Neural Networks For Real-Time Terrain Typing

Ian Lane Davis

CMU-RI-TR-95-06

The Robotics Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213

January, 1995

This work has been supported in part by a National Science Foundation Graduate Research Fellowship. The equipment used is provided in part by DACA76-89-C0014, Topographic Engineering Center, Perception for Outdoor Navigation and DAAE07-90-C-R059, TACOM, CMU Autonomous Ground Vehicle Extension.

The views and conclusions expressed in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. government.

Neural Networks For Real-Time Terrain Typing

Ian Lane Davis

CMU-RI-TR-95-06

Abstract

Many robotics tasks require an ability to determine quickly the nature of the terrain surrounding the robot. In cross country navigation in particular, the robot needs to know where the vegetation is and where the hard obstacles are. I have developed a general system which has successfully allowed real-time terrain typing in the NavLab II autonomous vehicle. This system and training paradigm are based on standard neural network technology and allow the robot to learn arbitrary non-linear mappings from color and texture space to terrain space.

1.0 Introduction

Much of the work in autonomous navigation has focussed on the idea of finding and avoiding obstacles [Kelly93] and the complementary problem of finding empty and level ground over which to maneuver [Pomerleau91]. This is appropriate for some navigation tasks, such as road following, and for some environments such as a lifeless planetary surface. But for many environments and navigations tasks, this simple concept of an obstacle is insufficient. For example, in an off-road navigation task almost anywhere on our very own planet, there are at least two kinds of obstacles: *hard-obstacles* such as rocks and *soft-obstacles* such as vegetation. Sometimes we can make a robot navigate by avoiding all obstacles, but there are times when it is preferable to drive, walk, or otherwise travel over or through the soft-obstacles. Hence, we need to locate the soft-obstacles.

1.1 Terrain Typing Background

Terrain typing is not unstudied, although real-time terrain typing has often been neglected. Some interesting work has been done by [Marra88] and [Wright89]. Wright used neural networks to find roads in images, but he performed significant image segmentation before applying the neural network, and not much emphasis was put on finding the terrain type quickly.

Marra, Dunlay, and Mathis used several techniques, and the early work we have done in our group is derivative primarily from one of their techniques, the "Image Based Neural Networks" of [Marra88]. 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 to use fewer hidden units in the neural networks; this will give us a chance to achieve 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]). Our system is quick to train and quick to run.

1.2 Our Research Goals

Our research group is studying navigation using the NavLab II military ambulance, also known as the HMMWV (High Mobility Multi-Wheeled Vehicle, pronounced "Humvee", Figure 1, on page 2). At one of our primary test sites on an old slag heap (2 km x 2km), vegetation runs rampant throughout the year. We most frequently use a laser range finder to navigate there, and to the laser range finder the vegetation looks just like the natural hills, ridges, and man made mounds which we must avoid. Usually we just avoid everything for simplicity's sake.

FIGURE 1. The HMMWV

However, this approach limits the reachable regions on the slag heap. Additionally, this approach makes all of our navigation less efficient. Only one system has been implemented so far that handles vegetation [Davis95], and the manner with which vegetation is dealt is fairly tightly integrated into that system's paradigm (modular neural network control). This paper describes a different system for finding vegetation in single images: IVY (for Ian's 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. In fact, IVY can be used to do more complicated terrain typing, too, but this paper assumes a single division of vegetation or non-vegetation.

2.0 The Approach: IVY 1

In order to best understand the problem of terrain typing, we are starting with a simple architecture and a simple problem to solve. In the work detailed here and in our future work, we shall expand our technique and goals as we expand our understanding of the complexity of the problem.

2.1 Architecture

Ivy 1, which is covered by this paper, uses a monolithic neural network approach to classification. The paradigm is referred to as the *operator architecture* [Davis93a] since we use the neural network much as we would any low level computer vision operator such as an edge detector. Ivy 1 uses a simple 3-layer backpropagation neural network [Rumelhart86]. Ivy 2 will use a MAMMOTH modular network architecture [Davis95a] (see "Future Directions" on page 10).The inputs to IVY 1 are three small retinas, a red, a green, and a blue one. This is for use with an unmodified color CCD camera[1]. 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 1 represents whether or not the center pixel is to be classified as vegetation. The size of the retina can be adjusted to accommodate different resolutions and the amount of texture or averaging you wish to have affect the classification (this will be discussed in "Averaging and Texture" on page 5 & "Larger Retina Experiment" on page 8). See Figure 2, on page 4 for a diagram of IVY 1.

2.2 Training

Training IVY 1 is straightforward. We hand-label each pixel 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 we randomly select a training set with usually 500 training exemplars. Each exemplar is a an ordered pair (x, y) in which x is a 3-band (red, green, & blue) 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. 500 exemplars is sufficient for providing a *functional approximation basis*[2] for the mapping we are trying to learn.

1. For possibly better results, we could have an Infra-Red retina. IR has been shown to be very useful in distinguishing vegetation from other terrains [Kelly95].

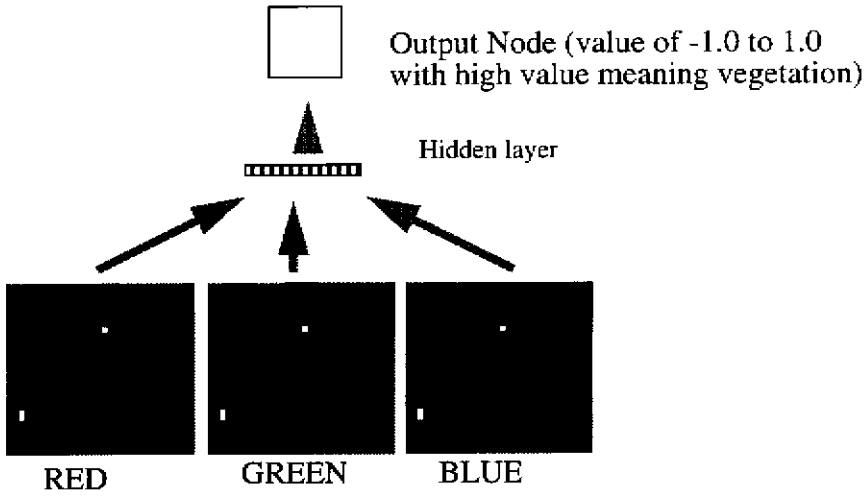


FIGURE 2. IVY 1 monolithic operator network

2.3 Justification

2.3.1 Color Space Complexity

The obvious question is why do we use a neural network approach to classification. The answer has two aspects. Consider the problem as if we were using retinas of one pixel so that the classification was based strictly on color space considerations. First, the mapping from RGB image space to terrain space can be nonlinear and complicated. Simple color space matching (such as used in hand-labelling training images in which all colors within a small neighborhood of a given (R,G,B) point are positively labelled) is not appropriate alone for complicated images for the same reason that a linear classification surface in color space does not always work. A linear classification surface (a plane in RGB color space) or a combination of linear classification surfaces will only work if the set of points in RGB that maps either to positive vegetation or negative vegetation is entirely convex (the proof of this is trivial). 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

2. A functional approximation basis, β_f, is a concept which we are beginning to examine in which we have a function we desire to learn, f from space X to space Y, and each element of β_f is a pair (x_i, y_i) where all of the x_i form a topological basis of the set in X which we wish to map from and the y_i form a topological basis of the set in Y which we want to map to. A rigorous exploration of the appropriate size of β_f for a given f, and of the relationship between the niceness of f and the details of the topological bases of X & Y will be undertaken in the future and promises to uncover good methods for determining good training sets for given architectures and training algorithms (such as the operator architecture and backpropagation in IVY 1).

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.

Even using a series of simple color space neighborhood matches impractical. Assume for the moment that the set, V, *in RGB space which maps to vegetation in terrain space is the* "simpler" *set; perhaps it could be called almost-convex.* For every "dent" in V (a separate convex region, R, of RGB space outside of V which fits in one of the concavities on the surface of set V), we would require at least one additional simple color neighborhood match to accurately reproduce the correct mapping. If V were convex, this would mean one match, but V is not guaranteed to be convex or even for that matter *connected* in RGB space.

2.3.2 Averaging and Texture

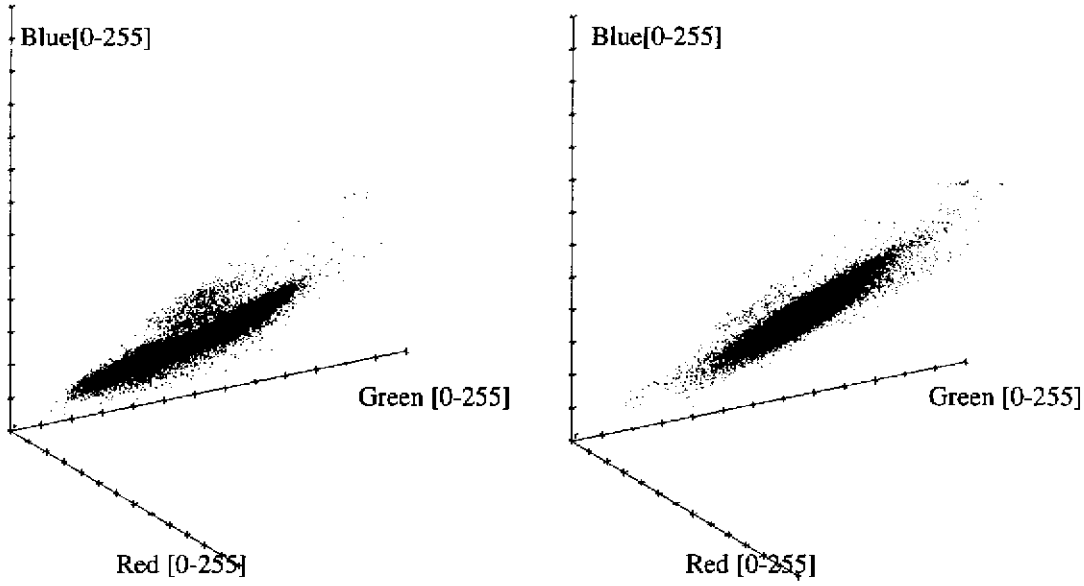
The second aspect is that when we use retinas larger than one pixel we want to achieve classification based in part on averaging or texture, and neural network techniques allow us to create these more complex mappings easily. Averaging is important for complete image classification because we get shadows 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.

In the work detailed here we deal with texture implicitly as did Marra, et al., though our next stage of research will concentrate more on utilizing texture data. Texture considerations become especially important when there are more man-made obstacles in the environment which might be vegetation colored. Since man-made items tend to be smoother and more continuously colored than vegetation in a CCD image, including a sufficiently big retina at a sufficiently high resolution can give us important clues for classification, and even our early work allows these clues to contribute to the mapping we construct.

In this paper, we show the results of both a purely color space based IVY 1 system and the results of an IVY 1 which used a 7 x 7 pixel retina to try to capture averaging and texture information. Both systems run in real-time on the HMMWV (where the Hz are determined by initial image resolution and whether or not we sample retinas to get a lower resolution output image - all of which is programmable on the fly).

3.0 Color Space Only Experiment

The simplest IVY 1 network which we trained had a single RGB pixel as an input (3 input units in the neural network). We used a series of 5 high resolution images (640x480) to generate 500 training exemplars and 500 initial testing exemplars. 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.



a. Vegetation **b. Non-vegetation**

FIGURE 3. Plot of training set for 1 pixel input IVY 1. 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 are almost two distinct 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 represented 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. You can see the relatively small portion of RGB space actually covered in Figure 3, on page 6. 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 misclassifications (by hand), and 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 1 network was trained with each number of hidden units from 1 to 15 and the results are shown in Figure 4, on page 7.

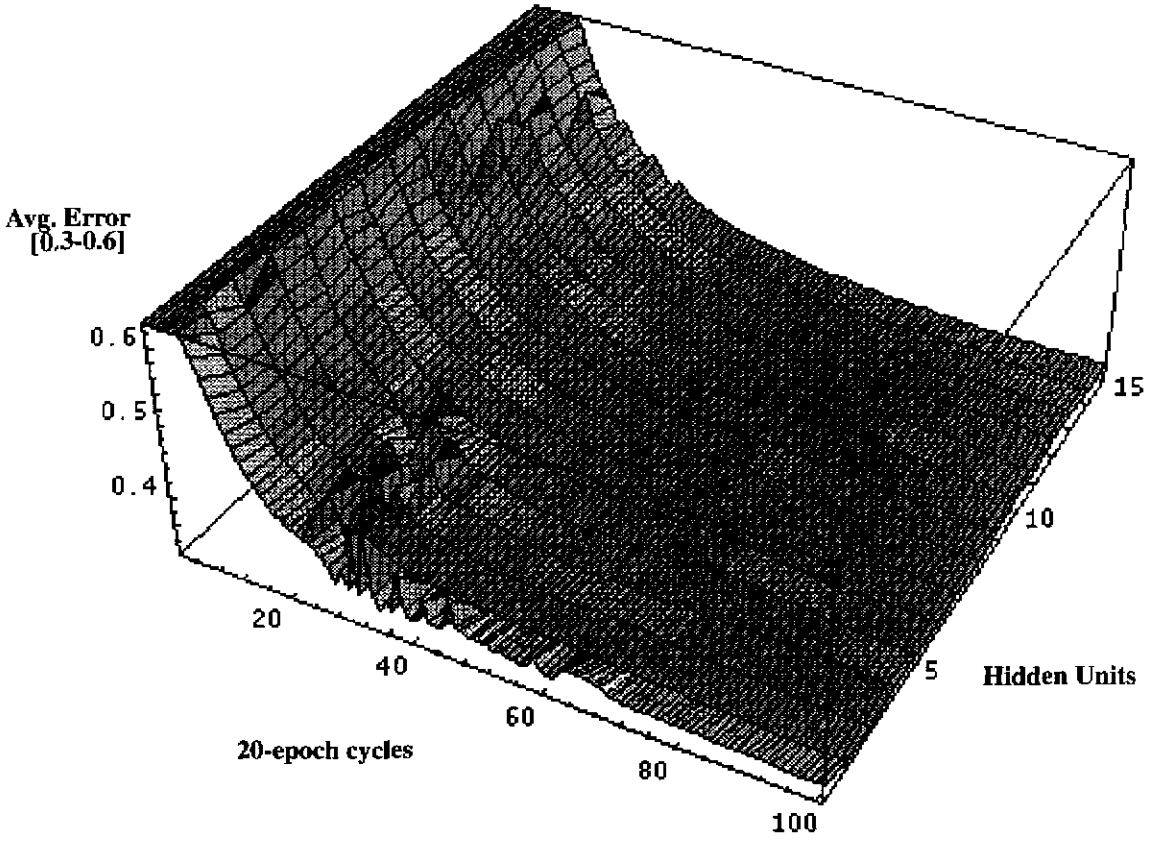


FIGURE 4. Average error per test pixel over (100) 20-epoch training cycles for 1 by 1 pixel IVY 1 with 1 to 15 hidden units. Note that average error on over 500 test cases never was < 0.3. Output of IVY 1 is single unit ranging from -1.0 to 1.0 with 1.0 being "vegetation" and -1.0 being "non-vegetation."

In the diagrammed 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 1 networks the average classification error on the given test set of 500 exemplars was more than 0.3. 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 (which make us use a looser threshold on the output for our terrain decision).

4.0 Larger Retina Experiment

In order to take advantage of averaging and texture, we also trained an IVY 1 network that had a 7 x 7 RGB pixel retina (147 input units). This network's architecture is very similar to that used in [Marra88], 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 1 network as for the simpler one described in "Color Space Only Experiment" on page 6. Again, 500 exemplars were used for a training set and 500 were used as a test set.

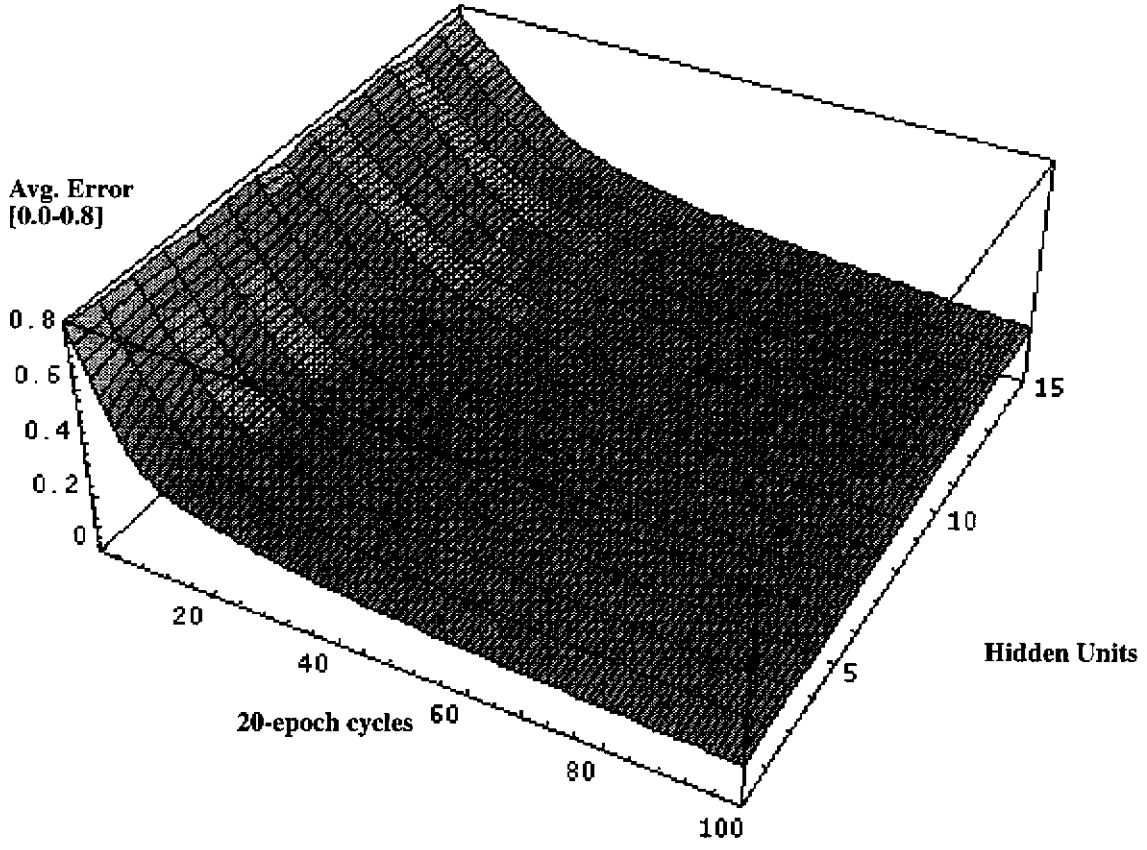


FIGURE 5. Average error per test pixel over (100) 20-epoch training cycles for 7 by 7 pixel IVY 1 with 1 to 15 hidden units. Note that average error on over 500 test cases went as low as 0.15. Output of IVY 1 is single unit ranging from -1.0 to 1.0 with 1.0 being "vegetation" and -1.0 being "non-vegetation."

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 has 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; see Figure 5, on page 8). This is partially because there was signifi-

cantly less overlap between the sets of vegetation and non-vegetation, both in the *ideal* mapping and in the *actual* training set. The high dimensionality of the inputs to the 7 x 7 input IVY 1 prevents us from generating an understandable plot of the vegetation and non-vegetation sets in their raw form.

5.0 Conclusions

5.1 Results

The average error for the 7 x 7 input IVY 1 was less than half of that for the 1 pixel IVY 1 network. This was a combination of misclassifications and a less well-learned mapping (the latter of which frequently causes the former). The reasons are that the larger retina IVY 1 simply had more information that was relevant to the ideal mapping. The implicit texture and averaging information of a large retina made the mapping less dependent on strict color space data, which is often insufficient for classifications (as can be seen in the set overlaps of Figure 3, on page 6).

If speed were of the utmost importance, the 1 pixel IVY 1 system performs adequately, but the network is sufficiently small for a 7 x 7 input IVY 1 (with few enough hidden units) that there would rarely be enough motivation to forego the greater reliability of the more complicated network. A nice side effect of the larger retinas is that they can be used to tessellate the image if sampling is desired to increase speed (Marra, et al, used a large scale tessellation to reduce a 512 x 512 raw image to 32 x 32 classification image). Such a tessellation allows each pixel to contribute to its region's classification due to the large retina. Thus, in practice we can classify real images in real-time[1] on our HMMWV vehicle.

As we noted earlier, even the RGB space considerations alone (such as non-convex classification sets) justify the use of neural networks or other equally sophisticated clustering and function approximation techniques. The additional averaging and texture information makes a sophisticated technique that much more important since different dimensions of the input space can imply entirely different classifications.

We have developed a system based on [Marra88] that we feel has practical usefulness which is due to focussing on a simpler classification problem. We suspect that combining the results of several such classifiers for different terrain

1. Where real-time is defined as faster than the cycle time of our other systems that perform navigation. This definition means that we can process an image in time to account for vegetation in each planning cycle.

types may even have superior results to a single classifier that tries to learn a more difficult classification problem. At this stage it is important for us to try to solve the simplest problems quickly and well.

5.2 Future Directions

One of the issues not addressed here is the separation of averaging effects from texture effects. Even for our simple initial tasks texture might be worth explicitly handling, and if we were to want to classify only certain types of vegetation, the texture could be even more important than the color space considerations. Thus, we are developing a series of tests to examine the utilization of texture in IVY 1. One test is to use a network trained at high resolution on a lower resolution image, in effect eliminating the texture data. Another test involves using artificially generated supplements to the training set and the test set which include retinas filled with colors appropriate to vegetation but in distinctly non-vegetation textures. Marra, et al, tried to have the neural network learn texture information implicitly, and a more explicit approach may be needed (though they explicitly represented where in the image a pixel was since its location affects its texture, and this idea is one we shall examine).

Since we know ahead of time that texture and averaging are important, if one of these crucial pieces of information is shown to be not fully utilized we will extend the IVY model to IVY 2, an operator architecture based on the MAMMOTH modular neural network architecture and training paradigm [Davis95]. With MAMMOTH we train subnetworks to recognize important known features and then integrate the pre-trained hidden units into the supervising task network (for IVY 2, we could have feature networks representing texture or the colors of surrounding pixels).

Finally, to extend the usefulness of either IVY 1 or IVY 2, we shall develop a regimen for developing a training set appropriate to multiple lighting and weather effects as well as a richer variety of vegetation. As we include these variations and seasonal color changes into our training sets, we will want to have texture and averaging and color space information well under control.

We hope to have many of these issues addressed in the near future, and plan to have some version of IVY integrated into several autonomous navigation systems within a very short time.

6.0 Acknowledgments

This work has been supported in part by a National Science Foundation Graduate Research Fellowship. The equipment used is provided in part by DACA76-89-C0014, Topographic Engineering Center, Perception for Outdoor Navigation and DAAE07-90-C-R059, TACOM, CMU Autonomous Ground Vehicle Extension. Thanks go to all of the Robotics Institute at Carnegie Mellon University and the Field Robotics Center, and in particular Jeremy Armstrong and Jim Frazier for keeping the vehicle working, and Tony Stentz, Dean Pomerleau, and Mel Siegel for keeping me working.

7.0 References

[Brumitt92] B. Brumitt, R. Coulter, A. Stentz. "Dynamic Trajectory Planning for a Cross-Country Navigator," *Proc. of the SPIE Conference on Mobile Robots*, 1992.

[Daily88] 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.

[Davis93a] 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.

[Davis93b] 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.

[Kelly93] A. Kelly, "A Partial Analysis of the High Speed Cross Country Navigation Problem," Carnegie-Mellon University Ph. D. Thesis Proposal, 1993.

[Kelly95] A. Kelly, personal correspondence, December, 1994.

[Langer93] D. Langer, J. K. Rosenblatt, M. Hebert, "A Reactive System For Off-Road Navigation," Carnegie Mellon University Technical Report, CMU-RI-TR-93-, 1993.

[Marra88] M. Marra, R. T. Dunlay, and D. Mathis, "Terrain Classification Using Texture for the ALV," Proceedings of the SPIE Conference on Mobile Robots, 1992.

[Pomerleau92] D. Pomerleau, *Neural Network Perception for Mobile Robot Guidance*, Ph.D. Dissertation, Carnegie-Mellon University Technical Report CMU-CS-92-115, 1992.

[Pomerleau91] D. Pomerleau, "Neural network-based vision processing for autonomous robot guidance," *Proceedings of SPIE Conference on Aerospace Sensing*, Orlando, FL, 1991.

[Rumelhart86] 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.

[Stentz93] A. Stentz, "Optimal and Efficient Path Planning for Unknown and Dynamic Environments," Carnegie Mellon University Technical Report, CMU-RI-TR-93-20, 1993.

[Wright89] W. A. Wright, "Contextual Road Finding With A Neural Network," Technical Report, Sowerby Research Centre, Advanced Information Processing Department, British Aerospace, 1989.