A Specialized Multibaseline Stereo Technique for Obstacle Detection

Todd Williamson and Charles Thorpe
Robotics Institute
Carnegie Mellon University
Pittsburgh PA 15213

Abstract

This paper presents a multibaseline stereo technique specially suited to detecting obstacles. A method is described for weakly calibrating a set of multibaseline stereo cameras with high accuracy. This method is then used to tailor the stereo search space to the special case where the world in front of the cameras consists mainly of nearly horizontal planar surfaces (the ground), where we are interested in deviations from those planar surfaces (obstacles). The resulting disparity maps are presented and compared to the output of a traditional stereo algorithm.

1. Introduction

Obstacle detection is one of the fundamental problems of mobile robotics. In order to navigate in the world, it is necessary to detect those portions of the world that are dangerous or impossible to traverse. For a large class of such navigation problems, the world in front of the robot can be modeled as a flat plane, and any points that deviate from the planar model can be considered to be obstacles. Examples of problems in this class are Automated Highway Systems (AHS) vehicles, and practically all indoor mobile robots.

Many different sensors have been used for obstacle detection, with varying degrees of success. Many groups (e.g. [3][4]) have chosen stereo vision because of the low cost and high reliability of the sensors and the fact that the sensing is passive. We have chosen *multibaseline* stereo because of those reasons and the improved reliability and accuracy of the recovered structure of the environment achievable with more cameras.

Conventional stereo systems for navigation often perform three steps. The first step is to find matching points between images. From these matches, it is then possible to compute the 3D coordinates of each point in the image. Finally, these points are arranged in a map so that possible vehicle motions can be evaluated.

In this paper we present a method, based on weakly calibrated stereo, that attempts to identify obstacles directly from the stereo imagery. The issue of camera calibration for multibaseline stereo has been addressed in a variety of ways. In this paper we first present a weak calibration technique that produces very accurate results in the general case. Then we present two variations to this technique that tailor the stereo search space for the obstacle detection problem. Finally, we compare the output of all three methods and discuss implementation issues.

2. Stereo Calibration

It is a well-known result of camera geometry that images from different pinhole cameras of the same planar surface, represented in 3D homogeneous coordinates, are related by a 3x3 homography matrix. Furthermore, this homography matrix has a specific structure:

$$x' = Hx = A_1(R + tn^T)A_0^{-1}x \tag{1}$$

Where the A matrices are 3x3 matrices containing information about the intrinsic parameters of the two cameras (focal length, aspect ratio, and image center). R is the 3x3 rotation matrix between the two camera axes, t is the 3D translation vector between the two camera focal points, and $n = \hat{n}/d$ is the 3D normal vector to the plane being observed, divided by the normal distance of the plane from the focal point of camera 0 [6][7][8]. Thus H encodes the relationship between the points in the image from camera 0, represented by x, and the points from camera 1, represented by x'

Given such homography matrices (H_1 and H_2) for two such planes, it is possible to weakly calibrate the set of cameras by simply taking the difference between the two matrices:

Copyright 1998 IEEE. Published in the Proceedings of CVPR'98, June 1998 Santa Barbara, CA. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works, must be obtained from the IEEE. Contact: Manager, Copyrights and Permissions / IEEE Service Center / 445 Hoes Lane / P.O. Box 1331 / Piscataway, NJ 08855-1331, USA. Telephone: + Intl. 908-562-3966.

$$\begin{aligned} H_2 - H_1 &= A_1(R - R + tn_2^T - tn_1^T)A_0^{-1} \\ &= A_1(t(n_2^T - n_1^T))A_0^{-1} \\ &= (A_1 t)[(n_2^T - n_1^T)A_0^{-1}] \end{aligned} \tag{2}$$

This matrix is rank 1, being the outer product of two vectors. Intuitively, $A_1 t$ is the position of the focal point of camera 0 in the image of camera 1 (which simply means that it is just the epipole of camera 1). The other half of the expression encodes the information about how the two planes being observed are different.

To perform the stereo computation using this result, we can simply compute the set of homographies:

$$H_\alpha = H_1 + \alpha(H_2 - H_1) \tag{3}$$

Where α is a scalar that controls what plane is actually represented by the homography. The normal vector n_α of this plane for H_α is given by:

$$n_\alpha = n_1 + \alpha(n_2 - n_1) \tag{4}$$

In practice, it is often convenient to aim the camera toward points that are so far away that their disparity is effectively zero. The distance at which this is the case depends on the focal length of the lenses and the length of the longest baseline. For most configurations, the sky near the horizon is sufficiently far away.

In this case, n_1 goes to zero, and equation (4) becomes

$$n_\alpha = \alpha n_2 \tag{5}$$

and α has the normal stereo relationship to 1/d. For the remainder of this paper, we will derive equations for the general case, but in the actual experiments we use the homography of the plane at infinity for H_1.

Thus, for any of the planes n_α we can compute the corresponding point in camera 1 for any point in camera 0. By stepping α through a range of values, we can get a series of planes and their point correspondences to use in the heart of the stereo matching algorithm.

The extension of this technique for multiple baselines is straight-forward. If the same two planes are viewed by multiple cameras, the corresponding homography matrices can be interpolated as above. No measurements of camera geometry are necessary.

In the following sections, we will first describe how we can compute homography matrices from image data, and then we will discuss three different methods for choosing the planes to use to perform the calibration.

2.1. Computing Homography Matrices

Since any given point in a 2D image can be represented by any of an infinite number of homogeneous coordinates, all scalar multiples of each other, we cannot expect to directly solve equation (1) for *H*. One way to represent this is to write the cross-product of the two homogeneous coordinates that are supposed to be equal, and set it equal to zero. This has the effect of constraining the two coordinates to be scalar multiples of each other.

Thus equation (1) becomes:

$$x' \times Hx = 0 \tag{6}$$

Since, given any solution *H* to this problem, all scalar multiples of *H* are also solutions, we can arbitrarily set one element of *H* to whatever value we like and solve for the other 8. Doing this we get two linear equations per image point, and a total of 8 unknowns, so four point correspondences are required to compute *H*.

Ideally, though, we would like to know the parameters of *H* to very high precision so that we can accurately compute depth using sub-pixel interpolation template matching.

For an AHS vehicle, the goal is to recognize small obstacles (as small as 20 cm or so) at long range (60-100 m in front of the vehicle). In order to accomplish this, a combination of telephoto lenses and a large baseline becomes necessary. In this situation, small inaccuracies in the calibration can cause large errors. In one particular situation that we have studied, using a 1 m baseline and 35 mm lenses, a 1 mm error (1 part in 1000) in the computed position of the camera can cause the epipolar line to be off by as much as 2 pixels in certain parts of the image at extreme disparities. Accurate computation of homography matrices is therefore essential.

Thus, some method of accurately determining calibration parameters is necessary. The most obvious way to do this is to minimize the residual error between one image and the other image when warped by the homography. Since the relationship between a pixel's location in one image and its position after transformation by the homography is a rational function, some type of nonlinear optimization must be used. For this, we use a program that has been in use at Carnegie Mellon for several years [4][6]. It asks the user to select four matching points to compute a starting point for a Levenberg-Marquart nonlinear optimization. The results of this computation are shown in Figure 1. For this case, the residual error was reduced by a factor of around 50%.

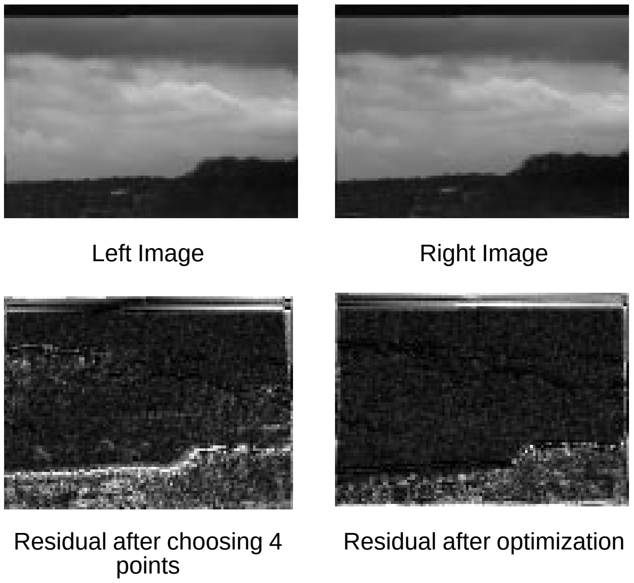


Figure 1: results of homography computation

After accurately computing the homography matrices of two planes, one issue remains. Since we were able to compute the homographies only up to a scale factor, the cancellation of R from the second line of equation (2) is not possible unless we compute the relative scale factors of the two matrices.

To accomplish this, we use the fact that the difference between the two homographies is rank 1 for the correct scale factor. So we simply have to find β such that $H_1 - \beta H_2$ is rank 1. In general, because of rounding errors and imperfect assumptions made by our model, there will be no β that accomplishes this exactly. We evaluate how good any given β is by computing the Singular Value Decomposition of $H_1 - \beta H_2$ and taking the ratio of the largest and second largest singular values. Finding the best value of β then becomes a simple 1D optimization problem.

In the following sections, we will assume that all homography matrices have already been normalized by β as computed by the above method.

2.2. Depth Calibration Method

If H_1 and H_2 are computed by viewing planes that are parallel to each other, then equation (4) becomes

$$n_\alpha = n_1 + \alpha(n_2 - n_1) = \frac{\hat{n}}{d_1} + \alpha\left(\frac{\hat{n}}{d_2} - \frac{\hat{n}}{d_1}\right) = \left(\frac{1-\alpha}{d_1} + \frac{\alpha}{d_2}\right)\hat{n} \quad (7)$$

where $\hat{n}$ is the common normal vector to both planes. So we get a set of planes, all of which are parallel to the

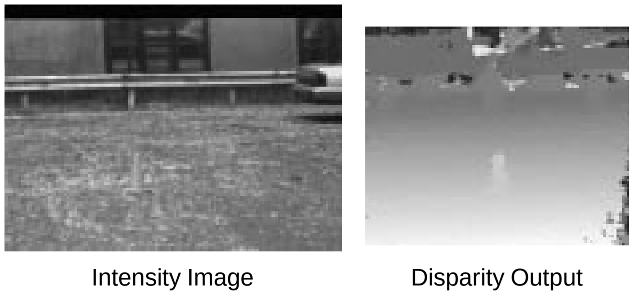


Figure 2: output with depth calibration

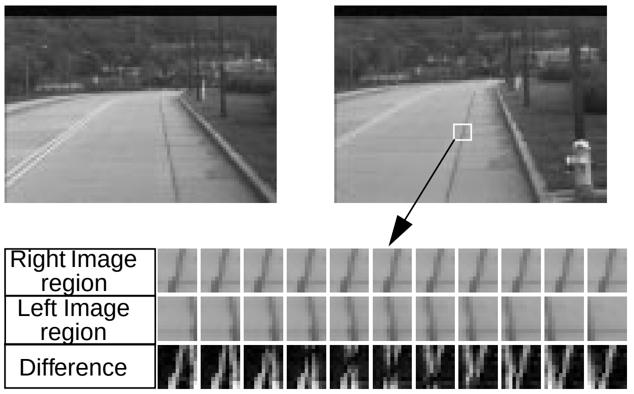


Figure 3: depth calibration method applied to a road surface

original planes.

If the initial planes are chosen such that they are roughly perpendicular to the camera axis, then $\hat{n}$ will be parallel to the camera axis, and the set of planes from equation (7) will be planes of constant depth. The geometry that is recovered allows us to compute the distance of points from camera 0. This corresponds exactly to the most common stereo vision method.

The multibaseline stereo output from a set of 3 cameras calibrated with this technique is shown in Figure 2.

Unfortunately, this method runs into problems in an environment consisting mostly of surfaces whose normals are not parallel to the camera axis. This is illustrated in Figure 3. The regions being matched in the left image have a different geometry arising from the fact that points that are higher in the image are actually farther away. This leads to a different shape within the image, which in turn leads to imprecise matching.

The method described here describes how to derive pixel-by-pixel differences which can then be summed over a window in order to compute the disparity. However, the same effect could be achieved when computing SSD (sum

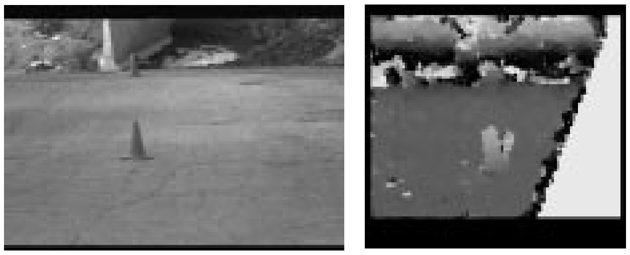


Figure 4: output with height calibration

of squared differences) over templates, simply by warping the templates using the homography for a given disparity level before computing the SSD.

2.3. Height Calibration Method

An interesting property of equation (7) is that the direction of the normal vectors of the planes being viewed do not have to be parallel to the camera axis. We can use this property to get around the problem with depth calibration described in the previous section.

An obvious adaptation would be to choose two planes that are parallel to the ground (note that the plane at infinity can be used for H_1 in this case as well). In this configuration, the output of the stereo algorithm would be the height of each point in the image.

The output of this height calibration is shown in Figure 4. The triangular section on the lower right represents pixels for which all possible matches lay outside the image for this calibration method. The scene being viewed in this figure is different from the one used for calibration.

The results of this simple height calibration are reasonably good, but it seems to have problems at long distances. To get at the core of why this is, it is helpful to go back to equations (1) and (2).

$$x' = H_1 + \alpha(H_2 - H_1)x = H_1 x + \alpha A_1 t(n_2 - n_1)^T A_0^{-1} x \quad (8)$$

Note that the quantity $(n_2\text{-}n_1)^T A_0^{-1} x$ is a scalar. This scalar is the dot product of the direction in world coordinates corresponding to the pixel x, $A_0^{-1}x$, with the direction of the ground plane. If the horizon appears in the image, pixels at the horizon will be perpendicular to the direction of the ground plane, and this scalar will be zero. The effect of this will be that pixels on the horizon will all map to the same location, regardless of α. Another effect is that pixels that are close to the horizon will move very little as α increases, while pixels that are far from the horizon will move much more. Pixels that are above the horizon will actually move backwards (as if looking at the ground behind the cameras).

It is clear from this that no single choice of step size for α will give good results for this method, and the result of that can be seen in the upper portion of Figure 4. On the other hand, the method has several good features:

- the regions being compared between the images automatically have the correct shape if the surface being viewed is a plane parallel to the ground
- the mapping from the output of this method (height) to whether a region should be considered an obstacle or not is obvious -- regions that are too high (or too low) should be considered to be obstacles.

2.4. Combined Method

Although the height calibration method is promising, the problems mentioned above seem to be too extreme make it practical. Ideally, there would be some combined method which captures the strong points of the height method with none of the drawbacks.

The problems associated with the height calibration method mostly seem to stem from the fact that the direction of $\hat{n}$ in equation (7) is far from the camera axis. The benefits, on the other hand, stem from the fact that the planes involved are similar to the ground plane.

Thus, we define a new set of homographies

$$H_\alpha = H_3 + \alpha(H_2 - H_1) \quad (9)$$

Where H_3 is computed for the ground plane, but H_2 and H_1 are computed from planes such that the difference between normal vectors lies near the camera axis (such as with the depth calibration method). With this set of homographies, the corresponding planes look like:

$$n_\alpha = n_3 + \left(\frac{1-\alpha}{d_1} + \frac{\alpha}{d_2}\right)\hat{n} \quad (10)$$

Where $\hat{n}$ is the common normal of H_2 and H_1, lying near the camera axis. An example of this set of planes for a simple camera geometry is shown in Figure 5.

The output of this scheme is shown in Figure 6. Again, the scene used for output is significantly different from that used for calibration.

This sampling method has a number of good properties:

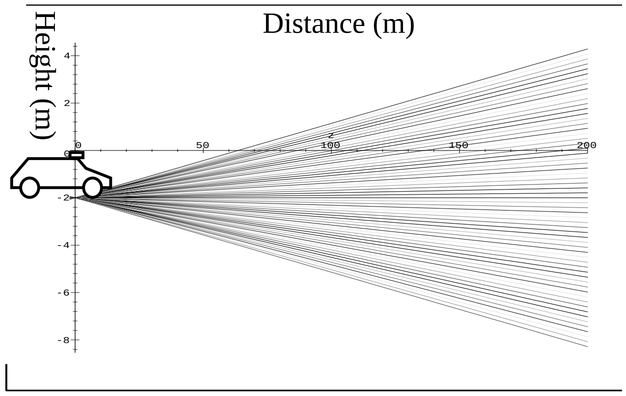


Figure 5: Planes of constant "disparity" for the second calibration method. Parameters are 1m baseline, 35mm lenses, 1/2" CCD, cameras aligned perfectly and 2m above the ground

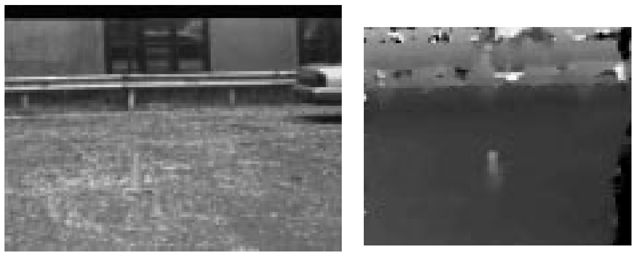


Figure 6: output with combined calibration

- the template shape is correct for objects whose surface corresponds to one of these planes

- the different planes are all distinct everywhere in the image

- the allocation of the limited number of disparity search levels is optimized for road geometry (since we do not expect 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)

- 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 roughly planar in the (row,column,disparity) space of the output of this algorithm (this is not true for the height calibration case).

In Figure 7, we present the regions used by this method, for comparison with those used by the depth method (shown in Figure 3). Since the regions are transformed appropriately for the expected surface normal

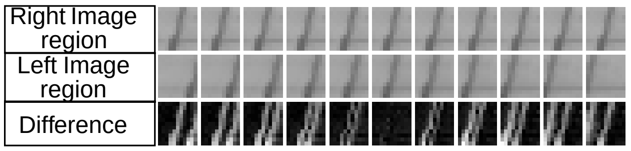


Figure 7: combined calibration method output for the situation shown in Figure 3

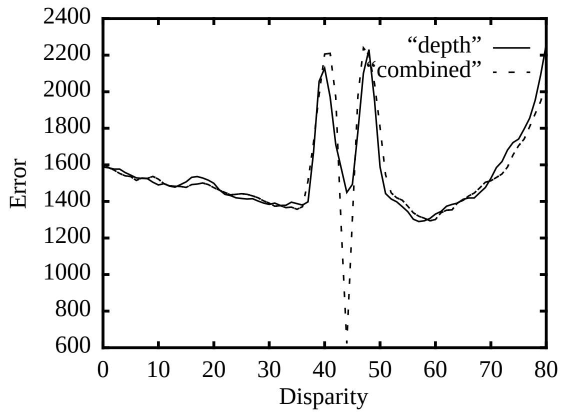


Figure 8: error for the cases shown in Figure 3

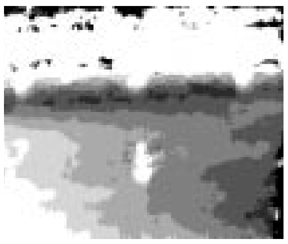
direction, the shape of features in the image is correct, and matching is much more likely to be accurate.

3. Comparison of Results

In Figure 8 we have plotted the error as a function of disparity for the case considered in Figure 3 and Figure 7. Here the benefit of this method becomes clear. The depth calibration method (shown with a solid line) achieves a local minimum of error at the correct value, but it is not less than the value attained in regions where the road surface has no texture. Thus, the result in this case is an incorrect disparity value. The combined method properly and strongly identifies the correct minimum.

Another means of evaluating the results of these methods is to look closely at the shapes of the regions that are identified by each algorithm as having a particular disparity value. We can compare the output of the two methods by mapping the output shown in Figure 2 into the space of the output in Figure 6. The results of this mapping are shown in Figure 9, with the regions corresponding to four disparity levels of the ground displayed as four different shades of gray.

As can be seen in the figure, the combined method maps the plane of the ground into around five regions, representing five different planes of the type shown in Figure 5. Details like the fact that the ground slopes down-



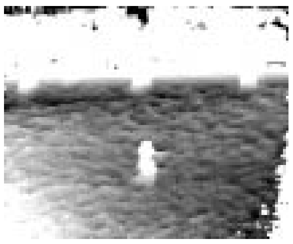


Figure 9: comparison of results for ground plane pixels -- four disparity levels for the ground are shown in levels of gray

ward from left to right (relative to the original calibration plane) are clearly visible. Given the results shown in Figure 8, it is not surprising that the depth method does not do as well on the road surface. Although the general trends are present, the accuracy is affected strongly by the incorrect feature shape used in matching.

4. Implementation

The algorithms described in this paper have been implemented both using special purpose hardware, the Video-Rate Multi-baseline Stereo Machine, and in C code on a Sun workstation.

4.1. Video-Rate Multi-baseline Stereo Machine

The stereo machine consists of a number of custom-built 9U VME boards connected in a system. The system is described in more detail in [2].

The algorithm used by the stereo machine works by first digitizing the images from each of the cameras (up to 6 in the current design), and then passing the image 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, to rectify these 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 ($I(i,j,\zeta)$, $J(i,j,\zeta)$). These 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 contain any values whatsoever, so it is possible to correct for lens distortion, or to operate with one camera upside down, or to use cameras with lenses of different focal lengths. The primary limitation of the geometry compensation circuit is that it assumes that each 4x4 pixel region of the base image will be offset by the same amount. For the kind of extreme geometry that has been presented in this paper, this assumption is often not a very good one, and the quality of the output suffers.

Calibration for the stereo machine consists entirely of computing the values to load into these lookup tables. These values can be computed directly from the homography matrices by the simple formula

$$\begin{bmatrix} aI(i,j,\zeta) \\ aJ(i,j,\zeta) \\ a \end{bmatrix} = H_\zeta \begin{bmatrix} i \\ j \\ 1 \end{bmatrix} \qquad (11)$$

and normalization by a to convert from homogeneous coordinates to 2D coordinates.

In the next stage of the stereo machine, the absolute value of difference (AD) is performed pixel-by-pixel for the base camera (camera #0) paired with each of the other cameras. The results of the AD computation are summed over all of the camera pairs, resulting in a sum of absolute differences (SAD) value for each pixel for each disparity level.

The resulting SAD values are now smoothed by summing over a window, 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.

The stereo machine processes images at a constant rate of roughly 30 million pixel-disparities per second (counting pixels processed in camera #0), regardless of the number of cameras in use. Thus the frame rate depends on the number of pixels processed and the number of disparities searched. When using the maximum values for each (256x200 image, 60 disparity levels searched), the frame rate is roughly 6 Hz.

For the experiments presented in this paper, the stereo machine was mounted on a vehicle and driven to a variety of different locations.

4.2. C Program Implementation

The implementation in C code emulates the stereo machine algorithm completely, except for the geometry compensation. Instead of using a lookup table, the geometry compensation is done by computing image coordinates directly from the homography matrices, for maximum accuracy. With the exception of Figure 4, all of the output shown in this paper was computed with this program.

5. Conclusion

A system has been presented that allows very precise calibration of multi-baseline stereo cameras. Such calibration is necessary in the case where long focal lengths and large baselines are necessary to observe objects at long range, or when maximum depth precision is required.

Building on this method, we have shown two possible modifications that allow stereo vision to be used to accurately detect planar surfaces in front of a vehicle. This then allows us to detect regions that do not lie on those planar surfaces, which can therefore be classified as obstacles.

A key principle of these methods is that distances to objects whose surface normals are very different from the camera axis direction can not be measured accurately with traditional stereo vision algorithms. If these surface normals are different enough, then the distance to the object may vary significantly over the areas being considered, thus violating a key assumption of traditional stereo algorithms.

Other methods to cope with similar situations have been presented in [1] and [7]. The former presents a method intended to reduce stereo vision processing while accounting for vehicle motion, and adaptively adjusts the parameters of the ground plane as the vehicle moves. The latter attempts to compute the height of objects in the scene using a weakly calibrated stereo system by making some simplifying assumptions.

The methods we have described in this paper have been tested in a variety of real-world environments and appear to be very robust to different situations. In future work we will be building a new frame-rate stereo machine that is capable of computing point correspondences instead of relying on a sparse, imprecise lookup table.

6. Acknowledgment

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

Most of the work described in section 2.1. was performed by the stereo machine group at Carnegie Mellon, primarily Kazuo Oda and Tak Yoshigahara.

In addition, we want to thank Martial Hébert for many hours of constructive conversation about the mathematics of stereo imaging.

7. References

[1] P. Burt, P. Anandan, K. Hanna, G. van der Wal, and R. Bassman. “A Front-End Vision Processor for Vehicle Navigation,” *Proceedings of the International Conference on Intelligent Autonomous Systems (IAS-3)*, 1993.

[2] 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.

[3] 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.

[4] 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.

[5] K. Oda. personal correspondence.

[6] K. Oda. “Calibration Method for Multi-Camera Stereo Head for NavLab II,” Internal Carnegie Mellon Document, 1996.

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

[8] 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.