Machine Learning, 15, 125-135 (1994)


# Neural Network-Based Vision for Precise Control of a Walking Robot

DEAN A. POMERLEAU (POMERLEAU+@CMU.EDU)
*School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213*

**Editor:** Alex Waibel

**Abstract.** This article describes a connectionist vision system for the precise control of a robot designed to walk on the exterior of the space station. The network learns to use video camera input to determine the displacement of the robot's gripper relative to a hole in which the gripper must be inserted. Once trained, the network's output is used to control the robot, with a resulting factor of five fewer missed gripper insertions than occur when the robot walks without sensor feedback. The neural network visual feedback techniques described could also be applied in domains such as manufacturing, where precise robot positioning is required in an uncertain environment.

**Keywords.** neural networks, vision, robot control.

## 1. Introduction

Building visual perception systems for robot control using traditional machine vision techniques is a labor-intensive, time-consuming process. The programmer must 1) determine what features in the image are important, 2) implement an image-processing scheme to detect those features, and 3) develop an algorithm to determine the appropriate robot response from the detected features.

This article illustrates that the learning power of artificial neural networks can effectively eliminate much of the difficulty involved in developing robust vision-based autonomous guidance systems. The article focuses on the use of artificial neural networks for the precise control of the Self Mobile Space Manipulator ($SM^2$), a robot designed to walk on the external structure of the space station Freedom (Brown, Friedman, & Kanade, 1990; Ueno, Xu, & Brown, 1990). Application of similar neural network techniques for autonomous robot driving can be found in Pomerleau (1991a, 1991b).

## 2. The task

Space is a dangerous environment for people. To reduce this danger in the construction and maintenance of the space station Freedom, a number of robot systems are under development. One of those robots, called the Self Mobile Space Manipulator ($SM^2$), is being designed at Carnegie Mellon University to perform visual inspection, transportation of parts, and light construction tasks (see figure 2). The $SM^2$, shown in figure 1, is a two-legged robot capable of rapid walking along the outside of the space station. Grippers at the end of each leg screw into threaded holes in the nodes where support struts join together.

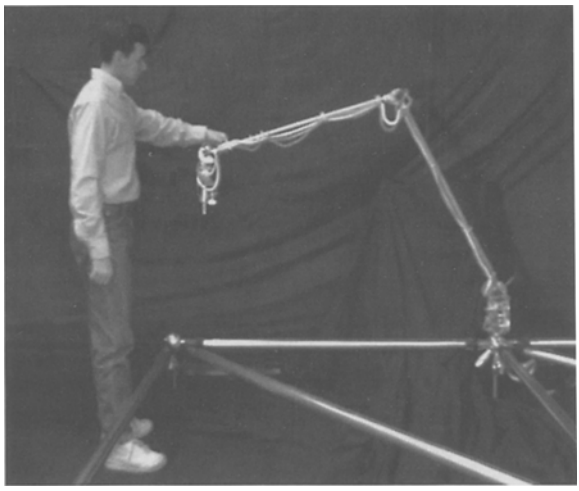

*Figure 1.* Image of the Self Mobile Space Manipulator ($SM^2$).

Locomotion is achieved by alternately unscrewing one gripper from its anchor hole and swinging it around to screw into the next hole (see figures 2 and 3).

In order to build a compact, power-efficient robot capable of fast walking, the robot's mass has been kept to a minimum by using lightweight aluminum legs. As a result of the flexibility in these legs, and the long distance between adjacent anchor holes (15 feet on the space station, and 5 feet on the one-third scale CMU testbed model), it is difficult to consistently position the gripper of the robot within the required 0.25 inches of the anchor hole for reliable insertion using only measured joint angles. In other words, sensor feedback is required for the precise positioning of the gripper. To facilitate this feedback, small monochrome video cameras are attached to the robot's grippers, as illustrated in figure 3. The next section describes the artificial neural network designed to guide the robot using these video cameras.

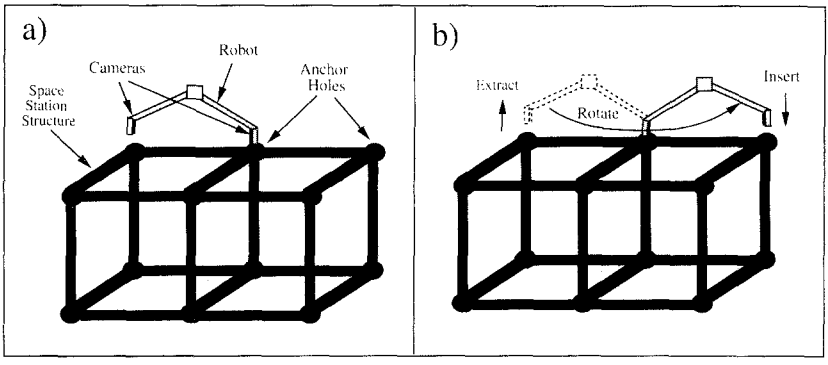


*Figure 2.* Schematic of the $SM^2$. (a) Robot and space station structure; (b) locomotion.

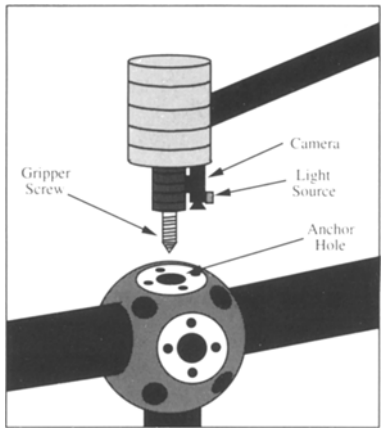


*Figure 3.* A schematic closeup of the gripper on the $SM^2$.

## 3. Network architecture

Given an image coming from the camera at the gripper, the vision system needs to provide the two-dimensional displacement of the gripper screw relative to the anchor hole. To accomplish this mapping, a neural network architecture very similar to that in Pomerleau (1991a, 1991b) is used. Specifically, the input layer consists of a 24 × 20 two-dimensional "retina" that receives input from the gripper's video camera. The activation level of each unit in the input retina represents the grey-scale intensity of the corresponding pixel in the low-resolution video image coming from the camera. Each unit in the input retina is fully connected to a layer of five hidden units, which are in turn fully connected to two vectors of 20 output units (see figure 4). The first output vector is a linear representation of the displacement of the gripper relative to the anchor hole in the X dimension, ranging from −1.25 inches for the lefmost output unit to +1.25 inches for the rightmost output unit. The second vector of output units is identical to the first except that it represents displacement in the Y dimension.

To control the precise positioning of the robot's gripper once it is in the vicinity of the anchor hole, a video image from the appropriate gripper camera is projected onto the input layer. After completing a forward pass through the network, the network's estimate of the gripper screw's X and Y displacements is extracted from the output vectors. The displacement indicated by the network in each dimension is taken to be the center of mass of the "hill" of activation surrounding the output unit with the highest activation level. Using the center of mass of activation instead of the most active output unit when determining the estimated displacement permits more precise estimates, and therefore improves the network's performance. For a more detailed analysis of the improvement achieved using this alternative output representation, see Pomerleau (1992).

The estimated X and Y displacements of the gripper screw are then converted to joint torques by the robot's PID controller in order to move the gripper over the anchor hole.

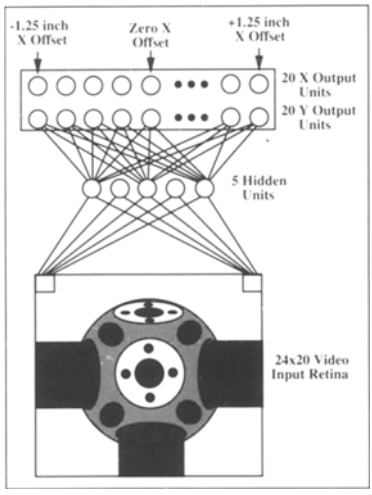


*Figure 4.* Architecture of network designed to control the $SM^2$.

Once the network has indicated that the gripper is within the 0.25-inch "threshold radius" of the anchor hole for over one second, the gripper is considered to be stably positioned over the hole, and the controller lowers the gripper to anchor the leg.

## 4. Network training and performance

The network is trained using the back-propagation supervised training algorithm (Rumelhart, Hinton, & Williams, 1986). In this technique, the network is repeatedly presented with input patterns and trained to produce the correct response for each by altering the strengths of the connections between units. On each presentation of each pattern, the network's connection strengths are slightly altered to decrease the discrepancy between the actual output response and the desired output response. The more influential a unit is in producing the current, incorrect response, the more its weights are altered.

In order to develop a general representation and to avoid simply memorizing the training set, a network trained with back-propagation must be presented with a relatively large and varied set of training examples. To provide the precise X–Y gripper displacement measures required by the network, a special training jig was constructed. The jig consisted of distance encoders that were temporarily attached to the robot gripper during the data collection phase (see figure 5). Approximately 500 video images are taken while the gripper's position relative to the anchor hole is manually varied in three dimensions. Each of the collected images is automatically tagged with its corresponding X–Y offset. These displacements in X and Y are converted into output vectors for use in training. Instead of being trained to activate only a single output unit in each vector, the network is trained to produce a gaussian distribution of activation centered around the actual displacement in each dimension. As in the decode stage, the actual displacement may fall between the displacements

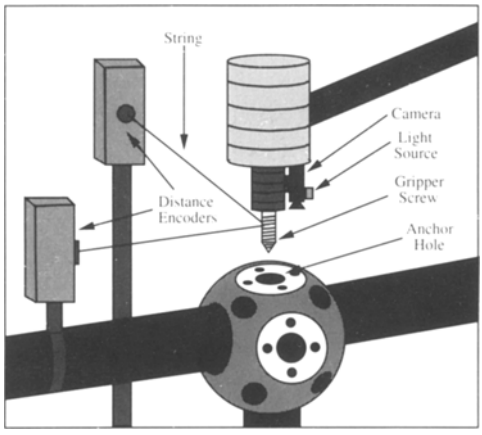


*Figure 5.* Distance encoders temporarily attached to robot provide accurate gripper displacement data during data collection.

represented by two output units. The following approximation to a gaussian distribution is used to precisely interpolate the target output activation levels:

$$x_i = e^{-\frac{d_i^2}{10}}$$

where $x_i$ represents the desired activation level for unit $i$, and $d_i$ is the $i$th unit's distance from the correct steering direction point along the output vector. The constant 10 in the above equation is an empirically determined scale factor that controls the width of the gaussian curve. The above equation corresponds to a normal distribution with a standard deviation $\sigma = \sqrt{10}$.

As an example, consider the situation in which the actual gripper displacement along one dimension falls halfway between the displacements represented by output units $j$ and $j + 1$. Using the above equation, the desired output activation levels for the units successively farther to the left and the right of the correct displacement value fall off rapidly, with the values 0.98, 0.80, 0.54, 0.29, 0.13, 0.04, 0.01, etc.

This gaussian desired-output vector can be thought of as representing the probability density function for the correct displacement value, in which a unit's probability of being correct decreases with distance from the gaussian's center. By requiring the network to produce a probability distribution as output, instead of a "one of N" classification, the learning task is made easier, since slight changes in the gripper screw's position relative to the anchor hole require the network to respond with only slightly different output vectors. This is in contrast to the highly non-linear output requirement of the "one of N" representation, in which the network must significantly alter its output vectors (from having one unit active in each vector and the rest off to having a different unit active in each vector and the rest off) on the basis of fine distinctions between slightly shifted images of the anchor hole.

During training, the back-propagation algorithm uses the 500 images collected with the camera and the corresponding displacement output vectors as examples of the mapping to be performed. After approximately 100 presentations of these exemplars, the network learns to use image features to accurately determine the gripper displacement. Figure 6 illustrates the evolution of the weights projecting into the five hidden units from the video retina at four different times during training. Prior to training, the network's connections are random, as illustrated by the unstructured distribution of positive weights (light squares) and negative weights (dark squares) at epoch 0 in figure 6. As training progresses, figure 6 shows the weights to the hidden units evolving to pick out important image features. The bottom row of figure 6 illustrates the final set of filters, or feature detectors developed by the network. Notice that the weights of the connections from the input retina into hidden unit 1, shown in the lower left corner of figure 6, form a center vs. surround detector, with hidden unit 1 being inhibited if there are active (bright) pixels in the center of the image and excited by bright regions in the periphery. The other feature detectors also exhibit spatially coherent structure, including preferences for bright regions towards the top or bottom, and towards the left or right or the image. In addition, hidden units 3 and 5 show a preference for annular patterns at particular locations, an important characteristic of the anchor hole's appearance. The network uses the position of features like the reflective ring around the anchor hole to determine the gripper offset relative to the hole itself. The data collection and training phases require a total of about 20 minutes running on a Sun-4 workstation.

Once trained, the system is able to provide 15 precise and accurate gripper displacement values per second under a variety of laboratory lighting conditions. Quantitatively, the net-

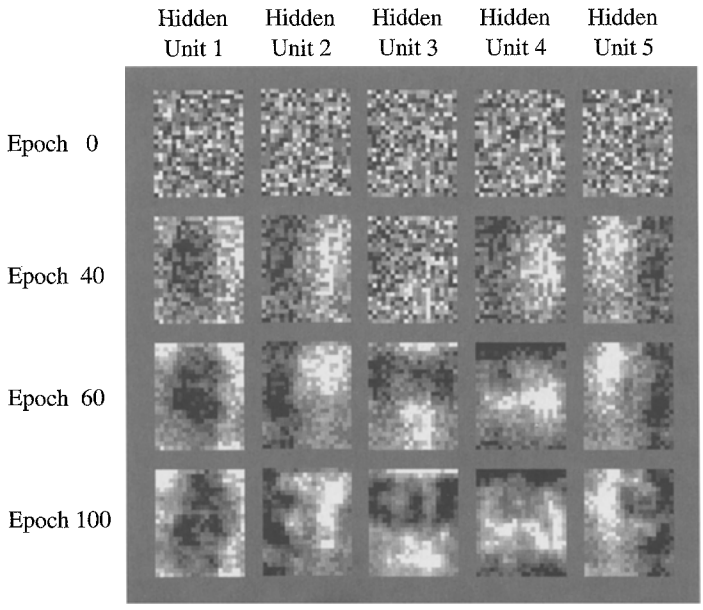


*Figure 6.* The weights projecting from the input retina to the five hidden units in the network at four points during training.

work's displacement estimate has a mean error of 0.1 inch and a standard deviation of 0.07 inch. The robot is equipped with its own light source for illuminating the area under the gripper, as can be seen in figure 5. In the laboratory, this local light source is powerful enough to eliminate shadows and other illumination variations caused by ambient lighting conditions. For a discussion of how to cope with the harsh lighting conditions of space, see section 6.

Figure 7 illustrates the low-resolution video input to a trained network and the resulting network responses for X and Y gripper displacement. In the case of figure 7, the robot gripper is actually centered over the anchor hole, although it does not appear to be, due to the camera's displacement relative to the gripper (see figure 5). The network responds correctly in this situation by most strongly activating the centermost units of the two output vectors, indicating that the gripper has zero offset in the X and Y dimensions.

When compared with using no sensor feedback, the neural network method for precisely guiding the $SM^2$ resulted in a factor-of-five decrease in missed gripper insertions. On a test of 20 "steps" (i.e., moving the gripper screw from one anchor hole to another), the robot failed to insert the gripper screw into the anchor hole on only one step when guided by the neural network. In contrast, it made five missed insertions when relying solely on the angles of its joints to determine the gripper's position. Because of the success of the neural network guidance system, the non-connectionist implementations described in the next section were never tested on the robot itself, making it impossible to compare their performance.

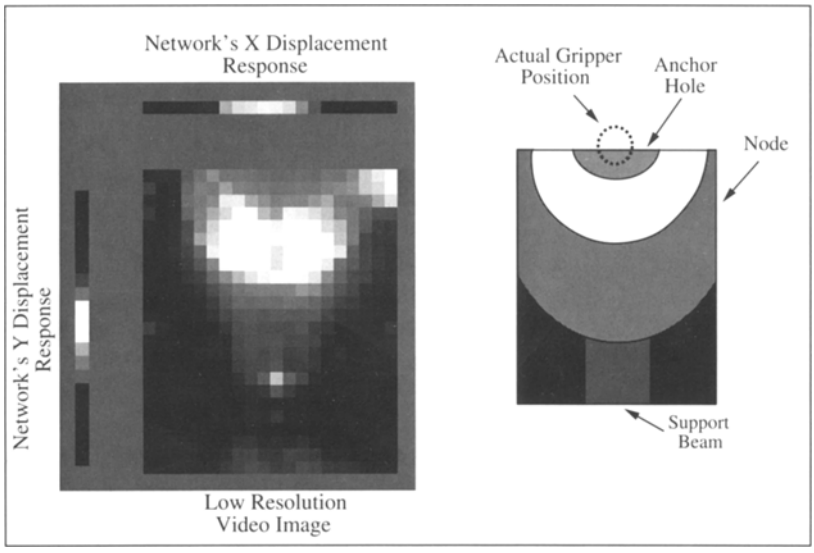


*Figure 7.* The low-resolution video image provided as input to a trained network and the network's X-Y displacement responses. The gripper screw is actually centered over the hole in this image, as indicated in the schematic to the right. The anchor hole does not appear centered in the image, since the camera is offset from the gripper screw.

## 5. Alternative architectures and approaches

Numerous alternative architectures were tested both for this task and for the task of autonomous driving. Dimensions along which the architecture was varied include the dimension of the input retina (ranging from 12 × 10 up to 60 × 50), the number of hidden layers (ranging from 0 to 2), the number of hidden units (ranging from 2 to 70), and the size of the output vectors (ranging from 1 to 50 units). Empirically, the architecture described above proved to be "optimal" in the following sense. Networks with more complex architectures (i.e., larger input retinas, more hidden units in a single layer, or more hidden layers) provided no significant performance benefits in terms of more accurate displacement estimates. But decreasing the retinal size, the number of hidden units, or the number of output units resulted in less accurate displacement estimates and therefore more missed gripper insertions.

Interestingly, when the input retina was connected directly to the output units, without an intervening hidden layer, the network's estimate of the gripper screw's displacement was *not* significantly less accurate, indicating that the task is linearly separable. However, connecting every input unit to every output unit resulted in a network with over seven times as many connections as the fully connected network with a single layer of five hidden units. The increased network size resulted in a large decrease in cycle rate, from 15 frames per second to 3 frames per second, which resulted in severe instability in the control loop, and very low insertion reliability. When connections in the perceptron network were randomly pruned before training so as to equal the number in the five-hidden-unit network, the sparsity and arbitrary nature of the connectivity pattern prevented the network from learning to precisely estimate the gripper screw displacement. Quantitatively, the network without hidden units was 30% less accurate than the five-hidden-unit network, with a mean error in its estimate of gripper-screw displacement of 0.13 inch vs. 0.10 inch for the network with five hidden units.

Work remains to be done in determining how other connectionist and non-connectionist learning techniques compare with the back-propagation approach to this task. The K-nearest-neighbors algorithm (MacQueen, 1967) in the limit should perform almost perfectly, but would probably require many templates, and hence more computation, to achieve equivalent performance. Other learning techniques, such as Kohonen's algorithm (Kohonen, 1990) and radial basis functions (Poggio & Girosi, 1990), might also be successfully applied in this domain. For a more detailed comparison of learning architectures and algorithms for mobile robot guidance, see (Pomerleau, 1992).

Two other methods that have been explored for guiding the $SM^2$ are based on classical image-processing techniques. The have utilized targets added to the space station structure to make the image-processing task more tractable. One system uses the black and white cameras currently on the robot and a white-on-black crosshair attached near each anchor hole. The system finds the crosshair in the image and uses its position to determine the tip displacement. This system fails if the target is obscured or out of the camera's field of view. The second system uses a color camera and a red ring surrounding each anchor hole. Again, image-processing techniques are used to find the ring and to use its position to determine tip displacement. This system is as accurate as the neural network, providing gripper-screw displacement estimates with a mean error of approximatley 0.1 inch, but requires a more sophisticated camera and, more importantly, alteration of the node itself

with a colored target. In addition, after choosing the target, the programmer of both these systems was required to develop algorithms for finding the target and determining the gripper-screw offset from the target's size and location. In contrast, the neural network was able to *learn* to estimate the gripper-screw offset from example.

As a result of its ability to learn the task from observation, the connectionist system required much less development effort than the handcrafted implementations. Once the camera and distance encoders were in place for data collection, it took under an hour to develop the connectionist vision system for estimating gripper offset. In contrast, approximately one man-month of effort by a vision graduate student was required to achieve equivalent performance with the ring-tracking system.

## 6. Conclusion and future work

The ease of development relative to other techniques provides a distinct advantage for the connectionist approach to visual robot guidance. But three questions remain regarding the reason for success of this approach, its limitations, and its generality. First, why was development of the neural network vision system for guiding the $SM^2$ so easy, particularly when compared to the effort required to achieve accurate handwriting or object recognition with connectionist methods? The answer is that finding the location of known features in an image is significantly easier than classifying the image based on the identity and relation of those features. In fact, to perform the task of object recognition, the system must be *invariant* to the absolute position of image features, since it must recognize objects regardless of their image position. This type of invariance is difficult to achieve in connectionist networks, and requires the use of specialized architectures (LeCun et al., 1989; Waibel et al., 1987).

So what are the limitations of this approach to connectionist vision? First, since it is based on finding the positions of specific features in the image, it is only applicable to tasks where feature positions are important. Fortunately, there are a number of tasks, particularly in robotics, that require determining the precise position of features in the environment. In addition to guiding the $SM^2$, these techniques have been successfully applied to steering an autonomous automobile at speeds of up to 55 mph (Pomerleau, 1991b). Work is also in progress in using these connectionist methods to guide the landing of an unmanned flying vehicle (Davis & Stentz, 1992).

Another limitation is that to provide accurate responses to slight positional differences in the image, the network must be trained for a relatively narrow range of situations. For the $SM^2$ experiments performed under laboratory lighting conditions, this was not a problem, since all the anchor holes appeared similar. Under widely varying lighting conditions like those encountered in space, it may be necessary to train separate networks for particular situations. The question then becomes one of selecting the appropriate network for the conditions at hand. Numerous techniques have recently been developed for just this type of multi-network integration, including the Meta-Pi architecture (Hampshire & Waibel, 1992), the task-decomposition architecture (Jacobs et al., 1990), and input reconstruction reliability estimation (Pomerleau, 1992).

Another approach we are currently pursuing to cope with harsh lighting conditions is to use an alternative sensor. Instead of a video camera, we are developing a ring of eight photosensors that surrounds the gripper screw. Each sensor is paired with its own light-emitting diode to illuminate the area below the photosensor. By measuring the relative strengths of the light reflecting back to the eight photosensors, the position of the gripper screw can be accurately estimated. A connectionist network identical to the one discussed in this article, but with eight input units instead of the 24 × 20 unit retina employed for the video camera, has already proven capable of estimating the gripper screw's displacement with an average error of 0.12 inch, only slightly worse than that of the video network. In addition, by strobing the LED light sources in very short but intense bursts, and then subtracting the signal from each photodiode under ambient lighting from its reading under the intense LED illumination, the effects of even extremely harsh ambient lighting can be effectively eliminated.

Since the neural network was easy to develop and can learn to robustly provide accurate tip-displacement data without altering the space station's appearance, the neural network is the vision system currently being employed in the development of the $SM^2$. The flexibility of the connectionist approach to robot vision is allowing us to test potentially useful new sensors with relatively little effort. In addition, we plan to employ these techniques for other space-related tasks requiring vision feedback, such as reliable grasping of cargo handles or hatch doors. The connectionist approach to vision feedback also has the potential to benefit non-space-related domains such as manufacturing, where precise robot positioning is required in an uncertain environment.

## Acknowledgments

This research was supported by Shimizu Corporation, by the Office of Naval Research under Contracts N00014-87-K-0385, N00014-87-K-0533 and N00014-86-K-0678, and by National Science Foundation Grant EET-8716324.

The research would not have been possible without the support $SM^2$ group at Carnegie Mellon University. In particular, the help of Ben Brown, Hiroshi Ueno, and David Simon has been crucial to the success of this project.

## References

Brown, H.B., Friedman, M.B., & Kanade, T. (1990). Development of a 5-DOF walking robot for space station application. *Proceedings of the IEEE International Conference on Systems Engineering* (pp. 194–197). Pittsburgh, PA: IEEE Press.

Davis, I., & Stentz, T. (1992). Personal communications.

Hampshire, J.B., & Waibel, A.H. (1989). The Meta-Pi network: building distributed knowledge representations for robust pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 14(7)*, 751–769.

Jacobs, R.A., Jordan, M.I., & Barto, A.G. (1991). Task decomposition through competition in a modular connectionist architecture: the what and where vision tasks. *Cognitive Science, 15(2)*, 219–250.

Kohonen, T. (1990). *Self-organization and associative memory*, 3rd. ed. Berlin: Springer-Verlag.

LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., & Jackel, L.D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation, 1(4)*, 541–551.

MacQueen, J. (1967). Some methods of classification and analysis of multivariate observations. In L.M. LeCam & J. Neyman (Eds.), *Proceedings of the 5th Berkeley Symposium on Mathematics, Statistics, and Probability.* University of California Press, Berkeley, CA.

Poggio, T., & Girosi, F. (1990). Regularization algorithms for learning that are equivalent to multilayer networks. *Science, 247,* 987–982.

Pomerleau, D.A. (1991a). Efficient training of artificial neural networks for autonomous navigation. *Neural Computation, 3(1),* 88–97.

Pomerleau, D.A. (1991b). Neural network-based vision processing for autonomous robot guidance. *Proceedings of SPIE Conference on Aerospace Sensing* (pp. 121–128). Orlando, FL.

Pomerleau, D.A. (1992). Neural network perception for mobile robot guidance. Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, PA. Also Technical Report CMU-CS-92-115.

Pomerleau, D.A. (1993). *Neural network perception for mobile robot guidance.* Norwell, MA: Kluwer Academic Publishers.

Rumelhart, D.E., Hinton, G.E., & Williams, R.J. (1986). Learning internal representations by error propagation. In D.E. Rumelhart & J.L. McClelland (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition. Vol. I: Foundations* (pp. 318–362). Cambridge, MA: Bradford Books/MIT Press.

Ueno, H., Xu, Y., & Brown, H.B. (1990). On control and planning of a space station robot walker. In *Proceedings of 1990 IEEE International Conference on Systems Engineering* (pp. 220–223). Pittsburgh, PA: IEEE Press.

Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., & Lang, K. (1989). Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech and Signal Processing, 37(3),* 328–339.

Received September 27, 1990
Accepted March 17, 1992
Final Manuscript December 15, 1992