# Obstacle Detection in Smooth High Curvature Terrain

Parag H. Batavia and Sanjiv Singh
[parag/ssingh]@ri.cmu.edu

Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

## Abstract

*Detection of obstacles for autonomous vehicles is more difficult when the terrain is not locally planar and remains an open problem. We have developed an approach suited for obstacle detection in those cases where the terrain has significant curvature but is smooth enough that the obstacles are discrete. Our system consists of a low-cost scanning laser rangefinder and a novel algorithm that can reliably detect obstacles as small as 15 cm in curving terrain. This paper presents an analysis of the effectiveness of our system and a summary of experimental results from an outdoor mobile robot.*

## 1. Introduction

Obstacle detection is a key capability for autonomous vehicles and remains an open problem. The problem is complicated during operation in hilly (high-curvature) terrain because the natural rise and fall of the terrain can be construed as an obstacle or worse, the curvature can lead to missing an obstacle altogether. In some cases, it is possible to assume that the terrain although curved, is locally smooth and hence the obstacles appear discretely. Parklands and golf courses are good examples of this type of terrain. Here we address an obstacle detection system suited to such environments. Since we target real-world commercial application, we add the requirements that the system be sensitive (able to detect obstacles as small as 15 cm), robust (operate in varying lighting and environmental conditions with a low false alarm rate) and inexpensive.

A large body of work in obstacle detection for outdoor vehicles uses range imagery from one of stereo vision, ladar or radar. Stereo vision provides dense range imagery but generally is not able to provide the range resolution to detect small obstacles reliably especially given a variety of natural textures and lighting conditions [3][9][16]. Radar is immune to a large range of environmental conditions and works well for detecting vehicle sized obstacles but its large beam size prevents detection of small obstacles and produces only coarse localization for the targets that it does detect [8]. Ladar has the ability to detect small obstacles because of small beam divergence but generally requires mechanical scanning. While flat indoor environments and operation at low speeds allow the use of a single horizontally scanned beam pointed straight ahead of the vehicle [12] this approach generally doesn't work outdoors and especially when the terrain curves significantly. One solution is to use a 2-D scanning ladar that has a scanning pattern similar to that of a television [6]. Such systems are not only very expensive but require accurate measurement of the vehicle motion during the scan to properly register the range data. In addition to the above methods there has been work on obstacle detection using passive vision. Our own work in the past has used two of these methods -- *color segmentation* and *homography* to detect obstacles [2]. Although well understood, these modalities can fail due to variations in lighting (color segmentation) and changes in elevation (homography). Additionally, color segmentation fails in cases where geometry does matter -- finding an unexpectedly colored patch of ground can result in a false alarm while true obstacles that might have the color of the expected terrain are not detected.

In this paper we present a novel, low-cost ladar scanner whose configuration removes the need for accurate measurement of vehicle motion for registration of range data. We also present a simple algorithm that is able to reliably detect obstacles as small as 15 cm traveling at speeds up to 2 m/s. The main advantage of this method is that it has a point density comparable to that provided by stereo algorithms, with the accuracy of a ladar. It is also robust to the problems that passive vision systems suffer from. The obstacle detection algorithm is suitable for terrain in which all areas are assumed to be equally traversable except for those occupied by obstacles. This is in contrast to algorithms which search for paths through unstructured terrain by building explicit traversability maps [16]. Our method does not provide traversability measures, but only information regarding the presence (location and geometry) of obstacles. It can, however, be used as a source of information to populate maps which can be evaluated for traversability.

Section 2 provides some detail on three modalities (color segmentation, homography, single axis ladar scanner) that we have used in prior work and discusses their shortcomings. Section 3 presents our new approach and an analysis of sensitivity of the method based on parameters such as sensor placement. Section 4 presents experimental results from our testbed.

## 2. Obstacle Detection

In this section, we discuss previous approaches to obstacle detection which make use of passive vision and single axis ladar scanning, and present their failure modes.

### 2.1. Color Segmentation

Color segmentation for navigation and obstacle detection uses color information to detect the presence of obstacles. Previous research has used various representations of color and different methods to generate models of obstacles and non-obstacles [7][10][15]. There are three underlying assumptions that are common to color segmentation algorithms regardless of the implementation details: 1) The "obstacle" and "freespace" classes can be adequately modeled using the chosen color representation, 2) in the chosen representation, the two classes are truly distinct, with little overlap, and 3) The geometry of the scene is not important to determining whether a feature is an obstacle. The first assumption holds if we assume that the color of the terrain the mobile robot will be operating over is homogenous. However, even with this assumption, lighting changes due to shadows, clouds, and time of day can dramatically alter the appearance of both the terrain and obstacles. This can result in a high number of false positives. This can be partially overcome by dynamically training the classes as lighting conditions change.

The second assumption, regarding the distinctiveness of the "obstacle" and "freespace" classes, is one which is not true in many cases, and results in false negatives. For example, if operating on grass, a false negative can occur if a green shrub or branch is encountered. The color of the obstacle in this case is very similar to, and overlaps, the color of the "freespace" class. Similarly, if the third assumption, that of geometry not being important, is violated, false positives result. Consider an area of white sand in the middle of a grassy area. The white sand will have a different color than the "freespace" color, and will appear as an obstacle, even though it is traversable. Since color segmentation does not take the geometry of the scene into account, it can fail in conditions where the geometry does matter. The next section discusses an approach to overcome this limitation.

### 2.2. Homography

Homography is a planar mapping, which can be computed using a stereo image pair. That is, if we assume that the features being imaged lie on a known plane and that the displacement between the two cameras is known, it is possible to predict where features in one image will be found in the other image. Therefore, it is possible to detect whether a particular image feature lies on the assumed ground plane. Unlike traditional stereo, there is no disparity search, as explicit range values are not computed. The complexity is *O(n)* in the number of pixels.

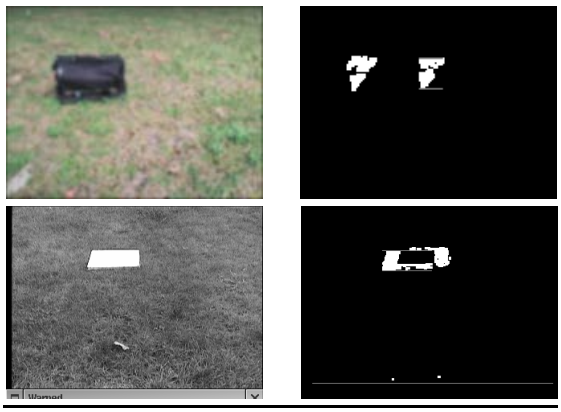

Figure 1: Examples of homography behavior. The images on the left are the right image from the stereo pair, and the images on the right are the output. The output is a thresholded difference image between the warped and actual images. The top pair is a true positive - the bag is properly detected. The bottom pair is a false positive. The paper is detected as an obstacle, since it is laying on a hill, above the assumed ground plane.

Given a two-camera stereo pair, we can compute the homography between a common plane in both images. If we assume that the world is locally flat, then the homographic transform between both cameras can be computed. This can be used to warp images from one camera into the expected image from the opposite camera. If all the image features in the first image are on the ground plane, then the correspondence between the warped image and the actual opposite image will be high. If, however, certain image features do not lie in the ground plane, the warping process will distort the features, and the difference can be detected.

The underlying assumption is that the ground is locally flat. Much of the previous work in using homography for obstacle detection has been on either indoor mobile robots [13][14], or highway settings [1][4], where this assumption holds. This approach breaks down when there are local elevation changes. Features which are on the (elevated) ground are warped as if they lay on the plane for which the homography matrix was computed. This leads to false positives, as shown in Figure 1. The left images are input (in this case, one image from the stereo pair), and the right images are classified output. The top pair shows a bag properly classified as an obstacle, since it rises above the (flat) ground plane. The bottom pair, however, shows a piece of paper lying on the upslope of a hill. Although flat on the ground, the paper is not lying on the assumed ground plane, and is instead above it. Therefore, it is not properly warped and is classified as an obstacle.

### 2.3. Single-Axis Range Scanning

Traditionally, single-axis ladar scanners are used in either a “forward looking” or “push-broom” configuration [12]. The former positions the laser such that each beam scans outwards, without intersecting a flat ground plane. The advantage of this configuration is that any return at all can be construed as an obstacle. The disadvantage is that the sensor height determines the minimum obstacle height which can be detected. In addition, platform pitching motion or elevated terrain can produce false alarms.

Alternatively, the sensor can be placed in push-broom configuration, in which it intersects the ground plane in a line at some distance ahead of the vehicle The sensor can then detect arbitrarily small obstacles, to the limit of its resolution. The disadvantage is that pitching motion is still confounded with obstacles. If the platform were to pitch downwards, all the returned ranges would be less than expected, making it appear as if an obstacle were present. It is possible to assume that the majority of laser beam returns are actually from the ground plane. We have implemented robust curve fitting to the return, using a Least Trimmed Squares (LTS) approach. Detected outliers from the computed fit are considered to be obstacles. We have implemented this and it is robust to pitching motion and small changes in local terrain elevation. However, this method fails in the presence of certain types of obstacles. In particular, if there is a large obstacle which subtends most of the laser field of view, LTS will fit a surface to that obstacle, determine there are no outliers, and report a false negative. This highlights one important failure mode of single-axis scanners, which is that they are not able to detect in the direction of travel. I.e., range discontinuities which are perpendicular to the sensor motion can be detected, but any discontinuities which are parallel to the direction of motion are not detected, unless time histories of scans are kept and maps are built. Keeping time histories allows range data to be registered precisely when vehicle attitude can be measured at a high rate but gaps between successive scans due to the vehicle motion can be larger than obstacles we would like to detect.

## 3. Two-Axis Range Scanning

Given the limitations and failure modes of the previous methods, we have developed a two-axis laser range scanner. Conventional two-axis scanners are expensive. Furthermore, they offer range and resolution beyond what is required for many applications. Therefore, we have adapted a low-cost single line laser range finder to operate as a two-axis scanner by rotating it so that the laser scans *vertically* instead of horizontally, and then mechanically sweeping it from side to side to provide horizontal coverage. This scanner provides 75 scans/second, where each scan covers 100 degrees, at one degree resolution. In this context, a *scan* is one line of laser data, scanned vertically. A *sweep* is a set of scans, collected as the laser is mechanically swept from left to right (or right to left). We conducted a simulation study to determine the ability of the laser to cover a desired area at a high enough resolution, along with expected obstacle detection performance.

### 3.1. Coverage and Sensitivity Analysis

There are a number of parameters which affect the behavior of the two-axis scanner. These include scanning rate, sweeping rate, sweeping range, and sensor placement (height and downangle). Platform motion and maximum velocity is also a consideration. We performed a sensitivity analysis, to determine how changes in the above parameters affect field of coverage and obstacle detection sensitivity.

For example, if the laser is placed high, the coverage pattern it generates tends to be relatively uniform. However, the “shadow” cast by a laser beam hitting an obstacle is concurrently smaller, and fewer beams hit the obstacle, reducing sensitivity. An increase in laser sweeping rate will decrease the time between a particular area being imaged, but will also reduce the horizontal resolution of each individual sweep. Physical and sensor constraints, such as angular rate limits, laser scanning rate limits, etc. have to be accounted for. We therefore evaluated parameters with the goal of being able to detect 6” (15cm) obstacles assuming a vehicle width of 1.8m. If we assume the platform is travelling at 1m/s, then obstacles should be detected at a distance of 2.0m to allow for adequate stopping distance, assuming a 0.8G braking acceleration, and a reasonable safety buffer.

We evaluate parameters by analyzing the resulting coverage pattern. A simulation is used to generate the synthetic range data from a moving laser. We first bin each laser beam’s intersection with a flat ground plane into a $5cm^2$ grid. After that, each grid cell is either *covered*, meaning that at least one laser beam intersects it, or *uncovered*, which means that no beam intersects it. We then compute the percentage of covered cells, and also do a blob size analysis on 8-neighbors of uncovered cells to determine the area distribution of uncovered regions. The top of Figure 2 shows the trend in coverage percentage for two laser heights, low (0.78m) and high (2.8m), as the fixed downangle is varied. The x-axis is downangle, and the y-axis is the percentage of $5cm^2$ cells which are covered. As downangle increases beyond a certain angle, the coverage deceases, due to a majority of the beams being concentrated very close to the sensor. This results in very dense coverage in certain

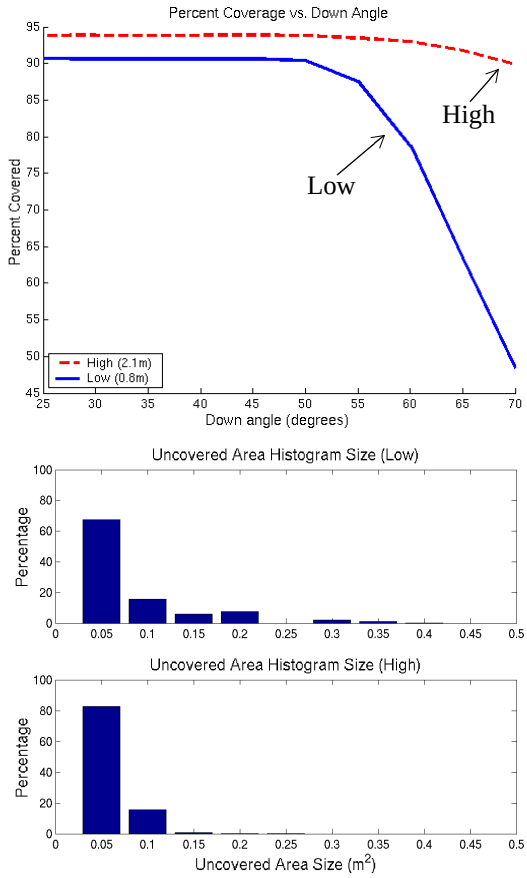


Figure 2: Comparison of coverage percentage vs. laser down angle for two laser height configurations (top) and histograms of uncovered area size (bottom)

areas, and very sparse coverage in other areas. Note that the higher placement results in better coverage. Figure 2 also shows two histograms of the area of uncovered cells. The top histogram is with the laser placed high, the bottom histogram is with the laser placed low. In both histograms, the size of the majority of uncovered areas is 1 grid cell, or $0.05m^2$. There is a slightly greater proportion of larger missed areas in the low configuration, due to coverage becoming less homogenous as the sensor is lowered.

However, coverage alone is not a sufficient predictor of obstacle sensitivity. What is most important is the number of laser beams that hit a particular obstacle, since that will determine how reliably the obstacle can be detected. The number of beams that hit an obstacle is related to the *shadow length* of the obstacle, which is the length of the shadow cast by the obstacle when illuminated by the laser sensor. The shadow length can be calculated as:

$$S = \frac{O_h O_d}{L_h - O_h} \qquad (1)$$

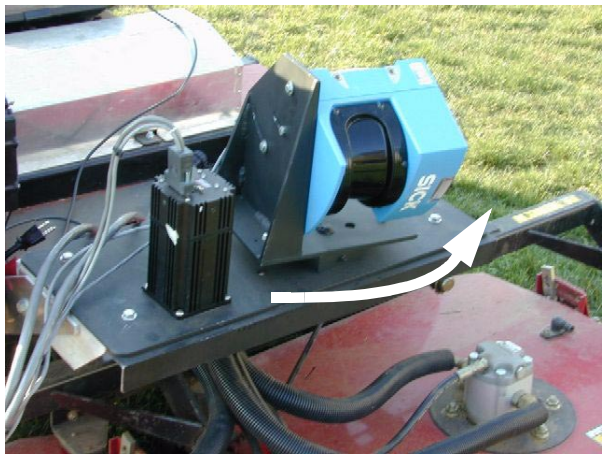

Figure 3: Two-axis scanner prototype

where $L_h$ is laser height, $O_h$ is obstacle height, $O_d$ is obstacle distance, and S is shadow length. From the above, it would appear that as obstacle distance increased, the shadow length would increase, resulting in greater detectability. However, at greater distances, the fixed angular interval of the laser scan results in a greater linear distance between beam hits. The following equation can be used to compute expected number of beam hits:

$$Hits = \frac{\operatorname{atan}\left(\frac{O_d}{L_h}\right) - \operatorname{atan}\left(\frac{O_d + S}{L_h}\right)}{R} \qquad (2)$$

where *R* is the angular resolution of the scan, and *Hits* is the expected number of hits. The other parameters are the same as in equation 1. From this, we can see that a lower sensor height results in a slightly greater number of hits, and a significantly larger shadow. Sensor coverage, however, is more uniform with the laser in the high configuration.

Therefore, there is a trade-off between a high mount and a low mount. The high mount provides better coverage, particularly at greater ranges. The low mount results in a slightly greater number of hits on a given obstacles. The better coverage of the high mount can significantly impact detection of small obstacles at greater range. However, at greater heights, unmeasured roll of the platform can induce significant registration error. Therefore, we have determined that a low mounting height is preferable.

We have constructed a prototype of this configuration, and it is shown in Figure 3. The following section describes a novel obstacle detection algorithm which operates on 2-D range data, and results.

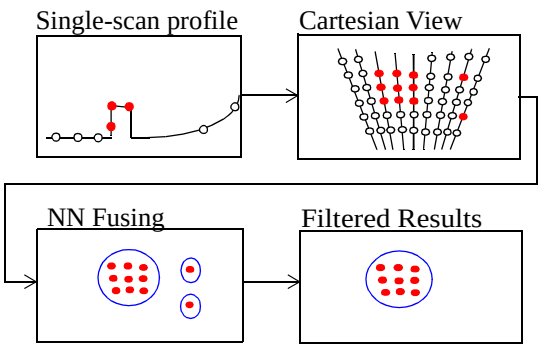


Figure 4: Obstacle Detection Algorithm overview. Each scan is classified as "obstacle" or "freespace". Obstacle pixels from groups of scans are clustered using nearest-neighbor to generate candidate obstacles. Candidate obstacles are filtered based on gross statistics.

### 3.2. Obstacle Detection Algorithm

The obstacle detection algorithm consists of two stages: classification, and fusing. As scan lines come in, each range point in the scan is classified as 'obstacle' or 'freespace'. Scans are accumulated, and then obstacle pixels are clustered using a nearest-neighbor criterion, and candidate obstacles are then filtered based on statistics such as mass and size. Figure 4 illustrates this procedure.

The novelty of this algorithm is that point clustering and scan fusing is done in such a way as to be robust to data mis-registration and hence false positives.

The first stage, classification, is fundamentally gradient-based. Gradient algorithms, as the name suggests, convert laser range data into cartesian coordinates in a vehicle-centric frame, and calculate the terrain gradient and look for areas which are not traversable based on slope. Our gradient calculation and classification method is similar to the method proposed by Chang [5]. Their algorithm processes each scan as it is accumulated, and classifies returns as obstacle or freespace based on gradients and heuristics. In our approach, as each scan is accumulated, it is registered to a fixed coordinate frame to account for platform motion. The gradient is computed along the dimension of the scan line. This is done in real time as each scan is received. A threshold is applied to the gradient to classify each pixel as 'obstacle' or 'freespace'. Each classified scan is then added to a circular buffer which contains a time-history of scans. The duration of this 'time-window' determines the amount of data which will be fused when clustering.

At regular (user specified) intervals, the points classified as obstacles are clustered using a nearest-neighbor approach. The nearest-neighbor approach is in contrast to the conventional approach of registering all the data to a common frame, and *then* classifying pixels, which

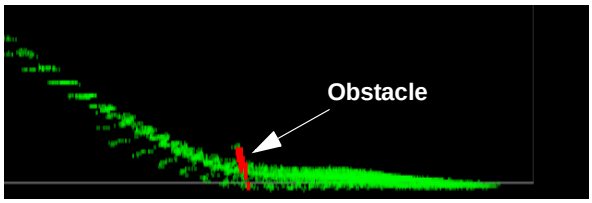


Figure 5: Profile view of a 22cm (9") obstacle at the foot of a slope.

can lead to false positives due to mis-registration of data. Our approach avoids this by clustering obstacle pixels *after* they are classified. Therefore, small mis-registrations in the data lead to small errors in gross obstacle statistics, rather than errors in deciding whether an obstacle is present. The latter is relatively minor, while the former can lead to undesired stopping or obstacle-avoidance behavior. This is similar in motivation to previous work which merges traversability maps created from instantaneous range images rather than merging elevation maps because of indeterminacy of vehicle motion in between images [16].

There are two parameters which control the behavior of the algorithm: the gradient threshold, and the time window. The output rate is another parameter, but does not affect results. It is a function of available computing power -- a faster CPU can support a faster output rate than a slower CPU.

## 4. Results

The following sections present results using real data taken from the two-axis scanner. The scanner has been mounted on an autonomous vehicle with high precision GPS-based localization, which provides 2cm accuracy. We present results of a test of obstacle detection sensitivity, using the algorithm presented in the previous section.

### 4.1. Obstacle Detection Results

We tested obstacle detection performance in two situations: flat terrain and curved terrain. Three different square obstacles were used, with side lengths of 15cm, 22cm, and 30cm. The tests were conducted on a moving platform. An initial pass over obstacle-free terrain was used to adjust the gradient threshold such that no false positives were generated. When testing on hilly terrain, each obstacle was placed at the foot of a steep (20% grade) hill. We started 10m away from the obstacle and drove towards it at 1.5m/s, stopping at a distance of 0.5m from the obstacle. Figure 5 shows a point cloud (in profile) of a 22cm obstacle at the foot of a 20 degree slope. The obstacle pixels are visible at the beginning of the slope. Note the unevenness at the beginning of the upslope. This is due to registration errors from unmodeled pitch motion. The obstacle detection algorithm is robust to such errors.

Table 1 shows results for combinations of obstacle size, terrain type, and time-window on flat and hilly terrain.

**Table 1: Max Distance (m) at which Obstacle (cm) is Reliably Detected for Flat (Hilly) Terrain Using Different Time Windows (s)**

| Obs Size / Window | 15cm (6”) | 22cm (9”) | 30cm (12”) |
|---|---|---|---|
| **0.5s** | **3.28 (2.13)** | **4.24 (4.00)** | **6.15 (7.01)** |
| **1.0s** | **3.15 (1.90)** | **4.38 (4.05)** | **7.45 (7.13)** |
| **1.5s** | **3.09 (1.83)** | **4.30 (4.09)** | **7.50 (7.07)** |
| **2.0s** | **3.25 (1.89)** | **4.24 (4.05)** | **7.73 (7.01)** |

The time-window parameter controls how much data is fused when detecting obstacles. The entries in the table show the distance to obstacle when it was *first* detected. I.e., this is the maximum distance at which we can expect to detect the obstacle. We see that in all cases, the obstacle is detected earlier on flat terrain than on hilly terrain. This is most likely due to our use of a steep gradient threshold, combined with low density sampling at far (> 3m) distances due to the geometry of the laser. In all cases, the obstacles were detected within the stopping distance of the mobile platform we use, which travels at 1.5-2m/s. Interestingly, the time window has very little impact on the detection distance. This is because the robustness to false positives is great enough that the algorithm can reliably detect an obstacle with a small number of hits at extreme range. The additional hits later on do not add to the detectability, although they do add to the confidence. For instance, at 2.0m, a 15cm obstacle generates 9 hits with a time window of 0.5s. However, if the time window is 2.0s, then there are 32 hits.

## 5. Conclusion

This paper has discussed failure modes of common obstacle detection methods which rely on passive cameras and single-axis laser scanners. Following that, we presented a simulation analysis of a proposed two-axis system which is low cost, and provides sufficient resolution to detect small obstacles. The simulated performance was bourne out using real hardware mounted on a mobile platform. Our proposed obstacle detection algorithm is robust to false positives, and is sensitive enough to detect small obstacles.

## References


[1] P. Batavia and D. Pomerleau and C. Thorpe, “Overtaking Vehicle Detection Using Implicit Optical Flow,” Proceedings of the IEEE Intelligent Transportation Systems Conference, Boston, MA, 1997.

[2] P. Batavia., and S. Singh, “Obstacle Detection Using Adaptive Color Segmentation and Color Homography,” Proceedings of the Int’l Conference on Robotics and Automation. IEEE, May, 2001.

[3] P. Bellutta et. al., “Terrain Perception for Demo III,” Proceedings of the IEEE Intelligent Vehicles Symposium 2000, Dearborn, MI, 2000.

[4] M. Bertozzi and A. Broggi and A. Fascioli, “Stereo Inverse Perspective Mapping: Theory and Applications,” Image and Vision Computing, Vol 16, pp. 585-590, 1998.

[5] T. Chang., et. al., “Concealment and Obstacle Detection for Autonomous Driving.” Proceedings of the Int’l Association of Science and Technology for Development - Robotics and Applications Conference, Santa Barbara, CA, 1999.

[6] J. Hancock et al., “Active laser radar for high performance measurements,” Proceedings of the IEEE International Conference on Robotics and Automation, 1998.

[7] J. Hyams and M. Powell and R. Murphy, “Cooperative Navigation of Micro-rovers using Color Segmentation,’ Journal of Autonomous Robots, Vol. 9, Num. 1, pp 7-16, August, 2000.

[8] D. Langer and T. Jochem, “Fusing Radar and Vision for Detecting, Classifying, and Avoiding Roadway Obstacles,” Proceedings of the IEEE Symposium on Intelligent Vehicles, 1996.

[9] R. Mandelbaum et. al. “Real Time Stereo Processing, Obstacle Detection, and Terrain Estimation from Vehicle-Mounted Stereo Cameras,” Proceedings of the Applications of Computer Vision, 1998.

[10] M. Ollis, “Perception Algorithms for a Harvesting Robot,” Carnegie Mellon University Doctoral Dissertation, CMU-RI-TR-97-43, August, 1997.

[11] P.J. Rousseeuw and K. Van Driessen., “Computing LTS Regression for Large Data Sets,” Technical Report, University of Antwerp, 1999

[12] N. Roy, et al., “Towards Personal Service Robots for the Elderly,” Proceedings of the Workshop on Interactive Robots and Entertainment, 2000.

[13] J. Santos-Victor and G. Sandini, “Uncalibrated Obstacle Detection Using Normal Flow,” Machine Vision and Applications, Vol 9, pp 130-137, 1996

[14] K. Storjohann and T. Zielke and H.A. Mallot and W. von Seelen, “Visual Obstacle Detection for Automatically Guided Vehicles,” Proceedings of the International Conference on Robotics and Automation, pp 761-766, 1990.

[15] I. Ulrich and I. Nourbakhsh, “Appearance-Based Obstacle Detection,” AAAI National Conference on Artificial Intelligence, Austin, TX, August 2000.

[16] C. Urmson and M.B. Dias, “Stereo Vision Based Navigation for Sun-Synchronous Exploration,” Proceedings of the International Conference on Robotics and Automation, 2002, IEEE, May, 2002.

[17] P.A. Veatch and L.S. Davis., “Efficient Algorithms for Obstacle Detection Using Range Data,” Computer Vision, Graphics, and Image Processing, 50(1), April 1990.


## Acknowledgements


The authors would like to thank Adam Szymanski for his efforts in developing the simulator used in this work.