iZoom: Using Eye Tracking Hardware to Present a More Intuitive Interface to Pliable Display Technology

Michael Ashmore*
Clemson University

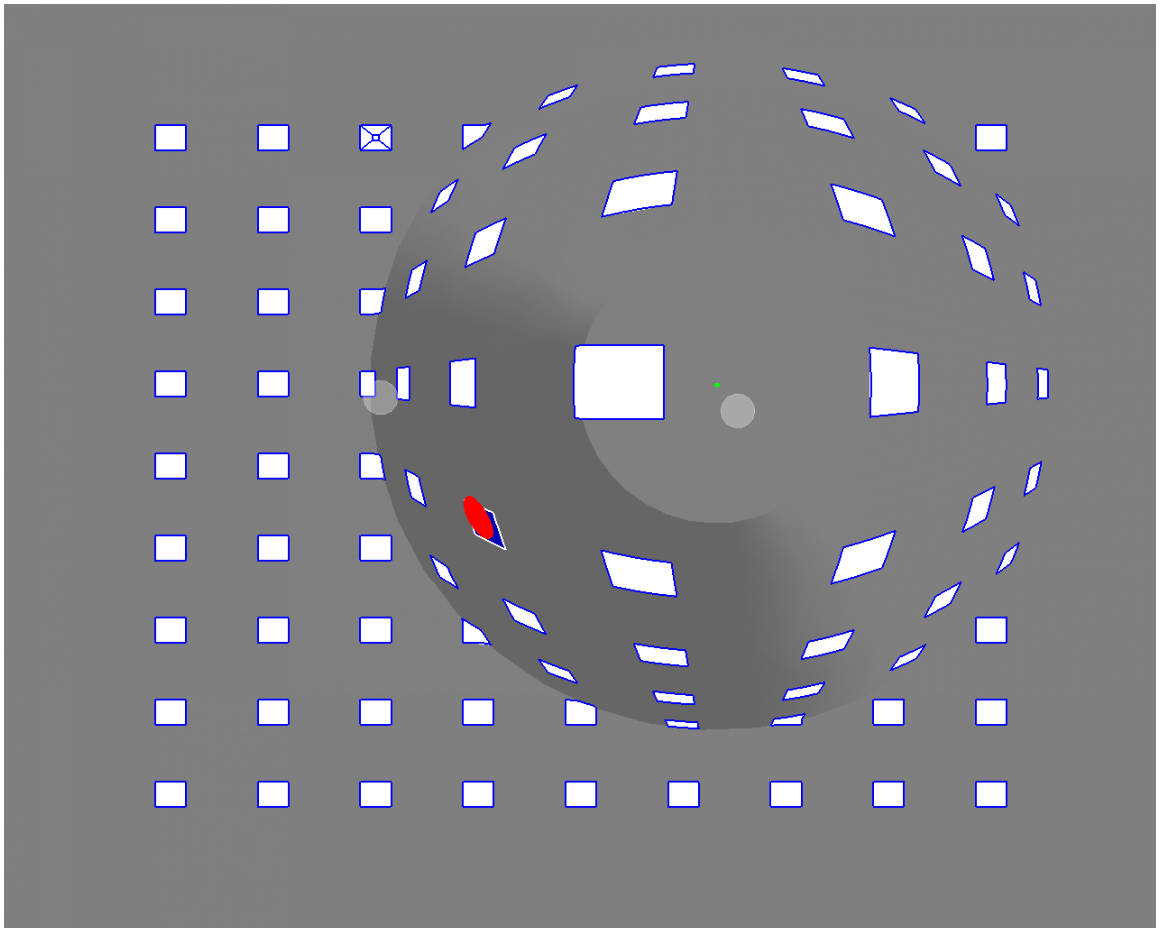

Figure 1: The iZoom fisheye display

Abstract

This paper investigates a refinement to existing eye mouse interfaces. iZoom is an application which monitors the user's gaze and interactively zooms the display to facilitate eye mouse pointing. Using Idelix Software's Pliable Display Technology (PDT), the user maintains a contextual view of the entire desktop while selectively zooming in on the region of interest. The novel interface may also be ergonomically superior to existing interfaces in applications where PDT is already in use.

CR Categories: B.4.2 [Hardware]: Input / Output and Data Communications—Input / Output Devices : Image Display; H.1.2 [Information Systems]: Models and Principles—User / Machine Systems; H.5.2 [Information Systems]: Information Interfaces and Presentation—User Interfaces; I.3.6 [Computing Methodologies]: Computer Graphics—Methodology and Techniques; I.4.10 [Computing Methodologies]: Image Processing and Computer Vision—Image Representation: Multidimensional;

Keywords: eye tracking, PDT, pliable display, zoom, interface

1 Introduction

Eye-based pointing devices, while technologically feasible, still face many challenges in the arena of practicality. Although technological refinements are allowing increasingly reliable detection of a user's focus, the eye's level of positional tolerance guarantees a certain degree of uncertainty in eye tracking systems [Jacob 1990]. Most graphical user interfaces in use today require selection of objects that are much smaller than a user's foveal view. When the region of the screen within foveal view encompasses more than one selectable object, not even the most accurate eye tracking system can determine with absolute certainty which of those objects is the

*e-mail: Mike.Ashmore@alumni.clemson.edu

true object of the user's attention.

Nevertheless, eye mouse interfaces are the only input alternative currently available for certain members of the motor-disabled community. Commercial packages designed for these users typically include a sort of "proxy" application which is designed for eye mouse interaction; the user issues commands to the proxy which the proxy then forwards to the underlying operating system. Unfortunately, this extra layer of software is generally "laborious and cumbersome" to use [Bates and Istance 2002]. A technique that allows direct interaction with the underlying user interface would be preferable, but as mentioned above, typical graphical user interfaces require finer-grained control over the cursor than eye mouse interfaces can reasonably be expected to provide.

The primary reason current GUIs are incompatible with eye mice, then, is because targets on the screen are too small to reliably select with an inherently inaccurate pointing device. It would seem that enlarging the targets would allow the eye mouse user to interact directly with the GUI, rather than relying on a proxy application. Statically enlarging our targets, though, is not a reasonable solution: if the all interface elements are larger, clearly fewer will fit within a given amount of screen real estate. Dynamically zooming interfaces seem to offer more promise.

2 Background

Zhai et al. have found that dynamic target expansion can increase pointing speed and accuracy with conventional input devices [Zhai et al. 2003]. Gutwin and Skopik have found that fisheye views (e.g. pliable displays), because of the context they include, are well-suited to steering tasks [Gutwin and Skopik 2003]; however, such views have been found to have an adverse effect on targeting tasks with input devices like handheld mice and trackballs [Gutwin 2002].

In 2002, Bates and Istance demonstrated an eye mouse system with a full-screen zoom facility. It was shown that the zoom capability improved eye mouse performance to a level competitive with head-controlled pointing devices [Bates and Istance 2002]. One of the disadvantages of the full-screen zoom, however, is the loss of context information: information outside the zoomed region is not displayed anywhere on the screen at all.

It seems a natural progression, then, to see if fisheye zooming is helpful for eye mouse tasks. Targeting was the task under study in this experiment.

3 Methodology

Our goal was to determine what effects a fisheye zoom mechanism may have on eye mouse selection accuracy and speed. We hypothesized that fisheye zoom would provide significant accuracy and speed benefits over non-zoomed interfaces; further, we considered it probable that a "smart" fisheye lens (one which only appeared when it was expected to be helpful in targeting) would be faster than a fisheye lens which was always present.

3.1 Apparatus

A Tobii 1750 eye tracker was used for this study. The Tobii 1750 is a 1280x1024 TFT flat panel monitor with an integrated eye tracking unit capable of binocular tracking at 0.5 degrees accuracy and 50Hz sampling. A Pentium 4 PC running Windows XP and Tobii software handled the task of interpreting the eye tracker's raw data; the processed data was then made available on a TCP/IP port. The TFT of the Tobii received its input from a dual-processor PC running RedHat Linux; the Linux machine also connected as a TCP/IP client of the Windows PC's gaze data. Custom software utilizing Idelix Software's Pliable Display Technology runs on the Linux machine, providing the gaze-contingent zooming interface and experimental data collection facilities.

3.2 Experimental Design

We wished to compare the merits of various interfaces in performing a simple selection task. A nine-by-nine grid of "windows" was drawn. Each window subtended 0.85 degrees of horizontal and 0.68 degrees of vertical visual angle when the user was positioned at the optimal 50 cm viewing range from the Tobii. A window was marked as a target by drawing an "X" through its center. The user's task was to select the target window by directing his gaze to the center of that window. A 500 ms fixation within the window was interpreted by the program as a selection of that window. The user was specifically instructed to try to adjust his gaze so that the cursor was centered in the window. The standard deviation σ of the points that lay within the window was then used to infer the smallest feature size that the user could have practically selected. Once the selection was made, the display visually confirmed the user's success by highlighting the window selected by the user and drawing a red circle centered at the fixation's mean position with a radius equal to σ.

Eye tracker status feedback was provided to the subjects by way of two translucent dots on the screen which gave a visual indication of the eye tracker's view of the subject's eyes, as well as by a small green dot which indicated the subject's gaze point as calculated by the eye tracker. Subjects were advised that the eye tracker functioned optimally when the eye-dots were roughly centered within the screen. Mapping was provided so that the eye-dots acted approximately as a mirror image of just the subject's pupils. When the eye tracker could not provide reliable data from one or both eyes, the eye dot corresponding to the troubled data disappeared from the screen to indicate that the eye tracker was functioning at reduced reliability and allow the user to adjust his orientation to correct the situation.

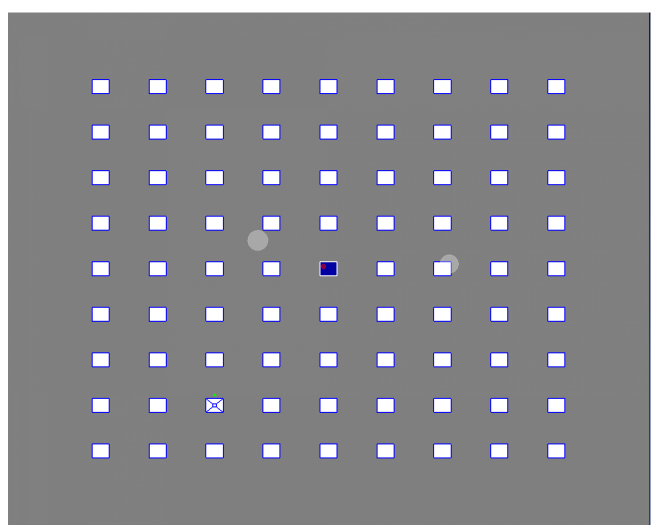

Figure 2: The iZoom user interface (non-fisheye condition).

The experimental study protocol presented three interface styles to the user:

- A non-zooming interface (the control condition)
- An interface using an ”always-on” fisheye view, with the magnified region of the screen selected based on the user's gaze fixation.
- An interface using an ”intelligent” fisheye view, similar to that described above, in which the fisheye magnifier stays hidden until certain fixation conditions are met.

Each subject completed three blocks of fifty trials. each block corresponding to one of the three interface styles. Each trial corresponded to a target window being randomly chosen, followed by the subject's correct selection of that target window (if the subject failed to correctly select the target window, the erroneous selection was logged, a new target window was selected, and the trial was repeated). Prior to each block, the subjects were given five practice trials in the new input mode to familiarize themselves with the interface. The order in which the interface styles were presented was counterbalanced across subjects to avoid confounds with time and practice.

3.3 Procedures

Candidate subjects were pre-tested with the eye tracker to ensure that they could achieve sufficiently precise calibrations. Eight subjects were used for pilot testing of the application; their input and results were used to guide further development of the iZoom application for future research.

Subjects were first placed 50 cm from the screen of the eye tracker and familiarized with the user interface elements of the iZoom application, particularly the relevance of the eye dots (see figures 2 and 1). At the beginning of the experiment, subjects performed a calibration routine to ensure the eye tracker was optimally detecting their gaze point. Nine calibration points (in a 3 x 3 grid) were used. At each calibration point, a yellow circle was first drawn centered on the point. The circle then shrank to a diameter of two pixels (still centered on the calibration point), at which point the eye tracker was instructed to gather 10 calibration samples. After the eye tracker had gathered its data, the circle grew back to its initial size and moved across the screen to the next sample point. At the new point, the circle shrank again, and so on, until a full set of calibration data had been acquired.

Once a complete set of calibration data was acquired, the Tobii was instructed to compute a calibration matrix and return results on the quality of the calibration. The quality results were inspected by the calibration routine and any points which showed unacceptable precision were automatically re-calibrated. Once all points were within acceptable thresholds (defined as a deviation of not more than 1/50th of the screen's dimension in either the X or Y axes) the experiment was allowed to proceed.

The experiment comprised three blocks, each consisting of fifty trials. Each block was presented in a different zoom condition: one block which included no zoom function (the control condition), one block using full-time fisheye zoom, and one block using the ”intelligent” fisheye zoom. The order of conditions was varied between subjects according to a standard latin square configuration. The experiment application automatically determined the necessary block order and created data log files based upon a subject number provided to it at runtime.

For each trial, the following items of data were logged:

- Trial identification information: the subject number, block number, and trial number within the block.
- The target window number. Target windows were numbered from 0 to 80, left-to-right and top-to-bottom.
- The selected window number. A window was considered selected if the subject had fixated on it for 500 ms. Subjects sometimes accidentally selected windows that were not the target; if this happened, the trial would log this fact and that trial would start over from the beginning.
- The mean position of the fixation that ended the trial. Ideally, this mean position would be in the center of the target window; in practice it rarely was. This allowed an analysis of the accuracy of the subject's targeting.
- The standard deviation of the fixation that ended the trial. This allowed an analysis of the precision of the subject's targeting.
- The mean deviation from the target's center. This allowed further analysis of targeting precision.
- And of course the time taken by the subject to complete the trial. Trials ran back-to-back; that is, the moment the first trial was complete, the timer started ticking on the next trial. The subject's objective was to complete all trials as quickly as practical.

A note on timing: As stated above, a trial started the moment the previous trial ended. The first trial within a block was preceded by a dialog box informing the user that they would be timed from the instant they dismissed the dialog. The final trial of a block obviously did not start timing any other trial; the timer was stopped pending the subject's response to the beginning-new-block dialog box.

A trial ended as soon as the subject had selected a window. Selection was considered to have occurred when the subject's gaze fell within any window on the screen for 500 consecutive milliseconds. If, during a trial, the user selected a window that was not the target window, that information was logged and that particular trial was restarted (i.e., a new target was selected and the trial number was not incremented).

Upon completion of all three blocks, subjects were thanked heartily and sent on their way.

4 Results

Data was initially analyzed by importing the log files collected during experimental runs into a PostgreSQL database and running a script of SQL commands developed for this particular experiment.

If a subject made n erroneous selections in the course of a single trial, that trial would have $n+1$ entries in the log file (the final entry being the one where the correct selection was made). A total of 1,262 entries were logged; 62 were immediately discarded because the subject had made an incorrect selection. Of those incorrect selections, thirty were made under the selective fisheye condition; sixteen were made in each of the non-fisheye and full-time fisheye conditions.

Correct selections were then analyzed to extract outliers. The outlier analysis algorithm defines the following for each trial record `tr`:

Condition	Mean Trial Time	Mean Speed Index	Mean Accuracy	Mean Accuracy Index
No Fisheye (control)	2919 ms σ=2227 ms	1	17.4 px σ=4.9 px	1
Selective Fisheye	3192 ms σ=1980 ms	0.91	16.5 px σ=4.7 px	0.94
Full-Time Fisheye	3803 ms σ=2866 ms	0.76	16.2 px σ=4.3 px	0.99

Table 1: Summary of results.

`mean` The mean of all records matching `tr.subject` and `tr.condition` which have not yet been determined to be outliers

`stddev` The standard deviation of all records matching `tr.subject` and `tr.condition` which have not yet been determined to be outliers

If a trial record `tr` has a time that is outside of `mean` plus or minus three times `stddev`, `tr` is considered an outlier and is placed in a ”discard” table. Of course doing so changes the values of `mean` and `stddev`, so the analysis algorithm has to start over each time it discovers a new outlier.

After multiple iterations, the analysis algorithm eventually reaches a stable equilibrium where there are no more trial records that can be placed in the discard table. At that point it exits, returning a table of trial data that is free of outliers on a per-subject, per-condition basis.

Eight subjects performing 50 trials in each of three conditions resulted in 400 ”correct” trial records per condition (1200 ”correct” trial records total). Of these, the outlier analysis algorithm discarded twenty trial records from the full-time fisheye condition and thirty-three each from the non-fisheye and selective fisheye conditions (86 trial records total).

Finally, means were calculated for the trials that remained.

A two-sample t-significance test on the speed difference between the selective fisheye and control conditions yields a t-value of 1.75, which with seven degrees of freedom implies an approximate 85% level of likelihood that the control condition is faster.

The same test between the full-time fisheye and control conditions yields a t-value of 4.72. Given seven degrees of freedom, we may conclude that there is a 99.7% probability that the control condition is faster than the full-time fisheye view.

Considering the accuracy results, our t-value for the control versus selective fisheye conditions is 2.54. This gives us a 96% probability that selective fisheye is more accurate than the control condition.

The t-value for the control versus full-time fisheye is 3.55, indicating 99% probability that full-time fisheye gives greater eye-pointing accuracy than the control condition.

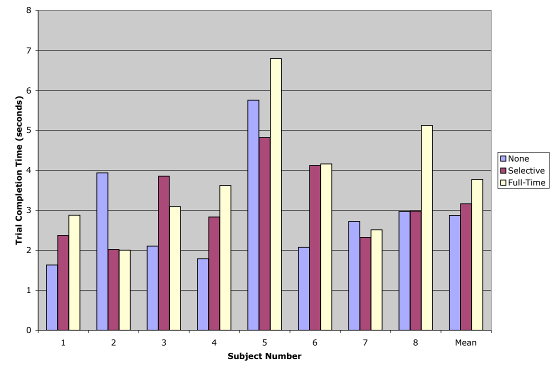

Figure 3: Mean trial times, per subject.

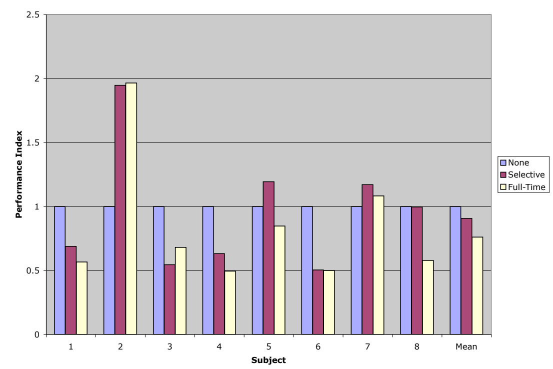

Figure 4: Speed indices, per subject.

When interviewed, five of the eight test subjects said they preferred using the selective fisheye zoom. Two preferred the non-zoomed interface, and one preferred the full-time fisheye zoom.

5 Discussion

It is encouraging to see increased accuracy when zoomed interfaces are used. It makes intuitive sense, though: with a larger target, accuracy ought to improve. The analogy might be made with a telescopic sight on a rifle: seeing the target in greater detail allows more precise positioning of the cross-hairs.

It is somewhat surprising, however, that the control condition appears to be significantly faster than either of the zoomed interfaces. One would have expected Fitts’ law to apply: although the pointing distance remains constant, the target is enlarged, so we expected to see target selection time decrease with magnified targets. As reliably as Fitts’ law has been applied in the past, we are led to wonder if flaws in our experimental method may have compromised the usefulness of our data. Certainly the experiment was not perfect.

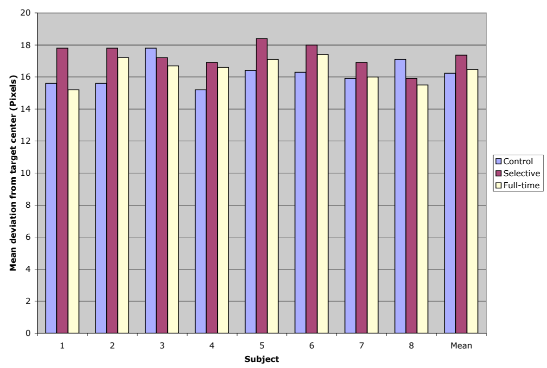

Figure 5: Mean accuracy, per subject.

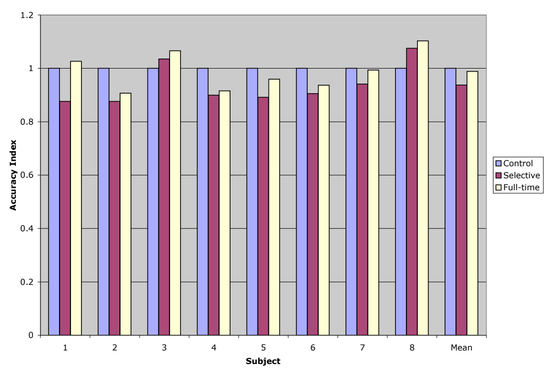

Figure 6: Accuracy indices, per subject.

While collecting data, we noticed several potential areas for improvement in a next-iteration version of the iZoom experiment.

- Many subjects had a great deal of difficulty achieving consistent accuracy across the entirety of the Tobii screen. This has been addressed by researchers in the past, notably Jacob. In that system, the user was allowed to adjust the eye tracker calibration by selecting a point on the screen with a hand-held mouse, looking at that point, and clicking the mouse to create an immediate calibration correction [Jacob 1990].

- Calibration had a tendency to drift, as is typical of eye tracking systems, but the drift was particularly exacerbated by the loopy lighting system installed in the VR lab. When lights turned themselves on and off (seemingly) spontaneously, users saw their accuracy decrease significantly. Duct tape over the motion sensors will definitely be used in future eye tracking research.

- As an additional approach to the drifting calibration problem, it may be possible to use each trial's selection as another calibration point. That is, since we know where the center of the selected rectangle is, and we know the position the eye tracker reported the subject's eyes in, we could have a calibration grid that updated itself each time a target was selected.

- When a user was struggling with a poor calibration, the zoomed modes became particularly difficult to use effectively. Since the lens' focus lags behind the user's gaze by about 80 ms, a slightly inaccurate targeting motion frequently degenerated into a positive feedback loop between the user's eye and the lens positioning logic that left the lens bouncing back and forth between positions that exactly managed to miss the intended target. Perhaps a filter could be designed to detect such feedback and suppress it.

- The start condition for each trial was not well controlled. The user's gaze could be almost anywhere on the screen when a trial started, so we had no way to determine how far their gaze traveled before fixating on the next target. This can be easily controlled for: a current development version of iZoom includes logic so that the user starts a trial by fixating on a target at the center of the screen.

- The targets used in this study are still quite large by the standards of current e.g. office application designers. Shrinking the targets further would slow performance in all conditions, granted, but would most likely have a disproportionately large effect on the unzoomed condition. Nevertheless, making the target windows five-by-five pixels when they aren't zoomed smacks of cheating, and offends this researcher's sense of fairness.

It is far too early to declare Fitts' law dead as applied to gaze-based selection. Nevertheless, we had presumed we would find such a significant effect that minor flaws in our experiment would be of minimal consequence. This was not, as it turns out, a good assumption to make.

6 Conclusion

As anticipated, zooming interfaces were responsible for a significant improvement in selection accuracy. As implemented for this experiment, though, they are detrimental to selection speed. Further research is ongoing in hopes of finding a method by which the selection speed of zoomed interfaces may be improved.

References

BATES, R., AND ISTANCE, H. 2002. Zooming interfaces!: enhancing the performance of eye controlled pointing devices. In *Proceedings of the fifth international ACM conference on Assistive technologies*, ACM Press, 119–126.

GUTWIN, C., AND SKOPIK, A. 2003. Fisheyes are good for large steering tasks. In *Proceedings of the conference on Human factors in computing systems*, ACM Press, 201–208.

GUTWIN, C. 2002. Improving focus targeting in interactive fisheye views. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM Press, 267–274.

JACOB, R. J. K. 1990. What you look at is what you get: eye movement-based interaction techniques. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM Press, 11–18.

ZHAI, S., CONVERSY, S., BEAUDOUIN-LAFON, M., AND GUIARD, Y. 2003. Human on-line response to target expansion. In *Proceedings of ACM CHI 2003 Conference on Human Factors in Computing Systems*, vol. 5, ACM, 177–184.