2010 11th Int. Conf. Control, Automation, Robotics and Vision
Singapore, 7-10th December 2010

Towards A Fully-Autonomous Vision-based Vehicle Navigation System in Outdoor Environments

Peyman Moghadam, Wijerupage Sardha Wijesoma, Moratuwage M.D.P
School of Electrical and Electronic Engineering
Nanyang Technological University
Singapore, 639798

***Abstract*— Colour Stereo visions are the primary perception system of the most Unmanned Ground Vehicles (UGVs), which can provide not only 3D perception of the terrain but also its colour and texture information. The downside with present stereo vision technologies and processing algorithms is that they are limited by the cameras' field of view and maximum range, which causes the vehicles to get caught in cul-de-sacs. The philosophy underlying the proposed framework in this paper is to use the near-field stereo vision 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. We propose an online, self-supervised learning method to learn far-field terrain traversability with the ability to adapt to unknown environments without using hand-labelled training data. The method described in this paper enhances current near-to-far learning techniques by automating the task of selecting which learning strategy to be used from among several strategies based on the nature of the incoming real-time input training data. Promising results obtained using real datasets from the DARPA-LAGR program is presented and the performance is evaluated using hand-labelled ground truth.**

***Keywords*— near-to-far learning, stereo vision, terrain classification**

I. INTRODUCTION

Autonomous navigation in outdoor, unstructured and cross-country environments introduces several challenging problems such as highly complex scene geometry, ground cover variation, uncontrolled lighting, weather conditions and shadows. Most of the autonomous navigation systems typically use range finding sensors (a stereo vision system or LIDAR) to sense their environment and predict the traversability of the terrain essentially based on geometry [1]. Perception based on stereo vision can provide the vehicle with sufficient information to build a three-dimensional geometric model of the environment enhanced with colour and texture information. However, the effective maximum range of typical off the shelf stereo vision systems is limited up to about 10-15 meters due to their short baseline (10-12cm). In order to drive at higher speeds while being able to avoid obstacles the path planning system needs information of traversability well beyond the range of a typical stereo vision system [2]. This problem can be solved by associating the geometry of terrain segments close to the vehicle (near-field) with their visual appearance attributes (e.g. colour and/or texture) and then, using such information to segment terrain and obstacles in the far-field of the camera images. However, finding a global correlation between terrain geometry characteristics and appearances that can be broadly applied is formidable due to the complex variability of appearance with terrain geometry, lighting, shadows, and weather conditions, especially in outdoor environments. Therefore, most hand-designed deterministic and rule-based system has proved to be ineffective as they are not robust to changing environments due to their inability to adapt to unforeseen ground cover variations. To address this problem, one promising approach is to use Machine-Learning techniques to replace hand-designed deterministic vision-based terrain classification systems for UGVs.

The term "self-supervised near-to-far learning" is used to refer to the Machine-Learning strategy that involves using near-field information for classifying the far-field image and is exploited by several teams in the DARPA-LAGR program including API [3], JPL [4], SRI [5] and NIST [6]. More specifically, the self-supervised near-to-far learning is an approach of generating near-field traversability labels from each incoming pair of stereo images using only the near-field stereo information. These labels are associated with their 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 is completed on every incoming pair of stereo images [7]. Although, this approach can adapt to the changing environments, it needs both negative (non-traversable) and positive (traversable) examples to train a conventional two-class classifier. However, in practical situations like outdoor terrain classification, the vehicle may not encounter any negative examples (i.e., non-traversable terrain) during some parts of its course. Bajracharya et al. [4] address this problem by using multiple one-class SVMs, although, this approach has not been extensively tested.

Moreover, since near-to-far learning approach autonomously associates the visual appearances of the near-field regions with traversability (supervisory labels provided by stereo information) as inputs for training a conventional two-class classifier, it has no control on the number of input examples in each negative (non-traversable) and positive (traversable) classes. This is the well known unbalanced data problem. That is a small proportion of the training data may constitute examples of the minority class (non-traversable terrain examples) whilst the rest belongs to the majority class

978-1-4244-7815-6/10/$26.00 ©2010 IEEE

(traversable terrain examples). Standard classification algorithms yield poor results when the training data is unbalanced [8]. In order to solve the problem of unbalanced data, Procopio et al. [9] randomly select a predetermined number of samples for both traversable and non-traversable classes. This method eliminates elements from majority class to match the size of the minority class and if the number of available samples in either class is less than the target number of training examples, learning is not performed for that frame and the vehicle navigates only based on its near-field stereo information. One drawback of this method is that it discards a portion of training samples from the majority class, despite the fact that they may contain crucial information about terrain traversability.

In this paper, we describe a framework for terrain classification using an online, self-supervised learning algorithm. This framework adapts to unknown environments without using any hand-labelled training data. The method described in this paper enhances current near-to-far learning techniques by automating the task of selecting which learning strategy should be chosen from among several strategies based on the nature of the incoming real-time input training data. Whilst many aspects of near-to-far learning are improved in the proposed methodology, the salient feature is the integration of the various techniques in a unified framework.

II. Self-Supervised Learning

The data flow of our self-supervised learning is illustrated in Fig. 1. This includes pre-processing, ground plane estimation, auto labeling, feature extraction, scaling, training and classification.

A. Image Pre-Processing

The vision sensor used in our experiments is a short baseline (12cm) Point Grey Bumblebee colour stereo vision system. Triclops SDK, a Stereo Vision Software Development Kit (SDK) provided by Point Grey is used 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 at a resolution of 320x240. 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 (SAD) correlation method to create a disparity map.

B. Ground Plane Estimation and Obstacle Detection

The obstacles and ground plane estimator module applies fast RANSAC algorithm directly to the disparity image to estimate the dominant ground plane. Firstly, invalid pixels in current disparity map are removed (0 and 255 values). Secondly, 3 points are randomly selected directly from the near-field of the disparity image (up to 7m around the vehicle). 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 fitted to the triple with maximum number of supporting points by least squares fitting is known as dominant ground plane. Complete details of the ground plane estimation method are given in [11]. Ground plane estimation is the most computational complex part of the algorithm. For a 320x240 image, the estimation takes approximately 200 ms.

The dominant ground plane estimated is projected back onto the disparity image to detect obstacles. Any point whose disparity is not within a preselected threshold of the expected ground plane will be considered as an obstacle. The ground plane (shown in green) and obstacles (shown in red) detected in the near-field in the disparity image when projected back to the original RGB image are depicted in Fig. 2.

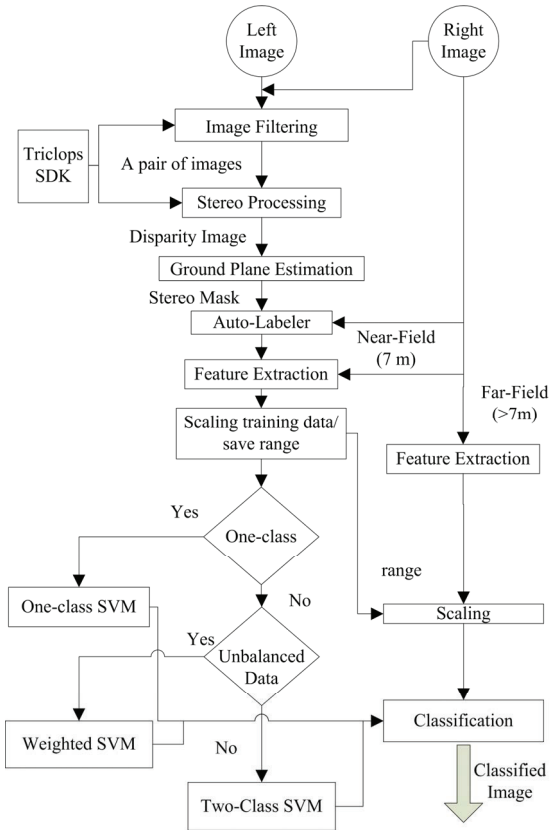

Figure 1. Data flow of proposed self-supervised learning.

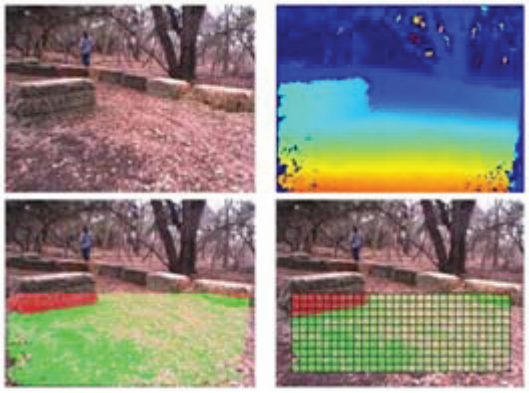

Figure 2. Original RGB image (top left), stereo disparity image (red to blue indicating decreasing disparity) (top right), ground plane (green) and obstacle (red) detection (bottom left), Traversability labels using fixed size patch traversable (green), non-traversable (red) (bottom right).

C. Auto-Labelling

After the ground plane and obstacles are estimated in the disparity image, their corresponding image pixels in the reference image (right image) are labelled as either ground plane or obstacle. In order to reduce computation time, a 320x240 pixels colour 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 on the sides of an image) 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 are counted in each patch, and based on the number of whichever is greater the patch is assigned a traversability label of either obstacle or ground plane. 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 labelled image using patches of size of 10x10. Only the near-field stereo range is labelled (280 patches).

D. Feature Extraction

The main aim of this block is to extract the most distinct features for each labelled patch. The colour information output from the camera is in RGB colour space and as such colour channels are highly correlated and vary drastically with illumination changes. The CIE LAB is nearly uniform colour space and close to human visual perception. Therefore, the Mean and Standard Deviation of CIE LAB colour space are chosen as colour descriptors for each patch. However, colour information is not enough for discriminating regions in outdoor environments (Fig.2). Texture is a fundamental property of surfaces. Image texture features represent a structure in terms of smoothness, coarseness, and regularity of an object. In this work, we use texture analysis using a bank of Gabor filters for constructing the texture descriptors. The mean and standard deviation of the Gabor filter banks responses for each patch with 4 orientations {0, 45, 90, 135 degrees} and 4 scales are used as texture descriptors. Next, the colour representation (6 features) and the texture representation (32 features) are included in one feature vector giving it a depth of 38.

E. Classifier Training

The learning method used is the Support Vector Machines (SVMs) which is currently enjoying much popularity in the machine learning community due to its good performance and less computational demands. SVM looks for the optimal separating hyper-plane between the two classes by maximizing the margin between the classes' closest points [14]. The points lying on the boundaries are called support vectors (SVs). Given a *d*-dimensional input *x* and *y* class label, the support vector machine's task is to minimize the following cost function:

$$\text{Minimize: } \frac{1}{2}\|\mathrm{W}\|^2 + \mathrm{C}\sum_{\mathrm{p}} \zeta_{\mathrm{p}} \qquad (1)$$

$$\text{Subject to: } \begin{cases} \mathrm{d_p}\left(\mathrm{W}^{\mathrm{T}}\mathrm{X_p} + \mathrm{b}\right) \geq 1 \\ \zeta_{\mathrm{p}} \geq 0 \end{cases}, \qquad (2)$$

where, p is the number of input samples, W is the weight matrix of the optimal hyper-plane, d is the desired label, and ζ are the slack variables, which basically allow some misclassification in the training aiming at better overall misclassification in test and train dataset. Value of $\mathrm{C} > 0$ reflects the cost of violating constraints. A large C generally leads to smaller margin but also fewer misclassifications of training data and vice versa. The data points are usually mapped into a higher-dimensional space where the points become linearly separable using $\mathrm{K}(\mathrm{x_i}, \mathrm{x_j}) = \varphi^{\mathrm{T}}(\mathrm{x_i}).\varphi(\mathrm{x_j})$ as a kernel function. Many kernel functions are available, but the four common kernels are Linear, Polynomial, RBF and Sigmoid kernel functions.

III. Proposed Method

A. Weighted SVM

Direct application of SVM yields poor results in the presence of unbalanced data. Some researchers have proposed different penalty parameters for the SVM to overcome the problem of unbalanced data [15]. Weighted support vector machines compensate for the undesirable effects caused by the uneven training class size, and improve on the classification accuracy of the minority class. However, this improvement is achieved at the cost of possible reduction of total classification accuracy, defined by:

$$\min \frac{1}{2}\|\mathrm{W}\|^2 + \mathrm{C}_{+}\sum_{\mathrm{y}=+1} \xi_{\mathrm{i}} + \mathrm{C}_{-}\sum_{\mathrm{y}=-1} \xi_{\mathrm{i}} \qquad (3)$$

where $C_i, i = +,-$ are two penalty parameters for the SVM to overcome the problem of unbalanced data and the rest of the SVM classification is the same.

B. One-Class SVM

Typical two-class classification needs both negative and positive examples for training phase. In practical outdoor settings, the vehicle may not encounter any negative examples (non-traversable) during certain parts of its course. Therefore, two-class classification cannot be used for long-range classification. Recently, one-class classification or novelty detection has received much attention and importance in the practical machine learning research community. One-class classification (distribution estimation) attempts to describe one class of objects (target) and distinguish it from all other possible objects (outliers). Only positive examples are used for training, as it is hard to identify models for negative examples. One-class SVM proposed by Schölkopf et al. [16] recently extended the SVM methodology to handle training using only positive information (one-class). First, it transforms the feature

space via a kernel function to the higher dimension representation. Then, it attempts to separate the feature vectors from the origin that corresponds to the second class.

C. Integrated Learning Method

On every processing cycle, traversability labels associated with a list of scaled feature vectors for all patches of near-field stereo vision are used as input training data to the proposed integrated learning method. The proposed method decides in real-time between three different learning strategies based on the nature of the input data. First, if the traversability label distribution of the incoming data stream contains only one-class examples (e.g. only traversable examples) or the size of minority class is less than 1% of the whole size of input training data, the learning strategy which is chosen is one-class classification (One-Class SVM). Only majority class examples are used for training and classifying the rest of the image. If input training data contains both negative and positive examples, but it is unbalanced, weighted SVM is chosen as the learning strategy to overcome the problem of unbalanced data. In each cycle, two penalty values $C_i, i = +,-$ depending on the percentage of minority and majority class sizes are assigned automatically to the two classes of training data. If input training data contain the same size for both negative and positive examples (each class has class size between 40%-60% of the whole size of input training data), the conventional two-class classification is applied. The data flow of our proposed integrated self-supervised learning is illustrated in Fig. 1. In the next section, we compare our proposed integrated method with baseline near-to-far learning strategy using real datasets from the DARPA-LAGR program.

IV. Experimental Evaluation

A. Baseline Near-to-Far Learning Strategy

To evaluate the performance of our system, we compare it against a baseline near-to-far learning strategy. For the classification module of the baseline near-to-far learning strategy, we chose the conventional two-class SVMs [14]. We have considered an SVM classifier with 4 different kernel functions: linear, second-degree polynomial, third-degree polynomial, and radial basis function (RBF). We observed that the performance gain obtained using a third-degree polynomial is marginal as compared to a linear SVM, which is simpler and computationally much more efficient. In order to evaluate the proposed learning strategy as applied to the baseline learning strategy, we use the same kernel for the classification module of the proposed system.

B. Datasets

The experimental results presented in this study, are based on the image datasets logged during live runs of the robot in the LAGR program, which is publicly available on the Internet [17]. Each dataset has 100 image frame sequences. Each frame consists of a raw RGB image, raw disparity information and a hand-labelled image. Each pixel in a hand-labelled image (Fig. 3(b)) has a label: green, red and blue indicate ground plane, obstacle, and unknown respectively.

C. Evaluation Measurements

The efficiency of the learning applied to computer vision can be measured either qualitatively by means of visual perception or quantitatively by using the criteria classification accuracy. Human eyes simply can tell which classifiers perform qualitatively better, but human visual perception may fail when the classifiers perform almost the same. Classification accuracy is defined as the percentage of corrected classified pixel to total number of pixels. However, when a dataset is unbalanced the error rate of a classifier is not representative of the true performance of the classifier.

To evaluate classification techniques in this study, we use Precision and Recall metrics based on confusion matrix elements. Precision (p) and Recall (r) are calculated using the equation:

$$\begin{cases} \mathrm{p} = \dfrac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \\ \mathrm{r} = \dfrac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \end{cases}, \tag{4}$$

where, the elements of the confusion matrix are the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). 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:

$$\mathrm{F-score} = \frac{(\alpha+1)\mathrm{r}\times\mathrm{p}}{\mathrm{r}+\alpha\times\mathrm{p}} \tag{5}$$

F-score commonly uses “average” of precision and recall (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 given in (8) is generated separately and added to F-score as one of our evaluation performance metrics. These evaluation performance metrics are used to measure differences between the classification output and the ground truth images hand-labelled by a human.

$$\mathrm{FP-rate} = \frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}} \tag{6}$$

D. Experimental Results

Fig. 3(a) shows an example of when a robot senses only traversable (shown in green colour) training examples in the near-field of its stereo vision system range. The baseline near-to-far learning strategy fails to classify the rest of the image correctly, since it does not have examples of both classes during the training phase. However in this case, in the proposed integrated method one-class SVM learning strategy is activated (dataflow in Fig. 1) which can classify the far-field terrain only based on positive examples. Fig. 3(c) and (d) compare representative outputs of baseline near-to-far learning strategy and proposed integrated method quantitatively.

At the next step, if training data contains both positive and negative examples but they are unbalanced, the system chooses in real-time two penalty values depending on the percentage of minority and majority class sizes. Typical classified outputs of baseline near-to-far learning strategy as against the proposed integrated method are shown in Fig. 4(c) and (d). The proposed method automatically assigns proper penalty values for both majority class (in this frame 0.07) and minority one (0.93) since as shown in Fig. 4(a) the robot senses only 7% of total examples as non-traversable (shown in red colour). The results of both baseline near-to-far learning strategy and the proposed integrated method for the 100 frames of one dataset (DS1) are shown in Fig. 5. The FP-rate of baseline SVM is "1" for the first 30 frames since the robot can sense only traversable training examples in its near-field of stereo vision system range, though proposed integrated method has less than "0.3" FP-rate.

After the first 30 frames, training data contains both positive and negative examples but it is unbalanced. In this case, proposed system automatically chooses in real-time penalty values for each frame. Fig. 6 shows comparison of overall FP-rate and F-score values of both baseline near-to-far learning strategy and proposed integrated method for dataset DS1. Table I summarizes the overall performance for the three different scenarios. Totally, our proposed integrated method performs significantly better compared to the baseline learning strategy in terms of greater F-score and less FP-rate while not increasing the computational complexity of the whole procedure.

V. Conclusion and Future Work

An integrated self-supervised learning system was proposed which uses 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. While we have improved many aspects of near-to-far learning, the integration of the techniques is the primary contribution.

Although this approach can adapt to the changing environments concurrently, it lacks memory or history of past information. Future work is to investigate strategies, and algorithms with strong theoretical underpinnings for an online, self-supervised learning algorithm that exploits multiple frames to develop adaptive models that can classify multitude of terrain types.

Acknowledgment

The authors gratefully acknowledge M.J. Procopio of the University of Colorado for providing the log data used as datasets considered in this study.

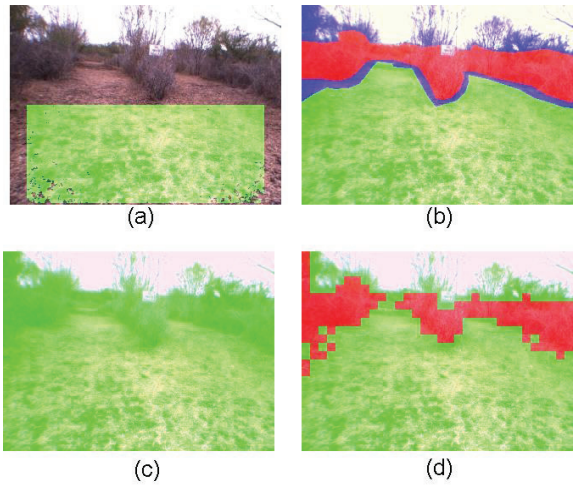

Figure 3. (a) Ground plane and obstacles detection using stereo information in near-field (b) Hand-labelled ground truth (c) Terrain classification using Baseline near-to-far learning (d) Terrain classification using proposed integrated method - frame 1-DS1.

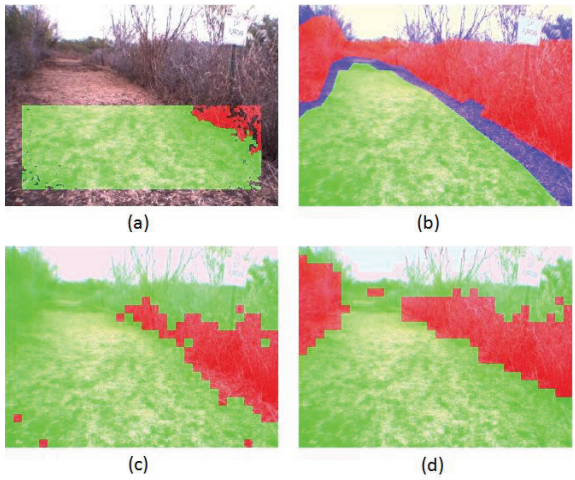

Figure 4. (a) Ground plane and obstacles detection using stereo information in near-field (b) Hand-labelled ground truth (c) Baseline near-to-far learning (d) Proposed integrated method (0.07:1) - frame 79-DS1.

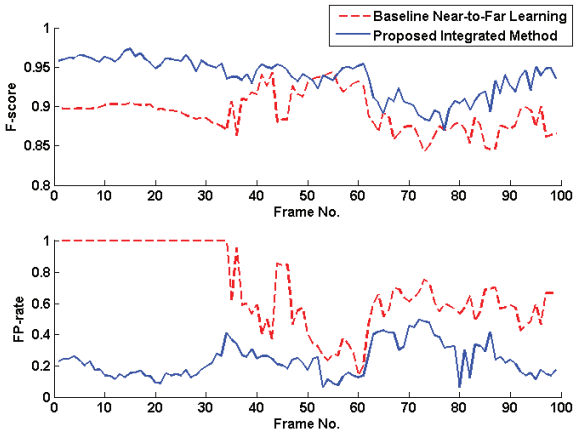

Figure 5. Baseline near-to-far learning (red) vs. proposed integrated method (blue)-100 frames-DS1.

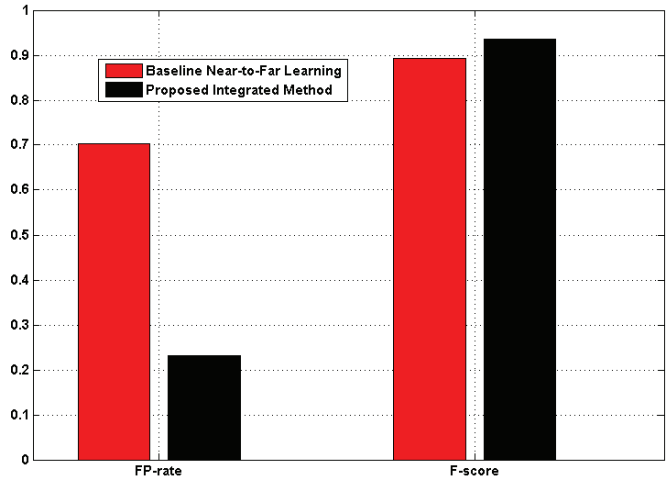

Figure 6. Overall Performance of baseline near-to-far learning (red) vs. proposed integrated method (black)-100 frames-DS1.

TABLE I. OVERALL RESULTS FOR THREE DATASETS

Dataset	Baseline		Proposed	
	F-score	***FP-rate***	***F-score***	***FP-rate***
DS1	89.2%	70.2%	93.6%	23.2%
DS2	93.6%	12.5%	94.1%	5.6%
DS3	93.2%	62%	93%	30.6%
Mean	92%	48.23%	93.56%	19.8%

REFERENCES

[1] P. Bellutta, R. Manduchi, L. Matthies, K. Owens and A. Rankin, "Terrain perception for DEMO III," *In Proc. of the IEEE Intelligent Vehicles Symp, 2000.*

[2] 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*, 2006, vol. 23, pp. 945–973.

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

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

[5] K. Konolige, M. Agrawal, M. R. Blas, R. C. Bolles, B. Gerkey, J. Sundaresan and A. Sola, "Mapping, Navigation, and Learning for Off-Road Traversal," *Journal of Field Robotics*, 2009, vol.26, pp. 88-113.

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

[7] J. Mulligan and G. Grudic, Guest Eds. "Special issue on machine-learning-based robotics in unstructured environments". *Journal of Field Robotics, 2006,* vol. 23, no. 9, pp. 655-835.

[8] G. M. Weiss and H. Hirsh, "Learning to predict extremely rare events," *Proceedings of learning from Imbalanced Data Sets. AAI Workshop. Technical Report 2000, WS-00-05*, 64-68.

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

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

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

[12] P. Moghadam, W. S. Wijesoma, "Online, Self-Supervised Vision-Based Terrain Classification in Unstructured Environments," *IEEE International Conference on Systems, Man and Cybernetics, pp.3100-3105, 11-14 Oct. 2009*

[13] T. Hastie, R. Tibshirani and J. Friedman, *The Elements of Statistical Learning*. Springer, Berlin, 2001.

[14] C. C. Chang, and C. J. Lin, "LIBSVM: a library for support vector machines. *software available at", 2001, http://www.csie.ntu.edu.tw/cjlin/libsvm/.*

[15] E. Osuna, R. Freund and F. Girosi, "Support vector machines: Training and applications" [R]. MIT, AI Memo, 1997, No. 1602.

[16] B. Schölkopf, A. J. Smola, R. C. Williamson and P. L. Bartlett, "New Support Vector Algorithms", *Neural Computation*, 2000, vol. 12, no. 5, pp. 1207-1245.

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