Submitted to the International Association of Science and Technology for Development - Robotics and Applications99 Conference, Santa Barbara, CA, October 28-30, 1999.

Concealment and Obstacle Detection for Autonomous Driving

Tommy Chang, Tsai-Hong, Steve Legowik, and Marilyn N. Abrams

Intelligent Systems Division
National Institute of Standards and Technology (NIST)
Building 220, Gaithersburg, MD 20899

{tchang, hongt, legowik, nashman}@cme.nist.gov

Abstract

Obstacle detection and mapping are essential for unmanned autonomous driving. This paper describes both the sensors and the supporting software used in our system for driving autonomously on cross-country roads. The Ladar Range Imaging Camera (EBK) is used for monitoring the environment. We describe an algorithm developed at the National Institute of Standards and Technology (NIST) for detecting obstacles and regions of concealment, and evaluate its ability to detect positive (e.g., rocks) and negative (e.g., ditches) obstacles and concealment regions. We discuss the mapping system used for representing general obstacles (positive, negative, and concealment.) In addition to using information provided by the EBK sensor, the mapping algorithm also uses information supplied by a Global Positioning System (GPS) and an Inertial Navigation System (INS). This system has been tested at NIST and has successfully detected obstacles and regions of concealment while driving cross country at speeds of 35 km/h.

1. Introduction

The ability to detect and avoid obstacles is a prerequisite for the success of the Demo III Unmanned Ground Vehicles (UGV) program[1]. The near term goals of the project are to be able to drive the High Mobility Multipurpose Wheeled Vehicle (HMMWV) autonomously on cross-country roads with the ability to:

1. Drive autonomously at speeds up to 35 km/h, controlled as appropriate for the terrain and vehicle dynamics.
2. Detect obstacles and rough terrain conditions in time to enable deceleration to a safe speed or the ability to steer around the problem condition.

3. Maintain a control station with an interface for limited mission planning and data collection.

This paper discusses an obstacle detection algorithm developed at NIST in support of the concealment-detection and rough-terrain conditions. The algorithm is a hybrid of grid-based and sensor-based obstacle detection and mapping techniques. The perception and obstacle detection/mapping module is part of the integrated 4D-Realtime Control System (RCS) [1,2]. It consists of two sections: an obstacle detection section and a mapping section. The obstacle detection section processes range data read from a Ladar[1] sensor. The algorithm converts range data into Cartesian coordinates in the vehicle coordinate frame and uses this information to detect obstacles. The second section, the mapping module, projects obstacle points onto a grid-based map. The map is used by the 4D-RCS planner module [12] to generate a traversable path for the vehicle. We have demonstrated autonomous driving with obstacle detection and avoidance on the NIST grounds at speeds of up to 35 km/h.

Section 2 summarizes the sensors used in the 4D-RCS autonomous driving system. Section 3 describes the obstacle detection algorithm and evaluates its performance on natural obstacles. Section 4 briefly describes the obstacle mapping module. In Section 5, we present our conclusions and plans for future work.

2. System Sensors

The sensors used in the system include a Dornier 1.2 Hz Ladar Range Imaging Camera (EBK); the Ashtech Z12, a Global Positioning System (GPS) sensor, and an Inertial Navigation System (INS) sensor[1] [15]. Multiple navigation sensors are used. A Kalman filter [10] computes vehicle position and orientation using data from an inertial, dead-reckoning system and a carrier phase differential GPS unit. Additional physical information about the characteristics and references of EBK can be found in [3,4,9].

3. Obstacle Detection Module

The purpose of the obstacle detection module is to extract areas that cannot or should not be traversed by the HMMWV. Rocks, fences, trees, and steep upward slopes are some typical examples. Many approaches for extracting obstacles from range images have been proposed. The most common approach is to fit a plane surface to a patch of points [6,8,19] in a local grid representation. For example, Hebert [6] and Kelly [11] have used the approach of fitting planar surfaces to a patch of points to detect obstacles. Their algorithms are computationally intensive and have difficulty detecting small obstacles [18]. We propose using differential depth and slope for an improved, fast obstacle detector. The algorithm was inspired by Veatch and Davis [18] and Lux and Schaefer [13].

[1] Certain commercial equipment, instruments, or materials are identified in this paper in order to adequately specify the experimental procedure. Such identification does not imply recommendation or endorsement by NIST, nor does it imply that the materials or equipment identified are necessarily best for the purpose.

In general, obstacles are classified into three categories: positive, negative, and overhanging. Examples of positive obstacles are rocks, trees, fences, etc. Negative obstacles refer to depressions on the ground, i.e., ditches. Overhanging obstacles refer to objects that are located directly above the ground. A tree branch is a typical example in an outdoor environment.

3.1 Obstacle Detection Algorithm

The LADAR image consists of 128 vertical scan lines, each containing 64 range values. Approximately half of the acquisition time is needed for the Ladar's back-scan. We process range data as they are read, rather than waiting for a full image. In this way, we can detect an obstacle as soon as the data are available. As each scan line is read, a filter is applied to the data. The filter removes outlier points caused by specular reflections. After the filtering, the Cartesian (x,y,z) location of each pixel in the scan line is computed in a coordinate system centered at the vehicle center. The x, y, z and r (range) values assigned to each pixel are used to compute the depth derivatives and surface slopes.

3.1.1 Positive Obstacle Classification

The main criterion used for detecting positive obstacles is the surface slope. A positive obstacle is one whose surface slope is steep. This condition is expressed mathematically in the following way:

Let p(i,j) be the i^{th} position pixel in the j^{th} scan line of the EBK image such that p(0,0) is the *lower right* corner in the image.

Let x(i), y(i), z(i) be the Cartesian coordinates of p(i), the i^{th} pixel in a scan line.

Let g be the pixel index pointing to the newest labeled ground pixel, G, described in the next paragraph.

p(k) is a positive obstacle if:

$$\frac{(z(k)-z(g))^2}{(x(k)-x(g))^2+(y(k)-y(g))^2+(z(k)-z(g))^2} \quad ? \quad \sin^2(\alpha)$$

α is a predefined constant representing the maximum slope allowed, i.e., 30°. For efficiency, the constant $\sin^2(\alpha)$ is pre-computed.

The index g is used for tracking the ground pixel G. It is updated with the value of k if p(k) is not a positive obstacle and has a small elevation change. A positive obstacle can have a shallow slope.

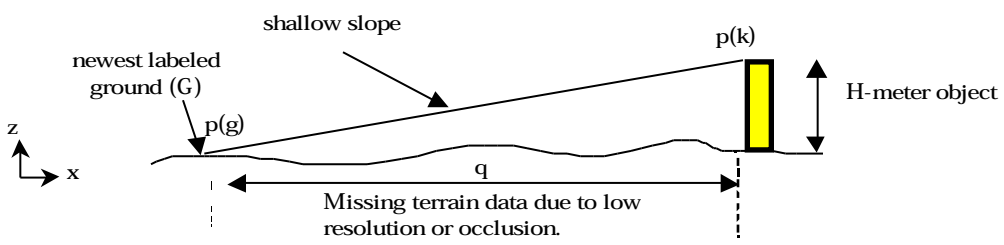


Figure 1. Positive obstacle with shallow slope

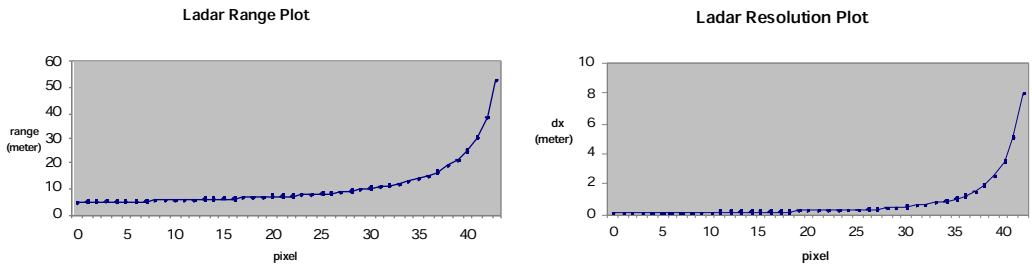
This is expressed mathematically as:

1. $$\frac{(z(k)-z(g))^2}{(x(k)-x(g))^2+(y(k)-y(g))^2+(z(k)-z(g))^2} < \sin^2(\alpha) \quad \text{and}$$

2. $$|z(k)-z(g)| < H$$

H is a pre-defined constant representing the maximum allowed ground discontinuity. Currently we assign H a value of 1.0 meter and assume that ground pixels are present in all scan lines.

In general, the slope in Figure 1 falls as q increases. Consider a Ladar image of flat terrain, the difference in the x-coordinates between successive pixels is an increasing function. Figure 2a shows the plot of the range in the x direction versus the pixel number of a Ladar scan line. Figure 2b shows the resolution of a single Ladar scan line. In this plot, dx represents the difference in value in the x direction.



(a) (b)

Figure 2. Ladar range

Because of decreasing resolution, the terrain visible in the Ladar image can be thought of as composed of two regions: low-resolution and high-resolution. The high-resolution portion of the image contains densely sampled terrain points and the low-resolution region represents coarse sampling of the terrain.

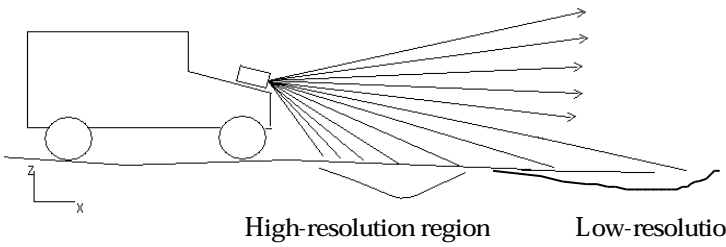


Figure 3. Sensing flat ground

3.1.2 Overhanging Cover Classification

In addition to detecting positive obstacles, the algorithm also examines "overhanging" regions. We assume that regions labeled "overhanging" are safe (or concealment) areas. These areas are referred to as "cover". If this isn't true, the regions are labeled as being obstacles. Currently, we use an object's height as the main discriminating factor for cover classification. Specifically, pixels representing positive obstacles with greater than a pre-defined height are classified as cover pixels. Some cover pixels may correspond to the upper region of a tall

object and it may be dangerous or even impossible for the HMMWV to go beneath it. These pixels may not represent true regions of cover in which we are interested. At the sensory processing level, it is not necessary that "false" cover pixels be sought and eliminated. They will be rejected by the mapping system described in Section 4. We could also label regions of "cover" by looking for pixels that are closer to the vehicle than its closest ground p(g) (see Figure 4). This method was implemented in earlier versions of the algorithm, but is no longer used because the same information is obtained from the mapping system.

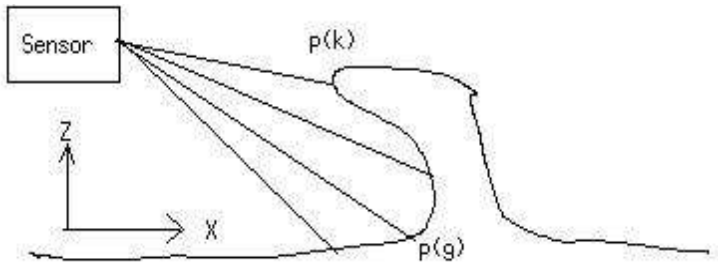


Figure 4. Cover at p(k)

Estimating an obstacle's height becomes critical for cover detection. The vehicle must detect areas of cover accurately from a distance because the cover may suddenly disappear from the sensor's field of view as the vehicle moves toward it. At that time, the cover might be above the vehicle. If the cover is actually lower than first detected and the sensor did not get additional data about it, a collision could occur.

Currently, our height estimation algorithm is based on the simple z-difference calculation. However, instead of taking the Z-difference with respect to the closest ground, we use the closest *flat* ground. We say a ground pixel is *flat* if it has the following property:

Let p(k-1), p(k-2) and p(k-3) be three ground pixels (see Figure 5).

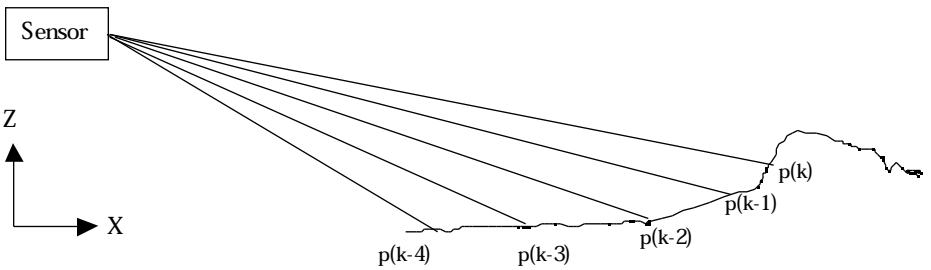


Figure 5. Flat ground at k-2; newest-labeled ground at k-1; j=2

$$p(k-j) \text{ is flat if } \left(x(k-j) - x(k-j-1)\right) > \left(x(k-j-1) - x(k-j-2)\right) \quad \text{for some } j > 0$$

The flat ground is not used in detection of positive obstacles. It is only used to improve the estimate of an object's height.

3.1.3 Negative Obstacle Classification

Negative obstacle detection presents a harder task at the sensory processing level. Unlike positive obstacles, negative obstacles generally do not have steep downward slopes because of occlusion. One possible solution would be to mount the sensor higher and increase the tilt angle. This would allow the slope to be a detecting factor, but it would also reduce the sensing area. Ground pixels just before the ditch can be easily detected as a large negative discontinuity in elevation. However, to reduce the number of false alarms, ditches are not considered until they are inside the high-resolution portion of the Ladar image. For our sensor mounting, this region is within 20 m from the vehicle. This range may not be sufficient for driving under high speed. It is helpful to understand that the minimum stopping distance required for a vehicle travelling at 8 km/h is approximately 5 meters. At 16 km/h, the stopping distance is 10 m, and at 24 km/h, it is 20 m (see Table 2 in Section 3.2.1). The far walls of a ditch, however, are positive obstacles and can be detected much further away.

3.2 System Performance Evaluation

Two types of tests were performed to evaluate the obstacle detection algorithm. Both tests involved collecting sets of Ladar data, which were analyzed at a later time. The data was collected at the Perryman site of the Aberdeen Proving Grounds (APG) in Aberdeen, Maryland. The surveyed area included rocks, trees, and ditches.

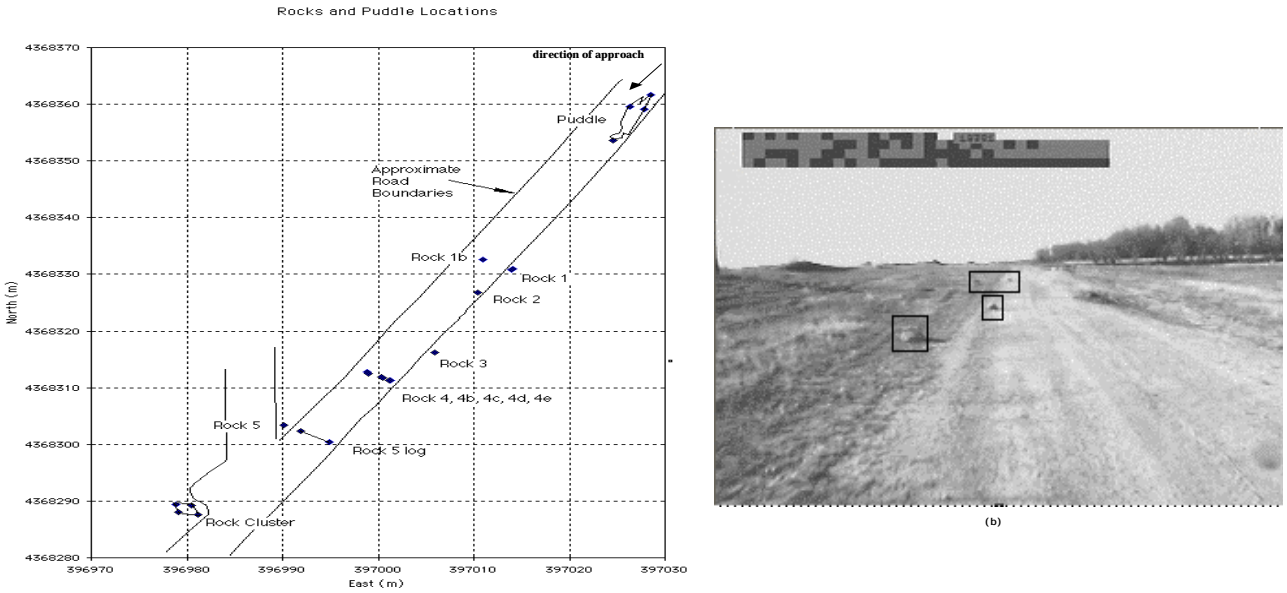
3.2.1 Positive Obstacles

Ideally, we would like to detect a positive obstacle when a single Ladar pixel hits that obstacle. This is the case when the corresponding, newest, labeled ground (G) is less than some value q (see Figure 1). In the low-resolution region of the image, we may need at least two pixels hitting the obstacle because the first pixel might get misclassified as a ground pixel. It is clear that detectability of small positive obstacles depends greatly on where they lie in the image and where ground points are detected. In particular, we are interested in learning the distances at which the algorithm can detect small obstacles. Some positive obstacles may be harmless. For example, short grass and pebbles should be ignored due to their small dimensions.

Figure 6a is a map of the Perryman area containing some large rocks. The rocks were placed both on-road and off-road. The rocks labeled "rock 1" and "rock 2" are approximately 15 cm high and 25 cm wide. Rocks 3 and 4 are approximately 30 cm high and 30 cm wide. Rock 5 is approximately 45 cm high and 30 cm wide. The rock cluster is approximately 45 cm high and 100 cm wide. Table 1 summarizes the rock data.

	rock 1	rock 2	Rock 3	rock 4	rock 5	Cluster
height (cm)	15	15	30	30	45	45
width (cm)	25	25	30	30	30	100

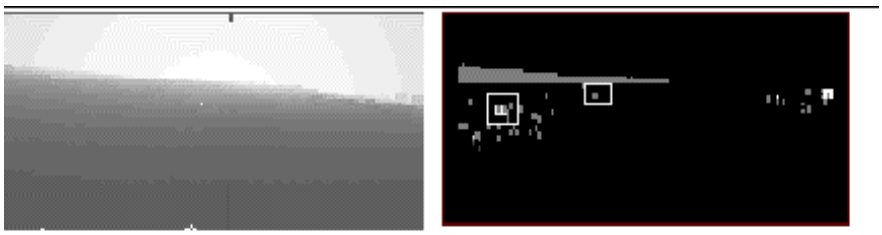
Table 1. Rock data



(a)

Figure 6. Perryman Survey of Rock Obstacles

Figure 6b is a video image showing rocks 1, 2, 3 and 4. Figure 7a shows a Ladar scene of rocks 1 and 2 and Figure 7b shows the obstacles detected in this image. Rocks 1 and 2 are highlighted by white boxes in the right image.



(a)Raw ladar data

(b) Rock 1 at a distance of 11.5m
Rock 2 at a distance of 19.2 m

Figure 7. Rock 1 and 2

Several metrics can be used to measure the performance of the system. The goal of the obstacle detection system is to detect an obstacle far enough ahead to provide time for braking or avoidance maneuvers. We use the "detected-in-time" metric to evaluate whether the system meets this goal. Other metrics could be based on error rate, i.e., false positives, false negatives and true positives. The "detected-in-time" metric is defined as:

```
If      Detected_Range ≥ Stop_Distance   THEN
        Detected_In_Time = YES;
Else
        Detected_In_Time = NO;
```

The quantity Detected_Range is the range at which an obstacle is first detected and the quantity Stop_Distance is the distance required to stop the vehicle given the current vehicle speed. The metric Detected_In_Time is a binary

value that tells whether the system was able to detect an obstacle in time for the vehicle to stop. The metric evaluates the sensor and the obstacle detection algorithms. If results indicate the obstacle is not detected in time, then apart from changing the sensor or algorithms, the vehicle speed should be decreased. Referring to the metric, the Stop_Distance has two components:

1. Reaction_Distance is the distance traveled from the time the obstacle is detected until the brake is fully depressed (emergency stop). It is computed as the product of Reaction_Time and the current speed. Reaction_Time is the sum of the planning time to decide to perform an emergency stop and the actuation time to depress the brake pedal fully.

2. Braking_Distance is the distance traveled from the time the brake pedal is fully depressed until the time the vehicle stops. It is computed as: $\text{Braking_Distance} = \left(\frac{-(\text{Speed})^2}{2\ \text{Acceleration}}\right)$

The vehicle can be driven at different speeds and the terrain and soil conditions will affect the deceleration capability of the vehicle. Table 2 gives the empirical stopping distance at the speeds used for our data collection while varying the vehicle's deceleration capability. For this evaluation, we set the Reaction_Time to a conservative value of 1 second and used a $-2\ m/s^2$ deceleration.

Driving speed	Stopping distance (m) at $-2\ m/s^2$	Stopping distance (m) at $-3\ m/s^2$	Stopping distance (m) at $-4\ m/s^2$
8 km/h	3.5 m	3.1 m	2.9 m
16 km/h	9.4 m	7.8 m	7.0 m
24 km/h	17.8 m	14.1 m	12.3 m

Table 2. Stopping distances based on speed and time to stop

Table 3 shows the evaluation results of the system's ability to detect positive obstacles. For each of the three speeds, the distance at which the obstacle was detected (Detected_Range) is listed. Below each range value is a YES or NO indicating the result of the "detected-in-time" metric. For example, rock 1 was detected in time at 8 km/h and at 16 km/h, but not at 24 km/h. The results from the data suggest that the system is able to detect obstacles in time at speeds up to 16 km/h but obstacles are missed at 24 km/h. At a Reaction_Time of 0.25 seconds, no obstacle would be missed.

Reaction_Time = 1 second	*Rock1*	*Rock2*	*Rock3*	*Rock4*	*Rock5*	*Rock Clump*	*Stopping distance (m) at $-2\ m/s^2$ deceleration*
8 km/h	13.9 m	20.2 m	18.6 m	25.0 m	12.8 m	31.8 m	3.5 m
Detected in time?	YES	YES	YES	YES	YES	YES	
16 km/h	13.2 m	20.5 m	22.3 m	24.1 m	17.0 m	26.0 m	9.4 m
Detected in time?	YES	YES	YES	YES	YES	YES	
24 km/h	15.0 m	24.3 m	22.2 m	26.6 m	13.7 m	30.6 m	17.8 m
Detected in time?	NO	YES	YES	YES	NO	YES	

Table 3. Result of positive obstacle detection performance evaluation

The variability of detection is a result of the following factors:

(1) Data acquisition rate (1 image per second).

(2) Relationship between vehicle velocity and the image resolution. If the vehicle is moving quickly, the scanning Ladar beam may miss the obstacle completely.

(3) Difficulty in identifying the targeted obstacle when it is hidden in tall grass.

The need for recognizing and classifying obstacles is clear from the outputs shown in these figures. In addition to the rocks and/or rock cluster, the algorithm has flagged tall grass, bushes, and background trees as obstacles. Our future work includes plans to fuse Ladar images with video images to generate windows of interest in the video images. Classification of obstacles will be based on the information in the video windows.

3.2.2 Negative Obstacles

In this section, we describe our analysis of negative obstacles. Our objective is to determine the farthest range at which pixels representing depressions[1] can be detected. In addition to the Ladar data, sets of images from 3 pairs of stereo cameras were collected.[1] The stereo data were analyzed by scientists at NASA's Jet Propulsion Laboratory (JPL) using algorithms designed for stereo inputs [14]. This report discusses only the results of the Ladar obstacle detection algorithm.

Figure 8 is a map of the sections of Perryman in which negative obstacle (ditch) data were collected. The ditches were dug by a back hoe. Ditch 1 is 0.61 m wide; Ditch 2 is 1.22 m wide, Ditch 3 is 1.83 m wide, and Ditch 4 is 2.44 m wide. The lengths of all the ditches are the same. These data are used to evaluate the effectiveness of the algorithm in detecting negative obstacles (depressions in the ground plane).

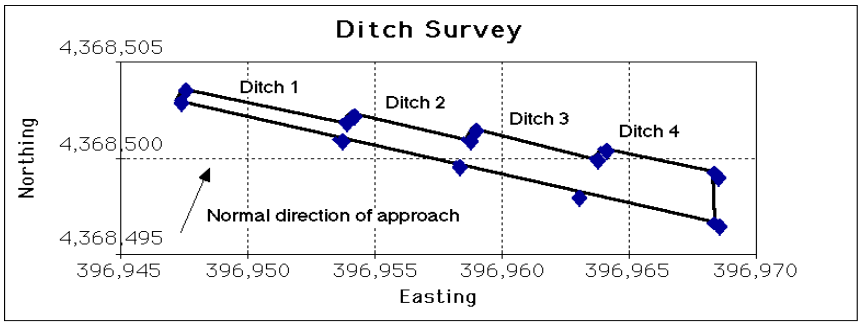


Figure 8. Ditch Survey Site

Data were collected at different times of day under varying weather conditions. Based on our experiments, a decrease in daylight did not affect the Ladar data which performed at the same level during dawn, daylight, and evening hours. For each of the ditches, the HMMWV was driven at speeds of 8 kilometers per hour (km/h), 16 km/h, and 24 km/h. The vehicle's approach to the ditches also varied: some runs were perpendicular to the

[1] Walls of the ditch.

ditches; some were oblique. The output of the obstacle detection algorithm is qualitatively evaluated by an experienced user.

Figure 9 shows two images taken from the "ditch 1" data set: the unprocessed Ladar image (Figure 9a) and the processed image (Figure 9b). The Ladar image is encoded as grey-scale values such that darker grey values represent ranges farther from the sensor. White pixels in the processed image represent pixels labeled as obstacle points. Although it is very difficult to recognize the ditch in the raw image, the ditch pixels are detected in the processed image. The ditch is 10 m from the vehicle in Figure 9.

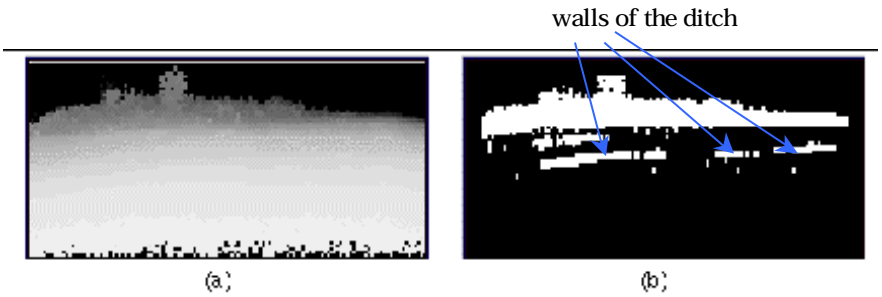


Figure 9. "Ditch 1" Raw and processed ladar data

Table 4 shows the results of the obstacle detection algorithm on the ditch data as the vehicle approached the ditches in the normal direction. The distances, measured in meters, represent the farthest distance at which the algorithm extracted pixels representing ditch data. The pixels associated with the ditch are qualitatively recognizable to a trained eye.

The performance of negative obstacle detection is very similar to that of positive obstacle detection. This is because the far wall of the ditch is itself a positive obstacle.

Driving speed	*Ditch1*	*Ditch2*	*Ditch3*	*Ditch4*	*Stopping distacne at –2 m/s^2 deceleration*
Speed = 8 km/h	10.0 m	19.0 m	17.0 m	23.0 m	3.5 m
Detected in time?	YES	YES	YES	YES	
Speed = 16 km/h	10.0 m	14.0 m	20.0 m	20.7 m	9.4 m
Detected in time?	YES	YES	YES	YES	
Speed = 24 km/h	9.5 m	15.0 m	20.5 m	21.2 m	17.8 m
Detected in time?	NO	NO	YES	YES	

Table 4. Results of negative obstacle detection performance evaluation

[1] A complete description of the data collection can be found at:
http://www.isd.cme.nist.gov/personnel/coombs/proj/mobility/data/apg-data/doc/datanotes.html

3.3 Overhanging obstacles

In this section, we describe the analysis of overhanging obstacles. Data were collected while driving at 8 km/h toward trees of unknown height. Our objective was to verify the algorithm's accuracy in determining an obstacle's height.

Figure 10a shows a Ladar image of trees at 45 meters while driving at 8 km/h. Although the trees are barely detectable by the human eye in the Ladar image, they are detected and classified in the corresponding processed image (Figure 10b). The light-grey regions on the processed image indicate tree branches higher than 2 meters. The dark-grey regions represent positive obstacles shorter than 2 meters. Note that some pixels are misclassified as ground pixels (mid-grey). This is due to the decreasing resolution discussed in sections 3.1.1. The white pixels represent either invalid data or positive obstacles of negligible dimension, i.e., pebbles and short grass.

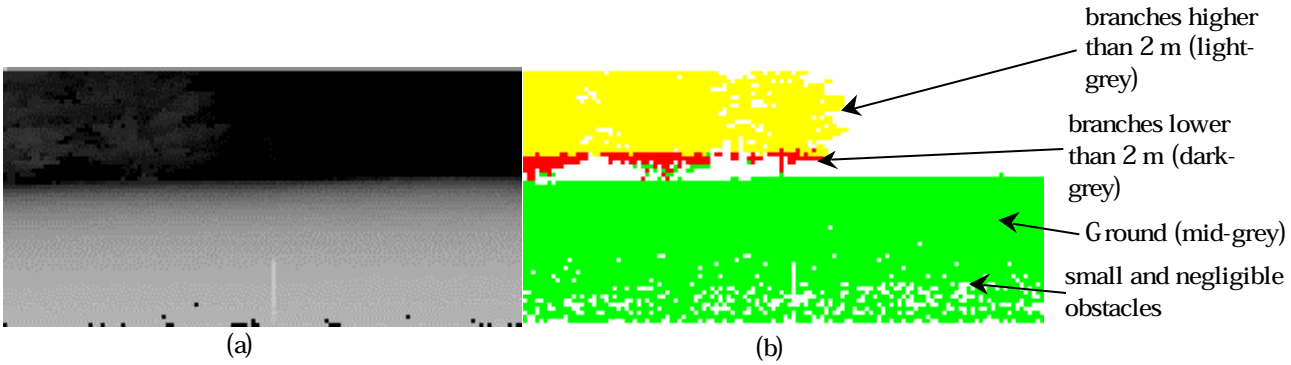


Figure 10. Trees at 45m

4. Map and Object Module

We adapted Kelly's [11] grid obstacle map for representing obstacles for path planning and vehicle control. The map representation contains terrain surface slopes, uncertainty measures, average elevation information, and a list of obstacles for every grid cell on the map. In some cases, a grid/map cell may contain both overhanging and positive obstacles. We give positive obstacles higher precedence and ignore overhanging obstacles. This has the effect of rejecting "false cover", as defined in Section 3.1.2.

The path-planning algorithm currently implemented in the 4D-RCS system [1] uses only a subset of the information contained in the map. It generates traversability paths from the map by using a binary flag indicating whether or not the grid cell contains an obstacle, a timestamp indicating when an obstacle was last updated, and an obstacle confidence value.

4.1 The World Model Representation

The world representation is based on the 2_-D model. In this model, the world is projected onto a ground plane with a fixed-size grid. Each grid cell contains the information mentioned in Section 4. This representation is

fast and easy to maintain when compared with typical tree data structures. Efficient use of memory can be achieved by allocating space only to grid cells that contain data.

4.2 The scrolling local map

We use an efficient, scrolling, local map that updates new sensor data while keeping the vehicle centered on the map. This approach has the advantage of minimizing grid relocation – no copying of data is done; only updating. Figure 11 shows a local map and its corresponding obstacle image. The light-grey regions represent areas of overhanging obstacles. The mid-grey regions represent detected ground and the dark-grey pixels positive obstacles. The dashed-line represents the path travelled by the vehicle.

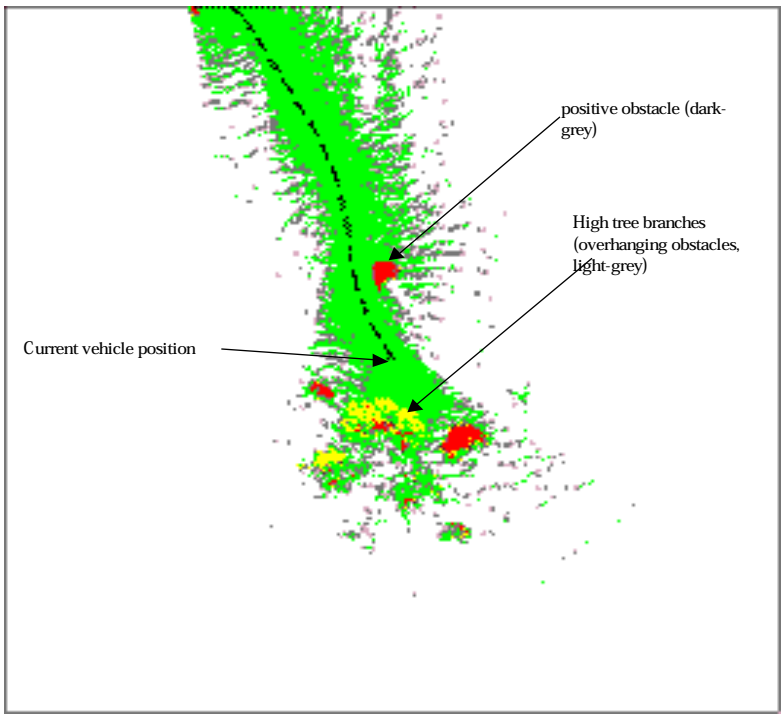


Figure 11. Local scrolling map.

4.3 Confidence-based Mapping

Our map updating algorithm is based on the concept of confidence-based mapping described in Oskard's paper [16]. In this algorithm, the confidence increases/decreases linearly when updated by the sensor. When a map cell receives a vote for "obstacle", the cell's obstacle confidence goes up by a predefined constant. Only when the confidence exceeds a certain threshold will the map cell get marked as "obstacle". A similar algorithm is used for marking "cover" map cells.

When a map cell receives a conflicting vote, the associated confidence goes down. Although this mechanism tends to resist noisy data from the sensor, it assumes a fairly static environment. When an obstacle moves, an obstacle trail is left on the map. This obstacle trail eventually fades as the confidence drops; however, there are instances where the confidence may not drop. For example, if the trail suddenly gets shifted outside the field of view

because of vehicle motion, no new updates will arrive to change the trail status. A similar situation occurs when the vehicle remains stationary with an object moving in the low-resolution region in the Ladar image. The map cell containing the moving obstacle's original position may not get updated again; the Ladar scan may simply miss that location due to low resolution. In order to fully address this problem, a moving-obstacle detection module is required. Our current approach for dealing with an obstacle trails is to decrease the reaction time needed to change their classification.

4.4 The Object representation

Object and feature descriptions are organized in a table indexed by their types and by their map locations. This data structure is very similar to the one described by Shneier [17] in which objects are indexed spatially by a pointer associated with each node in the octree. Additional entry fields in our representation include object type, velocity, GPS (Global Positioning System) position and dimension.

5. Future work and discussion

As noted in section 4.2, confidence-based mapping may not be adequate for tracking moving obstacles. Currently, we do not utilize a hypothesis/prediction model for predicting the motion of moving obstacles. We believe such an approach may be useful for maintaining knowledge of local obstacles at the sensory processing level. This knowledge could be used to improve detection accuracy and could also be used to detect moving obstacles.

We are also considering using temporal integration to address the problem of low-resolution Ladar images. In this way, we could use additional terrain information from overlapping regions from previous Ladar images. We believe that most of the missing terrain due to low-resolution could be filled and could be used during obstacle detection.

The current obstacle detection algorithm does not model the ground. Although this algorithm is fast, adequate, and provides satisfactory results, queries regarding terrain roughness or any terrain-related measure simply can not be answered. Our future obstacle detection algorithm will include a recursive least square filter for ground modeling. We believe a more accurate ground model will result in a more accurate estimation of an obstacle's height.

6. Conclusion

We have described an algorithm designed to detect obstacles in Ladar imagery. The algorithm is one of a suite of algorithms being tested in order to evaluate different sensors that might be used for obstacle detection in the Unmanned Ground Vehicle program. We have addressed the need for a moving-obstacle detection module and

discussed various improvements for future implementation. We have characterized the Ladar sensor being used, and described its performance on different types of surfaces.

In order to evaluate the obstacle detection algorithm, we collected extensive sets of Ladar data and defined a performance metric. The data were used as input for the obstacle detection algorithm. The algorithm was able to detect positive obstacles (e.g., rocks), overhanging areas (e.g., tree branches), and negative obstacles (e.g., ditches) with varying degree of success. Detectability was determined to be a function of both the range to the obstacle and the size of the obstacle. The results of the evaluation tests were reported. Conservative evaluations indicated current Ladar (1.2 Hz) limits vehicle speed to 16 km/h.

The obstacle detection algorithm was also tested in a real-time demonstration as part of the 4D-RCS control system. In conjunction with the planner module and behavior generation modules [1], we were able to detect obstacles on both the NIST grounds and the NIKE test site in Gaithersburg, Maryland.

7. Acknowledgments

This work was sponsored by the Joint Robotics Program of the Under Secretary of Defense for Acquisition and Technology (JRP OUSD(A&T)) and monitored by the U. S. Army Research Laboratory. We thank the members of the Intelligent Autonomous Vehicles Group at NIST, ARL, the Toole Army Depot, the Aberdeen Test Center, and JPL for their efforts in the Perryman data collection. In particular, we thank Maris Juberts, Martin Herman, David Coombs, Billibon Yoshimi, Karl Murphy, Jim Gilsinn, Larry Matthies, Todd Litman, Ken Owens, Tim Vong, Chuck Shoemaker, Jon Borenstein, Gary Haas, Fred Eldridge, Ish Sherali, and Bob Brooks. We thank Sandor Szabo for his helpful suggestions and comments. We apologize to anyone inadvertently left off this list.

8. Reference

1 Albus, James, "4-D/RCS Version 1.0, A Reference Model Architecture for Demo III," TBP.

2 S. Baten, R. Mandelbaum, M. Lutzeler, P. Burt, E. D. Dickmanns, "Area-based stereo image processing within the 4D architecture for autonomous off-road navigation," AutoNav Meeting, FL, January, 1998.

3 Dornier GmbH, "Evaluation Package User's Guide."

4 Dornier GmbH, "Transformation of EBK range Images to 3-D Images in a Sensor Coordinate System."

5 Hebert , M. and Kanade, T. , "First Results on Outdoor Scene Analysis Using Range Data," Proceedings: Image Understanding Workshop (Miami Beach, FL, December 9--10, 1985), pp. 224-231, Morgan Kaufmann, 1985.

6 Hebert, M. and Ponce, J., "A New Method for Segmenting 3-D Scenes into Primitives," 6th International Conference on Pattern Recognition, pp836-838, Oct. 1982.

7 M. Hebert, C. Thorpe, and A. Sentz, Intelligent Unmanned Ground Vehicles, Autonomous Navigation Research at Carnegie Mellon, Kluwer Academic Researchers, 1997.

8 Hoffman, R. and Jain, A.K., "Segmentation and Classification of Range Images," PAMI(9), No. 5, pp. 608-620., Sept. 1987.

9 Hong, T., Legowik, S., Nashman, M., "Obstacle Detection and Mapping System", NISTIR 6213, Auguest, 1998.

10 Kalman, R.E., "A new approach to linear filtering and prediction problems," Trans. ASME, Series D, J. Basic Eng., V. 82, pp. 35 - 45 , 1960.

11 Kelly, A., "An Intelligent, Predictive Control Approach to the High-Speed Cross-Country Autonomous Navigation Problem," Carnegie Mellon University, 1995.

12 Lacaze, A., Albus, J., Meystel, A., "Planning in the Hierarchy of NIST-RCS for Manufacturing," Proceedings of the International Conference on Intelligent Systems: A Semiotic Perspective, Gaithersburg, MD, October 20-23, 1996.

13 Lux, P.W. and Schaefer, C.H., "Range Imaging for Autonomous Navigation of Robotic Land Vehicles," Dornier Lufthahrt, Signal Processing 22, pp299-311, 1991.

14 Matthies, L., Litwin T., Owens, K. Rankin, A., et. al., "Performance Evaluation of UGV Obstacle Detection with LADAR and CCD/FLIR Stereo Vision," Proceedings of SPIE AeroSense Conference, April, 1998.

15 Murphy, K., Legowik, S., "GPS Aided Retrotraverse For Unmanned Ground Vehicles," Proceedings of the SPIE 10th Annual AeroSense Symposium, Conference 2738, Navigation and Control, Technologies for Unmanned Systems, Orlando, FL, April 1996.

16 Oskard, D., Hong, T., Shaffer, C., "Real-time Algorithms and Data Structures for Underwater Mapping," National Bureau of Standards, 1990.

17 Shneier, M., Kent, E. and Mansbach, P., "Representing workspace and model knowledge for a robot with mobile sensors," National Bureau of Standards, 1984.

18 Veatch, P. A. and Davis, L. S., "Efficient Agorithms for Obstacle Detection Using Range data," Computer Vision, Graphics, and Image Processing, 50(1), pp. 50-74, April 1990.

19 Yang, H.S. and Boyer, K.L. and Kak, A.C., "Range Data Extraction and Interpretation by Structural Light," pp 199-205, Conference on Artificial Intelligence Applications, 1984.