17026

Terrain perception for robot navigation

Robert E. Karlsen*[a] and Gary Witus[b]
[a]U.S. Army –TARDEC, Warren, MI 48397-5000
[b]Turing Associates, Ann Arbor, MI 48103

ABSTRACT

This paper presents a method to forecast terrain trafficability from visual appearance. During training, the system identifies a set of image chips (or exemplars) that span the range of terrain appearance. Each chip is assigned a vector tag of vehicle-terrain interaction characteristics that are obtained from simple performance models and on-board sensors, as the vehicle traverses the terrain. The system uses the exemplars to segment images into regions, based on visual similarity to the terrain patches observed during training, and assigns the appropriate vehicle-terrain interaction tag to them. This methodology will therefore allow the online forecasting of vehicle performance on upcoming terrain. Currently, the system uses a fuzzy c-means clustering algorithm for training. In this paper, we explore a number of different features for characterizing the visual appearance of the terrain and measure their effect on the prediction of vehicle performance.

1. INTRODUCTION

Most, if not all, unmanned ground vehicles currently in use are teleoperated. Typically, the operator relies exclusively on visual input from a video camera to select the route and speed. Teleoperation is robust and effective. Vision processing for autonomous and semi-autonomous navigation has not matched the human operator's visual terrain understanding. Current approaches to autonomous/semi-autonomous navigation employ a wide gamut of sensors including 3D imaging LIDAR, ground penetrating radar, multi-spectral stereovision, ultrasound, and other sensor modalities to detect potential obstacles and forecast trafficability. Inspired by the ability of human operators, our research is focused on methods to assess terrain trafficability directly from image appearance. We do not address obstacle detection, which is an important, but separate cognitive process.

We present an approach to automated image segmentation and terrain classification using exemplars, or small image samples, to represent the variety of terrain appearance. Each chip is assigned a set of measured vehicle-terrain interaction (VTI) parameters that describe the vehicle's performance while driving over that particular terrain. Important measures include vehicle slip, ground resistance and terrain roughness. This process requires three main functions: segmenting the terrain into areas that are visually similar, measuring and computing appropriate measures of the vehicle-terrain interaction, and matching the VTI parameters to the correct image chip. During runtime, local pieces of terrain are assigned to the exemplar to which they are most similar in appearance and inherit the VTI parameters of the exemplar. Previous work has been performed in determining meaningful and robust VTI parameters[1]. In this paper we will utilize measures of ground resistance and roughness.

Exemplar models assume that intact stimuli are stored in memory, and that classification or recognition is determined by the degree of similarity between a stimulus and the stored exemplars. Exemplar methods admit evolution of similarity metrics, since the entire sample is stored intact in memory and not merely a feature vector summary. Simple generalization effects explain correct classification of novel (i.e., previously unseen) instances of categories. Only the item information is used for classification decisions. Categorization relies on the comparison of a new stimulus with known exemplars of the category.

Exemplar models are the most parsimonious models of categorization in terms of the underlying associative mechanism[2]. Exemplar based learning has been proposed as a model of human learning[3] and has since been shown to

* robert.karlsen@us.army.mil

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
09 APR 2007	**N/A**	**-**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Terrain perception for robot navigation	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
Robert E. Karlsen; Gary Witus	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
US Army RDECOM-TARDEC 6501 E 11 Mile Rd Warren, MI 48397-5000 Turing Associates, Ann Arbor, MI 48103	**17026**
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	TACOM/TARDEC
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)
	17026

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
Pub.in Proceedings of SPIE -- Volume 6561, The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**SAR**	**11**	

explain both human and animal visual classification performance significantly better than alternative hypotheses of feature-based and prototype-based processing[4,5].

Various researchers have worked to develop methods to forecast traversability based on estimates of geometrical properties inferred from non-contract sensors. A fuzzy-rule-based system[6,7] was developed to mimic human "high/medium/low" trafficability assessment based on measures of roughness, slope and distance between obstacles computed from stereo imagery. The system was targeted for planetary rover environments. A stereo color vision system, together with a single axis LADAR, was used to classify terrestrial terrain cover and detect obstacles[8]. It was noted that the color-based classification system could be made more robust by considering the texture of regions and the shape features of objects. A trafficability index has been defined[9] that is equal to the weighted sum of the slope and roughness, estimated from line-scanning laser rangefinder data. A procedure has been described for classifying terrain as impassible (NoGo)[10] if any of several properties were above a threshold: height variation, the surface normal orientation, and the presence of an elevation discontinuity (all estimated from LADAR imagery). A rule-based system for terrain classification from LADAR and color camera imagery has also been developed[11].

Appearance based approaches do not attempt to directly estimate geometrical properties and then infer traversability. Instead, they classify the terrain appearance, and then assign the associated trafficability vector measured while traversing similar terrain. The trafficability assessment is not restricted to computations of geometrical properties, but can also reflect micro-surface properties (e.g., friction, resistance, sinkage, etc.). In previous work, we used an exemplar-based approach to segment terrain into Go and NoGo regions[12].

In addition to mobile robot navigation, other applications could benefit from automatic image-based methods to segment and classify terrain, such as virtual reality simulated terrain, combat engineering planning, and land cover analysis for ecological studies. These applications address different scales, terrain features and parameters of interest and it is unlikely that any specific segmentation criteria would be suitable for all of them. Nonetheless, the applications have important similarities. In all cases, we implicitly assume that local areas with similar appearance should be grouped together in any segmentation and that they are likely to be representatives of the same terrain. For the purposes of this research, we assume that the segmented terrain regions do not have any a priori constraints on their geometric shape or global organization.

The approach is currently implemented as a software system designed to provide considerable flexibility in the choices of perspective transformation, resolution, scale, sampling and difference metric. In general, different choices will be appropriate for different applications.

2. TECHNICAL APPROACH

The algorithm is organized into two routines: one for offline training, which is based on fuzzy c-means clustering, and one for online learning, which applies segmentation and parameter identification to test images. At the end of the offline training, an exemplar bank is created that contains image and parameter identification data. During online learning, the exemplar bank is updated.

2.1 Training images and data

The user must provide a set of representative training images and associated vehicle-terrain interaction (VTI) parameters. Ideally, the training images would be drawn from the same distribution as the downstream application images. In practice, it may not be possible to ensure this. The effect that different conditions between the training image set and test/application image set, such as different terrain, foliage, season, lighting, and weather, has on segmentation and parameter identification performance is a question for empirical investigation. In principle, the images can be multi-spectral with an arbitrary number of planes.

For each training sequence, a corresponding VTI data set is required that contains sensor data for the relevant patches in the imagery. If one does not have range information, assumptions, such as "flat earth," must be made concerning the terrain in order to associate the sensor data from the vehicle to the image data that the vehicle has not

traversed yet. In this paper, the vehicle is assumed to be traveling in a straight line and we estimate the distance that each image patch is from the vehicle. For online learning and arbitrary vehicle motion, one would need to cache image patches and use a more complex method for correlating VTI parameters with image patch location. Examples of image and VTI data are shown in Figure 1.

2.2 Perspective transformation, resolution, scale and sampling

In some cases, a transformation from original camera perspective may be appropriate. In the camera image view, pixels represent the same angle (assuming lens distortion effects are minimal), but do not project onto equal areas of ground. This is problematic since terrain appearance changes with range and thus, would require multiple instances of the same terrain for training (at different ranges).

Assuming the elevation of the camera is large relative to the variation in ground elevation in the scene, the pseudo plan view projection can be used to create a new image in which each pixel corresponds to the same ground area. The pseudo plan view projection is good for areas where the variation in elevation is small relative to the elevation of the camera, but produces distortion when this is not the case. An alternative projection is to restrict analysis to horizontal sub-bands within the image. The band view does not distort vertical objects, but retains the perspective distortion of the original camera image for flat earth regions. A third alternative is to use a stereovision camera to measure range and warp the image accordingly, such that each image chip roughly corresponds to equal areas of ground.

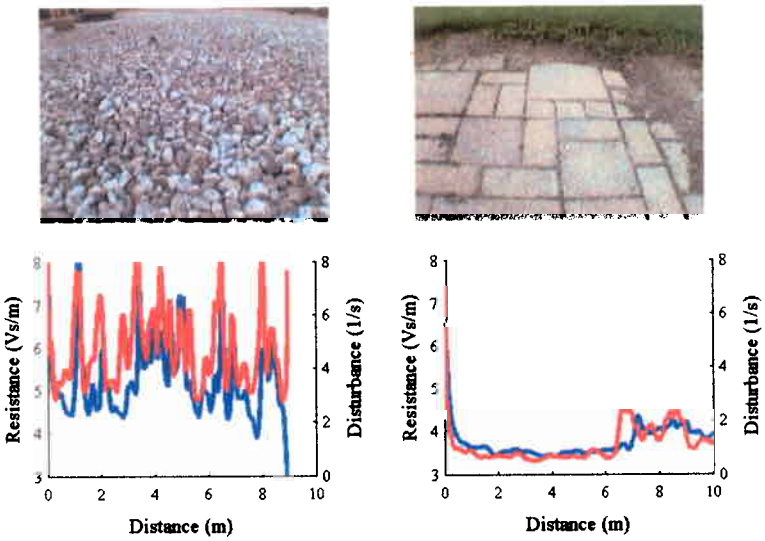

Fig. 1: Input training images and VTI parameters (resistance ($1/\alpha$) = blue, disturbance (ρ) = red).

The user must specify the analysis scale for terrain segmentation. The segmentation is based on exemplar image chips (square chips in the current software). The scale is the width of the exemplar chips. Membership in a terrain class is considered to be a bulk property of a local region, not a point-location property. The user must also specify the center-to-center spacing, or sampling distance.

2.3 Image space transformation

The purpose of the image space transformation is to amplify the importance of selected image properties. For example, the imagery can be transformed into a variety of color spaces. The importance of color could be strengthened or weakened by weighting different image planes. In addition to the RGB color coordinate system, we have experimented with the HSV (hue, saturation, value) and L*a*b* (luminance, red/green, yellow/blue) systems.

Another transformation option is to adjust the high spatial frequency content relative to low spatial frequency content by constructing a multi-resolution pyramid representation and then applying weights to the image planes. A common example is the Laplacian-of-Gaussian spatial bandpass pre-filtering, which is often used in stereovision processing. The space transformation could increase the dimensionality of the image space. Consider a monocular image input. The image could be processed through a bank of N spatial filters, such as edge and corner filters at different spatial scales and orientations, with each filter producing a single-plane output image.

2.4 Exemplar basis set

We use the fuzzy c-means (FCM) clustering algorithm to generate the initial exemplar basis set, since it is assumed that the system will be trained offline. In run-time operation, the online algorithm will characterize upcoming terrain, as well as generate new exemplars for unrecognized terrain.

The offline FCM algorithm processes all the images at the same time, which leads to an optimal segmentation of the images. The user chooses the number of clusters desired and the algorithm determines the cluster centers and the resulting cluster sizes. The closest image chips to the cluster centers are chosen as the exemplars and form the foundation of the exemplar bank.

Because the current online algorithm processes each image independently, it is naturally suboptimal, and therefore, additional information is considered when choosing exemplars. Here, each chip is compared to its neighbors within a specified radius to calculate the difference metric between it and each of its neighbors (the radius is a user input). The aggregate local difference between the chip and its neighbors is calculated as the weighted average of the mean and minimum differences (The weight is a user input. Weighting towards the minimum leads to a larger pool of exemplars, and weighting towards the mean leads to a smaller pool of exemplars). Chips similar to their neighbors are preferred over those that are different.

Based on previous work[13], we plan to explore the use of a decision tree algorithm for both the training and online portions of the system. Preliminary tests indicate that the decision tree algorithm provides comparable offline performance to the fuzzy c-means algorithm, while promising a method to incrementally add new information to the database without complete retraining.

2.5 Image chip difference metric

Image difference metrics remain an open issue in the evaluation of image compression schemes. While it is easy to measure the amount of compression and the encoding/decoding time, it is not clear how to measure the quality of the reconstructed image, i.e., its difference in appearance from the original. Different image characteristics are important depending on the image content, the questions at hand, and who is looking at the image.

Similarly, there is no obviously correct metric for measuring the difference between two images. Before the images are chopped into chips, they can be processed to balance the relevant image characteristics (see 2.3 Image Space Transformation). In principle, therefore, simple measures of the aggregate difference are all that are needed. Even so, there are many different ways to calculate the difference between two image chips. Some metrics are computed from the pixel-by-pixel difference between two chips, others are calculated from the difference in statistics computed from the individual chips.

2.6 Output illustration controls

The algorithm contains options to output different images to illustrate and provide insight into the processing:

- the pseudo plan view or camera band view perspective transformation of the image;
- the exemplar chips (at their location in the image) selected from the current image;
- the segmentation of the current image based on the current bank of exemplars;
- the VTI parameters for the image; and
- the image, color coded to the VTI parameter prediction for each chip.

There is no obvious and correct way to represent the different segments for purposes of visualization. Color-coding shows the different segments, but does not give much insight into the basis for the segmentation. The software illustrates the segmentation in a way that provides direct visual insight into the basis for the segmentation. To visualize the segmentation, the software replaces each image chip with the exemplar chip to which it is associated (image chips not associated with any exemplar appear black). When the sampling distance is less than the exemplar scale, the exemplars are blended in the reconstruction. The visualization image is the same size as the pseudo plan view or camera band view

perspective image, so it is easy to directly compare the two. By using the exemplar chips themselves, the visualization image shows what the exemplars look like, and which image chips they are associated with. Finally, comparing the visualization to the perspective image gives prima fascia evidence of the credibility of the segmentation. See Fig. 2 for reconstruction of the images in Fig. 1, which is based on a specific set of image features that does not include color. Hence, the lack of color matching between the reconstructions and original images in Fig. 1.

Fig 2: Reconstruction of training images from exemplars.

3. IMPLEMENTATION

3.1 Image processing

Image processing can be used to remove attributes of the imagery that can lead to misclassification, such as noise, color balance, and brightness. Automated features in cameras attempt to compensate for different lighting conditions and produce more life-like imagery. However, they are sometimes only partially successful, resulting in a time lag before compensation or applying the correction over the entire image when only a portion of the image needs correction. We were interested in applying a transform to the imagery such that consistent results would be obtained, irrespective of the lighting conditions. As an initial attempt at separating the luminance component from the color component, we tried the HSV (hue, saturation, value) color space. Although this resulted in some improvements over the RGB color space, the HSV system is unsatisfactory due to the cyclical nature of hue and the fact that HSV is far from perceptually uniform. This led to the implementation of an L*a*b* color space transform, where L^* refers to luminance and the a^* and b^* components encode the color information (red/green and yellow/blue color opponency, respectively). The transformation to L*a*b* is nonlinear, resulting in components that are closer to perceptually uniform. All the results depicted in this paper use the L*a*b* color space transformation.

Fig 3: Images showing fluctuating lighting and automated camera response correction.

As seen in Fig. 3, our image sequences also show evidence of spurious color effects, most likely due to automated features of the camera system. The top pair was separated by approximately 1/3 second and the images on the bottom were each separated by about 1/4 second. We are considering ways to alleviate this problem. In previous work[14], we tried having the system learn the color changes. For the current paper, we are not using color as a feature. However, one would like an unmanned vision system that is able to recognize terrain even in the presence of changing lighting conditions or color shifts, just like a human can.

We know that texture plays an important role in vision and even more so in the current work, where we are not concerned with object identification. We have explored two main measures of texture, the standard deviation and entropy. For the former, we created a texture image by computing the standard deviation over all patches of a given shape, centered on each pixel in the image. Because this also picked up the strong edges of objects and other texture

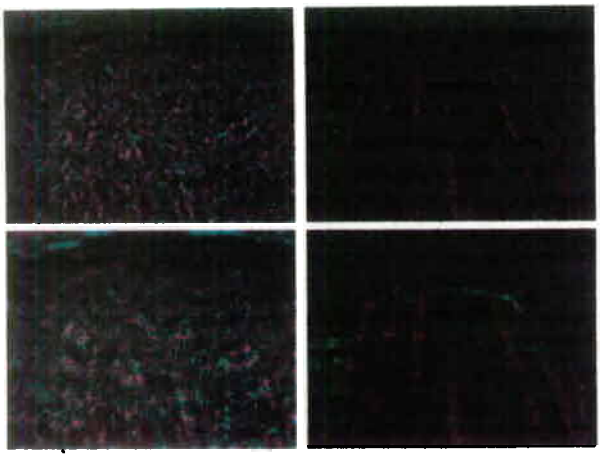

Fig 4: Texture images computed via standard deviation (top: 5 pixel filter, bottom: 11 pixel filter).

boundaries, we employed a Canny edge detector to find these strong edges and suppress them in the texture image. Fig. 4 shows examples of texture at two different resolutions for the images in Fig. 1. Each texture image has three planes, with the red and green planes containing the output of horizontal and vertical one-dimensional filters, respectively. The blue plane is computed from a two-dimensional standard deviation filter.

We also computed a texture measure based on entropy ($\Sigma\, x \log(x)$). Examples of this are shown in Fig. 5, where the different color planes correspond to different resolutions (5, 11, 17 pixels). In this case, we did not use Canny edge detection to suppress strong edges, since they appeared less pronounced. In the future, we intend to explore whether more specific shape filtering will provide improved recognition results.

For the current paper, based on a number of runs with a reduced data set, we found that the most effective features consisted of the mean luminance (L^*), the mean of the standard deviation texture images with a two-dimensional filter at resolutions 5 and 11, and the standard deviation of the entropy texture images at resolutions 5 and 11. It was found that the color information coded in a^* and b^* provided little to classification accuracy and hurt the results in most cases. The addition of the horizontal and vertical standard deviation filters also did not help the results significantly. The median of the image plane chips was also explored, but also added little to the classification accuracy. If an even smaller set of features was desired, the texture at resolution 11 pixels was more important than the resolution at 5 pixels. Eliminating the mean luminance caused only a small decrease in classification accuracy and would result in a feature vector computed entirely from texture.

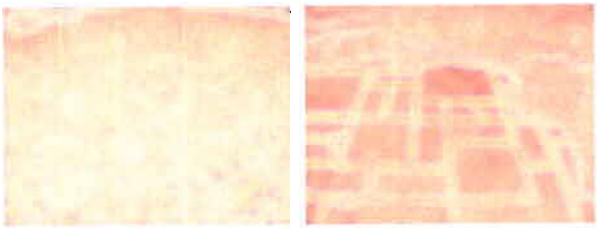

Fig 5: Texture images computed via entropy measure.

3.2 Fuzzy clustering

For the offline portion of the system, we have implemented a fuzzy c-means (FCM) clustering algorithm[15]. We are using the most basic form of FCM clustering with spherical clusters of the same size. Future work may look at non-spherical clusters of different sizes. As described earlier, our initial data set consisted of three features computed from each of the three L*a*b* image planes and twelve texture planes (nine multi-resolution standard deviation and three multi-resolution entropy). However, it was determined through experimentation that a five-element feature vector would suffice and could even be reduced to three or four elements with little loss of accuracy. Although our system includes the ability to transform the feature vector before presenting it to the FCM algorithm, the results in this paper have no additional transform applied. In the past we have taken the square root of the data so that the FCM algorithm, which computes a root-mean-square difference, could be compared to the results of the current online algorithm, which employs absolute differences.

The FCM algorithm provides a list of cluster centers and a matrix with the distance of each data point to each cluster. Since the cluster centers have no direct connection to the data, we move each cluster center to the location of the nearest exemplar in feature space and recompute the distances. From the resulting matrix, we can identify which exemplar (cluster center) should be assigned to each image chip in the data.

Since each chip has an associated set of VTI parameters, this gets naturally carried along with the corresponding exemplar. However, for the results in this paper, instead of using the VTI parameters for the particular exemplar, we

have averaged the parameters over all chips within the cluster and used the resulting values to tag each exemplar. There is also an option for using a weighted average, based on distance from the cluster center, where a distance of zero would yield a weight of one and a distance equal to the mean cluster distance would yield a weight of one-half. We modified existing computer code[16] for our implementation of the FCM algorithm.

3.3 Vehicle-terrain interactions

While the test vehicle, shown in Fig. 6, had a number of sensors, we are only using a subset in this study. All VTI measures that we are interested in have a dependence on vehicle speed and therefore, this is an important parameter to measure accurately. We currently use a wheel encoder attached to a fifth wheel trailing the vehicle to provide the speed of the vehicle, irrespective of track slip. Based on previous experiments[1], we assume that vehicle speed is linearly proportional to the motor voltage, $v = \alpha V$. Our measure of ground resistance is the proportionality constant α (m/Vs), with high α corresponding to low ground resistance and low α to high ground resistance. Ground resistance is more directly related to $1/\alpha$, as in Fig. 1.

Fig 6: Test vehicle.

The second VTI measure of interest is ground disturbance. We use the output of an accelerometer positioned over the front axle to collect data and we assume that disturbance is linearly proportional to speed, $D = \rho v$. The proportionality constant ρ is used as a measure of ground disturbance, with high ρ for large disturbance and low ρ for small disturbance. The disturbance D is determined by computing the standard deviation of the accelerometer output around the point of interest.

We use a "flat earth" assumption to determine the range to points in the images. By measuring the camera height off the ground and the distance from the front axle of the vehicle to the apparent top and bottom rows of the images, we can estimate the distance to all points in the images. The vehicle was commanded to travel in a straight line. By using the distance traveled, as measured by the fifth wheel, which was synchronized with the internal vehicle sensor data, offline we can tag each image chip with the appropriate VTI parameters. In more realistic operation, where the vehicle turns, where the terrain is not flat and where there are objects in front of the vehicle, more complex processing and data handling procedures will be required.

4. RESULTS

4.1 Data

The data collection that forms the basis for the results in this paper consists of 34 runs over different types of terrain, such as concrete, asphalt, dirt, grass, bricks, gravel, sand and rocks. Each run was between 15-25 seconds, with periods at the beginning and end where the vehicle was motionless; vehicle motion occurred for between 10-15 seconds. The vehicle-terrain interaction (VTI) parameters that we are currently exploring are for quasi-steady state conditions and so we are not considering effects due to acceleration or deceleration. For each run over a given terrain segment, we also had another run in the opposite direction.

For this paper, as in a previous work[14], we chose five runs to train the system and used the companion runs in the opposite direction for testing. Terrain 1 consisted of rocks, terrain 2 was brick pavers and grass, terrain 3 was cement and grass, terrain 4 was asphalt and cement, and terrain 5 was rough sand.

We collected distance data from the fifth wheel encoders, acceleration data from the accelerometer over the front axle, voltage data from the motor, and image data from the onboard camera. The VTI parameters of interest are the ground resistance and ground disturbance.

4.2 Data processing

We smoothed the voltage data with a Hamming-like filter of length 0.5 s. The acceleration data was converted to disturbance by a Hamming-like standard deviation filter of length 0.5 s. We used a heuristic algorithm to remove spikes from the wheel encoder data, which was then filtered by a derivative filter of length 0.5 s to produce vehicle speed. Vehicle speed divided by voltage yielded the terrain resistance parameter. Disturbance divided by vehicle speed yielded the ground disturbance parameter.

Fig 7: Measured vehicle-terrain interaction parameters (α = center and ρ = right).

We extracted every fifth frame in the center part of each of the training sequences, resulting in 325 images. We used every frame in the center part of each of the test sequences, resulting in 1575 images. Each frame in the video was 320x240. We cropped the images to 200x160 by taking 60 pixels off each side and 80 pixels off the bottom. We chose an image chip size of 24x24, which resulted in 48 chips per frame (except for frames that included terrain that would not be traversed by the vehicle). The resulting training set had 15,520 samples and the test set had 75,560 samples. A five-element feature vector was computed for each of the image chip samples in the training and test sets.

4.3 Training and test results

We chose to use 40 clusters for this test, although test error results were relatively flat beyond 20 clusters. This resulted in a training error of 6.2% and 34.2% for the ground resistance and ground disturbance predictions, respectively. The error on the test set was 9.7% and 47.0% for the ground resistance and ground disturbance, respectively. The error was computed as the absolute difference between prediction and measurement divided by the average of the two.

We implemented a color-coding scheme to graphically illustrate the predicted VTI measures using the image data. The color red corresponds to the least desirable end of the parameters (0.2 for α and 2.7 for ρ), while green corresponds to the most desirable end of the parameters (0.3 for α and 0.7 for ρ). Quantities outside that range were truncated. Image chips that were determined to be too far from any exemplar were color-coded blue, with those having desirable VTI parameter values being cyan-hued and those that were least desirable were magenta-hued. The unknown chips were included in the error computations.

Figure 7 shows an example of extrapolating the measured values for the VTI parameters to specific image locations via the "flat earth" assumption, which are input to the FCM algorithm for the training image on the left. The center image contains the terrain resistance parameter and the right image contains the terrain roughness parameter. This figure illustrates one of the places where errors can enter the process: synchronizing the onboard data with the image data. Errors can enter due to faulty range estimations, but here the terrain is fairly flat and the problem is due to distances being computed from the front axle of the vehicle. The terrain resistance is maximized when both the front and rear tracks are on the terrain, while the disturbance manifests when the front track encounters the boundary. This lag between terrain roughness and resistance can also be seen in the right plot of Fig. 1.

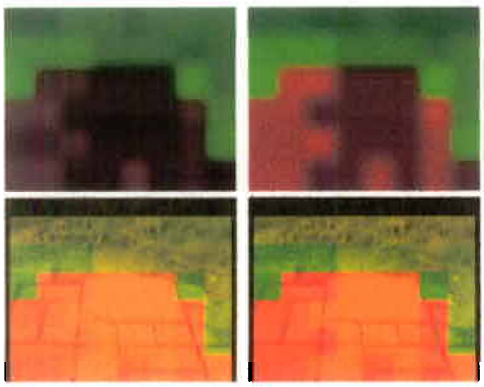

Fig 8: Predicted vehicle-terrain interaction parameters (top) and color-coded images (bottom) (α = left and ρ = right).

Figure 8 shows a prediction for the VTI parameters for the training image in Fig. 7 along with the color-coding scheme. The terrain resistance images are on the left and the terrain disturbance images are on the right. Note that the predictions are not very

accurate, they show the pavers being rough with high resistance and the grass being smooth with low resistance. The error can be traced with the aid of the reconstruction image on the right side of Fig. 2, where a poor choice for exemplars is shown for the image chips. In this case, our training database did not contain enough samples of brick pavers in different lighting conditions. In fact, the exemplar bank contained only two exemplars from the paver portion of the dataset.

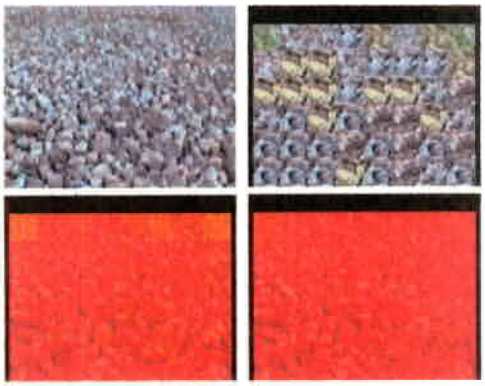

Fig. 9: Test image reconstruction (top right) and VTI predictions (bottom, α = left and ρ = right).

Figures 9-11 show further examples of test results. Fig. 9 contains data from the vehicle being run over rough rocks. Here the results are generally good and correspond to expectations for both the terrain resistance and the terrain roughness. Because of the high variability in the rock and sand images, they tended to dominate the exemplar bank, with 11 exemplars each out of 40. This is reflected in the reconstruction, which is reasonably close to the original. Exemplars derived from cement portions of the images were next with 8 exemplars and those with grass had 7 exemplars. Note the color variations within the reconstruction image in Fig. 9, which is due to an absence of a color element in the feature vector. In previous work, where color was a part of the feature vector, the reconstruction was more accurate in regards to color, but the overall VTI parameter prediction was less accurate.

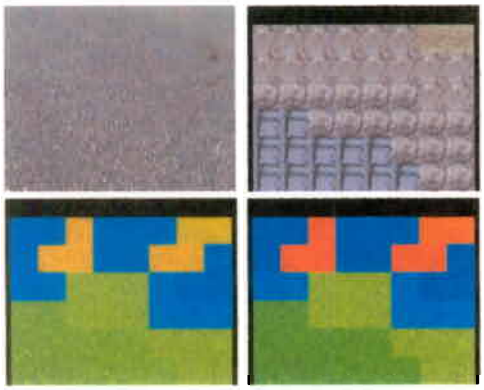

Fig. 10: Test image reconstruction (top right) and VTI predictions (bottom, α = left and ρ = right).

Figure 10 shows results for an image that contains asphalt. The system provided erroneous results for this whole image sequence since the exemplar bank only contained one exemplar derived from asphalt, likely due to a shortage of asphalt images in the training sequences. As the reconstruction image shows, the system tended to pick sand exemplars as the best match, which resulted in modest agreement with the terrain resistance parameter and poor agreement for the terrain disturbance parameter. The blue areas indicate that the image patches were too far from any exemplar, but the closest exemplar was one that was average in regards to both ground resistance and roughness. The reconstruction image in fact indicates that the top part of the resistance image would have been yellow and the top part of the roughness image would have been orange.

Figure 11 shows some interesting results for the cement images, which generally had good agreement in the cement portions, but the cracks were mistaken for pavers, which still resulted in accurate predictions since pavers and cement have the same VTI characteristics. In past work[14], with less emphasis on texture, the cracks were often mistaken for rocks, resulting in poor agreement in those specific portions.

The pavers/grass image sequence caused the most error in the training set, while the asphalt sequence caused the most error in the test set.

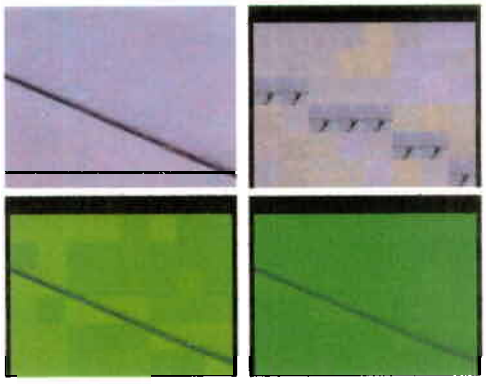

Fig. 11: Test image reconstruction (top right) and VTI predictions (bottom, α = left and ρ = right).

5. FINDINGS AND OBSERVATIONS

This paper has demonstrated an approach to image-based terrain segmentation using exemplars, as applied to vehicle-terrain interaction (VTI) prediction. Exemplars provide a simple way to represent the characteristic color/luminance and spatial patterns of terrain. Since the exemplars are drawn from training images in such a

way as to span the appearance of the training images, they are well suited to represent the variations of appearance without an a priori model of terrain appearance. The software system, as presented, allows for considerable flexibility to specify the perspective transformation, image space transformation, scale, resolution, sampling density, and image difference metric. Empirical research is needed to tune these options for specific applications.

Preliminary results indicate the approach has potential to segment terrain in a manner that is consistent with subjective perception. The segmentation appears to provide some robustness over changes in lighting, specific terrain, and automatic camera gain and contrast adjustments, but still needs some additional work. There are a number of other areas where additional work is needed. Among the most important is devising a method for determining range. The current "flat earth" assumption is not viable for real-world application. Solutions include using internal sensors to measure pitch and roll and then correct for them, although this just provides a correction for the "flat earth" assumption. More costly methods would use a stereo camera system or laser range finder. The intent of this system is not to characterize all objects in view of the vehicle and so a method to eliminate non-traversable obstacles from the imagery would be useful. This is obviously the same as an obstacle detection system and would require the same level of sophistication.

There are a number of other subjects to explore in the current system, including incorporating more complex cluster structures, although that would also require more sophisticated processing for the online algorithm. Exploring shape filtering and additional multi-resolution processing could yield improvements to the clustering, without much change to the existing architecture. Implementing bandpass multi-resolution techniques, such as wavelets or Gaussian-Laplacian pyramids may also be fruitful. Methods for tracking terrain segments could also prove to be valuable. We need to explore better procedures for selecting the training data. The non-selective method currently employed results in a training set that is over-weighted in some terrain types and under-weighted in others. We also need to explore methods to further compensate for automatic gain and color distortions of the current camera system.

We are also beginning to explore the use of decision trees to replace both the fuzzy c-means (FCM) clustering for offline classification and the heuristic online clustering algorithm. Similar classification accuracy to FCM has been found in preliminary tests of the decision tree algorithms, with faster training times. Future work includes investigating the ability to easily add new exemplars to the data bank without retraining the entire system, something difficult to achieve with FCM. While the heuristic online algorithm worked reasonably well, it was less accurate than FCM clustering and generated substantially more exemplars.

The system as a whole shows promise and we intend to explore further refinements in order to provide better results. The visualization tools that have been developed for this project have been very valuable in determining where the system performs correctly and where it does not and will greatly aid with upcoming enhancements.

REFERENCES

1. R.E. Karlsen, J.L. Overholt and G. Witus, "Run-time assessment of vehicle-terrain interactions," Proc. 24th Army Science Conference, Orlando, FL (2004).
2. S. Chase and E.G. Heinemann, "Exemplar memory and discrimination," *Avian Visual Cognition*, R.G. Cook, Ed., (2001).
3. D. Medin and M. Schaffer, "Context theory of classification learning", *Psychological Review* **85**(3), 207-238 (1978).
4. R.M. Nosofsky, "Tests of an exemplar model for relating perceptual classification and recognition memory," *J. Experimental Psychology: Human Perceptual Performance* **17**(1), 3-27 (1991).
5. C.W. Werner and G. Rehkamper, "Categorization of multidimensional geometrical figures by chickens (Gallus gaullus f. domestica): fit of basic assumptions from exemplar, feature and prototype theory," *Animal Cognition* **4**, 37-48 (2001).
6. A. Howard, E. Tunstel, D. Edwards and A. Carlson, "Enhancing fuzzy robot navigation systems by mimicking human visual perception of natural terrain traversability," Joint 9th IFSA World Congress and 20th NAFIPS Int. Conf., Vancouver, B.C., Canada, 7-12, July 2001.
7. A. Howard and H. Seraji, "Vision-based terrain characterization and traversability assessment," *J. Robotic Systems* **18**(10), 577-587 (2001).

8. R. Manduchi, A. Castano, A. Talukder and L. Matthies, "Obstacle detection and terrain classification for autonomous off-road navigation," *Autonomous Robots* **18**, 81-102 (2005).
9. C. Ye and J. Borenstein, "A method for mobile robot navigation on rough terrain," Proc. IEEE Int. Conf. Robotics and Automation, 3863-3869 (2004).
10. D. Langer, J.K. Rosenblatt and M. Hebert, "A behavior-based system for off-road navigation," IEEE Trans. Robotics and Automation **10**(6), 776-783 (1994).
11. A. Sarwal, D. Simon and V. Rajagopalan, "Terrain classification, *Unmanned Ground Vehicle Technology V*, SPIE Proc. **5083**, 156-163 (2003).
12. R.E. Karlsen and G. Witus, "Vision-based terrain learning," *Unmanned Ground Vehicle Technology VIII*, SPIE Proc. **6230**, 33-42 (2006).
13. G. Witus, "System to build fuzzy logic models from databases and application to multi-sensor data," *Data Mining and Knowledge Discovery: Theory, Tools, and Technology III*, SPIE Proc. **4384**, 171-79 (2001).
14. R.E. Karlsen and G. Witus, "Image understanding for robot navigation," Proc. 25th Army Science Conference, Orlando, FL (2006).
15. F. Hoppner, F. Klawonn, R. Kruse and T. Runkler, *Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition*, John Wiley & Sons, (1999).
16. B. Balasko, J. Abonyi and B. Feil, "Fuzzy clustering and data analysis toolbox" (www.fmt.vein.hu/softcomp/fclusttoolbox).