Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Online, Self-Supervised Vision-Based Terrain Classification in Unstructured Environments

Peyman Moghadam, Wijerupage Sardha Wijesoma
School of Electrical and Electronic Engineering
Nanyang Technological University
Singapore, 639798

***Abstract*—Outdoor, unstructured and cross-country environments introduce several challenging problems such as highly complex scene geometry, ground cover variation, uncontrolled lighting, weather conditions and shadows for vision-based terrain classification of Unmanned Ground Vehicles (UGVs). Color stereo vision is mostly used for UGVs, but the present stereo vision technologies and processing algorithms are limited by cameras' field of view and maximum range, which causes the vehicles to get caught in cul-de-sacs that could possibly be avoided if the vehicle had access to information or could make inferences about the terrain well beyond the range of the vision system. The philosophy underlying the proposed strategy in this paper is to use the near-field stereo information associated with the terrain appearance to train a classifier to classify the far-field terrain well beyond the stereo range for each incoming image. To date, strategies based on this concept are limited to using single model construction and classification per frame. Although this single-model-per-frame approach can adapt to the changing environments concurrently, it lacks memory or history of past information. The approach described in this study is to use an online, self-supervised learning algorithm that exploits multiple frames to develop adaptive models that can classify different terrains the robot traverses. Preliminary but promising results of the paradigm proposed is presented using real data sets from the DARPA-LAGR project, which is the current gold standard for vision-based terrain classification using machine-learning techniques. This is followed by a proposal for future work on the development of robust terrain classifiers based on the proposed methodology.**

***Keywords*—online, self-supervised learning, stereo vision**

I. INTRODUCTION

Autonomous navigation system requires a vehicle to move reliably, at a desired speed from a starting location to a goal point (optionally via waypoints) avoiding obstacles. In order to realize autonomous capability, the vehicle needs a set of sensors and algorithms to estimate the vehicle's position and orientation (pose), and to predict the traversability of the terrain. In this paper, we use color stereo vision as the main sensor that can provide not only 3D perception of terrain geometry but also color and texture information.

The present stereo vision technologies and processing algorithms are limited by cameras' field of view and maximum range. As of now, the maximum range of a typical stereo vision system comprising of short base line is effectively up to about 10 - 15 meters [1]. Beyond this, the accuracy of the estimates degrades as the distance from the camera pair increases. This near-sightedness with stereo vision often causes the vehicles to get caught in cul-de-sacs that could possibly be avoided if the vehicle had access to information or could make inferences about the terrain well beyond the range of the vision system [1]. This problem can be alleviated by associating the terrain geometry regions close to the robot with visual appearance mostly in terms of color and/or texture and use this association to segment terrain and obstacles in the far-field. However, finding a global correlation between terrain geometry characteristics and appearances that can be broadly applied is impossible at present due to the complex variability of appearance on the type of the terrains and weather conditions in outdoor, unstructured and cross-country environments. Therefore, almost any hand-designed deterministic and rule-based system has proved to be futile as it is not robust to changing environments and not able to adapt to unforeseen ground cover variations. Machine learning is a promising paradigm in order to replace hand-designed deterministic vision-based terrain classification systems for UGVs. The ALVIN (Autonomous Land Vehicle In a Neural Network) [2] by Pomerleau was one of the pioneering learning based approach to robot navigation. A supervised neural network is trained using image data associated with steering angle for road following by watching a human driver's actions when driving on roads of varying properties. Manduchi et al. [3] have successfully implemented learning color distributions of terrain classes by training over a large number of images taken under widely different illumination conditions. Since the outdoor off-road environment is highly unconstrained, collecting and hand labeling of a large amount of data may be difficult, time consuming and impractical in many real world applications and finally this type of learning will be limited to the certain environment types.

Recently, Self-Supervised Learning (SSL) has been introduced which is proving to be essential for many real-time navigation systems. Self-supervised learning is an approach for designing a system that can train on the incoming data stream, adapting to unknown environments without using any hand-labeled training data. The terms Online learning or Near-to-Far learning are often used to refer to Self-Supervised Learning. Thrun et al. [4] use a combination of self-supervised learning and reverse optical flow technique for adaptive road following.

The self-supervised learning takes information from the near-range local sensors such as physical bumpers and infrared range sensors about objects and terrain types and labels input data without human assistance. When an object is registered in the near-field of local sensors, the optical flow procedure is called to trace the view of that object in the current image back to where it appeared first in the field of view of the robot in order to extract visual appearance of the object at a greater distance. This information is then used to train the Mixture of Gaussian classifier. Therefore, the robot learns the visual characteristics of objects at different distances for making early navigation decisions.

More recently, single-model-per-frame near-to-far learning was addressed by the LAGR - Learning Applied to Ground Robots. Several teams in DARPA-LAGR (API [5], JPL [6], SRI [7] and NIST [8]) exploited standard Near-to-Far learning using single-model-per-frame approach. The near-to-far learning refers to an approach of generating near-field traversability labels from each incoming pair of stereo images using only the near-field stereo information. Next, these labels are associated with the visual appearance features to train a classifier, and the model obtained is used to classify the far-field terrain well beyond the stereo range. Later, this model is discarded. Therefore, one full training and classification cycle are completed on every incoming pair of stereo images. Although this single-model-per-frame approach can adapt to the changing environments concurrently, it lacks memory or history of past information. It is quite common that the autonomous robot moving in unstructured and cross-country environments may return back to the same or similar terrain previously traversed (e.g. recurring contexts). Moreover, it is worth mentioning that at each frame the robot can only sense the world partially so it cannot be expected that it learn everything at once even if it adapts to the changing environment on-the-fly. To address the shortcomings of common single-model-per-frame near-to-far learning, Procopio et al. [9] introduce near-to-far Best-K ensemble algorithm. This ensemble learns and stores terrain models for application in future terrains. For each incoming frame, a single model is trained on the appearance of near-field stereo labels, and then this model is added to the model library. Later, when a new image is received, all models in the library are evaluated on the new image. They are ranked by their performance on near-field validation data provided by stereo. The better a model performs on validation data, the more weight it will receive. Then the best K models are selected and the outputs of these models are combined using weighted average method. They use linear SVM as their baseline technique. A drawback of their method is that it has memory loss over time for long courses. Another drawback is that the evaluation of all models of the library on a new incoming frame is computationally expensive when the robot moves for long time.

In this paper, we describe an approach using an online, self-supervised learning algorithm to develop online model to classify the different terrains the robot traverses. This online learning can train incrementally over time on the incoming data stream, adapting to unknown environments without using any hand-labeled training data while improving its performance with each new training example. It is able to learn new knowledge while keeping previously learned knowledge.

II. Online Self-Supervised Terrain Classification

In online, self-supervised learning, stereo vision reading is used to find the ground plane (traversable region) and obstacles (non-traversable region) in the near-field. The visual appearances of the near field regions are then associated with traversability (supervisory labels provided by stereo vision) as inputs for training a classifier. The trained classifier is then applied over all remaining regions of the image in order to estimate obstacles and traversable terrain well beyond the stereo range. The data flow of online, self-supervised learning used in our system is illustrated in Fig. 1. This includes pre-processing, ground plane estimation, auto labeling, feature extraction, balancing, scaling, online training and classification.

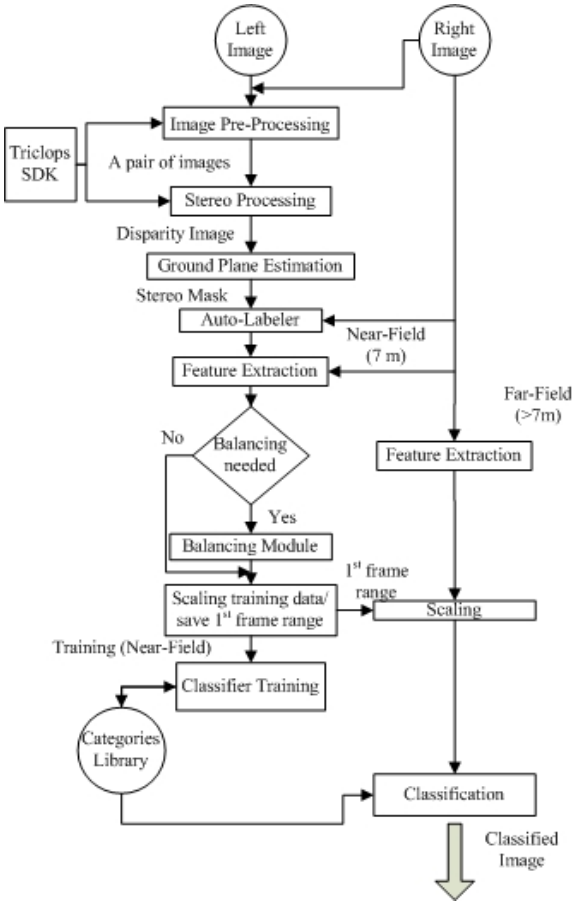

Figure 1. Data flow of Online Learning

A. Ground Plane Estimation and Obstacle Detection

The vision sensor used in our experiments is a short baseline Point Grey Bumblebee color stereo vision system. We use Triclops SDK, a Stereo Vision Software Development Kit (SDK) provided by Point Grey for stereo processing [10]. There are two main processing blocks in this library. The first one is the image pre-processing block that applies a low-pass filter, rectifies the images and performs edge detection. A

sample of rectified RGB image is shown in Fig. 2. A pair of pre-processed images is used as an input to the second block that performs stereo matching using Sum of Absolute Differences correlation method (SAD) to create a disparity map. The obstacles and ground plane estimator module applies fast RANSAC algorithm directly to the disparity image to estimate the dominant ground plane. First, the invalid pixels in current disparity map are removed (0 and 255 values). Second, 3 points are randomly selected directly from the near-field of the disparity image. A plane is fitted to these 3 points and then ranked by the number of points that are close enough to the plane (known as supporting points). A ground plane with maximum number of supporting points is selected as the dominant ground plane. Complete details of the ground plane estimation method are given in [11]. Once the ground plane is estimated, any point whose measured disparity is not within some threshold of the expected ground plane disparity range will be considered as an obstacle. This threshold is found by several tests in typical outdoor environments. Therefore, lower part of obstacles may not be detected as their disparity differences may not be significant enough. The ground plane (shown in green) and obstacles (shown in red) detected in the near-field are projected back to the original RGB image are depicted in Fig. 2.

B. *Auto-Labeling*

Once the ground plane and obstacles are estimated in the disparity image, their corresponding image pixels in the reference image (right image) are labeled. In order to reduce computation time, a 320x240 = 76800 pixels color image is quantized into patches of size 10x10. We set a 20–pixel margin for each side of an image (because invalid disparity values are often located in the side of images) and then decompose the entire image into 560 patches of size 10x10. Then, the lower half of the frame is selected as the supervision module (280 patches). The number of ground plane and obstacle labels is counted in each patch, and the patch is assigned traversability label of either obstacle or ground plane based on the number of whichever is greater. In order to fail on the safe side, if the numbers of both labels are the same the patch is marked as an obstacle. Fig. 2 shows a labeled image using patches size of 10x10. Only near-field stereo range is labeled (280 patches).

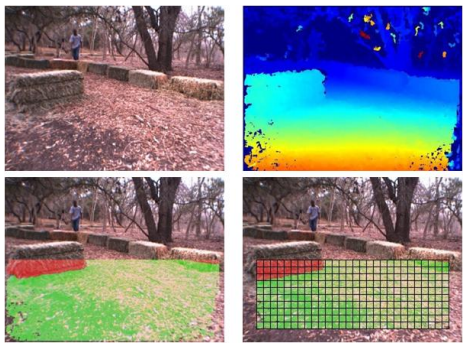

Figure 2. Original RGB image (top left), stereo disparity image (red to blue indicating decreasing disparity) (top right), ground plane (green) and obstacle (red) detection (bottom left), Traversability labels using fixed size patch traversable (green), non-traversable (red) (bottom right)

C. *Feature Extraction*

Next, a list of descriptors also known as a feature vector is extracted for each labeled patch. Here, we carry out an in-depth comparative analysis verified by experiments for common and state-of-art feature extractors. A comparison of different representations for feature vectors is shown in table I. It compares results (FP-rate and F-score) of two different data sets for one baseline classifier algorithm (SVM-Linear) using different sets of features. The most common color spaces which are good enough to describe all the ranges of colors are RGB, HSV (Hue, Saturation and value) and L*a*b (L stands for Luminance, while “a” and “b” represent the two color areas). Table I shows the baseline classifier performances based on using these 3 colors spaces with 2 commons color descriptors, mean and standard deviation and color histograms. The mean and standard deviation for three color channels produce a feature vector of depth 6. Besides, the fixed number of bins (here 5 and 8 bins) are used for 1D color histograms results respectively yielding feature depths of 15 or 24 values (three channel * five (eight) bins per channel). As it is shown in table I, the statistics information of L*a*b color space performs slightly better than the other models. Results of using only texture features are also illustrated in table I. The log Gabor filters with different orientations (3-4-8) and different scales (5-4-3) (respectively 15, 16, and 24 filter banks) are compared and tabulated. The best result is Log Gabor filter with 4 orientations and 4 scales, which covers the whole range of possible textures in outdoor environments. We measured mean and standard deviation of texture features in each patch of an image. Next, we combine the best color representation and the best texture representation (shown as red color in table I) in one feature vector with depth of 38. Before applying this feature vector to a classifier, the data has to be normalized to a specific range such as [0,1]. There are many normalization methods for scaling data such as Min-Max, Z-score, and decimal normalization. The algorithm used for this study is Min-Max normalization, which is simple and fast for real-time applications.

D. *Evaluation Measurements*

The efficiency of the learning applied to computer vision can be measured qualitatively by means of visual perception or quantitatively as the percentage of corrected classified pixel to total number of pixels. However, when a data set is unbalanced (when the number of samples in different classes varies considerably), the error rate of a classifier is not representative of the true performance of the classifier. On the other hand, human eyes simply can tell which classifiers perform qualitatively better than the others do, but the visual perception may fail to separate classifier when they perform almost the same. For quantitative evaluation measurements, there are many methods such as confusion matrix, Receiver Operating Characteristic (ROC), and the Area-Under-an-ROC-Curve (AUC). To evaluate classification techniques in this study, we use two metrics based on confusion matrix elements.

Table I. Comparison of different features using baseline classifier

Dataset		Color									Texture		
		RGB			HSV			L*a*b			Gabor15	Gabor16	Gabor24
		M/Std	Hist5	His8	M/Std	Hist5	His8	M/Std	Hist5	His8			
DS1B	FP-rate	0.11	0.09	0.10	0.12	0.12	0.12	0.07	0.21	0.19	0.10	0.10	0.11
	F-score	0.86	0.87	0.89	0.93	0.93	0.93	0.95	0.80	0.86	0.83	0.84	0.84
DS2A	FP-rate	0.23	0.27	0.25	0.26	0.30	0.29	0.21	0.53	0.54	0.14	0.14	0.15
	F-score	0.88	0.87	0.87	0.88	0.87	0.86	0.90	0.85	0.84	0.91	0.91	0.91
Avg	FP-rate	0.17	0.17	0.17	0.19	0.21	0.20	**0.14**	0.37	0.36	0.12	**0.12**	0.13
	F-score	0.87	0.87	0.88	0.90	0.9	0.89	**0.92**	0.82	0.85	0.87	**0.87**	0.87

Table II. Confusion Matrix

		Predicted class	
		T	N
True Class	T	TP	FN
	N	FP	TN

The elements of the confusion matrix are the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).Precision is the proportion of the predicted positive cases that were correct, as calculated using the equation:

$$\text{Precision} = \frac{TP}{TP+FP} \quad (1)$$

In addition, recall or true positive rate (TPR) is the proportion of positive cases that are correctly identified, and is calculated using the equation:

$$\text{Recall} = \frac{TP}{TP+FN} \quad (2)$$

Precision and Recall are better descriptors when one class is rare. To combine precision and recall in one metric we use F-score, which is a harmonic average of precision and recall. The weighted harmonic mean of recall(r) and precision (p) is:

$$\text{F-score} = \frac{(\alpha+1)r \times p}{r+\alpha \times p} \quad (3)$$

F-score commonly uses "average" of precision and recall [12] (i.e. α=1). In outdoor environments, where the cost of classifying an obstacle as ground point is too high, F-score is unable to rank the classifiers with their False Positive rate. Therefore, the FP-rate is generated separately and added to F-score as one of our evaluation performance metrics. The evaluation performance metrics are used to measure differences between the classification output and the ground truth images hand-labeled by a human.

$$\text{FP-rate} = \frac{FP}{FP+TN} \quad (4)$$

E. Classifier Training

Most existing learning algorithms don't allow incremental learning. They were designed in such a way that if a new data become available, they will tend to forget old information, re-initialize and try to train on new data. None had the ability to be able to learn new knowledge while keeping previously learned knowledge. This problem is known as the stability-plasticity dilemma. It means that how a learning system can remain plastic (adaptive) in response to new, unseen information, yet remain stable in response to irrelevant information and can filter out them. The adaptive resonance theory (ART) was initially developed by Grossberg [13] as a response to the stability-plasticity dilemma. There are several architectures in the ART family including unsupervised learning architectures to perform clustering like ART1 [14], and FuzzyART [15], supervised learning architectures like ARTMAP [16] and Fuzzy-ARTMAP (FAM) [17].

In order to address the problem of baseline single-model-per-frame approach, we use an online, self supervised learning method called Fuzzy ARTMAP (FAM). Fuzzy ARTMAP is a class of neural network architectures that perform incremental supervised learning. The network is capable to add new data items without the need of re-training. Fuzzy ARTMAP architecture includes a pair of unsupervised Fuzzy ART modules, ARTa and ARTb, linked via an inter-ART module called map field that implements a supervised learning control process between ARTa and ARTb categories and internal vigilance test that ensures autonomous system works in real-time as shown in Fig. 3. A summary of the Fuzzy ARTMAP (FAM) is given below:

1) Initialization:

The performance of ART networks depends on a learning rate β [0, 1] and a vigilance parameter ρ. In FAM, there are

two additional parameters: the baseline vigilance known as $\overline{\rho_a}$ and the vigilance parameter of map field ρ_{ab}. The baseline vigilance parameter $\overline{\rho_a}$ is set to zero initially to allow broad generalization, coarse categories, and abstract memories. On the other hand, ρ_{ab} set to one to have fine categories and detail memories for output.

2) Input Pattern:

The first network ARTa takes the stream of input data (a) and ARTb receives output classes (b) where they are the correct prediction of given inputs.

3) Category Selection:

When a pattern is applied to ARTa, a category will be chosen through the bottom-up and up-bottom ART competition. If the input vector does not match any stored category within a given vigilance parameter, then a new category is created by storing a new pattern similar to the input vector.

4) Map Field Activation:

The map field F_{ab} is activated when one of the ARTa or ARTb categories is activated. If both ARTa and ARTb are active, the map field becomes active only if ARTa predicts the correct category as ARTb. In the case of any mismatches match tracking is raised.

5) Match Tracking:

ARTMAP has the mechanism that if the prediction failed at ARTb, the vigilance parameter will be increased by minimum amount necessary to correct error. Initially, vigilance parameter ρ_a equals baseline vigilance. This feedback control mechanism is called match tracking. Match tracking enables ARTMAP to learn a prediction for rare events.

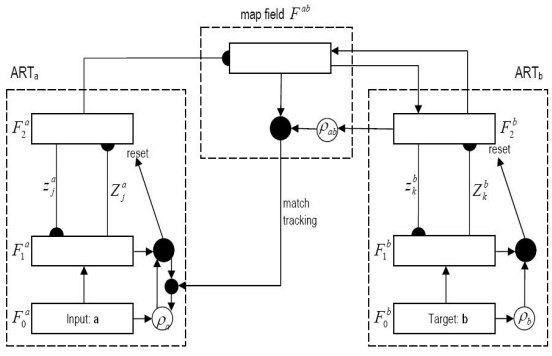

Figure 3. Fuzzy ARTMAP architecture [17]

III. Experimental Results

A. Baseline Learning

To quantify the performance of the online, self-supervised learning technique, we compare it against a baseline scenario of single-model-per-frame near-to-far learning using SVM-linear, which is simple with reasonable computational demands for real-time applications.

B. Datasets

For experimental results presented in this study, we have used image frames logged during live runs of the robot in the LAGR program, which have been made publicly available on the Internet [18]. The datasets used here were taken from three different scenarios, each with two separate lighting conditions. Each scenario has 100 image sequences. Each frame consists of a raw RGB image, raw disparity information and a hand-labeled image. Each pixel in a hand-labeled image indicates a label: 0 means ground plane, 1 means obstacle, and 2 means unknown. If it is hard for a human to assign a class label for a pixel, or it is from "don't care" regions (e.g. sky) is labeled as an unknown pixel. Representative images from datasets are shown in Fig. 4.

Figure 4. Representative images from six datasets [18]

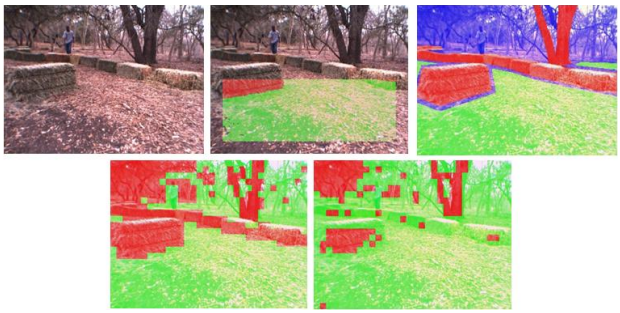

Figure 5. RGB image (top left), ground plane estimation (top middle), hand-labeled ground truth (top right), Online fuzzy ARTMAP learning (bottom left), SVM-Linear learning-baseline (bottom right)-DS1B

Fig. 5 and 6 show examples of online fuzzy ARTMAP learning against baseline SVM-Linear. The results for the 100 frames of dataset DS1B are shown in Fig. 7. The overall FP-rate of baseline SVM is "1" for the first 30 frames, though Online ARTMAP can learn new knowledge from each frame while keeping its previous models .On the whole, It has less false positive error rate and higher F-score rate compared to baseline SVM.

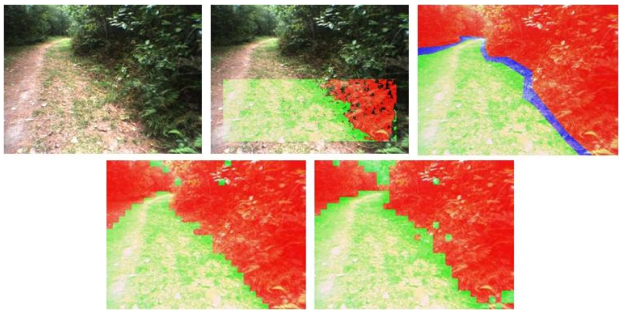

Figure 6. RGB image (top left), ground plane estimation (top middle), hand-labeled ground truth (top right), Online fuzzy ARTMAP learning (bottom left), SVM-Linear learning-baseline (bottom right)-DS3A

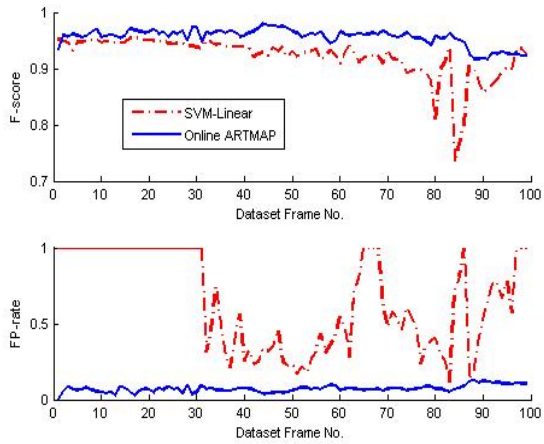

Figure 7. SVM-Linear vs. Online fuzzy ARTMAP learning, DS1B

IV. Conclusions and Future Works

To address the shortcoming of basic near-to-far learning, we have described an online, self-supervised Fuzzy ARTMAP (FAM) learning algorithm to develop an adaptive model to classify the different terrains the robot traverses. To the best of author's knowledge, this is the first study, where FAM is used for self-supervised terrain classification for UGVs. This online, self-supervised learning approach can train incrementally over time on the incoming data stream, adapting to unknown environments without using any hand-labeled training data while improving its performance with each new training example. It is able to learn new knowledge while keeping previously learned knowledge. For analysis of performance of the proposed methodology, we have used image frames logged during live runs of the LAGR project, which is the "gold standard" for learning based terrain classification.

Stereo information as supervision for our online learning may misclassify some parts of objects as ground plane (e.g. foot of obstacles – transition from obstacle to ground). Therefore, the input data to online ART learning may be noisy and this causes category proliferation. Moreover, when the robot keeps moving, the model library grows gradually. These will make it very difficult to adapt online when terrain appearance changes. Future work is aimed at algorithms with strong theoretical underpinnings to autonomously decide which categories are relevant, which need to be removed from library and how to select from existing categories and how to combine them, or to evolved or created any new category.

Acknowledgment

The authors gratefully acknowledge M. J. Procopio for providing the log data used as data sets considered in this study.

References

[1] L. Jackel, E. Krotkov, M. Perschbacher, J. Pippine, and C. Sullivan. "The DARPA LAGR program: Goals, challenges, methodology, and Phase I results". Journal of Field Robotics, 23, pp. 945–973, 2006.

[2] D. Pomerleau. "ALVINN: An autonomous land vehicle in a neural network". In Advances in Neural Information Processing Systems (NIPS), pages 305–313, 1989.

[3] R. Manduchi, A. Castano, A. Talukder, and L. Matthies. "Obstacle detection and terrain classification for autonomous off-road navigation". Autonomous Robot, 18, pages 81–102, 2005.

[4] A. Lookingbill, J. Rogers, D. Lieb, J. Curry, and S. Thrun. "Reverse Optical Flow for Self-Supervised Adaptive Autonomous Robot Navigation". Int. J. Comput. Vision 74, no.3, 287-302, 2007.

[5] M. Happold, M. Ollis, and N. Johnson. "Enhancing supervised terrain classification with predictive unsupervised learning". In Proceedings of Robotics: Science and Systems, Cambridge, USA, June 2006.

[6] M. Bajracharya, A. Howard, L. H. Matthies, B. Tang, M. Turmon. "Autonomous off-road navigation with end-to-end learning for the LAGR program." Journal of Field Robotics, 26, 3–25, 2009.

[7] K. Konolige, M. Agrawal,M.R. Blas, R.C. Bolles,B. Gerkey, J. Sol`a,A. Sundaresan, "Mapping, Navigation, and Learning for Off-Road Traversal" .Journal of Field Robotics, 26, 88-113,2009.

[8] M. Shneier, T. Chang, T. Hong, W. Shackleford, R. Bostelman, J. Albus, "Learning traversability models for autonomous mobile vehicles". Autonomous Robots, 24, 69–86, 2008.

[9] M. J. Procopio, J. Mulligan, G. Grudic. "Learning Terrain Segmentation with Classifier Ensembles for Autonomous Robot Navigation in Unstructured Environments". Journal of Field Robotics, 26, 145-175, 2009.

[10] PTGrey. Point grey research inc. http://www.ptgrey.com/,2004.

[11] P. Moghadam, W. S. Wijesoma, D. J. Feng. "Improving path planning and mapping based on stereo vision and lidar". ICARCV. 384-389, 2008.

[12] C. J. van Rijsbergen. "Information Retireval. Butterworths". London, 1979.

[13] S. Grossberg, "Adaptive Pattern Recognition and Universal Encoding II: Feedback, Expectation, Olfaction, and Illusions", Biological Cybernetics, Vol. 23, pp. 187-202, 1976.

[14] G. A. Carpenter, S. Grossberg, "A massively parallel architecture for a self-organizing neural pattern recognition machine", Computer Vision, Graphics, and Image Processing, Vol. 37, pp. 54-115,1987.

[15] G. A. Carpenter, S. Grossberg, and D.B. Rosen, "Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system", Neural Networks, 4(6), pp. 759-771, 1991.

[16] G.A. Carpenter, S. Grossberg, and J.H. Reynorlds, "ARTMAP: Supervised real-time learning and classification of non-stationary data by a self-organizing neural network", Neural Networks, Vol. 6, pp. 565-588, 1991.

[17] G.A. Carpenter, S. Grossberg, N. Markuzon, J.H. Reynorlds and D.B. Rosen, "Fuzzy ARTMAP: A Neural Network Architecture for Incremental Supervised Learning of Analog Multidimensional Maps", IEEE Transaction on Neural Networks, 3(5), pp. 698-713, 1992.

[18] M. J. Procopio. "Hand-labeled DARPA LAGR data sets". Available at http://ml.cs.colorado.edu/~procopio/labeledlagrdata/, 2007.