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

Raia Hadsell*, Pierre Sermanet*†, Ayse Naz Erkan*, Jan Ben†,
Jefferson Han*, Beat Flepp†, Urs Muller†, and Yann LeCun*
* The Courant Institute of Mathematical Sciences, New York University, New York, NY, USA
† Net-Scale Technologies, Morganville, NJ, USA
email: {raia,naz,jhan,yann}@cs.nyu.edu, {psermanet,jben,flepp,urs}@net-scale.com

***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 predict the traversability of the entire scene. A distance-normalized image pyramid makes it possible to efficiently train 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, thus training the classifier to be 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 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.**

I. Introduction

The method of choice for vision-based driving in offroad 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 [7, 9, 3]. 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 left). Human visual performance is not due to better stereo perception; in fact, humans are excellent at locating pathways and obstacles in monocular images (see Figure 1 right).

We present a learning-based solution to the problem of long-range obstacle and path detection, by designing an approach involving *near-to-far learning*. It is called near-to-far learning because it learns traversability labels from stereo-labeled image patches in the near-range, then classifies 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.

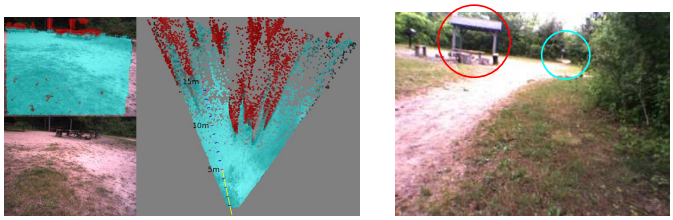


Fig. 1. *Left:* Top view of a map generated from stereo (stereo is run at 320x240 resolution). The map is "smeared out" and sparse at long range because range estimates from stereo become inaccurate above 10 to 12 meters.
Right: Examples of human ability to understand monocular images. The obstacles in the mid-range are obvious to a human, as is the distant pathway through the trees. Note that for navigation, *directions* to obstacles and paths are more important than exact distances.

In order to be effective, the long-range obstacle detection system must overcome some obstacles. A normalization scheme must be used because of the problem of relative sizes of objects in the near and far ranges. 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 the stereo module may be sparse or noisy, so we maximize their usefulness by using them to label not only the target window, but also *all other previously seen views of that target*. Thus the classifier can train on far-away views that were taken before the stereo label was available. This process of *spatial label propagation* allows the system to learn view-invariant classifications of scenes and objects. Finally, we use a ring buffer to simulate short term memory. The system 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, offroad terrain using only passive vision.

II. Previous Work

Considerable progress has been made over the last few years in designing autonomous offroad vehicle navigation systems. One direction of research involves mapping the environment from multiple active range sensors and stereo cameras [10, 14], and simultaneously navigating and building maps [9, 21] and classifying objects.

Estimating the traversability of an environment constitutes an important part of the navigation problem, and solutions have been proposed by many; see [1, 5, 15, 17, 18, 23]. 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 systems are constrained to a limited range of predefined environments. By contrast, the vision system presented in this paper uses online learning and adapts quickly to new environments.

A number of systems that incorporate learning have also been proposed. These include ALVINN [16] by Pomerlau, MANIAC [6] by Jochem et al., and DAVE [11] by LeCun et al. Many other systems have been proposed that rely on supervised classification [13, 4]. These systems are trained offline using hand-labeled data, with two major disadvantages: labeling requires a lot of human effort and offline training limits the scope of the robot's expertise to environments seen during training.

More recently, *self-supervised* systems have been developed that reduce or eliminate the need for hand-labeled training data, thus gaining flexibility in unknown environments. With self-supervision, a reliable module that determines traversability can provide labels for inputs to another classifier. This is known as *near-to-far* learning. Using this paradigm, a classifier with broad scope and range can be trained online using data from the reliable sensor (such as ladar or stereo). Not only is the burden of hand-labeling data relieved, but the system can robustly adapt to changing environments. Many systems have successfully employed near-to-far learning in simple ways, primarily by identifying ground patches or pixels, building simple color histograms, and then clustering the entire input image.

The near-to-far strategy has been used successfully for autonomous vehicles that must follow a road. In this task, the road appearance has limited variability, so simple color/texture based classifiers can often identify road surface well beyond sensor range. Using this basic strategy, self-supervised learning helped win the 2005 DARPA Grand Challenge: the winning approach used a simple probabilistic model to identify road surface based on color histograms extracted immediately ahead of the vehicle as it drives [2]. In a slightly more complicated approach by Thrun et al., previous views of the road surface are computed using reverse optical flow, then road appearance templates are learned for several target distances [12].

Several other approaches have followed the self-supervised, near-to-far learning strategy. Stavens and Thrun used self-supervision to train a terrain roughness predictor [20]. An online probabilistic model was trained on satellite imagery and ladar sensor data for the Spinner vehicle's navigation system [19]. Similarly, online self-supervised learning was used to train a ladar-based navigation system to predict the location of a load-bearing surface in the presence of vegetation [24]. A system that trains a pixel-level classifier using stereo-derived traversability labels is presented by Ulrich [22]. Recently Kim et al. [8] proposed an autonomous offroad navigation system that estimates traversability in an unstructured, unknown outdoor environment.

The work presented here uses self-supervised online learning to generalize traversability classification. Unlike other methods, our method relies solely on visual data and is efficient enough to re-train and re-classify each frame in realtime (roughly 4-5 frames/second). The system requires no human labeling or supervision.

III. The LAGR Vehicle: Overview of Path Planning and Local Navigation

This section gives an overview of 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. It stores cost and confidence data which is delivered by the different obstacle detectors.
- **Local Navigation.** The local navigation is based on the vehicle map. It determines a set of candidate waypoints based on cost, confidence, and steering requirements. The candidate waypoint is picked which lets the vehicle progress toward the goal. Driving commands are issued based on this choice.
- **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.

IV. Long-Range Vision from Distance-Normalized Monocular Images

A. 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 a vision system that uses online learning to provide the same capability to a mobile robot. Our approach, using *self-supervised* learning, is to use the short-range output of

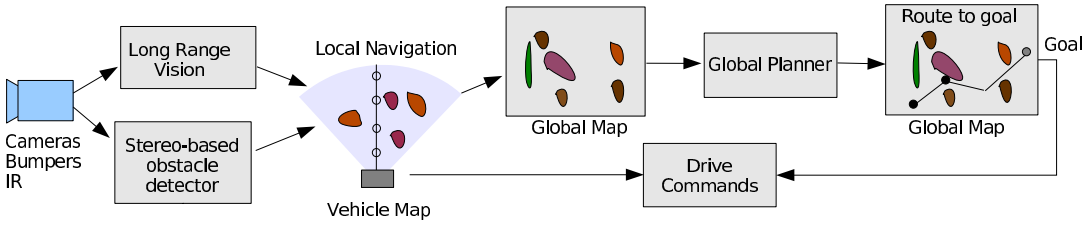


Fig. 2. A flow chart of the full navigation system. The long-range obstacle detector and the stereo obstacle detector both populate the vehicle map, where local navigation is done. The local map gets written to the global map after every frame, where route planning is done with the global planner.

a reliable module (stereo) to provide labels for a trainable module (a window-based classifier). There are three key components to the approach. First, we do horizon leveling and distance normalization in the image space, producing a multi-resolution 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, we use a ring buffer to hold a balanced set of traversable and non-traversable training samples.

The sequence of operations are 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 from image space to world space, and can be computed if the parameters of the ground plane are known. This is discussed in section IV-B.
2) The stereo module labels the traversability of visible points up to 12 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 quad-tree is queried to find all previously seen points 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 in the ring buffer, which are labeled with -1 (ground) or 1 (obstacle).
6) The trained module classifies all windows in the pyramid, at a range of 1 to 35 meters.

B. Horizon Leveling and Distance Normalization

Recent vision-based autonomous navigation systems have trained classifiers using small image patches or mere pixel information, thus limiting the learning to color and texture discrimination. However, it is beneficial to use larger windows from the image, thus providing a richer context for more accurate training and classification. Recognizing the feet of objects is critical for obstacle detection, and the task is easier with larger windows.

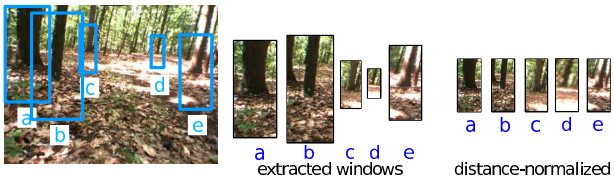


Fig. 3. This figure demonstrates the problem of distance scaling. *left*: The trees in the image vary widely in size because of different distances from the camera. This makes near-to-far learning extremely difficult. *center and right:* If windows are cropped from the image and resized such that the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

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 of an object sitting on the ground X meters away is identical to the appearance of the same object when sitting on the ground Y meters away (see Figure 3). This also makes the feet of objects appear at a consistent position in a given image, allowing for easier and more robust learning.

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 threshold 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.

After cropping a sub-image from around its footline, the sub-image is then subsampled to make it a uniform height (20 pixels), resulting in image bands in which the appearance of an object on the ground is independent of its distance from the

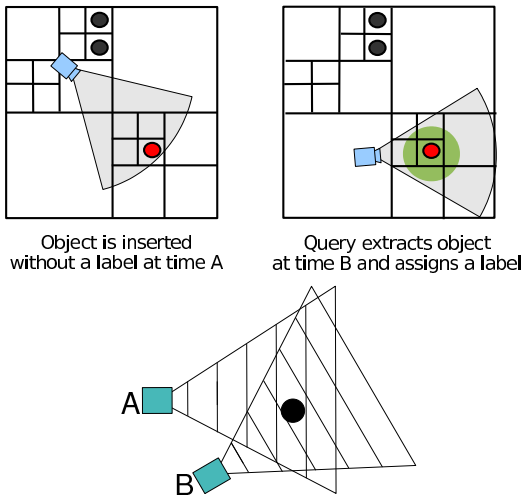


Fig. 5. Spatial label propagation. At position/time **A**, the robot can see the black location, but it is out of stereo range, so it cannot be labeled. It is inserted into the quad-tree. At position/time **B**, the black area is seen from a different, closer view and a stereo label is obtained. Now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

Fig. 6. Multiple views of a single object. The same label (non-traversable, in this case) is propagated to each instance.

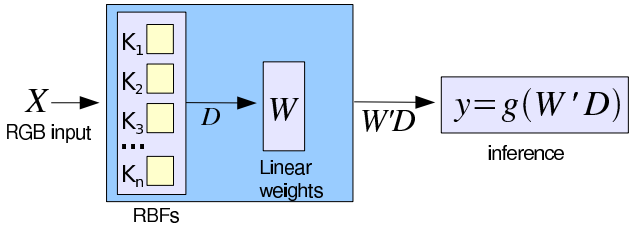


Fig. 7. The online learning architecture for the long-range vision system.X is the input window, D is the feature vector calculated from the RBF layer, and W is a weight vector. The function g is a logistic function for inference on y

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 $2^{\frac{1}{6}}$.

C. Spatial 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 and 6). 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.

D. Training Architecture and Loss Function

The long-range obstacle detector goes through a labeling, training, and classification cycle on every frame. First, each overlapping RGB window from the right camera is assigned a traversability label (ground or obstacle) if it is within stereo range (< 12 meters) and if stereo data is available. Then feature vectors are computed for the windows in the pyramid using a small convolutional network that is trained offline. A set of 120 radial basis functions was also implemented as a feature extractor, but tests showed that the trained convolutional network produced more discriminative feature vectors. Details on the feature extraction process, for the convolutional network and the RBF (radial basis function) network, are provided. The classification error rates, after offline training of the different feature extractors, is given in Table I.

The RBF feature vectors are constructed using 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 $D = [\exp(-\beta_1 ||X - K_1||^2) \ldots \exp(-\beta_n ||X - K_n||^2)$ 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.

Several convolutional networks were compared. The one chosen is a three-layer network with a 15x15 pixel field of view and an output vector of 120. The first layer (C0) has 16 6x6 convolutional kernels; the second layer (S0) is a 2x2 subsampling layer, and the third layer (C1) has 64 5x5 convolutional kernels. Although this network did not achieve the lowest offline error rate, it had the best speed/performance balance out of all the tested networks.

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

(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.

Fig. 4. Sub-images are extracted according to imaginary lines on the ground (computed using the estimated ground plane). *(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 size of the other bands. *(c)* All the sub-images are subsampled to 20 pixels high.

Test	Baseline (sec)	Our system(sec)	% Improved
11.1	327	310	5.48
11.2	275	175	57.14
11.3	300	149	101.34
12.1	325	135	140.74
12.2	334	153	118.3
12.3	470	130	261.54
12.4	318	155	105.16

TABLE II

GOVERNMENT TESTING RESULTS. A DARPA TEAM TESTED OUR SOFTWARE AGAINST THE BASELINE SYSTEM AT PERIODIC INTERVALS, AT UNKNOWN LOCATIONS WITH UNKNOWN ENVIRONMENTS AND OBSTACLES. THE PERFORMANCE OF EACH SYSTEM WAS MEASURED AS THE TIME TAKEN TO REACH THE GOAL. OUR SYSTEM TYPICALLY REACHED THE GOAL 2 TO 4 TIMES FASTER THAN THE BASELINE SYSTEM

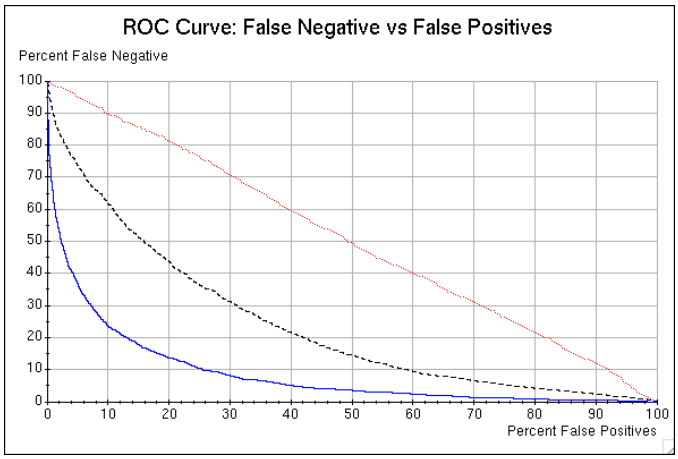


Fig. 8. ROC curves comparing classifier labels vs. stereo labels. *dotted/blue* The classifier was initialized with random weights, and the online learning was turned off. *dashed/black:* The classifier was initialized with default trained weights, but online learning was turned off. *solid/red:* The full system: trained default weights and online learning.

and gradient of the loss function are

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

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.

V. 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 (see Table II). The long-range vision module is efficient; it runs at 3-4 frames/second.

To specifically assess the accuracy of the long-range classifier, 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 8).

Figure 9 shows examples of the maps generated by the long-range obstacle detector. It not only yields surprisingly accurate traversability 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

Feature extractor	Input window size	% Test error (offline)	% Error (online)
RBF	24x6	45.94	23.11
RBF	6x6	47.34	22.65
RBF	12x3	48.49	21.23
CNN (3-layer)	15x15	20.71	11.61
CNN (3-layer)	20x11	20.98	12.89
CNN (3-layer)	12x9	24.65	14.56
CNN (1-layer)	24x11	16.35	11.28

TABLE I

A COMPARISON OF ERROR RATES FOR DIFFERENT FEATURE EXTRACTORS. EACH FEATURE EXTRACTOR WAS TRAINED, OFFLINE, USING THE SAME DATA. RBF = RADIAL BASIS FUNCTION FEATURE EXTRACTOR. CNN = CONVOLUTIONAL NEURAL NETWORK FEATURE EXTRACTOR.

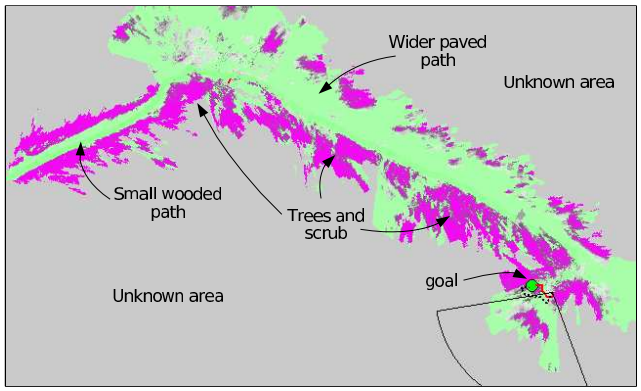


Fig. 10. This is a global map produces by the robot while navigating toward a goal. The robot sees and follows the path and resists the shorter yet non-traversable direct path through the woods.

stereo maps often have noisy spots or holes - disastrous to a path planner - but the long-range module produces maps that are smooth and accurate, without holes or noise.

The long-range module, when integrated with the whole driving system and tested in offroad environments, generally produced clear and accurate global maps (see Figure 10).

VI. SUMMARY AND FURTHER WORK

We demonstrated an autonomous, online 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 quad-tree for future reference. Windows within stereo range are assigned labels, together with windows at the same spatial location stored in the quad-tree during previous frames. A simple online logistic regression classifier fed with features extracted using a trained convolutional network is trained on a ring buffer of such windows. The system requires no human intervention, and can produce accurate traversability maps at ranges of up to 40 meters.

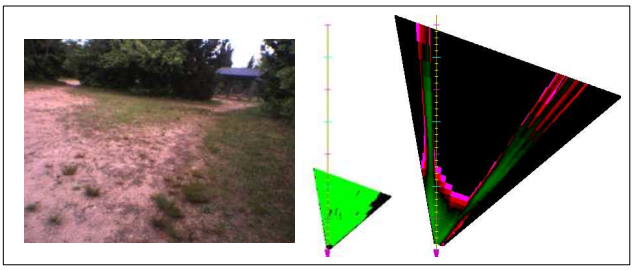
The method, not including the label propagation strategy, was implemented as part of a complete vision-based driving system for an offroad robot. The system navigates through unknown complex terrains to reach specified GPS coordinates about 2 to 4 times faster than the baseline system provided with the robot platform.

Acknowledgments

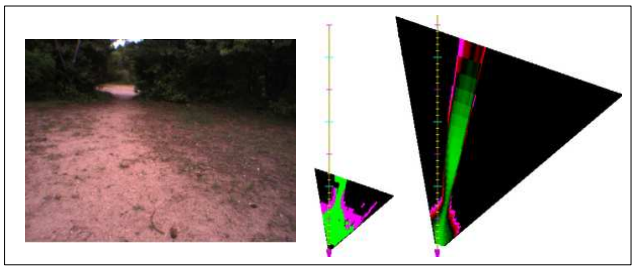
The authors wish to thank Larry Jackel, Dan D. Lee, and Martial Hébert for helpful discussions. This work was supported by DARPA under the Learning Applied to Ground Robots program.

REFERENCES

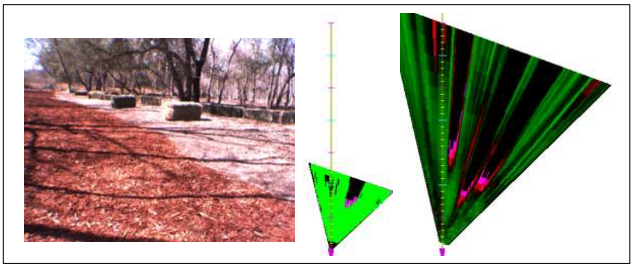
[1] P. Bellutta, R. Manduchi, L. Matthies, K. Owens, and A. Rankin. Terrain perception for demo iii. 2000.
[2] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. Bradski. Self-supervised monocular road detection in desert terrain. June 2006.
[3] S. B. Goldberg, M. Maimone, and L. Matthies. Stereo vision and robot navigation software for planetary exploration. March 2002.
[4] T. Hong, T. Chang, C. Rasmussen, and M. Shneier. Road detection and tracking for autonomous mobile robots. In *Proc. of SPIE Aeroscience Conference*, 2002.
[5] A. Huertas, L. Matthies, and A. Rankin. Stereo-based tree traversability analysis for autonomous off-road navigation. 2005.
[6] 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.
[7] 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.
[8] 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.
[9] 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.
[10] E. Krotkov, , and M. Hebert. Mapping and positioning for a prototype lunar rover. pages 2913–2919, May 1995.
[11] 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.
[12] D. Leib, A. Lookingbill, and S. Thrun. Adaptive road following using self-supervised learning and reverse optical flow. June 2005.
[13] R. Manduchi, A. Castano, A. Talukder, and L. Matthies. Obstacle detection and terrain classification for autonomous off-road navigation. *Autonomous Robot*, 18:81–102, 2003.
[14] L. Matthies, E. Gat, R. Harrison, V. R. Wilcox, B and, , and T. Litwin. Mars microrover navigation: Performance evaluation and enhancement. volume 1, pages 433–440, August 1995.
[15] R. Pagnot and P. Grandjea. Fast cross country navigation on fair terrains. pages 2593 –2598, 1995.
[16] 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.
[17] A. Rieder, B. Southall, G. Salgian, R. Mandelbaum, H. Herman, P. Render, and T. Stentz. Stereo perception on an off road vehicle. 2002.
[18] 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.
[19] B. Sofman, E. Lin, J. Bagnell, N. Vandapel, and A. Stentz. Improving robot navigation through self-supervised online learning. June 2006.



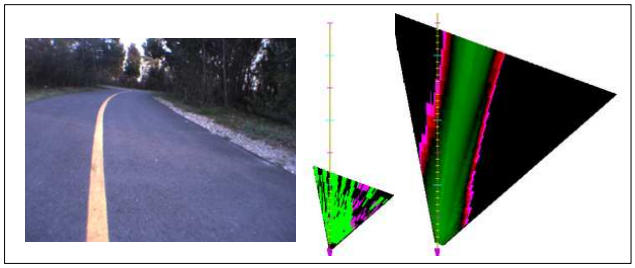
(a) 2 paths (and pavilion) detected at long range



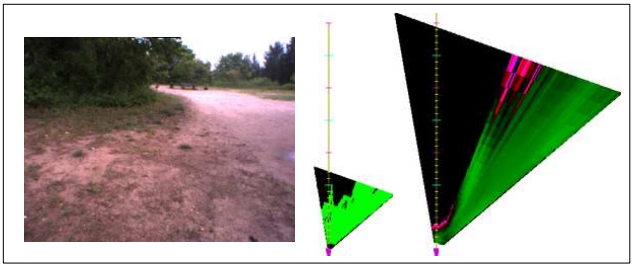
(b) Path and distant trees detected



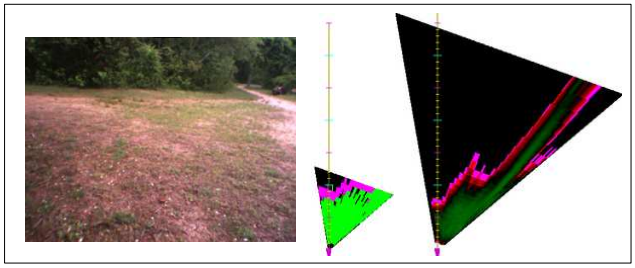
(c) Several hay-bales detected accurately



(d) Smooth road and tree barrier is seen



(e) Distant picnic tables detected



(f) Path seen at long range

Fig. 9. Examples of performance of the long-range vision system. Each example shows the input image (left), the stereo labels (middle), and the map produced by the long-range vision (right). Green indicates traversable, and red/pink indicates obstacle. Note that stereo has a range of 12 meters, whereas the long-range vision has a range of 40 meters.

[20] D. Stavens and S. Thrun. A self-supervised terrain roughness estimator for off-road autonomous driving. 2006.

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

[22] I. Ulrich and I. R. Nourbakhsh. Appearance-based obstacle detection with monocular color vision. pages 866–871, 2000.

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

[24] C. Wellington and A. Stentz. Online adaptive rough-terrain navigation in vegetation. April 2004.