Self-Supervised Classification for Planetary Rover Terrain Sensing

Christopher A. Brooks
Department of Mechanical Engineering
Massachusetts Institute of Technology
77 Massachusetts Ave
Cambridge, MA 02139
617-253-5095
cabrooks@mit.edu

Karl D. Iagnemma
Department of Mechanical Engineering
Massachusetts Institute of Technology
77 Massachusetts Ave
Cambridge, MA 02139
617-452-3262
kdi@mit.edu

Abstract—Autonomous mobility in rough terrain is key to enabling increased science data return from planetary rover missions. Current terrain sensing and path planning approaches can be used to avoid geometric hazards, such as rocks and steep slopes, but are unable to remotely identify and avoid non-geometric hazards, such as loose sand in which a rover may become entrenched. This paper proposes a self-supervised classification approach to learning the visual appearance of terrain classes which relies on vibration-based sensing of wheel-terrain interaction to identify these terrain classes. Experimental results from a four-wheeled rover in Mars analog terrain demonstrate the potential for this approach.[1][2]

1. Introduction and Related Work

The ability to explore other planets using mobile robots is fundamentally dependent on the autonomous mobility capabilities of these robots. Targets of scientific interest such as craters, ravines, and cliffs present dangers to landing, therefore planetary rovers must land at distant sites then travel to the target of interest. Limits in communication with observers on Earth mean that close supervision of robots puts significant restrictions on the distance a rover can travel during a mission lifetime. Thus, advances in robot autonomy will lead to payoffs in terms of scientific return.

One current limit in autonomous mobility lies in terrain sensing capability. Given an accurate map of the ease of traversability of terrain, existing path planning algorithms can generate a route to a target which avoids known obstacles [1,2,3]. Unknown hazards and uncertainties in the map are sources of potential danger. The ability to remotely detect possible hazards would enable safe autonomous traversal of previously unexplored rough terrain.

While geometric hazards, such as large rocks, can be sensed remotely using stereo vision [4], little research has addressed remote sensing of non-geometric hazards, such as the loose drift material in which the Mars Exploration Rover (MER) Opportunity became entrenched in April, 2005. Non-geometric hazards are highly dependent on wheel-terrain interaction, so the sensing of such hazards has focused on using the rover wheels as sensors. Examples include wheel sinkage measurement [5,6], parametric soil characterization [7], wheel slip detection [8] and explicit traversability estimation [9]. This sensing is inherently local—the rover wheel must be in contact with the terrain to make a measurement—so it is of limited use for hazard avoidance.

Remotely sensing non-geometric hazards depends on generalizing locally sensed data to gain information about more distant terrain, and this has only recently received attention from researchers. One group has attempted to predict future wheel slip based on the rover's past slip on terrain classes whose appearance is known *a priori* [10]. Another group has attempted to distinguish traversable terrain from non-traversable terrain using a self-supervised framework, to reduce the false positive detection of deformable obstacles such as tall grass [11]. Other researchers have used laser range sensors mounted on a ground vehicle to estimate a traversability cost of paths through natural terrestrial terrain, then learned a regression model based on color overhead imagery to predict traversability costs for terrain over the horizon [12]. However no research has addressed the detection of non-geometric obstacles where their appearance has not been known *a priori*.

This paper addresses the issue of generalizing local terrain sensing by training a visual classifier to recognize classes corresponding to different outputs of the local sensors. In the context of mobility, local sensors identify a terrain class, and a visual classifier identifies where that terrain appears in the distance. For this work, local terrain was classified based on vibrations in the rover structure, and distant terrain was classified as belonging to one of the locally identified classes. Experiments were performed using a four-wheeled rover in Mars-analog rough terrain. Results demonstrate the accuracy of this classifier at identifying the locally-sensed terrain class in the distance.

[1] 1-4244-0525-4/07/$20.00 ©2007 IEEE.
[2] IEEEAC paper #1164, Version 5, Updated December 5, 2006

Section 2 describes the self-supervised classification approach and the architecture of the component classifiers. Section 3 gives details about the experiments. Section 4 presents the results of the experiments. Section 5 describes the conclusions drawn from these results and suggests directions for future research.

2. APPROACH

Overview

In this paper, the self-supervised classification is composed of a previously trained vibration-based terrain classifier which provides labels for training the visual classifier. Once labeled training data is available, standard supervised classification techniques can be used to train the visual classifier. Details of the component vibration-based and vision-based classifiers are presented in this section followed by details the self-supervised training approach.

Local Vibration-Based Classification

The supervisory classifier, which performs the initial labeling of terrain patches, is a vibration-based terrain classifier. This classifier labels terrain based on vibrations recorded by a contact microphone fixed to the rover suspension near one of the wheels. This sensory mode was proposed in [13], and classification results were presented in [14]. Similar work has been presented for high-speed ground vehicles in [15].

The vibration data features used in this paper are the same as those used in [14]. Vibrations are represented as the log-scaled magnitudes of the power spectral density of the vibration time signal. For vibrations recorded at 44.1 kHz, 557 frequencies are used ranging from 20.5 Hz to 12 kHz. These magnitudes are averaged over 1-second windows to reduce noise. While 12 kHz is very high compared to the frequencies normally associated with vehicle vibrations, it is reasonable for sensing rigid wheels climbing on rocks.

This 557-element vector data representation is used as an input to a support vector machine (SVM) classifier [16]. Support vector machines are well-established classifiers which use kernel functions to estimate the similarity between new data and training data. For vibration-based classification, a second-order polynomial kernel is used, because it demonstrated the highest classification accuracy in previous experiments. For multi-class classification, the probability estimation approach described in [17] is used. This yields a class likelihood for each terrain class. For this work, SVM algorithms are implemented using the LIBSVM library, with additional optimizations for classification with a linear kernel [18].

Remote Vision-Based Classification

The classifier trained using self-supervised classification is a vision-based terrain classifier. It identifies terrain based on images collected using a forward-looking color stereo camera pair. Vision-based classification has been studied extensively for terrain classification (e.g. [19,20]). Features observable using stereo cameras include color, visual texture, and geometry. These features are represented as inputs to the vision-based classifier as described below.

Color data is directly available from the cameras as red, green, and blue (RGB) intensities. However, overall illumination intensity affects all three values in a raw RGB representation and can yield poor classification results, so the hue, saturation, and value (HSV) representation is used as in [12]. Here hue (an angle) is represented as two values sin(hue) and cos(hue) to eliminate the artificial discontinuity at 2π. Thus, color is represented as a 4-element vector: [sin(hue), cos(hue), saturation, value].

Visual texture is a measure of the local spatial variation in intensity in the image. Researchers have proposed many metrics for visual texture, using Gabor filters and local energy methods [21,22]. This work uses a wavelet-based approach, similar to the one demonstrated in [23]. Here the grayscale image is decomposed with the Haar wavelet. Three scales of wavelets are used, each scale having horizontal, diagonal, and vertical (HDV) wavelets, corresponding to estimating the derivative in the horizontal, diagonal, and vertical directions at each length scale. The scales used are 2, 4, and 8 pixels. Because this data is noisy, the magnitudes of the wavelet coefficients are averaged over windows of 7, 9, and 11 wavelets. Thus, visual texture is represented by a 9-element vector, composed of the window-averaged horizontal, diagonal, and vertical wavelet coefficients at each scale.

Geometry data is available for the terrain through stereo image processing. This raw output is a cloud of points. For this work, we consider the terrain to be a grid of 20-cm by 20-cm patches, and calculate statistics of the points within each grid. The first element of the geometric feature vector is the average slope of the terrain, defined as the angle the least-squares-fit plane makes with the horizontal. The second element is the mean-squared deviation from that plane along its normal. The third element is the variance in height of the pixels, and the fourth is the height difference between the highest and lowest pixels in the patch. Thus, the geometry of each patch is represented at a 4-element vector: [slope, plane fit deviation, height variance, height range].

As with the vibration-based classification, the vision-based classification uses a support vector machine classifier. The straightforward approach of concatenating the color, visual texture, and geometric features into a single feature vector resulted in poor classification, so naïve Bayes fusion is used

for the results presented here. This approach assumes that color, visual texture, and geometric features are conditionally independent given the terrain class. Thus, the likelihood of a terrain patch belonging to a terrain class is the product of the class likelihoods for each sensing mode. Note that since there may be many pixels observed in each terrain patch, the overall estimate of the class likelihood for the color mode is taken as the geometric mean of the individual pixel class likelihoods. The same approach is used for visual texture.

To estimate the class likelihoods for each sensing mode, a support vector machine is used, just as for the vibration data. Training is done separately for each sensing mode. For this SVM, linear and low-order polynomial kernels are considered because they can be computed efficiently independent of the amount of training data. This is especially important for the color and texture modes because the classifier is applied to each pixel individually.

Self-Supervised Training Framework

In this paper, two different approaches are used to obtain visual data needed for training the vision-based classifier. The first approach, termed "local training," uses images from a belly-mounted camera captured simultaneously with the vibration sensing. A sample image from such a camera is shown in Figure 1. This approach is very easy to implement, but has drawbacks in that range data from the terrain is not available in a monocular image, texture data is not easily comparable between the monocular images and forward-looking stereo images, and color calibration between the monocular images and forward-looking stereo images needs to be carefully controlled. The second approach, termed "remote training," stores forward-looking stereo images in memory and recalls the appearance of a patch of terrain when the rover is driving over it and "feels" which terrain it is. This second approach relies heavily on stereo processing to identify the distance to patches in the image and on accurate position estimation to identify when the rover has reached a patch of terrain. However, this approach does not depend on accurate color calibration between the cameras, and texture and geometric data are readily available to use for training the classifier.

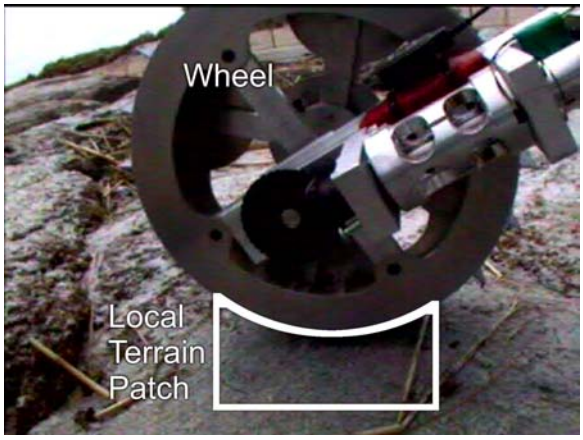

Figure 1. Sample image from belly-mounted camera

For either approach, training data is accumulated as the vehicle drives along, with a separate batch of training data for each terrain. To limit the training time for the visual classifier, each terrain is limited to a maximum of 400 data points—"old" data is forgotten if new data arrived that would exceed that maximum. At the end of each data set, the training data is used to train a SVM classifier. Classification of the terrain is implemented on a patch level, based on naïve Bayes fusion of the class likelihoods of the color, visual texture, and geometric sensing modes.

3. EXPERIMENT DETAILS

This self-supervised classification approach is compared to traditional manually trained classification using experiment data collected on TORTOISE, a four-wheeled mobile robot developed at MIT, in Mars-analog outdoor terrain.

Robot Configuration

TORTOISE, shown in Figures 2 and 3, is an 80-cm-long, 50-cm-wide, 90-cm-tall robot with 20-cm-diameter rigid aluminum wheels with grousers. The wheels on either side are connected to the main body and mast via a differential. The robot is outfitted with a forward-looking mast-mounted stereo camera pair, a belly-mounted monocular camera, a

Figure 2. TORTOISE rover, showing section enlarged in Figure 3

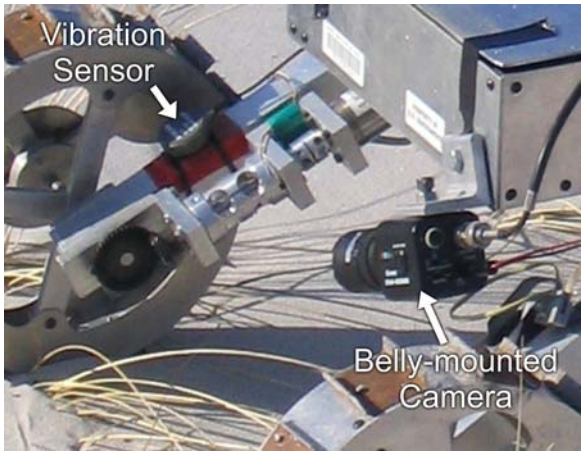

Figure 3. Local terrain sensors on TORTOISE

vibration sensor, and a body-mounted two-axis tilt sensor. Forward-looking images were captured with the stereo camera pair, a Videre Design "dual DCAM" capable of capturing color images with 640 x 480 resolution. Range data were extracted from the stereo images using Videre Design's commercial stereo processing software [24]. Color images of the right-front wheel and its immediate surroundings were captured using the belly-mounted monocular camera. Vibrations from the right-front wheel were sensed using a Signal Flex SF-20 contact microphone. Body pitch and roll were measured with a Crossbow CXTA02 two-axis tilt sensor. Additional sensors include a torque sensor and suspension configuration sensors, though these sensing modes are not used in this work.

During experiments, the rover traveled at an average speed of 6 cm/sec. Vibrations were recorded at 44.1 kHz, body pitch and roll were captured at 25 Hz, images from the belly-mounted camera were captured at 2 Hz, and forward-looking stereo images were captured every 1.5 seconds. These data were stored during experiments and processed offline.

Experiment Environment

Experiments were performed at Wingaersheek Beach, in Gloucester, MA. This is a sandy beach with a mixture of small and large outcrops, relative to the size of the rover, and loose rocks. This site was chosen due to its similar in appearance to the MER landing sites on Mars. In this environment, sand and rock were considered to be distinct terrain classes. To demonstrate the ability of the classification approach in a multi-class setting, matted piles of beach grass were used as a third terrain class. A sample scene is shown in Figure 4. In this image, sand appears as a uniform gray flat surface, rock appears tan and orange with some steep slopes and fine uniform texture, and beach grass appears highly-textured with mixed browns and shadows.

Experiments used in this analysis were conducted over two days. Each data set was collected during a single rover

Figure 4. Sample scene from Wingaersheek Beach, with terrains labeled

traverse approximately 10 meters along a straight-line path containing a combination of the three terrains. No two paths were identical. During the experiments cloud cover ranged from overcast to direct sun.

Data Processing

Data were collected during the experiments and stored for processing offline. For data sets used for testing the accuracy of the vision-based classifiers, every 5th image from the stereo pair was hand-labeled to identify a ground-truth terrain class for each pixel. Separate hand-labeled images were used for training the manually trained vision-based classifiers. Similarly, the vibration-based classifier was trained using hand-labeled training data; the true terrain was identified using images collected by the belly-mounted camera. Five data sets with hand-labeled images contained vibration data from all three terrain types; these 108 images were used for assessing the accuracy of the self-supervised visual classifier.

4. RESULTS

Approach 1: Local Training

The first approach for self-supervised training of the visual terrain classifier uses images from the belly-mounted camera labeled using the vibration-based classifier as training data. Since geometry information is not available from the belly-mounted camera and visual texture is not easily comparable between the belly-mounted camera and the stereo cameras, color information is used exclusively.

Figure 5 shows the receiver operating characteristic (ROC) curves for the locally trained visual classification using the self-supervised approach applied to 16 test images from one data set. Here, the horizontal axis indicates the percentage

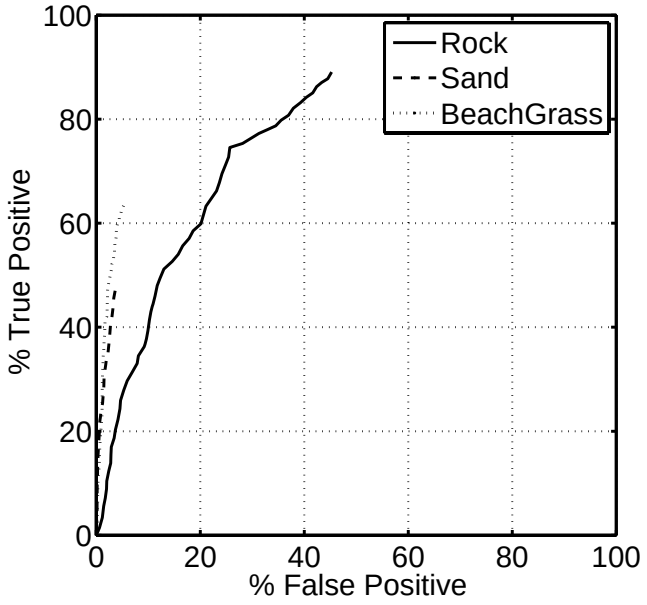

Figure 5. ROC curves for locally trained classifier

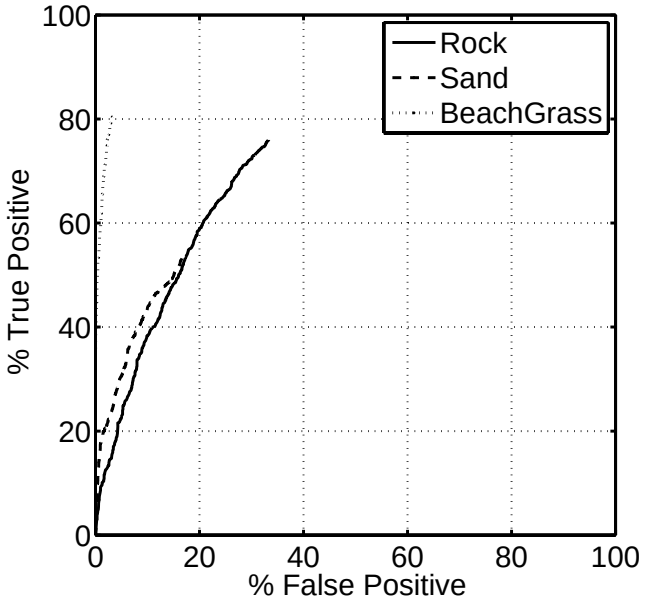

Figure 6. ROC curves for manually trained classifier

of false positives (%FP) (for example, instances when sand or beach grass were falsely identified as rock) and the vertical axis indicates the percentage of true positives (%TP). Each terrain forms a curve on the plot, as the threshold for leaving terrain "unclassified" is adjusted. In this plot, it can be seen that the classifier was very good at discriminating beach grass from the other two terrains. With no data unclassified, 63% of the beach grass was positively identified, with only 5% of the other terrain being falsely identified as beach grass. Classification of sand was also very accurate, with 47% of the sand positively identified and only 4% of other terrains falsely identified as sand. Classification of rock wasn't as successful, however. With no data remaining unclassified, 89% of the rock was successfully identified, though 45% of the other terrains falsely identified as rock. Note that this is still significantly better than random guessing, which would have equal values for the true positive and false positive percentages.

For comparison, a manually trained classifier was implemented. This classifier was trained using color, texture, and geometry data from 5 images from the data set not included in the test set. For each terrain 400 data points for each sensing mode were randomly selected and used to train a SVM classifier. Shown in Figure 6 are ROC curves for this classifier applied to the same test images as used for the self-supervised classifier. Here again, beach grass was well discriminated from the other two terrain classes, and rock was discriminated less accurately. The main difference between the manually-trained classifier and the locally-trained self-supervised classifier in Figure 5 is the classification rate of sand. In the manually-trained classifier, far more terrain was falsely identified as sand. This could be due to small changes in illumination over the course of the test run decreasing the performance of the manually-trained classifier. On the other hand, there were improvements in the rate of classification of beach grass, which suggests that addition of texture and geometry data may make the classification of beach grass easier.

Self-supervised classifiers and manually trained classifiers were implemented for each of the five data sets, and the results are shown in the Table 1. To test whether the difference was due to the use of texture and geometric

Table 1. Comparison of local semi-supervised training to manual training

	Local Training Self-supervised Classifier (Color only)	Manually trained Classifier (Color only)	Manually trained Classifier
Mean % True Pos. [95% Conf. Int.]	48.8% [30.1% - 67.5%]	67.7% [56.1% - 79.3%]	67.1% [50.6% - 83.5%]
St. Dev. of % True Pos.	36.6%	22.6%	32.2%
Mean %TP/(%TP + %FP) [95% Conf. Int.]	0.72 [0.62 – 0.82]	0.82 [0.76 – 0.89]	0.87 [0.82 – 0.92]
St. Dev. of %TP/(%TP + %FP)	0.20	0.13	0.10

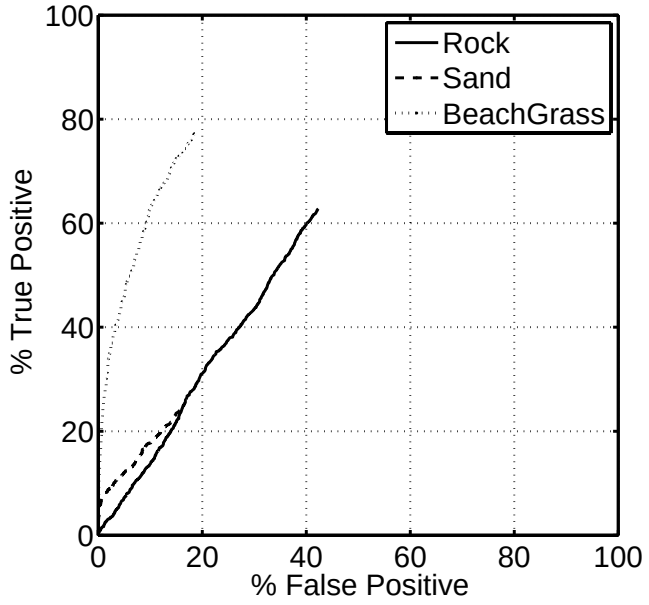

Figure 7. ROC curves for remotely trained visual classifier

features, an additional manually trained classifier was implemented using only color data, and results for that classifier are also shown in Table 1. The first two rows show statistics of the true positive percentage of the classifiers when no data is left unlabeled, corresponding to the vertical coordinate of the ROC curve endpoints. The second two rows show statistics related to the ratio between the true positive percentage and the false positive percentage. The metric, %TP/(%TP + %FP), can be considered the fraction of labeled patches which are labeled correctly.

In this table, it can be seen that both of the manually trained classifiers outperform the self-supervised classifier (both mean %TP and mean %TP/(%TP + %FP) are higher, though not beyond their 95% confidence intervals), suggesting that the vibration-labeled training data for the self-supervised classifier is not a robust representation of the data as observed by the stereo camera pair. This is likely due to a combination of miscalibration of the colors between the belly-mounted camera and the forward-looking stereo pair, and the different appearance of the terrain as observed close-up. It can also be seen in Table 1 that the standard deviation of these performance measures are largest in the self-supervised classifier, so the relative performance of these classifiers as shown in Figures 5 and 6 is not reflected across all data sets. This suggests that additional factors may be involved, such as the angle of each camera relative to the sun. These challenges motivate the second, remote-training approach, which trains a vision-classifier using the same cameras which will be used for classification under as similar lighting conditions as possible.

Approach 2: Remote Training

The second approach for self-supervised training of the visual classifier is to maintain a record of the appearance of terrain patches in the stereo images and recall that appearance later when the rover drives over that terrain. Since this approach uses the same cameras for collecting training data and test data, color, texture, and geometry data are all available for training, and the potential for color miscalibration is eliminated.

Figure 7 shows the ROC curves for the remotely trained visual classification of one test set. It can be compared directly to Figures 5 and 6. In this data set, the remote training approach performs best in classifying beach grass. Classification of rock and sand were significantly less accurate.

Remotely trained self-supervised classifiers and manually trained classifiers were implemented for each of the five data sets, and the results are shown in the first two columns of Table 2. The rows of the table are the same as Table 1. Here it can be seen that the manually trained classifier outperforms the self-supervised classifier, by a statistically significant difference in the case of mean %TP/(%TP + %FP). This difference may be attributed to the main challenge of self-supervised classification—getting properly labeled training data.

However, the self-supervised approach is intended for situations when a manually trained classifier is not a viable option. Thus, the accuracy of a self-supervised classifier is

Table 2. Comparison of remote semi-supervised training to manual training

	Remote Training Self-supervised Classifier	Manually trained Classifier	Manually trained Classifier (Prior Data Set)	Local Patch Manually trained Classifier
Mean % True Pos. [95% Conf. Int.]	48.3% [29.2% - 67.4%]	67.1% [50.6% - 83.5%]	57.8% [38.1% - 77.5%]	61.4% [43.3% - 79.5%]
St. Dev. of % True Pos.	37.4%	32.2%	38.5%	35.3%
Mean %TP/(%TP + %FP) [95% Conf. Int.]	0.70 [0.60 – 0.79]	0.87 [0.82 – 0.92]	0.77 [0.63 – 0.90]	0.82 [0.73 – 0.90]
St. Dev. of %TP/(%TP + %FP)	0.18	0.10	0.27	0.15

more fairly compared to a manually trained classifier trained on the previous data set, because the time it takes to manually label data would delay the classifier's implementation. The accuracy of such a classifier is indicated in the third column of Table 2. Here the difference between the remotely-trained classifier and the manually-trained classifier is not as significant, though it still suggests that the manually-trained classifier is more accurate.

The fact that the self-supervised classifier does not achieve the performance of a manually trained classifier indicates that some improvements may be possible. To distinguish problems with the self-supervised approach from problems with the supervising classifier (the vibration-based terrain classifier) an additional visual classifier was trained using manually labeled data drawn only from terrain patches which had vibration data associated with them. The accuracy of this classifier is shown in the last column of Table 2. Here it can be seen that training using only traversed terrain patches has the potential to yield performance nearly as good as the manually trained classifier. The main difference between this local patch manually trained classifier and the self-supervised classifier is the number of terrain patches used for training. The self-supervised classifier only uses terrain patches for which the vibration-based terrain classifier is confident, so improving the classification rate of the vibration classifier and increasing the length of the data sets should yield significantly better performance.

Computation time

An effort was made to limit the computational complexity of training and testing of these classifiers, so the most computationally intensive tasks are the stereo data extraction and the texture feature computation. Extraction of the geometric features from the 3D point cloud takes an average of 5 seconds per image using a Matlab script on a Pentium 1.8 GHz desktop. Texture feature extraction takes 17.3 sec, using an unoptimized Matlab script. A C-code implementation would be expected to run much faster.

Because it doesn't rely on stereo data or texture feature extraction, the locally-trained classifier is very quick to train and classify data. Training takes 1.7 seconds on a Pentium 1.8 GHz desktop, using the LIBSVM library. Color-based classification takes only 0.7 seconds for a full 640x480 image. Classifying terrain patches requires that stereo data be computed to identify which pixel belongs to which patch; once that data is computed, each patch took an average of 0.01 seconds to classify in Matlab, with an average of 400 patches per image. The remotely-trained classifier took slightly less time to train, 1.5 sec on average, because some of the terrains had fewer than 400 sample data points. Classification was slower for the remotely-trained classifier, 4.1 seconds per image, due to the use of texture and geometric data.

5. CONCLUSIONS

In this work, two self-supervised classification approaches were described to create a visual classifier based on terrain classes sensed locally using a vibration-based terrain classifier. The first approach uses images from a belly-mounted camera as training data. It was shown to have good performance in classifying terrain, but robustness issues and the challenge of doing accurate color calibration motivated the development of the second approach. The second approach uses a memory of past images of the terrain patch currently being traversed to supply the training data for the visual classifier. This approach was shown to perform nearly as well as a manually trained classifier trained on data from a prior data set, which is the main alternative to self-supervised classification. Experiments also demonstrated that improvements in the supervising classifier, in this case the vibration-based terrain classifier, have the potential to increase the accuracy of semi-supervised classification to that of the manually supervised classifier. It is likely that improved classification could be realized using the current classifiers with longer data sets to increase the amount of training data.

Future Work

Since classification accuracy seems to be currently limited by the amount of training data, longer data sets need to be collected to assess self-supervised classifier performance. Also, though vibration-based classification accuracy is already good, improvements in its classification rate and accuracy could lead to further improvements self-supervised visual classifier's accuracy. Finally, since the overall goal of this research is to be able to predict the traversability properties of the terrain, incorporation of terrain properties into the terrain class estimates would be a logical next step.

ACKNOWLEDGEMENT

This work was supported by the NASA Jet Propulsion Laboratory (JPL) through the Mars Technology Program.

REFERENCES

[1] Nilsson, N. J. (1980) *Principles of Artificial Intelligence*, Tioga Publishing Company, Palo Alto, CA. 1980.

[2] Stentz, A. (1994). "The D* algorithm for real-time planning of optimal traverses" Technical Report CMU-RI-TR-94-37, Robotics Institute, Carnegie Mellon University, October 1994.

[3] Goldberg, S., Maimone, M., and Matthies, L. (2002) "Stereo vision and rover navigation software for planetary exploration" in *Proceedings of the IEEE Aerospace Conference,* Big Sky, Montana, March 2002.

[4] Talukder, A., Manduchi, R., Castaño, R., Owens, K., Matthies, L., Castano, A., Hogg, R. (2002). "Autonomous Terrain Characterization and Modeling for Dynamic Control of Unmanned Vehicles" in *Proceedings of the IEEE Conference on Intelligent Robots and Systems (IROS),* Lausanne, Switzerland, September 2002.

[5] Wilcox, B. (1994). "Non-geometric hazard detection for a Mars Microrover", in *Proceedings of the AIAA Conference on Intelligent Robotics in Field, Factory, and Space,* Vol. 2, 1994.

[6] Brooks, C., and Iagnemma, K. (2006). "Visual wheel sinkage measurement for planetary rover mobility characterization," *Autonomous Robots*, Vol. 21, pp. 55-64. DOI 10.1007/s10514-006-7230-9.

[7] Iagnemma, K., Shibly, H., and Dubowsky, S. (2002). "On-line terrain parameter estimation for planetary rovers." in *Proceedings of the 2002 IEEE International Conference on Robotics and Automation (ICRA 02)*, Washington, DC, May 2002.

[8] Reina, G., Ojeda, L., Millella, A., and Borenstein, J. (2006). "Wheel slippage and sinkage detection for planetary rovers," *IEEE Transactions on Mechatronics, Special Issue on Novel Aspects in Robotics*, Vol. 11, No. 2, April 2006, pp. 185-195.

[9] Kang, S. (2003), "Terrain parameter estimation and traversability assessment for mobile robots" MS Thesis, Department of Mechanical Engineering, MIT, 2003.

[10] Angelova, A., Matthies, L., Helmick, D., Sibley, G., Perona, P. (2006). "Learning to predict slip for ground robots," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2006.

[11] Kim, D., Sun, J., Oh, S. M., Rehg, J. M., and Bobick, A. F. (2006). "Traversability Classification using Unsupervised On-line Visual Learning for Outdoor Robot Navigation" in *Proceedings of the 2006 IEEE International Conference on Robotics and Automation,* Orlando, Florida. May, 2006.

[12] Sofman, B., Lin, E., Bagnell, J. A., Vandapel, N., Stentz, A. (2006). "Improving Robot Navigation Through Self-Supervised Online Learning" in *Proceedings of Robotics: Science and Systems*, August 2006.

[13] Iagnemma, K. D. and Dubowsky, S. (2002). "Terrain estimation for high speed rough terrain autonomous vehicle navigation," in *Proceedings of the SPIE Conference on Unmanned Ground Vehicle Technology IV*, Orlando, FL, 2002, Paper 4715-31.

[14] Brooks, C., and Iagnemma, K. (2005). "Vibration-based terrain classification for planetary rovers." *IEEE Transactions on Robotics*, Vol. 21, No. 6., pp. 1185-1191, December 2005.

[15] Sadhukhan, D., Moore, C., and Collins, E. (2004). "Terrain estimation using internal sensors," in *Proc. of the IASTED Int. Conf. Robot. Appl.,* Honolulu, HI, 2004, Paper 447-800.

[16] Vapnik, V. (1995). *The Nature of Statistical Learning Theory.* Springer: New York. 1995.

[17] Wu, T.-F., Lin, C.-J., and Weng, R. C. (2004). "Probability estimates for multi-class classification by pairwise coupling" *Journal of Machine Learning Research*, 5:975–1005.

[18] Chih-Chung C., and Chih-Jen L. (2001). LIBSVM: a library for support vector machines. Software retrieved January, 2006 available at *http://www.csie.ntu.edu.tw/~cjlin/libsvm.*

[19] Rasmussen, C. (2002) "Combining Laser Range, Color, and Texture Cues for Autonomous Road Following", in *Proceedings of the 2002 IEEE International Conference on Robotics & Automation*, Washington, DC, May 2002.

[20] Shi, X., and Manduchi, R. (2003). "A Study on Bayes Feature Fusion for Image Classification," IEEE Workshop on Statistical Algorithms for Computer Vision.

[21] Bouman, C. and Liu, B. (1991). "Multiple Resolution Segmentation of Textured Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 2, pp. 99-113.

[22] Reed, T., and Hans du Buf, J. (1993). "A review of recent texture segmentation and feature extraction techniques," *CVGIP: Image Understanding*, Vol. 57(3), May 1993, pp. 359-372.

[23] Espinal, F., Huntsberger, T., Jawerth, B., Kubota, T. (1998) "Wavelet-Based Fractal Signature Analysis for Automatic Target Recognition," Optical Engineering, Vol. 37, No. 1, pp166-174, January 1998.

[24] Videre Design (2006) Retrieved on May 25, 2006 at *http://www.videredesign.com/index.htm*.

BIOGRAPHY

Christopher Brooks *is a graduate student in the Mechanical Engineering department of the Massachusetts Institute of Technology. He received his B.S. degree with honor in engineering and applied science from the California Institute of Technology in 2000, and his M.S. degree from the Massachusetts Institute of Technology in 2004. His research interests include mobile robot control, terrain sensing, and their application to improving autonomous robot mobility. He is a member of Tau Beta Pi.*

Karl Iagnemma *is a research scientist in the Mechanical Engineering department of the Massachusetts Institute of Technology. He received his B.S. degree* summa cum laude *in mechanical engineering from the University of Michigan in 1994, and his M.S. and Ph.D. from the Massachusetts Institute of Technology, where he was a National Science Foundation graduate fellow, in 1997 and 2001, respectively. He has been a visiting researcher at the Jet Propulsion Laboratory. His research interests include rough-terrain mobile robot control and motion planning, robot-terrain interaction, and robotic mobility analysis. He is author of the monograph* Mobile Robots in Rough Terrain: Estimation, Motion Planning, and Control with Application to Planetary Rovers *(Springer, 2004). He is a member of IEEE and Sigma Xi.*