Effective Data Conversion Algorithm for Real-Time Vision Based Human Computer Interface

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.
http//www.cate.fiu.edu

Abstract: -Human computer interfaces (HCI) for assisting persons with disabilities may employ eye gazing as the primary computer input mechanism. These systems rely on the use of remote eye-gaze tracking (EGT) devices to compute the direction of gaze and employ it to control the mouse cursor. Regrettably, the performance of these interfaces is traditionally affected by inaccuracies inherited from the eye tracking devices and ineffective EGT to mouse-pointer data conversion mechanisms. This study addresses this problem and proposes a new optimized data conversion mechanism. It analyzes in more details the correlation between the two data types resulting in a considerable increment in the accuracy of the system. This improved data conversion interface integrates the following procedures: (a) map the correlation between the EGT data and the mouse cursor position, (b) apply a curve fitting method that best suits the behavior of the data, (c) interpret the direction of gaze in order to determine the appropriate mouse cursor response, and (d) use effective means to monitor and evaluate the system performance.

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

1. Introduction

Alternative computer input devices have been developed to enable persons with disabilities 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 focus of this study is to implement a vision based user-computer interface, which can be achieved by including an eye gaze tracking (EGT) system to provide user interaction with the computer using only eye movement [1, 2]. 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.

Unfortunately, use of eye-gaze activity as the primary computer input device has been complicated and inaccurate due to unwanted collateral effects. Such undesirable effects are: the constant jittering of the mouse pointer due to saccade behavior of the eye, and the eye-mouse pointer deviation due to inaccuracy of the data conversion mechanism. The eye-mouse deviation refers to the distance between the EGT cursor and mouse cursor.

Mouse cursor control 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 allowing the user to control the mouse pointer through eye gazing.
(4) 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 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].

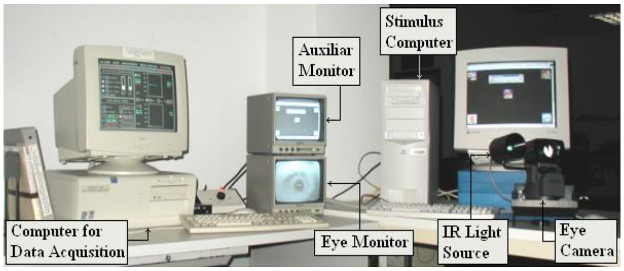


Fig. 1: Eye gaze based human-computer interface components and setup

2.2. Data Range

The calibration system divides a scene NTSC video signal into 512H x 512V pixel matrix. 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.

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:

$$x_M = \frac{x_{E\_Max}}{x_{M\_Max}} \cdot x_E \quad , \quad y_M = \frac{y_{E\_Max}}{y_{M\_Max}} \cdot y_E \qquad (1)$$

where x_{M_Max} and y_{M_Max} are the monitor resolution value, x_{E_Max} and y_{E_Max} are the maximum EGT values, and x_E and y_E are the current EGT coordinates.

Experiments demonstrated that these equations were not appropriate for this type of data conversion because there was always an offset between the eye gazing point and the mouse cursor coordinates. This made controlling of the mouse cursor through eye movement 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

Successful 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 ensure accurate data conversion, important steps where performed: (a) collection of EGT and mouse cursor data simultaneously, (d) mapping of the data 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 gazing position (Fig.2 (c)). The Mouse-EGT Correlation (MEC) application consists on a GUI displaying a matrix of buttons (Fig.2 (a)) with a default dimension of 23x17. The default dimension of the matrix was determined based on the current screen resolution (1024x768 pixels), and the size of the buttons (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. As each of calibration points become active, a circle appears around the square (Fig.2 (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.2 (c)). The active calibration point can be moved around the screen using the computer arrow keys.

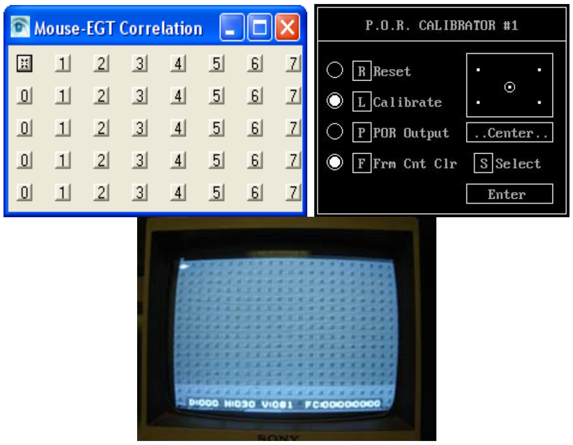


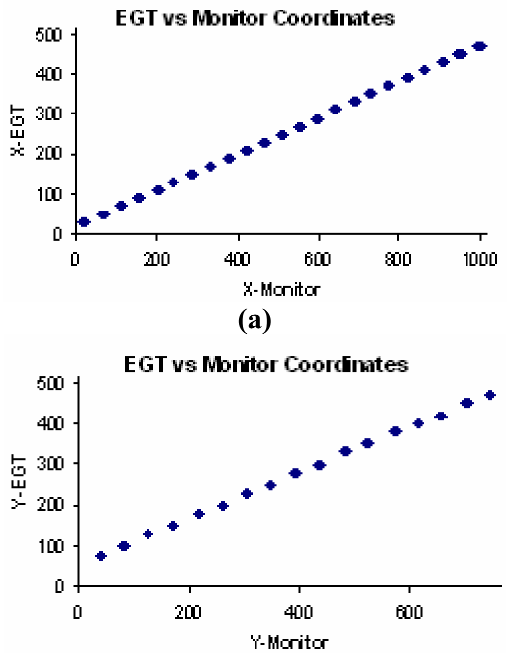
Fig. 2. (a) EGT-Correlation Application Snapshot (b) REMDA Calibration Control Panel (c) EGT System Scene Monitor

During data collection, the mouse pointer was placed on top of one of the buttons in the MEC application matrix, while the EGT-cursor was also placed on top of the same button. This was done using the image displayed in the scene monitor as reference. When 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), coming from the EGT, were saved into a table (Table 1). Notice that the position 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.

Table 1. Data correlation table format

Matrix Position	$\mathbf{X_E}$	$\mathbf{Y_E}$	$\mathbf{X_M}$	$\mathbf{Y_M}$
0,0	22	42	30	79
0,1	65	41	50	79
0,2	110	42	70	79
0,3	156	41	90	79
0,4	203	40	110	79
0,5	244	42	130	79
0,6	288	41	150	79

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. 3.



(a)

(b)

Fig. 3: 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 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_1, 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:

$$\Pi = d_1^2 + d_2^2 + ... + d_n^2 = \sum_{i=1}^{n} d_i^2$$

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

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:

$$a = \frac{\left(\sum_{i=1}^{n} y_i\right)\left(\sum_{i=1}^{n} x_i^2\right) - \left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} x_i y_i\right)}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2} \tag{3}$$

$$b = \frac{\left(\sum_{i=1}^{n} x_i y_i\right) - \left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} y_i\right)}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2} \tag{4}$$

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 since the values fall outside of the screen resolution values 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 system performance.

The correlation data set collected using the MEC software was divided into two subsets corresponding to the horizontal and vertical axis. Equations (3) and (4) 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 2. Least Square Coefficients

	1024x768		**800x600**	
	a	**b**	**a**	**b**
Horizontal	-45.235	2.2188	-49.339	-78.989
Vertical	-101.672	1.7907	1.78667	1.40903

Plugging the values into the straight line equation yield to the following data translation equations using 1024x768 monitor's resolution:

$$x_M = -45.2347 + 2.21879 \cdot x_E \tag{5}$$

$$y_M = -101.67155 + 1.79065 \cdot y_E \tag{6}$$

5. Tool for System Performance Evaluation

The Metric Monitoring (MM) application (Fig. 4) [9] was used in order to evaluate the performance of the data conversion algorithm. This interface 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.

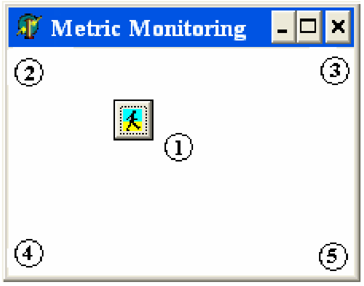


Fig. 4: Graphical evaluation application

The degree of discrepancy of the mouse cursor and the visual point (center of button) is determined by 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 Δt. 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 Δt, and the target was the average position of the moving button at that time.

The degree of discrepancy equation can be written as:

$$D_e = \sqrt{(x_c - x_b)^2 + (y_c - y_b)^2} \tag{7}$$

where $x_c = \frac{\sum_{i=1}^{6} x_i}{6}$, $y_c = \frac{\sum_{i=1}^{6} y_i}{6}$, x_i and y_i are the mouse cursor coordinates, and $i = 1,2,...,6$.

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

$$D_\mu = \frac{\sum_{i=1}^{n} D_e}{n} \tag{8}$$

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

6. Results

Experiments were conducted with six subjects in order to determine the prospects for EGT-mouse offset reduction. 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. 5. HCI testing setup

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

Table 3: EGT-mouse offset reduction using 640x480 monitor resolution.

Subject	Previous Average Offset (a)	New Average Offset (b)	Ratio of Improvement ((a – b) / a) %
1	73.0542	11.8192	83.82132718
2	81.36279	78.53229	3.478863004
3	52.86821	34.55053	34.64781577
4	82.72291	16.61711	79.91232417
5	79.77081	16.1265	79.78395857
6	100.2122	4.371254	95.63800216

Table 4: EGT-mouse offset reduction using 1024x768 monitor resolution.

Subject	Previous Average Offset (a)	New Average Offset (b)	Ratio of Improvement ((a – b) / a) %
1	554.8309	50.46035	90.905274
2	113.583	112.6977	8.65851404
3	67.05093	33.7582	49.6528982
4	573.0231	24.13928	95.787381
5	550.8675	16.1265	97.0725265
6	569.1422	26.4758	95.3481221

The results from Table 3 and 4 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 the user has better eye gazing control of the mouse cursor.

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

Subject	3 x 3 (a)	5 x 5 (b)	Ratio of Improvement ((a – b) / a) %
1	89.21197	50.46035	43.43769115
2	123.4176	112.6977	8.685876245
3	43.85943	33.7582	23.03091946
4	44.13928	24.13928	45.31111518
5	20.5669	16.1265	21.59003058
6	73.5468	26.4758	64.00142494

Furthermore, the data collected in Table 5 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, the mouse cursor was found to be more stable and easier to control since it could reach the target and click on it with an improved degree of accuracy.

7. Conclusion

The challenge 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 it 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 [10]. 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

[1] T.E. Hutchinson, K.P. White Jr., K.C. Reichert, & L.A. Frey. Human-computer interaction using eyegaze input, *IEEE Transactions on Systems, Man, and Cybernetics, 19*, Nov/Dec 1989, 1527-1533.

[2] Tobii Technology, Home page: http://www.tobii.com/

[3] C. A. Chin, A. Barreto, and M. Adjouadi, "Enhanced Real-Time Cursor Control Algorithm, Based on the Spectral Analysis of Electromyograms, *Biomedical Sciences Instrumentation*, Vol. 42, pp. 249-254.

[4] C. A. Chin, A. Barreto, and M. Alonso, "Electromyogram-Based Cursor Control System for Users with Motor Disabilities", *Lecture Notes on Computer Science*, K. Miesenberger et al. (Eds.ICCHP-2006, LNCS 4061, pp. 905-912, 2006.

[5] A. Sesin, M. Ayala, M. Cabrerizo, and M. Adjouadi, "Jitter Reduction in Eye Gaze Tracking System and Conception of a Metric for Performance Evaluation", *WSEAS Transactions on Computers*, Issue 5, Vol. 3, pp. 1268-1273, 2004. * Also Presented at the 4th International Conference on Information Science and Applications.

[6] ISCAN INC., Raw Eye Movement Data Acquisition Software, Instruction Manual.

[7] A. Sesin, M. Adjouadi, and M. Ayala, "A Calibrated, Real-Time Eye Gaze Tracking System as an Assistive System for Persons with Motor Disability," *Proceedings of the 7th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2003)*, Vol. VI, pp. 399-404, Orlando, Florida, July 27-30, 2003.

[8] C. Daniel, & F. S. Wood, *Fitting equations to data: computer analysis of multifactor data, 2nd edition* (John Wiley & Sons Inc, NY 1980).

[9] M. Ayala, M. Adjouadi, "Eye-Gaze Tracking System Evaluator," Programming Tool [Online]. Available: www.cate.fiu.edu/CATEHomePage/Software/index.htm.

[10] Jacob, R. J. K. Eye-gaze computer interfaces: What you look at is what you get. *IEEE Computer 26*, 7 (July 1993), 65-67.