Autonomous Robot Navigation:
Using Multiple Semi-supervised Models for Obstacle Detection

Adam Bates
University of Colorado at Boulder

Abstract: *This paper proposes a novel approach to efficiently creating multiple semi-supervised models of obstacles for life time learning applied to autonomous robot navigation. While previous techniques, which used predefined models of obstacles and terrain, have had success in constrained environments, this paper provides a framework to improve upon this naive technique in obstacle-rich environments.*

Introduction:

With recent success coming from both planetary exploration rovers and ground robots, interest has increased in autonomous navigation. Autonomous ground robot navigation presents a number of difficult problems, which are unsolved as of this time. The largest and most difficult problem concerns how to create a model representing the area in-front of the robot. Currently this problem is unsolved except for very constrained environments. As more successful autonomous navigation systems are developed, more insight is gained into how to incorporate higher levels of knowledge.

In the past, these autonomous navigation systems were designed to compensate for large communication delays. When a rover is sent to another planet there is a large amount of lag rendering remote control of the rover imposable. This lead to the autonomous navigation systems placed in Sojourner and the MER rovers, Spirit and Opportunity. These programs added momentum to exploration into other areas of autonomous navigation systems. The DARPA Grand Challenge is one such program. The goal of the program is to create a sensor rich robot with a robust, autonomous navigation system. Meant to navigate a robotic ground vehicle through the desert, the DARPA Grand Challenge focus on navigation of relatively obstacle free terrain. This program differs from the rover missions in that, there is no need to compensate for communication lag, but instead it is meant to remove the need for human supervision in potentially lethal situations.

A second DARPA sponsored project for autonomous navigation is LAGR, standing for Learning Applied to Ground Robots. LAGR focuses on creating autonomous ground robot navigation systems in obstacle-rich environments. Each team in the LAGR program is provided with a pre-built robot with fixed sensors providing a standard platform from which to create a robust navigation system. This paper will focus on the LAGR robot as the platform for a new technique for autonomous ground robot navigation and modeling in an obstacle-rich environment, using semi-supervised machine learning.

The autonomous navigation system described in this paper is targeted toward obstacle-rich environments. The system uses a robust stereo vision subsystem to locate and model obstacles as the robot explores. The stereo vision is only usable within 3 meters of the robot, thus a secondary system must be used to locate obstacles in the far-view. This paper focuses on locating obstacles in the far-view, beyond stereo vision range, and how stereo vision can be used to locate obstacles which are modeled by a semi-

supervised machine learning algorithm.

The first part of this paper will consist of a brief overview of current techniques for autonomous ground robot navigation, and why they have been successful or have failed. Next a description of the technique and the algorithms used. The final sections will consist of an in-depth discussion of experiments conducted using this technique, and an overview of potential problems and future work pertaining to this technique.

Past Work:

The MARS rovers are examples of autonomous ground robot navigation in a constrained environment. For relatively long amounts of time these robots navigate the surface of Mars, avoiding obstacles and reaching predefined goal locations. Unlike Earth the atmosphere on Mars does not have a large effect on the color and brightness of the light coming from the Sun. Because of this, sensors and algorithms can be calibrated to take very small changes into account. This allows the rovers to continue functioning properly without the need to continuously update the navigation system. Earth's continuously shifting atmosphere, however, has a larger effect on lighting conditions, thus making calibration extremely difficult. The surface of Mars also lacks the large number of different obstacles that exist here on Earth. Only having to avoid a small number of obstacles, such as stones and larger rocks formations, the problem of modeling the robots environment is greatly reduced[1]. Because of the relatively obstacle-free environment and low number of atmospheric changes, the Mars Rovers can successful navigate the surface of Mars.

The DARPA Grand Challenge provided momentum to create navigation systems that could successfully navigate through a relatively obstacle-free environment. During the 2004 DARPA Grand Challenge every robot failed. This was due to the inability for the navigation systems to react and respond to the changing conditions encountered when navigating through the desert. Examples of these changing conditions are shifting lighting conditions, the movements of the robotic ground vehicle and the effect on the sensors. During the year preceding the 2005 DARPA Grand Challenge, the teams turned to techniques that were able to compensate for these changing conditions. Both machine learning and human-skill transfer were used to train the navigation system and subsystems[2]. With this improved flexibility six robots were able to complete the course 100% autonomously.

Navigation Algorithm:

This section describes the new technique for autonomous ground robot navigation. First, the robot used to conduct test and gather data will be described, then an overview of the subsystems will follow. Next, an in-depth discussion of the Eye subsystem will be presented. Finally the semi-supervised machine learning algorithm will be discussed and explained.

The LAGR robot is the perfect platform for conducting autonomous ground robot navigation research. The bulk of perception sensors on the LAGR robot are two pairs of Point Grey Research Bumblebee stereo cameras. These are used in the navigation system to construct a representation of the ground plane in-front of the robot, and to search for obstacles in the far range of the robot using 2D vision techniques. Figure 1 shows a front view of the LAGR robot. The sensor mast holds the two stereo camera pairs and a GPS receiver.

Figure 1. LAGR robot.

The LAGR also has three main position sensors. The first is a global positioning unit. This allows the robot to locate itself with in 30 meters from its true location on the Earth. An accelerometer and optometry make up the local position sensors. These sensors allow the robot to know how far it has moved on a fine scale. For example, the robot rotated 90%, or moved forward 20 feet. These systems are used to create the autonomous ground robot navigation system.

Three main subsystems make up LAGR's autonomous navigation system. There are two stereo vision subsystems, one for each side of the robot. These stereo eyes allow the robot to construct a 3D representation of the terrain in-front of the robot. Each stereo eye also provides images from each camera. These images supply input for the semi-supervised machine learning models. Each eye subsystem transmits knowledge about the area in-front of the robot to a planner subsystem. The planner subsystems constructs a 2D cost map using the information from the eye systems. This cost map is then used to construct a path to the goal location. Once a path is constructed, the planner sends a direction and velocity to the controller subsystem. The controller subsystem then translates the direction and velocity into servo signals that are sent to the wheel servos.

Eye subsystem:

The Eye subsystem consists of two parts. A 3D stereo vision system and a semi-supervised machine learning system. These two systems work in conjunction to find obstacles. The 3D stereo vision system is capable of navigating a variety of terrain. When only using stereo vision the robot is near-sighted, tends to get stuck in cul-de-sacs and takes long and complex paths to the goal. By attaching the semi-supervised machine learning algorithm on top of the stereo vision system the robots vision can be greatly extend.

The 3D stereo vision system uses a pair of cameras to create a disparity image that describes the difference between the left and right images. This disparity image can them be used to infer height from the ground plane. By calculating the disparity image that would be created by a flat ground plane and comparing it to the true disparity. Taking the absolute value of this difference in disparities, it is possible to determine the ground plane. A binary image showing ground plane and obstacles is found by thresholding the absolute values of the difference in disparities. In this paper the threshold value of 1.00 was found to work well. This binary image shows the XY location of objects that are taller than the robots wheelbase in a rectified image.

Figure 2.
RGB rectified image from the left eye camera.

Figure 3.
Disparity image created using the difference between the right and left images from the stereo camera.

Figure 4.
Binary image of the ground plane. White defines areas that are to high for the root to drive over.

After the binary ground plane image is constructed it is cut up into windows. Each window is then flooded with the dominating color, either black or white.

Figure 5.
Binary ground plane image cut into windows.

At this point the windows in the bottom half of the binary image are used to create color models of obstacles. The bottom half of the image is about 3 meters from the front of the robot, thus it does not produce robust ground plane readings. 12.5% of the right edge of the left eye must be removed due to the areas in the two images that don't have corresponding pixels. This is true of the right eye also, but the left edge must be removed. This produces a matrix of labeled windows that can them be used for semi-supervised learning.

Each window that is classified as an obstacle is used to create a semi-supervised machine learning model. This model is then projected into the distance and tested against the labeling produced by stereo vision. If the model contradicts with the stereo vision ground plane (i.e. classifies ground plane as obstacles) then it is thrown out. The models that don't contradict with the stereo labeling are saved. After all the new obstacles have been classified the saved models are projected into the far-view producing a matrix showing which windows contain obstacles. The resulting matrix can them be sent to the planner system.

Semi-supervised Learning

Due to the limitations of stereo vision a method for classifying obstacles in the far-view is needed. A semi-supervised learning technique was chosen to meet this need. Semi-supervised learning has the benefit of being able to spread labels to unlabeled points by using structures in the existing data. This allows the semi-supervised learning algorithm to use the stereo labeled windows to locate obstacles in the far-view. Semi-supervised learning also has the advantage that it enables a model to be created with examples of only one class, in this case only obstacles are

modeled. If a supervised learning algorithm was used both obstacles and ground plane would have to be present to create a model. This means that the model of an obstacle is not only tied to the particular obstacle, but also to the ground plane that the robot currently occupies. If the ground plane has a particular feature, such as being a red path, and the obstacle is a hay bale, then when the robot moves off the red path onto short grass the model of the hay bale will cause the ground path to classify as an obstacle.

The problem of obstacle avoidance can be approached in two ways. Classify obstacles, which this system does, or classify ground plane. The reason this for classifying obstacles and not ground plane stems from the need to keep a relatively small number of models. The number of models needed to fully classify obstacles is relatively low, around twenty models for a hay bale. Over time false positive rate from the ground plane models increases rapidly. This does happen with the obstacle models but does not result in the robot avoiding clear patches of ground.

The particular semi-supervised learning algorithm used in this paper uses a simple k-nearest neighbors to classify windows. The window based approach enables a distance to be defined which can then be used to find other windows that are similar. Thus, the algorithm only needs a single labeled window to infer the labels of the rest of the windows in the image. The distance used by the algorithm is defined by the mean distance between every other pixel in the window (either Euclidean squared distance or the mahalanobis distance). A thresholding parameter defined by the max distance between the mean and all the pixels in the window allows the scaling of the output during classification. During classification the distance between the mean values of the model and the pixels in the windows divided by the threshold parameter, are used to produce a class label. This produces a real value between one and zero. The values from all the models are then combined using a max function. This yields a matrix that has small values in windows that are very similar to obstacles.

Experimental Results and Future Work

This section discusses the experiments conducted using this technique and the results from those experiments. Four main experiments were conducted. The first involved adjusting the size of the windows the images are cut into. Next, an investigation to find the optimal number of training examples, when finding the distance parameter for the window was conducted. Thirdly, a comparison of Euclidean and Mahalanobis distance measures was completed. Finally, two processes for selecting models to label the image are tested.

In the following experiments all images have the same width and height, 512 x 384 pixels. Normalized RGB was used for the rectified image. The disparity image has values ranging from 1 to 256. In the binary ground plane image, white defines an obstacle and black defines ground plane. The images are all from the left eye system.

Window Size

Three window sizes where tested; 8 x 8, 16 x 16, and 32 x 32 pixels. The reason for these windows sizes were chosen was due to the ability to optimize window segmentation algorithms. These window sizes also decrease the amount of cache misses when touching a window multiple times. For this experiment 2 examples and mahalanobis distance were used.

Window Size	***Number of Models***
8x8 pixels	131
16x16 pixels	29
32x32 pixels	7

Table 1.
As the window size increases the number of models it take to describe that obstacles deceases.

Figure 6.
Labeled image using 131 models and 8x8 pixel windows

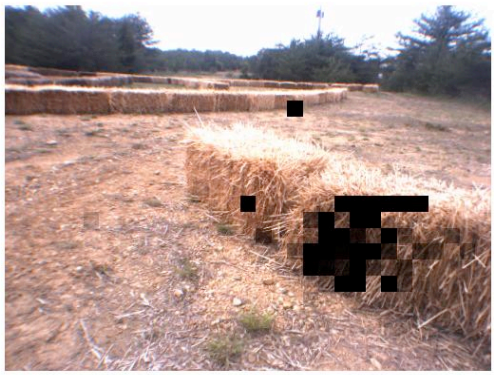

Figure 7.
Labeled image using 29 models and 16x16 pixel windows.

Figure 8.
Labeled image using 7 models and 32x32 pixel windows.

It was found that the models built with 16x16 pixel windows produced more robust models than 8x8 or 32x32 pixel windows. The models created by 8x8 pixel windows resulted in classification of ground plane as obstacles in later frames. 32x32 pixel windows where unable to capture obstacles in the far-view because of there size.

Number of Examples

The second experiment consisted of tests with varying numbers of examples to produce a model. Varying the number of examples moves the window used to select the training example to surrounding areas close to the original window. This was found to create lager numbers of models that were highly specialized to the window they were created in. When the number of examples was larger than ten, the models overlapped the ground plane and were discarded, thus the number of models decreased.

# of examples	***# of models***
2	29
3	65
5	60
10	25
15	21

Table 2.

With the number of examples set to 3 the hay bale in the image is almost entirely labeled correctly. The models robustly classified hay bales from frame to frame without misclassification of ground plane, as *Figure 9* shows.

Figure 9.
Labeling of the image after 10 frames using 3 examples for training.

Figure 11.
10 frames from the creation of the initial models. Notice the labeled hay bales in the far-view.

Distance Measure

The last experiment compared the different distance measures. The mahalanobis distance measure using 16x16 pixel windows and 3 examples for training was found to produce the most robust models. By repeating the above experiments it was found that the euclidean optimal learning parameters were 16x16 pixel windows and 5 examples. The labels that were produced by the euclidean distance measure are not as proficient at classifying obstacles close to the robot, but in the far-view perform at a higher level. This can be seen in *Figure 10* and *Figure 11*.

Figure 10.
Labeled windows after model creation.

Model Selection

When creating multiple models the process used to choose which models to keep for label propagation becomes difficult. One could choose to remove any model that contradicts with the pre-labeled windows. This was done in the previous experiments. This type of selection process presents problems when areas without obstacles are encountered. The reason being that most of the models build will tend to disagree with one or more frame from the sequence. This in turn removes that model and also removes the ability to label obstacles in the future. Because of this problem an experiment was connected which compares a sequence of labeled frame with model removal enabled and with model removal disabled.

During the comparison of the two methods two properties were found. First, when models are removed after conditions the ability to label obstacles in the far-view is greatly reduced, see *Figure 12*. Secondly, when areas are miss labeled by stereo the incorrect models are then used to miss-label windows in future frame, see *Figure 13*.

Figure 12.
Left shows labeled image using model removal. Right shows labeled image without model removal. Notice with label removal all models of hay bales have been removed over time.

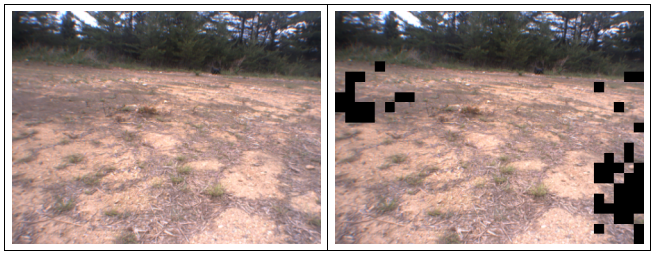

Figure 13.
Left shows labeled image using model removal. Right shows labeled image without model removal. Notice the number of misclassifications when not using model removal.

Future Work

Currently this technique is only laying the foundation for a life long learning process for autonomous navigation systems. The ability to build and save models of obstacles allows the autonomous navigation system to navigate more quickly, and also allows the system to robustly navigate new and obstacle-rich environments. There are four areas that need further work before this system can be reliable enough to use on real ground robots. First a better method for selecting training examples must be developed. As well further exploration into using a different semi-supervised learning algorithm is needed. Third, a timing method for model creation needs to be devised. Lastly, the current model system uses a max function to combine labels, but there might be a better method for this.

Currently models are built from all the windows that are classified as obstacles by stereo vision. This produces a large number of models that overlap. A system for trimming these overlapping models would improve performance when labeling obstacles in the far-view. The current semi-supervised learning algorithm could be modified to use a number of windows to create a single model. This would allow for a model to capture a larger amount of windows and reduce the overlap from other models. Another option is to combine overlapping models. Since the model consists of a distance and a threshold it should be conceivable to combine models easily.

Currently only one semi-supervised learning algorithm was used to label windows. Other semi-supervised algorithms should be used to check whether a better method of labeling is possible. The current algorithm is slow because of the need to calculate the distances between pixels. A random walk may provide a faster way to produce a suitable distance. The method of label propagation could be used as described in [3]. Both methods would allow the creation of one model pre-obstacle and not pre-window. This would allow for even fewer models to be created and maintained, thus increasing the speed of the system.

A timing method or metric must be defined to reduce the number of models created every frame. The most time consuming part of this technique is creating the different models. Even after an obstacle has been modeled the next few frames produce slightly different views of the same obstacle. When an obstacle is lightly classified, the whole obstacle is not classified due to misclassification of ground plane; multiple models could be made that produce the same light classification. If the system were allowed to create models every ten frame this effect would be reduced. Secondly an obstacle that is lightly classified still can be sent to the cost map and navigated around. So a method for telling if there is a new obstacle is present would improve performance, and reduce the number of models

needed.

Since the semi-supervised learning algorithm outputs distances, a max function is used to combine multiple models. A better procedure for combining models should be possible. A voting system could be implemented to allow the models to vote which windows should be labeled as obstacles. This would make models noisier, but reduce miss-classification of the ground plane.

Works Cited

[1] J. N. Maki and Steven W. Squyres and Mark A. Schwochert and Todd Litwin and Reg G. Willson and Mark W. Maimone and Eric T. Baumgartner and Anne Collinot and T. S. Elliot and E. C. Hagerott and L. M. Scherr and Robert G. Deen and D. Scott Alexander and J. J. Lorre. *Mars Exploration Rover Engineering Cameras* , 2003.

[2] Bob Davies and Rainer Lienhart. *Using CART to Segment Road Images,* 2005.

[3] X. Zhu and Z. Ghahramani. *Learning from labeled and unlabeled data with label propagation.* Carnegie Mellon University, 2002.