A Real-Time Vision Based Human Computer Interface as an Assistive Technology for Persons with Motor Disability
Anaelis Sesin, Malek Adjouadi, Melvin Ayala, Armando Barreto, Naphtali Rishe

Hardware Design, Cost and Diameter Analysis of Super-Hypercube Array, Master-Slave Star-Ring Super-Hypercube and Master-Slave Super-Super-Hypercube 4-Cube Architectures
M. Amiri-pour, H. Abachi, K. Dabke

On a MTTF Analysis of Fault-tolerant Hybrid MINs
Nitin, Vivek Kumar Sehgal, P.K. Bansal

Extracting Knowledge from Multiple Genetic Resources
Chien-I Lee, Hsiu-Min Chuang

Development of 64-bit Message Passing Interface for Large Scale Computing System

Hopfield Neural Network: Model, Applications and Implementations
Humayun Karim Sulehria, Ye Zhang

On Specification and Evaluation of Object-Relational Data Models
Erki Eessaar

On Using UML to Model a Building Information System
Wang Zhiguang, Chen Ming, Xu Junjie, Liu Ming

An Empirical Approach to Audio-Visual Guided Electronic Commerce
D. Rigas, A. Stergiou

The Development Method for Process - Oriented Data Warehouse
Laila Niedrite, Darja Solodovnikova, Maris Treimanis, Aivars Niedrits
<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaptive Knowledge Base using Rules and Cases and its Application to Japanese-to-Braille translation</td>
<td>199</td>
</tr>
<tr>
<td>Utoshi Ono, Takashi Yamasaki, Shigeru Nakayama</td>
<td></td>
</tr>
<tr>
<td>Security Software Engineering with a Knowledge Based Engineering</td>
<td>207</td>
</tr>
<tr>
<td>Ahmad Alazzazi, Asim El Sheikh</td>
<td></td>
</tr>
<tr>
<td>Neural Networks Modeling in Greenhouse with Spatial Variability Identfication</td>
<td>214</td>
</tr>
<tr>
<td>Tarcos Alberto Bussab, Joao Israel Bernardo, Andre Riyuki Hirakawa</td>
<td></td>
</tr>
<tr>
<td>Local-global Approach to Generalized Network Design Problems</td>
<td>220</td>
</tr>
<tr>
<td>Rica Pop, Corina Pop Sitar, Nicolae Pop, Ioana Zelina</td>
<td></td>
</tr>
<tr>
<td>Tended Protocol for Secure Multi-party Computation using Ambiguous Identity</td>
<td>227</td>
</tr>
<tr>
<td>K. Mishra, M. Chandwani</td>
<td></td>
</tr>
<tr>
<td>Proposed Software Process Framework for Web-Based Applications Development in Small Italian Web Firms</td>
<td>234</td>
</tr>
<tr>
<td>room Tarawneh, Asim Elsheikh, Sattam Lahawiah</td>
<td></td>
</tr>
<tr>
<td>SS Customisers Encounter Intelligent Agents</td>
<td>240</td>
</tr>
<tr>
<td>Orhan Aydin, Semra Tunali</td>
<td></td>
</tr>
<tr>
<td>Maintaining Generalized Association Rules for Record Deletion</td>
<td>248</td>
</tr>
<tr>
<td>Ng-Pei Hong, Tzu-Jung Huang</td>
<td></td>
</tr>
<tr>
<td>Themetic Development for Software Requirements using Natural Language Processing Techniques</td>
<td>255</td>
</tr>
<tr>
<td>Sad Hudaib, Bassam Hammo, Yara Alkhader</td>
<td></td>
</tr>
<tr>
<td>Ore Outsourcing of DNA Sequences Comparisons in a Grid Environment</td>
<td>262</td>
</tr>
<tr>
<td>Hel Akimana, Olivier Markowitch, Yves Roggeman</td>
<td></td>
</tr>
<tr>
<td>Backup System Based on XML Format</td>
<td>270</td>
</tr>
<tr>
<td>Eymane Oumtanaga, Tey Pierre, Kadjo Tanon Lambert, Dowa N'sreke Florent</td>
<td></td>
</tr>
<tr>
<td>Understanding Visitor Behaviors from Web Log Data</td>
<td>277</td>
</tr>
<tr>
<td>Astellano, A.M. Fanelli, M.A. Torsello</td>
<td></td>
</tr>
<tr>
<td>Rectifying Method of Music Impression Space for Impression-based Satisfactory Retrieval</td>
<td>285</td>
</tr>
<tr>
<td>Ositi Takayama, Yuuta Hashiba, Shigeru Kikuchi, Tetsuo Ikeda, Yoshiyoshi Murata</td>
<td></td>
</tr>
<tr>
<td>Role-Based Intelligent Refinement Method for Web Application Systems</td>
<td>291</td>
</tr>
<tr>
<td>Mano, Takakazu Kaneko</td>
<td></td>
</tr>
</tbody>
</table>
A System-Level HW/SW Co-Design Methodology for Ad-Hoc Implementation of DBMS Operators

Luigi Pomante, Paolino Di Felice

Representing Situation Model as Interconnected Frames: One Step to Make Computer Comprehend Stories

Hideji Enokidt, Masahiro Kiyota, Jun Ohashi, Seiichi Komiya

Differential Scenario for both Generating and Classifying Scenarios

Masayuki Makino, Hiroya Itoha, Atsushi Ohnishi

Aggregating and Fine Tuning Remote Map Services with Local Cascading WMS Server

Tomas Brunclik

Employing a Genetic Algorithm Based on Knowledge to Address the Job Shop Scheduling Problem

Hung-Pin Chiu, Kun-Lin Hsieh, Yi-Tsung Tang, Wan-Jung Chien

Study on an Architecture for Web-based DSS

Huabin Chen, Xiaodong Zhang, Tianhe Chi

A Novel Approach to Address the Job-Shop Scheduling Problem by using a Tabu Genetic Algorithm

Hung-Pin Chiu, Kun-Lin Hsieh, Yi-Tsung Tang, Ching-Yu Wang

Implementing Model-Based Technique using Recursive Methodology

Azlinah Mohamed, Noor Habibah Arshad, Norhamimi Mohd Hidzir

Software Development Projects: Risk Factors and Occurrences

Noor Habibah Arshad, Azlinah Mohamed, Zaiha Mat Nor

A Stepwise Approach to Managing Software Projects

Syed Jawad Hussain, Khalid Rashid, H. Farooq Ahmad, Syed Fawad Hussain
EDITORIAL BOARD

EDITOR-IN-CHIEF

Prof. Nikos E. Mastorakis, Military Institutes of University Education (ASEI) Hellenic Naval Academy, GREECE

ASSOCIATE EDITORS

Prof. P. Anderson, Rochester Institute of Technology, USA
Prof. M. Araki, Kyoto University, Japan
Prof. L. Chua, University of California, USA
Prof. H. Bothe, Technical University of Denmark
Prof. M.J. Er, Nanyang Technological, Singapore
Prof. E. Fornasini, Universita di Padova, Italy
Prof. M. Gori, Universita di Siena, Italy
Prof. M. Guziani, University of West Florida, USA
Prof. W. Gruver, Simon Fraser University, USA
Prof. T. Hinamoto, Hiroshima University, Japan
Prof. M. Jamshidi, University of Mexico, USA
Prof. R. Jarvis, Monash University, USA
Prof. L. Jozwik, Eindhoven University of Technology, The Netherlands
Prof. W. Karplus, University of CA at LA, USA
Prof. T. Kaczorek, Warsaw Technical Univ., Poland
Prof. A. Kummert, University of Wuppertal, Germany
Prof. V. Lakshmikantham, FIT, USA
Prof. P. Loucopoulos, UMIST, UK
Prof. V. Mladenov, Technical Univ. of Sofia, Bulgaria
Prof. N. Steele, Coventry University, USA
Prof. A. Sage, George Mason University, USA
Prof. D. Siljak, University of Santa Clara, USA
Prof. J. Tien, Rensselaer Polytechnic Institute, USA
Prof. S. Vassilev, Siberian Branch of Russian Academy of Science, Russia
Prof. M. Vidyasagar, CAIR, India
Prof. R. Yager, Iona College, USA


SUBSCRIPTION: The subscription rate for each journal is 100 Euros (per year) for individuals and 200 Euros (per year) for institutions or companies.

FORMAT OF THE PAPERS: http://www.worldses.org/journals
ISSN: 1991-8755
WSEAS E-LIBRARY: http://www.wseas.org/data
WSEAS CHAPTERS: http://www.wseas.org/chapters

Each paper of this issue was published after review by 3 independent reviewers

WSEAS Headquarters: Ag. I. Theologou 17-23, 15773, Zographou, Athens, Greece.
Tel: (+30) 210 7473313. Fax: (+30) 210 7473314
A Real-Time Vision Based Human Computer Interface as an Assistive Technology for Persons with Motor Disability

ANAELIS SESIN, MALEK ADJOUADI, MELVIN AYALA, ARMANDO BARRETO, NAPHTALI RISHE
Department of Electrical & Computer Engineering
Florida International University
10555 W. Flagler Street, Miami FL 33174
U.S.A.
Anaels.Sesin@fiu.edu, http://www.cate.fiu.edu

Abstract: - The aim of this study is to develop a human computer interface (HCI) to empower people with disability with hand free communication and computer access. The interface employs eye gazing as the primary computer input mechanism. It relies on the use of remote eye-gaze tracking (EGT) device to compute the direction of gaze and utilizes it to control the mouse cursor. Unfortunately, the performance of these interfaces is traditionally affected by inaccuracies inherited from the eye tracking devices and ineffective eye-tracking data to screen coordinates normalization algorithm. This study focuses on the development of new optimized EGT to screen coordinates data conversion mechanism which minimizes considerable the disparity between gaze-point and the actual fixation of the eye. It analyzes in more details the correlation between the two data types resulting in an increment in the accuracy of the system. In the development of this data conversion mechanism, the following steps were implemented: (a) compilation of EGT data and mouse cursor coordinates simultaneously in order to find the correlation between the two data types, (b) search for a curve fitting method that best describes the behavior of the data, (c) computation of the data conversion coefficients based on the collected data set (d) implementation of a decision mechanism to determine the appropriate mouse cursor action based on the direction of gaze, and (e) application of visualization tools to monitor and evaluate the system performance.

Key-words: - Human-computer interaction, eye gaze tracking, least-squares line

1. Introduction
Unconventional computer input devices have been developed based on the perception of the human senses to harness the power of computing and access the variety of resources made available thereby. These aids allow people to control computer through devices beyond the standard keyboard and mouse. Examples of adaptive equipments include smaller or larger keyboards, eye-gaze pointing devices, sip-and-puff systems controlled by breathing, and other alternative pointing devices.

The aim of this study is to provide universal access to computers to individuals with severe motor disabilities via a vision based human computer interface (HCI). Such access will make it possible for a motor disabled individual to use computers and issue command and control tasks using only eye-gazing [1, 2]. This is achieved through the integration of an eye gaze tracking (EGT) system to the interface.

The EGT interface is inherently integrated in terms of modalities of use and in hardware-software assimilation. The integrated aspect of the design is based on the use of the less intrusive (passive) remote eye-gaze tracking system in contrast to the head mounted version in order to free the user from any physical constraint. The eye tracking device computes eye coordinates in accordance to the visual field, in this case a computer monitor. These are then sent to the stimulus computer through the serial port, where they are normalized into mouse-pointer coordinates.

Regrettably, using eye-gaze activity as the primary computer input mechanism is less stable and less accurate than most manual input devices due to unwanted collateral effects. Such undesirable effects are: the ubiquitous jittering of the mouse pointer due
to saccade behavior of the eye, and the disparity between the gaze point and the mouse pointer position due to unsuitable data normalization algorithm.

Accuracy of mouse cursor control through eye gazing activity can be enhanced using different approaches, such as: (a) integration of a complementary technology -- i.e., Electromyogram (EMG) [3, 4], and (b) application of different smoothing algorithm to the mouse cursor trajectory -- i.e. training an artificial neural network to learn how the EGT inputs statistically relate to the jittering [5].

This study addresses the data translation problems by proposing an effective conversion algorithm which considerably reduces the discrepancy between the EGT coordinates and the mouse pointer position. In order to ensure accurate cursor displacement, the following important tasks have been performed:

1. Develop an interface for data collection which records EGT and screen coordinates simultaneously.
2. Enhance the data conversion mechanism based on the equations for the Least-Squares Line using previously collected data.
3. Use the new data conversion mechanism to calculate the mouse cursor coordinates.
4. Develop a decision mechanism to determine which mouse cursor action should be executed based on the already normalized coordinates.
5. Evaluate system performance using a Metric Monitoring (MM) application.

2. Eye Gaze System Overview

2.1. System Configuration

The vision based human-computer interface as proposed is based on a benchtop eye gaze setup. It integrates a CPU for raw eye movement data acquisition, a CPU for user interaction, an eye monitor, a scene/auxiliary monitor, an eye imaging camera, and an infrared light source (Fig. 1).

The computer for data acquisition (DA) communicates with the stimulus computer through the serial port. Eye image is detected and analyzed using the Raw Eye Movement Data Acquisition (REMDA) software which resides in the DA computer. The EGT system used in this study is the ISCAN® ETL-500 [6].

2.2. EGT Data Range

The EGT calibration system divides a scene NTSC video signal into 512H x 512V pixel matrix (Fig. 2).

The sync and blanking portion of the video signal are not part of the viewed image in the monitor. However, they use up the edge portions of the matrix. The section of the matrix that corresponds to the viewed image is from 21 to 481 in the horizontal position and from 57 to 481 in the vertical position [6]. When computing the direction of gaze, the values correspond to a horizontal and vertical position in the matrix.

```
(0,0) Horizontal Coord. (511,0)

Vertical Coord. (0,511) (511,511)
```

Fig. 2 Scene NTSC Video Coordinate Matrix

2.3. Serial Communication

During serial communication, the EGT system can output up to six parameters. The table below lists all the possible parameters that can be output through the serial port.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>P.O.R. H</td>
<td>Point of regard horizontal ordinate</td>
</tr>
<tr>
<td>P.O.R. V</td>
<td>Point of regard vertical ordinate</td>
</tr>
<tr>
<td>Pupil H</td>
<td>Pupil horizontal position ordinate</td>
</tr>
<tr>
<td>Pupil V</td>
<td>Pupil vertical ordinate</td>
</tr>
<tr>
<td>Pupil D</td>
<td>Pupil horizontal ordinate</td>
</tr>
<tr>
<td>Pupil VD</td>
<td>Pupil vertical diameter</td>
</tr>
<tr>
<td>C.R. H</td>
<td>Corneal reflection horizontal ordinate</td>
</tr>
<tr>
<td>C.R. V</td>
<td>Corneal reflection vertical ordinate</td>
</tr>
<tr>
<td>P-C.R.</td>
<td>Pupil and corneal reflection relationship</td>
</tr>
</tbody>
</table>

The parameter values, \( x_k \), were not defined as constants because the eye coordinates varied during the experiment.

where \( x_k \) is a constant

```
Fig. 1: Eye gaze based human-computer interface components and setup

3. Preliminary

In previous human-computer interactions, a mouse was used to control the cursor on which the mouse was placed.

4. Algorithm

Mouse: Precise positioning of the mouse is a crucial component in assistive environments. In order to control the mouse, eye gaze information is recorded. The mouse computer is synchronized to the monitor's data stream. The mouse's data is then captured and used to calculate the gaze point within the image.

4.1. Data Collection

An appropriately calibrated eye gaze system (EGT) is used to record eye movements in the scene (Fig. 3). The calibration data is used to map the eye gaze to the mouse position. The EGT-mouse interface is then defined and applied to the mouse. The mouse button is then pressed to give the user a choice to select one of the options.
The parameters that represent the actual position of the eye gaze of the user are the Point of Regard Horizontal (P.O.R.H) and Point of Regard Vertical (P.O.R.V). Only these two parameters are sent through the serial port to the stimulus computer.

3. Preliminary Data Conversion Algorithm

In previous studies, mouse pointer coordinates were defined as the product of the EGT coordinates and a constant of proportionality, given by the ratio of the maximum screen value and the maximum EGT value [7]. The resultant data conversion equations were:

\[ x_M = \frac{x_E}{x_{E_{\text{Max}}}} \cdot x_E, \quad y_M = \frac{y_E}{y_{E_{\text{Max}}}} \cdot y_E \]  

where \( x_{E_{\text{Max}}} \) and \( y_{E_{\text{Max}}} \) are the monitor resolution value, \( x_E \) and \( y_E \) are the maximum EGT values, and \( x_M \) and \( y_M \) are the current EGT coordinates.

Experiments demonstrated that these equations were not suitable for this type of data conversion because there was always a disparity/offset between the eye gazing point and the mouse cursor coordinates. This made eye gazing control of the mouse cursor very difficult. The offset was not constant throughout the screen. It varied depending on which of the four quadrants of the monitor and how close to the center of the screen the visual point was.

4. Algorithm for Enhanced EGT to Mouse Pointer Data Normalization

Precise EGT to mouse pointer coordinate conversion is a crucial step in the implementation of vision based assistive system for persons with motor disability. In order to ensure accurate data conversion, important steps where performed: (a) collection of EGT and mouse cursor data simultaneously, (d) analysis of the data’s behavior in order to find the appropriate curve fitting equations, and (b) implementation and testing of the data conversion equations.

4.1. Data Collection

An application was developed in order to collect and map the EGT-cursor to mouse-cursor coordinates. EGT-cursor refers to the cross hair that appears in the scene monitor indicating to the user’s gazig position (Fig.3 (c)). The Mouse-EGT Correlation (MEC) application consists on a GUI displaying a matrix of buttons (Fig.3 (a)) with a default dimension of 23x17.

The default dimension of the matrix was determined based on the current screen resolution (1024x768 pixels), and buttons size (250 twips). The dimension of the matrix can be changed at user demand. When running the application in the stimulus computer, the matrix is also displayed in the scene monitor.

Data was collected using the REMDA software in the calibration mode in which calibration points are displayed in the panel pseudo screen as small squares. Each of calibration points can be in the active stage one at a time denoted by a circle appearing around the square (Fig.3 (b)). The active cursor becomes visible on the scene monitor superimposed to the image displayed in the stimulus computer, and its point-of-regard (P.O.R.) coordinates are output through the serial port (Fig.3 (c)). The active calibration point can be moved around the screen using the computer arrow keys.

![Fig. 3. (a) EGT-Correlation Application Snapshot (b) REMDA Calibration Control Panel (c) EGT System Scene Monitor](image)

During data collection, the mouse pointer was placed over one of the buttons in the MEC matrix while the EGT-cursor was placed over the same point using as reference the image displayed in the scene monitor. As the button was clicked, its matrix position, the mouse cursor position \((x_M, y_M)\), and the P.O.R. coordinates \((x_E, y_E)\) were saved in a table format (Table 2).

<table>
<thead>
<tr>
<th>Matrix Position</th>
<th>X_E</th>
<th>Y_E</th>
<th>X_M</th>
<th>Y_M</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>22</td>
<td>42</td>
<td>30</td>
<td>79</td>
</tr>
<tr>
<td>0.1</td>
<td>65</td>
<td>41</td>
<td>50</td>
<td>79</td>
</tr>
<tr>
<td>0.2</td>
<td>110</td>
<td>42</td>
<td>70</td>
<td>79</td>
</tr>
<tr>
<td>0.3</td>
<td>156</td>
<td>41</td>
<td>90</td>
<td>79</td>
</tr>
<tr>
<td>0.4</td>
<td>203</td>
<td>40</td>
<td>110</td>
<td>79</td>
</tr>
<tr>
<td>0.5</td>
<td>244</td>
<td>42</td>
<td>130</td>
<td>79</td>
</tr>
</tbody>
</table>
Notice that the coordinates of the mouse pointer is equivalent to the position of the button on the computer screen. This procedure was repeated for each of the buttons in the matrix. The collected data was then used to map the correlation between the EGT data and the mouse pointer location.

EGT vs. mouse-cursor plots were generated using the collected data, which indicated a linear relationship between the two data types as shown in Fig. 4.

![EGT vs Monitor Coordinates](image)

**Fig. 4:** EGT vs. monitor coordinates from one of the data sample files (a) x-axis (b) y-axis

### 4.2. Curve Fitting Method

The method used to map the collected data set based on its characteristics was the least squares method [8] which is a mathematical procedure used to find the best-fit curve for a given set of points by minimizing the sum of the squares of the offsets (deviation) of the points from the curve. Using the sum of the squares of the offsets instead of the absolute values allows the residuals to be treated as a continuous differentiable quantity.

The least-squares line method uses a straight line $y = a + bx$ to approximate the data set $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$, where $n > 2$. The best-fit curve $f(x)$ is defined as the curve that has the least square error. For the best fitting curve, the sum of the deviations squared yields:

$$
\Pi = \sum_{i=1}^{n} [y_i - f(x_i)]^2 = \sum_{i=1}^{n} [y_i - (a + bx_i)]^2 = \text{min.}
$$

where $a$ and $b$ are unknown coefficients while all $x_i$ and $y_i$ are given. To obtain the least square error, the first derivatives of $a$ and $b$ must yield zero. Solving for $a$ and $b$ in equation (2) yields:

$$
a = \frac{\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2}
$$

and

$$
b = \frac{\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2}
$$

As mentioned before in section 2.3, even thought the EGT data values are based on a 512x512 matrix, not all the points in the matrix are part of the image viewed on the stimulus computer screen. For those points, the equations above do not result in valid mouse coordinates on the screen. The resulting values fall outside of the screen resolution causing the mouse pointer to roll over. Thus, the equations are only valid for the EGT values that fall within the range of the viewed image. The interface does not process the EGT values that correspond to the blanking portion of the video. Since data is being received at a frequency of 60 Hz, this does not affect the performance of the system.

The correlation data set, collected using the MEC software, was divided into two subsets corresponding to the horizontal and vertical axis. Equations (4) and (5) were applied to each data set independently in order to compute the least square coefficients. This process was done using a monitor resolution of 1024x768 and 800x600 in the stimulus computer, and the default matrix dimension in the MEC application. The resultant coefficient values are tabulated below.

<table>
<thead>
<tr>
<th>Table 3: Least Square Coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1024x768</strong></td>
</tr>
<tr>
<td><strong>a</strong></td>
</tr>
<tr>
<td>Horizontal</td>
</tr>
<tr>
<td>Vertical</td>
</tr>
</tbody>
</table>

Plugging the yield to the following equation:

$$
\Pi_M = \Pi_{x_M} + \Pi_{y_M}
$$

The implemented system to implement number of operations to move function the fact coordinates.

### 5.1. Horizontal

Within the horizontal coordinate, the different directions are valid mouse actions as the computer.

### 5.2. Vertical

If the vertical direction is up or down, it indicates the eye blink. The remaining actions are based on the zero condition.
Plugging the values into the straight line equation yield to the following data translation equations:

- using 1024x768 monitor’s resolution:
  \[ x_M = -45.2347 + 2.21879 \cdot x_E \quad (6) \]
  \[ y_M = -101.6716 + 1.7907 \cdot y_E \quad (7) \]
- using 1024x768 monitor’s resolution:
  \[ x_M = -49.339 + 1.78667 \cdot x_E \quad (8) \]
  \[ y_M = -78.989 + 1.40903 \cdot y_E \quad (9) \]

The Data Normalization Module (DNM) implements both pairs of equations allowing the system to perform accurately in any of the two screen resolutions. It automatically detects the current number of pixels across and down on the stimulus monitor, and selects the corresponding coefficient values for data translation.

5. Eye Motion Interpretation Algorithm

Once the coordinates are converted to pixel values, they are used as input to the Mouse Control Module (MCM) whose purpose is to translate user’s gazing behavior into mouse cursor actions. The two main actions that are executed when using the mouse are: to move the cursor, and to employ a left click. The functionality of the mouse was implemented based on the fact that if the eyes are closed the EGT coordinates will assume a value of zero.

5.1. How to Trigger Cursor Movement

Within the MCM, a subroutine determines if the eye coordinates are different than zero or not. If they are different, it uses the P.O.R.H. and P.O.R.V. values, already converted to screen coordinates, to move the mouse cursor to that position.

5.2. How to Trigger Mouse Left Click

If the values coming from the EGT are equal to zero, it indicates that the subject closed the eye which can be either a regular eye blink or a voluntary closing of the eye where the duration is assumed longer than the blink. In order to differentiate between the two actions, once zero coordinates are detected by the MCM, the program counts how many consecutive zero coordinates are received. If the counter value is very small (less than 15 counts), no action is executed. However, if the counter value is equal to 15, then the program initiates a left mouse click event. In order to determine the dwelling time at which the left click should be executed, different counter values were tested and the more accurate results were obtained when using 15. It is good to point out that the EGT coordinates are sent through the serial port at a frequency of 60 Hz, thus the interval in which 15 points are received is very small and is equivalent to 1/4 s.

6. Tool for System Performance Evaluation

The tool used during the testing section was the Metric Monitoring (MM) application (Fig. 5) [9] which provides indicators to evaluate the performance of the data conversion algorithm based on the interaction of the user. It consists on having a target (button) moving through out the screen which the subject should follow with the eye (only one eye is being tracked). The trajectory of the button can be changed at user’s request and it is defined by markers placed on different positions on the screen. As the target moves, the coordinates of the center of the button as well as the position of the mouse cursor is recorded.

![Fig. 5: Graphical evaluation application](image)

The degree of discrepancy of the mouse cursor and the visual point (center of button) is defined as the Euclidean distance \( D_e \) between the centroid \((x_c, y_c)\) of the mouse pointer and the center of the target button location \((x_b, y_b)\) for each time frame \( \Delta t \). The time frame was defined based on the frequency of the EGT system (60 Hz) to be equal to 1 second. The centroid was determined using the six points that fall within \( \Delta t \), and the target was the average position of the moving button at that time. The degree of discrepancy equation is written as:

\[
D_e = \sqrt{(x_c - x_b)^2 + (y_c - y_b)^2} \quad (10)
\]
where \( x = \frac{\sum_{i=1}^{n} x_i}{n} \), \( y = \frac{\sum_{i=1}^{n} y_i}{n} \), \( x_i \) and \( y_i \) are the mouse cursor coordinates, and \( i = 1,2,\ldots,n \).

After calculating the Euclidean distance for each button position, the average offset was computed as:

\[
D_{\mu} = \frac{\sum_{i=1}^{n} D_i}{n}
\]

where \( n \) is the number of time frames in the data set.

7. Results

Experiments were conducted with six subjects in order to determine the prospects for EGT-mouse offset reduction mechanism. The objective was to evaluate the behavior of the system using the two data translation algorithms mentioned in this article and compare the results.

EGT and mouse pointer data were collected with the MECDC application using different calibration matrix dimensions and different monitor resolutions. For each data set, conversion coefficients were calculated and used to compute the mouse pointer position, and the results were tested using the MM application.

![Fig. 6. HCI testing setup](image)

The average degree of discrepancy obtained for all subjects is shown in the following tables.

Table 4: EGT-mouse offset reduction using 640x480 monitor resolution.

<table>
<thead>
<tr>
<th>Subject</th>
<th>Previous Average Offset (a)</th>
<th>New Average Offset (b)</th>
<th>Ratio of Improvement ((a - b) / a) %</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>73.0542</td>
<td>11.8192</td>
<td>83.82132718</td>
</tr>
<tr>
<td>2</td>
<td>81.36279</td>
<td>78.5322</td>
<td>3.478863004</td>
</tr>
<tr>
<td>3</td>
<td>52.86821</td>
<td>34.55053</td>
<td>34.64781577</td>
</tr>
<tr>
<td>4</td>
<td>82.72291</td>
<td>16.17111</td>
<td>79.91232417</td>
</tr>
<tr>
<td>5</td>
<td>79.77081</td>
<td>16.1265</td>
<td>79.78395857</td>
</tr>
<tr>
<td>6</td>
<td>100.2122</td>
<td>4.371254</td>
<td>95.63800216</td>
</tr>
</tbody>
</table>

The results from Table 4 and 5 exhibit an average reduction of eye-mouse deviation of 62.88% when using 640x480 monitor resolution and of 72.91% when using 1024x768 with the new data conversion mechanism, which represents a substantial improvement in the use of eye gaze to control the mouse pointer. The difference in the offset values of 10.3% demonstrates that when using small screen resolution (icons are displayed bigger on the screen) the user has better eye gazing control of the mouse cursor.

Table 5: EGT-mouse offset reduction using 1024x768 monitor resolution.

<table>
<thead>
<tr>
<th>Subject</th>
<th>Previous Average Offset (a)</th>
<th>New Average Offset (b)</th>
<th>Ratio of Improvement ((a - b) / a) %</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>554.8309</td>
<td>50.46035</td>
<td>90.905274</td>
</tr>
<tr>
<td>2</td>
<td>113.583</td>
<td>112.6977</td>
<td>8.65851404</td>
</tr>
<tr>
<td>3</td>
<td>67.05093</td>
<td>33.7582</td>
<td>49.6259892</td>
</tr>
<tr>
<td>4</td>
<td>573.0231</td>
<td>24.1392</td>
<td>95.787381</td>
</tr>
<tr>
<td>5</td>
<td>550.8675</td>
<td>16.1265</td>
<td>97.0725265</td>
</tr>
<tr>
<td>6</td>
<td>569.1422</td>
<td>26.4758</td>
<td>95.348122</td>
</tr>
</tbody>
</table>

Table 6: EGT-mouse offset reduction using 3x3 and 5x5 matrix dimensions with 1024x768 monitor resolution.

<table>
<thead>
<tr>
<th>Subject</th>
<th>3 x 3 (a)</th>
<th>5 x 5 (b)</th>
<th>Ratio of Improvement ((a - b) / a) %</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>89.21197</td>
<td>50.46035</td>
<td>43.43769115</td>
</tr>
<tr>
<td>2</td>
<td>123.4176</td>
<td>112.6977</td>
<td>8.68576245</td>
</tr>
<tr>
<td>3</td>
<td>43.85943</td>
<td>33.7582</td>
<td>23.03091946</td>
</tr>
<tr>
<td>4</td>
<td>44.13928</td>
<td>24.1392</td>
<td>45.3111518</td>
</tr>
<tr>
<td>5</td>
<td>20.5669</td>
<td>16.1265</td>
<td>21.59003058</td>
</tr>
<tr>
<td>6</td>
<td>73.5468</td>
<td>26.4758</td>
<td>64.0014294</td>
</tr>
</tbody>
</table>

Furthermore, the data collected in Table 6 reveals that the offset was reduced by 34.34% when changing the matrix dimensions from 3x3 to 5x5. This implies that by using more data points to calculate the data conversion coefficients, the system becomes more accurate. However, incrementing the number of calibration points makes the data collection process last longer, which may be very tedious. There needs to be a trade off between the calibration matrix dimensions and the data collection time interval.

When experimenting with the control application and web browsing, users reported the mouse cursor to be more stable and easier to control. The mouse pointer could reach the target and click on it with an improved degree of accuracy.
8. Conclusion
The aim of this study was to increment the accuracy of a real-time vision-based assistive human computer interface that will allow individuals with severe motor disabilities to use most of the Windows applications. This was accomplished by: (1) implementing an enhanced data conversion mechanism that decreases the degree of discrepancy between EGT and mouse cursor coordinates, and (2) evaluating the performance of the system by experimenting with different Windows applications.

Using eye gazing as primary computer input mechanism will give individuals with disabilities means to communicate with their eyes. Its main advantage is that it reacts faster than other input mechanisms resulting in a highly responsive system. Its downside is that eye tracking equipments are still less stable and less accurate than most manual input devices due to the saccade behavior of the eye.

However, this study confirms that, with the implementation of suitable data conversion mechanism and algorithms that overcome the eye jittering behavior, the use of eye movement as a source of computer input is feasible.

ACKNOWLEDGEMENTS
The authors gratefully acknowledge the support from the National Science Foundation Grants under grants HRD-0317692, CNS 042615 and CNS 063439, and the NSF graduate fellowship for Ms. Sesin.

References