Robotics: Science and Systems 2006
Philadelphia, PA, USA, August 16-19, 2006

Enhancing Supervised Terrain Classification with Predictive Unsupervised Learning

Michael Happold, Mark Ollis, and Nik Johnson
Applied Perception, Inc.
Pittsburgh, PA 16066
Email: happ@appliedperception.com

***Abstract*— This paper describes a method for classifying the traversability of terrain by combining unsupervised learning of color models that predict scene geometry with supervised learning of the relationship between geometric features and traversability. A neural network is trained offline on hand-labeled geometric features computed from stereo data. An online process learns the association between color and geometry, enabling the robot to assess the traversability of regions for which there is little range information by estimating the geometry from the color of the scene and passing this to the neural network. This online process is continuous and extremely rapid, which allows for quick adaptations to different lighting conditions and terrain changes. The sensitivity of the traversability judgment is further adjusted online by feedback from the robot's bumper. Terrain assessments from the color classifier are merged with pure geometric classifications in an occupancy grid by computing the intersection of the ray associated with a pixel with a ground plane computed from the stereo range data. We present results from DARPA-conducted tests that demonstrate its effectiveness in a variety of outdoor environments.**

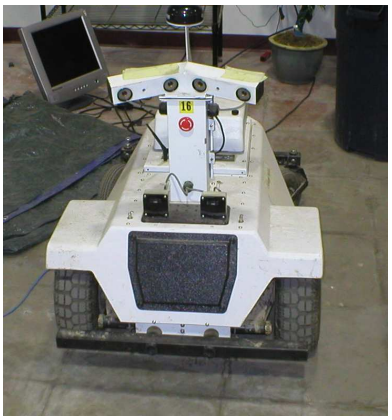

Fig. 1. The LAGR Test Platform

I. Introduction

Autonomous navigation in outdoor environments requires accurate and dense sensor data. In unstructured, natural settings, where lighting conditions cannot be controlled and the scene geometry is highly complex, stereovision often provides neither. Given a noisy and sporadically sparse description of the world, it is extremely difficult to formulate rules for assessing the traversability of terrain that are robust to both stereo artifacts and changes in environment type. The more varied the terrain a robot must traverse, the more information that is needed to distinguish terrain types. The higher the dimensionality of the features used to distinguish terrain, the harder it is to discern rules that broadly apply. Nonlinear relationships crop up, and the human expert is overwhelmed.

Despite some of the successes of the rule-based approach [1],[2], its limitations are substantial. Machine learning techniques have shown the promise of being able to replace the human expert's struggle to find hand-coded rules that generalize well with automated methods for discovering how best to carve up the feature space into broadly applicable classes [3],[4],[5]. Supervised learning, though still relying on a human for labeling, is useful for classifying geometric features that possess a reasonable level of invariance across environments. The relationship between color and traversability, however, can alter rapidly and dramatically due to changes in lighting or terrain. Such substantial changes in the mapping between features and classes often render supervised learning too brittle. This lack of robustness can be addressed in part by computing invariances in the color data prior to training or training a set of classifiers that span the known space of lighting and terrain conditions. Achieving the former is an open problem in computer vision, while the latter is extremely burdensome and still requires a judgment by the robot as to which conditions obtain, and, hence, which classifier to use. The alternative is to make use of online learning to adapt to the fluctuations. Color information can augment geometric assessments of terrain in a variety of ways. In [6], it is used to filter out vegetation that would be considered an obstacle from a purely geometric standpoint. It can also be used to provide an independent classification of traversability, which is then combined with the geometric classification. For example, [7] computes separate geometric and color-based classifications of terrain as road or non-road, using the color-based classification to increase the confidence of the geometric classification when they agree. A more monolithic approach can be found in [8], where neural networks are trained on color and geometry combined in a single feature vector. The color constancy problem is tackled by training a set of classifiers on different color models of the terrain.

Greater attention has recently been devoted to adaptive methods for determining scene geometry from color information. Much of this work can be found in the area of road-following, where online and self-supervised techniques are

gaining prominence. For example, rather than attempting to train a system in a supervised fashion to handle all road types, [9] made use of the assumption that the vehicle starts on a road and can therefore use the current appearance of the road to detect future roads. A reverse optical flow technique is used to trace back the current road appearance to how it appeared in previous image frames in order to extract road templates at various distances. The templates can be then matched with distant possible road regions in the imagery.

The system described in [9] attempts to answer the question: How did something judged traversable nearby look when it was far away? It then makes use of the assumption that other things that look similar far away will be traversable nearby. Our approach is instead to judge the traversability of proximal terrain on the basis of its geometry, learn the relationship between proximal color and geometry, and make use of the assumption that if proximal colors map to a particular geometry, then distal colors will as well. This enables our system to fill in proximal regions with terrain assessments using the geometry predictions of the color models when there is little actual geometric information available, as well as to extend the range of our perception beyond that of stereo. This is extremely beneficial in complex, unstructured environments where the stereo range data may contain large gaps, particularly when the robot is very close to an obstacle. The online learning of the color/geometry relationship is rapid and ongoing, enabling the system to adapt to lighting and terrain type changes without explicitly addressing the color constancy problem. Seeding the color learning system is unnecessary, and adaptation is so rapid that a couple of seconds wait after starting the robot is sufficient to develop a useful characterization of the surroundings.

In the following three sections, we will provide a more detailed explanation of our approach, examine results from both DARPA-conducted tests and our own experiments, and discuss possible future improvements to this work.

II. Approach

There are three major components of the terrain assessment system: a supervised classifier that exclusively uses geometry information, the unsupervised color learning module, and the terrain assessment merging system that produces cost grids for a planner. The supervised geometry classifier is trained on a set of 8-dimensional vectors which characterize the geometry in cells of a voxelized world filled with stereo range data. The geometry classifier outputs its classification of the input into one of four classes, which is then transformed into a terrain cost. While the trained classifier is running on the robot, statistics describing the relationship between the colors of the points in the cell and the geometry vector for that cell are accumulated. Once a sufficient amount of data has been gathered, a cost for each pixel in the rectified reference stereo image is determined using the geometry predicted by its color. The position for a pixel's cost is computed by intersecting the pixel's associated ray with the ground plane computed from the stereo range data. This in itself would result in large, erroneous cost regions in an occupancy grid caused by the projection of vertical obstacles onto the ground plane. Therefore, they are trimmed as described in section B before being merged. Finally, pure geometric costs are merged into an occupancy grid using a confidence metric described in section C, and this is in turn merged with an occupancy grid filled with color-predicted costs to enhance detection of obstacles and extend the range of perception beyond viable stereo.

A. Supervised Learning of Terrain Traversability

Supervised learning techniques have proven their value in relieving the human expert of the task of deducing complex rules for terrain classification while still incorporating the expert's knowledge [10]. For this reason, we employ a neural network to learn the relationship between geometry and traversability. The human expert is asked only to judge the traversability of stereo range data, something that humans are quite good at. The labeling process consists of stepping through logs collected from the two stereo pairs on the LAGR robot [11]. The color range data is computed and placed in a modified occupancy grid [12] consisting of 20cm x 20cm cells with infinite height. The human expert is asked to assign one of four cost classes (low, intermediate, high, or lethal) to the data in each selected cell according to how difficult it would be for the robot to traverse that cell. Lethal cells are those that the robot cannot traverse.

Once the labeling process is complete, eight geometric features are computed for each labeled cell. Because some of these features rely on the identification of a ground plane, we first compute a robust plane fit using RANSAC to those points that fall between two planes emanating from the vehicle's control point (center of the front wheels on the ground) at 15 degree angles above and below a flat ground plane. Also, for each 3D point from a stereo pair, a ray trace through the grid is computed, placing the point and the ray pass-throughs in the appropriate cells. The eight geometric features that are then computed within each of the cells are as follows:

1) $Height\ Variation$: The difference between the minimum and maximum heights of points in a cell.
2) $Terrain\ Slope$: The slope of a RANSAC plane fit to the points in the cell relative to a flat world.
3) $Point\ Count\ (PC)$: The raw number of points within a cell adjusted to compensate for the increased lateral spread of the points at greater ranges.
4) $Point\ Count\ Above\ the\ Ground\ Plane\ (PCGP)$: The number of points that fall 10 cm above the computed ground plane, again adjusted for range.
5) $Density$: The ratio of points in the cell to the sum of points and pass-throughs for that cell.
6) $Mean\ Vertical\ Distance\ (MeanZ)$: The mean vertical distance of the points 10 cm or more above the ground plane to the ground plane.
7) $Standard\ Deviation\ of\ Vertical\ Distance\ (StdZ)$: The standard deviation of the vertical distances of the points 10 cm or more above the ground plane to the ground plane.

8) *Percent of Points above Ground Plane* (NGP): The percentage of points 10 cm above the ground plane out of the total number of points within a cell.

The ray-trace is undertaken for each 3D point within 10m of the robot's control point (CP), although the grid extends only 5.5m from the CP. This ensures that the densities at the limits of our usable range are not automatically 1.0, but rather more truly reflect visual penetrability.

Having accumulated a set of 4000 labeled cells, we train a multi-layer perceptron (MLP) with one hidden layer to output the correct cost class given the input 8-dimensional geometric feature vector. The MLP is trained by means of the Conjugate Gradient algorithm using the NETLAB package from Sussex University. The inputs are adjusted to be zero mean with unit variance. The hyperbolic tangent activation function is used for the hidden layer, while the output layer uses a logistic function. The cost of a cell, ranging from 25-255, is computed as the scaled sum of the activations of the output neurons, divided by the sum of the activations. The scale factors for the neurons are 25, 100, 150, and 255, corresponding to low, intermediate, high, and lethal class neurons respectively.

Once trained, the MLP is incorporated into a log playback tool and used to augment the original training set. The human expert steps through logs, adding instances where the MLP misclassified cells. Of particular interest are those cells where mismatches in stereo happen to fall. These false matches occur often when there are regular geometric structures in the background, such as buildings or lines of trees. They generally produce small clusters of range data above the ground plane, confounding some geometric features such as height variation. The goal is to teach the MLP to assign a cost class more indicative of the rest of the geometry of the cell, rather than be thrown off by the false matches.

Because we interface with the Carnegie Mellon-supplied path planner, which makes an absolute distinction between lethal and non-lethal obstacles, we introduce a lethality threshold for our MLP-produced costs. This is an arbitrary figure at first, set by observing the values produced in logs and the behavior of the robot on test runs. However, given that the robot is equipped with a bumper sensor, we can adjust the threshold in light of this feedback. Whenever the bumper is triggered, the system looks at the set of cells that could have plausibly been responsible for the event given the robot's position when it happened, singling out the most expensive cell as determined by the MLP as the culprit. The threshold is adjusted downward to ensure that a similarly expensive cell will be considered lethal in the future. Because the blame-assignment process is not perfect, we set a lower limit on this threshold to constrain this parameter to a plausible range.

B. Predicting Geometry from Color

Given a grid filled with geometric feature vectors and a set of colored 3D points that give rise to those feature measurements, we can set about creating a predictive model that maps color inputs to geometric features. It is common of late to choose parametric or semi-parametric models of color for scene interpretation. For example, [13] uses a Mixture of Gaussians to model road appearance for a robot driving on unpaved roads. Because we have little reason to believe that a particular parameterized model such as a Mixture of Gaussians faithfully captures the color/geometry relationship found in the sort of unstructured, outdoor environments involved in the LAGR program, and because we have no lack of examples (up to 600,000 samples a second) with which to build a non-parametric model, we have opted for the latter.

The choice of mapping color to geometry rather than directly to cost was motivated by two considerations. First, we expect to find a greater averaging effect when mapping color to cost that will result in poorer discrimination of obstacles. For some geometric features, such as height or slope, we would expect the neural network to learn fairly sharp boundaries between classes corresponding to small changes in the feature value around a threshold. Suppose now that we take several feature measurements for a color, most of which are above the threshold, but a few are below. The cost for this color, if determined directly, will be the average of the outputs of the neural network run on the individual measurements, which will be a blend of obstacle and non-obstacle costs. On the other hand, the mean geometric feature vector for this color, when run through the neural network, will preserve the sharp distinction so long as its elements remain above the learned threshold. Of course, it is possible to produce a step-change in the opposite direction in the cost of a color computed from predicted geometry with enough feature measurements that lie below the learned threshold. Determining cost from color-predicted geometry means that when we are wrong, we are more dramatically wrong. It is more important, however, to pick out as many lethal obstacles as possible, so the degree to which the system is wrong in classifying the type of obstacle is of less consequence. Results provided in Section III.C. confirm not only the learning of sharper distinctions in cost computed from color-predicted geometry, but also its more accurate fit to cost computed directly from measured geometry than that provided by cost mapped directly to color.

Our second motivation for mapping color to geometry was to preserve the ability to change our interpretation of geometric features online and have this reinterpretation be immediately applicable to color. This is directed toward future work on online learning where the neural network does not provide the only interpretation of geometry.

We have chosen to build the color predictive models of geometry using 16 3D histograms, two for each of the 8 geometric features. The histograms are indexed using intensity normalized red and green values as well as intensity itself. The first two dimensions are quantized into 64 bins, while the intensity dimension is quantized into 16 bins, yielding 65536 bins. A pair of histograms maintain a maximum likelihood estimate of the mean and standard deviation for each of the geometric features. The histogram bin $H_{\mu_j}^{RI,GI,I}$ storing the estimated mean value for feature j indexed to color

(RI, GI, I) is computed as follows:

$$H_{\mu_j}^{RI,GI,I} = \hat{\mu}_j^{RI,GI,I} \tag{1}$$

$$\hat{\mu}_j^{RI,GI,I} = \frac{1}{N}\sum_{i=1}^{N} x_j^{RI,GI,I} \tag{2}$$

where $x_j^{RI,GI,I}$ is the measurement for feature j for a point with color (RI, GI, I) and N is the total number of measurements of j for color (RI, GI, I). Similarly, for the estimated standard deviation of feature j indexed to color (RI, GI, I), we have:

$$H_{\sigma_j}^{RI,GI,I} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_j^{RI,GI,I} - \hat{\mu}_j^{RI,GI,I})^2} \tag{3}$$

With a data rate of 600,000 samples a second, there are on average 9 updates per bin per second, enabling rapid and dense populating of the histograms.

Once the histograms have been populated with an acceptable number of samples, the geometry prediction module begins filling in its own map of the environment. This is accomplished by looping over the rectified stereo reference image (in this case, the rectified image from the right camera) and retrieving the maximum likelihood estimate of the mean for each component $v_j^{RI,GI,I}$ of the geometric feature vector $\vec{v}^{RI,GI,I}$ by using the intensity normalized color values for each pixel in the image as indices into the histograms:

$$v_j^{RI,GI,I} = H_{\mu_j}^{RI,GI,I} \tag{4}$$

Once the predicted geometric feature vector has been assembled for a pixel, it is passed to the MLP to produce a cost. Costs for pixels for which there exists a range value computed from stereo can be placed directly into the color cost grid. The 3D position associated with a pixel that has no corresponding stereo range measure can be estimated using the known pose of the camera and intersecting the ray through the camera center and the pixel with the estimated ground plane.

Simply adding to the grid all of the costs from pixels whose positions are determined by ground plane intersection has a serious side-effect. Pixels on vertical objects are by definition not on the ground plane. By assuming that they are, we create large regions of possibly high cost that resemble shadows of the objects. These *cost shadows* can block off valid routes that the robot would otherwise explore. There are many ways that one might trim the excess cost regions. We have opted for a simple, fast method of tracing along each column in the image from the bottom to the top looking for the first group of high costs. The assumption is that this is the base of an obstacle, so only these first high costs are placed in the grid. This still produces a cost shadow for an object that is not perfectly vertical along its outside edge. In future work, we will be examining alternatives to this simple approach.

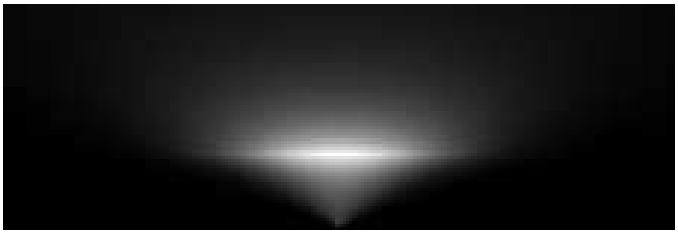

Fig. 2. Confidences in Stereo FOV

C. Merging Costs

The merging of costs occurs at two levels: (1) within a particular grid over the course of time as new assessments are made, and (2) between the stereo and the color-predicted grids. Merging within the stereo grid is governed by a confidence metric that takes into account the position of an updated cell within the stereo field-of-view. The color-predicted grid updates by simply overwriting cell data with the most recent information. Merging between these two grids is determined by a set of simple heuristics. Recent work in outdoor robotics [2],[4] using occupancy grids has highlighted the need for stable terrain assessments over time. If a robot's perception system makes rapid, substantial changes in its judgment of cost for a particular region in its grid, there can be corresponding dramatic changes in its desired path. If these changes occur as the robot is moving, the robot can be drawn along a path which is merely the result of sampling from two competing paths. If these paths pass on either side of an obstacle, the robot might be led toward the obstacle for a period of time as its changing terrain assessments cause an oscillation in which path is chosen.

One cause of unstable perceptual judgments is the variability of stereo matching results, particularly in outdoor environments. This is less a problem of mismatches than one of not finding matches at all. Stereo algorithms fail to find matches in the presence of complex shaped objects in part because no fixed relationship between the shapes of the correlation windows in two images can capture the true distortion of shapes of the objects in the images [14]. As the robot approaches an object, the viewpoints of the two cameras become increasingly different (reflected in the increasing disparity), and occlusion effects begin to dominate. However, as the distance to the object becomes greater and stereo range error increases as a second-order function of the range, we encounter a different problem, namely that the computed 3D points are increasingly placed in the wrong grid cells. This leads to the conclusion that there is a (possibly optimal) distance at which distortion and occlusion effects are limited but range accuracy is still acceptable. We have attempted to empirically determine this optimal range from logged stereo data and have placed it at about 2m from the cameras. We have also noted a degrading of stereo matching as the correlation windows are moved toward the edges of the field-of-view, due in part to the poorer quality of the images near the boundaries. The confidence metric incorporates these two constraints by multiplying an

exponential function of lateral displacement from the center of the stereo FOV by a function of range that has a maximum at 2m and drops off according to the square of the difference in range from 2m. It is expressed as follows (where x is the lateral offset from the center of the stereo FOV ϑ, r is the range from the stereo camera, and p is the distance of the beginning of the stereo FOV from the stereo camera):

$$Confidence(x, r) = \frac{G(F(r), x)}{U(r)} \tag{5}$$

$$U(a) = \begin{cases} (3.0 - a)^2 & \text{if } a \leq 2.0 \\ (a - 1.0)^2 & \text{if } a > 2.0 \end{cases} \tag{6}$$

$$G(a, b) = \exp\left(-\frac{b^2}{a^2}\right) \tag{7}$$

$$F(a) = 2.0(a - p)tan\left(\frac{\vartheta}{2}\right) \tag{8}$$

Equation 8 expresses the width of the stereo FOV at the given range. Figure 2 illustrates the distribution of confidences in the stereo FOV, with grayscale value encoding confidence. An update to a cell within the stereo grid that has a confidence value equal to or greater than the confidence of the current cell data can overwrite that cell.

The color-predicted grid will likely contain data both in regions that overlap with the stereo grid and regions that lie beyond the range of stereo. The costs from the latter regions are written directly into a merged grid, as are the regions of the stereo grid that do not overlap with filled regions in the color-predicted grid. Corresponding cells in the stereo and color-predicted grids that contain data in each are merged as follows: if the height variation from the stereo grid is above a threshold and the color-predicted cost is greater than the cost from stereo, update the merged cell with the color-predicted cost.

The reason for this update rule is that we only wish to enhance the costs in regions where stereo has some data that indicates the presence of an obstacle. Ruling out color-predicted updates to regions where stereo is normally useful but no data is present is done to avoid labeling with high cost portions of the ground that yield no stereo due to shadows, image saturation, or lack of texture.

III. Results

A. Test Platform and Procedure

Each participant in the Learning Applied to Ground Robotics program was supplied with two small robots developed by Carnegie Mellon University shown in Figure 1. They are differential drive vehicles with rear caster wheels. A WAAS enabled GPS provides position updates at 1 Hz, which are combined with odometry and the output of an inertial measurement unit (IMU) in an extended kalman filter (EKF). Sensors include two IR range finders and two stereo heads. The stereo baseline for each stereo pair is 11cm and the focal length is 4mm. The two heads are rotated toward the center of the vehicle by 20 degrees and tilted downward by 15 degrees. The horizontal FOV spanned by both stereo heads is 101 degrees. Baseline software from Carnegie Mellon includes a planner, a controller, a perception module, a pose estimation system, and a stereo library from PointGrey. Except for the controller, teams could replace any of these components. We opted to replace only the perception module with our own software as described here.

Fig. 3. Examples of LGT Courses

The teams participating in the program provide the LAGR Government Team (LGT) with a flash disk containing their executables before each test. The LGT has identical robotic platforms to those of the participants, and so can simply plug in the teams' flash disks and conduct tests at sites selected by the LGT. Participants are generally unaware of the exact nature of the test courses, so there is little opportunity for tuning a system for particular terrain. Courses have been reused for subsequent tests, though generally with modification. The goal of each test is to navigate the robot from a given start point to a given end point as set by the LGT prior to the test and not revealed to the participants. Each team has three runs per test. Figure 3 offers a sampling of some of the test sites. After each test, the LGT provides the teams with data logged from the test. The LGT scores each run on the basis of the time taken to reach the goal, and, if the robot fails to do so in the allotted time, the remaining distance to the goal. Failure to reach the goal is significantly penalized; driving fast has a considerable reward. The course lengths have ranged from 50-140 meters. The system described here has been the top performer according to the LGT's metric, averaged over all tests, and in terms of total number of successful runs. This is in spite of the fact that our system tends to drive slowly because of the speed-control logic in the baseline planner that ramps down speed in the presence of obstacles.

B. Neural Network-based Terrain Classification

To quantify the classification performance of the neural network, we computed the 4-fold cross validation score using the training set, evaluating several different network configurations. We also computed this score for a number of alternative classifiers. These alternative classifiers are naive Bayes, k-nearest neighbors (k=7), Parzen windows, the Fisher linear discriminant, and two SVMS, one using a 3rd degree

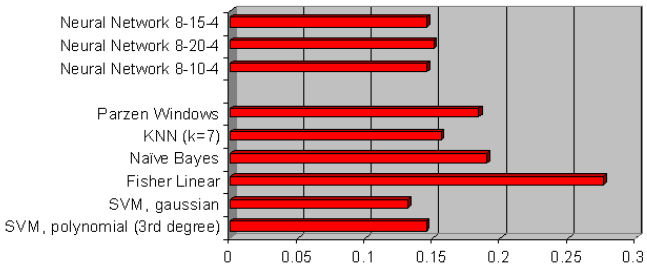

Fig. 4. Four-fold Cross Validation Error Score

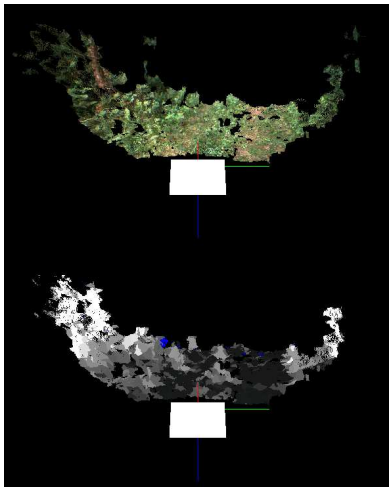

Fig. 5. Neural Network Terrain Classification (Grayscale maps to cost)

polynomial kernel, the other a gaussian kernel. As shown in Figure 4, the gaussian kernel SVM provides the best performance (13.1% error), slightly better than the neural networks (14.5-15% error) and the polynomial kernel SVM (14.4% error). However, both the gaussian and polynomial kernel SVMs required an excessive number of support vectors (750 and 1000 respectively). This in turn meant that using an SVM to classify a scene added significant lag when compared to the processing time of the neural networks (100 ms versus 2 ms to classify stereo snapshots from the right and left stereo heads). The slight gain in classification performance did not justify the considerable additional computational requirements.

Figure 5 provides a qualitative illustration of the effectiveness of the network on typical test terrain. Increasing grayscale values map to increasing cost. The network has proven to be extremely proficient at distinguishing terrain that a human expert would deem traversable but a naive measurement such as mere height variation would characterize as lethal. The robot averages less than one bumper hit per run over the course of government testing. It has also been demonstrated to be very robust to the presence of stereo artifacts.

C. Feature Saliency Analysis

Measurements of height and slope obtained from range data are undoubtedly the most widely used features for classifying terrain traversability for outdoor robots. In fact, these two features, either individually or in combination, are often deemed sufficient for determining traversability. Because our feature set includes elements beyond height and slope, we wished to determine the importance of these additional features within the neural net context. To do this, we have computed the saliency of each feature using the method of Optimal Cell Damage (OCD)[15], a feature-pruning algorithm inspired by the weight-pruning method of Optimal Brain Damage [16]. Optimal Brain Damage uses the Hessian matrix H to determine the weight saliency $S(w_j)$, which is the change in the training error that would occur if weight w_j were set to zero:

$$S(w_j) = \frac{1}{2}\frac{w_j^2}{H_{jj}^{-1}} = \frac{1}{2}\frac{\partial^2 J}{\partial w_j^2} w_j^2 \tag{9}$$

where

$$J = \frac{1}{2}\sum_{i=1}^{N}(t_i - z_i)^2 \tag{10}$$

is the sum-squared-error between the targets t_i and outputs z_i over the N elements of the training data. The saliency of an input variable, $S(i)$, offers a measure of how the training error would change if the ith variable were eliminated. It can be computed as the sum of the saliencies of the weights fanning out from the corresponding input unit i:

$$S(i) = \sum_{j \in fan-out(i)} S(w_j). \tag{11}$$

We have divided the computed input variable saliencies by the maximum to give a range of 0-1. The $Full$ dataset of Table I confirms that height and slope are the most important features for classifying traversability by our neural network. However, the remaining features have fairly strong saliencies and we would expect a noticeable increase in error if they were to be eliminated. Their importance is further confirmed by an analysis of feature saliency over those training examples that were included to improve robustness to stereo noise, shown in the $Outlier$ dataset of Table I. The relative saliencies of density and PCGP are greatly increased, while those of slope and raw height have diminished.

TABLE I
OCD FEATURE SALIENCY

Data Set	Feature							
	Height	Slope	PC	PCGP	Density	MeanZ	StdZ	NGP
Full	0.77	0.87	0.44	0.37	0.21	1.0	0.26	0.21
Outlier	0.57	0.36	0.05	0.66	0.88	1.0	0.06	0.09

D. Cost Enhancement from Color-predicted Geometry

Analysis of datasets from the LAGR program indicate that, on average, approximately 15-20% of the possible range data is lost due to failure to find a stereo match. This measure has a very high variance, with near complete matching in open terrain and almost total failure (up to 90% missing) when

TABLE II
ACCURACY OF COST PREDICTIONS USING GEOMETRIC COST AS BASELINE

Test Set	Mean Absolute Difference from Baseline Cost	
	Color-predicted Geometric Cost	Direct Color-predicted Cost
N.H. Forest	46.9	52.3
Virginia Brush	58.2	67.3
Texas Woods	67.9	83.6

close to obstacles. It is in the latter case that cost enhancement from color-predictions can make a dramatic difference in robot behavior, highlighting why emphasis should be placed on the accuracy of predicting obstacles.

To measure the effectiveness of color-predicted geometry cost enhancement, and to determine its worth relative to cost enhancement using a direct color-to-cost mapping, we have selected three datasets from the LAGR tests that represent very different environments and computed the mean difference of the costs computed by each method from those provided by the neural network. Because of our focus on finding obstacles, we look strictly at points lying above the ground plane. For each 3D point in a stereo image, we compute the geometry within its map cell and pass this to the neural network to determine a baseline cost. We then compute the absolute difference of this cost and the cost learned directly from color-cost associations, collecting a running total of these differences. Similarly, we compute the absolute difference between the baseline cost and that computed from color-predicted geometry. Using this measure, Table II illustrates that cost from color-predicted geometry is 10-20% more accurate than direct color-predicted cost.

Figure 6 offers a qualitative illustration of the typical color-predicted cost labeling of Viriginia brush. Increasing grayscale levels map to increasing cost. Obstacles in Figure 6 are clearly labeled as high cost. Figure 7 shows the merging of cost assessments in Figure 6, which clearly enhances the detection of obstacles. The red circles pick out the same bush in all three grids, showing the increase of its cost due to the color-predicted assessment. Testing has shown that the color-predicted cost enhancement can save the robot from collision with obstacles that are too close for adequate stereo measurements. This can happen often in cases where the robot makes a point turn in the midst of obstacles that it has not yet seen from its previous point of view (e.g., in a maze).

We have also argued that using color-predicted geometry can preserve step-changes in the cost function that would be smoothed out using direct color-cost predictions. To find the presence of these step-changes, at the end of runs we computed a final cost histogram, indexed by color, from the geometry histograms. To do this, we looked at every color index and extracted the corresponding 8 features from their histograms, passed these feature vectors to the neural network

Fig. 6. Color-predicted cost in test 7 (Grayscale maps to cost)

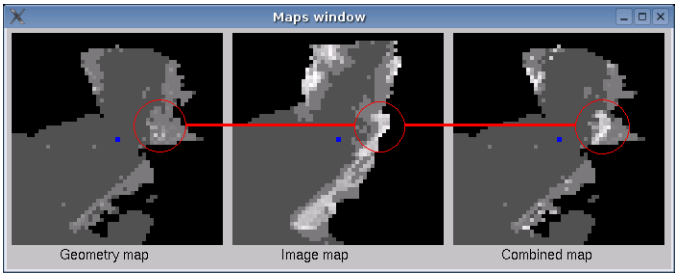

Fig. 7. Merging of costs shown in Figure 6

to compute a cost, then placed the cost in a new histogram indexed by the color. We then compared this histogram to the cost histogram representing the direct color-cost mapping. As expected, cost histograms derived from color-predicted geometry show prominent steps changes in cost, whereas the direct color-cost mappings display smooth surfaces. Figure 8 shows the cost histogram from direct color-cost mapping plotted against normalized red and green for a particular intensity level (red and green values are discretized into 64 bins). Figure 9 shows the same mapping for cost from color-predicted geometry for the same dataset. The differences in color-cost mapping between these two examples—smooth in the former, step-changes in the latter—can be found across intensity levels and across datasets.

E. Color-based Long Range Vision

To try to quantify the effects of color-based long range vision on robot behavior, we constructed a haybail cul-de-sac in otherwise open terrain. The robot is placed 25m away, near two haybails in view of the cameras. Ten runs were conducted using long range vision, and ten without. The goal was placed 20m beyond the cul-de-sac such that the direct path of the robot led into the cul-de-sac. All ten runs using long range vision avoided the cul-de-sac, all tens runs without it went directly into the cul-de-sac. On average, the cul-de-sac was fully detected by long range vision at a range of 15m. Figure 10 offers a qualitative assessment of long range color-predicted cost labeling in a more natural environment, showing the corresponding regions in the image and grid.

IV. CONCLUSION

We have presented an unsupervised method for learning color-based predictions of scene geometry that improves the terrain classification provided by a neural network trained to assess the traversability of unstructured, natural environments

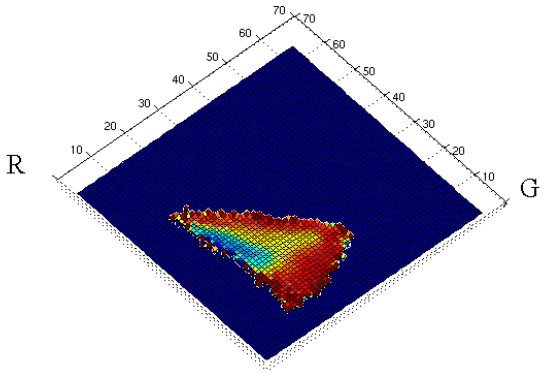

Fig. 8. Cost from direct color-cost mapping

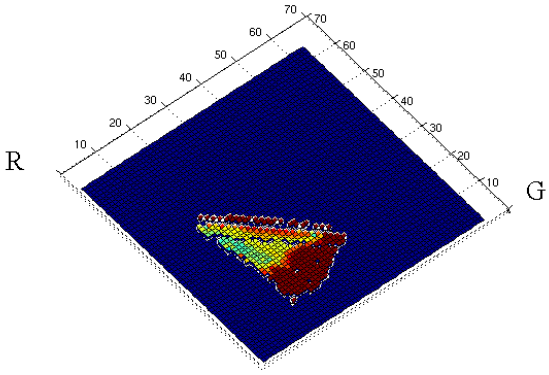

Fig. 9. Cost from color-predicted geometry

for a small mobile robot. The neural network is trained on eight-dimensional feature vectors representing measurements beyond height and slope, a departure from much of current terrain classification schemes for outdoor robots. The unsupervised learning method employs a non-parametric representation to capture the relationship between color and geometry in the scene. Imagery can then be used to enhance the geometric classification by filling-in in the near range when stereo fails to find matches and by adding in long range updates. Cost from color is computed by retrieving the color-predictions of geometry and classifying these with the neural network. Pixel positions are determined by ground plane projection in the absence of range data.

This system offers rapid, online, and adaptive classification of scenery without requiring a correct guess as to the underlying form of the distribution of colors in an environment. Learning geometry rather than cost from color provides the flexibility of being able to change the interpretation of geometry on the fly while preserving past learning.

Fig. 10. Long Range Color-predicted Cost with corresponding image regions

We have demonstrated in DARPA-conducted tests the efficacy of this method for terrain classification and mobile robot navigation. These tests were conducted over a wide range of environments and lighting conditions using both artificial and natural obstacles. Our system has been the top overall performer in average score and times to goal over the course of these tests.

ACKNOWLEDGMENT

This research has been financially supported through the DARPA LAGR program. The views and conclusions contained in this document are those of the authors, and should not be interpreted as necessarily representing policies or endorsements of the US Government or any of the sponsoring agencies.

REFERENCES

[1] A. Talukder, R. Manduchi, A. Rankin, and L. Matthies, "Fast and reliable obstacle detection and segmentation for cross-country navigation," *Proc. IEEE Intelligent Vehicles Symposium*, Versailles, France, June, 2002.

[2] A. Kelly, O. Amidi, M. Happold, H. Herman, T. Pilarski, P. Rander, A. Stentz, N. Vallidis, and R. Warner. "Toward reliable off road autonomous vehicles operating in challenging environments," *Proc. International Symposium on Experimental Robotics*, Singapore, June, 2004.

[3] C. Dima, N. Vandapel, and M. Hebert, "Classifier fusion for outdoor obstacle detection," *Proc. International Conference on Robotics and Automation*, April, 2004.

[4] M. Ollis and T. Jochem, "Structural method for obstacle detection and terrain classification," *Unmanned Ground Vehicle Technology*, 2003.

[5] C. Wellington and A. Stentz, "Learning predictions of the load-bearing surface for autonomous rough-terrain navigation in vegetations," *Proc. of the International Conference on Field and Service Robotics*, July, 2003.

[6] R. Manduchi, A. Castano, A. Talukder, and L. Matthies, "Obstacle detection and terrain classification for autonomous off-road navigation," *Autonomous Robots*, 19, 81-102, 2005.

[7] T. Hong, C. Rasmussen, T. Chang, and M. Schneier, "Road detection and tracking for autonomous mobile robots," *Proc. of the SPIE 16th Annual Symposium on Aerospace/Defence, Sensing, Simulation, and Controls*, Orlando, Florida, April, 2002.

[8] C. Rasmussen, "Combining laser range, color, and texture cues for autonomous road following," *Proc. of the International Conference on Robotics and Automation*, 2002.

[9] D. Lieb, A. Lookingbill, and S. Thrun, "Adaptive road following using self-supervised learning and reverse optical flow," *Proc. Science and Systems 1*, August, 2005.

[10] D. Pomerleau, "Alvinn: an autonomous land vehicle in a neural network," *Advances in Neural Information Processing Systems 1*, pp 305-313, 1989, Denver, Colorado.

[11] L. Jackel, "DARPA LAGR," *NIPS 2005 Workshop on Machine Learning Based Robotics in Unstructured Environments*, December, 2005.

[12] A. Elfes, "Occupancy grids: A stochastic spatial representation for active robot perception," *Proc. of the Sixth Conference on Uncertainty in AI*, July, 1990.

[13] G. Bradski, A. Kaehler, and V. Pisarevsky, "Learning-based computer vision with Intel's Open Source Computer Vision Library," *Intel Technology Journal*, May, 2005.

[14] A. Kelly and A. Stentz, "Stereo vision enhancements for low-cost outdoor autonomous vehicles," *International Conference on Robotics and Automation*, May, 1998.

[15] T. Cibas, F. Fogelman Soulie, P. Gallinari, and S. Raudys, "Variable selection with optimal cell damage," *Proc. of ICANN'94*, 1994.

[16] Y. LeCun, J. Denker, and S. Solla, "Optimal brain damage," *Advances in Neural Information Processing Systems 2*, 1990.