Online Learning for Offroad Robots: Using Spatial Label Propagation to Learn Long-Range Traversability

Raia Hadsell[1,2], Pierre Sermanet[2], Jan Ben[2], Ayse Naz Erkan[1], Jeff Han[1], Matthew K. Grimes[1], Sumit Chopra[1], Yury Sulsky[1,2], Beat Flepp[2], Urs Muller[2], Yann LeCun[1]

(1) Courant Institute of Mathematical Sciences, New York University

(2) Net-Scale Technologies, Morganville, NJ 07751, USA

Abstract

We present a solution to the problem of long-range obstacle/path recognition in autonomous robots. The system uses sparse traversability information from a stereo module to train a classifier online. The trained classifier can then accurately classify the entire scene. A distance-normalized image pyramid makes it possible to efficiently train a learning network on each frame seen by the robot, using large windows that contain contextual information as well as shape, color, and texture. Traversability labels are initially obtained for each target using a stereo module, then propagated to other views of the same target using temporal and spatial concurrences, producing a classifier that is largely view-invariant. A ring buffer simulates short-term memory and ensures that the discriminative learning is balanced and consistent. This long-range obstacle detection system (LROD) sees obstacles and paths at 30-40 meters, far beyond the maximum stereo range of 12 meters, and adapts very quickly to new environments. Experiments were run on the LAGR robot platform.

1 Introduction

The method of choice for vision-based driving in off-road mobile robots is to construct a traversability map of the environment using stereo vision. In the most common approach, a stereo matching algorithm, applied to images from a pair of stereo cameras, produces a *point-cloud*, in which the most visible pixels are given an XYZ position relative to the robot. A traversability map can then be derived using various heuristics, such as counting the number of points that are above the ground plane in a given map cell. Maps from multiple frames are assembled in a global map in which path finding algorithms are run [5, 7, 2]. The performance of such stereo-based methods is limited, because stereo-based distance estimation is often unreliable above 10 or 12 meters (for typical camera configurations and resolutions). This may cause the system to drive as if in a self-imposed “fog”, driving into dead-ends and taking time to discover distant pathways that are obvious to a human observer (see figure 1). Human visual performance is not due to better stereo perception; in fact, humans are excellent at locating pathways and obstacles in monocular images.

We present a learning-based solution to the problem of long-range obstacle and path detection, using the paradigm of *near-to-far learning* or *bootstrap learning*. Near-to-far learning provides a way to see beyond the limits of stereo by learning traversability labels from stereo-labeled image patches in the near-range, then classifying image patches in the far-range. If this training is done online, the robot can adapt to changing environments while still accurately assessing the traversability of distant areas. However, the approach is limited to classifying simple color or texture information and can be easily foiled by monochromatic environments, shadows, lighting changes, etc.

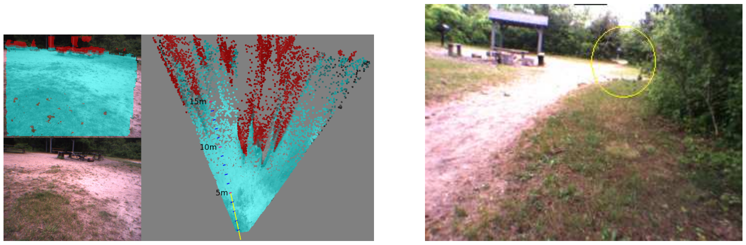


Figure 1: *Left:* Top view of a map generated from stereo (stereo is run at 320x240 resolution). The map is ”smeared out” and sparse at long range because range estimates from stereo become inaccurate above 10 to 12 meters. Traversibility maps built from this are either short range or inaccurate.
Right: Examples of human ability to discern navigational information from monocular images. The obstacles in the mid-range are obvious to a human, as is the pathway through the trees in the far-range. Note that for navigation, directions to obstacles and paths are more important than exact distances.

Building on this paradigm, our long-range obstacle detection system (LROD) overcomes the limitations of other approaches by applying three innovative techniques. First, we use a distance-normalized pyramid to train on large, context-rich windows from the image. This allows for improved path and obstacle detection (compared to just learning from color or texture). Secondly, the traversability labels from stereo are used to label not only the target window, but also *all other previously seen views of that target*. Thus LROD can train on far-away views that were taken before the stereo label was available. This process of *temporal label propagation* allows the system to learn view-invariant classifications of scenes and objects. Finally, LROD uses a ring buffer to simulate short term memory. LROD allows the robot to reliably “see” 35-40m away and opens the door to the possibility of human-level navigation.

Experiments were run on the LAGR (Learning Applied to Ground Robots) robot platform. Both the robot and the reference “baseline” software were built by Carnegie Mellon University and the National Robotics Engineering Center. In this program, in which all participants are constrained to use the given hardware, the goal is to drive from a given start to a predefined (GPS) goal position through unknown off-road terrain using only passive vision.

2 Previous Work

Considerable progress has been made over the last few years in designing autonomous off-road vehicle navigation systems. One direction of research involves mapping the environment from multiple active range sensors and stereo cameras [8, 10], and simultaneously navigating and building maps [7, 15] and classifying objects.

Estimating the traversability of the terrain of the environment constitutes an important part of the navigation problem, and solutions have been proposed by many; see [1, 3, 11, 13, 14, 16]. However, the main disadvantage of these techniques is that they assume that the characteristics of obstacles and traversable regions are fixed, and therefore they cannot easily adapt to changing environments. The classification features are hand designed based on the knowledge of properties of terrain features like 3-D shape, roughness etc. Without learning, these sytems are constrained to a limited range of predefined environments. However, the infinite variety of the possible obstacles in the real world constitues a major challenge that remains to be solved in offroad robot navigation. By contrast, the vision system presented in this paper uses online learning and is adaptable to any environment.

A number of systems that incorporate learning have also been proposed. These include ALVINN [12] by Pomerlau, MANIAC [4] by Jochem et. al, and DAVE [9] by LeCun et. al. The main disadvantage of these systems is that they are trained purely in a supervised setting. This requires large amount of labeled data which often requires a lot of human effort. In addition, these systems don’t learn a traversability mapping and don’t have a representation and mapping of the environment.

Recently Kim et. al [6] proposed an autonomous off-road navigation system that estimates the traversability of terrains in an unstructured, unknown outdoor environment. Their method is unsupervised; the robot learns traversability from its experiences as it navigates through the world.

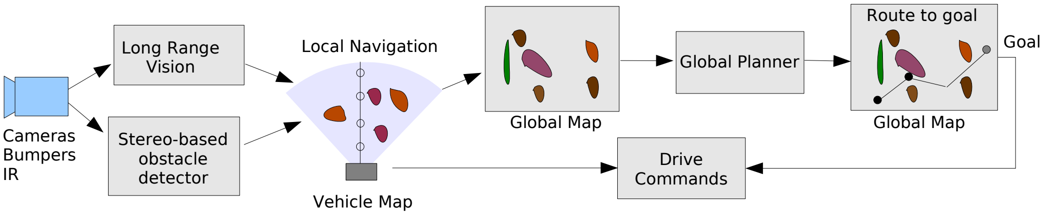


Figure 2: A flow chart of the full LAGR navigation system. The LROD and the stereo obstacle detector both populate the vehicle map, where local navigation is done. The local map gets written to the global map after every frame, where route planning is done with the global planner.

The work presented here combines unsupervised and semi-supervised online learning with the ability to generalize traversibility classification. The LROD system was developed in the LAGR program, as was the approach of Kim et. al. As compared to their approach we are able to identify the traversibility of regions that are far beyond the stereo range, using the labels generated by the stereo module and propogating them to long range, using temporal and spatial information.

3 The LAGR Vehicle: Overview of Path Planning and Local Navigation

This section gives a general overview of the LAGR navigation system. This section discusses the full navigation system developed for the LAGR robot. Although reference "baseline" software was provided, none was used in our system. Our LAGR system consists of 4 major components (see figure 2).

Vehicle Map. The vehicle map is a local map in polar coordinates that is fixed relative to the robot position. It is 100° wide and has a 40m radius, including the outputs from the long-range vision system described in this paper. It stores cost and confidence data which is delivered by the various obstacle detectors. The map has no memory.

Local Navigation. The local navigation is based on the vehicle map. It determines a set of candidate way points based on cost, confidence, and steering requirements. The candidate way point is picked which lets the vehicle progress toward the goal. Driving commands are issued based on this choice. The local navigation is robust against positioning errors.

Global Map. The global map is a Cartesian grid map into which cost and confidence information from the vehicle map is copied after each processed frame. The global map is the system's "memory".

Global Planner. The global planner finds a route to the goal in the global map, starting with candidate points proposed by the local navigation module. The algorithm is a modified A-Star algorithm which operates on rays rather than grid cells. It is much faster than A-Star and generates far fewer waypoints.

4 Long-Range Vision from Distance-Normalized Monocular Images

4.1 Motivation and Overview

Humans can easily locate pathways from monocular views, e.g. trails in a forest, holes in a row of bushes. In this section, we present LROD, a vision system that uses online learning to provide the same capability to a mobile robot. One approach for autonomous learning in robots is to use the short-range output of reliable modules (such as stereo) to provide labels for a trainable module (such as a long-range obstacle detector). LROD builds on this paradigm in three important ways. First, LROD does horizon leveling and disance normalization in the image space, producing a multiresolution pyramid of sub-images. This transformation is essential for generalizing the classifier to long-range views. Second, the system propagates labels temporally using spatial concurrences in a quad-tree. This allows us to directly train on previously seen views that are out of stereo range. Last, LROD uses a ring buffer to hold a balanced set of traversable and nontraversable training samples. This creates a sort of short-term memory.

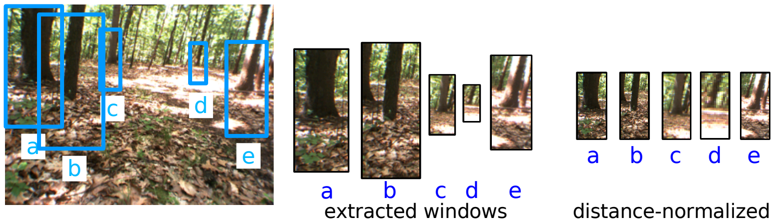


Figure 3: This figure uses isolated objects in an image to demonstrate the problem of distance scaling. *left*: The trees in the image vary widely in size because of different distances from the camera. This makes near-to-far learning extremely difficult. *center and right:* If windows are cropped from the image and resized such that the cropping window is proportional to their distance from the camera, then a classifier can train on uniformly sized and positioned objects.

The operations in LROD can be summarized here and details are discussed in the following sections.

1. All points in the current frame are inserted into a quad-tree according to their XYZ coordinates. The XYZ coordinates are determined by mapping pixels in image space to world space, and can be computed if the parameters of the ground plane are known. This is discussed in section 4.2.
2. The stereo module labels the traversability of visible points up to 15 meters.
3. Each point that was given a stereo label in the previous step is now used as a *query point* for label propagation. The XYZ quad-tree is queried to find all previously seen points (and their associated windows) that are within a radius r of the queried location. These points are labeled with the stereo label of the *query point*.
4. All stereo-labeled query points and points returned by the quad-tree are added to a ring buffer.
5. A discriminative classifier trains on the samples associated with points in the ring buffer to map windows in the image pyramid to the traversibility at the corresponding location on the ground.
6. The trained module classifies all windows in the pyramid, at a range of 1 to 35 meters (far beyond stereo range).

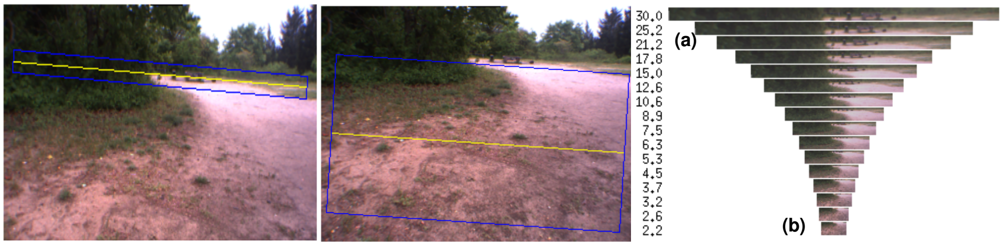
4.2 Horizon Leveling and Distance Normalization

Recent applications of near-to-far bootstrap learning in robots have trained classifiers on small image patches, thus limiting the learning to color and texture discrimination. However, it is preferable to use larger windows from the image, thus providing a richer context for more accurate training and classification. Larger windows can encompass, and learn, characteristic vertical patterns such as *road-trees-sky* or *grass-rocks-sky* or *dirt-foot-of-object-sky*. Recognizing the feet of objects is critical for obstacle detection, and the task is easier with large, context-rich windows.

There is an inherent difficulty with training on large windows instead of color/texture patches. In image space, the size of obstacles, paths, etc. varies greatly with distance, making generalization from near-range to far-range unlikely. Our approach deals with this problem by building a *distance-invariant pyramid of images at multiple scales*, such that the appearance in the image at scale X of an object sitting on the ground X meters away is identical to the appearance in the image of scale Y of the same object when sitting on the ground Y meters away. This also makes the feet of objects appear at a consistent position in a given image, allowing for easier and more robust learning. See figure 3 for further explanation of this strategy.

In order to build a pyramid of horizon-leveled sub-images, the ground plane in front of the robot must first be identified by performing a robust fit of the point cloud obtained through stereo. A Hough transform is used to produce an initial estimate of the plane parameters. Then, a least-squares fit refinement is performed using the points that are within a certain distance of the initial plane estimate.

To build the image pyramid, differently sized sub-images are cropped from the original RGB frame such that each is centered around an imaginary *footline* on the ground. Each footline is a predetermined distance (using a geometric progression) from the robot's camera. For [*row, column, disparity, offset*] plane parameters $P = [p_0, p_1, p_2, p_3]$ and desired disparity d, the image coordinates (x_0, y_0, x_1, y_1) of the footline can be directly calculated.

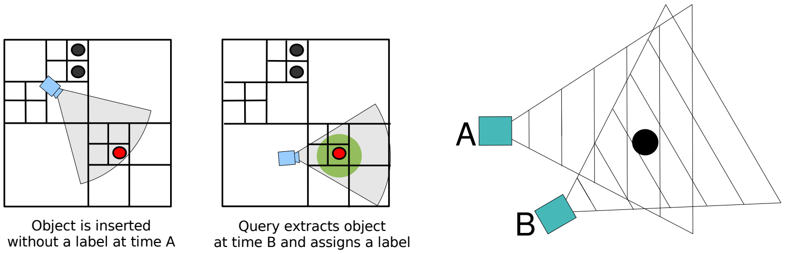


(a). sub-image extracted from far range. (21.2 m from robot).

(b). sub-image extracted at close range. (2.2 m from robot).

(c). the pyramid, with rows (a) and (b) corresponding to sub-images at left.

Figure 4: Sub-images are extracted around imaginary lines on the ground that are computed using the ground plane fit. *(a)* Extraction around a footline that is 21m away from the vehicle. *(b)* Extraction around a footline that is 1.1m away from the robot. The extracted area is large, because it is scaled to make it consistent with the scale of the other bands. *(c)* All the sub-images are reduced down to 12 pixels high. The size-normalized image pyramid is shown, containing 20 bands (36 are actually used in the system).



Object is inserted without a label at time A

Query extracts object at time B and assigns a label

Figure 5: Temporal label propagation. At position/time **A**, the robot can see the black object, but the object is out of stereo range, so it cannot be labeled. It is inserted into the quad-tree. At position/time **B**, the object is seen from a different, closer view and a stereo label is obtained. Now the view of the object from position **A** can be extracted from the quad-tree and trained on using the label from position **B**.

After cropping a sub-image from around its footline, the sub-image is then subsampled to make it a uniform height (12 pixels), resulting in image bands in which the apperance of an object on the ground is independent of its distance from the camera (see figure 4). These uniform-height, variable-width bands form a distance-normalized image pyramid whose 36 scales are separated by a factor of $2^{\frac{1}{6}}$.

4.3 Temporal Label Propagation

We expand the number of labeled training samples per frame by propagating labels backward in time. This is done using a quad-tree that indexes XYZ locations in the world surrounding the robot. The quad-tree is a very efficient data structure for storing spatial information, and concurrent views can be inserted and queried in $O(lgn)$ time. Given a labeled window and its corresponding world XYZ location, we can query the quad-tree to retrieve all previously stored views of the same location (see figure 5). Label propagation on a graph is a variant of semi-supervised learning that exploits knowledge about the robot's position and heading to expand the training set on every frame.

The stereo-labeled points and the query-extracted points are stored in 2 large ring buffers, one for traversable points, and one for non-traversable points. On each frame, the classifier trains on all the data in both ring buffers. The ring buffer acts like short-term memory, since samples from previous frames persist in the buffer until replaced. The ring buffer also balances the training, ensuring that the classifier always trains on a constant ratio of traversable/non-traversable points.

4.4 Training Architecture and Loss Function

The long-range OD goes through a labeling, training, and classification cycle on every frame. First, each overlapping 12x3 pixel RGB window from the right camera is assigned a traversibility label

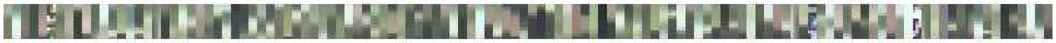


Figure 6: The centers of the 120 radial basis functions, trained using unsupervised K-means learning. The width of each RBF is variable, and is determined by the variance of the associated K-means cluster. The RBFs are used to generate feature vectors from each window of each band of the image pyramid

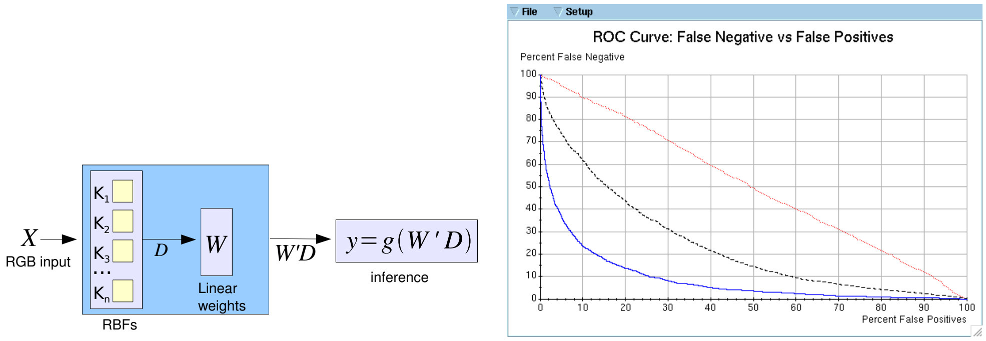


Figure 7: *left:* The online learning architecture for the LROD system.X is the input window, D is the feature vector calculated from the RBF layer, and W is a weight vector. The function g is a logistic function for inference on y
right: ROC curves comparing classifier labels vs. stereo labels. *dotted* The LROD was initialized with random weights, and the online learning was turned off. *dashed:* The LROD was initialized with default trained weights, but online learning was turned off. *solid:* The full system: trained default weights and online learning.

(ground or obstacle) if it is within stereo range ($<$ 12 meters) and if stereo data is available. Then feature vectors are computed for overlapping windows in the pyramid using a set of 120 radial basis functions. A feature vector is constructed with Euclidean distances between a 12x3 RGB window and the 120 fixed rbf centers. For an input window X and a set of n radial basis centers $\mathbf{K} = [K_1 ... K_n]$, the feature vector D is $D = [\exp(-\beta_1 ||X - K_1||^2)$ ... $\exp(-\beta_n ||X - K_n||^2)$ where β_i is the variance of rbf center K_i. The radial basis function centers $\mathbf{K}$ are trained in advance with K-means unsupervised learning, using a wide spectrum of logfiles from different environments (see figure 6). The width of RBF i, β_i, is determined by the inverse variance of the points that clustered to it during K-means learning. A logistic regression on the feature vectors is trained using the labels provided by stereo. The resulting classifier is then applied to all feature vectors in the pyramid, including those with stereo labels. The training architecture is shown in Figure 7.

The classifier is a logistic regression trained with stochastic gradient descent to produce binary labels (0 for traversable, 1 for non-traversable). Weight decay towards a previously learned set of default weights provides regularization. The loss function is cross-entropy loss. For weights W, feature vector D, label y, and logistic function $g(z) = \frac{1}{1+e^z}$, the loss function and gradient of the loss function are

$$\mathcal{L} = -\sum_{i=1}^{n} \log(g(y \cdot W'D)) \qquad \frac{\partial L}{\partial W} = -(y - g(W'D)) \cdot D$$

Learning is done by adjusting the weights with stochastic gradient descent. Although the loss function is convex and optimal weights could be found exactly, using stochastic gradient descent gives the system a natural and effective inertia that is necessary for generalizing well over successive frames and environments.

After training on the feature vectors from labeled windows, all the windows are classified. Inference on y is simple and fast: $y = sign(g(W'D))$ where $g(z) = \frac{1}{1+e^z}$ as before. Although this classifier is small and relatively simple, it is appropriate for the learning task, and it is fast enough to meet the tight computational requirements of an online, real time robot.

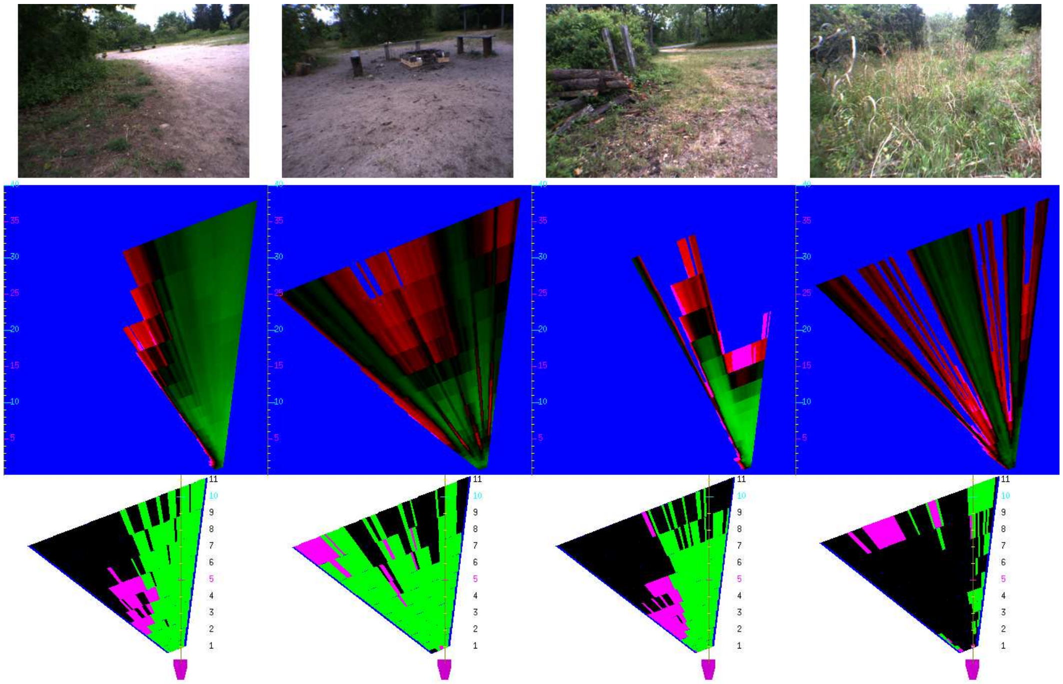


Figure 8: Examples of performance of the long-range obstacle detector. Each example shows the input image (top), the map produced by the long-range OD (middle), and the stereo labels (lower). Note that the stereo (lower map) is shown at a range of 0-11m, whereas the long-range OD map is shown from 0-40 meters.

5 Results

The robot drives smoothly and quickly under the full navigation system described in section 3. It typically gets to a goal 2 to 4 times faster than the baseline system.

To specifically assess the accuracy of the LROD system, the error of the classifier was measured against stereo labels. If the classifier was initialized with random weights and no online learning was used, then the error rate was, predictably, 52.23%. If the classifier was initialized with a set of default parameters (average learned weights over many logfiles) but with no online learning, then the classifier had an error of 32.06%. If the full system was used, i.e., initialized with default weights and online learning on, then the average error was 15.14%. ROC curves were computed for each test error rate (see figure 7).

Figure 8 shows examples of the maps generated by the long-range obstacle detector. The LROD not only yields surprisingly accurate traversibility information at distance up to 30 meters (far beyond stereo range), but also produces smooth, dense traversability maps for areas that are within stereo range. The stereo-derived maps often have noisy spots or holes - disastrous to a path planner - but the adaptive long-range OD produces maps that are smooth and accurate, without holes or noise.

The LROD module, when integrated with the whole driving system and tested in offroad environments, generally produced clear and accurate global maps (see figure 9) that could be successfully navigated by the robot.

6 Summary and Further Work

We demonstrated an autonomous, on-line learning method for obstacle detection beyond the range of stereo. At each frame, a pyramid of image bands is built in which the size of objects on the ground is independent of their distance from the camera. Windows on locations beyond stereo range are stored in a spatially-indexed quadtree for future reference. Windows within stereo range are assigned labels, together with windows at the same spatial location stored in the quadtree during previous frames. A simple on-line logistic regression classifier fed with pre-trained RBF features

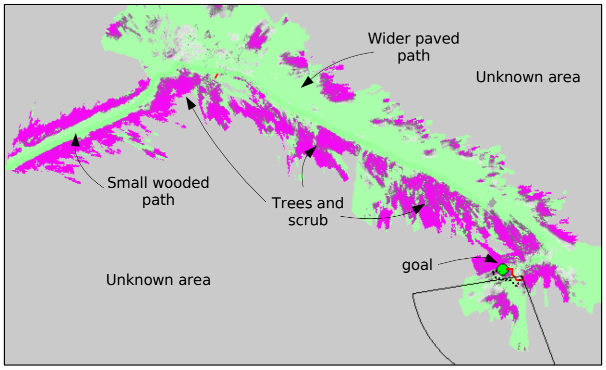


Figure 9: This is a global map produces by the robot while navigating toward a goal (green circle). The robot sees and follows the path and resists the shorter yet non-traversable direct path through the woods.

is trained on a ring buffer of such windows. The system requires no human intervention, and can produce accurate traversability maps up at ranges of up to 40 meters.

The method (minus the quadtree-based label propagation), was implemented as part of a complete vision-based driving system for an off-road robot. The system navigate through unknown complex terrains to reach specified GPS coordinates about 2 to 4 times faster than the baseline system provided with the robot platform (see supplementary video).

Future work will include the use of more sophisticated learning architectures such as multilayer convolutional networks [9].

References

[1] P. Bellutta, R. Manduchi, L. Matthies, K. Owens, and A. Rankin. Terrain perception for demo iii. 2000. 2

[2] S. B. Goldberg, M. Maimone, and L. Matthies. Stereo vision and robot navigation software for planetary exploration. March 2002. 1

[3] A. Huertas, L. Matthies, and A. Rankin. Stereo-based tree traversability analysis for autonomous off-road navigation. 2005. 2

[4] T. M. Jochem, D. A. Pomerleau, and C. E. Thorpe. Vision-based neural network road and intersection detection and traversal. *Intelligent Robot and Systems (IROS)*, 03:344–349, 1995. 2

[5] A. Kelly and A. Stentz. Stereo vision enhancements for low-cost outdoor autonomous vehicles. *Int'l Conf. on Robotics and Automation, Workshop WS-7, Navigation of Outdoor Autonomous Vehicles*, May 1998. 1

[6] D. Kim, J. Sun, S. M. Oh, J. M. Rehg, and B. A. F. Traversibility classification using unsupervised on-lne visual learning for outdoor robot navigation. May 2006. 2

[7] D. J. Kriegman, E. Triendl, and T. O. Binford. Stereo vision and navigation in buildings for mobile robots. *IEEE Trans. Robotics and Automation*, 5(6):792–803, 1989. 1, 2

[8] E. Krotkov, , and M. Hebert. Mapping and positioning for a prototype lunar rover. pages 2913–2919, May 1995. 2

[9] Y. LeCun, U. Muller, J. Ben, E. Cosatto, and B. Flepp. Off-road obstacle avoidance through end-to-end learning. In *Advances in Neural Information Processing Systems (NIPS 2005)*. MIT Press, 2005. 2, 8

[10] L. Matthies, E. Gat, R. Harrison, V. R. Wilcox, B and, , and T. Litwin. Mars microrover navigation: Performance evaluation and enhancement. pages 433–440, August 1995. 2

[11] R. Pagnot and P. Grandjea. Fast cross country navigation on fair terrains. pages 2593 –2598, 1995. 2

[12] D. A. Pomerlau. Knowledge based training of artificial neural networks for autonomous driving. *J. Connell and S. Mahadevan, editors, Robot Learning. Kluwer Academic Publishing*, 1993. 2

[13] A. Rieder, B. Southall, G. Salgian, R. Mandelbaum, H. Herman, P. Render, and T. Stentz. Stereo perception on an off road vehicle. 2

[14] S. Singh, R. Simmons, T. Smith, A. Stentz, V. Verma, A. Yahja, and K. Schwehr. Recent progress in local and global traversability for planetary rovers. pages 1194–1200, 2000. 2

[15] S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, pages 21–71, February 1998. 2

[16] N. Vandapel, D. F. Huber, A. Kapuria, and M. Hebert. Natural terrain classification using 3d ladar data. 2004. 2