A High-Performance Vision System for Obstacle Detection

Todd Williamson

CMU-RI-TR-97-39

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

September, 1997

This research was sponsored by the collaborative agreement between Toyota Motor Corporation and Carnegie Mellon University.

Abstract

I have developed a method for obstacle detection in a highway environment using the CMU Video Rate Multi-Baseline Stereo Machine. One key feature of this method is the computation and output of a confidence measure at each pixel, so that regions of the scene with low reliability can be recognized and filtered out. This method has proven to be capable of detecting 30cm high obstacles at distances of close to 100m. Unfortunately, the system requires significant of post-processing of the output data from the CMU stereo machine, and therefore cannot be run in real-time. Additionally, the design of the CMU stereo machine imposes several limitations on the system, both computationally (limited resolution, speed, and programmability) and practically (large size, cost, power consumption).

I propose to construct a system, consisting of both hardware and software, to perform this obstacle detection task in real-time. In the process, I will develop a general-purpose stereo vision machine that improves upon the state-of-the-art. Furthermore, I will show that the system can perform obstacle detection in real-world environments and thus improve vehicle safety.

Table of Contents

1 Introduction

One important problem in the field of autonomous navigation systems is that of obstacle detection. In the broadest terms, an obstacle is any region in the environment of the vehicle that should be avoided. In the Automated Highway Systems domain that is considered in this document, typical obstacles are other vehicles and debris that has fallen onto the road surface. In some situations, it is also convenient to consider other objects, such as guard rails, as obstacles. These obstacles need to be detected at large distances (on the order of 50-100m) and with a high cycle rate (on the order of 10 Hz or better) to be useful for improving vehicle safety on the highway.

In attacking the problem of detecting such obstacles, I have used the CMU Video-Rate Multi-Baseline Stereo Machine. This machine has some unique features which inspired me to develop a new calibration method. The new calibration scheme extends the abilities of the machine to include the detection of the deviation of objects from the expected planar environment. Using this algorithm to calibrate the stereo machine, and with some further post-processing of the output of the machine, I have been able to detect 30cm high obstacles at ranges approaching 100m.

Unfortunately, this system has many limitations. Currently the post-processing algorithms do not run in real-time, despite the fact that the output of the stereo machine is available at around 6 Hz. A large number of other limitations come from the stereo machine itself, including limited source image resolution, a limited number of possible disparity levels to search, limited calibration accuracy, and limited speed. Additionally, a number of factors render the stereo machine inappropriate for the AHS domain, including its size, power consumption, and high price.

Given these limitations, and the fact that no other stereo vision processing hardware with suitable performance is commercially available, it seems natural to propose to build a better stereo machine to use for obstacle detection in the AHS domain.

To build the best possible vision-based obstacle detection system, it will be necessary to address the following issues:

- hardware that is fast, while being modular and easily upgradable
- low power consumption and small size
- low cost
- a stereo vision algorithm specially tailored to the obstacle detection domain
- high-speed implementation in software

In investigating how it might be best to go about this, I have found that custom computing hardware, no matter how clever the design, is soon overtaken by the power of general-purpose CPUs. Therefore I propose to build a modular vision machine, with arbitrarily expandable (and upgradable) hardware, and the accompanying software, to perform the necessary computation for stereo vision based obstacle detection.

The remainder of this paper is organized as follows. In Section 2 I will discuss related work in vision-based obstacle recognition, other obstacle detection systems, and stereo vision machines. In Section 3 I will explain in detail how I use the CMU stereo machine for obstacle detection. In Section 4 I will discuss my proposed research, and in Sections 5 and 6 I will show my research schedule and thesis contributions.

2 Related Work

Since this thesis touches on several areas, there is a large amount of related work. I have divided the discussion here into three areas: vision-based (including stereo) obstacle detection, obstacle detection using other sensors, and other stereo vision machines.

2.1 Vision-based obstacle detection

There has been a lot of research on vision-based obstacle detection. Much of this work falls into two categories: indoor static obstacles [1][3][7], outdoor cross-country environment with static obstacles [15][18]. Since these systems are generally more concerned with detecting and navigating along a safe path than with speed, it is sufficient to detect obstacles at relatively close range and with slow cycle times. Another class of systems deal with a highway environment and moving obstacles [4][13][14]. Most of these methods attempt to detect large obstacles such as other vehicles, but do not address the problems of smaller static obstacles.

One interesting system based on stereo vision using 1-D cameras is described in [5][6]. They have demonstrated detecting a pedestrian crossing in front of the vehicle at short range.

2.2 Obstacle Detection with other Sensors

A variety of other sensors have been used for obstacle detection. Radar and ladar (and to a lesser extent sonar) systems have been used for obstacle detection research for a number of years. All of these systems use a limited spectrum in which some possible obstacles may appear transparent. They are also susceptible to multi-path reflection effects. Another concern for AHS applications is how to prevent interference between the active signals from multiple vehicles travelling on the same roadway. While vision-based systems have their share of problems (chief among them being computational complexity), the low cost of camera systems makes them an attractive option.

2.3 Stereo Vision Machines

A number of groups have developed real-time stereo vision processing systems. In addition to the CMU system [11], groups at SRI [12], Cornell [22], and Point Grey Research [21] among others have real-time stereo vision systems. While all of these systems are very competitive with the CMU machine in terms of performance and cost, they all lack one or more of the following useful features present in our machine:

- multibaseline processing for increased accuracy
- the ability to use any arbitrary camera geometry - in particular long baselines and long focal length lenses
- the ability to calibrate the machine for non-standard epipolar geometry

3 Previous Work

During the past two semesters, I have developed a new algorithm for detecting obstacles using the CMU stereo machine and a modified epipolar geometry. In the following sections I will describe briefly the how CMU stereo machine works, a modified epipolar geometry and accompanying calibration scheme for obstacle detection, and some ideas about how the machine could be improved.

3.1 The CMU Video-Rate Multi-Baseline Stereo Machine

The CMU stereo machine consists of a number of custom-built 9U VME boards connected together in a system as shown in Figure 1. The system is described in more detail in [11].

The algorithm works by first digitizing the images from each of the cameras (up to 6 in the current design). The image is then passed through an 11x11 LOG (Laplacian of Gausssian) filter. The resulting data is then compressed to 4 bits by a non-linear weighting function. All subsequent processing is done with 4-bit pixel data.

The 4-bit LOG filtered data is then passed on to the geometry compensation unit. This unit is of particular importance because it performs a very general transformation on each of the input images, the purpose of

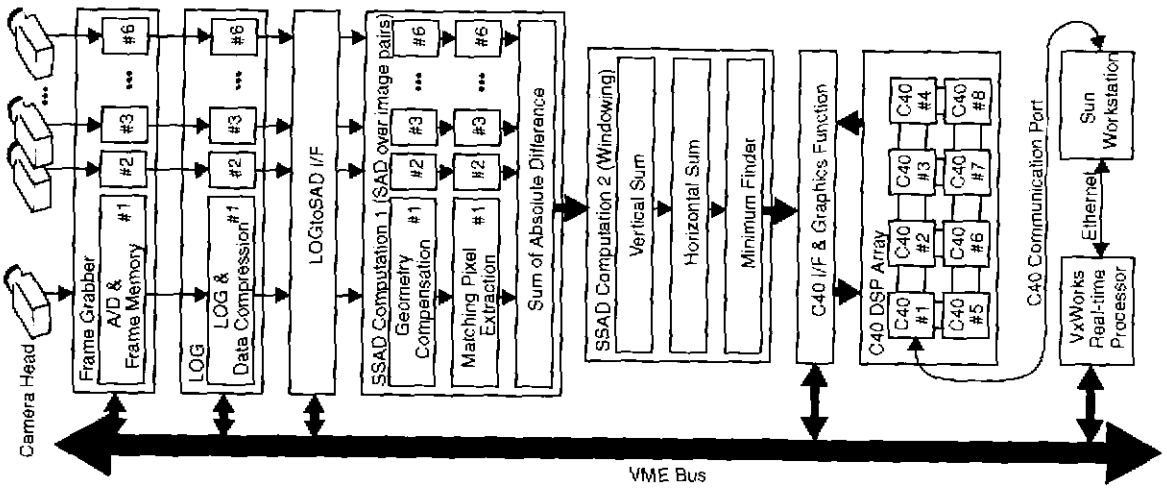

Figure 1: Architecture of the CMU Stereo Machine

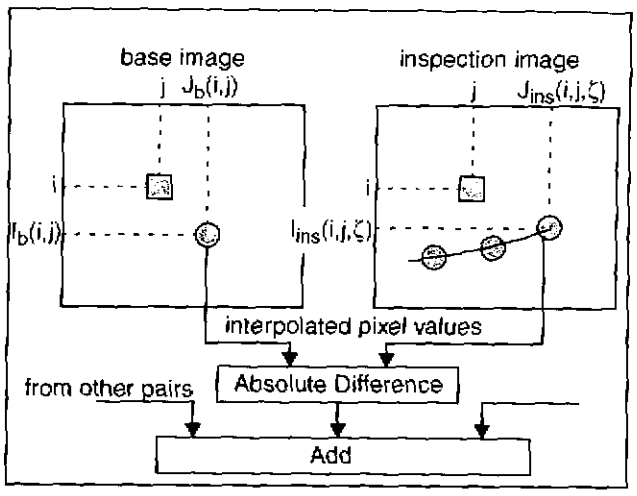

Figure 2: Geometry compensation

which is to rectify the images before performing the SAD computation which comes next. For each postulated distance from the cameras, all of the images are effectively reprojected as if all objects in the scene were at that distance. This reprojection is accomplished via a 3-dimensional lookup table whose inputs are the postulated disparity ζ and the image coordinates (i,j), and which outputs the corresponding image coordinate for each camera ($I(i,j,\zeta)$, $J(i,j,\zeta)$). The output image coordinates are given to sub-pixel accuracy (in 1/16ths of a pixel), and the four adjacent pixels are linearly interpolated to produce the reprojected result pixel. Note that the lookup table can actually contain any values whatsoever, so it is possible to correct for lens distortion, or to operate with one camera upside down, or using cameras with different focal length lenses.

In the next stage, the absolute value of difference (AD) is performed pixel-by-pixel for the base camera (camera #1) paired with each of the other cameras. The results the AD computation are summed over each camera, resulting in a SAD (sum of absolute difference) value for each pixel for each disparity level.

The resulting SAD values are now smoothed by summing over a window whose size is programmable from 1x1 to 13x13, producing the SSAD. In the final stage, for each pixel, the disparity level with the minimum

SSAD value is found, and the SSAD values of the minimum and its neighbors are sent to the C40 DSP processing board, where the disparity levels can be interpolated for higher accuracy.

3.2 The Importance of Multi-Baseline Techniques

My experiments with our three camera system on the NavLab II have convinced me of the usefulness of multiple baselines. Multi-baseline techniques provide at least three advantages:

1. disambiguation of repeating patterns
2. sensitivity to image texture in multiple directions
3. improved accuracy, since all of the cameras contribute to the result

Since our cameras are currently mounted in a horizontal line, we don't see the advantage of #2 yet.

Experiments with disconnecting individual cameras from the stereo machine while it is operating show that using all three cameras in tandem produces significantly better results than when limited to any pair.

3.3 Stereo Calibration

In traditional stereo vision algorithms, the first stage of the computation is to compute a disparity value at each pixel. For any pixel in one image, there is a single location where it would project into another image when observing a point infinitely far away. The distance from that location at which the point actually appears is called the disparity. In this scenario, the planes of constant disparity are parallel to the reference image plane, and the distance to a point can be easily computed from its disparity.

This model does not necessarily map well onto the situation where the object to be observed is a planar surface that is not parallel to the camera plane. In this case, pixels from different rows of the image are actually at different distances from the camera. Since in the AHS domain we expect to observe relatively flat roads with the occasional obstacle, it makes sense to redefine disparity in such a way that a constant disparity value represents a single ground plane.

In the following sections, I will first describe the traditional calibration method for the CMU stereo machine, and then describe two modifications of the algorithm to handle road geometry better.

3.3.1 Traditional calibration scheme

The camera calibration problem for the CMU stereo machine consists primarily of measuring or computing the values that go into the lookup tables in the geometry compensation circuit. The stereo machine group has investigated a number of ways to do this, the most successful of which uses ideas from weakly calibrated stereo research [18][17].

It is a well-known result that the homogeneous coordinates of corresponding points in two images of the same planar surface are related by a homography matrix (assuming a pinhole camera model) [19]. Furthermore, given homography matrices for two parallel planes H_{z_1} and H_{z_2} at distances from the camera z_1 and z_2 respectively, the homography matrix for any other parallel plane can be determined by simply interpolating the two homographies:

$$H_z = \frac{\frac{1}{z} - \frac{1}{z_2}}{\frac{1}{z_1} - \frac{1}{z_2}} \cdot H_{z_1} + \frac{\frac{1}{z_1} - \frac{1}{z}}{\frac{1}{z_1} - \frac{1}{z_2}} \cdot H_{z_2}$$

This result has been used extensively to calibrate the stereo machine at CMU. The stereo machine group has written an application that asks the user to choose 4 pairs of corresponding points (the minimum needed to compute a homography matrix) in two images, and a region of the image to match. The program then does a nonlinear optimization to match the two regions by finding the H matrix that minimizes the SSD between them. If we use this program with, for example, images of a wall 10m in front of the vehicle and 100m in front of the vehicle, then we can interpolate the homography matrices thus obtained, and generate the look-up tables for the geometry compensation hardware to compute distance in front of the vehicle.

This algorithm effectively matches square templates from one image to the other. This is the proper thing to do for surfaces that are parallel to the camera image plane, and in those cases it produces very accurate results. In the highway environment, however, we expect the dominant portion of the image to be the road surface itself. When viewing the road surface, the distance to points in the image can vary greatly from one image row to the next, thus we expect a square region in the left image to map to a parallelogram-shaped region in the right image.

3.3.2 Road plane calibration schemes

In order to detect obstacles in front of the vehicle, my idea was to use the same algorithm, but with images taken in a slightly different manner. The first step is to treat the ground as if it were a planar surface. One possible second step could be to construct another plane parallel to the ground plane, so that the two planes could be interpolated to generate a "height" map. While this approach works reasonably well, it was very difficult to find a plane that is parallel to the ground, but at a different height. Instead, if we could take an image where all of the points were effectively infinitely far away (compared to the distance between cameras), then it would be possible to compute the homography for the "plane at infinity". In this case ($z_2=\infty$), the equation simplifies to:

$$H_z = \frac{z_1}{z} H_{z_1} + \left(1 - \frac{z_1}{z}\right) H_{z_2}$$

To get homographies for H_∞, I pointed the cameras at the sky near the horizon. Using this method, I was able to generate the output image shown in Figure 3.

The results of simple height calibration were pretty good, but as the image approaches the horizon, the results get more and more noisy. This is due to the fact that the horizon line for parallel planes (the planes of constant height above the ground) is always in the same location. Furthermore, for rows near the horizon line all different height levels appear at almost the same pixel location, at the bottom of the image different height levels are very far apart in the image, and above the horizon, the disparity is negative, representing the fact that the plane is actually behind the camera (see Figure 4).

Obviously, this type of height computation has too many problems to be really practical. But is there a method that captures the same template shape characteristics, while keeping the disparity levels distinguishable from each other? The answer seems to be yes, if we use the homography for the ground plane to determine the initial disparity levels, and then move in one pixel increments along the epipolar line. For an ideal camera configuration, the planes of constant "disparity" for this method are as shown in Figure 5.

This sampling method has a number of good properties:

- the template shape is correct for objects that lie along one of these planes
- the different planes are all distinct everywhere in the image

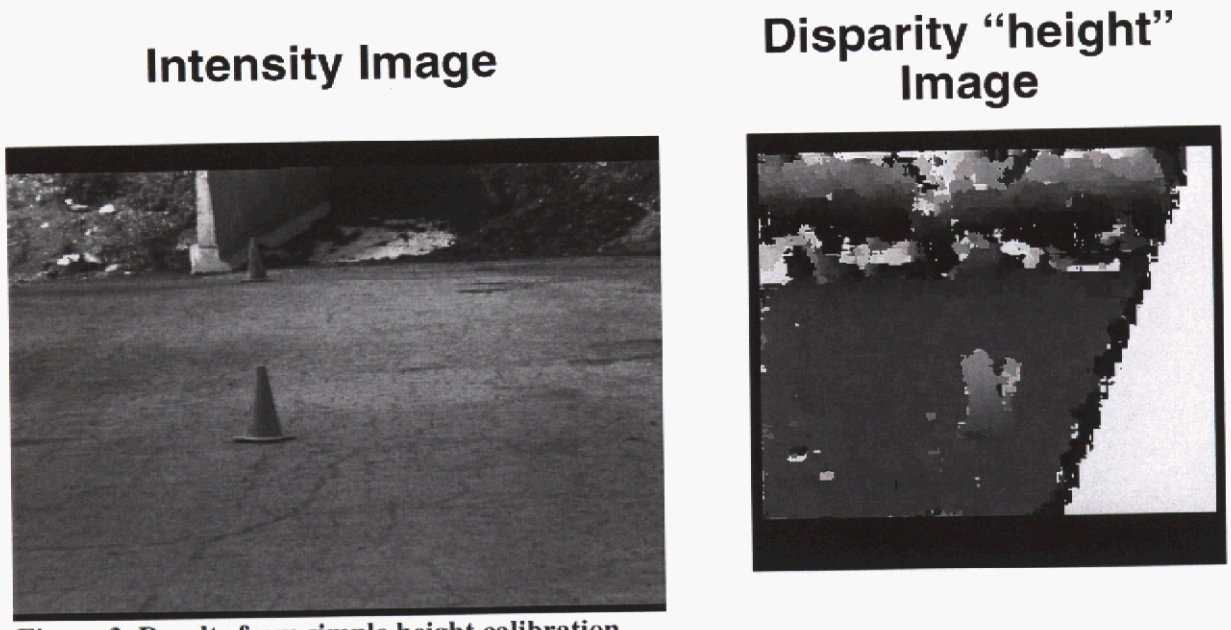

Figure 3: Results from simple height calibration

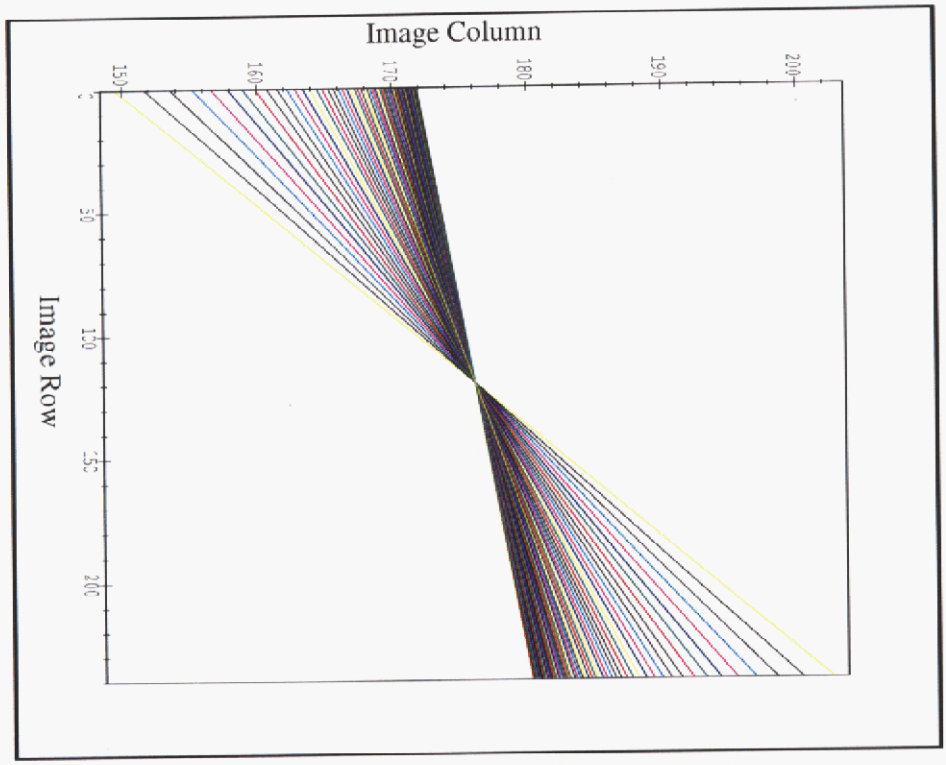

Figure 4: Position of matching pixel in right image for the pixels in column 175 of the left image, one line for each possible height.

- the allocation of the limited number of disparity search levels is optimized for road geometry (since we don't expect for the pixels at the bottom of the image to be 200m away, nor do we expect objects at the top of the image to be close).

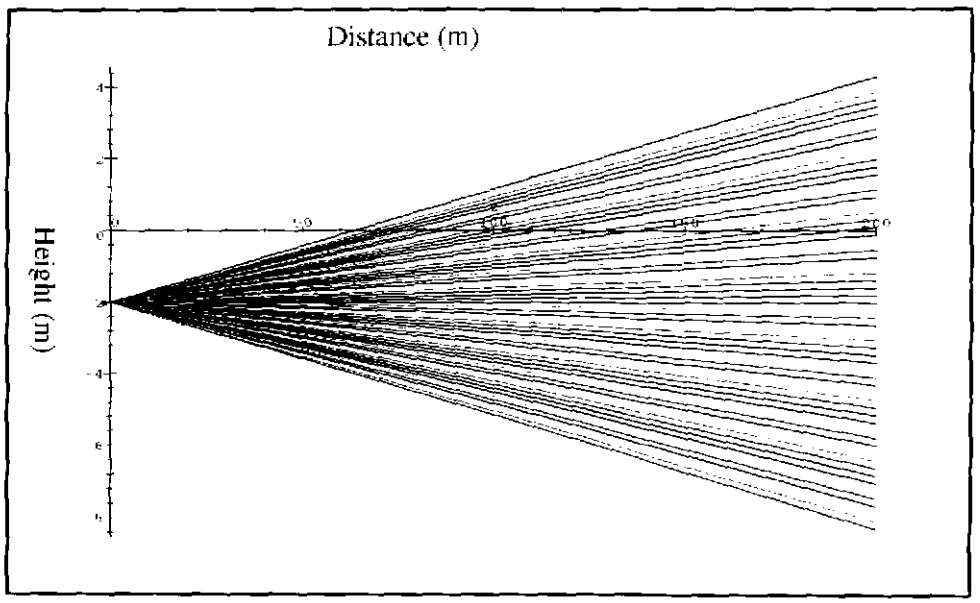

Figure 5: Planes of constant "disparity" for the second calibration method. Parameters are 1m baseline, 35mm lenses, 1/2" CCD, cameras aligned perfectly and 2m above the ground

- even if the ground plane does not match up with one of the constant disparity planes exactly (because of the curvature of the road and the fact that the bottom of the image may in fact be very far away from the vehicle), a planar road surface must also be planar in (row,column,disparity) space.

3.4 Confidence Measurement

Even with a good camera calibration tailored to the task, there will always be regions of the world that lack sufficient texture to support a reliable depth measurement. One of the most common ways to measure the reliability of the output of stereo matching is to look at the SSAD curve as a function of disparity level, and to judge the "sharpness" of the minimum.

Unfortunately, in the CMU stereo machine, the image is passed through a LOG filter, the output of which is sensitive to the second derivative of the image intensity. This computation is very susceptible to noise in the input image, and tends to amplify small image gradients. Thus the sharpness of the SSAD minimum can indicate whether or not there was any signal in the LOG filtered data, but it can't indicate whether that signal was reliable or just an amplification of noise.

Therefore, a separate confidence metric is necessary to determine the confidence of the output of the LOG filter. For this purpose, I have chosen to use the absolute value of the derivative of a gaussian filter, taken in the direction of the epipolar line in the image, and with the same parameters as the LOG filter. This confidence measure does the right thing intuitively - it roughly measures the amount of texture in the direction of the disparity search. Mathies [15] derives a similar metric for stereo matching.

Currently, this computation is done off-line on a UNIX workstation, and is one of the most time-intensive parts of my obstacle detection algorithm.

3.5 Further Post-Processing

The output from the CMU stereo machine using the above calibration method is a 2-dimensional image, where the value of a pixel represents which of the planes resulted in the best match. From this output we must decide which parts of the image represent obstacles and which parts represent road.

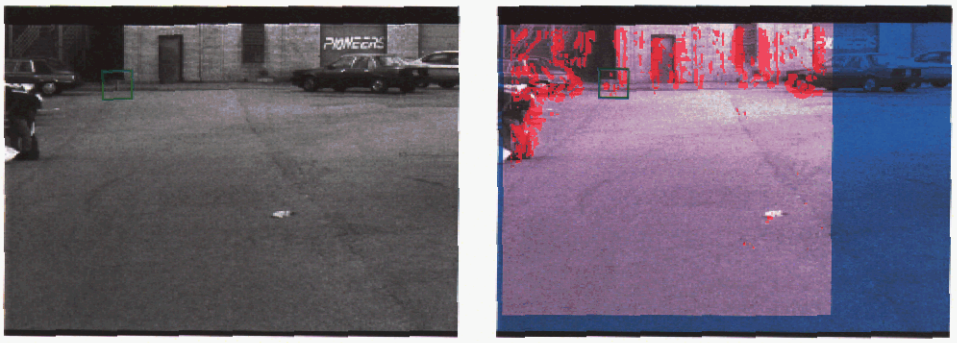

Figure 6: An image of a parking lot, and the obstacle image. Obstacles identified are marked in red. The blue-ish region was not processed.

Since planar surfaces in the world map to planar surfaces in this image representation, the first step is to determine the ground plane. For this I use a simple Hough transform on the parameters of the plane. Of a set of possible candidate planes, the one for which the majority of pixels votes is assumed to be the ground plane. A method of this sort results in better rejection of outlier points than a simple least-squares fit.

Once the ground plane has been determined, obstacles can be detected by measuring the difference between the output of the stereo machine and the best-fit plane, and thresholding. Further thresholding based on confidence produces an obstacle map such as that shown in Figure 6. Note the traffic cone (in the green box) whose edges are detected by this algorithm. This traffic cone is approximately 100m ahead of the vehicle, and is only about 40cm high. Note also that the edges are not very well defined due to the 11x11 pixel smoothing window.

3.6 Disadvantages of the current system

The system I have described so far is very promising, but it still has a number of problems. The post-processing is very CPU-intensive, taking about 20 seconds per frame on a Sparcstation 5. Ideally, this post-processing could be moved onto the stereo machine itself.

Additionally, the CMU stereo machine some drawbacks for use in the AHS domain:

- too big - it uses a 9U VME cage
- too power-hungry - it draws over 400W
- too expensive - our prototypes cost around $100K each
- limited image size and number of possible disparity levels

It would also be nice to be able to freely program which portion of the image is used for computation as well as the number and range of candidate planes to be searched at each point as the machine is operating, for example to track an obstacle as it moves across the field of view. This sort of capability is not currently available on our machine.

4 Proposed Research

My proposed research falls generally into two categories:

- Developing a new stereo machine
- Demonstrating its use for high-relibility long-range obstacle detection

I intend to perform both parts of the research in parallel, and at my thesis defense, I should be able to show a complete, functioning obstacle detection system based on the new machine.

4.1 Developing a new Stereo Machine

I propose to solve each of the problems that I listed in Section 3.6, and hopefully improve performance as well, by developing a new stereo machine.

In investigating how a new stereo machine can be built, I have found that the performance of general-purpose CPUs tends to improve exponentially, at any given time operating with clock rates that are several times faster than custom hardware which is available (such as FPGA technology). This fact suggests that the best approach to building a stereo machine might be to build a modular system with replaceable CPU modules.

In fact, there is a project at CMU, the Reconfigurable Vision Machine (RVM) project, that has been producing modular vision processing systems. The idea is that there is a baseboard with a number of slots, each of which has communication ports that can be wired arbitrarily. Each slot holds a vision processing module, which can be a DSP processor card, a digitizer, a convolver, etc.

I have been working together with the RVM project to evaluate a number of very fast CPUs that have come out recently. Our conclusion was that the Philips TriMedia TM-1000 is probably the fastest chip for vision processing at the moment, although the TI C60 is very close. The RVM lab is currently designing a CPU board for their machine based on this CPU.

My plan is to work with the RVM project to build any necessary hardware to handle the special communication requirements of the stereo algorithm. At the same time, I will be developing the stereo machine software in such a way that it will work with any number of CPU modules, and with programmable image size and number of disparity levels.

One important feature that the current stereo machine is lacking is some measure of the confidence level of the output. I intend to implement the new stereo machine in such a way that the confidence measure is available in real-time. One area that deserves further attention is exactly what formula is appropriate for computing this confidence measure. My approach of using a tuned derivative of gaussian filter, while it makes intuitive sense, was simply an ad-hoc procedure. I would like to develop a more principled formula that combines elements of both image texture and the shape of the SSAD curve.

In order to get the stereo algorithm to run on a general-purpose CPU at reasonable speeds, a great deal of algorithmic optimizations will have to be made. In my experimentation with the Philips Trimedia simulator, I have found that it is possible to get large speedups (as much as 80x!) by incrementally adapting the algorithm to fit the types of operations that the CPU can do quickly. In the processes, it is necessary to make trade-offs: for instance, the fastest way to perform the geometry compensation would be to use a large lookup table. Unfortunately, the size of this lookup table would be around 20 megabytes per baseline. Since the memory for the lookup table would have to be very fast, and local to each processor in the system, this is not a viable solution. On the other hand, the most compact representation is the 8 floating-point numbers in the homography matrix, but since the computation of image coordinates using the homography matrix involves a floating-point division, it is very slow. Thus, some intermediate representation is necessary.

4.2 Improving and Validating the Obstacle Detection Algorithm

Further experiments need to be done to validate the functioning of this obstacle detection algorithm in a variety of circumstances. I need to test the detectibility of various obstacles in different lighting conditions, at different distances, on different road surfaces, and finally, under different weather conditions and at night.

Currently, I do not attempt to localize the detected obstacles in the 3-D. In order to be useful in a system, this needs to be done so that the vehicle can take appropriate action. Also, no handling of negative obstacles

(e.g. potholes) is currently done, though it is a minor modification to detect any visible regions of this sort of obstacle (though by their very nature, such regions must be small and therefore may not be detectable as obstacles).

Additionally, I want to experiment with different spatial configurations of the three cameras to determine the effect on obstacle detection accuracy. I predict that accuracy will be significantly improved by arranging the cameras in a triangular configuration.

5 Conclusions

The goal of this thesis is to build the best possible vision-based obstacle detection system. In order to do that, I intend to build a new stereo vision machine which will be better than every other contemporary system in at least one of the following ways:

- speed
- modularity (both expandability and upgradability)
- cost
- size
- power consumption
- accuracy

Additionally, I will develop an obstacle detection scheme using this machine which will be better than currently available systems in the following ways:

- able to detect smaller obstacles (30cm or less) at greater distances (80-100m)
- available confidence metric to judge the reliability of the output
- real-time, at least 10fps

6 Schedule

Summer 1997	complete thesis proposal RVM lab begins board design
Fall 1997	validation experiments experiment with different camera configurations design stereo machine software in simulation design any necessary additional hardware RVM lab finishes board design
Spring 1998	assemble and test hardware finish developing and testing software show operation of stereo machine
Summer 1998	verify complete system operation implement other vision algorithms data collection
Fall 1998	write thesis defend

7 Contributions

I expect to demonstrate the following:

- a stereo vision machine that is cheap, compact, relatively low power, and completely modular, with processing speed determined simply by the number of processors available
- a method for intelligently tailoring the search space for multibaseline stereo to the obstacle detection problem.
- a obstacle detection system based on the above machine and method, capable of detecting 30cm (1ft) obstacles at distances of up to 80m reliably, and smaller objects at shorter ranges.
- a modular, automatically configured software system to run on the machine
- real-time computation of confidence level of disparity values

8 Acknowledgements

This research was sponsored by the collaborative agreement between Toyota Motor Corporation and Carnegie Mellon University.

9 Bibliography

[1] N. Ancona, "A Fast Obstacle Detection Method based on Optical Flow," *Proceedings of the European Conference on Computer Vision (ECCV '92)*, 1992.

[2] M. Betke, E. Haritaoglu, and L. Davis. "Multiple Vehicle Detection and Tracking in Hard Real-Time." *Proceedings of the Intelligent Vehicles '96 Symposium*, 1996.

[3] S. Bohrer, M. Brauckmann, and W. von Seelen. "Visual Obstacle Detection by a Geometrically Simplified Optical Flow Approach." *Proceedings of the 10th European Conference on Artificial Intelligence (ECAI '92)*, 1992.

[4] S. Bohrer, T. Zielke, and V. Freiburg. "An Integrated Obstacle Detection Framework for Intelligent Cruise Control on Motorways," *Proceedings of the Intelligent Vehicles '95 Symposium*, 1995.

[5] J.-L. Bruyelle and J.-G. Postaire. "Direct Range Measurement by Linear Stereovision for Real-Time Obstacle Detection in Road Traffic." *Proceedings of the International Conference on Intelligent Autonomous Systems (IAS-3)*, 1993.

[6] J.-C. Burie and J.-G. Postaire. "Enhancement of the Road Safety with a Stereovision System Based on Linear Cameras," *Proceedings of the Intelligent Vehicles '96 Symposium*, 1996.

[7] S. Cornell, J. Porrill, J.E.W. Mayhew. "Ground Plane Obstacle Detection Under Variable Camera Geometry Using a Predictive Stereo Matcher," *Proceedings of the British Machine Vision Conference (BMVC '92)*, 1992.

[8] W. Enkelmann. "Obstacle Detection by Evaluation of Optical Flow Fields from Image Sequences." *Image and Vision Computing (UK)*, vol. 9, no. 3, June 1991.

[9] V. Graefe and W. Efenberger. "A Novel Approach for the Detection of Vehicles on Freeways by Real-Time Vision," *Proceedings of the Intelligent Vehicles '96 Symposium*, 1996.

[10] B. Heisele and W. Ritter. "Obstacle Detection Based on Color Blob Flow," *Proceedings of the Intelligent Vehicles '95 Symposium*, 1995.

[11] T. Kanade, A. Yoshida, K. Oda, H. Kano, and M. Tanaka. "A Stereo Machine for Video Rate Dense Depth Mapping and its New Applications," *Computer Vision and Pattern Recognition Conference (CVPR '96)*, June, 1996.

[12] K. Konolige and B. Bolles, (SVM reference)http://www.ai.sri.com/~konolige/svm/index.html.

[13] W. Kruger, W. Enkelmann, and S. Rossle. "Real-Time Estimation and Tracking of Optical Flow Vectors for Obstacle Detection," *Proceedings of the Intelligent Vehicles '95 Symposium*, 1995.

[14] Q.T. Luong, J. Weber, D. Koller and J. Malik, "An integrated stereo-based approach to automatic vehicle guidance," *Fifth International Conference on Computer Vision (ICCV '95)*, Cambridge, Mass, June 1995, pp. 52-57.

[15] L. Mathies and P. Grandjean. "Stereo Vision for Planetary Rovers: Stochastic Modeling to Near Real-Time Implementation," *International Journal of Computer Vision*, 8:1, pp. 71-91, 1992.

[16] L. Mathies, A. Kelly, T. Litwin, and G. Tharp. "Obstacle Detection for Unmanned Ground Vehicles: A Progress Report," *Proceedings of the Intelligent Vehicles '95 Symposium*, 1995.

[17] K. Oda. "Calibration Method for Multi-Camera Stereo Head for NavLab II." Internal CMU Document, 1996.

[18] L. Robert, M. Buffa, M. Hébert. "Weakly-Calibrated Stereo Perception for Rover Navigation," *Proceedings of the International Conference on Computer Vision (ICCV '95)*, 1995.

[19] L. Robert, C. Zeller, O. Faugeras, and M. Hébert. "Applications of Nonmetric Vision to Some Visually Guided Robotics Tasks," chapter from *Visual Navigation: From Biological Systems to Unmanned Ground Vehicles*, Lawrence Erlbaum Associates, 1997.

[20] (Raven Project Reference)

[21] (Point Grey Research/TriClops Reference) http://www.ptgrey.com/stereo/stereo.html

[22] J. Woodfill. (reference for Cornell stereo machine).