In Proceeding of the ETRA, ACM
This is the corrected version of the paper. The definitive copy of the paper can be downloaded from portal.acm.org

PursuitAdjuster: An Exploration into the Design Space of Smooth Pursuit –based Widgets

Oleg Špakov*, Poika Isokoski, Jari Kangas, Deepak Akkil, and Päivi Majaranta
Tampere Unit for Human Computer Interaction
School of Information Sciences
University of Tampere, Tampere, Finland

Abstract

In a study with 12 participants we compared two smooth pursuit based widgets and one dwell time based widget in adjusting a continuous value. The circular smooth pursuit widget was found to be about equally efficient as the dwell based widget in our color matching task. The scroll bar shaped smooth pursuit widget exhibited lower performance and lower user ratings.

Keywords: eye tracking, smooth pursuit, uncalibrated.

Concepts: **• Human-centered computing ~ Human computer interaction (HCI);** *Interaction techniques*;

1 Introduction

Eye tracker error is conventionally discussed in terms of two components: precision and accuracy [Nyström et al. 2013]. Problems with accuracy are difficult to overcome. Conventionally the gaze is considered as a pointer and objects are activated by dwelling on them. A typical design for dealing with the inaccurate gaze point estimation is to make the GUI widgets so large that they can accommodate some pointing inaccuracy.

A different approach based on smooth-pursuit eye movement has been proposed recently [Vidal et al. 2013; Vidal et al. 2015]. In smooth pursuit techniques targets move on a display. Each target has a unique track. A user follows the desired target and the system recognizes this based on the timing and direction of the eye movements. The recognition then launches a state change in the user interface. It can be a single event such as the entry of a character [Cymek et al. 2014; Lutz et al. 2015] or a continuous adjustment that begins and continues as long as the target is followed [Esteves et al. 2015].

The advantage of smooth pursuit tracking is that the gaze point estimation error becomes a non-issue. As long as the gaze moves on a trajectory that is similar enough to the trajectory of the desired target, the system can operate regardless of accuracy.

We investigated smooth pursuit in adjusting a continuous value using a linear (scroll bar) and circular (knob) widgets. Both widgets had two moving targets used as cues for gaze. One target was for decreasing the value and the other for increasing it. In both widgets the smooth pursuit targets traveled in opposite directions: in the circular widget smoothly around it, and in the linear widget jumping back to the other end when they reached the end of the scrollbar. These widgets were then compared to a dwell time based widget that had two stationary fixation targets. Our goal was to compare the performance of dwell and smooth pursuit operated GUI widgets in the same task.

2 Related Work

Gaze based interaction techniques can be divided into four subtypes 1) dwell-based interaction, 2) gesture-based interaction, 3) steering-based interaction and 4) smooth pursuit –based interaction.

In dwell based interaction a central trade-off is the duration of the dwell time. Longer dwell time is slow, but reduces accidental activations. In some contexts, such as eye typing, experienced users can use shorter dwell-times. For example, Majaranta et al. [2009] found that novice users may need up to 1000 ms dwell time, but experts can manage with less than 400 ms. Dwell-time selection needs accurate per-user calibration.

Gaze gestures [Drewes and Schmidt 2007] are sequences of relative eye movements and hence tolerant to gaze tracking inaccuracy. Hyrskykari et al. [2012] compared gaze gestures and dwell-based interaction for discrete commands and found that gestures are a robust alternative to dwell-based interaction.

Unlike gestures, steering requires a fairly accurate tracker. The simplest gaze-based steering interface is to make the vehicle move where the user looks at. Examples of steering actions include steering of robots [Tall et al. 2009] and wheelchairs [Wästlund et al. 2010]. Such techniques have also been used in virtual worlds [Stellmach and Dachselt 2012]. Also the Dasher text entry system [Ward et al. 2010] utilizes steering control techniques.

Smooth pursuits are slow eye movements (<30 °/s) where the eyes follow a moving object. Vidal et al. [2013; 2015] were the first to use smooth pursuit eye movements for spontaneous, calibration free human computer interaction. They concluded that a) the shape of the trajectory does not impact performance for small number of targets, and b) a circular trajectory allows for more robust detection when number of objects is large. They recommend setting a window size of 500 ms for pattern matching algorithm.

Khamis et al. [2015] demonstrated that the smooth pursuit technique is well perceived by novice users for spontaneous interactions with public displays. Pfeuffer et al. [2013] used the smooth pursuit technique for gaze tracker calibration. Esteves et al. [2015] investigated interactions with a smartwatch. They used

*e-mail:oleg.spakov@uta.fi.

ETRA '16, March 14-17, 2016, Charleston, SC, USA
© 2016 ACM. ISBN 978-1-4503-4125-7/16/03…$15.00
DOI: http://dx.doi.org/10.1145/2857491.2857526

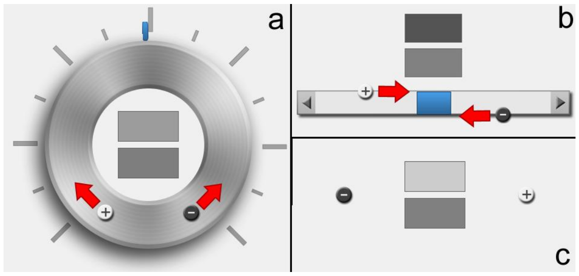

Figure 1. *The rotary (knob, a), linear (scrollbar, b) and dwell-time (c) controls. The arrows show the target movement direction. They were not shown to users*.

multiple objects moving in circular trajectory in the watch interface and invoked a command when user followed an object by gaze. They also noted that the smooth-pursuit technique could be used for continuous control by using the duration of the pursuit to modify a continuous parameter. Lorenceau [2012] used sustained smooth-pursuit eye movement for cursive writing. Lutz et al. [2015] achieved a text entry rate of 3.34 words per minute without training. Cymek et al. [2014] demonstrated that smooth pursuit technique can be used to enter PIN information robustly even with an uncalibrated tracker.

Most of the previous work has shown promising results for smooth-pursuit based interaction but no study has compared the technique with the dwell-time. All the previous works on smooth-pursuit use it for discrete interactions. We investigated continuous interactions and compared smooth pursuit with a dwell-based technique.

3 PursuitAdjuster

3.1 Design Rationale

We designed two variants of a control that can adjust a number in a range. The first variant, in Figure 1a, was inspired by a rotary knob. It is operated by following one of the pursuit targets (small circular object with + or - on it) with the gaze. These targets circulate around the control into opposite directions. Similar design was used by Esteves et al. [2015] in the wristwatch context. The second variant, in Figure 1b, was inspired by a scroll bar. The targets travel to one end of the control and then jump back to the other end. This causes a discontinuity in the following, but occupies a smaller space than the circular control. The two gray rectangles in Figure 1 are related to the grayscale adjustment task in our experiment.

Because the smooth pursuit needs to go on for a while for the algorithm to detect it and start adjusting the number, the users need feedback. We added a “tick” sound that was played re-peatedly as long as the number was adjusted. The rotary control had an indicator that showed its position in the range of possible values. The scrollbar had the usual scroll box.

3.2 Implementation

The task in our experiment was to adjust the brightness level (range 0-255) of a gray rectangle to approximately match the target tone given on a rectangle above it. The color codes were not shown. The starting level was 128. Ten levels on each side of the starting level at a distance from 28 to 100 levels with a step of 8 levels were the targets. The targets were the same for all participants but their order was randomized.

The experiment application implemented three control methods, labeled (1) rotation, (2) scrollbar, and (3) dwell time. The first two methods implemented the rotary and scrollbar versions of the PursuitAdjuster shown in Figures 1a and 1b. The third method, shown in Figure 1c, was based on dwell time.

The visual appearance of the gaze targets (ø = 0.9°) was the same in all methods: the target with a “+” sign indicated increasing the brightness of the box, and the target with “-” sign indicated decreasing it. A short “tick” sound was heard every 100 ms as long as the adjustment continued. Between the ticks the value was adjusted three times. It was possible to stop at any value.

The rate of change of color was 33 levels/s. The speed of the targets in both techniques was 3.3 °/s[1]. The path radius in the rotary control was 3.5°, i.e. the length of the path was 22.3°, and targets made a full circle in 6.8 s. The length of the scrollbar was 11°; the targets traveled it in 3.3 s.

Both smooth pursuit based methods utilized the same pursuit detector. The pursuit detection algorithm measured the standard deviation (SD) of distances between the gaze points and the pursuit targets during 600 ms. The idea was that the distance between the gaze point and the pursuit target stays approximately the same as long as the user is following a given target . Thus, the SD of the distances will be the smallest for the followed target. Gaze data was resampled at 30 Hz and smoothed before calculating the distance vectors using a mild low-pass filter $G_i = (G_i + \alpha G_{i-1}) / (1 + \alpha)$ with α=0.5.

Because users do not always follow targets, an additional rule for detecting this was needed. In such situations the standard deviations for all targets are “large”. In pilot testing we found that 0.4° worked fine and set that as the threshold.

We also found that when targets were moving in parallel, applying the algorithm resulted in similar standard deviations. Therefore, we restricted rapid target switching. Unless the standard deviation of another target was smaller for at least 200 ms, the current target would remain tracked.
Also, with the rotary widget the standard deviations could decrease below the threshold while the gaze was still (e.g. while inspecting the adjusted gray level in the middle). Therefore, we required that the length of gaze path was in the range of 0.5 – 1.5 times of the length of a target path. These improvements resulted in robust pursuit detection.

Overall, competitive smooth pursuit detection algorithms can be implemented in many ways as demonstrated in the works discussed earlier. We do not claim that our algorithm is any better than the other known algorithms. Any algorithm that was robust enough would have served our purpose in the experiment.

The dwell time based method (see Figure 1c) mapped gaze points on fixation targets using large gaze-responsive areas (4.2° wide).

[1] Because the angular speed is critical for smooth pursuit and the signal to noise ratio of the tracker is also expressed in degrees, we report the path lengths, target sizes, and velocities in degrees and °/s, assuming the participants’ eyes were 60 cm away from the display.

The dwell time measurement was based on the modified competitive time accumulation algorithm described by Räihä [2015 p.5]. However, the accumulated time was not reset after exceeding the threshold as in the original algorithm, but the controllable value was adjusting as long as the gaze was focusing on the target. To match the smooth pursuit techniques, the dwell time threshold was set to 600 ms, and the gray level changing speed was 33 levels/s.

4 Method

4.1 Participants

We recruited 12 participants (7 males, 5 females, mean age 30) from the university community. The participants had normal or corrected to normal vision. None of the 12 participants reported difficulties in perceiving colors, none had used gaze tracking applications before this experiment. The participants received credits towards satisfying course requirements.

4.2 Procedure

The experiment began with a short introduction to the goals of this study followed by signing a consent form. The participants were seated in front of the eye tracker's monitor at a distance about 60 cm. The eye tracker was then calibrated.

The experimental software displayed a control widget and two boxes in the center of the application window. The upper box displayed the target level of gray. The participants were instructed to make the gray level in the lower box approximately the same as it was in the upper box, as fast as they could.

The participants were familiarized with one of the widgets at a time. The order of methods was counterbalanced between the participants. It was emphasized that simply looking at the dark and light pursuit or fixation targets was all that was necessary to change the brightness.

The participants first completed a practice block (20 trials), and then continued with two experiment blocks. The participants then filled in a questionnaire with five subjective ratings regarding to the usability of the control method just used: performance, controllability, easiness, pleasantness, and speed of change. The rating scales consisted of 9 steps, from -4 (negative experience) to 4 (positive experience).

Using the same procedure, the two other control methods were then evaluated. At the end of the session the participants were asked to rate their tiredness, pleasantness of the audio feedback and to rank the control methods from the most preferable to the least preferable. Finally, they were asked to express their experience, thoughts, and comments freely.

4.3 Apparatus

We used a Tobii T60 gaze tracker with a 19" display (1280 x 1024 pixels) to collect gaze data. The experiment software was developed using .NET 4.5 framework in C# and was running on a standard PC (Intel i5-2520M 2.5Hz CPU, 8GB RAM, MS Win7). We wanted to test the smooth pursuit techniques under realistic conditions with a poorly calibrated tracker. However, we needed a well calibrated tracker for the dwell control blocks. Thus, we did

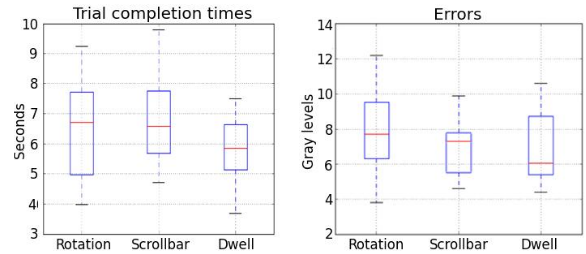

Figure 2. *Task completion times (left) and errors (right). Note that in this boxplot the line in the middle shows the median.*

normal calibration and applied an offset to the tracker data before our algorithm processed it. For each participant a random offset of 5-11° was computed. The same offset was active in all smooth pursuit blocks for that participant.

5 Results

For each trial the software recorded the gray levels in the boxes and the task completion time (TCT). The difference between the box colors was the level adjustment error. Before analysis we removed trials that took three times longer than the mean value (0.8% of the trials, all longer than 20 s).

Objective measures

The trial completion times are shown in Figure 2, left. According to the Shapiro-Wilk test, one of the compared pairs (rotational-dwell) had non-normal distribution of paired differences. Thus we used non-parametric Wilcoxon's matched pairs signed ranks test. Only the difference between the scrollbar control and the dwell time control was statistically significant ($z=-2.4$, $p=0.015$). The adjustment errors are shown in Figure 2, right. Typical error was just below 8 levels. There were no statistically significant differences between the widgets.

Subjective measures

In subjective ratings the main finding was that the scrollbar control was rated lower than the others. Using Wilcoxon's test, the difference in controllability was statistically significant ($p<0.05$) against rotational control and the dwell. In easiness dwell was rated higher than scroll ($p<0.003$) and rotational ($p<0.048$). No other statistically significant differences were found.

The ranking results are shown in Figure 3. Eight participants ranked the dwell time control as the best. Four ranked the rotary control as best. The favorite method was often characterized as the easiest, most accurate, requiring least effort, and most natural. Also easy inspection of the grey level adjustment result was given as justification for the highest rank.

Many participants noted that although the audio feedback was

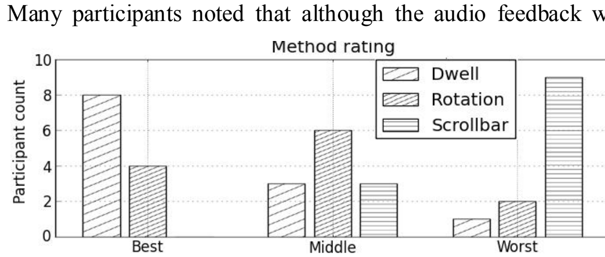

Figure 3. *Method ratings*

useful, it was not pleasant. Also, following the moving cues resulted in eye tiredness. At least three participants noted that the “+” and “-“ signs on the targets were confusing, and they would expect them to be switched. About half of the participants reported that at the beginning (mostly, in the practice session) it was hard to remember that the adjustable gray level was shown in the lower box. Sometimes, participants started the adjustment to the wrong direction. Some participants noted that it was easier to inspect the gray level adjustment result with peripheral vision when using “dwell time” method.

6 Discussion, Conclusions and Future work

The main finding was that the smooth pursuit techniques were competitive with dwell time in performance and user preference. The scrollbar technique was somewhat inferior to the dwell and the rotary techniques, but the difference was not large. In user interfaces with limited screen space it might be useful and usable.

It was not a surprise that the scrollbar technique performed worse than the others. It had the inherent handicap of a 0.6s pause at each jump from one end of the bar to the other. This pause was a natural consequence of the eye following the jump with a delay that interrupted the smooth pursuit detection.

The positive findings regarding the competitive performance of smooth pursuit controls should be taken with caution given that we only tested approximate continuous adjustment. The results may not generalize to other kinds of tasks. Perhaps the smooth pursuit techniques were competitive only because the pursuit detection window and the dwell time were both 600ms. Dwell time can be shorter in some tasks but it is not clear whether the smooth pursuit detection window can be shortened.

Also, our experiment included only 12 young adults. Given the small number of participants our estimates of the population performance are not very accurate. Also, other user populations may exhibit different performance.

One participant noted that, after checking the color boxes, it was easy to continue the adjustment with the “dwell time” method because the targets stayed at a fixed location. With the smooth pursuit methods, the targets moved and it took a moment to find the correct target to follow. This would not be an issue in real life scenarios, such as changing the level of sound or lighting, where the user perceives the change immediately without the need to divert the gaze from the followed target.

Much work remains to be done in optimizing the smooth pursuit detection algorithms and the rates of adjustment. Also other tasks besides the color matching must be experimented with to map the area of validity of the present results.

The smooth pursuit techniques showed great promise in allowing precise control with an eye tracker regardless of the tracker offset. They can be very valuable interaction techniques with wearable trackers in real-world control tasks. Other smooth pursuit-based widgets should be constructed to make building complete user interfaces possible.

Acknowledgements

This work has been funded by the Academy of Finland projects HAGI (decision number 260179) and MIPI (decision number 266285).

References

CYMEK, D. ET AL. 2014. ENTERING PIN CODES BY SMOOTH PURSUIT EYE MOVEMENTS. *IN J. OF EYE MOVEMENT RESEARCH,* 7(4), 1–11.

DREWES, H. AND SCHMIDT, A. 2007. INTERACTING WITH THE COMPUTER USING GAZE GESTURES. *IN PROC. OF INTERACT'07*, 475–488.

ESTEVES, A., ET AL. 2015. ORBITS : GAZE INTERACTION FOR SMART WATCHES USING SMOOTH PURSUIT EYE MOVEMENTS. *IN PROC. OF UIST'15*, 10 P.

HYRSKYKARI, A., ISTANCE, H., AND VICKERS, S. 2012. GAZE GESTURES OR DWELL-BASED INTERACTION? *IN PROC. OF ETRA'12*, 229–232.

KHAMIS, M., ALT, F., AND BULLING, A. 2015. A FIELD STUDY ON SPONTANEOUS GAZE-BASED INTERACTION WITH A PUBLIC DISPLAY USING PURSUITS. *IN PROC. OF UBICOMP'15*, 863–872.

LORENCEAU, J. 2012. CURSIVE WRITING WITH SMOOTH PURSUIT EYE MOVEMENTS. *CURRENT BIOLOGY* 22.16, 1506-1509.

LUTZ, O., ET AL. 2015. *SMOOVS :* TOWARDS CALIBRATION-FREE TEXT ENTRY BY GAZE USING SMOOTH PURSUIT MOVEMENTS. *IN J. OF EYE MOVEMENT RESEARCH,* 8(1):2, 1–11.

MAJARANTA, P., AHOLA, U.-K., AND ŠPAKOV, O. 2009. FAST GAZE TYPING WITH AN ADJUSTABLE DWELL TIME. *IN PROC. OF CHI'09*, 357–360.

NYSTRÖM, M., ET AL. 2013. THE INFLUENCE OF CALIBRATION METHOD AND EYE PHYSIOLOGY ON EYETRACKING DATA QUALITY. *BEHAVIOR RESEARCH METHODS 45*, 1, 272–288.

PFEUFFER, K., VIDAL, M., AND TURNER, J. 2013. PURSUIT CALIBRATION: MAKING GAZE CALIBRATION LESS TEDIOUS AND MORE FLEXIBLE. *IN PROC. OF UIST'13*, 261–269.

RÄIHÄ, K.-J. 2015. LIFE IN THE FAST LANE: EFFECT OF LANGUAGE AND CALIBRATION ACCURACY ON THE SPEED OF TEXT ENTRY BY GAZE. *IN PROC. OF INTERACT'15*, 402-417.

STELLMACH, S. AND DACHSELT, R. 2012. DESIGNING GAZE-BASED USER INTERFACES FOR STEERING IN VIRTUAL ENVIRONMENTS. *IN PROC. OF ETRA'12*, 131–138.

TALL, M., ET AL. 2009. GAZE-CONTROLLED DRIVING. *IN PROC. OF CHI EA'09*, 4387-4392.

VIDAL, M., BULLING, A., AND GELLERSEN, H. 2013. PURSUITS: SPONTANEOUS INTERACTION WITH DISPLAYS BASED ON SMOOTH PURSUIT EYE MOVEMENT AND MOVING TARGETS. *IN PROC. OF UBICOMP'13*, 439–448.

VIDAL, M., BULLING, A., AND GELLERSEN, H. 2015. PURSUITS: SPONTANEOUS EYE-BASED INTERACTION FOR DYNAMIC INTERFACES. *ACM SIG-MOBILE MOBILE COMPUTING AND COMMUNICATIONS REVIEW* 18, 4, 8–10.

WARD, D. J., AND D. J. C. MACKAY. 2002. FAST HANDS-FREE WRITING BY GAZE DIRECTION. *NATURE*, 418.6900.

WÄSTLUND, E., SPONSELLER, K., AND PETTERSSON, O. 2010. WHAT YOU SEE IS WHERE YOU GO. *IN PROC. OF ETRA'10,* 133-136.