Analyzing Eye-Tracking Effectiveness With and Without Cursor Feedback During On-Screen Button Presses

Yiannis E. Papelis
Virginia Modeling Analysis & Simulation Center
Old Dominion University
Norfolk, VA, USA
ypapelis@odu.edu

Ginger S. Watson
Dept. of STEM Education and Professional Studies
Old Dominion University
Norfolk, VA, USA
gswatson@odu.edu

Kathryn Hicks
Virginia Modeling Analysis & Simulation Center
Old Dominion University
Norfolk, VA, USA
Kcatl001@odu.edu

ABSTRACT

Eye-tracking has traditionally been used as a means of measuring eye scanning behavior while reading, interacting with Graphical User Interfaces (GUIs) or in providing a user interface for assistive devices for individuals who may not be able to use more standard input mechanisms. As the technology matures there is extensive interest in using eye-tracking as a control mechanism, either augmenting or completely replacing an existing user interface device. Use of eye-tracking technology has the potential of increasing the control bandwidth between a user and a device, something that can be a critical advantage in complex interfaces. At the same time low accuracy or poorly designed interfaces could have a negative effect on ease of use. Thus, it is important to gain an understanding of how eye-tracking accuracy can affect the performance of different control tasks under various control approaches. In this paper preliminary results are presented on eye-tracking accuracy in a simple GUI button selection task with and without cursor feedback. This research is a precursor to a larger study that investigates eye-tracking as a means of controlling a tele-operated robot in a simulated search and the results will be used to optimize the design of discrete selection events in the GUI for robot control.

Analyzing Eye-Tracking Effectiveness With and Without Cursor Feedback During On-Screen Button Presses

Yiannis E. Papelis
Virginia Modeling Analysis & Simulation Center
Old Dominion University
Norfolk, VA, USA
ypapelis@odu.edu

Ginger S. Watson
Dept. of STEM Education and Professional Studies
Old Dominion University
Norfolk, VA, USA
gswatson@odu.edu

Kathryn Hicks
Virginia Modeling Analysis & Simulation Center
Old Dominion University
Norfolk, VA, USA
Kcatl001@odu.edu

INTRODUCTION

To date, eye tracking has been used to study users' attention patterns during task performance or as an aide that allows hands-free interaction with a computer for persons unable to use the traditional mouse and keyboard-based control inputs. As eye-tracking technology matures and the cost drops, it becomes feasible to utilize eye-tracking as a replacement for traditional control tasks. In certain environments, use of eye-tracking over traditional control approaches can have significant advantages. For example, there are several situations where a user cannot utilize their hands either because they are already busy with a different task as when operating multiple pieces of equipment or because of wearing clothing and/or protective equipment that prevents use of hands, i.e., wearing thick gloves. It may also be possible to increase the control bandwidth by allowing eye-tracking to complement traditional control methods. And in some cases, use of eye tracking fits naturally with the intended task, for example a camera that utilizes the user's eyes to focus the lens at the location that the user is currently looking.

At the same time, performance of eye tracking equipment can vary and corresponding interface techniques can have a negative effect on use, either due to low accuracy and/or poorly or inappropriately designed interfaces. In this paper results are presented from a study designed to 1) evaluate the typical accuracy of desktop tetherless eyetracking, 2) characterize the types of errors and their effect on use of eye-tracking for control tasks, and 3) evaluate the effect on providing on-screen feedback on a simple task that involves the virtual pressing of a GUI button using eye control. This research is preliminary and a pre cursor to a larger study that focuses on use of eyetracking as a means to teleoperate a ground robot and its on-board camera in a typical search task.

This paper is organized as follows. First, an overview of relevant prior work is provided followed by the design of the experiment, then results are provided along with a discussion on the findings.

PRIOR WORK

Eye tracking is currently used in a number of research applications, usually split between two categories: diagnostic and interactive. Diagnostic applications use the eye tracker to provide objective of quantitative feedback concerning the user's attention or reactions to stimulation. Interactive applications use the eye tracker as another input device, by replacing the usual computer mouse and allowing the eyes to control the location of a cursor on a screen or using knowledge of the user's gaze to alter what is seen on the display (Duchowski, 2002).

Control theories primarily focus on relying on eye tracking for disabled persons; however (Kumar et al, 2007) proposed the use of eye tracking to enter ATM passwords in order to reduce the chance of the password being stolen by a "shoulder surfer". Their study touched factors that may attribute to the effectiveness of using eye tracking, such as the distance between keys on the screen and modes of input: dwell or trigger. With the dwell method, users only had to focus on a key for a certain amount of time to activate it. With the trigger method, users had to focus on a key and then also press a button to activate it. This study concluded that the speed difference between the dwell and trigger methods was inconclusive. The trigger approach, however, showed significantly higher error rates due to users having difficulty properly timing their hands and eyes to coordinate perfectly (Kumar et al, 2007).

Such an approach to security is inspired by the systems that allow disabled persons to interact with a computer using just their eyes. One such system, called the Erica project, uses a tier fixation system for selection (Hutchinson et al, 1989). When the user's eyes have fixed for two to three seconds on a certain point, Erica plays a sound and a cursor appears in line with the gaze. If the user continues to focus on this cursor, a second tone sounds and the point on which they are focused is selected (Hutchinson et al, 1989).

When it comes to robotics, eye tracking is commonly used in the diagnostic sense. In one study eye tracking was used to study situational awareness while a single user attempted to control several robots (Ratwani et al, 2010). The user was required to direct five semi-autonomous UAVs to specific targets on a map while avoiding dynamically moving hazard areas. The eye tracker was used to monitor the user's attention in order to measure the user's cognitive processing and predict his situational awareness. This was done by measuring how much time a user focused on a specific robot or task as well as tracking his scan patterns (Ratwani et al, 2010).

Such scan patterns have also been used to study the effectiveness of graphical user interfaces for UAVs. By determining where on the screen the pilot of the UAV tends to focus, a user interface can be streamlined to make sure information is easily available. During a study that required users to execute certain actions with a UAV (turning a certain number of degrees, maintaining a certain altitude, etc.), speed of action changes was tested as well as the ability to perform such tasks while eye-gaze was tracked. This proved the necessity of an ergonomic user interface layout for the effective piloting of a UAV (Tvaryanas, 2004).

More recently, eye-tracking gaze control has been used in mobile phones as a means to facilitate usage. For example, phones manufactured by Samsung (King 2013) have an option that will scroll a page when the phone detects the user's eyes gazing toward the bottom of the screen. In addition, the user has the ability to enable an option that will prevent a phone from turning off the display while it detects that the user is reading the page (King 2013). Both of these features utilize the phone's on-board camera to assess the user's gaze in order to control a specific function.

On the technology front, Apple Inc. was recently awarded a patent (Patent 8,937,591) that addresses Troxler's fading, a phenomenon that causes visual stimuli to fade around a point, should a user focus on that point too long. Another company, FOVE has developed a head-mounted display that incorporates eye-tracking as a means to improve game player's performance; for example a player glancing at an on-screen character will begin the process of aiming a weapon toward the gaze target thus improving the player's chances of hitting their target. Also, while display a 3D scene, the user's gaze is used to determine the projection focus point and create appropriate 3D effects.

STUDY DESIGN

The aim of this study was to assess eye-tracking accuracy and characterize the types of errors that are typical when using eye-tracking technology, and finally to assess the effects of such errors on simple control tasks involving pressing a virtual on-screen button. The study consisted of two portions; an engineering 'sanity check' where the raw errors of the eye-tracking equipment were characterized and a controlled study where the performance of subjects using eye-tracking to virtually press an on-screen button can be compared with and without cursor feedback. Results of the second portion of this study are preliminary and presented here in descriptive form. The researchers served as participants for all experimental tasks in the pilot study. This allowed testing and refinement of the protocol prior to the larger study. Given the psychomotor nature of these tasks, researchers as participants introduced limited bias.

Equipment

For this test and subsequent experiment, a tetherless eye tracker manufactured by SmartEye was used. The model used was Smart Eye Pro, and consisted of a dedicated PC equipped with a four camera digitizer to which up to four cameras can be attached. Specific cameras were also equipped with an off-axis infrared illuminator. A software application run on a PC and utilized the attached cameras for performing the eye- and head-tracking function. Version 6 of the SmartEye software was used. Prior versions of the software required that a profile be generated for each user by manually registering specific points on the face which were used to identify the orientation of the head and further to estimate the eye gaze. Version 6 of the software performed these steps automatically and generated a profile of a user within a few seconds after initialization. The only required manual action was performing a calibration step that identified any fixed bias errors in the tracking and was used to reference to position of the head with the coordinate system established by the placement of the cameras.

Once a profile was generated and the gaze calibration completed, the eye-tracker estimated numerous variables that included but were not limited to the position and orientation of the head, the direction of the eye gaze, pupil diameter and eye closures. This data was broadcast within the local area network at 60 Hz. Both filtered and unfiltered versions of these variables were provided. For purposes of this study only the unfiltered intersection of the eye-gaze with the screen was used.

For this experiment, three cameras were used. Two of the cameras were mounted below the screen and the third camera was mounted on top of the screen. The cameras were mounted on a flat screen whose panel size was measured at 0.378 m in width and 0.301 m in height. The resolution of the panel was 1280 by 1024 pixels on the horizontal and vertical axis respectively. This information was provided to the eye-tracking software to ensure that the gaze to screen intersection is properly calibrated. Figure 1 depicts the screen with the eye-tracking cameras attached.

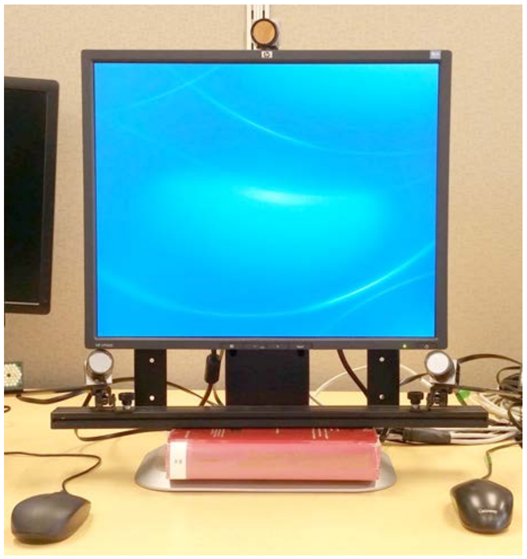
Figure 1 – Screen and Camera Configuration

Under the specified conditions the data of interest consists of a pair of numbers representing the pixel at which the user is looking at:

$$(x, y) \ : \ \begin{cases} 0 < x < 1279 \\ 0 < y < 1023 \end{cases} \qquad (1)$$

Engineering Assessment Portion

This portion of the study focused purely in the accuracy of the equipment and the characterization of any errors. To perform this assessment, the researchers utilized the equipment to temporarily stare at specific points on the screen with known coordinates and the data delivered by the eye-tracker were collected so it can be compared against the expected value. A pattern containing 49 dots arranged in a 7 by 7 grid was displayed on the screen. A number was printed above each dot providing the unique index of each point, from 1 to 49. The dots were laid out in row-order, starting on the upper left corner of the screen. Figure 2 depicts the image displayed on the screen.

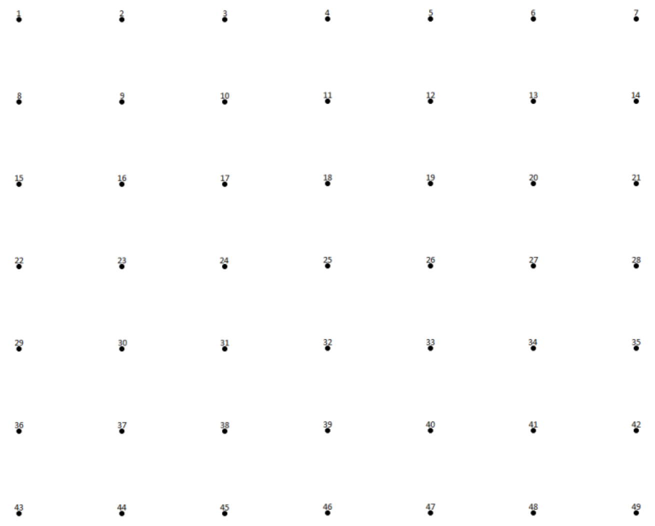

Figure 2 – Dot Pattern utilized for Accuracy Test.

To perform the test, each of the researchers took turns testing the equipment using the following process:

- A random ordering of the numbers between 1 and 49 was generated.
- For each number in this randomized order, the participant was told the number and asked to find the corresponding dot on the screen. This was to ensure that gaze samples were not captured while the user was searching for the dot on the screen.
- Once the participant indicated they had identified the location of the dot, they begun staring at the dot for a short interval during which the eye-tracker estimate of the gaze-screen intersection was collected.
- After the expiration of the interval, the participant was given the next number in the sequence and asked to identify its location on the screen.
- This process was repeated until all 49 points had been used.

A software tool was developed to facilitate the data collection. The tool generated the random sequence, provided the proper data collection interval and captured the eye-tracker data reflecting the intersection of the user's gaze with the screen. The data was then tabulated and analyzed.

Engineering Assessment Results & Discussion

Figure 3 shows the raw data for two of the test, in particular tests two and three. The x and y axis represent pixel locations on the screen. Red crosses mark the 49 reference points. The blue dots depict the corresponding samples of the gaze-screen intersection. An ellipse is drawn for each reference point. The center of the ellipse represents the numerical mean of the horizontal and vertical coordinates of the samples for each reference point (centroid), whereas the horizontal and vertical size of the ellipse reflects two standard deviations of the samples on the respective axis.

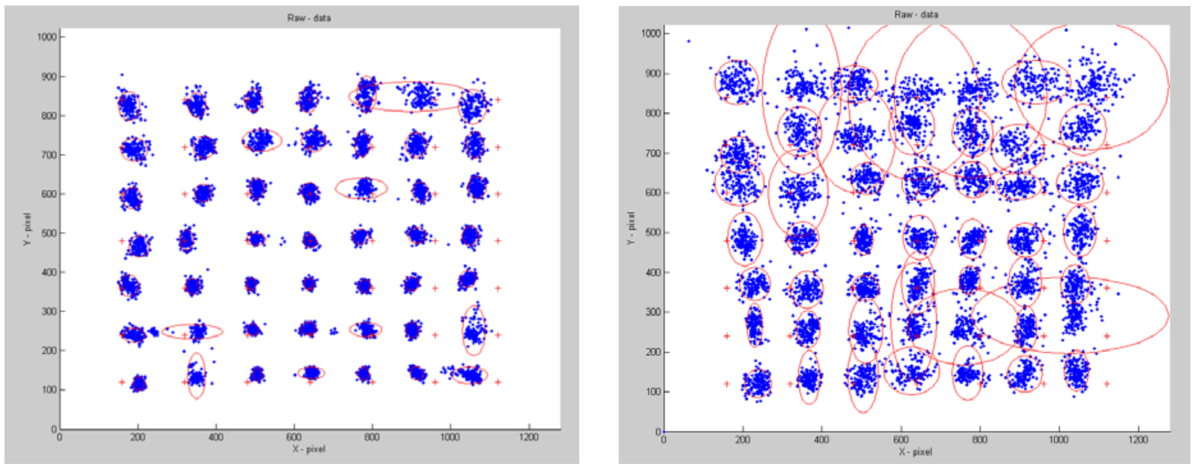

Figure 3 – Raw Data From Accuracy Test.

Table 1 shows the numerical values for the errors, in particular the mean and standard deviation for the centroid error on the horizontal and vertical axis for each of the tests.

	Test 1		Test 2		Test 3	
	Avg	StdDev	Avg	StdDev	Avg	StdDev
Horizontal	15.96	67.2	8.46	37.4	5.57	43.2
Vertical	-15.32	26.8	-7.43	10.3	-18.82	15.6

Table 1. Example of a Single Column Table

Figure 4 shows the error in pixels plotted against the Cartesian distance between the reference point and the center of the screen. This error is calculated by computing the Cartesian distance of each target point to the center of the screen. For example, point # 25 which is the 4th column of the 4th row is located at the center of the screen and its distance would be 0. Then the Cartesian error of each point is plotted against its distance to the screen. Effectively, points near the center of the screen appear on the left of the graph and the points on the perimeter of the screen appear on the right. The graph in Figure 4 shows the individual errors for all three tests, their average and a best-fit second order curve.

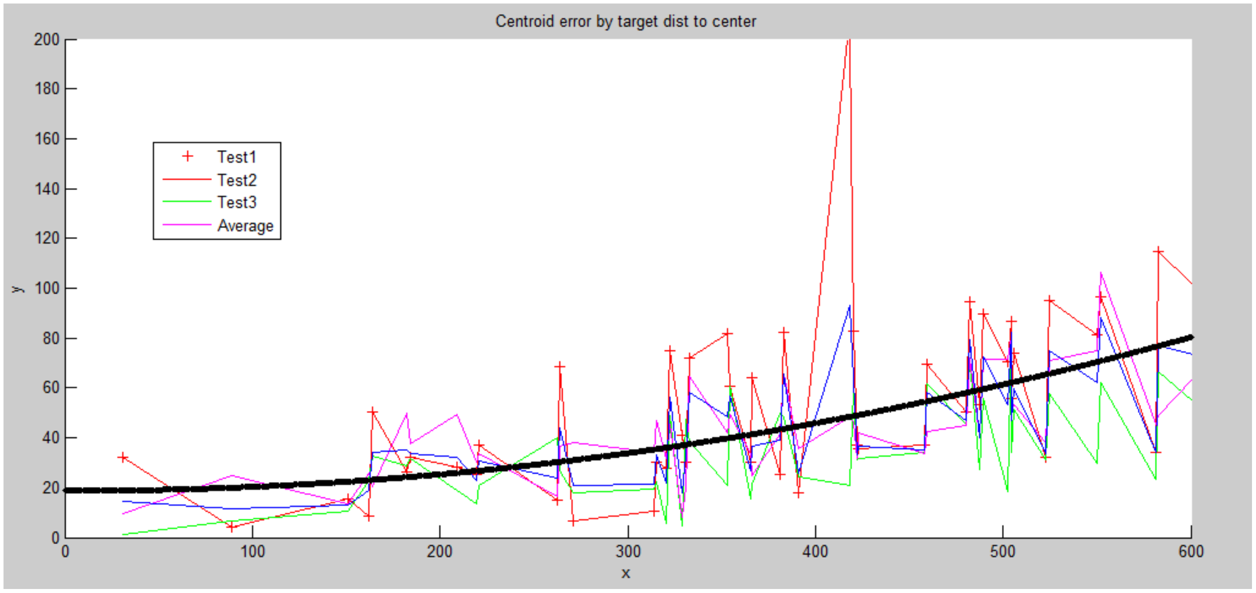

Figure 4. Cartesian Pixel Error vs Target Cartesian Distance to Screen Center.

The data from these tests provide significant insights into the potential usefulness of using eye gaze for control tasks. In particular, there are two relevant types of error; noise and bias.

Noise is variation of the gaze intersection even though the user is fixated on a point. Noise is caused by a variety of factors including quantization error of the cameras, visual noise in the camera images, and vibration of the camera mounts, among others. In addition, the eye direction does not remain constant but instead changes due to involuntary muscle movements. It is also possible that fatigue or external stimuli may cause a user's eye to rapidly scan away from an intended reference point even while the user staring at a specific target. Numerically, the noise is reflected by the standard deviation of the samples for each target – these are depicted as the horizontal and vertical size of the ellipses in Figure 3 and tabulated in Table 1.

The data here gives us some indication of the noise magnitude that one can expect in a typical environment. This information can be used to appropriately size GUI elements or active regions that are to be used for control tasks purely based on eye gaze inputs. As an example, one could utilize the data in Table 1 to properly size on-screen buttons, or estimate spacing for adjacent controls that are designed for activation through eye-gaze.

Bias represents a fixed error between the actual gaze location and the reported gaze location. This is numerically reflected by the Cartesian distance between the intended gaze location and the centroid of the samples captured for that target point. This is depicted as the center of the ellipse in Figure 3 and is also plotted explicitly against each target's Cartesian distance to the screen center in Figure 4.

It is important to note that as part of the initial calibration process, the eye-tracker uses four on-screen points to generate a reference that allows mapping the eye vector described from a camera-based coordinate system to a screen based coordinate system. It is unclear if the software utilizes these calibration points to correct for the off-center bias error. However, the data in this test suggests that the bias error varies significantly with respect to the distance between the gaze point and the screen center. This is evident in Figure 3 which shows that for most of the testers the bias is minimal for the reference points near the screen but grows for the reference points away from the center. Figure 4 shows a definite trend of increasing error as the distance to the center of the screen increases. Since this is an engineering test, there is not enough data to statistically analyze the trend; however, such biases can be eliminated by performing an additional multi point calibration for each user. This calibration will calculate the bias error at several screen locations and then can apply a correction to the gaze samples before use. This is something that is planned for implementation on future studies involving this eye-tracking system.

It is interesting to note that in all of these tests, the raw performance of the eye-tracking system is within the manufacturer specifications for angular error. This can be calculated based on the size of the screen, the screen resolution and the typical distance between the user's eyes and the screen. Our interest in characterizing the eye-tracking error is in understanding the degree to which the system can be utilized for control tasks involving on-screen GUI elements.

Virtual Button Press Portion

At the time of this writing, preparation for this portion of the research is ongoing. Preliminary engineering data was obtained for three tests and is presented here in descriptive form.

For this portion, a gray button was drawn against an empty black screen and the user was asked to focus on the center of the button. The size of the button was 80 pixels in width and 60 pixels in height. The button could be placed on one of 25 locations, arranged along a 5 x 5 grid. The procedure used for gathering the data in this portion is described below:

- One of the 25 buttons was selected at random and drawn on the screen.
- One second after the button appeared the system begun gathering data on the intersection of the eye-gaze with the screen.
- Two seconds later the button blinked to indicate selection and disappeared.
- The process repeated until all 25 button locations had been tested.

The process was repeated for two conditions; the first condition utilized no feedback on the actual gaze to screen intersection, while the second condition utilized a small cursor to depict the location of the user's gaze during the experiment. These conditions are referred to as cursor and no-cursor respectively. Presentation order was randomized for each research participant.

Figure 4 illustrates a button drawn in all possible grid locations; keep in mind that only one button was drawn at a time during data collection.

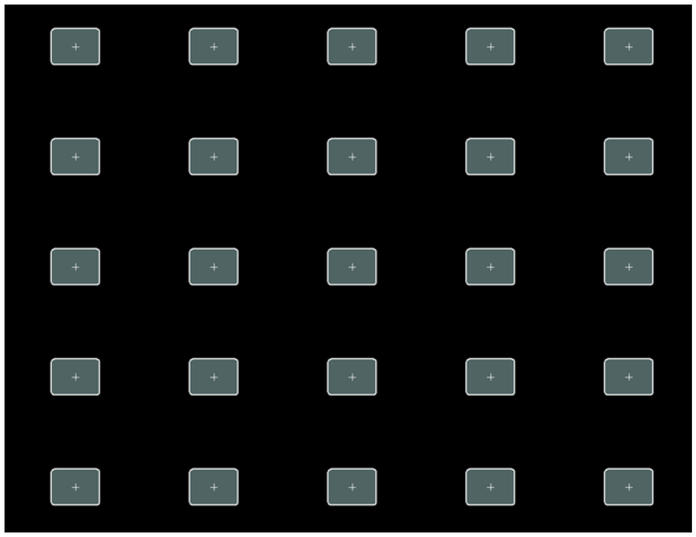

Figure 5. Illustration of button screen locations.

Figures 6 and 7 depict the raw data for the cursor and no-cursor conditions respectively. As in prior illustrations, a red cross depicts the location at which the respective button was drawn and the blue dots depict samples of the eye gaze to screen intersection collected for the two second period following the appearance of the button on the screen. Because of the approach utilized for drawing the cursor, the number of samples obtained for the cursor condition is less than the number of samples obtained for the no-cursor condition. While drawing the cursor some the eye-tracker packets were lost causing the difference in the number of samples obtained in each condition. This deficiency will be corrected in the future by utilizing a more efficient drawing routine but did not affect accuracy.

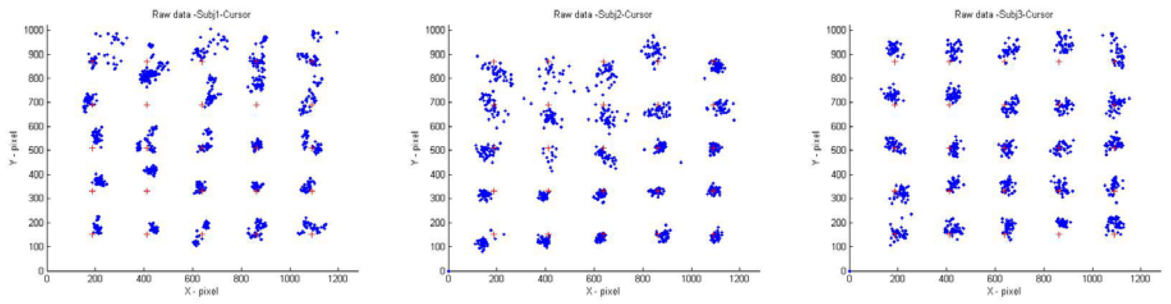

Figure 6. Raw Data for the Cursor Condition.

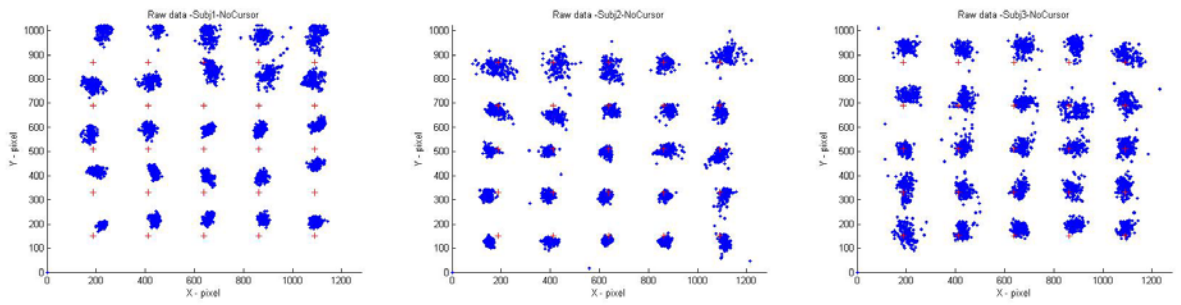

Figure 7. Raw Data for the No-Cursor Condition.

Table 2 provides a numerical summary of the data. For the cursor and no-cursor condition the average and standard deviation of the horizontal and vertical error are shown.

Cursor Condition			
Horizontal Mean	**Horizontal StdDev**	**Vertical Mean**	**Vertical StdDev**
-10.34	16.75	-34.43	27.87
20.74	66.4	48.64	100.55
-11.04	10.97	-22.63	22.62
	Mean: 31.37		Mean: 50.35
No Cursor Condition			
Horizontal Mean	**Horizontal StdDev**	**Vertical Mean**	**Vertical StdDev**
-2.05	38.9	-65.76	77.74
3.93	16.86	19.62	12.82
-15.81	11.42	-21.71	15.72
	Mean: 22.39		Mean: 35.43

Table 2. Preliminary Summary Statistics on Centroid Error.

Because this is engineering data collected on a limited number of subjects without the full experimental protocol, a full statistical analysis is not yet possible, however the data provides some evidence of trends. As expected, the larger bias error for target points away from screen center remains in this data set. However, there is a strong trend in the size of sample dispersion between the two conditions with a larger dispersion when the cursor is visible. One possible explanation for this trend is that when the cursor is visible and there is a bias error, the cursor is not drawn at the user's focus point and as a result, the user is more likely to shift their gaze to the cursor; in response, the cursor moves even further away thus creating the higher dispersion. Eventually, the user will focus back to the intended target point but the net effect is a larger amount of dispersion on the gaze data when the cursor is visible.

Figure 8 depicts the average error between the target point (button center) and the centroid of the samples for the two seconds after the button appeared on the screen. Given the few test runs the data is inconclusive, although there is a trend indicating higher error for the cursor condition.

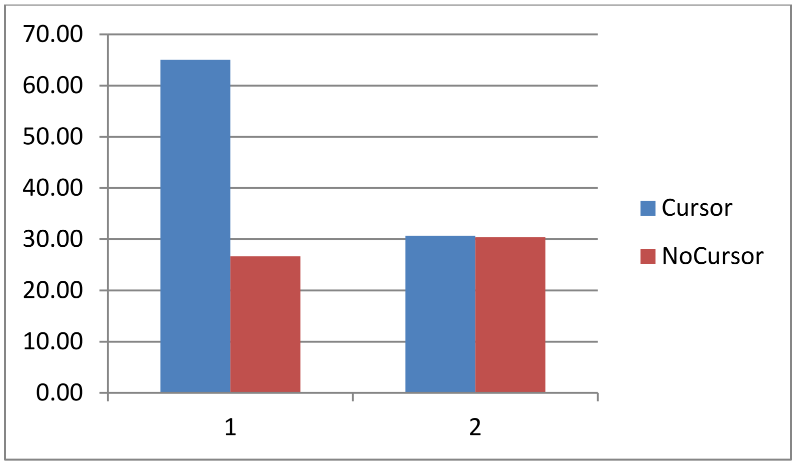

Figure 8. Average of Centroid Error Across All 25 Buttons.

Finally, Figure 9 depicts the standard deviation of the error for all 25 points.

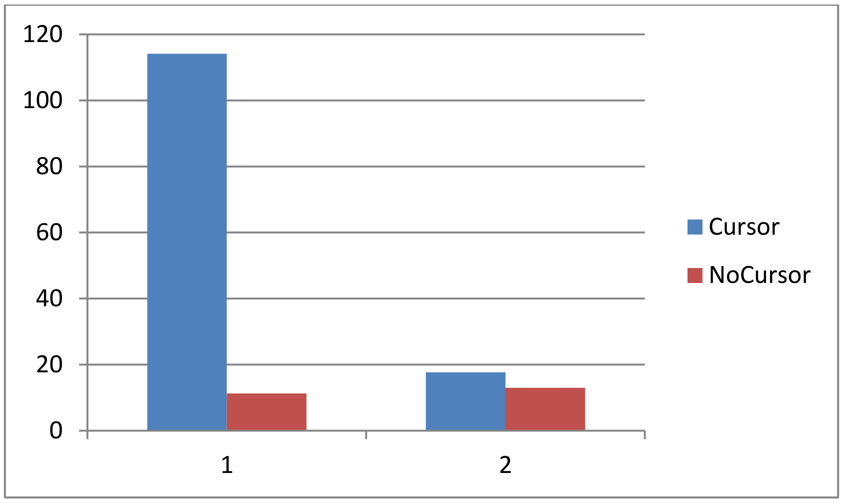

Figure 9. Standard Deviation of Centroid Error Across All 25 Buttons.

DISCUSSION

Preliminary data was presented on the basic engineering performance of a typical reconfigurable tetherless eyetracking system and the effect of cursor presence in accuracy when utilizing eye gaze to press a virtual on-screen button. Preliminary review indicates that there are several sources of systemic errors that need to be accounted for in order to successfully utilize eye-tracking for control tasks. Screen position dependent bias error affects accuracy away from a sweet spot but multi-point calibration could be used to compensate for such type of error. In addition to accuracy, the system appears to exhibit increased sample dispersion when the gaze is away from the sweet spot, something that is harder to compensate for, although dynamic filtering could be used. Another option to address screen-dependent system errors is to utilize additional cameras. The system utilized in this assessment can operate with a minimum of two cameras and up to four cameras but three cameras were used to gather the data presented here. It is also possible to configure the system with more than four cameras. It is expected that use of additional cameras would decrease both bias error and dispersion, although cost would also increase, something that would make a system less attractive for commodity uses.

The use of a cursor providing feedback for the on-screen estimate of the eye-gaze yielded mixed results. There appears to be increased error and increased dispersion of samples when a cursor is present versus when the cursor is absent. Anecdotally, this can be attributed to the fact that when there is a bias error, cursor presence has a distracting effect because it does not appear at the gaze location and has a tendency to create a gaze drift when the user adjusts their gaze to look at the cursor. At the same time, a user could potentially benefit from using the cursor as they could dynamically adjust their gaze based on the feedback and effectively eliminate bias error. Additional research is required to measure the effect of cursor not only on engineering data but also on the efficacy of button selection. A straightforward measure would be the time it takes to select a button based on the number of samples within the bounding box of the button. This work is currently ongoing.

REFERENCES

Duchowski, A. T. (2002) A breadth-first survey of eye tracking applications. Behavior Research Methods, Instruments, & Computers, 455-470.

Duchowski, A. T. (2007). Eye tracking: methodology theory and practice (2nd ed.). London: Springer.
Hutchinson, T. E., White, K. P., Martin, W. N., Reichert, K. C., & Frey, L. A. (1989) Human-computer interaction using eye-gaze input. IEEE Transactions On Systems, Man, and Cybernetics, 1527-1534.

Kumar, M., Garfinkel, T., Boneh, D., & Winograd, T. (2007) Reducing shoulder surfing by using gaze-based password entry. Proceedings of the 3rd symposium on usable privacy and security, Pittsburgh, Penn.: July 18-20.

Ratwani, R. M., McCurry, J. M., & Trafton, J. G. (2010) Single operator, multiple robots: an eye movement based theoretic model of operator situation awareness. 5th ACM/IEEE International Conference on Human-Robot Interaction, 243-250.

Tvaryanas, A. P. (2004) Visual scan patterns during simulated control of an uninhabited aerial vehicle (UAV). Aviation, Space, and Environmental Medicine, 75, 531-538.

King, R., (2013) Eye tracking and gesture will control future mobile devices, online article, accessed February 2015, http://www.biometricupdate.com/201303/eye-tracking-and-gesture-will-control-future-mobile-devices.

Ware, C. (2013). Information visualization: perception for design (3rd ed.). Waltham, Mass.: Morgan Kaufmann.