Terrain Typing for Real Robots

Ian Lane Davis[†], Alonzo Kelly[†], Anthony Stentz[†], Larry Matthies[††]
[†]The Robotics Institute, School of Computer Science, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh PA 15213, USA
Phone (+1)412-268-6587, Fax (+1)412-268-5895, akiam+@cmu.edu
[††]Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Drive, Pasadena, CA 91109, USA
Phone (+1)818-354-3722, Fax (+1)818-354-8172, lhm@robotics.jpl.nasa.gov

Abstract

Many robotics tasks require an ability to determine quickly the nature of the terrain surrounding the robot. While much attention has been given to the general problem of terrain typing, the problem of effective *real-time* terrain typing remains open. For robot missions such as construction site work, military reconnaissance, hazardous waste removal, and planetary exploration this problem must be addressed. In particular, for cross country navigation with a wheeled vehicle, the robot needs to know where the vegetation is and where the rigid obstacles are because frequently the optimal, if not the only, path will pass through vegetation. Our groups have independently researched the problem of finding vegetation in a scene, and have developed systems tuned to the specific demands of real-time terrain typing for robots. This paper looks at three classifiers of increasing dimensionality and describes their applicability to different aspects of the terrain typing problem.

1. Background

Our research groups are studying unmanned ground navigation. At CMU (Carnegie Mellon University), we implement our systems on the NavLab II military ambulance, also known as the HMMWV (High Mobility Multi-Wheeled Vehicle). At JPL (Jet Propulsion Laboratory) there are several robots (including a HMMWV) which benefit from our research. We frequently use range images for navigation on our robots, and in range images the vegetation looks just like the hills, ridges, rocks, and mounds which we must avoid. Usually we just avoid everything for simplicity's sake. However, this approach limits the reachable regions in natural environments. Additionally, this approach makes all of our navigation less efficient, so we are developing systems to classify every pixel in a color video image as vegetation or not vegetation.

Terrain typing is not unstudied, although real-time terrain typing has often been neglected. For a real robot, speed of processing is paramount: if a technique for terrain typing is a perfect classifier for all terrain, but can't get the results before the robot next has to decide on a direction, that technique is useless. The trade-off between accuracy and speed can be cheated, though, by matching as well as you can a technique to the simplest sufficient terrain typing problem for your navigation task. In other words, don't solve a problem that's any harder than you must.

2. Our Research Goals

Towards this end, we are exploring several problems and several techniques. Our two questions are "How much data do we need to perform the terrain typing at hand?", and "Given the right data, what technique will suffice for classification?".

The simplest sense space we've used for classification is RG-space (Red-Green space). For this task, we are given an image, and at each pixel we decide whether the pixel is vegetation or not based on its red and green components (which have shown to be effective in some scenarios). The next more complicated sense space is RGB. This problem is the same, but we base our decisions on the pixel's position in the three dimensional red, green, & blue coordinates. Finally, we perform classification in "RGB retina-space" based not just on the one pixel in question, but on the qualities of a small retina about that pixel; this allows context and texture information to be available.

The techniques we use range from linear classifiers to classifiers capable of highly non-linear classification. This research in the paper applies a Fisher Linear Discriminant [4] to the RG problem, and Backpropagation Neural Networks [10] to the RG, RGB, and RGB-retina problems. For all of our techniques, we start with a color video image, and generate an equal sized image in which the intensity at each pixel is the vegetation classification of the corresponding pixel from the input color image.

3. The Approach: Fisher Linear Discriminant

Our simplest approach for generating the classification image is to use a linear decision surface. The idea behind a linear decision surface is simple: we have data in a given space of dimension N belonging to classes A & B, and we perform a classification by drawing a surface of dimension N-1 and calling everything on one side A and everything on the other side B. The trick is to find the correct surface.

The Fisher Linear Discriminant (FLD) does this by finding the line in N-space which maximizes the ratio of "between-class scatter" to "within-class scatter" in the projections of classes A & B onto that line [4]. Once we have this line, it is a simple matter to find a classification point (everything on one side is class A and everything on the other is class B) that maximizes the correct classifications of sets A & B. Given the Fisher line and the classification point, the N-1 dimensional decision surface is simply the surface through that point perpendicular to that line in all dimensions.

The advantage of a linear decision surface is that the computation needed for classification in high dimensions is very low. The necessary calculations can also be easily implemented with special purpose hardware.

4. The Approach: IVY

The neural network system we use for finding vegetation called IVY (Instant Vegetation Yielder). IVY's role is to process an image directly after digitization. The output is a new image, or an overlay on the raw image, which has intensity values whose range denotes the degree of vegetation at each pixel. Every pixel whose intensity in the output image surpasses a certain threshold is called vegetation. IVY could be used to do more complicated terrain typing, too, but this paper assumes a single division of vegetation or non-vegetation.

IVY uses a monolithic neural network approach to classification. The paradigm is referred to as the *operator architecture* [2] since we use the neural network much as we would any low level computer vision operator (such as an edge detector). IVY uses a simple 3-layer backpropagation neural network [10]. The inputs to IVY are three small retinas, a red, a green, and a blue one. In this paper we look at IVY networks where the retinas are 1x1 and 7x7. The input level units are activated with appropriately scaled pixel values from a square retina centered around a particular pixel in question. The single output of IVY represents whether or not the center pixel is to be classified as vegetation; the value of this output ranges from -1.0 (not vegetation) to 1.0 (vegetation). The size of the retina can be adjusted to accommodate the amount of texture or averaging you wish to have affect the classification (See Figure 1, on page 2 for a diagram of IVY.

The neural network approach is derivative primarily from one of the techniques used by Marra, Dunlay, and Mathis: the "Image Based Neural Network" [7]. In that work, Marra, et al, tried to classify terrain as one of six things: brush, dirt, grass, hill, road, or sky. The Image Based Neural Network technique had some success, but we hope to improve on those results by focussing more closely on classifying just vegetation. By having such a complicated function to learn as they had, it is not surprising that their networks had difficulty developing internal texture representations as there must have been significant interference (crosstalk) caused when the training examples from different terrains were used. An additional benefit of a narrower problem is that we should be able a simpler neural networks; this will give us a chance to achieve high-resolution real-time classification (Marra, et al, could turn a 512 by 512 raw image into a 32 x 32 classification image in more than 2 seconds on a very fast and expensive parallel computer [using > 50 hidden units]).

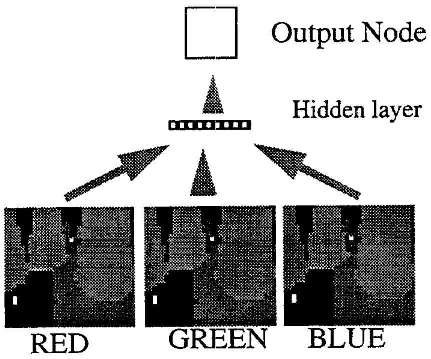


Figure 1. IVY monolithic operator network

5. Training the FLD and IVY

Training either the FLD or IVY is straightforward. We hand-label a large number of pixels in several images at whatever resolution we desire the classification to occur. With a sophisticated "paint" program which allows color range matching as well as the selection of polygonal regions and hand drawn regions, we can do this quite painlessly. For training the FLD we look at every pixel that has been hand classified and store the statistics (histogram) for both classes, vegetation and non-vegetation. We then run the FLD algorithm on the class data. For the IVY neural network, we randomly select a training set with as few as 500 or as many as 5000 training exemplars (for simple mappings, 500 exemplars can be sufficient to span the set of possible inputs). Each exemplar is a an ordered pair (x, y) in which x is a retina about a random point in one of the training images and y is the label for the pixel at that point in the hand-labelled image.

6. Convexity, Linearity, and Complexity

The answer to the questions of which input space to use and which technique to use hinges on the complexity of the classification task. If the sets to be classified are convex and disjoint, a linear decision surface is all that is needed. For some tasks, though, the mapping from image space to terrain space can be nonlinear and complicated. A simple example is that we often encounter very green vegetation, as well as red and orange vegetation. We also see brown dirt roads. To linearly classify all of the vegetation or to classify it with a single color neighborhood match would mean that we would incorrectly classify the road. For this reason, some tasks demand the use of a nonlinear classifier.

6.1. Averaging and Texture

In even more complex tasks, we may want to use retinas larger than one pixel to achieve classification based in part on averaging or texture. Neural network techniques allow us to create these more complex mappings easily. Averaging is important for complete image classification because we get shadows both on vegetation and on rocks and other obstacles. If we look at individual pixels only, we will get very dark pixels that cannot be properly classified. If, however, we look at the surrounding pixels, too, and all of them are vegetation-colored or vegetation-textured, then we can classify a pixel as vegetation.

7. RG Input Space Experiments

Our first set of experiments involved classification based only on the red and green components of each pixel in the input image. Although we could guess from the start that there will be scenarios in which these two bands will be insufficient for terrain typing, we were surprised to find some cases in which they provided enough data to do reasonable classification. More importantly, using two bands affords us the luxury of virtually instantaneous classification of pixels: with either the FLD or IVY classification scheme, we need only generate a 256 by 256 array (with the indices corresponding to Red & Green pixel components) storing the classification for each RG pair. When classifying in real time, we simply look up the classification.

The first set of images and hand-classifications used to test both techniques were generated by JPL. For the FLD, we used a training set of 531434 pixels taken from five 480x512 color images. Of these, 251413 had been hand labelled as vegetation and 490776 had been labelled as non-vegetation. For the neural network, we used a simple IVY network with a 1x1 retina for red, green, & blue (but disabling the connections from the blue retina to the rest of the network). Our training set consisted of 5000 exemplars from the set of 531434 known pixels. Exactly half of the IVY training set was vegetation and half was non-vegetation. IVY networks with every number of hidden units from 1 to 20 were trained; the best one had 8 hidden units (for a 2 to 8 to 1 feed-forward network).

Both techniques were test on all 531434 known pixels. Listed below are the percent of correct classifications over the known pixels in each of the five images for both techniques (+ is % right of known vegetation, - is % right of known non-vegetation, T is % right of total known pixels per image):

Table 1: RG Input Space

		FLD			*IVY*	
Image	+	-	*T*	+	-	*T*
1	87	77	82.9	90	71	80.3
2	69	87	77.0	72	86	78.3
3	84	94	91.8	88	92	91.4
4	97	90	93.8	98	88	93.7
5	97	95	96.4	98	95	96.8
Total	88.6	89.1	88.9	90.6	87.1	88.9

Both the FLD technique and the IVY network performed equally well on the given data. The FLD correctly classified 472769 pixels, and the IVY network correctly classified 472157 pixels.

8. RGB Experiment

While we achieved a decent level of ultra-fast terrain typing with both the FLD and the IVY network in the RG-space experiments, there seems to be an upper bound on how well we could do with that data using only the red and green bands. We next tried a full RGB-IVY network on the same images. In fact, we used the exact same training set as used for the RG-IVY network.

Again, 20 networks were trained with all numbers of hidden units from 1 to 20. The one with 6 hidden units provides the results listed below:

Table 2: RGB-IVY

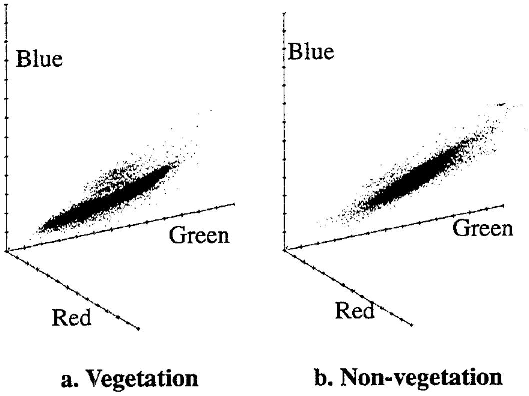
		RGB-IVY	
Image	+	-	*Total*
1	96	84	90.2
2	88	90	89.0
3	90	93	92.7
4	96	90	93.3
5	96	95	95.6
Total	94.0	90.9	92.4

The IVY network using the RGB data performed better than either of the techniques on just the RG Input space. Red and green data is simply not enough to do a 100% classification; there is overlap between the vegetation and non-vegetation sets. However, once we jump to three dimensions (RGB) we lose the instant look-up tables of 2D RG space.

We did one small side experiment to strengthen the position that sometimes we need the additional input dimensions and non-linearity. We used a single image taken from CMU in Autumn to illustrate the point. In this image, there is dark green grass, reddish grass, and tan, strawlike grass. The image had poor lighting and we expected none of the classifiers to perform especially well, and we were right, but there was a distinct difference in performance. The FLD got 65% correct classifications, the RG-IVY got 67% correct, and the RGB-IVY got 70% correct.

9. Second Set of Testing Data

We also tested the RGB IVY network on some images gathered at CMU that had a slightly more complicated scene (in terms of vegetation coloration). In the set of images used in experiments detailed in previous sections, all of the vegetation was more or less green; in this second set, which was gathered in Autumn, there were some red and orange plants. We used a series of 5 high resolution images (640x480) to generate 500 training exemplars and 500 initial testing exemplars. Again, each pixel is defined as three integers from 0 to 255. All of the images used were digitized live from a CCD camera on the same day. It is worth taking a close look at the training data.



a. Vegetation b. Non-vegetation

Figure 2. Plot of second training set for RGB

The dots on graph "a" represent the set in RGB space that we wish to map to "vegetation" in terrain space. The graphed set is a subset (randomly sampled) of all of the points labelled "vegetation" in the training images. Notice that there at least two major clusters, one "above" the other with reference to the Blue axis. The dots on graph "b" represent the set in RGB space that we wish to map to "non-vegetation".

The pixels in the training set for both positive and negative exemplars of vegetation do not cover the RGB space, which means that the *ideal* function which we wish to approximate is not even defined on those other areas of RGB space. Performance on other days and lighting conditions is not guaranteed with such a training set. For more generality, the training set must cover the desired subset of RGB space.

Furthermore, there is some overlap in the positive and negative sets. This is due largely to shadows and specular effects, but also occurs due to natural overlap in the two *ideal* sets of vegetation and non-vegetation.

The mapping is fairly easy to learn and convergence occurs quickly. An IVY network was trained with each number of hidden units from 1 to 15. In these trials and in others, the function was learned reasonably well by each network. One hidden unit usually seemed to be not good enough, but anything from 2 or 3 up was good. With higher numbers of hidden units, convergence took more time, as one would expect, but the long term results were not better. In even the best 1 pixel IVY networks the average classification error on the given test set of 500 exemplars was more than 0.3. The error for a given exemplar (input + known output) is the difference between the real valued output which ranges from -1.0 to 1.0 and the known terrain classification (with vegetation being 1.0 and non-vegetation being -1.0). Thus, the maximum error possible was 2.0 and average error with random weights would be expected to be 1.0. The average classification error results both from misclassifications and weaker classifications (the network will never learn to output exactly 1.0 or -1.0, and the amount of "slop" is related to how well the function was learned).

10. Larger Retina Experiment

In order to take advantage of averaging and texture in the second set of test images, we also trained an IVY network that had a 7 x 7 RGB pixel retina (147 input units). This network's architecture is very similar to that used in [7], though the mapping it is to learn is more focussed - which is an important distinction. The same hand-labelled high resolution images were used to generate the training and test sets for this IVY network as for the simpler one described in the previous section. Again, 500 exemplars were used for a training set and 500 were used as a test set.

Training was slower in real-time for these networks since for a given number of hidden units, the 7 x 7 input IVY 1 network had 49 times as many connections as a 1 pixel IVY 1 network. However, the learning "flattened out" in fewer epochs (passes through the entire training set. This is partially because there was significantly less overlap between the sets of vegetation and non-vegetation, both in the *ideal* mapping and in the *actual* training set. Again we used a network with each number of hidden units from 1 to 15. The average error in the output over 500 test cases (not seen in training) was as low as 0.15 (out of a maximum of 2.0).

We also used this network on the data from the first set of images (from JPL). The 7 x7 input IVY is not fast enough (done in software on a serial computer) to do terrain typing of each pixel of an image in real-time, but we can compare the average error in the outputs over test sets of 500 unseen retinas between the RG-IVY, the RGB-IVY, and the 7x7-IVY. The results can be seen in Table 3, "RG v RGB v 7x7 IVYs," on page 5.

Table 3: RG v RGB v 7x7 IVYs

	RG	*RGB*	*7x7*
Avg Error	0.284	0.220	0.163

11. Conclusions

The results of our experiments highlight several important aspects of the terrain typing problem. First, the sense space you use can limit your performance no matter what technique you use. On the other hand, although a sufficiently complex sense space can let you correctly classify more pixels, you may not be able to classify a whole image in time for it to be of use to a robot.

Several strategies fall out of these results. If speed is paramount, either a low input space such as RG-space or a simple classifier will suffice. Even the RG-space allowed almost 90% correct classification. For scenarios in which vegetation and non-vegetation come in large patches in an image, this could be entirely sufficient. Also, the FLD promises to perform well in higher dimensions (such as RGB) provided the sets of vegetation and non-vegetation in the higher dimensional space remain relatively disjoint and convex.

The IVY neural networks are appropriate when the input space is of high dimension or when the different sets to classify are not as nice as the ones we used here. The first benefit was seen when we were able to add texture and averaging data, which can be essential when correctness of classification is needed. The second benefit, although not the focus of these tests, would come about in the case with red and green vegetation and brown roads: a properly trained IVY network could classify many different colors of vegetation.

12. Future Directions

There are three directions we wish to explore in the immediate future as a result of this work:

First, we believe it is worthwhile to continue investigating simple classifiers such as the FLD and the simpler IVY networks. The FLD promises to perform well and quickly in higher dimensions. Furthermore, the shortcoming of the simple techniques is that they fall apart when the sets to be classified are not continuous or convex. However, if, as often happens in the real world, the sensing space has sets that are "piecewise" continuous or convex, we can simulate higher order behavior by combining several simpler classifiers. Two FLDs could be used to pick out green vegetation and red vegetation respectively, and the union of their outputs would be correct for the appropriate sensing environment.

Second, we will continue to develop classifiers such as neural networks that take advantage of richer sensing spaces (texture, averaging, etc.). We are developing a technique using modular neural networks that allows us to pre-select features we know to be important in order to supplement the neural network's learning[3]. In this way, we can guarantee the use of texture information in the classification process. Also, by carefully choosing the size of the retina in an IVY network, we can get texture information and speed of classification at the same time. Another possible speed up is to perform classification at several layers of resolution from low resolution to high resolution (where necessary).

Finally, the RGB images we use are not necessarily the most useful images for sensing vegetation. Currently experiments are being done with the FLD and IVY techniques on images that have a Near Infra-Red (NIR) band in addition to the Red, Green, and Blue bands[6]. In NIR-space, or Red-NIR space, or RGB-NIR space, the vegetation set is sometimes completely disjoint from (and not especially close to) the non-vegetation set. A well-chosen input space allows use to use faster and simpler classifiers.

13. Acknowledgments

This work has been supported in part by ARPA, a National Science Foundation Graduate Research Fellowship (for I. L. Davis), by DACA76-89-C0014, Topographic Engineering Center, Perception for Outdoor Navigation, and DAAE07-90-C-R059, TACOM, CMU Autonomous Ground Vehicle Extension.

14. References

[1] M. Daily, et al., "Autonomous Cross Country Navigation with the ALV," *Proceedings of the 1988 IEEE International Conference on Robotics and Automation*: 718-726, 1988.

[2] I. L. Davis and M. W. Siegel, "Automated Nondestructive Inspector of Aging Aircraft," *International Symposium on Measurement Technology and Intelligent Instruments*, Huazhong University of Science and Technology, Wuhan, Hubei Province, People's Republic of China, October 1993.

[3] I. L. Davis and M. W. Siegel, "Visual Guidance Algorithms for the Automated Nondestructive Inspector of Aging Aircraft," *SPIE Conference on Nondestructive Inspection*, San Diego, July 1993.

[4] R. Duda and P. Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons, Inc., New York, 1973. pp 114-118.

[5] A. Kelly, *A Partial Analysis of the High Speed Cross Country Navigation Problem*, Carnegie-Mellon University Ph. D. Thesis Proposal, 1993.

[6] L. Matthies, A. Kelly, T. Litwin, "Obstacle Detection for Unmanned Ground Vehicles: A Progress Report," to appear in the *Proceedings of the IEEE International Symposium on Intelligent Vehicles*, 1995.

[7] M. Marra, R. T. Dunlay, and D. Mathis, "Terrain Classification Using Texture for the ALV," *Proceedings of the SPIE Conference on Mobile Robots*, 1992.

[8] D. Pomerleau, *Neural Network Perception for Mobile Robot Guidance*, Ph.D. Dissertation, Carnegie-Mellon University Technical Report CMU-CS-92-115, 1992.

[9] D. Pomerleau, "Neural network-based vision processing for autonomous robot guidance," *Proceedings of SPIE Conference on Aerospace Sensing*, Orlando, FL, 1991.

[10]D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning Internal Representations by Error Propagation," *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, D. E. Rumelhart and J. L. McClelland, Ed. MIT Press, 1986.

[11]A. Stentz, "Optimal and Efficient Path Planning for Unknown and Dynamic Environments," Carnegie Mellon University Technical Report, CMU-RI-TR-93-20, 1993.

[12]W. A. Wright, "Contextual Road Finding With A Neural Network," Technical Report, Sowerby Research Centre, Advanced Information Processing Department, British Aerospace, 1989.