan Kutlu, Sashi Buluswar, Alan Boulanger and Greg Glosser for carrying out many test runs of the system on the vehicle, and, Fred Weiss for his excellent video production skills.

References

[1] J. H. M.J. Daily and K. Reiser, "Detecting obstacles in range imagery," *Proc. of [ARPA] Image Understanding Workshop*, pp. 87–97, 1987.

[2] P. Veatch and L. Davis, "Efficient algorithms for obstacle detection using range data," *CVGIP 50*, pp. 50–74, 1990.

[3] B. Barshan and R. Kuc, "A bat-like sonar system for obstacle localization," *IEEE Transactions on Systems, Man, and Cybernetics 22*, pp. 636–646, 1992.

[4] J. Borenstein and Y. Koren, "Obstacle avoidance with ultrasonic sensors," *IEEE J. Robotics Automation 4*, pp. 213–218, 1988.

[5] N. Ancona, "A fast obstacle detection method based on optical flow," *Proc. of European Conference on Computer Vision*, pp. 267–271, 1992.

[6] W. Enkelmann, "Obstacle detection by evaluation of optical flow fields from image sequence," *Image and Vision Computing 9*, pp. 160–168, 1991.

[7] M. H. G.S. Young, T.H. Hong and A. Yang, "Obstacle detection for a vehicle using optical flow," *SAE Intelligent Vehicle. IEEE*, 1992.

[8] R. Nelson and J. Aloimonos, "Using flow field divergence for obstacle avoidance: towards qualitative vision," *Proc. of International Conference on Computer Vision*, pp. 188–196, 1988.

[9] Y. Yagi and M. Yachida, "Real-time generation of environmental map and obstacle avoidance using omnidirectional image sensor with conic mirror," *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 160–165, 1991.

[10] H. Sawhney and A. Hanson, "Affine trackability aids obstacle detection," *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 418–424, 1992.

[11] H. Sawhney and A. Hanson, "Tracking detection and 3d representation of potential obstacles using affine constraints," *Proc. of [ARPA] Image Understanding Workshop*, pp. 1009–1017, 1992.

[12] R. Z. Zhang and A. Hanson, "Qualitative obstacle detection," *Tech. Report COMPSI TR94-20*, 1994.

[13] R. C. B. H. H. Baker and M. J. Hannah, "The JISCT stereo evaluation," *Proc. of [ARPA] Image Understanding Workshop*, 1993.

[14] B. Ross, "A practical stereo vision system"," *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 148–153, 1993.

[15] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *International Journal of Robotics Research 5(1)*, pp. 90–99, 1986.

[16] J. Kim and P. Khosla, "Real-time obstacle avoidance using harmonic potential functions," *Tech. Report COMPSI T-RA 8*, pp. 338–349, 1992.

[17] B. Krogh, "A generalized potential field approach to obstacle avoidance," *SME Conf. Proc. Robotics Research: The Next Five Years and Beyond,(ed),Bethlehem, PA, SME*, 1984.

[18] C. I. Connolly, "Harmonic control," *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, 1992.

[19] C. I. Connolly and R. A. Grupen, "Applications of harmonic functions to robotics," *Tech. Report UM-CS-1992-012*, 1992.

[20] R. C. Arkin, "Motor schema based navigation for a mobile robot," *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, 1987.

[21] D. W. Payton, "An architecture for reflexive autonomous vehicle control," *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, 1986.

[22] S. S. Ravela, "A survey of reactivity," *Tech. Report COMPSCI TR-92-61, UMASS*, 1992.

[23] M. S. B. Thomas C. Wikman and W. S. Newman, "Reflexive collision avoidance: A generalized approach," *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, 1993.

[24] W. S. Newman, "Automatic obstacle avoidance at high speeds via reflex control," *PhD thesis, Massachusetts Institute of Technology, Department of Mechanical Engineering*, 1987.

[25] R. M. Haralick and L. G. Shapiro, "Computer and robot vision Vol. I and II," *Addison-Wesley Pub. Co.*, 1993.

[26] D. Marr and T. Poggio, "A theory of human stereo vision," *Roy. Soc. London, vol B 204*, pp. 301–328, 1979.

[27] D. Marr, "Vision," *Freeman*, 1982.

[28] W. E. L. Grimson, "From images to surfaces," *MIT press*, 1981.

[29] T. Lozano-Perez, "Spatial planning: A configuration space approach," *IEEE Trans. Computers, C-32, NO. 2*, pp. 108–120, 1983.

[30] D. A. Pomerleau, "Neuralnetwork perception for mobile robot guidance," *Kluer Academic Publishing, Boston*, 1983.