Feedback for Smooth Pursuit Gaze Tracking Based Control

Jari Kangas
School of Information Sciences
University of Tampere, Finland
jari.kangas@uta.fi

Oleg Špakov
School of Information Sciences
University of Tampere, Finland
oleg.spakov@uta.fi

Poika Isokoski
School of Information Sciences
University of Tampere, Finland
poika.isokoski@uta.fi

Deepak Akkil
School of Information Sciences
University of Tampere, Finland
deepak.akkil@uta.fi

Jussi Rantala
School of Information Sciences
University of Tampere, Finland
jussi.e.rantala@uta.fi

Roope Raisamo
School of Information Sciences
University of Tampere, Finland
roope.raisamo@uta.fi

ABSTRACT

Smart glasses, like Google Glass or Microsoft HoloLens, can be used as interfaces that expand human perceptual, cognitive, and actuation capabilities in many everyday situations. Conventional manual interaction techniques, however, are not convenient with smart glasses whereas eye trackers can be built into the frames. This makes gaze tracking a natural input technology for smart glasses. Not much is known about interaction techniques for gaze-aware smart glasses. This paper adds to this knowledge, by comparing feedback modalities (visual, auditory, haptic, none) in a continuous adjustment technique for smooth pursuit gaze tracking. Smooth pursuit based gaze tracking has been shown to enable flexible and calibration free method for spontaneous interaction situations. Continuous adjustment, on the other hand, is a technique that is needed in many everyday situations such as adjusting the volume of a sound system or the intensity of a light source. We measured user performance and preference in a task where participants matched the shades of two gray rectangles. The results showed no statistically significant differences in performance, but clear user preference and acceptability for haptic and audio feedback.

CCS Concepts

•**Human-centered computing** → *Haptic devices; Auditory feedback; Gestural input; Ubiquitous and mobile devices;*

Keywords

wearable computing; interactive eye-wear; gaze tracking; smooth pursuit

AH 2016, February 25-27, 2016, Geneva, Switzerland

DOI: http://dx.doi.org/10.1145/2875194.2875209

1. INTRODUCTION

The goal of our work is to enable the use of wearable devices that augment human abilities in interacting with the environment. In future scenarios with the Internet of Things (IoT) we envision that many everyday objects such as doors, electronic appliances, lights, cars, etc. could be activated and operated remotely just by looking at them. The commands would be given with gaze only or with combinations of gaze and other modalities. Such user interfaces would involve many interaction techniques. We investigated in this paper one of the many methods that need to be evaluated to find a suitable set of interaction techniques that enables sufficient efficiency and user experience.

To be able to use the gaze as input for ubiquitous computing systems, eye movements need to be measured. Smart glasses offer a convenient platform for gaze tracking. The frames are often ideally located for placing eye tracking components. Many smart glasses also include a forward-looking camera. Video from that camera can be used to map the gaze to objects in the world and to detect nearby interactive objects.

In gaze tracking the goal is to closely follow the location and orientation of the eyes in order to compute where the gaze is aimed. Gaze tracking has been widely used in studying vision and the physical parameters of eye movements. However, because gaze and visual attention are often colocated, gaze tracking has also been useful in studying visual cognition. Modern gaze tracking systems typically utilize the pupil center and corneal reflection technique (PCCR). PCCR systems usually consist of one or more video cameras and one or more infrared light sources. The video recorded by the camera is analyzed to find the pupil and the reflection(s) of the light source(s). Based on the relative positions of these and knowledge on the location of the eye in space the gaze vector can be computed.

In addition to being used off-line for after-the-fact analysis of eye behavior, gaze trackers can also be used in real-time to provide input for computing devices. This interactive use of gaze tracking is the domain of interest in this paper. Gaze trackers in smart glasses would enable gaze based interaction in mobile use.

Gaze tracking systems in forms that resemble glasses are already available (see, e.g. the products of Pupil Labs, SMI, and Tobii [1, 2, 3]). Such systems are usually designed for recording the eye behaviour, not for interaction purposes.

One specific problem with gaze tracking devices is the need to calibrate the gaze tracker for each user. Such calibration usually involves the need to make the user look at known spots in the scene and then adjusting the gaze vector computation based on the captured eye images. Thus, the calibration in current systems requires user co-operation.

The need to calibrate trackers has not conventionally been a problem. However, in wearable consumer devices frequent calibrations are undesirable because they interrupt the user's task flow. Automatic calibration without user cooperation is one potential solution to this problem. Another solution is to develop interaction techniques that do not require exact knowledge of the gaze direction. Such interaction techniques can be built by observing only the changes of gaze orientation. Gaze gestures and smooth pursuit based techniques are two approaches utilizing relative gaze movements.

In this paper we work on smooth pursuit based techniques. In this approach the idea is that the system displays one or more moving targets to the user. The user then follows one of these targets with the gaze. The system knows both the movements of the targets and the movements of the gaze. It keeps records on the similarity between the gaze movements and the movements of each of the targets. Thus, it can notice if the gaze moves similarly to one of the targets. When this happens, the system can take an action. What action would be appropriate in each situation depends on the user interface design. In this paper we report on experiments where we displayed two targets. Following one of them caused an increase in the adjusted value and following the other caused a decrease.

2. RELATED WORK

2.1 Gaze-based interaction

Eye tracking has mostly been used as a research tool for studying visual perception, visual cognition, and eye physiology. Recently marketing studies have also been informed by gaze data from the viewers of advertisements or scenes in stores. However, the use of eye tracker data as input in interactive computing systems has also been studied. A small, but important user group of such interactive systems are people with disabilities that prevent the use of conventional user interface techniques. Spinal cord injuries high enough in the neck can lead to situations where the eyes are the most effective input method. Also degenerative neural and muscle disorders such as Amyotrophic Lateral Sclerosis (ALS) can lead to similar situations. Significant parts of earlier work in Eye Typing [16], for example, have been motivated by the needs of disabled users.

Some of the early work, however, saw gaze as a promising input for all users. For example the works by Bolt *et al.* [5, 21] and Jacob *et al.* [13] were motivated by the benefits that were seen in gaze-based interaction regardless of the user. Later possibilities in using gaze as an additional input channel to help in automatically triggering dictionary lookups [12] or additional multimedia while reading [4] have been investigated. Approaches such as these are useful because they do not require eye tracking to be operational all the time. When the gaze is being tracked, additional features will be available. However, the systems remain usable also without the gaze input.

When the gaze is used as the main input modality, there are many situations where explicit commands must be given by gaze. The difficulty in the use of eyes for explicit commands in user interfaces is that eyes must simultaneously be available for their primary purpose - seeing. Thus, eye movements intended for commanding the system must be different from the eye movements that occur when looking at things. The two conventional approaches are dwell-based and gesture-based inputs. Dwell involves looking at targets so long that it is safe for the system to conclude that this is an attempt at activating the target. Gesture-based techniques involve sequences of eye movements that are unlikely to happen by accident [8, 7, 11].

2.2 Smooth Pursuit Interaction

Studies on smooth pursuit eye movement have a long history, see e.g. [20]. However, smooth pursuit based interaction techniques are a relatively new invention. The first experimental work on smooth pursuit for interaction was published by Vidal *et al.* [22, 24, 23]. In their implementation the participants were using smooth pursuit to select between several objects that were independently (and uniquely) moving on the display. The selection was then interpreted as an interaction command. It was shown that the technique has potential and is especially useful for public displays as the system does not need individual calibration.

The robustness of smooth pursuit tracking and the suitability for public displays has later been studied by Khamis *et al.* [14], who implemented a simple game with smooth pursuit selection. Another user study was described by Esteves *et al.* [9, 10] where the participants were interacting with a smart watch using smooth pursuit based gaze interaction. The interaction was based on targets that circulated around the clock-face.

Even as smooth pursuit tracking does not require tracker calibration, the technique could be used as an implicit calibration technique. Pfeuffer *et al.* in [18] demonstrated a smooth pursuit based calibration method that can work without participant co-operation and even without participant awareness.

Some specific use cases for smooth pursuit interaction have been described by Cymek *et al.* [6] and Lutz *et al.* [15]. Cymek *et al.* demonstrated that one can enter PIN codes using smooth pursuit type gaze tracking. They were using numbers that moved on the display along different paths. The system followed the user's gaze to identify the unique path that s/he was following. Lutz *et al.* utilized a similar idea for text input. The user was following a character through a two step move (first a character group, then single character), which would uniquely identify each character.

While there have been several studies on using smooth pursuit for interaction, the studied systems have usually been tailored for certain applications only, with the only exception of the Orbits by Esteves *et al.* [10]. Following the design of Esteves *et al.* we investigated the performance of generic adjustment techniques under gaze control. We wanted to see if different feedback designs lead to different performance and user preference.

Most of the earlier work mentioned above does not actually verify that it is smooth pursuit that is being used to interact. Many smooth pursuit based algorithms, including ours, will work regardless of what type of eye movement is under way. The only thing the algorithms typically measure is whether the target and the gaze move approximately in synchrony or not. If the gaze breaks the smooth pursuit every now and then to do short saccades to catch up with the target is unessential. Even if there is no smooth pursuit, the systems will still work as long as the gaze moves on a trajectory similar enough to the target's trajectory. Because of this it would perhaps be preferable to drop the "smooth" and talk about pursuit controls. The essence of the concept is that the gaze is following a target. However, in keeping with the existing literature, we will use the term smooth pursuit. The reader should remember that while smooth pursuit is most likely what is going on most of the time, there may be exceptions and that the algorithms do not care.

3. SMOOTH PURSUIT CONTROL

The smooth pursuit based control that we were using in the experiment is used to adjust a continuous variable, like the volume of an audio source or the intensity of a light. The control needs to have two moving objects as the parameter value can be increased and decreased. The control needs not to be exact but reflect the approximate nature of the intended use. The smooth pursuit control method is well fitting to that purpose as it has some resemblance to an automatic slider that is active as long as one looks at it.

In real-world setups the moving targets could be implemented through physical constructions in the environment. A lighting fixture or a sound system could have a few LEDs that would be activated when the user looks at the lamp. LEDs in different colors would blink in different sequences. The eye tracker would detect which sequence the eyes follow and adjust the light through the IoT accordingly. Another possible implementation would not require user interface components in the IoT devices. Instead they would be shown to the user using augmented reality techniques by the smartglasses.

In this paper we do not address the user interfaces presentation. We studied the interaction mechanisms using a conventional LCD display as the graphics output device.

3.1 Design of the control

We built a simple experimental control widget to test the effect of different feedback methods. In the experiment the participant was able to control the gray level of a box by smoothly following one of the moving buttons. The control setup is shown in Figure 1.

The task was chosen as it was similar to one of the intended use scenarios (adjusting an intensity of a lamp). The buttons were moving in the opposite directions on a tight circle around the two boxes that were the focus of the action. The diameter of the circle that the buttons were moving around was about 85 mm. As the user of the control needs to keep his/her gaze on the button, the controlled gray level had to stay in the vicinity of the moving buttons all the time, so that the value change can be perceived without a need for a fixation to the box itself. As the participants were sitting approximately 50 to 70 cm away from the display, the width of the circle was around 7 to 10 degrees in visual angle.

3.2 Shape matching algorithm

The smooth pursuit control is based on the idea that as long as the gaze path is similar to the path that the followed object is taking, the object's absolute location does not matter. However, gaze path will never match the path of the followed object exactly due to tracker noise, tracking error, and the nature of eye movements. Therefore, we need to make it clear what is meant by a "similar shape", and what assumptions should be made about paths and their similarity.

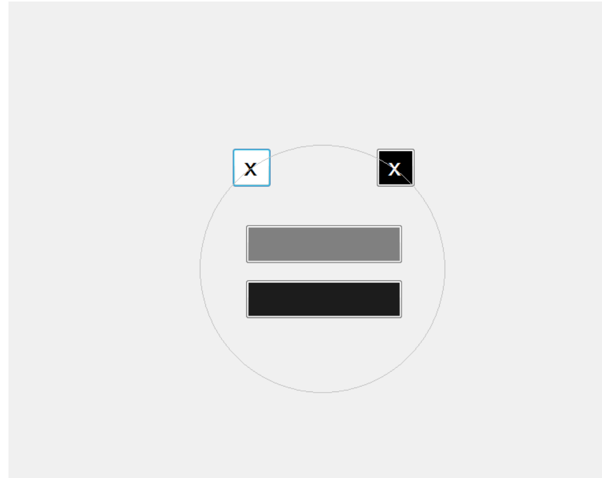

Figure 1: Experimental software UI. The upper box was showing the target gray level, and the lower box was showing the controlled gray level. The buttons with "x" mark were moving around the center by the circular path (not shown to the participants). The black button moved clockwise and the white button moved counter clockwise. Following the black button would make the gray level darker in the lower box, and following the white button would make the gray level lighter.

3.2.1 *Ideal case*

Let E be a sequence of observations $\mathbf{e}_t = (e_x, e_y)_t$ of gaze coordinates at time step t:

$$E = [\mathbf{e}_0, \dots, \mathbf{e}_t]. \tag{1}$$

Similarly let L be a set of observation sequences L^o, $o \in [1..N]$ of locations $\mathbf{l}^o_t = (l^o_x, l^o_y)_t$ of several objects o:

$$L = \{L^0, \dots, L^n\}, \tag{2}$$

$$L^o = [\mathbf{l}^o_0, \dots, \mathbf{l}^o_t]. \tag{3}$$

To check which object o is followed by the gaze during a time span $t \in [i, k]$ we would compute a distance $D_o(E, L^o)$ between the gaze observations E and object location observations L^o within a given time span, and find the object o_m with the trajectory "closest" to the given gaze trajectory, i.e. which has the smallest distance:

$$D_{o_m}(E, L^{o_m}) \le D_o(E, L^o) \forall o \tag{4}$$

Since gaze may not follow any object within the given time span, the smallest distance must be checked against some predefined threshold. If this distance is greater than the threshold, then the result of detection is void. We discuss the threshold estimation procedure later on.

A straight forward method to compute the distance implies computing the squared distances between each sample of gaze coordinates and corresponding object locations, and taking the mean of these values over the given time frames:

$$D_o(E, L^o) = \frac{\sum_{t=i}^{k}(\mathbf{e}_t - \mathbf{l}_t^o)^2}{k-i+1} = \frac{\sum_{t=i}^{k}[(e_{xt} - l_{xt}^o)^2 + (e_{yt} - l_{yt}^o)^2]}{k-i+1}. \quad (5)$$

If the gaze trajectory is exactly the same as the object location trajectory, then the sum would become zero.

3.2.2 *Offset in gaze trajectory*

In Equation 5 we assumed that we have an absolute knowledge of where the gaze is pointing to. In many cases, however, the gaze coordinates contain an unknown but constant offset $\mathbf{d}_{off}$ and we can not directly use Equation 5. Even if the shape of the gaze trajectory is an exact duplicate of an object trajectory, then the Equation 5 would equal to the squared offset $D_o(X) = \mathbf{d}_{off}{}^2$, which might become large. However, if we know that the trajectories are the same, then we can correct the observed gaze trajectory simply by subtracting the offset value. An estimate of the offset can be computed from the observed coordinates, for example, by calculating the mean value of the differences between the gaze and object trajectories:

$$\hat{\mathbf{d}}_{off}^o = \frac{1}{k-i+1}\sum_{t=i}^{k}(\mathbf{e}_t - \mathbf{l}_t^o), \quad (6)$$

for a given object o. Including the offset estimate into Equation 5 leads to:

$$D_o(E, L^o) = \frac{\sum_{t=i}^{k}(\mathbf{e}_t - \mathbf{l}_t^o - \hat{\mathbf{d}}_{off}^o)^2}{k-i+1}, \quad (7)$$

where the offset estimate is, of course, object dependent. One should notice that the removal of offset estimate also leads to a new problem, as Equation 7 will now give exactly the same distance for all objects that share the same trajectory shapes, i.e. which move in synchrony (or stay in place). Equation 7 will result in different distance values only for objects that follow different (partly, at least) trajectories. Therefore certain variability of object paths should be ensured to make the search of the "closest" trajectory calculation based on Equation 7 valid.

As the offset estimate is removed from the difference components in Equation 7, the distance measure $D_o(E, L^o)$ last derived is location independent and was used in the experiments.

3.3 Implementation details

All constants needed by the detection algorithm were defined by running a number of pilot tests during the application development and iterating the values until a suitable combination was found. A more thorough study would be needed in future to get more general guidelines for the values.

The speed of the buttons around the track was such that it took around 4.2 seconds to make one round[1]. The speed was found to be comfortable for the participants.

[1]Given the viewing distance of 50 to 70 centimeters, the speed translates to around 5 to 7 degrees per second in visual angle.

The maximum adjustment speed was 60 grey levels per second. This speed of change would be too fast for very precise adjustment but as the given task was to do an approximate control it was found to be a suitable compromise between accuracy and speed. The feedback (visual, audio or haptic) for gray level changes was activated only once for every eight gray value changes. The maximum feedback frequency was thus 7.5 Hz.

The window length in the distance calculation was 500 ms, i.e. location data of the last half a second was used in calculating the path similarities. The calculation was done 60 times per second, i.e. once every 16.7 ms.

The threshold value for the distance $D_{o_m}(E, L^{o_m})$ was set to 700, i.e. the average distance between the gaze and an object should be less than $\sqrt{700} \approx 26$ pixels to consider this object as currently tracked[2]. The threshold value was determined empirically during the development. However, the optimal value may vary depending on tracker accuracy and noise, as well as the delay (and changes in the delay) between eye movements and the time the corresponding gaze data samples are available for the software. In our simple two objects setup it was enough to find one working value and there was no need for further optimizing.

4. METHOD

4.1 Participants

We recruited 16 participants (between 18 and 33 years, median age 22.5 years; 12 males, 4 females) from the University community. 12 participants had normal vision, 4 participants were using glasses. All the participants had some experience of gaze tracking, but none had tried any smooth pursuit based interaction. All the participants had some experience of haptic feedback, mostly from using mobile devices or game controllers.

4.2 Apparatus

The experiment application software was running on a Windows 7 PC. We used C# with NET 4.5 framework to implement the experiment application, a Tobii EyeX gaze tracker to collect gaze data and a 24" screen of 1920 x 1200 pixels resolution to display stimuli. As the smooth pursuit method does not require tracker calibration, the device was calibrated once for the experiment supervisor prior to the test to start the system[3].

For haptic feedback we used two vibrotactile actuators (LVM8, Matsushita Electric Industrial Co., Japan) built into the frame of glasses similar to the construction by Rantala *et al.* [19]. The actuators were situated in the ends of the temples and gave the feedback simultaneously on both sides of the head. The stimulation signal consisted of 20 ms 150 Hz sine wave pulses to get a sensation that would resemble a short "tap".

For audio feedback we were using a 12 ms long audio clip that was played using a separate speaker behind the display.

[2]The radius of the circle that the buttons were following was 160 pixels. 26 pixels translates to around 0.6 to 0.8 degrees in visual angle.

[3]The Tobii EyeX tracker requires a calibration before it will produce gaze data. We calibrated on the experimenter instead of the participant to intentionally make the calibration inaccurate.

For visual feedback we alternated characters "x" and "+" shown inside the two moving button shapes (see Figure 1). Character switch was perceived as the visual feedback event.

4.3 Procedure

The experiment started with a short description of the experiment followed by reading and signing a consent form and by filling a demographic questionnaire. The participant was seated in front of the display at the distance of about 50 to 70 centimeters.

The experiment application was showing two boxes in the center of the display (see Figure 1). The upper box showed the target gray level, and the lower box showed the controlled gray level. The lowest grey level 0 corresponded to the black color, and the highest grey level 255 corresponded to the white color. The buttons with "x" mark were moving around the center with a constant speed (see Figure 1).

The participant was instructed to adjust the gray level that was set to the lower box at the beginning of each trial *as fast as possible* to make it *approximately* the same (gray level 127) as in the target box, and confirm the selection by pressing the space bar key on the keyboard. In real usage situations the confirmation would not be needed. Instead, the user would simply cease to adjust and look elsewhere when the desired setting would be reached. For experimental purposes, however, we needed a clear sign for the end of the task. This is why the space bar key press was required.

Although the instruction given to participants did not define strictly when to stop the adjustment, we assumed that each individual participant would interpret it similarly for each feedback condition. Our motivation for not giving strict instruction was that in everyday adjustments in the real world: sound volume and light intensity adjustments, scrolling a document or a movie, and many other similar actions usually do not have precise targets (with the obvious exceptions for minimum and maximum values). In the analysis phase we were able to check that the average error (between the target and selected gray levels) stayed the same in all conditions, which indicated that our assumption on interpreting the instruction similarly for each condition was correct.

Next, the software was configured to provide a certain feedback (or no feedback), and a set of trials was initialized. Each trial had a unique difference between the start and target gray levels. The differences varied from -100 to 100 with the step 10 (the difference equal to 0 was excluded). The order of differences was randomized.

The participant first completed a practice block of five random trials, and then the full experimental block of 20 trials. It took between 3 and 4 minutes to complete the experimental block. Next, the participant was asked to evaluate how well the control worked with the given feedback. We used a 7-point scale from -3 "very poorly" to 3 "very well".

The experiment consisted of four rounds, one per feedback condition (see Table 1). The order of the conditions was counterbalanced between the participants using the Latin square method.

After all four feedback conditions were completed, the participant was asked to rank the conditions by giving a ranking number 1 to the condition that s/he considered most viable, 2 to the second most viable, and so on. In the very end we gave the participants a chance to give free form comments.

The experiment had a within-subject design with every participant performing all conditions. The dependent variables that we measured were the trial completion time, the trial gray level error and the ratings that the participants gave in the subjective evaluations. The trial completion time was measured from the moment when the new trial gray level was shown to the participant to the moment when the participant pressed the space bar. The trial gray level error was measured at the moment of space bar press, as well. The experiment duration was between 15 and 25 minutes per participant.

Table 1: The experiment conditions

Condition	Description
NoFeedback	No feedback
Audio	Short click sound
Haptics	Short tap on the temples of glasses
Visual	Alternating characters on the followed object

5. RESULTS

In testing for statistically significant differences, we used a non-parametric permutation test (see e.g. [17]). In the test an observed value of a measurement is compared against a distribution of measurements produced by resampling a large number of sample permutations assuming no difference between the sample sets (null hypothesis). The relevant p-value is given by the proportion of the distribution values that is more extreme or equal than the observed value. The relevant measure and the resampling method is test dependent and will be described together with the results.

5.1 Objective measures

On analysing the trial completion times we need to consider the conflicting target of getting good accuracy in gray level matching. One participant may spend more time trying to get more accurate matching, while another would target for faster action and care less about the accuracy. As long as the accuracy criteria are the same for each participant separately for all conditions we can assume the results are valid. It might also be possible that the participant would react to the conditions and the speed requirement so that s/he would change the accuracy requirement per condition. That would lead to biased trial completion times. In a permutation test we measured the difference between median values of gray level errors per condition for a participant, and then assuming no difference between the conditions, we pooled the errors from two compared conditions and resampled from that generating 10'000 permutations. We found only one case of statistically significant differences between conditions (at the level of $p < 0.01$). This frequency of statistically significant differences matches well with what would be expected to happen by chance only given that there were 96 pairwise comparisons to complete ($16\ participants \times 6\ comparisons = 96\ comparisons$). Therefore, we assume that all participants were using the same accuracy criteria (per participant) on every feedback condition.

The mean values of the trial completion times are shown

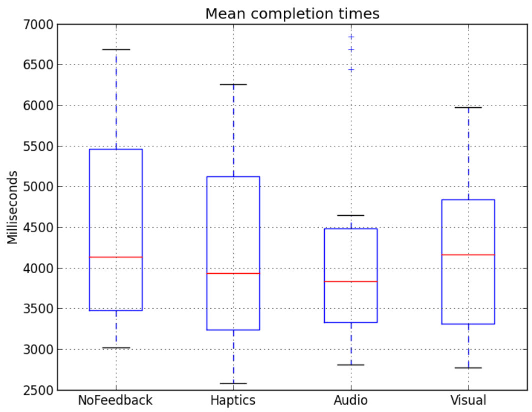

Figure 2: The mean values of trial completion times. Each participant was doing 20 trials for each condition.

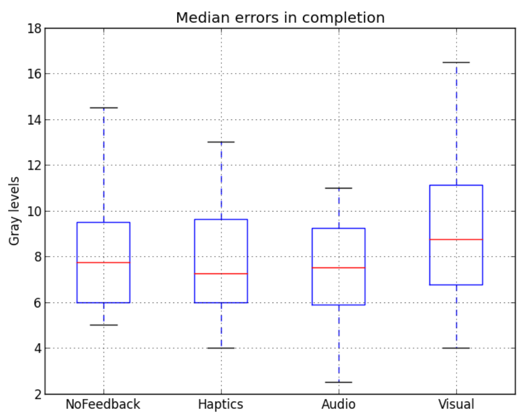

Figure 3: The median values of the gray level matching errors in the end of the trials.

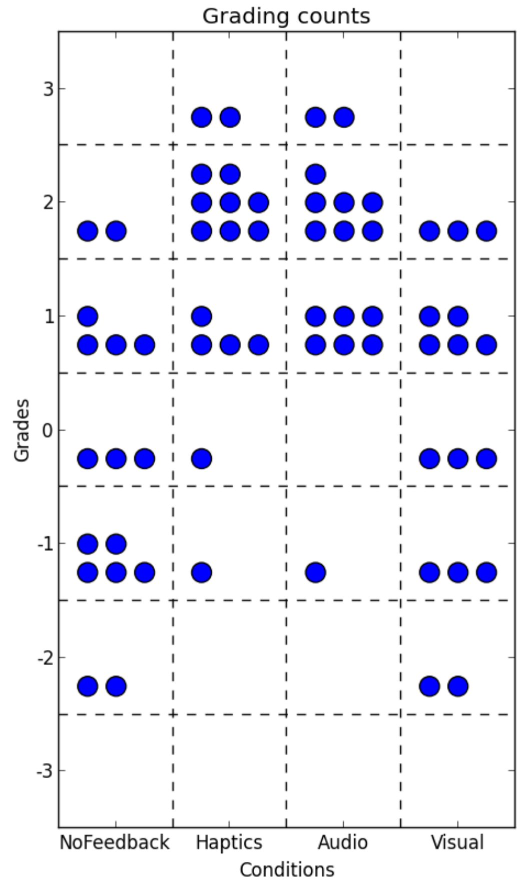

Figure 4: The participants were asked to evaluate how well the control worked with each condition giving grades between -3 "very poorly" and 3 "very well".

on Figure 2[4]. There were small differences between the trial completion times on different conditions, but no statistically significant differences. As a measure in the significance test we used the sum of differences between (each participant's) paired samples, and computed 10'000 resamples of the differences with random reversals of the signs.

The median values of the errors in gray levels at the end of trials are shown on Figure 3. The differences between the median values were small and there were no statistically significant differences. We used the same significance test method as above.

5.2 Subjective measures

The individual grades that the participant were giving to different conditions are shown in Figure 4. There are obvious differences on the overall distribution of values. The grades given to *Haptics* and *Audio* were generally positive with median grade of 2. The grades given to *NoFeedback* and *Visual* varied more between participants. The median value for *NoFeedback* was 0 and for *Visual* it was 0.5. Conditions *Haptics* and *Audio* were thus consistently liked, while there was more uncertainty of the usefulness of the conditions *NoFeedback* and *Visual*. Some participants, anyhow, gave higher grades to *NoFeedback* and *Visual*.

In the significance test we computed the signs of the differences of each pair of grades given to two conditions by the participants, and the measure was the sum of the signs, adding 1 for positive sign, -1 for negative sign and 0 for even grades. Assuming a null hypothesis of no difference we computed 10,000 resamples with random reversals of the difference signs between the conditions. The permutation test shows that there are statistically significant differences between the condition *NoFeedback* and both conditions *Haptics* ($p < 0.004$) and *Audio* ($p < 0.003$), and separately between the condition *Visual* and condition *Haptics* ($p < 0.002$).

The distribution of the ranking positions that the participants were giving to different conditions are collected into

[4] Even as separate trials naturally require different completion times as target difference was varied, the mean value is linearly dependent on the sum of the completion times of all 20 trials.

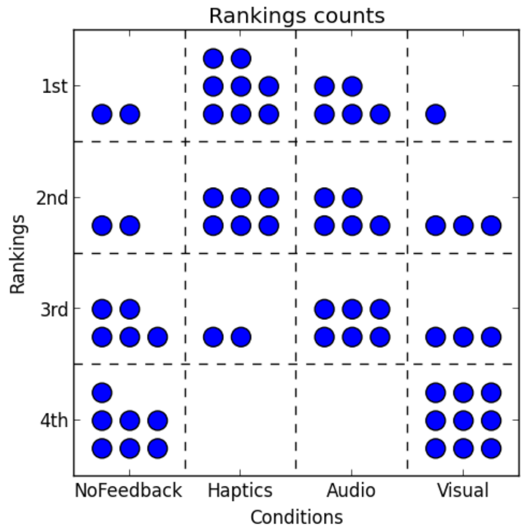

Figure 5: The participants were asked to order the conditions, giving a ranking number 1 to the most viable condition, 2 to the second most viable condition, etc.

Figure 5. The two conditions that were graded higher were also given higher rankings. The condition *Haptics* got 50 % and the condition *Audio* got 37 % of the highest rankings (1^{st}). On the other hand those two conditions got none of the lowest rankings (4^{th}), which were then divided between the other two conditions. The condition *NoFeedback* got 44 % and the condition *Visual* got 56 % of the lowest rankings.

To check for significance of the differences in rankings we used the same permutation test as above. For the ranks there are no equal values, so every ranking pair leads to either positive or negative sign. The test showed that there is a statistically significant difference between the *Haptics* and *Visual* feedback ($p < 0.001$).

Only five participants gave free-form comments in the end of the experiment. Some of the participants commented that the feedback (in general) gives the user more confidence that the tracker is recognizing their action. One comment was about the interfering case of *Visual* condition: *Visual is disturbing, too much happening for one sense*. Even as the visual feedback is giving basically the same information as the other modalities it can be confusing as the user observes simultaneously the visual feedback, the motion of the targets and the gray level change.

6. DISCUSSION

Earlier work on smooth pursuit control has involved single event type interaction, like the character input in [6, 15]. An exception has been by Esteves *et al.* [10] who describe an idea that a control would be adjusted continuously as long as the pursuit target is followed. Our results add on that and show that smooth pursuit control with suitable feedback can be utilized on continuous values.

The results show that the participants were able to do the control tasks using the smooth pursuit gaze tracking. The results also indicate that the feedback method has little effect in the objective performance measures of the control. The trial completion times varied quite a lot between participants but only a little on an aggregate level. The same is true for the trial completion gray level error values. On the aggregate level we notice only small differences between conditions on the error values.

The subjective evaluations, on the other hand, showed clear differences between the feedbacks. The participants preferred *Haptics* and *Audio* feedback more than the others. It was also clear that most of the participants preferred to have some kind of feedback, as only 2 out of 16 participants ranked *NoFeedback* as 1^{st} in the ranking question. That would indicate that when we implement smooth pursuit based interaction devices, it would be advisable to activate feedback when the system detects the smooth pursuit. If the implementation is based on smart glasses then either *Audio* or *Haptics* would be an easy and effective choice. If we expect that the system will be potentially used in noisy environments then *Haptics* would probably suffer less interference.

Visual feedback had a disadvantage in the task, as the participants had to simultaneously observe the gray level change and were seeing the feedback changes in the followed button. That may have been one specific reason that the *Visual* condition was given poorer grades. Similar disadvantage might arise if the controlled parameter was audio based (e.g. audio volume) and the feedback was given by sound.

7. CONCLUSIONS AND FUTURE WORK

The control system based on smooth pursuit worked well and was accepted by the participants. There were no statistically significant differences between the feedback conditions on objective measures, the completion times and the accuracy of setting the gray levels. There were, however, statistically significant differences between the feedback conditions on subjective measures, as the participants graded conditions *Haptics* and *Audio* higher than conditions *NoFeedback* and *Visual*.

The results give a good basis to continue on more experiments. Before a complete pursuit-based user interface can be built we need more work on other interaction techniques for various other tasks. Even the simplest cases mentioned in this paper, light intensity and volume controls, require a further ability to turn the device on and off.

Also, within the domain of continuous adjustment, more studies are needed, especially on the relation of the feedback methods to the controlled parameters. Also, we certainly need more studies on more realistic environmental settings, to demonstrate that the system is robust enough for general use.

8. ACKNOWLEDGMENTS

The work was supported by the Academy of Finland, projects HAGI (decisions numbers 260026 and 260179) and MIPI (decision 266285).

9. REFERENCES

[1] Pupil labs. http://pupil-labs.com/. Accessed: 2015-11-27.

[2] SMI eye tracking glasses 2. http://www.eyetracking-glasses.com/. Accessed: 2015-11-13.

[3] Tobii glasses 2. http://www.tobiipro.com/product-listing/tobii-pro-glasses-2/. Accessed: 2015-11-13.

[4] R. Biedert, G. Buscher, S. Schwarz, J. Hees, and A. Dengel. Text 2.0. In *Extended abstracts of CHI 2010*, pages 4003–4008, New York, NY, USA, 2010. ACM Press.

[5] R. A. Bolt. Gaze-orchestrated dynamic windows. In *Proc. Siggraph 1982*, pages 109–119, New York, NY, USA, 1982. ACM Press.

[6] D. H. Cymek, A. C. Venjakob, S. Ruff, O. H.-M. Lutz, S. Hofmann, and M. Roetting. Entering pin codes by smooth pursuit eye movements. *Journal of Eye Movement Research*, 7(4):1–11, 2014.

[7] H. Drewes, A. De Luca, and A. Schmidt. Eye-gaze interaction for mobile phones. In *Proceedings of the 4th International Conference on Mobile Technology, Applications, and Systems and the 1st International Symposium on Computer Human Interaction in Mobile Technology*, Mobility '07, pages 364–371, New York, NY, USA, 2007. ACM.

[8] H. Drewes and A. Schmidt. Interacting with the computer using gaze gestures. In *Proc. INTERACT 2007*, pages 475–488, New York, NY, 2007. Springer.

[9] A. Esteves, E. Velloso, A. Bulling, and H. Gellersen. Orbits: Enabling gaze interaction in smart watches using moving targets. In *Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers*, UbiComp/ISWC'15 Adjunct, pages 419–422, New York, NY, USA, 2015. ACM.

[10] A. Esteves, E. Velloso, A. Bulling, and H. Gellersen. Orbits: Gaze interaction for smart watches using smooth pursuit eye movements. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, UIST '15, pages 457–466, New York, NY, USA, 2015. ACM.

[11] A. Hyrskykari, H. Istance, and S. Vickers. Gaze gestures or dwell-based interactionfs. In *Proc. ETRA'12*, pages 229–232. ACM Press, 2012.

[12] A. Hyrskykari, P. Majaranta, A. Aaltonen, and K.-J. Räihä. Design issues of idict: a gaze-assisted translation aid. In *Proc. ETRA 2000*, pages 9–14, New York, NY, USA, 2000. ACM Press.

[13] R. J. K. Jacob. The use of eye movements in human-computer interaction techniques: What you look at is what you get. *ACM Trans. Inf. Syst.*, 9(2):152–169, Apr. 1991.

[14] M. Khamis, F. Alt, and A. Bulling. A field study on spontaneous gaze-based interaction with a public display using pursuits. In *Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers*, UbiComp/ISWC'15 Adjunct, pages 863–872, New York, NY, USA, 2015. ACM.

[15] O. H.-M. Lutz, A. C. Venjakob, and S. Ruff. Entering pin codes by smooth pursuit eye movements. *Journal of Eye Movement Research*, 8(1):1–11, 2015.

[16] P. Majaranta and K.-J. Räihä. Twenty years of eye typing: Systems and design issues. In *Proceedings of the 2002 Symposium on Eye Tracking Research & Applications*, ETRA '02, pages 15–22, New York, NY, USA, 2002. ACM.

[17] T. E. Nichols and A. P. Holmes. Nonparametric permutation tests for functional neuroimaging: A primer with examples. *Human Brain Mapping*, 15(1):1–25, 2002.

[18] K. Pfeuffer, M. Vidal, J. Turner, A. Bulling, and H. Gellersen. Pursuit calibration: Making gaze calibration less tedious and more flexible. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13, pages 261–270, New York, NY, USA, 2013. ACM.

[19] J. Rantala, J. Kangas, D. Akkil, P. Isokoski, and R. Raisamo. Glasses with haptic feedback of gaze gestures. In *Proceedings of the Extended Abstracts of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, CHI EA '14, pages 1597–1602, New York, NY, USA, 2014. ACM.

[20] D. A. Robinson. The mechanics of human smooth pursuit eye movement. *The Journal of Physiology*, 180(3):569–591, 1965.

[21] I. Starker and R. A. Bolt. Gaze-responsive self-disclosing display. In *Proc. CHI 1990*, pages 3–10, New York, NY, USA, 1990. ACM Press.

[22] M. Vidal, A. Bulling, and H. Gellersen. Pursuits: Spontaneous interaction with displays based on smooth pursuit eye movement and moving targets. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '13, pages 439–448, New York, NY, USA, 2013. ACM.

[23] M. Vidal, A. Bulling, and H. Gellersen. Pursuits: Spontaneous eye-based interaction for dynamic interfaces. *GetMobile: Mobile Comp. and Comm.*, 18(4):8–10, Jan. 2015.

[24] M. Vidal, K. Pfeuffer, A. Bulling, and H. W. Gellersen. Pursuits: Eye-based interaction with moving targets. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '13, pages 3147–3150, New York, NY, USA, 2013. ACM.