A Comparison of Smooth Pursuit- and Dwell-based Selection at Multiple Levels of Spatial Accuracy

Dillon James Lohr
Texas State University
San Marcos, TX 78666, USA
djl70@txstate.edu

Oleg V. Komogortsev
Texas State University
San Marcos, TX 78666, USA
ok11@txstate.edu

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author. Copyright is held by the owner/author(s).
CHI'17 Extended Abstracts, May 06-11, 2017, Denver, CO, USA
ACM 978-1-4503-4656-6/17/05.
http://dx.doi.org/10.1145/3027063.3053233

Abstract
In this paper, we present a smooth pursuit–based alternative to dwell-based selection for eye-guided user interfaces. Participants attempt to perform both dwell- and pursuit-based selections while we artificially reduce the spatial accuracy of an affordable eye tracker to see how resilient both selection methods are. We find that the time to perform a pursuit-based selection remains consistent even as spatial accuracy degrades, unlike dwell-based selection which takes considerably longer to perform the worse the spatial accuracy becomes. We argue that smooth pursuit–based selection will be important in eye-tracking systems with low spatial accuracy, such as very low cost trackers, certain self-made systems, and calibration-free systems.

Author Keywords
Eye tracking; gaze interaction; selection; smooth pursuit

ACM Classification Keywords
H.5.2 [Information interfaces and presentation (e.g., HCI)]: Input devices and strategies (e.g., mouse, touchscreen)

Introduction
Eye tracking is becoming a popular method for interacting with devices. With the introduction of affordable eye-tracking devices intended for mainstream use, such as the FOVE [5], a virtual-reality headset with eye tracking ca-

pabilities, eye tracking is reaching a larger audience than ever before. But eye tracking is not always accurate or user-friendly. It typically requires an initial calibration phase before each use. Even then, all but the highest quality eye trackers frequently suffer from poor spatial accuracy, especially if users are not positioned correctly and secured by a chin rest. The FOVE, for example, has a tracking accuracy as poor as 1 degree [5], which could prove problematic when more exact control is needed.

Likewise, self-made eye trackers and calibration-free systems have rather poor accuracy. The self-made tracker used in [3] had an average accuracy of 3.55°. Hennessey and Lawrence demonstrated that a traditional calibration-free system might have an average accuracy of 1.34 cm (1.24°), and their enhanced calibration-free system averaged 0.85 cm (0.79°) [2]. Though impressive, even an error of 0.79° could be troublesome for navigating interfaces on smaller displays.

To select objects on the screen–arguably the most important task of any user interface–most eye-guided interfaces use dwell-based selection, which requires a user to stare at a target for a predetermined amount of time [9]. This method of selection can work very well in a controlled environment when using a high quality eye tracker with exceptional spatial accuracy. However, dwell-based selection might not work well if an eye tracker does not have great spatial accuracy (possibly in the case of affordable mobile and wearable devices) or if a user does not remain still while navigating an eye-guided interface. In the case of poor accuracy, a user's recorded gaze may not match where he or she is actually looking, which makes it difficult to use dwell-based selection. Similarly, head movements interfere with an eye tracker's ability to accurately identify where on the screen a user is looking (though, for head-mounted devices such as the FOVE, this may be less of a problem).

Ideally, the accuracy of the tracking device would be no worse than half the size of the smallest selectable object on the screen. This would maximize the user-friendliness of the interface by allowing a user to select any object by (at the very least) looking at its center. For the aforementioned eye trackers [5, 3, 2], objects on the screen would need to be unreasonably large to satisfy this usability requirement.

That is where pursuit-based selection excels. Smooth pursuit selection needs only the relative movement of the eye rather than its exact position, meaning spatial accuracy is virtually meaningless for performing pursuit-based selections. Therefore, an initial calibration phase would be unnecessary, so a user could simply, say, put on a wearable device and immediately begin navigating a pursuit-based eye-guided interface.

Background

Many studies have attempted to address the problems that plague dwell-based selection. Head gestures [8] and saccadic eye movements [6] have been explored as alternative selection methods with promising results. However, these methods have their own downfalls. Using head gestures could increase the rate of fatigue onset in users, and accurately detecting saccades requires an eye tracker with both great spatial accuracy and high temporal precision to function best.

Pursuit-based selection is a relatively new alternative to dwell-based selection. Videl, Bulling, and Gellersen first showed how performing selections with smooth pursuits makes it possible for a user to instantly begin using an eye-guided interface, even without calibration or instructions on how to navigate it [10]. Esteves et al. employed smooth

$$r = \frac{\sum_{i=1}^{n}(g_i-\bar{g})(s_i-\bar{s})}{\sqrt{\sum_{i=1}^{n}(g_i-\bar{g})^2}\sqrt{\sum_{i=1}^{n}(s_i-\bar{s})^2}}$$

Figure 1: The Pearson Product-Moment Equation

Consider a window of length n with mean gaze position $\bar{g}$ and mean stimulus position $\bar{s}$. The correlation between the gaze and stimulus positions within that window is r, given by this equation.

A shorter temporal window (a smaller n) can lead to faster selection times, while a longer window (a larger n) may be more robust to accidental selections and noisy signals.

pursuit selection as a means to navigate a smart watch interface [1]. Špakov et al. were the first to compare pursuit- and dwell-based selection techniques [7].

Though pursuit- and dwell-based selection techniques were compared in [7], this comparison did not investigate how both methods might be affected by different levels of spatial accuracy. This information could be used to find the point at which one method becomes better than the other.

We expand upon previous works by comparing both dwell- and pursuit-based selection at various levels of spatial accuracy to see how resilient both selection methods are against the poor spatial accuracy that may be experienced when using affordable, self-made, or calibration-free eye trackers.

Additionally, our method of smooth pursuit selection differs from those used previously. Videl et al. used randomly moving targets as the stimuli; Esteves et al. used circular stimulus movements; and Špakov et al. used a combination of circular and linear stimulus movements. Our method, however, involves stimuli moving back and forth along straight lines connected to the target objects, fitting the style of a radial menu. Also, the nodes in our experiment gradually progress toward being selected, which differs from how nodes instantly become selected in other experiments.

Pursuit-Based Selection Method

The key component of smooth pursuit selection is having a moving stimulus for a user to follow and comparing its movement with the movement of the user's eyes. For our method, we maintain a short temporal window of gaze and stimulus positions and compare the correlation of the two sets of positions using the Pearson product-moment test (see Figure 1).

If r, that is the correlation between the data sets in the temporal window, exceeds a threshold, then the object progresses toward being selected. With this test, the distance between any given gaze point and the stimulus is unimportant, so the spatial accuracy of an eye tracker would have virtually no effect on the correlation.

More specific implementation details are described in the following section under *Selection nodes*.

Experimental Method

Hardware & software

The experiment was conducted on a computer with a 4.0 GHz quad-core processor and 16 GB RAM. Stimuli were presented on a 19" monitor with a pixel resolution of 1280×1024 and a 60 Hz refresh rate. Participants' eye positions were recorded with a consumer-grade eye tracker, the Tobii EyeX Controller (which has a temporal resolution of roughly 60 Hz). We recorded each participant in binocular mode, but we only used information from the left eye.

Participants

We recorded 11 participants (8 males, 3 females) with normal or corrected-to-normal vision. 3 participants were unable to perform the smooth pursuit selections, possibly due to excessive noise in their eye position signal, so their data were removed, leaving 8 participants (6 males, 2 females) for consideration in our results.

Participants were positioned 550 mm away from the monitor, and a simple chin rest was used to keep their heads relatively still to prevent unnecessary bias against dwell-based selection. For each participant, the eye tracker was calibrated once before any selections were made to determine the base spatial accuracy. The average base spatial accuracy for the 8 participants was 0.55° ± 0.31°.

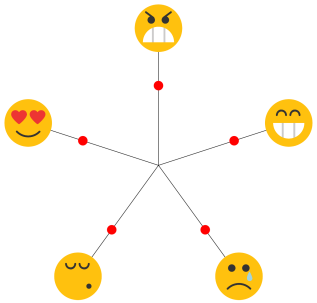

Figure 2: Selection Nodes

The layout of the 5 different emoji that were used for the nodes. They were presented to participants as (clockwise from the top): angry, happy, sad, tired, and in love. In this figure, the nodes are pursuit-based, so there are stimuli moving back and forth along lines. (These emoji were designed by Roundicons from Flaticon.)

Each node has a diameter of 19 mm (about 2° of the visual angle, or 64.5 pixels on the monitor we used), equal to the side length of a square Windows 10 start menu tile as it appears on a Microsoft Surface Book laptop. This reference size was chosen because it is a practical size for use in eye-guided interfaces. The layout was designed to mimic a very simplistic radial menu.

Selection nodes

The objects which participants were instructed to select (henceforth called nodes) are circular emoji representing different emotions (shown in Figure 2).

For dwell-based selection, each node simply checks whether the newest gaze in the temporal window of eye positions lies within the circular bounds of that node. If it does, the node progresses toward being selected.

For pursuit-based selection, each node is attached to a line along which a stimulus moves (as seen in Figure 2). One end of the line connects to the center of the node, and the other end connects to the center of the screen. The stimulus we used is a small, red circle with a diameter of about 5.3 mm (0.55° of the visual angle, or 18 pixels), and it travels along the line with a speed of 50.8 mm/s (about 5.3°/s, or 172 pixels/s). As the stimulus moves along the line from the center of the screen toward the node, the position of the stimulus is added to the temporal window at the same rate as new gaze positions are added to the window. Once the edge of the stimulus collides with the edge of the node, the stimulus reverses direction and begins moving along the line away from the node. It reverses direction once again when its edge collides with the opposite end of the line, and it repeats this motion until a selection is made. If the correlation coefficient from the Pearson product-moment test is above a threshold (we used 0.6, chosen empirically for our experiment), the node progresses toward being selected.

As a node progresses toward being selected, the user receives visual feedback in the form of a semi-transparent color filling in the node, starting from the center and radiating outward.

Methodology

First, participants performed the calibration routine designed by Tobii, the manufacturer of the eye tracker we used. This is a modified 5-point calibration routine in which participants pop the displayed dots (located either at the center or one of the four corners of the screen) by staring at them. The purpose of this calibration was to allow for the calculation of the base spatial accuracy with a verification procedure. Outside of this experiment, a calibration routine would not be necessary for smooth pursuit selection to work.

Participants next performed a verification procedure which was used to compute the base spatial accuracy. For this procedure, 30 stimuli (evenly arranged in a 6x5 grid filling the whole display) were presented one at a time. Each stimulus was displayed for 2 seconds, and the first and last 0.5 seconds of gaze information was discarded for each one. Once all 30 points finished displaying, we used the calculations detailed in [4] to determine the base spatial accuracy.

After the verification procedure, each participant was presented with 5 different emoji (see Figure 2) in a pentagonal pattern and instructed to select a specific one by a prompt at the top of the screen such as "Select the angry one." Each time a selection was made, there was a one-second pause before the next selection started. If a selection was not made after 10 seconds, that selection attempt was deemed a failure and the next selection began. After 10 selections were made, the selection method changed. Once both selection methods were performed 10 times, the accuracy of the eye tracker was artificially reduced (described in Figure 3). We randomized the order of selection methods for each participant. The experiment lasted approximately 15 minutes.

$$g'_x = g_x + d\cos\theta$$
$$g'_y = g_y + d\sin\theta$$

Figure 3: Artificial Accuracy Reduction Equations

To artificially reduce the accuracy of the eye tracker by d degrees, we first chose a random value, θ, in the range [0,2π). Then, the gaze components, g_x and g_y, become g'_x and g'_y from these equations.

For our experiment, the value of θ was randomized for each selection attempt to prevent a learning bias, and d held one of the values {0,1,3,6} depending on the participant's progress.

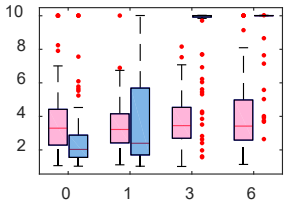

Figure 4: A boxplot of the selection times for both selection methods at each level of accuracy reduction. Failed selections were treated as 10-second selections (the timeout period). In each cluster, pursuit is on the left and dwell is on the right. The horizontal axis is the amount of accuracy reduction (degrees), and the vertical axis is the selection time (seconds).

Results

The selection times for both selection methods as the spatial accuracy degraded are shown in Figure 4.

We performed a Mixed Model Analysis of Variance with selection method (SM) modeled as a repeated measures factor and accuracy reduction (AR) as a covariate. The effect of SM, AR, and their interaction were tests. There was a power interaction between SM and AR. This effect was followed up with a series of estimates of least-squares means and estimates of differences in least-square means for SM at various levels of AR.

At the baseline accuracy (AR = 0), pursuit-based selection (P) had an average selection time of 3.7$\pm$2.0 s, and dwell-based selection (D) had an average selection time of 2.7$\pm$1.9 s. This difference was not significant (t = -0.76, p = .4456).

At AR = 1, P had an average selection time of 3.4$\pm$1.5 s, and D had an average selection time of 4.1$\pm$3.2 s. This difference was found to be significant (t = 4.08, p < .0001).

At AR = 3, P had an average selection time of 3.7$\pm$1.4 s, and D had an average selection time of 8.8$\pm$2.5 s. This difference was found to be significant (t = 16.51, p < .0001).

At AR = 6, P had an average selection time of 4.1$\pm$2.2 s, and D had an average selection time of 9.4$\pm$1.7 s. This difference was found to be significant (t = 19.13, p < 0.0001).

To address the possibility of a speed-accuracy bias, we also compared the number of successful but incorrect selections (i.e., the selection of a node that was not displayed in the prompt) made for each selection method. Of the 320 overall selections attempted using dwell selection, 162 (50.6%) were correct, 10 (3.1%) were incorrect, and 148 (46.3%) were failed attempts. Of the 320 overall pursuit selection attempts, 258 (80.6%) were correct, 52 (16.3%) were incorrect, and 10 (3.1%) were failed attempts.

Discussion

These results show that pursuit-based selection had a significantly faster selection time on average than dwell-based selection; however, the former had a noticeable increase of unwanted selections compared to the latter, constituting just over 16% of all pursuit-based selection attempts compared to only 3% of dwell-based selection attempts.

We realize that artificially degrading accuracy does not accurately reflect how either selection method would perform on, say, a custom-built eye tracking device. Therefore, the results obtained through this experiment should be viewed more as rough performance estimates.

To better imitate an interface, it may have been more ideal if each emoji remained in the same position every time (e.g., the angry emoji would always appear at the top position instead of being randomly placed for each selection). Additionally, the prompt being at the top of the screen may have created a bias toward the node at the top position. This could be remedied by placing the prompt in the center of the nodes, which would have the added benefit of better mimicking what a real interface might look like.

Also, some of the emoji appeared ambiguous for a few participants. For example, some participants confused the *in love* and *happy* ones, or the *tired* and *sad* ones. The prompt could display a smaller version of the target emoji instead of text to address this problem.

There were also those few participants who were unable to perform any smooth pursuit selections. While we are not certain what caused this issue, a potential explanation is

excessive noise in those participants' eye movement signals. Employing some kind of filter on the raw signal–such as a Kalman filter or even a simple averaging filter–might fix the issues we had.

Lastly, the pursuit-based selections in our experiment required over 3.5 seconds on average to perform. In real use cases, this amount of time would probably not be practical. The process of gradual selection we employed in our experiment is certainly at fault, so this process would likely need to be refined.

We would be interested in expanding upon our research by also investigating how changes in spatial precision affect both pursuit- and dwell-based selection. Another interest of ours is seeing how well pursuit-based selection would work on mobile and wearable devices like the FOVE.

Conclusion

In this paper, we compared the performance of both smooth pursuit- and dwell-based selection under multiple levels of spatial accuracy. We found that the time to perform a pursuit-based selection remains consistent even as spatial accuracy degrades, unlike dwell-based selection which takes considerably longer to perform the worse the spatial accuracy becomes. Pursuit-based selection significantly outperforms dwell-based selection with as little as one degree worse accuracy over the baseline accuracy. Even at baseline accuracy, the difference between the two methods is not significant. Our findings support the results of the study by Špakov et al. [7].

Clearly, dwell-based selection does not work well when the spatial accuracy of the eye tracker is poor, and it requires a user to keep his or her head still for optimal performance. With smooth pursuit selection, self-made and calibration-free eye-tracking devices with poor spatial accuracy could easily and effectively be used, leading to a more affordable and user-friendly eye tracking experience.

Acknowledgments

This work is supported in part by Google Faculty Research Award #2014_R1_308 and NSF CAREER Grant #CNS-1250718.

References

[1] Augusto Esteves, Eduardo Velloso, Andreas Bulling, and Hans Gellersen. 2015. Orbits: Gaze Interaction for Smart Watches using Smooth Pursuit Eye Movements. *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology* (2015), 457–466.

[2] Craig A. Hennessey and Peter D. Lawrence. 2009. Improving the Accuracy and Reliability of Remote System-Calibration-Free Eye-Gaze Tracking. *IEEE Transactions on Biomedical Engineering* 56 (2009), 1891–1900. Issue 7.

[3] Corey Holland and Oleg Komogortsev. 2012. Eye Tracking on Unmodified Common Tablets: Challenges and Solutions. *Proceedings of the Symposium on Eye Tracking Research and Applications* (2012), 277–280.

[4] Kenneth Holmqvist, Marcus Nyström, and Fiona Mulvey. 2012. Eye Tracker Data Quality: What It Is and How to Measure It. *Proceedings of the Symposium on Eye Tracking Research and Applications* (2012), 45–52.

[5] FOVE Inc. 2016. FOVE: Eye Tracking Virtual Reality Headset. https://www.getfove.com/. (2016). Accessed: 2 January 2017.

[6] Oleg V. Komogortsev, Young Sam Ryu, Do Hyong Koh, and Sandeep M. Gowda. 2009. Instantaneous Saccade Driven Eye Gaze Interaction. *Proceedings of the International Conference on Advances in Computer*

Entertainment Technology (2009), 140–147.

[7] Oleg Špakov, Poika Isokoski, Jari Kangas, Deepak Akkil, and Päivi Majaranta. 2016. PursuitAdjuster: An Exploration into the Design Space of Smooth Pursuit–based Widgets. *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research and Applications* (2016), 287–290.

[8] Oleg Špakov and Päivi Majaranta. 2012. Enhanced Gaze Interaction using Simple Head Gestures. *Proceedings of the 2012 ACM Conference on Ubiquitous Computing* (2012), 705–710.

[9] Linda E. Sibert and Robert J.K. Jacob. 2000. Evaluation of Eye Gaze Interaction. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2000), 281–288.

[10] Mélodie Vidal, Andreas Bulling, and Hans Gellersen. 2013. Pursuits: Spontaneous Interaction with Displays based on Smooth Pursuit Eye Movement and Moving Targets. *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (2013), 439–448.