

Adaptive eye-gaze tracking using neural-network-based user profiles to assist people with motor disability

Anaelis Sestin, PhD; Malek Adjouadi, PhD; * Mercedes Cabrerizo, PhD; Melvin Ayala, PhD; Armando Barreto, PhD
Center for Advanced Technology and Education, Department of Electrical and Computer Engineering, Florida International University, Miami, FL

Abstract—This study developed an adaptive real-time human-computer interface (HCI) that serves as an assistive technology tool for people with severe motor disability. The proposed HCI design uses eye gaze as the primary computer input device. Controlling the mouse cursor with raw eye coordinates results in sporadic motion of the pointer because of the saccadic nature of the eye. Even though eye movements are subtle and completely imperceptible under normal circumstances, they considerably affect the accuracy of an eye-gaze-based HCI. The proposed HCI system is novel because it adapts to each specific user's different and potentially changing jitter characteristics through the configuration and training of an artificial neural network (ANN) that is structured to minimize the mouse jitter. This task is based on feeding the ANN a user's initially recorded eye-gaze behavior through a short training session. The ANN finds the relationship between the gaze coordinates and the mouse cursor position based on the multilayer perceptron model. An embedded graphical interface is used during the training session to generate user profiles that make up these unique ANN configurations. The results with 12 subjects in test 1, which involved following a moving target, showed an average jitter reduction of 35%; the results with 9 subjects in test 2, which involved following the contour of a square object, showed an average jitter reduction of 53%. For both results, the outcomes led to trajectories that were significantly smoother and apt at reaching fixed or moving targets with relative ease and within a 5% error margin or deviation from desired trajectories. The positive effects of such jitter reduction are presented graphically for visual appreciation.

Key words: artificial neural network, assistive technology, eye-gaze tracking, human-computer interface, jitter reduction, mouse cursor trajectory, rehabilitation, saccadic eye movement, severe motor disabilities, user profile.

INTRODUCTION

Computer interface research has known respectable growth in the last decade, and the deployed assistive technology tools have enabled persons with disabilities to harness the power of computers and access the variety of resources available to all [1–3]. Despite recent advances, challenges still remain for extending access to users with

Abbreviations: ANN = artificial neural network, CPU = central processing unit, EGMPC = Eye-Gaze Mouse-Pointer Control, EGT = eye-gaze tracking, HCI = human-computer interface, MLP = multilayer perceptron, MM = Metric Monitoring, NSF = National Science Foundation.

*Address all correspondence to **Malek Adjouadi, PhD; Florida International University, Department of Electrical and Computer Engineering, 10555 W Flagler Street, Miami, FL 33174; 305-348-3019; fax: 305-348-3707.**

Email: adjouadi@fiu.edu

DOI: 10.1682/JRRD.2007.05.0075

severe motor disabilities. A number of human-computer interfaces (HCIs) have integrated eye-gaze tracking (EGT) systems as one possible way for users to interact with the computer through eye movement [4–6]. Other studies have integrated different modalities, such as eye gazing, gesture recognition, and speech recognition, to allow the user more flexible interactions with computers [7–8].

Unfortunately, the use of EGT systems as the primary mechanism for controlling the mouse pointer and the graphical user interface has been complicated by inaccuracies arising from saccadic eye movement. Such natural involuntary movement of the eye results in sporadic, discontinuous motion of the pointer, or “jitter,” a term used herein to generally refer to any undesired motion of the pointer resulting from a user’s attempts to focus on a target, regardless of the specific medical or other reason or source of the involuntary motion. Some attempts to increase the accuracy of mouse cursor control through eye-gazing activity involve the integration of a complementary technology such as electromyogram [9–11]. However, these approaches require the users to wear devices such as electrodes, which may be uncomfortable.

To make matters worse, the jitter effect generally varies in degree as a function of inherent user characteristics, which vary from one user to another. The jitter effect across multiple users may be so varied that a single control scheme to address each user’s jitter effect would likely require significant and complex processing requirements that would impose unrealistic constraints on the demand for real-time processing. As a result, the system would then be unable to control the mouse pointer position in real time and would add cost to such processing power. But without real-time control and processing, users would experience noticeable delays between eye movement and positioning of the pointer, which would be frustrating.

Some studies attempt to resolve the jitter dilemma based on Fitt’s law, which defines the time needed to move the pointer to a target area MT as a function of the distance to A (amplitude of the movement) and size of the target W , as in **Equation 1**.

$$MT = a + b \cdot \log_2\left(\frac{A}{W} + 1\right) \quad , \quad (1)$$

where a corresponds to the start/stop time of the device and b is the inherent speed of the device.

This type of study facilitates selection of a target by enlarging the target size. For instance, Špakov and Min-

iotas suggest the use of dynamic target expansion for menu item selection, which would require developing specialized applications [12]. Also, Bates and Istance use a full-screen zoom-in technique to increase eye-based interaction performance [13]. However, one downside to this approach is the loss of contextual information, since the peripheral region of the zoomed area is lost. Another approach uses a fisheye lens to expand the target and the area surrounding it [14]. The approach has two stages, one to activate the fisheye lens and another to lock and click on the target. Each stage lasts in accordance with predefined dwelling times. Since this technique is based on the user fixating a target, a conflict may arise when the user fixates on some item solely to obtain information and the program interprets this fixation as an input command. This problem is known as the Midas touch. Furthermore, other studies use a combination of eye gazing and standard computer input devices, such as the keyboard, for selecting a target [15].

The objective of our research endeavor is to develop an eye-gaze-based HCI system that accommodates and adapts to different users through artificial neural network (ANN) design customization and configuration. Generally speaking, the methodology relies on a user profile that customizes eye-gaze tracking by using neural networks that, up to this point, have been used on eye-image localization and gaze-positioning algorithms [16–17]. The user-profile concept will facilitate universal access to computing resources and, in particular, enable an adaptable interface for a wide range of individuals with severe motor disability, such as amyotrophic lateral sclerosis, muscular dystrophy, spinal cord injury, and other disabilities characterized by lack of muscle control or body movement. More specifically, each individual user has a unique ANN configuration that helps smooth the trajectory of eye movement based on his or her unique user profile generated during the training session. After gaining experience with the proposed EGT-based system, the individual can conduct additional training sessions to fine-tune the specific ANN configuration to optimally minimize jitter, since jitter characteristics can change as experience is gained. This constitutes another adaptive feature of the system that will allow continual improvements and therefore enhanced practicality.

This approach does not change the appearance of the image display on the computer monitors. Instead, it reduces the mouse pointer jitter, which makes its trajectory smoother and, consequently, allows the user to better

control the eye-based pointing device. At the same time, this approach keeps selecting or clicking icons as standard as possible with instantaneous response.

To develop a neural network to effectively reduce the jitter of the mouse due to eye movement, we implemented the following steps:

1. We analyzed the original mouse cursor trajectory without ANN intervention.
2. On the basis of step 1, we defined a suitable configuration of the ANN.
3. We acquired data and extracted training patterns for training the ANN.
4. We trained the ANN.
5. We evaluated the jitter-reduction algorithm with the ANN.

METHODS

System Overview

The EGT-based HCI, as illustrated in **Figure 1**, is based on a remote eye-gaze setup that is less intrusive (passive) than the head-mounted version and thereby frees the user from any physical constraint. The system consists of a central processing unit (CPU) for eye-data acquisition, another CPU for user interaction (stimulus computer), an eye monitor, a scene monitor, an eye-imaging camera, and an infrared light source. The integrated EGT system in this research was developed around the

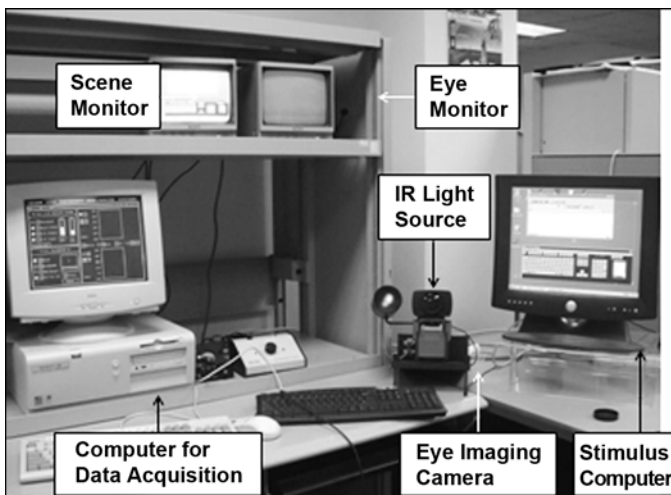


Figure 1. Eye-gaze-based human-computer interface components. IR = infrared.

ISCAN[®] ETL-500 technology (ISCAN, Inc; Burlington, Massachusetts) [18].

Figure 2 illustrates the system setup during a working session with a nondisabled subject using a standard headrest to prevent any abrupt head movement and with an individual with a motor disability in a wheelchair.

In EGT-based systems, the direction of a user's gaze accordingly positions a mouse pointer on the display of the stimulus computer. More specifically, the EGT system reads and sends eye-gaze position data, in the form of a 512×512 pixel matrix, to the stimulus computer, where the eye-gaze data is translated into display coordinates



Figure 2. Eye-gaze-based human-computer interface setups. (a) Nondisabled subject using standard headrest to prevent abrupt head movement and (b) individual with motor disability in wheelchair.

that guide the position of the mouse pointer. To that end, remote EGT systems often track the reflection of an infrared light from the limbus (i.e., the boundary between the white sclera and the dark iris of the eye), pupil, and cornea together with an eye image to determine the point of regard (i.e., point of gaze) as an (x, y) coordinate point with respect to the visual field [19–21]. The eye coordinates are determined based on the pupil/corneal reflection disparity with an accuracy typically $>0.3^\circ$ over a $\pm 20^\circ$ horizontal and vertical range, as reported in the ISCAN manual. For this particular HCI system, the field of view is the monitor of the stimulus computer. These coordinates are then translated to determine the position and movement of the mouse pointer. This task is performed by the Eye-Gaze Mouse-Pointer Control (EGMPC) application, which was developed to receive raw eye-gaze data as input and to output the equivalent mouse pointer actions (mouse movement, left click, etc.). The data-conversion algorithm implemented for this interface is based on the least square line method [22]. **Figure 3** shows a screenshot of the EGMPC application.

The user can disable the gazed mouse-pointer control at any time by selecting the “Disconnect” option on the File menu or tool bar, by pressing Ctrl+d, or by voice command. Similarly, the user can restore gazed mouse-pointer control by selecting the “Connect” option in the File menu or the tool bar, by pressing Ctrl+t, or by voice command. When the gazing control is disabled, the mouse cursor does not respond to eye movements; gazing control can be established by dwelling, or staring, at the “Connect” button in the tool bar.

Mouse Cursor Trajectory Analysis

Controlling eye position consciously and precisely at all times is relatively difficult because of the eye’s jerky behavior. Therefore, predicting the actual position of the mouse is easier if its trajectory is subdivided into smaller sections, which allows the trajectory of the mouse to be described linearly. The size of a subset is defined by the x

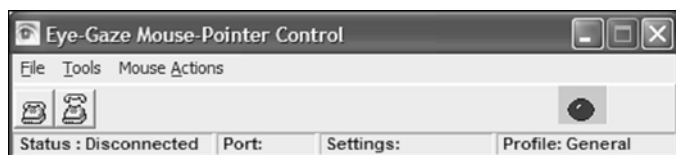


Figure 3. Screenshot of Eye-Gaze Mouse-Pointer Control application.

and y ordinates generated by the EGMPC module during a time interval Δt , which is later used as input to estimate the actual gaze point (desired output value). Hence, defining the size of Δt was a crucial step in determining the number of mouse coordinates that would be used as input to the ANN.

The EGT data are generated at a frequency of 60 Hz, but to obtain the required inputs to the neural network, the input/output relation would require a sampling frequency <60 Hz in order to construct an $n \times 2$ ANN architecture. Simultaneously, the output needs to be generated at a frequency that (1) still guarantees a smooth mouse pointer movement perception and (2) permits sufficient input data to accurately determine the mouse pointer position.

Different sampling rates were tested, and the best empirical results were obtained at a frequency of 10 Hz. At this value, the trajectory of the mouse pointer is still relatively smooth and the size of the sampling window is now $60/10 = 6$, which is an acceptable number of reference points with which to compute the desired position of the mouse pointer. **Figure 4** shows how the trajectory of the mouse is divided into 6 points, where P_{Start} and P_{End} are the initial and final positions of the pointer, respectively. The three-dimensional (x, y, t) sequence of the mouse pointer in time frame Δt is illustrated in **Figure 5**.

Artificial Neural Network Architecture

The ANN architecture we sought relies on the multi-layer perceptron (MLP) model, also known as a supervised network since it requires a desired output (target) in order to learn. This type of network correctly maps the input (gaze coordinates) to the output (anticipated mouse pointer location) relationship based on historical data [23].

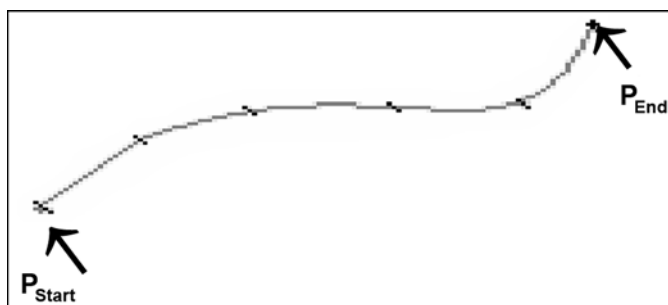


Figure 4. Mouse pointer trajectory is fragmented using time interval t . P_{Start} = initial position of pointer, P_{End} = final position of pointer.

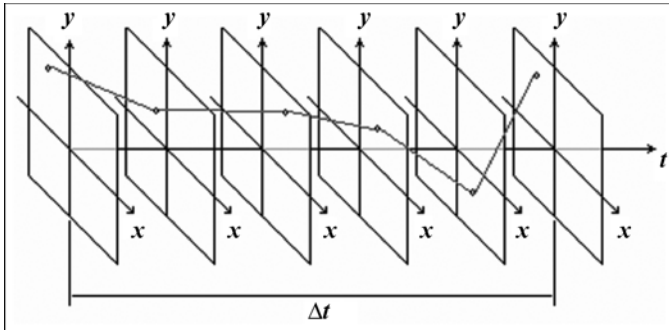


Figure 5. Mouse cursor sequence in time frame t .

Since the sampling frequency was defined to be one-tenth of a second, 6 samples, consisting of x and y ordinates, are collected for each sampling window. Consequently, the ANN contains a total of 12 neurons in the input layer, or 6 (x, y) positions.

The composition of the hidden layers was determined by testing the ANN with different numbers of hidden layers and units. The best results were obtained with one hidden layer that had 24 nodes and sigmoid activation functions. Since the outputs of the network are the x and y ordinates, only two output units are needed (x_{out} and y_{out}) in the last layer.

In summary, the ANN default configuration shown in **Figure 6** contains three layers: 12 input neurons, 24 hidden units, and 2 output units. The ANN was trained with the backpropagation algorithm. The default activation functions are (1) linear for the input layer, (2) logsig for the hidden layer, and (3) linear for the output layer.

The configuration of the network can be customized by changing the number of hidden layers and the number of neurons in the model according to the characteristics of the data. This change can be performed by selecting the Neural Network tab in the Application Settings window, as shown in **Figure 7**. Not all the ANN configuration parameters can be changed. Only the number of hidden units and the activation functions for the hidden and output layers can be modified. The training stopping conditions can also be changed.

Data Acquisition Phase and Training Pattern Extraction

A Metric Monitoring (MM) application was developed to assist in the collection of data and in the evaluation of implemented algorithms [24]. The MM software monitors the mouse cursor movements and computes a set

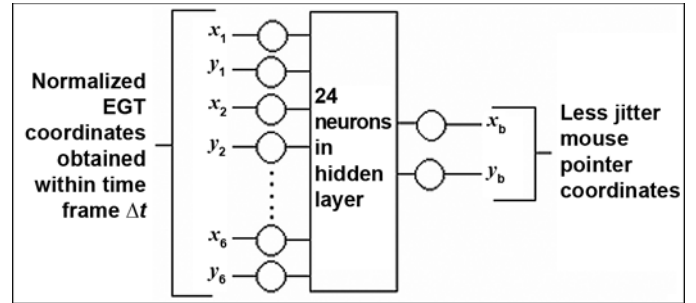


Figure 6. Default artificial neural network configuration. EGT = eye-gaze tracking.

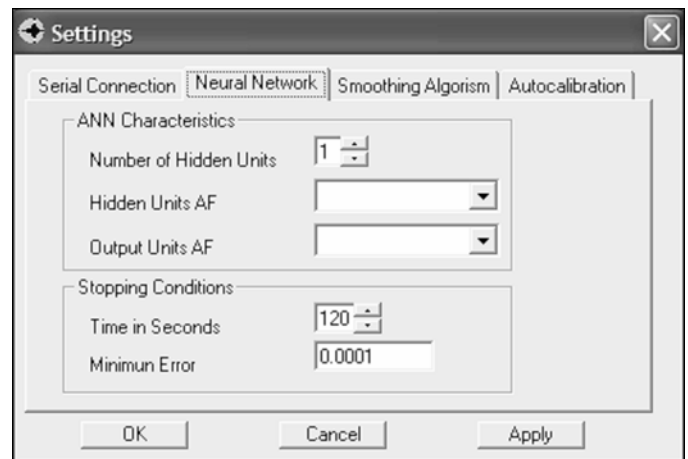


Figure 7. Screenshot of Application Settings window for artificial neural network configuration.

of indicators, which will be later used for jitter-reduction algorithm assessment.

The data collection process involves a moving target, such as a button, that is rendered on the display device of the stimulus computer (**Figure 8**) and that the user must follow throughout the entire session. As the user looks at the button, the mouse pointer coordinates generated by the EGMPC module are taken as the input training set, while the actual average display coordinates of the moving button (\bar{x}_b, \bar{y}_b) are taken as the ANN target set.

The training set is then divided into sample frames of one-tenth of a second. Each frame F_i can be written in a vector format as illustrated in **Equation 2**. Since the EGMPC module generates mouse pointer coordinates at a rate of 60 Hz, each segment corresponds to 6 separate coordinate pairs, shown as (x_1, y_1) through (x_6, y_6) , for a total of 12 input data values for each training pattern. A

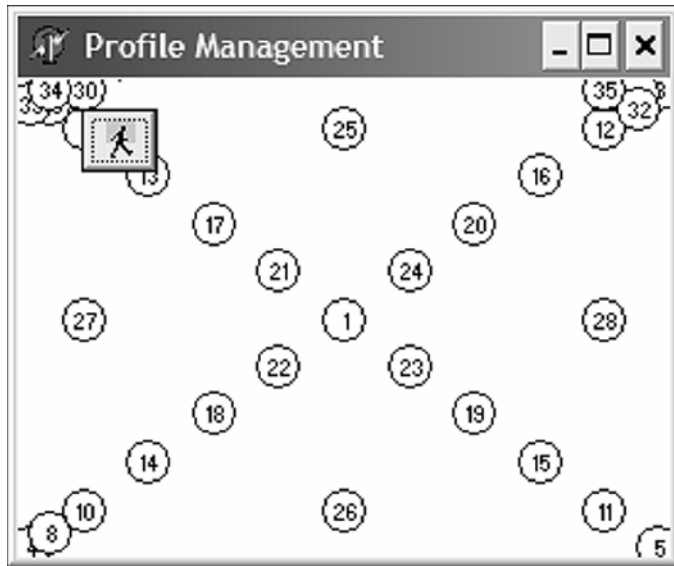


Figure 8.
Metric Monitoring graphical evaluation application.

nonoverlapping scrolling window is used for data collection, which simplifies the programming code and reduces the training set size.

$$F_i = [x_{i1} y_{i1}, x_{i2} y_{i2}, \dots, x_{i6} y_{i6}, \bar{x}_b \bar{y}_b] \quad (2)$$

The number of sample frames (n_F) depends on the duration of the data acquisition process and is defined as

$$n_F = \frac{t}{f_s} \quad (3)$$

where t is the duration of the data collection in seconds, and f_s is the sampling rate, which equals one-tenth of a second or 6 points. For instance, a recording section of 1 minute generates a total of 600 sample frames. Since each frame contains 6 (x, y) input points, the training set contains 3,600 points. This is illustrated below in **Equations 4–5**:

$$\frac{1 \text{ min} \cdot \frac{60\text{s}}{\text{min}}}{\frac{1}{10} \text{ s}} = 600 \text{ sample frames} \quad (4)$$

and

$$600 \text{ frames} \cdot 6 \text{ point/frame} = 3,600 \text{ points} \quad (5)$$

Network Training Phase

In the MLP model, the inputs are fed into the input layer, multiplied by interconnection weights, and then

passed into the first hidden layer. Within the first hidden layer, all the values input to the same neuron are summed, biased, and then processed by a nonlinear function (activation function). The data processed by the first hidden layer are again multiplied by interconnection weights, then summed and processed by the following layer. This process is repeated for the output layer, where the neural network is expected to produce the desired outcome.

The MLP acquires knowledge through backpropagation, a learning algorithm in which the input data is repeatedly presented or passed to the neural network. At the end of each iteration, an error is computed by comparing the output of the ANN with the desired outcome. This error is then fed back (backpropagated) to the neural network and used to adjust the weights such that the error decreases with each iteration. In this way, the ANN model gets closer and closer to its final configuration by adjusting the weights and biases for all the layers.

The user may specify or control the length of the training time by using the Application Settings window shown in **Figure 5**. Before the training starts, the user can change the stopping conditions of the learning algorithm by changing either the maximum training time or the minimum training error. Training of the ANN stops as soon as either of the two stopping conditions is met. The default stopping conditions are set to 3 minutes for the maximum training time and 0.0001 for the minimum mean square error. Furthermore, if the neural network is taking a long time to converge, the User Profile Management module provides the user the option of halting the training at any point by clicking the Stop Calculation button, which is represented by the boxed “x” in **Figure 9**.

Personalization of Artificial Neural Network to Accommodate User’s Variability

At this design stage, the proposed smoothing algorithm using ANN fails to address how jitter effects may vary widely between different users of the system. Specifically, the initialization of the EGT system, as proposed, may result in a trained neural network that performs inadequately with another user who was not involved in the initialization. Furthermore, the EGT system may also fail to accommodate single-user situations, because each individual may exhibit varying jitter characteristics over time with changing circumstances or operational environments.

To improve the performance results, we need to tailor ANN data by creating user profiles. To this end, data are collected for each user and applied for training the ANN.

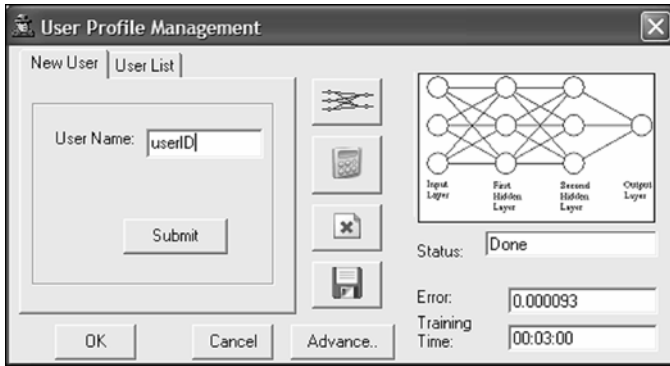


Figure 9. Screenshot of User Profile Management application.

The results from the training (i.e., weights and biases) are then saved as a user profile, which basically defines the EGT behavior of that particular user from a prior experience. The customized ANN based on these users' profiles will in time allow all users in the database (i.e., users with existing profiles) to interact with a computer in real time with minimized jitter effects. Customizing the HCI system does not require the user to have any knowledge of ANNs, much less the manner in which such networks are trained and structured. In other words, the reduction in jitter effects via the customized ANN is accomplished in a manner transparent to the user.

Furthermore, besides creating a new profile, users may also edit an existing profile. By retraining an existing ANN, the system adapts in time to the user as subtle differences are learned with each recorded experience. Thus, users may update an existing user profile to accommodate additional changes in jitter characteristics.

Time Required for Generation of User Profile

The amount of time needed to generate a user profile (which is the same as generating an individual ANN) is 5 minutes, broken down into 2 minutes for data collection and 3 minutes for training with a stopping condition. The user could increase the training period to exceed 3 minutes if needed. Empirical results show that 3 minutes of training for the ANN was sufficient since it yielded an 0.0004 mean square error. The ANN, however, always converges to an optimal solution after 6 minutes, on average, with a mean square error of 0.0001. We must emphasize that once the ANN or user profile is established for the individual user, any subsequent use of the interface by a user whose profile has

already been created will be performed in real time and with diminished jitter as previously experienced.

Indicators Used for Performance Evaluation

The intent of this study was to smooth the jitter in the trajectory while a user attempts to move his or her gaze from a given point to another point (without consideration as to which is the start point and which is the end point and, for that matter, where these points are on the screen). With this in mind, the degree of jitter measures the spread of the mouse coordinates and is computed from a given point to another by approximating the pointer trajectory (with jitter) with linear segments (S_K) consisting of 6 points (Module 6) each distanced temporally by one-tenth of a second in accordance with the 60 Hz sampling rate. **Figure 10** and **Equations 6–7** illustrate how jitter is computed. Module 6 is used for notational convenience, such that points d_0 through d_5 will be used again for all subsequent segments:

$$J_K = \frac{\sum_{i=1}^5 d_i \angle_{1,i} \angle_{0,5}}{d_{0,5}} \quad (6)$$

and

$$d_{j,i} = \sqrt{(X_i \angle X_j)^2 + (Y_i \angle Y_j)^2} \quad (7)$$

The degree of jitter was computed for each subtrajectory (J_K) and the results were averaged as defined in **Equation 8**:

$$\frac{1}{n} \sum_{k=0}^{n-1} J_K \quad (8)$$

With regard to the jitter metric, the Euclidean distance $d_{0,5}$ is considered the optimal trajectory, which means a straight line with no jitter. In this case, the equation of the degree of jitter yields zero.

RESULTS AND DISCUSSION

We conducted the experiments using the same MM application used during the preliminary training stage. The first test involved 12 subjects, 7 males and 5 females, ranging from 25 to 46 years of age. One of the subjects had a spinal cord injury and used the system setup

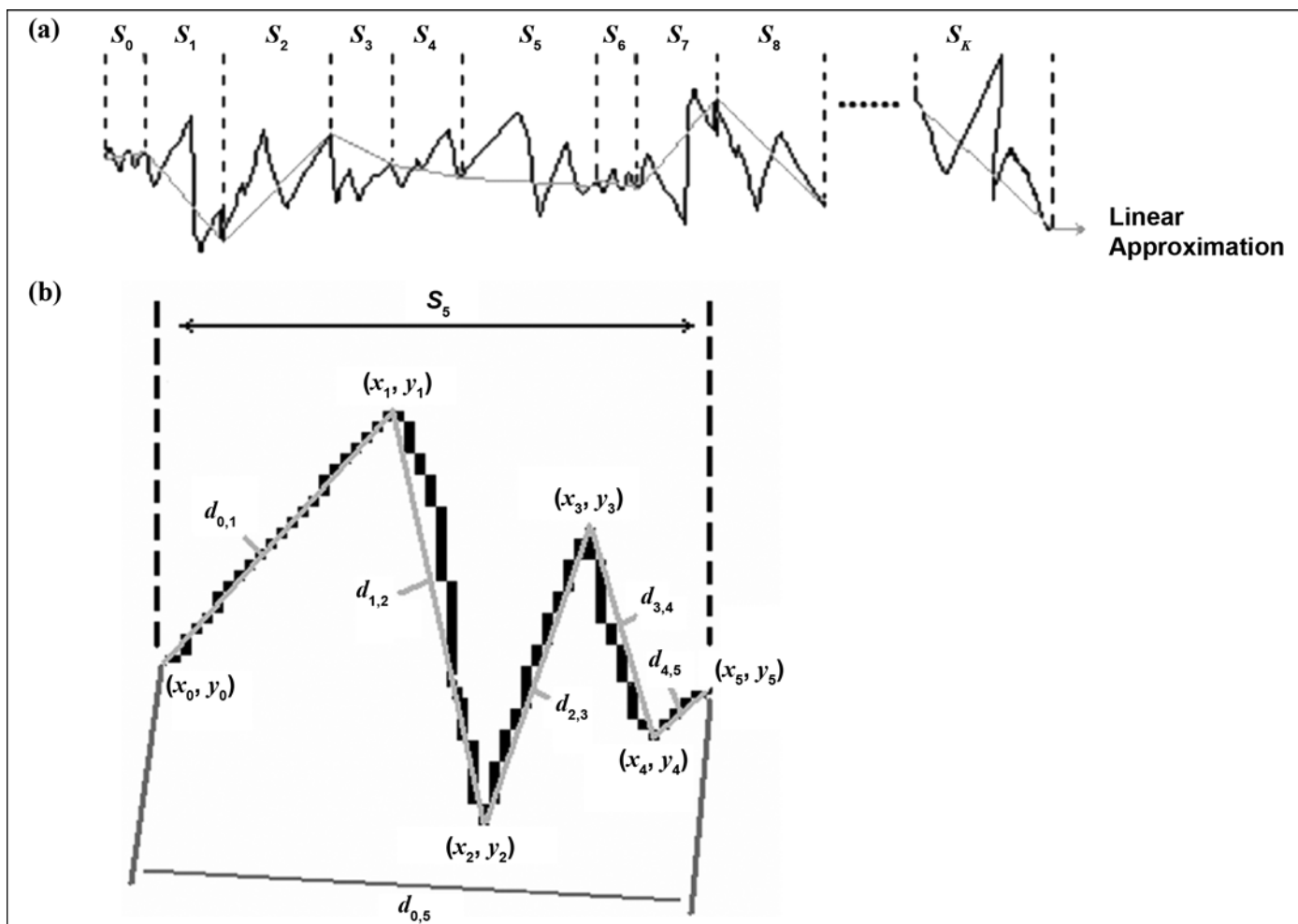


Figure 10.

Computing degree of jitter. (a) Trajectory of mouse pointer is parceled into 6 point segments (S_K). (b) Zoom-in of S_5 segment in (a); linear approximation of segment is defined by $d_{0,5}$ and represented by gray line.

shown in **Figure 2(b)**. (This person participated in all the experiments conducted for this study.) The remaining participants used the headrest as illustrated in **Figure 2(a)**.

The experiment used the MM application to collect the training data set and test the ANN. Each subject went through a three-step process. First, the training data set, also referred to as the raw data, was collected as the user gazed at a moving target that covered most of the screen area. The data set included the position of the mouse cursor, corresponding to the gaze coordinates, and the center coordinates of the button. The data collection process took an average of 2 minutes for each subject. Second, the training of the ANN started using the collected data, which required approximately 3 minutes to obtain the weights and biases of the ANN. This ANN defined the

user profile that was saved for future use. Third, data were again collected as the user gazed at the moving target but this time using the intervention of the personalized ANN. The subject gazed at a moving target in the third phase of the test for constancy purposes only.

We computed the degree of jitter for the raw data (J) and the processed data with ANN intervention (J_{ANN}), using **Equations 6–7**, as well as the ratio of improvement (RI) as given in **Equation 9**:

$$RI = \frac{(J_{RAW} \angle J_{ANN})}{(J_{RAW})} \quad (9)$$

The results from all subjects and the overall performance of the system, computed by averaging the results from all the users, are summarized in **Table 1**.

Table 1.

Jitter reduction using artificial neural network (ANN) across different subjects.

Subject	J	J_{ANN}	RI
1	0.424	0.301	0.290
2	0.290	0.136	0.531
3	0.546	0.322	0.410
4	0.439	0.358	0.185
5	0.528	0.367	0.304
6	0.625	0.332	0.469
7	0.304	0.174	0.427
8	0.546	0.331	0.394
9	0.305	0.218	0.286
10	0.420	0.275	0.345
11	0.215	0.150	0.301
12	0.372	0.293	0.213
Overall Jittering Degree (%)	41.8	27.1	35.1

J = raw jitter data, J_{ANN} = processed jitter data with ANN intervention, RI = ratio of improvement.

The results reveal an average of 35 percent reduction in jitter error when the EGT is supported with ANN intervention, which represents a substantial improvement in the use of eye gaze to control the mouse pointer. Experimenting with the control application and Web browsing, the mouse cursor was found to be more stable and easier to control since the trajectory was significantly smoother and could reach the target and click on it within the 5 percent error margin.

Furthermore, to test how the system adapts even further to the user every time the user's profile is edited (i.e., the ANN is retrained), we repeated the same test several times with the same user. For illustrative purposes, **Table 2** shows the results from two of the subjects. The trend of decreasing degree of jitter as a result of the ANN retraining can be observed in **Figure 11**.

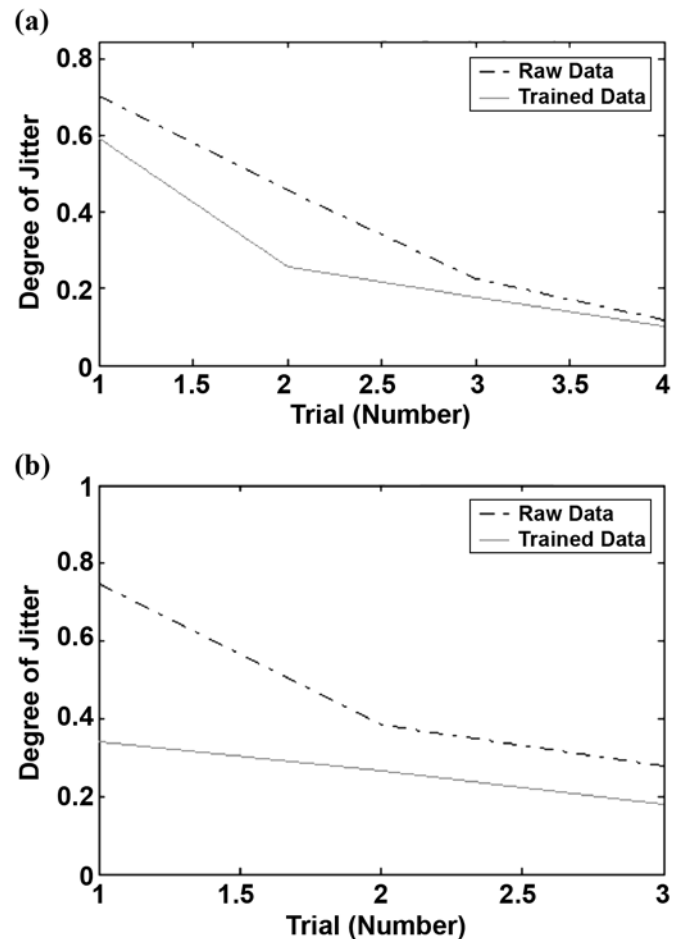
The results prove that as a user profile is edited and the ANN is retrained for the same user, the system further learns how to overcome the jitter behavior for that particular user. If the initial degree of jitter, before training the system and without any ANN intervention (J of first trial), and the final degree of jitter, after training the system several times and with ANN intervention (J_{ANN} of last trial), are compared for both subjects, the results reflect a 75.9 and 85.4 percent reduction in jitter error for subjects 1 and 2, respectively. Additionally, as in any other system, the more the user uses the system, the more the user gets familiarized with the interface.

Table 2.

Jitter reduction as artificial neural network (ANN) system adapted further to user characteristics. Subjects 1 and 2 shown for illustrative purposes.

Test No.	J	J_{ANN}	RI
Subject 1			
1	0.744	0.341	0.542
2	0.385	0.267	0.306
3	0.278	0.179	0.355
Subject 2			
1	0.704	0.593	0.157
2	0.454	0.257	0.434
3	0.226	0.179	0.207
4	0.117	0.103	0.123

J = raw jitter data, J_{ANN} = processed jitter data with ANN intervention, RI = ratio of improvement.

**Figure 11.**

Degree of jitter trend as artificial neural network is retrained for (a) subject 1 and (b) subject 2.

Given the complexity of the problem, assessing further the weights associated with the neural network was relevant in order to visualize the various outcomes given the diverse user population. We also intended to estimate the merit of using user profiles to minimize jitter. To do so,

we provided two different assessments: one using gray scale maps of the weights of the ANN (**Figures 12 and 13**) and the other using histograms of the same weights.

From the results shown in **Figures 12 and 13**, and given the striking variations of these weights between

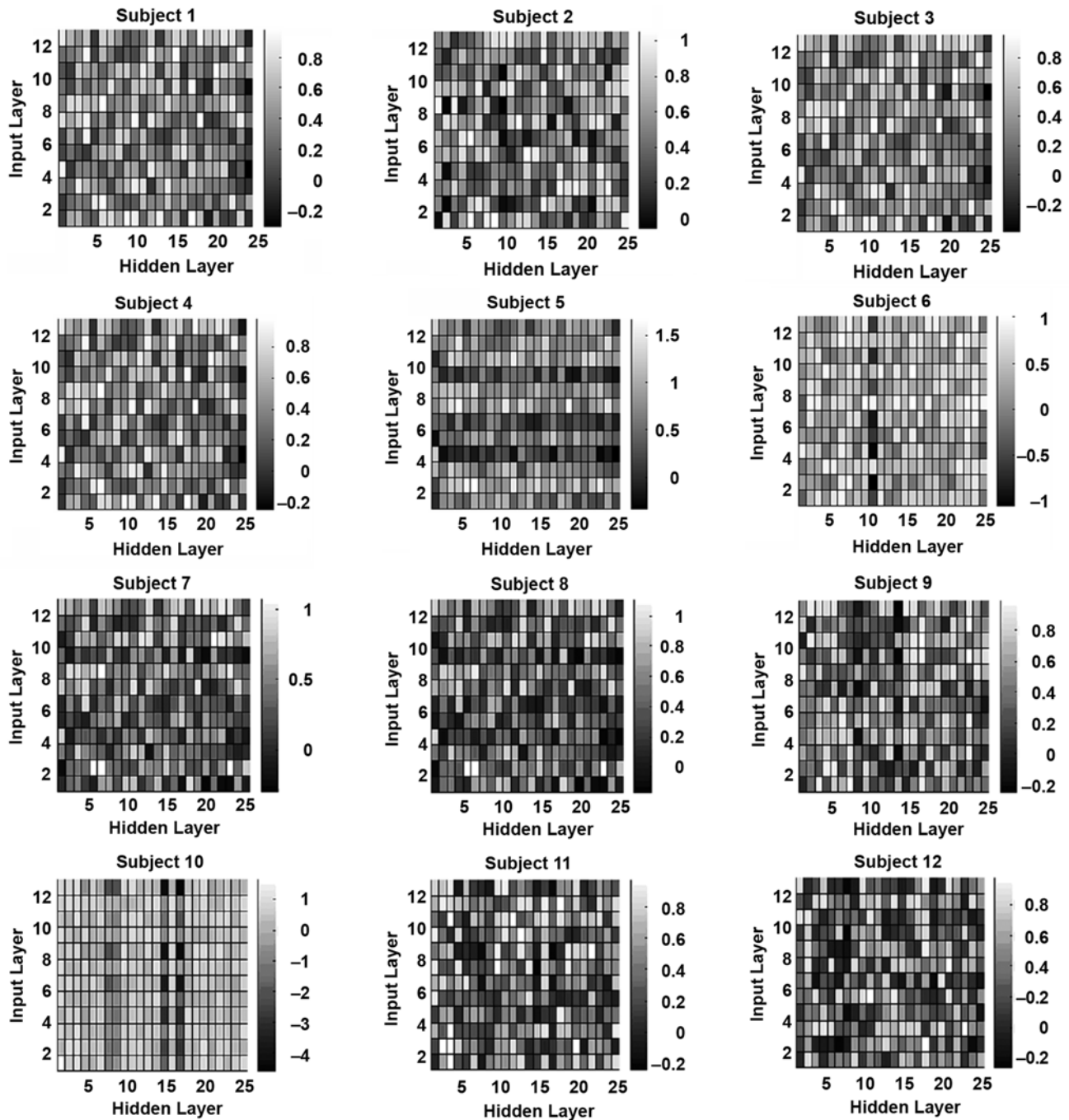


Figure 12.

Gray scale maps showing behavior of weights between input and hidden layers generated by artificial neural network for subjects 1 through 12.

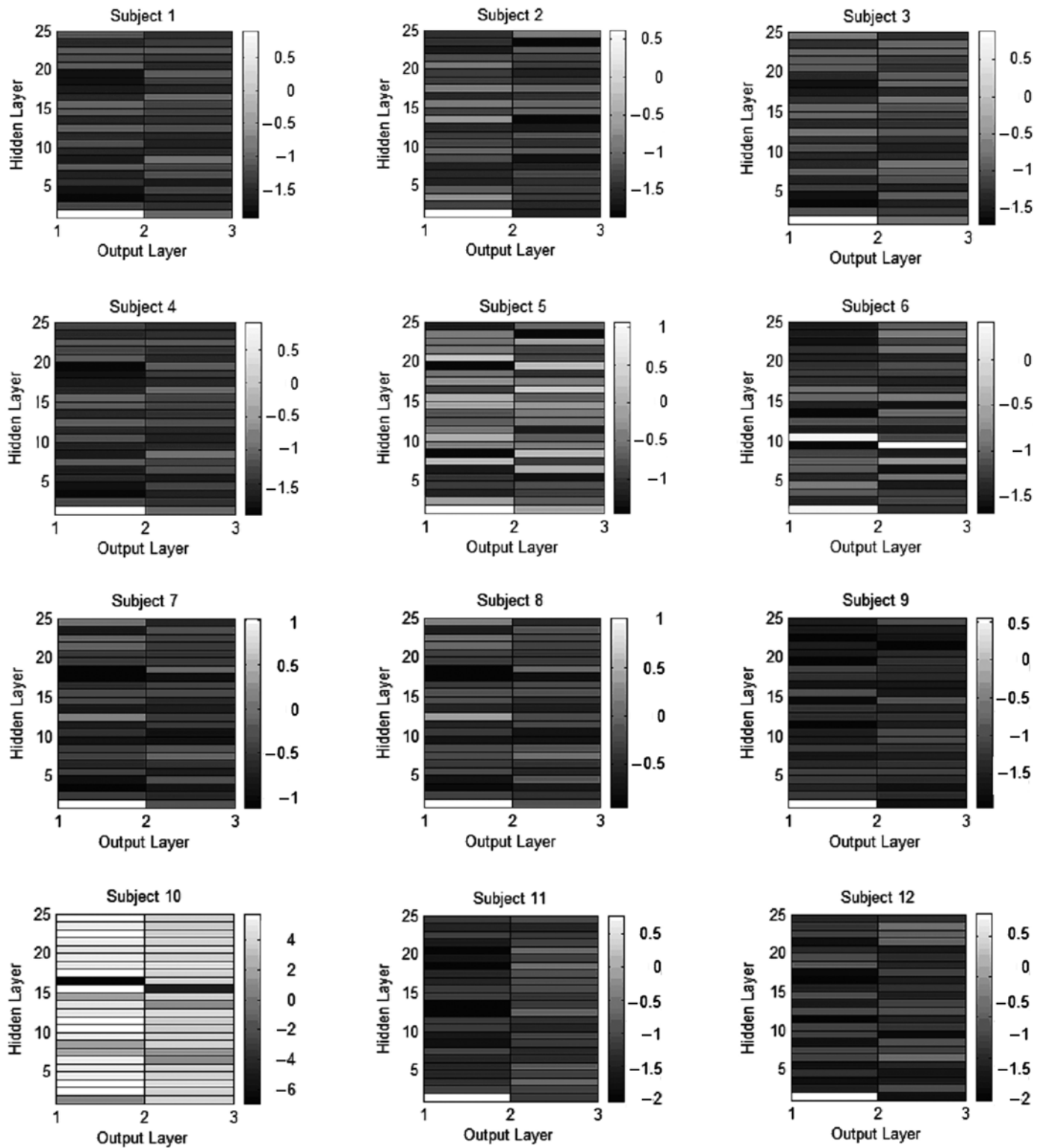


Figure 13.

Gray scale maps showing behavior of weights between output and hidden layers generated by artificial neural network for subjects 1 through 12.

users, creating a single ANN that would work (reduce jitter) for all users clearly would not be feasible. This outcome is what led to the creation of individualized profiles.

To further emphasize the need for individualized profiles, **Figure 14** shows the histograms generated for each of the user's profiles. From each histogram, major features were extracted, such as average, variance, standard deviation, skewness, kurtosis, energy, and power, and are

summarized in **Table 3**. Here again, determining a set of values that would have worked in an optimal fashion for the entire user population, in order to generalize the profile for all users, was not possible.

In **Table 3**, except for subject 10, most of the weight distributions have a negative skew; this means that the mass of the distribution is concentrated on the right side of the histogram. However, this assessment would not

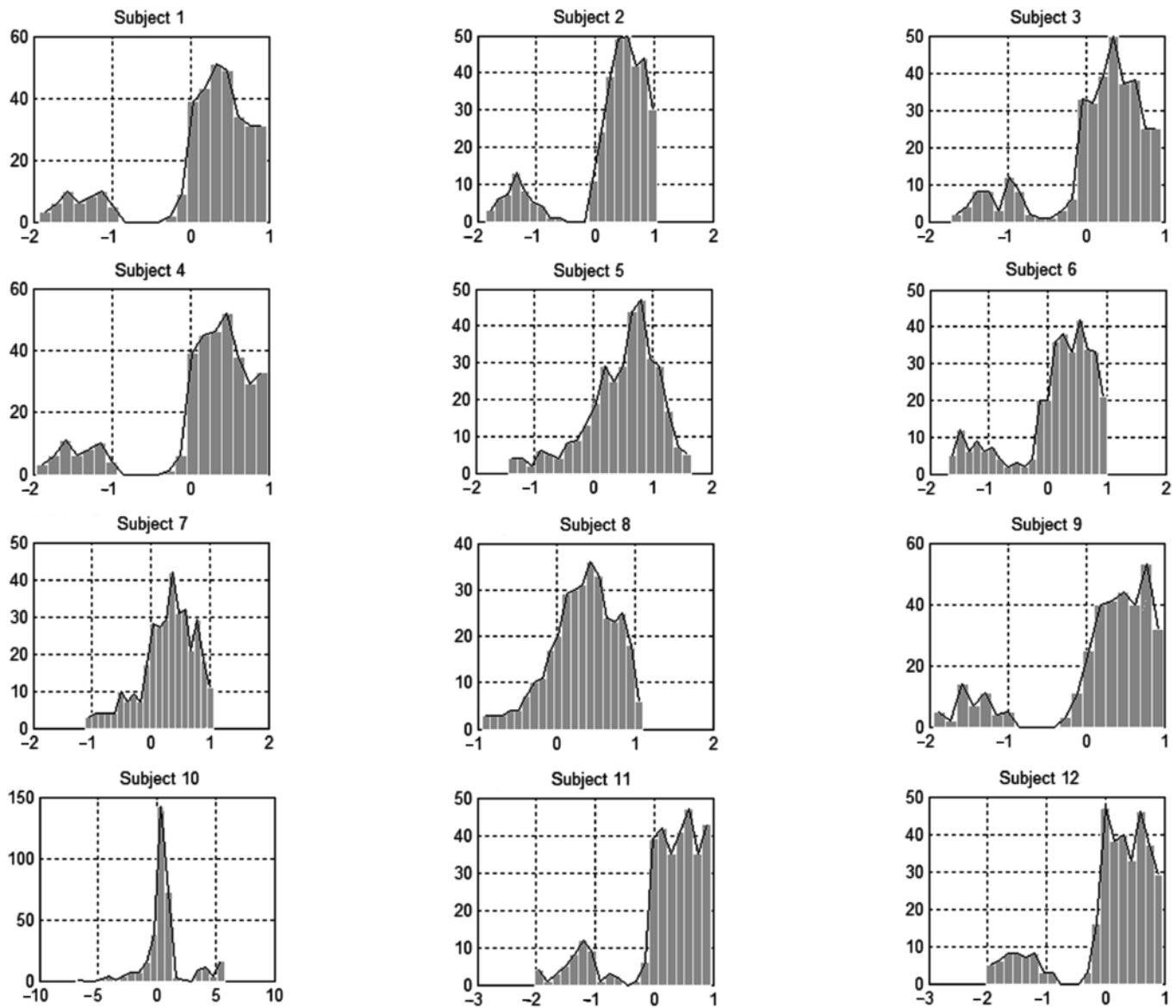


Figure 14. Distribution of weights generated by artificial neural network for subjects 1 through 12.

Table 3.

Features extracted from distribution of weights generated by artificial neural network for each subject.

Subject	Average	Variance (Biased)	Standard Deviation (Biased)	Skewness	Kurtosis	Energy	Power
1	0.162	0.472	0.687	-1.475	4.279	167.823	0.498
2	0.274	0.494	0.703	-1.566	4.428	191.958	0.570
3	0.164	0.367	0.606	-1.300	4.009	132.942	0.394
4	0.167	0.493	0.702	-1.513	4.351	175.544	0.521
5	0.495	0.359	0.616	-0.876	3.633	210.251	0.624
6	0.158	0.442	0.665	-1.222	3.600	157.421	0.467
7	0.286	0.211	0.459	-0.735	3.266	98.727	0.293
8	0.337	0.173	0.416	-0.575	3.087	96.338	0.286
9	0.201	0.527	0.726	-1.526	4.300	191.149	0.567
10	0.638	3.259	1.805	0.566	5.836	1,235.697	3.667
11	0.188	0.458	0.677	-1.447	4.421	166.416	0.494
12	0.122	0.523	0.723	-1.454	4.278	181.148	0.537

help in the final analysis given the variations in magnitude of the skewness measurement as well as of the other measurements, such as kurtosis, energy, and power.

We conducted a second test with nine (five females and four males) of the participants from test 1 in order to visualize graphically the jitter reduction results. In this case, the moving target followed a rectangular shape centered on the stimulus monitor. Once again, the subject was asked to gaze at a moving button for standardization purposes. As in the previous test, data were collected without and with the intervention of the ANN. However, in this case, the raw data were not used to train the ANN since the subjects involved in this test had already had a personalized profile created during test 1. **Figure 15** shows the graphs generated for each user without ANN and with ANN. This test was conducted with a 17 in. monitor with a screen resolution of 800×600 . Furthermore, as in test 1, we computed the degree of jitter for the raw data (J) and the processed data with ANN intervention (J_{ANN}), using **Equation 5** as well as the ratio of improvement as defined in **Equation 9**. The results are as shown in **Table 4**.

The mouse cursor trajectory plots reflect substantially improved control of the mouse pointer through eye gazing when the mouse cursor coordinates were computed based on the user's profile. With the intervention of the ANN, the pointer trajectory is now smoother; more impressive still is the fact that when substantial offsets occurred, they were still corrected by the ANN.

The results in **Table 4** reveal an averaged jitter reduction of 53 percent when the EGT is supported with ANN

intervention. Since the participants had already used the system in test 1, they were familiar with the system and some had retrained the ANN to further adapt the interface to their jittering characteristics. This retraining is reflected in the results for the degree of jitter, which is reduced even more in comparison with the results from test 1.

We used the data collected during test 2 to compute the accuracy of the proposed EGT-based system. The accuracy of the system is indicated by the disparity between the center of the moving button (B) target (x_B, y_B) and the actual coordinates of the mouse (M) pointer (x_M, y_M). This offset was defined as the Euclidean distance between the two points as given by **Equation 10**:

$$\text{Offset}_i = \sqrt{(x_{M_i} \angle x_{B_i})^2 + (y_{M_i} \angle y_{B_i})^2} \quad (10)$$

In this case, the trajectory of the mouse cursor was not subdivided as initially reported for test 1. In test 2, the offset was computed for all the points in the data set, averaged as in **Equation 11**. These offsets are given in **Table 5**.

$$\frac{1}{n} \sum_{i=1}^n \text{Offset}_i \quad (11)$$

The results reflect that the accuracy of the system reflected by Offset_{RAW} ranges from 16 to 25 pixels in a 19 in. monitor with a screen resolution of 800×600 . Also, the results indicate that the use of ANN profiles as reflected by Offset_{ANN} does not generate a significant difference in the offset values, since they varied between 16 and 18 pixels. The existing disparity between the

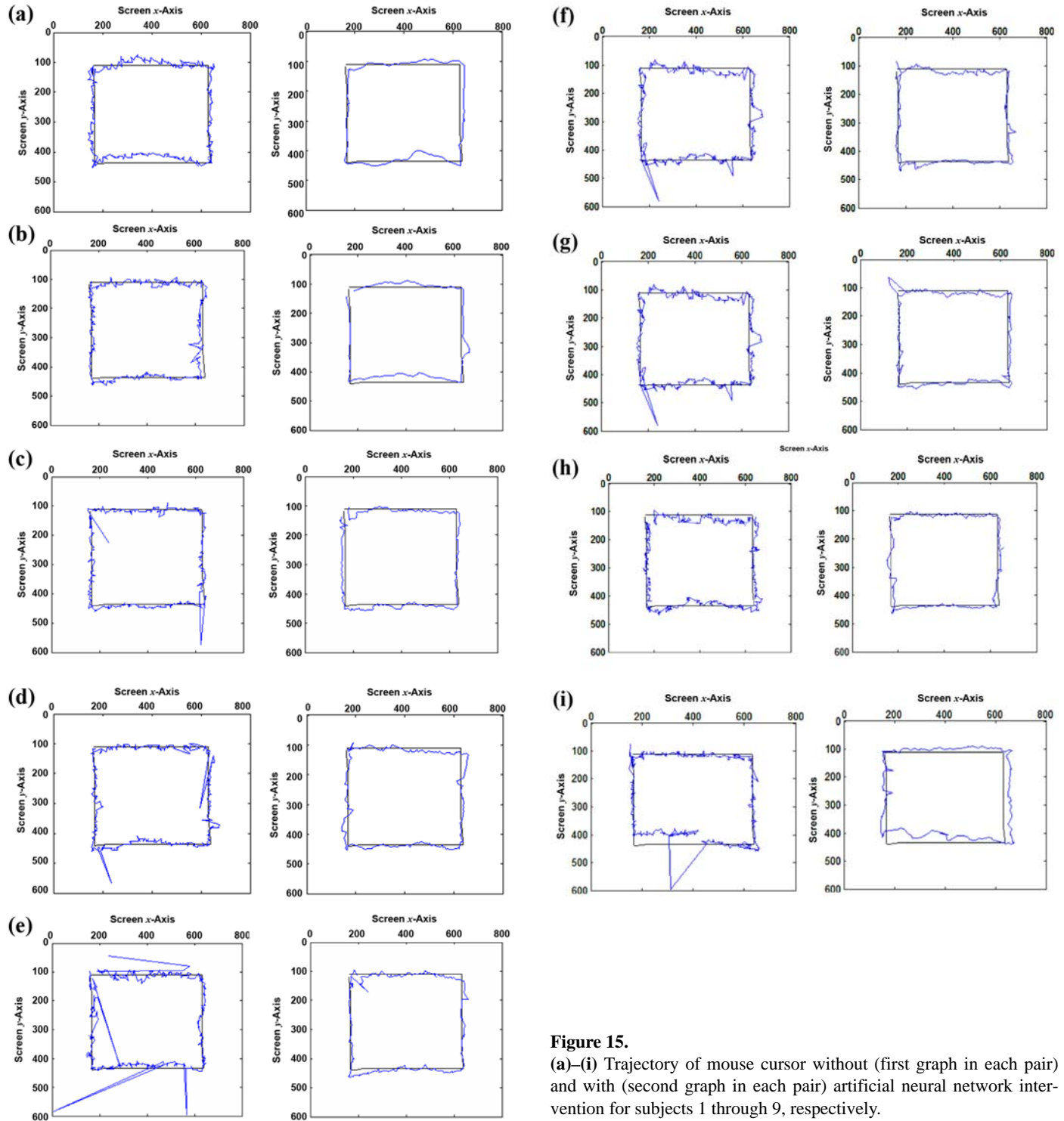


Figure 15.

(a)–(i) Trajectory of mouse cursor without (first graph in each pair) and with (second graph in each pair) artificial neural network intervention for subjects 1 through 9, respectively.

point of gaze and the position of the mouse cursor is inherited from the EGT system used in this interface. In terms of system setup, the subject is sitting 48 in. away from the computer screen, which is 14 in. wide. Given

that the accuracy ($\text{Offset}_{\text{ANN}}$) varies between 16 and 18 pixels, this translates to 0.33° to 0.37° .

In a similar experimental design reported in Kim and Varshney [25], the eye-tracker calibration process reached

Table 4.

Jitter reduction using customized artificial neural network (ANN) across different users in test 2.

Subject	J	J_{ANN}	RI
1	0.790	0.022	0.972
2	0.482	0.034	0.928
3	0.458	0.187	0.592
4	0.929	0.689	0.258
5	0.772	0.529	0.314
6	0.504	0.346	0.313
7	0.617	0.385	0.376
8	0.807	0.395	0.511
9	0.790	0.022	0.972
Average	0.670	0.323	0.533

J = raw jitter data, J_{ANN} = processed jitter data with ANN intervention, RI = ratio of improvement.

Table 5.

Disparity between center of target and mouse cursor in pixels before ($Offset_{RAW}$) and after ($Offset_{ANN}$) training of artificial neural network (ANN).

Subject	$Offset_{RAW}$	$Offset_{ANN}$
1	19	17
2	23	16
3	17	17
4	19	17
5	25	16
6	22	18
7	20	17
8	23	17

an accuracy of 30 pixels (or 0.75°) given the EGT interface setup on the basis of tracking 13 points that are displayed sequentially on the screen.

We must stress at this time that although the ANN has to some extent improved the offset error and provided a more consistent outcome across the subjects, the main objective of this study was to produce a steady and smooth trajectory not only for navigation but also for facilitating clicking actions on small icons or buttons on the computer screen.

A third experiment was conducted to assess the jitter reduction impact in the performance of a real-world task. The test involved six participants, three male and three female, who had also participated in tests 1 and 2. The subjects were asked to execute a simple task, such as clicking a stationary button, for 1 minute. During this time, the number of clicks performed inside (C_{IN}) and outside (C_{OUT}) the area of the button was recorded. The

button had a predefined area of 31×26 pixels. The recorded values were used to compute the click efficiency (C_{EFF} , **Equation 12**), which is defined as the ratio between the number of click events effectively triggered when the mouse pointer is over the desired location (C_{IN}) and the total number of clicks that are commanded by the user during the whole working session ($C_{IN} + C_{OUT}$):

$$C_{EFF} = \frac{C_{IN}}{C_{IN} + C_{OUT}} \quad (12)$$

Subjects executed the same task without and with the assistance of their user profile, and the click efficiency was computed for each case. We then compared the results by computing the RI as given in **Equation 13**. These results are given in **Table 6**.

$$RI = \frac{C_{EFF(ANN)} \angle C_{EFF(RAW)}}{C_{EFF(RAW)}} \quad (13)$$

The results revealed substantial improvement of 176 percent in click efficiency with the assistance of personalized ANN. Also, participants were able to perform more useful clicks when using their ANN profiles during each 1-minute trial. Therefore, target selection time was reduced significantly.

CONCLUSIONS

This study designed an adaptive, real-time assistive system as an alternative HCI that uses eye gaze only to facilitate computer access for individuals with severe motor disability. This type of assistive technology tool intends to broaden the functional capability of persons with motor disability [26] through the intervention of unique ANN configuration and individualized user profiles.

More specifically, this study focused on the implementation of an algorithm to smooth out abrupt and unwanted jerky behavior of the mouse cursor, as a result of the saccadic nature of eye movement, via the configuration of an ANN that minimized the jitter effect on the basis of user characteristics. These characteristics were extracted via the creation of user profiles through an embedded graphical interface. The smoothing algorithm resulted in an average jitter reduction of 35 to 53 percent, depending on the complexity of the experiment. Consequently, the trajectory of the mouse cursor was significantly smoother and could reach the target with improved accuracy within a 5 percent error or deviation margin.

Table 6.Click efficiency (C_{EFF}) improvements with artificial neural network (ANN) across different users.

Subject	$C_{IN(RAW)}$	$C_{OUT(RAW)}$	$C_{EFF(RAW)}$	$C_{IN(ANN)}$	$C_{OUT(ANN)}$	$C_{EFF(ANN)}$	RI
1	3	12	0.20	27	9	0.75	2.75
2	5	15	0.28	34	6	0.85	2.04
3	6	12	0.27	29	10	0.74	1.74
4	12	17	0.41	43	13	0.77	0.88
5	12	25	0.32	32	7	0.82	1.56
6	9	21	0.30	29	8	0.78	1.60
Average	8	17	0.297	32	9	0.785	1.76

C_{IN} = number of clicks inside area of button, C_{OUT} = number of clicks outside area of button, RAW = without ANN intervention, RI = ratio of improvement.

The separate dedicated management of each user profile allows the ANN to be trained and retrained for each user. Retraining the ANN involved updating the weights and biases, thereby building upon prior customization and configuration efforts. More generally, a user-profile-based approach to reducing jitter effects addresses the user-specific, or user-dependent, nature of jitter. Our experiments proved that after retraining the ANN three or four times, the system further learns the gazing behavior of that particular user and yields an 80 percent average reduction of the jittering behavior of the mouse cursor. The relevance of these results has led to a U.S. patent application (20070011609, "Configurable, multimodal human-computer interface system and method").

The main advantage of the designed EGT-based interface is that it responds instantly to broad displacements of the user's eye gaze on the computer screen. Hence, eye-gaze interaction gives the individual the feeling of a highly responsive system. All the results obtained from the three different experiments revealed that eye-gaze performance was considerably improved for all users with the intervention of the specific ANN configurations and associated user profiles regardless of the complexity of the experiment.

ACKNOWLEDGMENTS

We are grateful to the support provided by the National Science Foundation (NSF). This research was conducted with Assistive Technology Research Institutional Review Board Approval 011707-02.

This material was based on work supported by NSF grants EIA-9906600, HRD-0317692, CNS-0426125, and CNS-0540592; NSF graduate fellowships for Drs. Sesin

and Cabrerizo; and additional infrastructure support from NSF grants CNS-0520811 and IIS-0308155.

The authors have declared that no competing interests exist.

REFERENCES

- Hong P, Huang T. Natural mouse—A novel human computer interface. In: Proceedings of the IEEE International Conference on Image Processing; 1999 Oct 24–28; Kobe, Japan. p. 653–56.
- Hutchinson TE, White KP Jr, Martin WN, Reichert KC, Frey LA. Human-computer interaction using eye gaze input. *IEEE Trans Syst Man Cybern.* 1989;19(6):1527–34.
- Adjouadi M, Sesin A, Ayala M, Cabrerizo M. Remote eye gaze tracking system as a computer interface for persons with severe motor disability. In: *Lecture notes in computer science.* Vol. 3118. Berlin (Germany): Springer; 2004. p. 761–69.
- Barreto AB, Scargle SD, Adjouadi M. A real-time assistive computer interface for users with motor disabilities. *ACM SIGCAPH Comp Phys Handicap.* 1999;64:6–16.
- Duchowski AT. *Eye tracking methodology: Theory and practice.* New York (NY): Springer; 2003.
- Glenstrup AJ, Engell-Nielsen T. *Eye controlled media: Present and future state [thesis].* Copenhagen, Denmark: University of Copenhagen; 1995.
- Pantic M, Sebe N, Cohn JF, Huang T. Affective multimodal human-computer interaction. In: *Proceedings of the 13th Annual ACM International Conference on Multimedia;* 2005 Nov 6–11; Singapore. p. 669–76.
- Ovita SL. Multimodal interfaces. In: Jacko J, Sears A, editors. *The human-computer interaction handbook: Fundamentals, evolving technologies and emerging applications.* Mahway (NJ): Lawrence Erlbaum Associates; 2002. p. 286–304.
- Barreto AB, Scargle SD, Adjouadi M. A practical EGM-based human-computer interface for users with motor disabilities. *J Rehabil Res Dev.* 2000;37(1):53–63. [\[PMID: 10847572\]](https://pubmed.ncbi.nlm.nih.gov/10847572/)

10. Chin CA, Barreto A, Adjouadi M. Enhanced real-time cursor control algorithm, based on the spectral analysis of electromyograms. *Biomed Sci Instrum.* 2006;42:249–54. [PMID: 16817616]
11. Chin C, Barreto A, Alonso M Jr. Electromyogram-based cursor control system for users with motor disabilities. In: *Lecture notes on computer science.* Vol. 4061. Berlin (Germany): Springer; 2006. p. 905–12.
12. Špakov O, Miniotas D. Gaze-based selection of standard-size menu items. In: *Proceedings of the 7th International Conference on Multimodal Interfaces;* 2005 Oct 4–6; Toronto, Italy. p. 124–28.
13. Bates R, Istance H. Zooming interfaces!: Enhancing the performance of eye controlled pointing devices. In: *Proceedings of the 5th International ACM Conference on Assistive Technologies;* 2002 Jul 8–10; Edinburgh, Scotland. p. 119–26.
14. Ashmore M, Duchowski AT, Shoemaker G. Efficient eye pointing with a fisheye lens. In: *Proceedings of the ACM International Conference of Graphics Interface;* Canadian Human-Computer Communications Society; 2005 May 9–11; Victoria (Canada). p. 203–10.
15. Kumar M, Paepcke A, Winograd T. EyePoint: Practical pointing and selection using gaze and keyboard. In: *Proceedings of ACM CHI 2007 Conference on Human Factors in Computing Systems;* 2007 Apr 28–May 3; San Jose, California. p. 421–30.
16. Baluja S, Pomerleau D. Non-intrusive gaze tracking using artificial neural networks. *Adv Neural Inform Proc Syst.* 1994;6:753.
17. Piratla NM, Jayasumana AP. A neural network based real-time gaze tracker. *J Network Comp Appl.* 2002;25(3):179–96.
18. ISCAN Inc. Raw eye movement data acquisition software: Instruction manual. Burlington (MA): ISCAN Inc; 1997.
19. Guestrin ED, Eizenman M. General theory of remote gaze estimation using the pupil center and corneal reflections. *IEEE Trans Biomed Eng.* 2006;53(6):1124–33. [PMID: 16761839] Erratum in: *IEEE Trans Biomed Eng.* 2006;53(8):1728.
20. Ohno T, Mukawa N. A free-head, simple calibration, gaze tracking system that enables gaze-based interaction. In: *Proceedings of the 2004 Symposium on Eye Tracking Research and Applications;* 2004 Mar 22–24; San Antonio, Texas. p. 115–22.
21. Morimoto CH, Koons D, Amir A, Flickner M. Pupil detection and tracking using multiple light sources. *Image Vision Comput.* 2000;18(4):331–35.
22. Sesin A, Adjouadi M, Ayala M, Barreto A, Rishe N. A real-time vision based human computer interface as an assistive technology for persons with motor disability. *WSEAS Trans Comp Res.* 2007;2(2):115–21.
23. Hertz J, Krogh A, Palmer RG. *Introduction to the theory of neural computation.* Redwood City (CA): Addison-Wesley; 1991.
24. Ayala M, Adjouadi M. (2004). Eye-gaze tracking system evaluator programming tool. Miami (FL): Center for Advanced Technology and Education. Available from: www.cate.fiu.edu/soft.html/.
25. Kim Y, Varshney A. Saliency-guided enhancement for volume visualization. *IEEE Trans Vis Comput Graph.* 2006;12(5):925–32. [PMID: 17080818]
26. Hedrick B, Pape TL, Heinemann AW, Ruddell JL, Reis J. Employment issues and assistive technology use for persons with spinal cord injury. *J Rehabil Res Dev.* 2006;43(2):185–98. [PMID: 16847785]

Submitted for publication May 30, 2007. Accepted in revised form January 22, 2008.

