Non-Intrusive Gaze Tracking Using Artificial Neural Networks

Shumeet Baluja & Dean Pomerleau

5 January 1994

CMU-CS-94-102

School of Computer Science
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Portions of this paper appear in: Baluja, S. & Pomerleau, D.A. "Non-Intrusive Gaze Tracking Using Artificial Neural Networks", **Advances in Neural Information Processing Systems (NIPS) 6**. Cowan J.D., Tesauro, G. & Alspector, J. (eds.) Morgan Kaufmann Publishers, San Francisco, CA., 1994.

Abstract

We have developed an artificial neural network based gaze tracking system which can be customized to individual users. A three layer feed forward network, trained with standard error back propagation, is used to determine the position of a user's gaze from the appearance of the user's eye. Unlike other gaze trackers, which normally require the user to wear cumbersome headgear, or to use a chin rest to ensure head immobility, our system is entirely non-intrusive. Currently, the best intrusive gaze tracking systems are accurate to approximately 0.75 degrees. In our experiments, we have been able to achieve an accuracy of 1.5 degrees, while allowing head mobility. In its current implementation, our system works at 15 hz. In this paper we present an empirical analysis of the performance of a large number of artificial neural network architectures for this task. Suggestions for further explorations for neurally based gaze trackers are presented, and are related to other similar artificial neural network applications such as autonomous road following.

Shumeet Baluja is supported by a National Science Foundation Graduate Fellowship. This research was supported by the Department of the Navy, Office of Naval Research under Grant No. N00014-93-1-0806. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the National Science Foundation, ARPA, or the U.S. government.

Keywords

Gaze Tracking, Artificial Neural Networks, Human Computer Interaction, Machine Vision, Real-Time Vision, Input Modality, Facial Feature Tracking

1. Introduction

The goal of gaze tracking is to determine where a subject is looking from the appearance of the subject's eye. The interest in gaze tracking exists because of the large number of potential applications. Three of the most common uses of a gaze tracker are as an alternative to the mouse as an input modality [Ware & Mikaelian, 1987], as an analysis tool for human-computer interaction studies [Nodine et. al, 1992], and as an aid for the handicapped [Ware & Mikaellian, 1987].

Viewed in the context of machine vision, successful gaze tracking requires techniques to handle imprecise data, noisy images, and a potentially infinitely large image set. The most accurate gaze tracking has come from intrusive systems. These systems either use devices such as chin rests to restrict head motion, or require the user to wear cumbersome equipment, ranging from special contact lenses to a camera placed on the user's head to monitor the eye. The system described here attempts to perform non-intrusive gaze tracking, in which the user is neither required to wear any special equipment, nor required to keep his/her head still.

2. Gaze Tracking

2.1. Traditional Gaze Tracking

In standard gaze trackers, an image of the eye is processed in three basic steps. First, the specular reflection of a stationary light source is found in the eye's image. Second, the pupil's center is found. Finally, the relative position of the light's reflection to the pupil's center is calculated. The gaze direction is determined from information about the relative positions, as shown in Figure 1. In many of the current gaze tracker systems, the user is required to remain motionless, or wear special headgear to maintain a constant offset between the position of the camera and the eye.

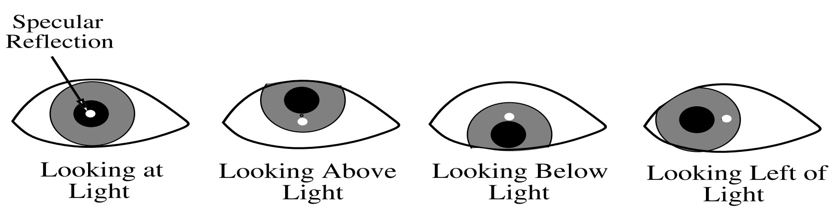

Figure 1. Relative position of specular reflection and pupil. This diagram assumes that the light is placed in the same location as the observer (or camera).

2.2. Artificial Neural Network Based Gaze Tracking

One of the primary benefits of an artificial neural network based gaze tracker is that it is non-intrusive; the user is allowed to move his head freely. In order to account for the shifts in the relative positions of the camera and the eye, the eye must be located in each image frame. In the current system, the right eye is located by searching for the specular reflection of a stationary light in the image of the user's face. This can usually be distinguished by a small bright region surrounded by a very dark region. The reflection's location is used to limit the search for the eye in the next frame. A window surrounding the reflection is extracted; the image of the eye is located within this window, as shown in Figure 2.

To determine the coordinates of the point the user is looking at, the pixels of the extracted window are used as the inputs to the artificial neural network. The forward pass is simulated in the ANN, and the coordinates of the gaze are determined by reading the output units. The output units are organized with 50 output units for specifying the X coordinate, and 50 units for the Y coordinate. A gaussian output representation, similar to that used in the ALVINN autonomous road following system [Pomerleau, 1993], is used for the X and Y axis output units. Gaussian encoding represents the network's response by a Gaussian shaped activation peak in a vector of output units. The position of the peak within the vector represents the gaze location along either the X or Y axis. The number of hidden units and

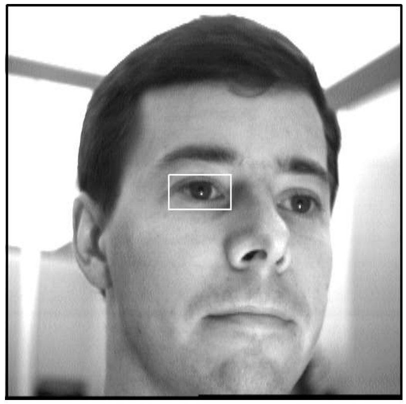

Figure 2. A window of pixels, centered on the user's right eye, is used as the input to the ANN.

the structure of the hidden layer necessary for this task are explored in section 3.

The training data is collected by instructing the user to visually track a moving cursor. The cursor moves in a predefined path. The image of the eye is digitized, and paired with the (X,Y) coordinates of the cursor. A total of 2000 image/position pairs are gathered. All of the networks described in this paper are trained with the same parameters for 260 epochs, using standard error back propagation. This training procedure is described in greater detail in the next section.

3. The ANN Implementation

In designing a gaze tracker, the most important attributes are accuracy and speed. The need for balancing these attributes arises in deciding the number of connections in the ANN, the number of hidden units needed, and the resolution of the input image. This section describes several architectures tested, and their respective performances.

3.1. Examining Only the Pupil and Cornea

Many of the traditional gaze trackers look only at a high resolution picture of the subject's pupil and cornea. Although we use low resolution images, our first attempt also only used an image of the pupil and cornea as the input to the ANN. Some typical input images are shown below, in Figure 3. The size of the images is 15x15 pixels. The ANN architecture used is shown in Figure 4. This architecture was used with varying numbers of hidden units in the single hidden layer; experiments with 10, 16 and 20 hidden units were performed.

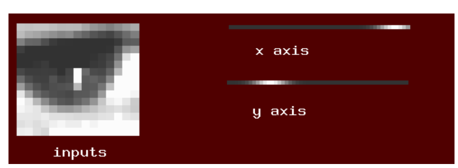

Figure 3. 15 x 15 Input to the ANN. Target outputs also shown.

As mentioned before, 2000 image/position pairs were gathered for training. The cursor automatically moved in a zig-zag motion horizontally across the screen, while the user visually tracked the cursor. In addition, 2000 image/position pairs were also gathered for testing. These pairs were gathered while the user tracked the cursor as it followed a vertical zig-zag path across the screen. See Appendix A for a diagram of cursor movements. The results reported in this paper, unless noted otherwise, were all measured on the 2000 testing points. The results for training the ANN on the three architectures mentioned above as a function of epochs is shown in Figure 5. Each line in Figure 5 represents the average of three ANN training trials (with random initial weights) for each of the two users tested.

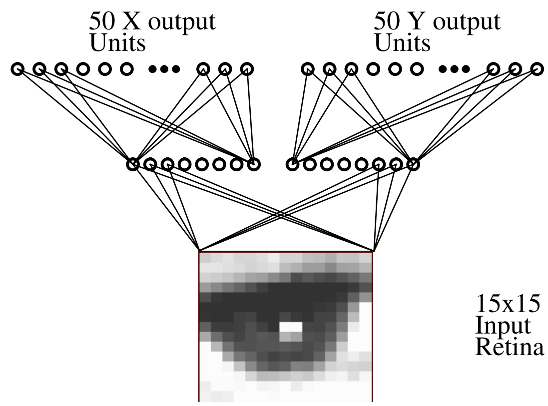

Figure 4. The ANN architecture used. A single, divided, hidden layer is used.

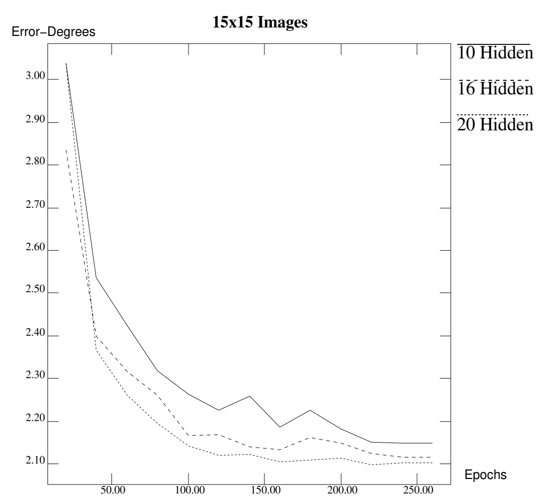

Figure 5. Error vs. Epochs for the 15x15 images. Errors shown for the 2000 image test set. Each line represents three ANN trainings per user; two users are tested.

Using this system, we were able to reduce the average error to approximately 2.1 degrees, which corresponds to 0.6 inches at a comfortable sitting distance of approximately 17 inches. In addition to these initial attempts, we have also attempted to use the position of the cornea within the eye socket to aid in making finer discriminations. These experiments are described in the next section.

3.2. Using the Eye Socket for Additional Information

In addition to using the information present from the pupil and cornea, it is possible to gain information about the subject's gaze by analyzing the position of the pupil and cornea within the eye socket. Two sets of experiments were performed using the expanded eye image. The first set used the network described in the next section. The second set of experiments used the same architecture shown in Figure 4, with a larger input image size. A small sample of images used for training is shown below, in Figure 6.

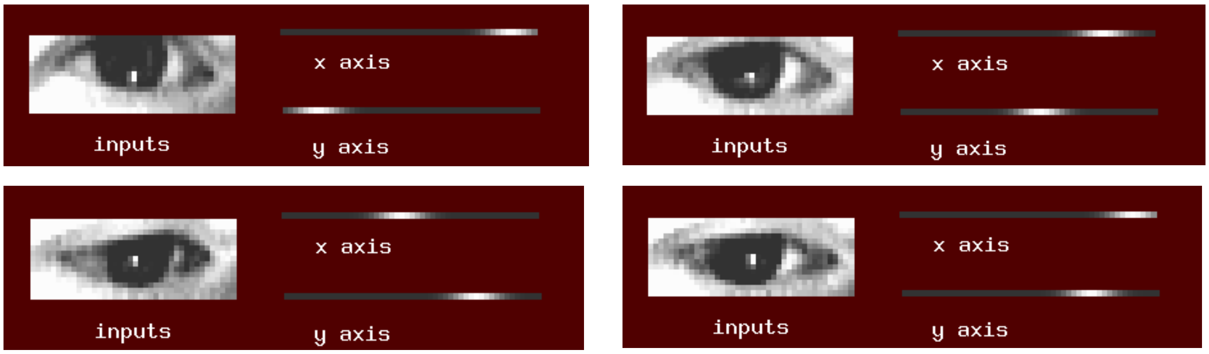

Figure 6. Images of the pupil and the eye socket, and their corresponding target outputs. 15 x 40 input image shown.

3.2.1. Using a Single Continuous Hidden Layer

One of the remaining issues in creating the ANN to be used for analyzing the position of the gaze is the structure of the hidden unit layer. In this study, we have limited our exploration of ANN architectures to simple 3 layer feed-forward networks. In the previous architecture (using 15 x 15 images) the hidden layer was divided into 2 separate parts, one for predicting the x-axis, and the other for the y-axis. Selecting this architecture over a fully connected hidden layer makes the assumption that the features needed for accurate prediction of the x-axis are not related to the features needed for predicting the y-axis. In this section, this assumption is tested. This section explores a network architecture in which the hidden layer is fully connected to the inputs and the outputs. Figure 7 shows the ANN architecture graphically.

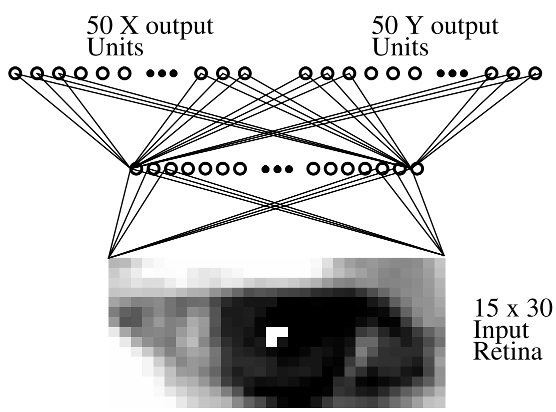

Figure 7. The ANN architecture used. A single, continuous, hidden layer is used. 15 x 30 input retina shown.

In addition to deciding the architecture of the ANN, it is again necessary to decide on the size of the input images. Several input sizes were attempted, 15x30, 15x40 and 20x40. Surprisingly, the 20x40 input image did not provide the most accuracy. Rather, it was the 15x40 image which gave the best results. Figure 8 provides three charts showing the performance of three image sizes as a function of the number of hidden units and epochs.

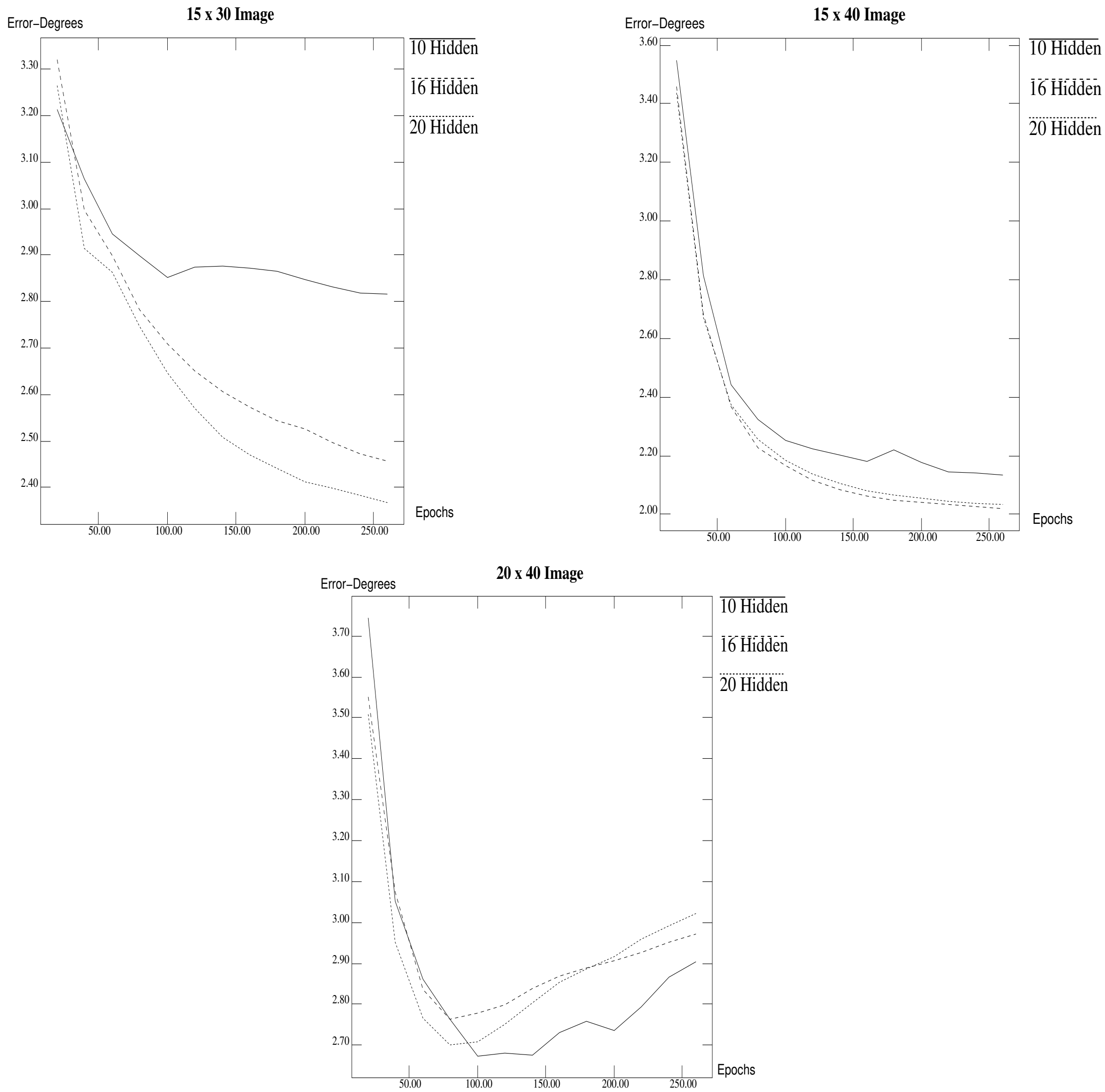

Figure 8. Performance of 15x30, 15x40, and 20x40 input image sizes as a function of epochs and number of hidden units. Each line is the average of 3 runs. Data points taken every 20 epochs, between 20 and 260 epochs.

The accuracy achieved by using the eye socket information, for the 15x40 input images, is better than using only the pupil and cornea. In particular, the 15x40 input retina worked better than both the 15x30 and 20x40. Nevertheless, another factor which should be considered when deciding the size of the input image is the average speed of the gaze tracker using each image size. Increasing the image size not only increases the burden of transferring larger portions of the image from the digitizer, but also increases the number of connections in the neural network. This has the potential to increase both the training time and the forward simulation time. This study has not yet addressed the problems associated with rapid training. However, as this system incorporates real time vision and interaction with users, minimizing forward simulation delays is crucial. The timings for several input image sizes and number of hidden units are given below, in Table 1. The set of tests shown is not complete; however, the result provide an indication of the potential time penalties incurred by the differences in ANN architectures. Although the tests to judge accuracy were done with 10, 16 and 20 hidden units, the tests shown in Table 1 are for 4, 10, 20, & 30 hidden units to emphasize the potential differences image size and number of connections can make.

The effects of increasing the number of connections is much more pronounced for the 20x40 image size than the 15x30 image size. Possible reasons for this, other than the increased number of computations because of the larger number of connections, may be very implementation and computer dependent. Possible factors outside the ANN architecture include cache size, memory size, and image acquisition and digitization time. The timings are measured on a Sun SPARC Station 10, with 32 MB of memory, and a Datacell S2200 frame grabber.

Table 1: Average time for forward Pass of 100 Images - Using 15x30, 15x40 and 20x40 Input Image Sizes

Image Size	Hidden Units	Number of Connections	Time to Process 100 Images	Cycle Rate (hz.)
15 x 30	4	2304	6.5 s.	15.4
	10	5610	6.5 s.	15.4
	20	11120	6.5 s.	15.4
	30	16630	6.6 s.	15.2
15 x 40	4	2904	6.7 s.	15.0
	10	7110	6.7 s.	15.0
	20	14120	7.0 s.	14.3
	30	21130	8.5 s.	11.7
20 x 40	4	3704	6.7 s.	15.0
	10	9110	7.3 s.	13.7
	20	18120	9.2 s.	10.9
	30	27130	11.4 s.	8.8

3.2.2. Using a Divided Hidden Layer

The final set of experiments which were performed were with 15x40 input images and 3 different hidden unit architectures: 5x2, 8x2 and 10x2. The hidden unit layer was divided in the manner described in the first network, shown in Figure 4. Two experiments were performed, with the only difference between experiments being the selection of training and testing images. The first experiment was similar to the experiments described previously. The training and testing images were collected in two different sessions, one in which the user visually tracked the cursor as it moved horizontally across the screen and the other in which the cursor moved vertically across the screen. The training of the ANN was on the “horizontally” collected images, and the testing of the network was on the “vertically” collected images. In the second experiment, a random sample of 1000 images from the horizontally collected images and a random sample of 1000 vertically collected images were used as the training set. The remaining 2000 images from both sets were used as the testing set. The second method yielded reduced tracking errors. If the images from only one session were used, the network was not trained to accurately predict gaze position independently of head position. As the two sets of data were collected in two separate sessions, the head positions from one session to the other would have changed slightly. Therefore, using both sets should have helped the network in two ways. First, the presentation of different head positions and different head movements should have improved the ability of the network to generalize. Secondly, the network was tested on images which were gathered from the same sessions as it was trained. The use of mixed training and testing sets will be explored in more detail in section 3.2.3.

The results of the first and second experiments are presented here, see Figure 9. In order to compare this architecture with the previous architectures mentioned, it should be noted that the performance of this architecture, with 10 hidden units, more accurately predicted gaze location than the architecture mentioned in section 3.2.1, in which a single continuous hidden layer was used. In comparing the performance of the architectures with 16 and 20 hidden units, the performances were very similar. Another valuable feature of using the divided hidden layer is the reduced number of connections decreases the training and simulation times. This architecture operates at approximately 15hz. with 10 and 16 hidden units, and slightly slower with 20 hidden units.

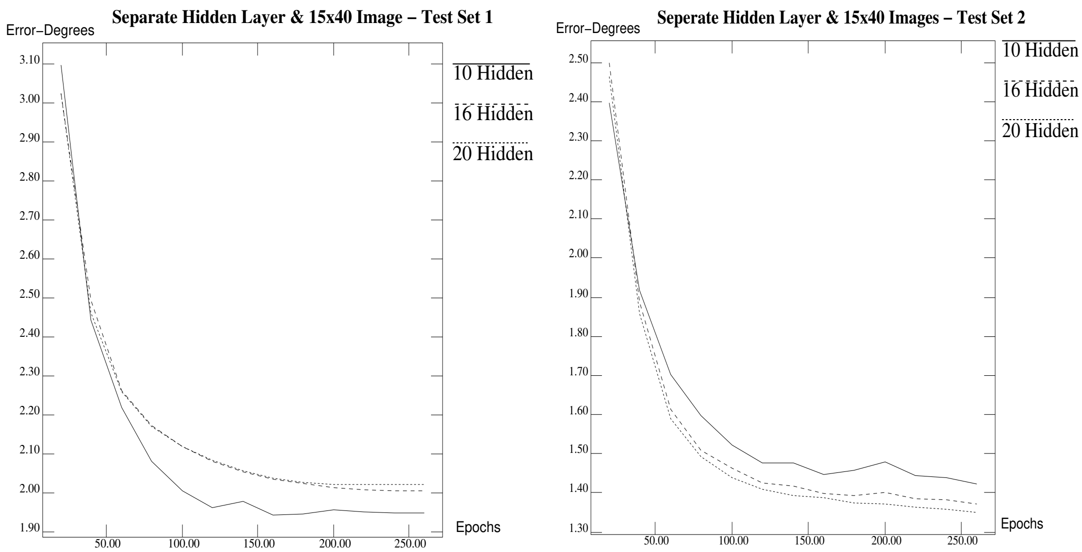

Figure 9. (Left) The average of 2 users with the 15x40 images, and a divided hidden layer architecture, using test setup #1: horizontal and vertical sets are used for training and testing, respectively. (Right) The average performance tested on 5 users, with test setup #2: training and testing set are disjoint sets drawn randomly from the horizontal and vertical image sets. Each line represents the average of three ANN trainings per user per hidden unit architecture.

3.2.3. Mixed Training and Testing Sets

It was hypothesized, above, that there are two reasons for the improved performance of a mixed training and testing set. First, the network's ability to generalize is improved, as it is trained with more than a single head position. Second, the network is tested on images which are similar, with respect to head position, as those on which it was trained. In this section, the first hypothesized benefit is examined in greater detail using the experiments described below.

Four sets of 2000 images were collected. In each set, the user had a different head position with respect to the camera. The first two sets were collected as previously described. The first set of 2000 images (horizontal train set 1) was collected by visually tracking the cursor as it made a horizontal path across the screen. The second set (vertical test set 1) was collected by visually tracking the cursor as it moved in a vertical path across the screen. For the third and fourth image sets, the camera was moved, and the user was seated in a different location with respect to the screen than during the collection of the first training and testing sets. The third set (horizontal train set 2) was again gathered from tracking the cursor's horizontal path, while the fourth (vertical test set 2) was from the vertical path of the cursor.

Three tests were performed. In the first test, the ANN was trained using only the 2000 images in horizontal training set 1. In the second test, the network was trained using the 2000 images in horizontal training set 2. In the third test, the network was trained with a random selection of 1000 images from horizontal training set 1, and a random selection of 1000 images of horizontal training set 2. The performance of these networks was tested on both of the vertical test sets. The results are reported below, in Figure 10. The last experiment, in which samples were taken from both training sets, provides more accurate results when testing on vertical test set 1, than the network trained alone on horizontal training set 1. When testing on vertical test set 2, the combined network performs almost as well as the network trained only on horizontal training set 2.

These three experiments provide evidence for the network's increased ability to generalize if sets of images which contain multiple head positions are used for training. These experiments also show the sensitivity of the gaze tracker to movements in the camera; if the camera is moved between training and testing, the errors in simulation will be large

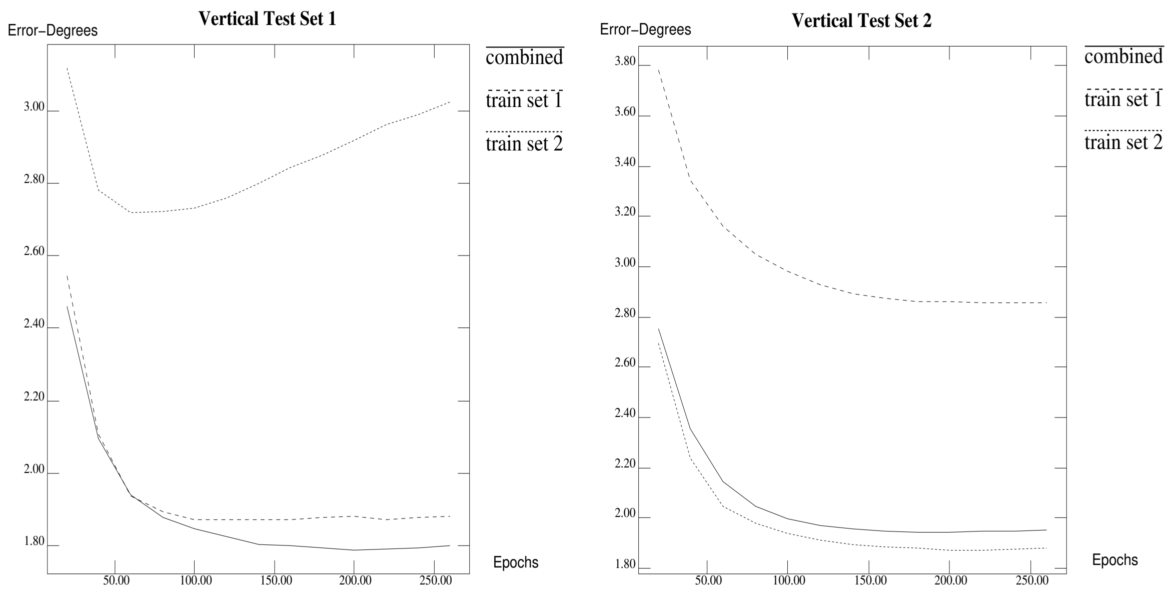

Figure 10. Comparing the performance between networks trained with only one head position (horizontal train set 1 & 2) and a network trained with both.

3.3. A Brief Look Into the Hidden Units

In watching the weights develop through the training of the ANN, easily recognizable images of the eye appear very early in almost all of the weights into the hidden units. However, as training progresses, the images become less defined. One possible explanation is that as the units represent increasing number of positions of the eye, the images become less recognizable. The weights entering the hidden units, after training, are shown in Figure 11. There is a recognizable image of an eye in the weights of several units.

The technique used to visualize the contents of the hidden units is to add a term to the weight update of the standard back propagation rule which not only takes into account the direction of the gradient in selecting the next move, but also considers the gradient of spatially close neighbors. Of course, the contribution of the neighbor's gradient is very small in comparison to its own gradient. If this revised weight update rule was not used, the images would be quite a bit more distorted. The distortion increases as the contribution of the neighbor's weights decreases. It should be noted, however, that with neighbor smoothing the performance of the network in predicting gaze location decreases, as the gradient is not always directly followed.

4. Using the Gaze Tracker

The experiments described to this point have used static test sets which are gathered over a period of several minutes, and then stored for repeated use. Using the same test set has been valuable in gauging the performance of different ANN architectures. However, a useful gaze tracker must produce accurate on-line estimates of gaze location. The use of an "offset table" can increase the accuracy of on-line gaze prediction. The offset table is a table of corrections to

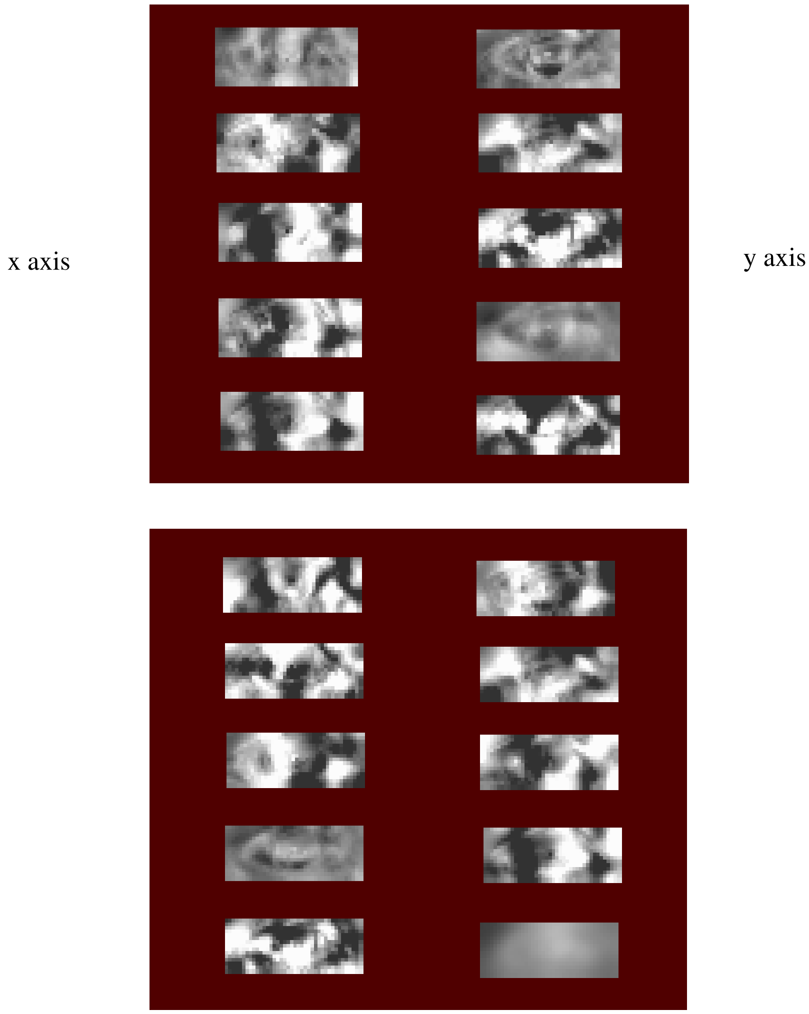

Figure 11. Weights entering the hidden units. (Top) Hidden Units for the 5 x 2 Hidden Layer (Divided Hidden Layer) Architecture. (Bottom) Hidden units for the 10 x 1 Hidden Layer (Single, Continuous Hidden Layer). Note the "blurred" images of the eye present in the weights of several hidden units.

the output made by a gaze tracker. The network's gaze predictions for each image are hashed into the 2D offset-table, which performs an additive correction to the network's prediction. The offset table is filled after the network is fully trained. The user manually moves and visually tracks the cursor to regions in which the ANN is not performing accurately. The offset table is updated by subtracting the predicted position of the cursor from the actual position. This procedure can also be automated, with the cursor moving in a similar manner to the procedure used for gathering testing and training images. However, manually moving the cursor can help to concentrate effort on areas where the ANN is not performing well; thereby reducing the time required for offset table creation.

With the use of the offset table, the current system works at approximately 15 hz. The best accuracy we have achieved is 1.5 degrees. Although we have not yet matched the best gaze tracking systems, which have achieved approximately 0.75 degree accuracy, our system is non-intrusive, and does not require the expensive hardware which many other systems require. We have used the gaze tracker in several forms; we have used it as an input modality to replace the mouse, as a method of selecting windows in an X-Window environment, and as a tool to report gaze direction, for human-computer interaction studies.

The gaze tracker is currently trained for 260 epochs, using standard back propagation. Training the 8x2 hidden layer network using the 15x40 input retina, with 2000 images, takes approximately 30-40 minutes on a Sun SPARC 10 machine.

5. Conclusions and Future Directions

We have created a non-intrusive gaze tracking system which is based upon a simple artificial neural network. Unlike other gaze-tracking systems which use traditional methods, such as a edge detection and circle fitting, this system develops its own features for successfully completing the task. The system's average on-line accuracy is 1.7 degrees. It has successfully been used in human-computer interaction studies and as an input device. Many extensions to the system are possible, some of the most intriguing are presented below.

One of the largest problems in existing eye trackers is their inability to handle user motion, for example the user's eye may leave the field of view of the camera. To address the problem of user motion, a pan-tilt mobile camera can be used. The motion of the camera can either be controlled by a separate ANN or through more traditional techniques.

Currently, the heuristic used to find the eye in the image of the face is to locate a bright spot surrounded by dark regions. However, neural networks have been applied to facial feature tracking [Hutchinson, 1990] [Debenham, 1991] and this technology may help here.

When using low resolution images, using only the pupil and cornea as inputs to the ANN does not provide enough information for accurate gaze tracking. In order to obtain more information from the appearance of the eye, we have used the position of the cornea in the eye-socket. One of the drawbacks of this method is that it makes the eye tracker less invariant to head position. For example, when the user looks at the same position on the screen with two different head positions, the cornea's position in the eye-socket can change dramatically. However, if only the pupil and cornea are examined, the relative change of the specular reflection is independent of small movements of the head [Starker, 1990]. One method of addressing this problem of keeping as much accuracy as possible with as much head position invariance as possible, is training on multiple head positions. This has been explored previously in this paper, and is returned to in the following paragraphs.

One of the potential drawbacks in attempts to make the ANN head position invariant is the large amount of data which must be collected. In the current system, data collection requires approximately 3 minutes of the user visually tracking the cursor. In this time, 2000 images of the user's eye paired with the position of the cursor are gathered. If the system were to be invariant to distance from the screen, and relative position with respect to the screen, more training image/gaze location pairs would have to be gathered. Images should include those in which the user is situated at different depths away from the screen, and different positions relative to the screen and camera.

A second method of maintaining position invariance in the eye tracking system is through the addition of extra inputs units to represent the head position. Because the camera used in this system has a relatively wide field of view, the same image from which the user's eye is extracted can be used to extract information about the head position. Similar techniques may not be feasible in other gaze tracking systems without the addition of extra hardware; many systems require a very high resolution image focused on the subject's cornea, and cannot maintain information of the relative position of the cornea in the eye or of the position of the eye with respect to the screen. A simple technique in which to incorporate head position in the ANN system is to use the information of where the eye is located in the image. For example, in Figure 2, the eye is located very close to the center of image. As the camera currently used is stationary, if the eye were located anywhere else on the screen, it would give a good indication to the head position relative to the camera and the screen.

In order to rapidly train the neural network for new users, a potential method may be to use a multiple network architecture, as was used in the MANIAC autonomous road following system [Jochem et. al, 1993], and is commonly used in connectionist speech recognition systems [Waibel et. al, 1990]. In the modular system for autonomous road following, several smaller "expert" networks were trained on different road types, i.e. one lane, two lane etc. An arbitrating network, which resided "on top" of the expert networks, is used to select which of the expert networks is yielding the best response to the current road or to combine the response of several expert networks. In an analogous manner, expert networks can be trained on the eye images of different users. Arbitration between experts could involve an arbitration network which receives input from the expert networks, as is the case in the MANIAC system. Alternatively, arbitration could be through the use of metrics which estimate the output reliability [Pomerleau, 1993]. Preliminary results have shown that the use of a modular network topology yields noticeable performance improvements. A benefit of the modular network approach is that each expert network can be trained independently of the others.

As another attempt to make the same network robust to a variety of people, preliminary experiments have also shown that training a large network with the images of several user's eyes improves the performance for each user. In comparison to the modular network approach described in the previous paragraph, the entire network must be trained on all images, while in the modular system, only the arbitration network needs to be trained on all images.

One of the criticisms which can be made about this system and the ALVINN autonomous vehicle steering system is that context is not preserved from one image to the next. In particular, each image is examined without considering any context which could have been developed in the previous image examined; the prediction of the gaze relies only on the current image. Perhaps a method of improving the system would be to use feedback connections which preserve some of the information from previously seen images. This may aid in limiting the set of possible responses for prediction of gaze location in future images.

With the additions described above, we hope to increase the system's accuracy without the addition of any intrusive hardware. Although we do not have as much invariance to head position as is desired, head position is not unnaturally restrained, and the user does not wear any extraneous equipment. This already makes the connectionist gaze tracker much less intrusive than many existing systems. We would like to test the viability of entirely replacing the mouse with the connectionist gaze tracker. Other potential uses for the system include aiding disabled people in interacting with their environment, and as a tool for data collection in psychological and human-computer interaction experiments.

6. Acknowledgments

The authors would like to gratefully acknowledge the help of Kaari Flagstad, Tammy Carter, Greg Nelson, and Ulrike Harke for letting us scrutinize their eyes, and being "willing" subjects. Profuse thanks are also due to Henry Rowley for aid in revising this paper. Finally, thanks are due to Todd Jochem who helped create the ANN software used for simulation and training, and to Roy Maxion for his discussions.

7. References

Debenham, R.M. & S.C.J. Garth (1991), "The Detection of Eyes Using Radial Basis Functions". In Proceedings of the *1991 International Conference of Artificial Neural Networks ICANN-91*. Amsterdam, Netherlands, North-Holland.

Hutchinson, R.A. (1990), "Development of an MLP feature location technique using preprocessed images". In proceedings of *International Neural Networks Conference*, 1990. Kluwer.

Jochem, T.M., D.A. Pomerleau, C.E. Thorpe (1993), "MANIAC: A Next Generation Neurally Based Autonomous Road Follower". In *Proceedings of the International Conference on Intelligent Autonomous Systems* (IAS-3).

Nodine, C.F., H.L. Kundel, L.C. Toto & E.A. Krupinksi (1992) "Recording and analyzing eye-position data using a microcomputer workstation", *Behavior Research Methods, Instruments & Computers* 24 (3) 475-584.

Pomerleau, D.A. (1991) "Efficient Training of Artificial Neural Networks for Autonomous Navigation," *Neural Computation 3*:1, Terrence Sejnowski (Ed).

Pomerleau, D.A. (1993) *Neural Network Perception for Mobile Robot Guidance*. Kluwer Academic Publishing.

Pomerleau, D.A. (1993) "Input Reconstruction Reliability Estimation", *Neural Information Processing Systems 5*. Hanson, Cowan, Giles (eds.) Morgan Kaufmann, pp. 270-286.

Starker, I. & R. Bolt (1990) "A Gaze-Responsive Self Disclosing Display", In *CHI-90*. Addison Wesley, Seattle, Washington.

Waibel, A., Sawai, H. & Shikano, K. (1990) "Consonant Recognition by Modular Construction of Large Phonemic Time-Delay Neural Networks". *Readings in Speech Recognition*. Waibel and Lee Eds.

Ware, C. & Mikaelian, H. (1987) "An Evaluation of an Eye Tracker as a Device for Computer Input", In J. Carrol and P. Tanner (ed.) *Human Factors in Computing Systems - IV.* Elsevier.

8. Appendix A: Cursor Movements

To gather training and testing images, the cursor is automatically moved in a pre-defined, easily predictable path. The user visually tracks the cursor. The image of the eye is paired with the (x,y) coordinates of the cursor. Either the path is a horizontal zig-zag or a vertical zig-zag. The paths are shown below, in Figure A.1.

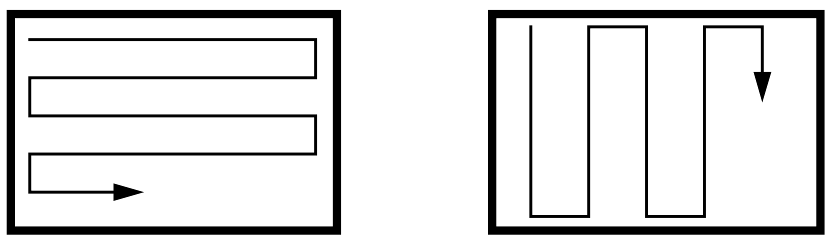

Figure A.1. (Left) Horizontal zig-zag. (Right) Vertical zig-zag. Outer square represents the screen. The lines with an arrow represent the cursor's path.