

# Learning for Ground Robot Navigation with Autonomous Data Collection

Jie Sun James M. Rehg Aaron Bobick  
GVU Center / College of Computing  
Georgia Institute of Technology  
{sun, rehg, afb}@cc.gatech.edu

## Abstract

Robot navigation using vision is a classic example of a scene understanding problem. We describe a novel approach to estimating the *traversability* of an unknown environment based on modern object recognition methods. Traversability is an example of an affordance jointly determined by the environment and the physical characteristics of a robot vehicle, whose definition is clear in context. However, it is extremely difficult to estimate the traversability of a given terrain structure in general, or to find rules which work for a wide variety of terrain types. However, by learning to recognize similar terrain structures, it is possible to leverage a limited amount of interaction between the robot and its environment into global statements about the traversability of the scene. We describe a novel on-line learning algorithm that learns to recognize terrain features from images and aggregate the traversability information acquired by a navigating robot. An important property of our method, which is desirable for any learning-based approach to object recognition, is the ability to autonomously acquire arbitrary amounts of training data as needed without any human intervention. Tests of our algorithm on a real robot in complicated unknown natural environments suggest that it is both robust and efficient.

## 1 Introduction

Robot navigation using vision is a classic example of a scene understanding problem, requiring inference beyond pure geometric reasoning. In particular, successful navigation requires the robust assessment of the *affordance* of traversability. As originally conceived by Gibson, an affordance is not just a property of the environment, but is jointly determined by the environment and the organism [8]. The problem domain for this paper is an outdoor autonomous robot. In this case, the affordance of traversability is determined by whether the robot can successfully move over a particular ground location. In this paper we present a novel method of learning to infer the affordance of traversability from appearance of terrain in an outdoor environment.

Typically, in constrained environments, is it is known a priori that a particular affordance can be directly inferred from a measurable physical property. For example, robot navigation in an indoor office environment relies heavily on geometry [3, 15]. The assumption employed is that if the world geometry in front of the robot is a flat plane, then that area is traversable. This methodology has been actively pursued precisely because, in fact, many indoor robot applications are situated in contexts where this assumption is valid. In other domains, such as conventionally marked automobile highways, different mappings can be exploited [4]. Special sensors, such as stereo pairs and range sensors certainly help navigation [1, 16, 11, 13, 2], yet they do not fundamentally solve the obstacle detection problem, because geometry does not directly encode traversability.

## 2 Learning to Map from Appearances to Affordance

Our approach to inferring traversability is premised on two key assumptions. The first of these is the assumption that image appearance and geometry contain information about traversability affordance. More specifically, two terrain locations which have similar appearances across viewpoints will have the same traversability properties (which we define to be `obstacle` or `not-obstacle`). We exploit this assumption by asserting that if a particular terrain location is similar in appearance to locations predominantly known to be traversable, then it also should be traversable. Notice this does not imply that all traversable terrain locations look similar — there is no simple appearance or geometric description of traversable regions.

The informal basis of this assumption is that in a given environment there will be a variety of terrain structures. For example, an area may consist of grassy slopes, gravel roads, dirt paths, woven thickets, and trees. At each point of time the robot can measure both the photometric appearance with a standard camera and the geometric structure from a stereo pair. An example of the imagery is shown in Fig-

ure 1. On occasion the geometry is such that the robot can make a direct inference of traversability, e.g. a broad vertical wall. But more importantly, the robot will assume that currently viewed location will behave similarly with respect to traversability as a previously sampled structures that have a similar appearance.

A key element of our approach with respect to this assumption is that the robot often views the same terrain location multiple times, from varying distances and viewpoints. Our method is designed to integrate this information by utilizing all observations and comparing them to previously sampled observations from similar distances. In this manner, the confidence that a terrain region is not an obstacle is reinforced through accumulated evidence.

The notion of previously sampled locations invokes our second key assumption: in general it will not be possible in complex vegetative domains to specify a priori a “traversability detector”, an algorithm that reliably compute whether a terrain area will support robot traversal. Rather the system will need to *learn* which structures are traversable and which are not.

Recently, machine learning has been successfully applied in both the vision based navigation [9, 7, 10] and object detection/recognition [12, 17]. However, such learning approaches involve a large amount of manual work in labelling and collecting training data. The object categorization work in [5] requires few learning instances for each class, yet manual data collection for a great number of classes is still a problem.

Our approach to learning is different in two significant respects. First, we take advantage of the robot being able to directly perform experiments in the environment. Given a potentially traversable terrain location, the robot can proceed (cautiously!) using its bumper and motor state sensors to determine whether its forward motion is being impeded. This gives the learning system the ability to collect arbitrary amounts of training data as needed through this independent channel.

The second difference is that the robot will receive an unbalanced number of samples of the two categories of `obstacle` and `non-obstacle`. Most of the time (one hopes) the system will be proceeding through appropriately planned trajectories such that the terrain structures it visits are usually not obstacles. It is thus not feasible to learn either the prior probability of obstacles  $P(O)$  or any probabilities conditioned on obstacles reliably due to lack of observational data. Thus we model the likelihood of observations associated with non-obstacles. If a particular appearance has been observed numerous times when viewing non-obstacles in the past, then it is unlikely that this appearance actually arises from an obstacle. This likelihood is updated whenever new traversability information is obtained. Repeated sampling of this appearance when observing non-



Figure 1: Input from one stereo pair. (a)RGB image from one camera; (b)Height map from stereo.

obstacles strengthens the robot’s expectation that such appearance is linked with “traversable”. On the other hand, encountering an actual obstacle greatly reduces the likelihood of all the appearances associated with this obstacle to be views of non-obstacles.

### 3 Feature Space Representation

As an index over which to integrate our observations, we carve the terrain into a two-dimensional cartesian grid of cells, with each cell representing a  $0.2m \times 0.2m$  square element of terrain [14]. Over time, traversability about a cell location is being made based on observations of it at different distances. In this section we describe the feature space that we design to represent all of the observations associated with a cell location.

We represent one observation by a pair consisting of the appearance  $b_i$  and the distance  $d_i$  at which the cell is viewed  $(b_i, d_i)$ . Distance  $d_i$  can be reliably measured by the stereo system, and we design  $b_i$  to be a fixed size image patch corresponding to that cell. With the assumption that similar observations do infer similar traversability, we want to map the appearance  $b_i$  in a feature space that reflects visual similarity. To handle the large variation of the lighting condition and the terrain structure reflectance, we use the orientation histogram representation. For each image patch, its gradient in the image intensity domain is calculated, and a histogram of the gradient orientations is calculated [6]. Orientational histogram is robust against the change of lighting condition because although intensity values change, the orientations of intensity derivatives do not change. It also works if the image patch is completely in a shadow. In practise, we calculate the orientation histogram in 8 bins, 45 degrees each. Since for each pixel, stereo gives its corresponding 3D position and height, we can also take advantage of the geometry information. Thus we calculate a 5 bin histogram of the corresponding height for each pixel, with the bin division based on the obstacle height range with respect to the robot size. The appearance  $b_i$  is thus in a 13 dimensional space  $[p_x p_z]$ ,

which is a cartesian product of the 8 dimensional visual feature space of orientation histogram ( $p_v$ ) and a 5 dimensional geometry feature space of height distribution ( $p_z$ ). Note that the appearance vectors are intrinsically in a 11 dimensional space, because  $\sum_i p_v^i = \sum_j p_z^j = 1$  as by definition. The dissimilarity measure between two feature vectors is based on the standard Chi-square distance for distributions:

$$Dist(p, q) = w_v \sum_{i=1}^8 \frac{(p_v^i - q_v^i)^2}{2(p_v^i + q_v^i)} + (1-w_v) \sum_{j=1}^5 \frac{(p_z^j - q_z^j)^2}{2(p_z^j + q_z^j)} \quad (1)$$

The weight  $w_v$  weights visual feature more than geometry feature, with the assumption that visually similar image patches usually have similar geometry. However, this combinatory feature also prevents clustering visually similar but geometrically distinct pathes together. In the following section, we describe a procedure for mapping the appearance  $b_i$  to a set of class labels  $a_i$ . With this feature space for one observation, the feature of a cell over time is represented by the powerset of these features  $(A, D) = \{(a_i, d_i)\}$ . In the next section, we show how to group similar observations together, so as to associate with traversability information.

## 4 Online Appearance Quantization

The traversability information is to be associated with a group of similar observations rather than to individuals. It is important to further reduce the dimensionality such that a reasonable number of groups can to be maintained. It is easy to discretize the distance  $d_i$  into a fixed number of bins, so here we concentrate on grouping visually similar patches together.

Our algorithm represents each cluster as a hypersphere of constant radius  $r$  in the 11 dimensional vector space. The radius is predefined so as to control the maximum number of clusters that can possibly be formed. Since the terrain structures in a natural environment are usually not arbitrarily distributed, a certain number of clusters do reliably cover the majority of terrain structure views. We start with the first frame using a K-means clustering algorithm with a small value of  $K$ . Later observed patches are compared to existing cluster centers and one of the following actions are performed:

1. If the patch is within radius  $r$  (as measured by Chi-square distance) to a particular cluster  $C_i$  with mean  $\mu_i$ , we label this patch as belonging to class  $C_i$ , and update the cluster mean  $\mu_i$ .
2. If all existing  $n$  cluster means are further than radius  $r$  away, we construct a new class  $C_{n+1}$ , using the feature vector of the image patch as the class mean  $\mu_{n+1}$

This clustering algorithm serves as an online appearance quantization method, that each single appearance will be assigned to one of the clusters based on the similarity between this single appearance and the cluster mean. Note that we don't require the clusters to have semantic meaning, such as a cluster that corresponds to "grass". With this representation, observations are recorded as "an instance of appearance class  $C_j$  at a distance  $d_i$ " ( $a_i = C_j, d_i$ ). Consider a typical example of the robot moving to a particular location. If most single observations ( $a_i, d_i$ ) of the location on the way are frequently encountered when the robot in the past moved to non-obstacles, it is very likely that the destination is also a non-obstacle. On the other hand, if the observations on the way rarely appeared before when the robot was moving to a non-obstacle, then conservatively suspecting the destination being an obstacle is a wise choice. This goes back to our assumption that there are no instances in the environment which look like non-obstacles from most view points, but are in fact obstacles. This online clustering algorithm is fast, and it reduces the dimensionality of patch feature vectors significantly. Experiments suggest that even in a very complicated natural environment, fewer than 1000 clusters suffice.

Later when image patches are observed, they are classified to the nearest cluster center, and the height distribution of that cluster is used instead of the output from stereo, for the purpose of classifying traversability. This addresses the noisy stereo output, which is often the case in an outdoor unconstrained environment (figure 1(b)). One important point in clustering is that we don't prune small clusters, because obstacles are usually rare, so it is very likely that the small clusters are from image patches on obstacles. The fact of greatly reduced dimensionality has enable us to keep all the likelihood information as associated with every possible observation.

## 5 Learning Traversability with Autonomous Data Collection

Here we describe two key mechanisms in our overall learning framework: an ensemble classification rule for predicting the traversability of a grid location in world coordinates based on a set of observations obtained by the robot, and a learning rule for updating the ensemble rule over time to reflect the experience of the robot. The interaction between these two components allows the robot to make increasingly accurate predictions about the traversability of regions of its environment based on its navigation experience.

## 5.1 Ensemble Classifier for Traversability

We assume that we have  $n$  observations of a given cell location  $x, y$ . The observations make up the composite feature  $(A, D) = \{(a_i, d_i)\}_{i=1}^n$ , where for each  $i$ ,  $a_i$  is the class label observed by the robot at a distance  $d_i$  away. We also assume that we have access to a base classifier  $h(a_i, d_i)$ , which predicts whether location  $x, y$  is traversable given a single observation. A simple choice for  $h(a_i, d_i)$  is to take the center of the cluster corresponding to the label  $a_i$  and count the percentage of times the height distribution exceeds a threshold above the ground plane. The threshold is chosen to identify obstacles based on their height above the ground plane. Note that if the base classifier made consistently good predictions, then there would be no need to use learning methods to achieve successful navigation. In practice, however, it is very difficult to make accurate predictions about traversability based solely on the image and stereo data obtained over a small window of pixels. Note also that more sophisticated and powerful base classifiers could be employed without changing our basic approach to ensemble classification.

We further assume that the base classifier is tuned conservatively, meaning that its false negative rate is very low (it will rarely miss an obstacle when one is present) but as a consequence its false alarm rate is fairly high (it often reports obstacles in locations where no obstacle is actually present). This is a reasonable choice given our assumption that non-traversable regions are fairly rare. Many real-world robot navigation systems take a similar conservative approach.

Under the given conditions, the primary goal of the ensemble classifier is to learn an effective discount factor  $\beta_i$  for each base classifier. The purpose of the discount factor is to ignore the votes of the base classifiers in cases where the likelihood that the feature values come from a traversable region are high. Let  $H(A, D)$  where  $0 \leq H \leq 1$  be the ensemble prediction, corresponding to the likelihood of an obstacle being present. When  $H(A, D) = 0$  the region is traversable. We use the following form for the ensemble rule:

$$H(A, D) = \frac{\sum \beta_i w_{d_i} h(a_i, d_i)}{\sum w_{d_i}}$$

$$\beta_i = \begin{cases} 1 & \text{if } L(a_i|\overline{O}, d_i) < T \\ 0 & \text{if } L(a_i|\overline{O}, d_i) \geq T \end{cases} \quad (2)$$

where  $w_{d_i}$  are the weights associated with the discrete bins of  $d_i$  and  $T$  is a pre-defined threshold. The  $w_{d_i}$  are chosen to weight close-by observations more heavily than far-off ones.

The factor  $L(a_i|\overline{O}, d_i)$  measures the likelihood of observing a particular class label  $a_i$  at a distance  $d_i$  given that the cell is traversable. The higher this likelihood is

for a particular feature  $(a_i, d_i)$ , the less attention should be paid to the vote by the corresponding base classifier at that particular location. Although we could allow  $\beta_i$  to be a continuously-varying weight, we have found that we get better results if we threshold it and obtain a zero-one ensemble weight. Only in cases where we have not observed a particular feature as being traversable a significant number of times will we rely upon the (conservative) base classifier.

Note that other classification approaches are possible. Perhaps the most straight-forward approach would be to model the probability of traversability,  $P(\overline{O}|A, D)$ , directly and obtain an expression for it using Bayes rule:

$$P(\overline{O}|A, D) = \frac{P(A|\overline{O}, D)P(\overline{O}|D)}{P(A|\overline{O}, D)P(\overline{O}|D) + P(A|O, D)P(O|D)} \quad (3)$$

$$= \frac{P(A|\overline{O}, D)P(\overline{O})}{P(A|\overline{O}, D)P(\overline{O}) + P(A|O, D)P(O)}$$

One difficulty in this approach is the complex structure of the joint observation density  $P(A|\overline{O}, D)$ . This is a distribution over the power set of combinations of observations. Another difficulty is the fact that it is very hard to determine the prior distribution over traversability,  $P(O)$ . This prior probability will change dramatically with the characteristics of the site and can even change significantly for different regions within a site.

## 5.2 Autonomous Learning of Traversability

The robot obtains high quality traversability information as it explores its environment. Each cell which the robot can successfully pass through can be labelled as traversable, otherwise it is labelled as untraversable. Bumper and IR sensors mounted at the front of the robot, as well sensor data on the state of the robot's motors, can be used to collect evidence of the lack of forward progress by the robot which is associated with an untraversable cell. A key point about this data collection process is that an essentially arbitrary amount of high-quality labelled data can be obtained without any human intervention. The process for inferring traversability is extremely robust and accurate. Through learning we leverage this high-quality information obtained for a sparse number of map locations into a substantial increase in the overall (global) accuracy of the map.

Let  $O(x, y)$  denote the traversability of a ground cell with center  $(x, y)$ .  $O(x, y) = 1$  if the cell is occupied by an obstacle,  $O(x, y) = 0$  if it is not. Initially all  $O(x, y)$  are set to  $-1$  throughout the occupancy grid. The main data structure for traversability learning is the list of observation likelihoods  $\{L(a_i|\overline{O}, d_i)\}_{i=1}^n$  corresponding to each possible feature value. These likelihoods are initialized to zero and then updated according to the navigation experience of the robot. The update rule goes as follows:

1. If the ground cell is traversable, set  $O(x, y) = 0$ ; up-

date class likelihoods:

$$\text{Count}(C_j|\bar{O}, d_i) = \text{Count}(C_j|\bar{O}, d_i) + 1 \quad (4)$$

for each evidence  $(C_j, d_i)$  associated with this cell  $(x, y)$ , where  $1 \leq j \leq n$  are quantized appearance clusters, and  $1 \leq i \leq 6$  are the 6 distance bins. Essentially, we have estimates of:

$$L(C_j|\bar{O}, d_i) = \frac{\text{Count}(C_j|\bar{O}, d_i)}{\sum_j \text{Count}(C_j|\bar{O}, d_i)} \quad (5)$$

2. Else, the ground cell is occupied by an obstacle, set  $O(x, y) = 1$ ; update observation likelihoods:

$$\text{Count}(C_j|\bar{O}, d_i) = \lambda \times \text{Count}(C_j|\bar{O}, d_i) \quad (6)$$

where  $0 < \lambda < 1$  is a penalization factor (we use  $\lambda = 0.5$ ).

Note that there is a strong resemblance between the preceding update rule and reinforcement learning techniques for navigation in grid-worlds. The primary difference is our use of class labels based on the visual appearance of image patches to bias the update strategy towards regions with visual similarity. This reflects our hypothesis that regions in the scene with similar appearance over distance will have similar traversability characteristics. The efficacy of our approach over standard temporal integration techniques demonstrates the value of this additional source of information.

### 5.3 Algorithm Implementation

Our algorithm consists of online learning and classification of traversability. In the world frame, the ground is divided into equal sized square cells, with  $(x, y)$  representing the center of the cell. Let  $O(x, y)$  denote the traversability of this ground cell,  $O(x, y) = 1$  if it's occupied by an obstacle,  $O(x, y) = 0$  if not. Initially all  $O(x, y)$  is set to  $-1$ . For each frame, do the following:

1. *Online appearance quantization.* For each image patch observed in the current frame:
  - (a) Classify this patch with a base class label, let  $a_i = C_j$ . Let  $d_i$  be the discretized distance from the center of the image patch to the robot center.
  - (b) Record evidence  $(a_i, d_i)$  associated with the ground cell  $(x', y')$  where the image patch is located in the world frame. Update the class mean vector and construct new classes if necessary, as described in section 4.

2. *Learning traversability.* Examine the traversability of the current cell. If  $O(x, y) = 0$  or  $1$ , the robot already has visited this location, no new information is obtained. Otherwise,  $O(x, y) = -1$  and the class likelihoods can be updated as in section 5.2.
3. *Traversability classification.* Use equation 2 in section 5 to update the traversability belief of each ground cell  $(x, y)$ , given its observations  $(A, D) = \{(a_i, d_i)\}$ .

One major focus of our algorithm is to learn the likelihoods of observations about non-obstacles at certain distance, as represented by  $L(C_j|\bar{O}, d_i)$ . Since chances of bumping into an obstacle are rare, especially when the algorithm is working, we don't model  $L(C_j|O, d_i)$  directly. Instead, we penalize  $L(C_j|\bar{O}, d_i)$  so as to reduce our belief of a non-obstacle, when a patch of class  $C_j$  is viewed about an obstacle at distance  $d_i$ . The novel aspect of this approach is to strengthen the belief of a non-obstacle when evidence are obtained, yet severely reduced the belief when obstacles are encountered.

Figure 2 illustrates the information flow in our algorithm. There are number of ground cells  $(x, y)$  which we want to predict the likelihood of having an obstacle in the cell. Each ground cell has a number of observations, and cells share similar observations. We provide an example to show how information flows for learning and classification. Suppose the cells in the figure are not visited yet, thus  $\{O(x_i, y_i) = -1\}_{i=1}^p$ . Now a measurement  $M$  of  $(x_1, y_1)$  is given. If there is no obstacle in this cell, meaning  $M = 0$ , then the observation class likelihoods that are associated with this cell are increased. These three observations, i.e.  $(a_1, d_1), (a_2, d_2)$  and  $(a_4, d_4)$ , get higher likelihood as being views of non-obstacles. They then affect the obstacle classification for cells  $(x_2, y_2), (x_3, y_3)$  and  $(x_k, y_k)$ , to reflect the newly obtained traversability data. The likelihoods of having an obstacle in each of these three cells are decreased. On the other hand, if measurement  $M$  of  $(x_1, y_1)$  is actually an obstacle, then the class likelihoods of the three observations are greatly reduced (according to equation (6) in section 5.2), thus causing  $(x_2, y_2), (x_3, y_3)$  and  $(x_k, y_k)$  to be more likely an obstacle.

## 6 Experiments and Results

Our algorithm is tested on a real robot (figure 3). The robot has two stereo pairs, each of them consisting of two cameras, that give us both the images from the camera and the stereo information. It also has an a GPS and an Inertial Measurement Unit (IMU), that gives the global position and pose of the robot. The bumper switch on the robot can give control feedback whether it has hit an obstacle or not. Here we present two test results, one in a parking deck and the other in a forest.

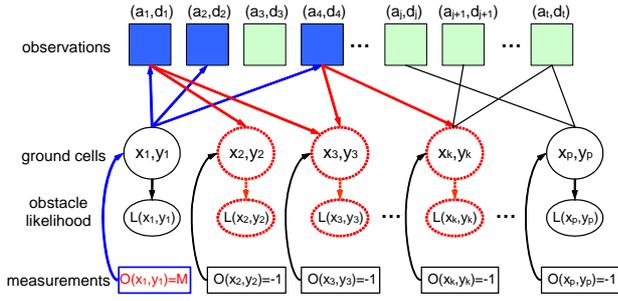


Figure 2: Learning and classifying traversability. Blue arrows: newly acquired traversability data updates the related observation class (blue squares) likelihoods. Red arrows: the change of observation class likelihoods is then reflected in traversability classification for the cells that share similar observations.

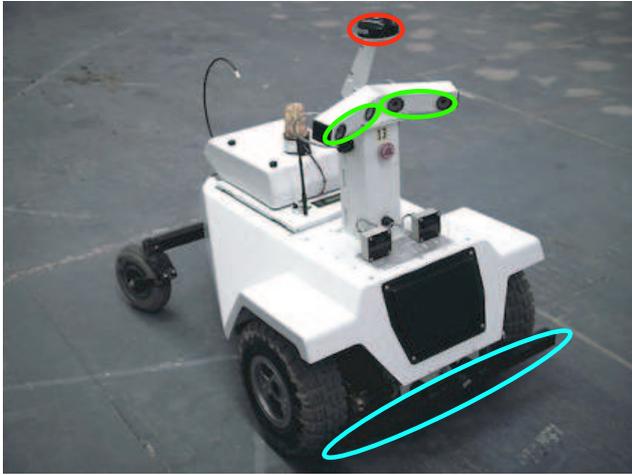


Figure 3: Robot for algorithm testing in outdoor natural environment. It has a GPS system (red), two stereo pairs (green) and a bumper (blue).

The parking deck experiment has 3 trash cans with camouflage cover as the only obstacles. Here we show two comparisons. The robot drives with the control of a simple algorithm that only looks at the current frame to find obstacles based on the stereo output. The robot turns whenever an obstacle is observed within 2 meters range. The red curve in figure 4 denotes the robot path, each sharp turn reflects a close obstacle being seen by this algorithm. It is clear that such algorithm, without taking advantage of temporal integration, suffers from significant false alarms. Figure 4(a) is the obstacle evidence grid build from our algorithm. The three blue circles denote the trash cans. During the experiment, the robot never hit a trash can, but with the traversable information collected about the ground, our algorithm filtered the ground out nicely, and conservatively classified

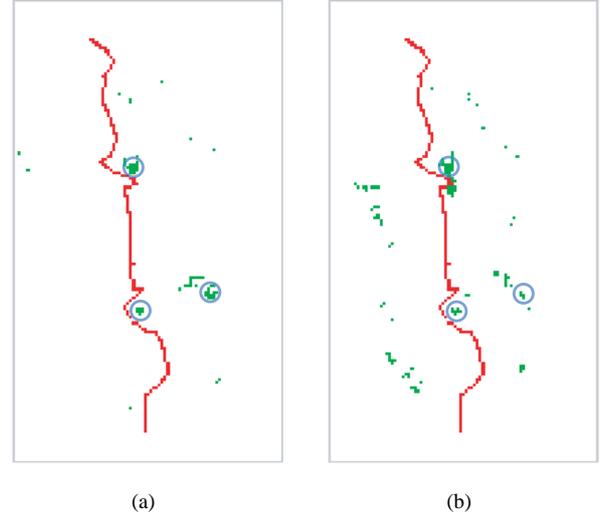


Figure 4: Obstacle evidence grid (size  $19m \times 33m$  with each pixel corresponding to  $0.2m \times 0.2m$  cell on the ground). Red curve denotes the robot’s path from bottom to top, blue circles denote ground truth where the obstacles are, green pixels denote high likelihood of an obstacle in that cell. (a) Output of our algorithm; (b) Output of standard temporal integration.

the trash cans as obstacles.

Figure 4(b) is the result from the standard temporal integration for a comparison. Two major problems contribute to the noisy result in this case. First, traversability evidence is integrated over time so as to eliminate accidental false alarms but not consistent ones. Consistently, parking slot marking lines on the ground are reported by stereo as having height above 50cm, thus these “obstacles” cannot be filtered out by only temporal integration. Second, the traversability belief of a cell location never gets updated if no new observations are made about the cell. Our approach can solve both of these problems, because it associates visually similar appearances together, and infers about the class traversability in general rather than doing it separately for every single observation.

The second experiment is performed in a forest under significant tree cover. We manually control the robot to drive in a glade that has a variety of terrain structures. Big trees sparsely distributed are the major obstacles in the scene. There are also a number of saplings that the robot can drive over 5(c)(d). The comparison result between our algorithm and the standard temporal integration algorithm is presented in figure 5(a)(b). The key observation here is that saplings have 3D geometry structure resembling obstacles, thus with temporal integration alone, they cannot be filtered out as non-obstacles. Our algorithm learns the traversability of similar saplings by driving over several of them. Fig-

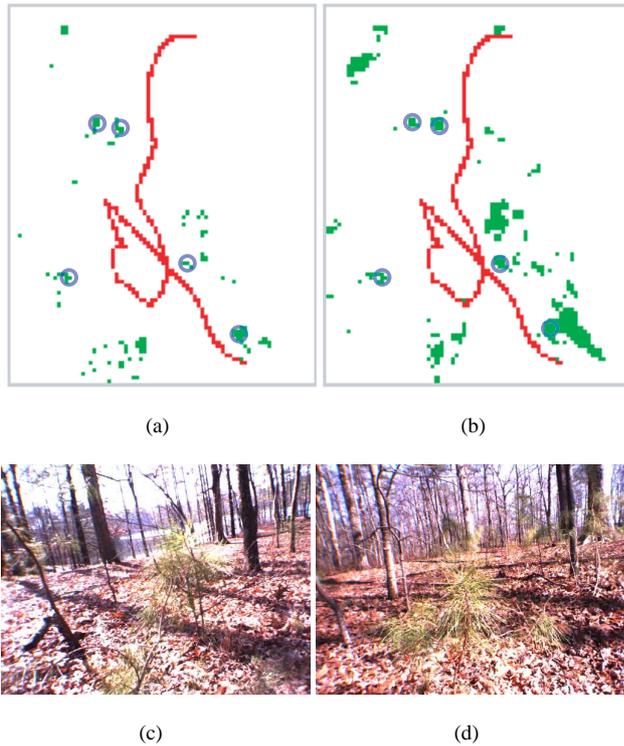


Figure 5: Obstacle evidence grid (size  $16m \times 20m$  with  $0.2m$  resolution). Trees of big size are marked with blue circles. (a) Output of our algorithm; (b) Output of standard temporal integration; (c) The robot drives over a sapling; (d) Similar saplings in the environment that are traversable.

Figure 5(a) suggests that our algorithm produces far less false alarms while locating the major obstacles (big trees) accurately.

## 7 Conclusions and Future Work

We have described a novel method for learning about the traversability affordance of terrain locations. Based on the assumption that terrain locations with similar appearance do have similar traversability, our method learns to integrate evidence from appearances to predict traversability. Our method has the desirable property of collecting arbitrary amounts of training data as needed, and the performance improves over time. The algorithm is being implemented on a real robot as a part of the DARPA LAGR project, which will be integrated with the robot planning module.

## References

[1] P. Bellutta, R. Manduchi, L. Matthies, K. Owens, and A. Rankin. Terrain perception for demo iii. In *Intelligent*

*Vehicle Symposium*, 2000.

- [2] H. Choset, K. Nagatani, and N. A. Lazar. The arc-transversal median algorithm: A geometric approach to increasing ultrasonic sensor azimuth accuracy. *IEEE Trans. Robotics and Automation*, 19(3):513–523, 2003.
- [3] G. N. DeSouza and A. C. Kak. Vision for mobile robot navigation: A survey. *IEEE Trans. PAMI*, 24(2):237–267, 2002.
- [4] E. D. Dickmanns and B. Mysliwetz. Recursive 3d road and relative egostate recognition. *IEEE Trans. PAMI*, 14(2):199–213, 1992.
- [5] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In *Workshop on Generative-Model Based Vision, CVPR 2004*, 2004.
- [6] W. T. Freeman and M. Roth. Orientation histograms for hand gesture recognition. In *Int'l. Workshop on Automatic Face and Gesture Recognition*, pages 296–301, 1995.
- [7] P. Gaussier, C. Joulain, S. Zrehen, and A. Revel. Visual navigation in an open environment without map. In *IEEE Conf. Intelligent Robots and Systems*, pages 545–550, 1997.
- [8] J. J. Gibson. *The Ecological Approach to Visual Perception*. Houghton Mifflin, Boston, 1979.
- [9] T. Jochem, D. Pomerleau, and C. Thorpe. Vision-based neural network road and intersection detection and traversal. In *IEEE Conf. Intelligent Robots and Systems*, pages 344–349, 1995.
- [10] C. Joulain, P. Gaussier, A. Revel, and B. Gas. Learning to build visual categories from perception-action associations. In *IEEE Conf. Intelligent Robots and Systems*, pages 857–864, 1997.
- [11] A. Rieder, B. Southall, G. Salgian, R. Mandelbaum, H. Herman, P. Rander, and T. Stentz. Stereo perception on an off-road vehicle. In *Proc. Intelligent Vehicles*, 2002.
- [12] H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Trans. PAMI*, 20(1):23–38, 1998.
- [13] A. Stentz, A. Kelly, P. Rander, H. Herman, O. Amidi, R. Mandelbaum, G. Salgian, , and J. Pedersen. Real-time, multi-perspective perception for unmanned ground vehicle. In *Proc. AUVSI2003*, 2003.
- [14] S. Thrun. Learning occupancy grids with forward models. *Autonomous Robots*, 15:111–127, 2003.
- [15] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 128(1), 2000.
- [16] N. Vandapel, D. F. Huber, A. Kapuria, and M. Hebert. Natural terrain classification using 3-d ladar data. In *IEEE Int'l Conf. Robotics and Automations*, 2004.
- [17] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. CVPR*, pages 511–518, 2001.