

## Performance Analysis of Unmanned Vehicle Positioning and Obstacle Mapping

Roger Bostelman, Tsai Hong, Raj Madhavan, Tommy Chang, and Harry Scott

National Institute of Standards and Technology  
Gaithersburg, MD 20899-8230

{roger.bostelman, tsai.hong, raj.madhavan, tommy.chang, harry.scott}@nist.gov

**Abstract:** As unmanned ground vehicles take on more and more intelligent tasks, determination of potential obstacles and accurate estimation of their position become critical for successful navigation and path planning. The performance analysis of obstacle mapping and unmanned vehicle positioning in outdoor environments is the subject of this paper. Recently, the National Institute of Standards and Technology's (NIST) Intelligent Systems Division has been a part of the Defense Advanced Research Project Agency LAGR (Learning Applied to Ground Robots) Program. NIST's objective for the LAGR Project is to insert learning algorithms into the modules that make up the NIST 4D/RCS (Four Dimensional/Real-Time Control System) standard reference model architecture which has been successfully applied to many intelligent systems. We detail world modeling techniques used in the 4D/RCS architecture and then analyze the high precision maps generated by the vehicle world modeling algorithms as compared to ground truth obtained from an independent differential GPS system operable throughout most of the NIST campus. This work has implications, not only for outdoor vehicles but also, for indoor automated guided vehicles where future systems will have more and more onboard intelligence requiring non-contact sensors to provide accurate vehicle and object positioning.

**Index Terms** —world modeling, global positioning system, ground truth, performance analysis, stereo vision, mapping.

### I. INTRODUCTION

The National Institute of Standards and Technology's (NIST) Intelligent Systems Division (ISD) has been developing the RCS [1, 2] reference model architecture for over 30 years. 4D/RCS is the most recent version of RCS developed for the Army Research Lab Experimental Unmanned Ground Vehicle program. ISD has studied the use of 4D/RCS in defense mobility [3], transportation [4], robot cranes [5], manufacturing [6, 7] and several other applications.

In the past year, ISD has been applying 4D/RCS to the DARPA LAGR Program [8]. The DARPA LAGR program aims to develop algorithms that enable a robotic vehicle to travel through complex terrain without having to rely on hand-tuned algorithms that only apply in limited environments. The goal is to enable the control system of the vehicle to learn which areas are traversable and how to avoid areas that are impassable or that limit the mobility of the vehicle. To accomplish this goal, the program provided small robotic vehicles to each of the participants (Figure 1). The vehicles are used by the teams to develop software and a separate DARPA team, with an identical vehicle, conducts tests of the software each month to compare against the DARPA baseline controller software. DARPA operators load the software onto an identical vehicle and command the vehicle to travel from a start position to a goal waypoint through an obstacle-rich environment. They measure the performance of the system on multiple runs, under the expectation that improvements will be made through learning. Developing high quality obstacle maps is therefore, critical to vehicle performance and learning.

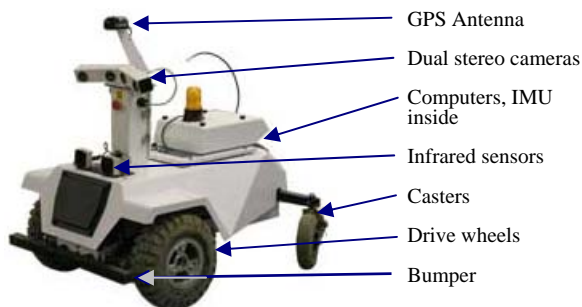


Figure 1. The DARPA LAGR vehicle

The vehicles are equipped with four computer processors (right and left cameras, control, and the planner); wireless data and emergency stop radios; GPS receiver; inertial navigation unit; dual stereo cameras; infrared sensors; switch-sensed bumper; front wheel encoders; and other sensors.

Section II of this paper describes in detail, the 4D/RCS World Model (WM) followed by a detailed description of the color image classification in Section III. NIST's world modeling performance is then discussed in section IV with differential GPS (ground truth) being compared to the onboard vehicle mapping system. Section V includes conclusions and suggested future work.

## II. WORLD MODEL

The world model is the system's internal representation of the external world. It acts as a bridge between sensory processing (SP) and behavior generation (BG) in the 4D/RCS hierarchy by providing a central repository for storing sensory data in a unified representation. It decouples the real-time sensory updates from the rest of the system. The WM process has two primary functions:

1. To create a knowledge database (map) and keep it current and consistent. In this role, it updates existing data in accordance with inputs from the sensors, and deletes information no longer believed to be representative of the world. It also assigns (multiple) confidence factors to all map data and adjusts these factors as new data are sensed. The types of information included in the map are state variables (e.g., time, position, orientation), system parameters (e.g., coordinate transforms, sensor-to-vehicle offsets, etc.), and lists or classes of sensed objects. The WM process also provides functions to update and fuse data and to manage the map (e.g. scrolling)
2. To generate predictions of expected sensory input based on the current state of the world and estimated future states of the world. For the LAGR autonomous, off-road, learning application, very little *a priori* information is initially available to support path planning between the vehicle's current position and a final goal position. However, as it learns, the world model memorizes, constructs and maintains all the information necessary for intelligent sensing.

The objective of the LAGR project is to have a small vehicle reach its off road navigation goal within 5 m on a 100 m course. To achieve this goal, we use two pairs of stereo camera systems, two infrared sensors and bumpers along with onboard GPS and inertial guidance systems. Two levels of 4D/RCS are implemented as described in [8]. Each level of the world model obtains a different map resolution which temporally fuses information from each sensory processing module. The sensory processing modules include a stereo obstacle detection module, a bumper obstacle detection module, an infrared obstacle detection module, an image classification module, a slippery detection module, and a learning module.

Each level, World Model 1 and 2 (WM1 and WM2) of the LAGR world model contains a different resolution of a two dimensional array (200 cells x 200 cells). Each cell in the map contains the fused information extracted from processed sensor data. Figure 2 shows two different resolutions of maps constructed from the stereo obstacle detection module in the world model.

The position of the vehicle is shown on the map as a green box with a white triangle protruding from it to signify stereo cameras field-of-view; the red, yellow, blue, light blue, and green are cost values ranging from high to low cost, respectively; and the black represents unknown areas. The violet, skewed-rectangle represents the local vehicle plan while the white line represents the long range plan with the dots being planned waypoints. The orange line is the direction to the goal.

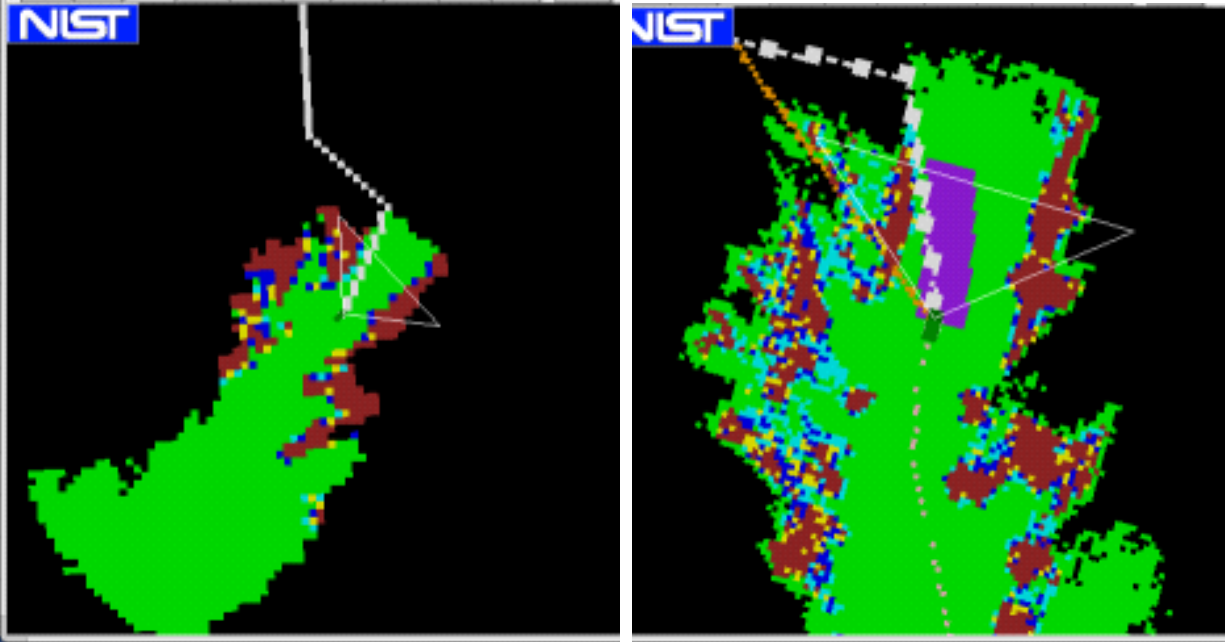


Figure 2: Cost Maps with 0.6 m resolution (left) and 0.2 m resolution (right). Obstacle costs = 0 (lowest cost - green) through red (highest cost - 256).

Each map grid cell represents an area defined by the header grid size and is marked with the time it was last updated. The total extent of the map is 40 m for WM1 and 120 m for WM2. The information stored in a cell includes:

1. The average ground and obstacle elevation height; the variance, minimum and maximum height; and a confidence measure reflecting the “goodness” of the elevation data.
2. A data structure describing the terrain traversability cost, confidences of the cost which are updated by the stereo obstacle detection module, image classification module, bumper module, infrared module, etc. The costs and the confidences are combined to determine the relative safety of traversing the grid with the following equation:

$$Cost_{cell} = W_s * Cost_{stereObs} + W_l * Cost_{lagrLearn} + W_c * Cost_{classification}$$

where  $Cost_{cell}$  is the cost to traverse for each grid cell.

$Cost_{lagrLearn}$  is the fused cost in the world model based only on the output from the lagr stereo module.

$Cost_{classification}$  is the fused cost in the world model based only on the output from the classification detection module.

And  $W_{s,l,c}$  is the weighting constant on each cost.

The final cost represents the best estimate of the terrain traversability based on information fused over time. Each cost has a confidence associated with it, and the world modeling selects the map grid label with the highest confidence. The final cost maps are constructed by taking each fused cost from all the sensory processing modules and calculating the smooth cost maps which indicates the cost is function of distance to the obstacles.

### A. Maintaining and Updating the World Model

The world model map must be maintained and updated in a timely manner. World model functions have been developed to scroll the map as the vehicle moves, to update map data, and to fuse data from the sensory processing modules with some uncertainty.

A scrolling, local map is updated with new sensor data while keeping the vehicle centered on the map. This approach has the advantage of minimizing grid relocation. For efficiency, no copying of data is done; only updating. When the vehicle moves out of the center grid cell of the map, the scrolling function is enabled. The scrolling function includes re-centering the map and reinitializing the map borders. Because the map is vehicle-centered, only the borders of the map contain new regions that must be re-initialized. The initialization information can be obtained from remembered maps which are saved from previous test runs. However, as shown in Figure 3 (left), the first run will have only information very local to the vehicle at startup. Figure 3 (right) shows a map saved from a previous vehicle run from start to the goal with vehicle ready to begin another run.

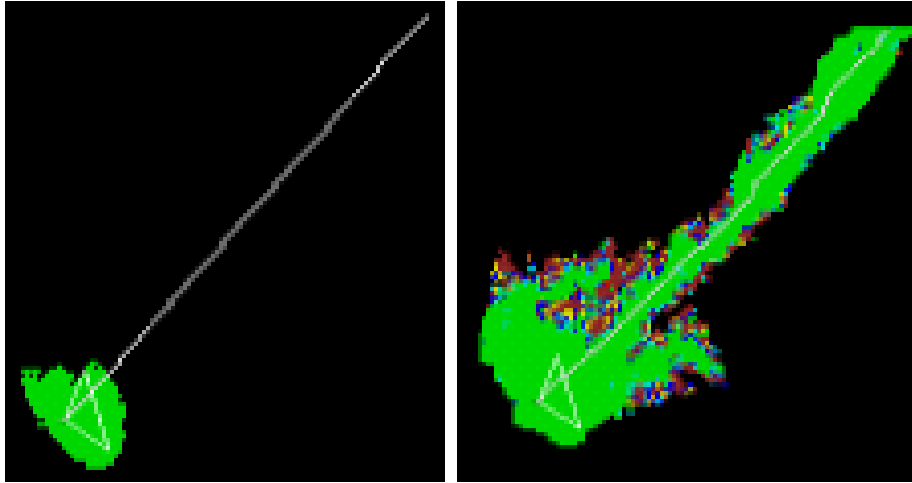


Figure 3. Left: Initialization map without previously saved obstacle information. Right: Map with previously saved obstacle information being passed on to the next vehicle run.

The map updating algorithm is based on the concept of confidence-based mapping described in [9]. In this algorithm, confidence measures increase or decrease linearly as the model receives updated information from the sensors. When a map cell receives a vote for a class such as an obstacle, an elevation measurement, a terrain classification, etc., the cell's confidence in that class is incremented by a predefined constant.

The cost to traverse each cell (region) is updated and temporally fused based on the traversability measurements which are computed from a learning module [10], stereo range data [11], infrared and physical bumpers. The obstacle's confidence increases by an empirically predefined weight constant.

The cost and elevation confidence of each grid cell are updated every sensor cycle: e.g., 5 Hz for the stereo obstacle detection module, 3 Hz for the learning module, 5 Hz for the classification module and 10 Hz -20 Hz for the bumper modules. The confidence values are used as a cost factor in determining the traversability of a grid.

### III. COLOR IMAGE CLASSIFICATION

The basic idea of the color image classification algorithm is to determine traversable areas using a road segmenting algorithm as proposed in [12]. We have modified the algorithm to not only segment the road but to segment the image based on a color model. We first represent the color distributions seen in the ground and the background. A 30 x 30 histogram of red and green for the color distribution has given the best results in the color modeling of ground and background in the LAGR environments.

The most critical assumption for the approach is that the area in front of the vehicle is good/safe to traverse. For most driving this is the usual case. A white, trapezoidal region (see Figure 4 (left, bottom)) is assumed to be good/safe ground for traversing and is histogrammed at each frame to construct the initial ground model. The trapezoidal region is the projection of a 1 m wide by 2 m long area of ground in front of the vehicle under a ground plane assumption.

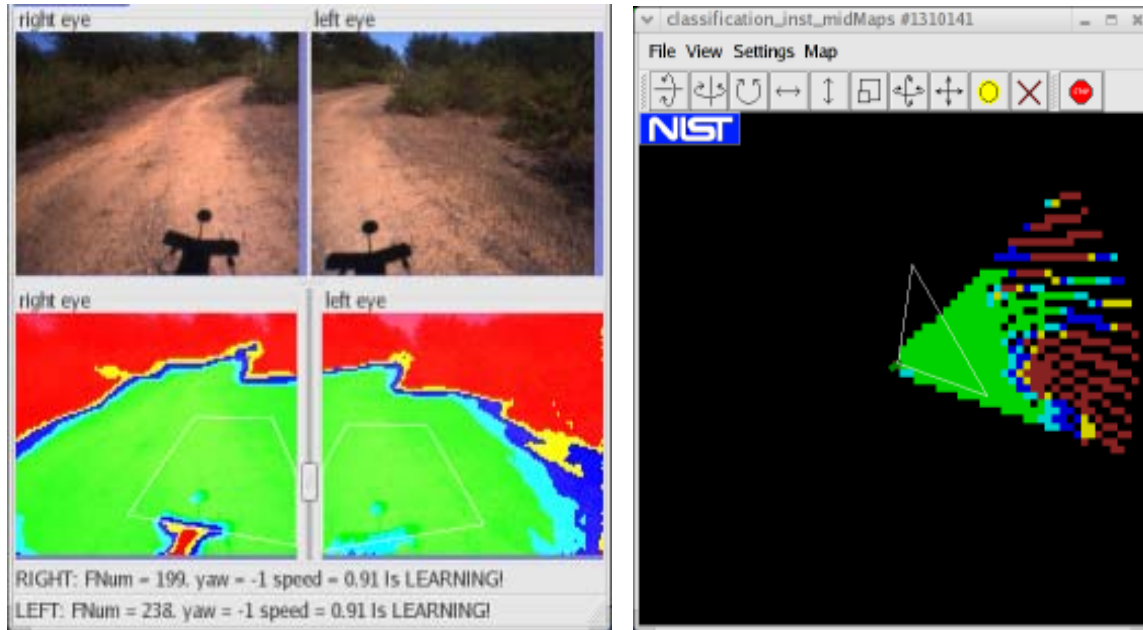


Figure 4. Left, top shows the original color images. Left, bottom shows the classification images based on the histogram color model (defines cost image). Right shows the projection of an image pixel based on ground plane assumption.

The construction of the background model initially and randomly samples the area above the horizon (image top). This can be problematic since the area above the horizon in the image might only be sky. However, once the algorithm is running, construction of the background model can be extended to the pixels beyond the horizon which the previous classification result believed to be background. Using the previous temporal fusion result, the algorithm randomly samples pixels in the current frame that the previous temporal fusion result identified as background. These random samples are used to construct the background color model.

In addition, color models are constructed from the original 512 x 384 image. Construction from the original image in this case means that more points are available to the color models, leading to more accurate models. Because the original image contains noise data, the color models will also inherently have some robustness against noise.

While the algorithm shows good results on single model ground regions when the ground is homogenous with respect to time and space, it has trouble with large amounts of non-homogeneity, e.g. driving on unshaded ground into completely shaded ground. Even if the algorithm had previously seen both ground conditions, the algorithm will have trouble correctly segmenting the “newly” encountered ground condition. This is due to the fact that the ground is modeled using a single color model while the ground clearly consists of multiple color distributions. Even though the single color model is capable of temporarily remembering multiple ground conditions through temporal fusion of histograms, this memory is very short. If the algorithm sees shaded ground for a long enough period of time, it forgets what unshaded ground looks like. As such, the single ground color model can usually remember only one color distribution at a time.

In order to remember multiple color distributions, we must use multiple ground color models. This extension leads to a drastic improvement in the algorithm's results by allowing the algorithm to remember a number of previous ground conditions and to apply these learned conditions to future images. This ability to learn leads to a powerful capacity to correctly segment in very difficult situations. As new data are processed, each color distribution model is updated with new histograms, changing with time to fit changing conditions. The new ground histograms to existing color models using a comparison function. If the difference exceeds a specific threshold and a maximum number of ground models has not yet been reached, then the algorithm enters a period known as the learning mode which will generate a new color model. Otherwise, the algorithm uses the histogram to update the closest ground color model.

The learning mode is a period of time in which the algorithm monitors new histograms in an attempt to pick out the histogram that is most different from existing ground models. This is done to avoid picking color models that contain significant amounts of overlap. In the learning mode, if a histogram is found to be more different than a previous histogram, the learning mode is extended. Eventually the learning mode will end, and the most different

histogram is used to create a new color model. Note that during the learning mode, none of the preexisting color models are updated. This, however, means that new histograms are unnecessarily being thrown away until the learning mode has ended. To deal with this, another ground model is included that essentially implements the single model case, taking a summation of a number of previous histograms. This color model helps to represent color distributions that would normally have color models, but do not while learning is on going.

#### A. Ground Probability Calculation

In this step, the algorithm goes through every pixel in an image and calculates a ground probability based upon its color. The end result is a probability map that represents the likelihood that an area is ground. Given a specific color, a specific ground color model and the background color model, ground probability is calculated as:

$$P_{ground} = \frac{N_{ground}}{N_{ground} + N_{background}}$$

where:  $N_{background}$  and  $N_{ground}$  are the number of hits in the corresponding histogram bin.

In the case of a single ground model, a single probability is calculated. With multiple ground models, multiple ground probabilities are calculated at each pixel. The largest ground probability is selected as the ground probability for that pixel. The justification is that a color may fit into one ground model better than all of the other models. For example, a dark pixel may fit into a model of shaded ground better than into a model of unshaded ground.

#### B. Temporal Fusion

This step takes ground probabilities across multiple frames and temporally fuses them together,  $P_t$ , to achieve a more stable result. The end result is a final probability map updated from frame to frame and expected to be more consistent than individual probability maps constructed from frame to frame.  $w$  is a weighting constant and  $w_{max}$  is the maximum number of images for temporal fusion.

The temporal fusion algorithm used in this project can be best described as a running average with a parameter to adjust for the influence of new data. The algorithm is presented here:

$$P_t = \frac{(w * P_{t-1}) + P}{(w + 1)}$$

$$w = w + 1 \text{ if } w < w_{max}$$

The ability to adjust the influence of new data is an important aspect of the temporal fusion algorithm. A number of situations exist where this capability can have an important role. For example, while the vehicle is turning, it is likely that the probability maps from frame to frame will have very little overlap. In such a case, it is likely helpful to set the influence of new data very high.

The final probability map is used to determine the traversability cost. The traversability cost is determined by the probability:

$$\text{Cost} = \text{probability of pixel being on the ground} * 250.$$

In order to reduce processing requirements, probabilities are calculated on a reduced version of the original image. The original image is resized down to 128 x 96 pixels through an averaging filter. This step has the additional benefit of noise reduction. Experimentation shows that this step does not significantly impact the final segmentation. A noteworthy aspect of this algorithm is that the color models were constructed from the original image for better accuracy, whereas probabilities are calculated on a reduced version of the image for greater speed. Each pixel with cost in the image will be placed in the world model maps using ground plane projection.

## IV. WORLD MODEL QUALITY: GROUND TRUTH COMPARED TO VEHICLE POSITIONING

#### A. Ground Truth Generation

In order to support evaluation of the quality of the world model produced by the vehicle, a series of measurements was performed to establish ground truth. The vehicle was tested on a course consisting of a number of obstacles, including several lines of fencing and obstacles constructed by placement of hay bales. The locations of these objects were precisely determined to provide ground truth. The ground truth determination was performed



using a two-part system available on the NIST campus. The two components are a differential GPS base station and a portable GPS with data logging capability.

A differential GPS base station has been installed in the tallest NIST building with antennas placed on the building's roof. The base station consists of the following components:

- A Novatel DL4 dual frequency, carrier phase GPS receiver\*
- A FreeWave 915 MHz spread spectrum data radio
- An uninterruptible power supply (UPS).

The base station antenna location was surveyed to millimeter level accuracy by collecting several hours of data, and post processing that data to obtain a relatively accurate position. The GPS unit uses this known location and its currently computed location to provide correction messages to other (rover) receivers in the area.

The base station GPS unit is configured to output differential corrections in CMR+ format. These are transmitted by the data radio at a 1 Hz rate, and available to rover receivers throughout the NIST site. Proper operation of the base station was confirmed by evaluating position reports of a rover GPS receiver located over National Geodetic Survey markers on the NIST campus.

A portable, rover component was used to collect data describing the obstacle locations (ground truth). A complete, portable position logging system consisting of another Novatel DL4 GPS receiver and antenna, a data radio receiver and antenna, and battery power supply has been configured in a small, portable enclosure for this purpose (see Figure 5).

This "ground truth box" receives the CMR+ messages from the base station GPS via the data radio link, and in favorable GPS conditions, operates in Real-time Kinematic (RTK) mode, providing position data with accuracy of a few centimeters. Data was collected by simply carrying the unit around the course obstacles, using care to position the antenna center above the obstacle edges. Data was collected at a rate of 1 Hz, which was suitable for walking around the obstacles. Higher data rates could be used. The data were inspected to verify that each position data record to be considered ground truth was logged while the unit was operating in RTK mode and that each such record indicated that the standard deviation of the northing and easting position were below 6 cm. This data quality information is provided by the GPS receiver for each point. Most of the records considered as ground truth displayed standard deviations of 1 cm to 2 cm. Position records not meeting these criteria were experienced on parts of the course near buildings and large trees (unfavorable GPS conditions due to loss of satellite line-of-sight and multipath), and were not considered to be valid ground truth points.



Figure 5. The ground truth data logging unit consists of a GPS receiver with memory card for logging data, a differential GPS data link radio, antennas, and battery power.

Data collection times were picked to avoid particularly poor GPS conditions through the use of an Applanix Mission Planning Tool. Figure 6 shows spikes in the Position Dilution of Precision (PDOP) values due to satellite constellation geometry at various times of the test day. These periods often correspond to times when the number of satellites visible from the NIST location is reduced, but can also occur when the satellites are not well spread out. These times were avoided for the ground truth data collection.

---

\* Commercial equipment and materials are identified in order to adequately specify certain procedures. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that these materials or equipment identified are necessarily the best available for the purpose.

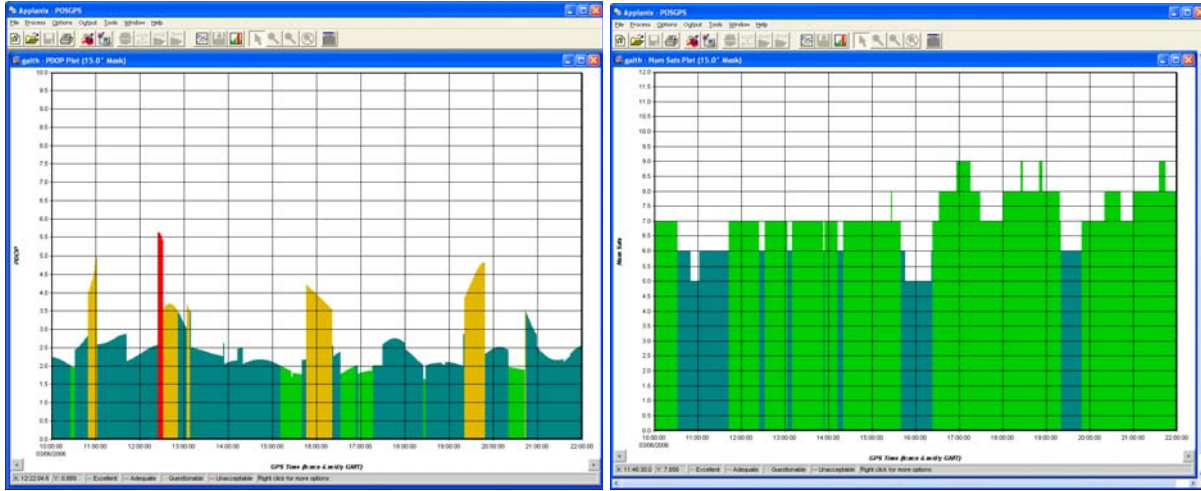


Figure 6. Left: Spikes in Position Dilution of Precision (PDOP) for the day of the test. Right: The number of satellites visible from the NIST location. The horizontal axis is time-of-day. Ground truth data collection avoided times of high PDOP and low numbers of satellites.

### B. Performance Analysis

To determine the quality of the collected world model data, a map was generated from the world model data as shown in Figure 7 (left). Obstacle costs shown in the map vary from 0 (lowest cost - blue) through red (highest cost - 256) as shown in the scale next to the map. The mapping algorithm calculates where each obstacle data point is relative to the vehicle position using onboard GPS and inertial data. The map data was then compared to collected ground truth data. Figure 7 (right) shows ground truth data overlaid onto the obstacle map shown Figure 7 (left). To test how well the obstacle map compares to ground truth, an average error between map and ground truth closest points and the corresponding standard deviation were determined. The equations to compute these errors are as below:

For each ground truth point ( $n, e$ ), find the closest obstacle point ( $on, oe$ ) in the map and compute the distance between these points according to:

$$d = \sqrt{((n - on)(n - on) + (e - oe)(e - oe))}$$

Then compute Average Error = sum of distances ( $d$ ) / (number of points and standard deviation of the error) = sqrt (variance) where variance = (sum of the difference between the distance and average error) / number of points.

Table 1 shows the average distance and standard deviation for each major obstacle including: large fence, small fence and hay bales. The results show that the vehicle places obstacles within the map relative to the vehicle that can include a very large average error in distance relative to the actual location. This error has little effect on the vehicle avoiding an obstacle that is seen for the first time as in Figure 3 (left) or one that remains within the local map. However, once the vehicle drives beyond the local map that includes an obstacle, the next time the vehicle needs to plan around that same obstacle, its position uncertainty can cause the path to be planned wrong. It could plan a path that is a large distance from the actual obstacle, even plan a path to pass it on the opposite side, or many other possible path planning scenarios.



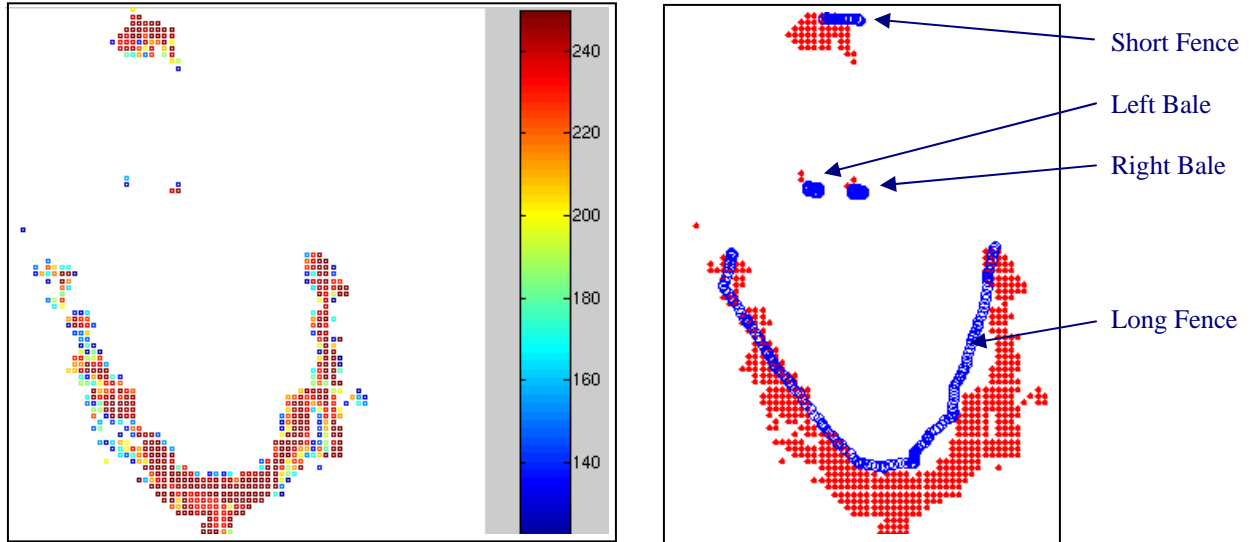


Figure 7. Left: Map generated from world model obstacle data. Obstacle costs = 0 (lowest cost - blue) through red (highest cost - 256). Right: Map of obstacle data (red) above a cost of 160 compared to ground truth (blue).

	<i>Long Fence</i>	<i>Left Bale</i>	<i>Right Bale</i>	<i>Short Fence</i>
<i>Average Error (m)</i>	0.91	0.83	1.52	0.48
<i>Standard Deviation (m)</i>	0.71	0.28	0.37	0.30

Table 1. Average distance error and standard deviation of each obstacle compared to ground truth.

A few possible causes to this obstacle location error are a combination of:

- the vehicle’s combined GPS and inertial information is obviously not as accurate as the ground truth GPS system,
- the data collected from the stereo cameras is not accurate, adding further error to the obstacle map, and
- the algorithm that places the obstacles in the map is not accurate.

The first point cannot be corrected without replacing the onboard positioning sensors with more accurate GPS equipment. The second point can perhaps be improved slightly with better stereo camera algorithms and/or calibration. The third point may also be a calibration issue that consistently places obstacles in an offset location from actual. Ideally, better sensors for obstacle detection and vehicle positioning are the best remedy.

## V. CONCLUSIONS AND FUTURE WORK

This paper has described the implementation of a 4D/RCS world modeling system as applied to the LAGR learning problem. The world model was designed to act as a bridge between multiple sensory inputs and a behavior generation (path planning) subsystem for the LAGR off-road, autonomous driving within 100 m. Higher 4D/RCS levels can accommodate longer distances. The world model maps have been described, as well as the functions used to maintain the model. The examples include integrating and fusing sensory data from multiple sources into the world model map. The representation is currently being used as part of the LAGR Project test runs at DARPA-selected test sites.

Due to time and funding limitations, parts of the 4DRCS world model [8] have not yet been implemented. For example, the confidence-based mapping may not be adequate for the task of modeling moving objects (cars, targets, etc.). Additional research is also needed to broaden the system's terrain and object classification capabilities. The number of terrain and object classes currently used is small; the ability to recognize and label bodies of water, rocky roads, buildings, fences, etc. would enhance the vehicle's autonomous driving performance.

World model data was compared to ground truth data in the form of obstacle map comparison. The results showed that obstacles placed in the cost map have relatively large errors from their actual locations. Errors can be caused by a combination of low quality onboard GPS and inertial systems, stereo cameras and perhaps their distance

to objects algorithms and calibration, and/or by offset obstacle placement in the map. The last two points could perhaps be improved by their calibration.

Future efforts toward improved performance should therefore include: calibration of obstacles placed in the map, evaluating obstacle offset locations when placing them in the map, and comparison of stereo cameras versus other non-contact sensors, such as LADAR (LAsER Detection And Ranging) sensors. Also, towards indoor applications, effort should be placed on measuring the performance of an intelligent vehicle mapping a facility, such as for manufacturing, requiring even greater accuracy of obstacle placed in cost maps than for outdoor environments. Intelligent vehicles that navigate through flexible and unstructured indoor facilities with close tolerance machine access, maneuverability around people and expensive equipment, etc. must include real time map updates that allow appropriate vehicle behavior for safe, yet effective vehicle control.

#### REFERENCES

- [1] Albus, J.S., Huang, H.M., Messina, E., Murphy, K.N., Juberts, M., Lacaze, A., Balakirsky, S.B., Shneier, M.O., Hong, T.H., Scott, H.A., Proctor, F.M., Shackelford, W., Michaloski, J.L., Wavering, A.J., Kramer, Tom , Dagalakis, N.G., Rippey, W.G., Stouffer, K.A., *4D/RCS Version 2.0: A Reference Model Architecture for Unmanned Vehicle Systems*, NISTIR, 2002.
- [2] Albus, J.S., Balakirsky, S.B., Messina, E., Architecting A Simulation and Development Environment for Multi-Robot Teams, Proceedings of the International Workshop on Multi Robot Systems, Washington, DC, March 18 – 20, 2002.
- [3] Balakirsky, S.B., Chang, T., Hong, T.H., Messina, E., Shneier, M.O., A Hierarchical World Model for an Autonomous Scout Vehicle, Proceedings of the SPIE 16th Annual International Symposium on Aerospace/Defense Sensing, Simulation, and Controls, Orlando, FL, April 1-5, 2002.
- [4] Albus, J.S., Juberts, M., Szabo, S., RCS: A Reference Model Architecture for Intelligent Vehicle and Highway Systems, Proceedings of the 25th Silver Jubilee International Symposium on Automotive Technology and Automation, Florence, Italy, June 1-5, 2000.
- [5] Bostelman, R.V., Jacoff, A., Dagalakis, N.G., Albus, J.S., RCS-Based RoboCrane Integration, Proceedings of the International Conference on Intelligent Systems: A Semiotic Perspective, Gaithersburg, MD, October 20-23, 1996.
- [6] Shackelford, W., Stouffer, K.A., Implementation of VRML/Java Web-based Animation and Communications for the Next Generation Inspection System (NGIS) Real-time Controller, Proceedings of the ASME International 20th Computers and Information in Engineering (CIE) Conference, Baltimore, MD, September 10 – 13, 2000.
- [7] Michaloski, J.L., Warsaw, B.A., Robot Control System Based on Forth, Robotics Engineering, Vol. 8, No. 5, pgs 22-26, May, 1986.
- [8] Albus, J., Bostelman, R., Hong, T., Chang, T., Shneier, M. Shackelford, W., The LAGR Project - Integrating learning into the 4D/RCS Control Hierarchy, Proceedings of the International Conference on Informatics in Control, Automation and Robotics (submitted), Setubal, Portugal, August 1-5, 2006.
- [9] Oskard, D., Hong, T., Shaffer, C., Real-time Algorithms and Data Structures for Underwater Mapping, National Institute of Standards and Technology, 1990.
- [10] Shoemaker, C. M., Bornstein, J. A., The Demo3 UGV Program: A Testbed for Autonomous Navigation Research, Proceedings of the IEEE International Symposium on Intelligent Control, 1998.
- [11] Herbert, M., Thorpe, C., Stentz, A., Intelligent Unmanned Ground Vehicles, Autonomous Navigation Research at Carnegie Mellon, Kluwer Academic Researchers, 1997.
- [12] Tan, C., Hong, T., Shneier, M., and Chang, T., Color Model-Based Real-Time Learning for Road Following, NISTIR (submitted).