Orbits: Gaze Interaction for Smart Watches using Smooth Pursuit Eye Movements

Augusto Esteves[1], **Eduardo Velloso**[2], **Andreas Bulling**[3], **Hans Gellersen**[4]
[1,2,4]Lancaster University, Lancaster, United Kingdom
[3]Max Planck Institute for Informatics, Saarbrücken, Germany
{a.esteves, e.velloso}@lancaster.ac.uk, bulling@mpi-inf.mpg.de, hwg@comp.lancs.ac.uk

ABSTRACT

We introduce Orbits, a novel gaze interaction technique that enables hands-free input on smart watches. The technique relies on moving controls to leverage the smooth pursuit movements of the eyes and detect whether and at which control the user is looking at. In Orbits, controls include targets that move in a circular trajectory in the face of the watch, and can be selected by following the desired one for a small amount of time. We conducted two user studies to assess the technique's recognition and robustness, which demonstrated how Orbits is robust against false positives triggered by natural eye movements and how it presents a hands-free, high accuracy way of interacting with smart watches using off-the-shelf devices. Finally, we developed three example interfaces built with Orbits: a music player, a notifications face plate and a missed call menu. Despite relying on moving controls – very unusual in current HCI interfaces – these were generally well received by participants in a third and final study.

Author Keywords

Eye tracking; smart watches; small devices; small displays; pursuits; gaze interaction; gaze input; wearable computing.

ACM Classification Keywords

H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION

In 1904, Brazilian aviation pioneer Alberto Santos-Dumont complained to his friend Louis Cartier, the French jeweller, about the difficulties in reading his pocket watch while flying with both hands on the plane's controls [12]. Soon after, Cartier emerged with a prototype of a wrist-mounted watch that enabled the aviator to time his flights without taking his hands off the controls. Wristwatches quickly gained widespread popularity, and since then, have become a staple of fashion and functionality. Fast forward over 100 years and wristwatches are once again in the spotlight, in the form of smart watches that enable digital information to be consumed at a glance. In this paper, we propose to complement this field by provision of *input at a glance* and, thereby, retain hands-free use as a key affordance of the original wristwatch.

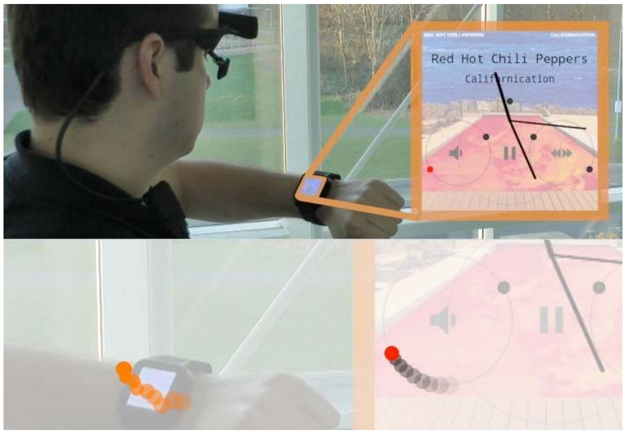

Figure 1. Top: a user raises the volume of his smart watch music player using Orbits gaze input controls. The UI shows the volume, pause/play and previous/next controls with orbiting targets for gaze selection. Bottom: how Orbits enables gaze input on smart watches. The technique can robustly detect which of the controls is actively being followed by correlating each Orbits' target with the user's gaze.

We introduce Orbits, a novel technique that enables gaze-only input in a design that accounts for both the limited display space of smart watches and the spontaneous nature of glancing at a watch. Orbits are graphical controls that display one or multiple targets moving on a circular orbit around the control. Users provide input to a control by following one of its orbiting targets briefly with their eyes, leading to trigger functionality associated with the target. Figure 1 illustrates Orbits in a smart watch interface for a music player. In this example the volume control displays a target orbiting clockwise to increase the volume and a target moving anticlockwise to decrease it.

Gaze input with Orbits leverages *smooth pursuits*, a distinctive form of eye movement that occurs when we follow a moving stimulus with our gaze [1]. Our technique builds on three principles: (1) smooth pursuits exhibit a characteristic behaviour that facilitates robust distinction

UIST '15, November 08 - 11, 2015, Charlotte, NC, USA.

ACM 978-1-4503-3779-3/15/11…$15.00.
DOI: http://dx.doi.org/10.1145/2807442.2807499

from regular eye movements, i.e. fixations on static points of regard and saccades as fast transitions between fixations; (2) the human eye cannot generate smooth pursuits without external stimuli, making it hard for a system to interpret input when none was intentionally provided (a *false positive*); (3) the eyes' trajectory during a smooth pursuit closely matches the relative trajectory of a target permitting disambiguation when multiple moving targets are displayed to the user (a principle first demonstrated by Vidal et al. [30]). Contrasting conventional use of gaze for input, our technique presents two significant advantages with respect to eye tracking. First, because our approach relies on the relative movement of the eyes, no calibration between the eye tracker and the display is necessary. Second, because we identify a target by its movement pattern and not its position, our approach works independently of the target's size and is robust against inherent inaccuracies in eye tracking and the natural jittery movement of the eyes.

We provide the following contributions through this work:

- First, the design and implementation of Orbits as a first technique that enables and demonstrates hands-free input 'at a glance' on smart watches.
- Second, an experimental evaluation of Orbits recognition performance depending on the duration of pursuits and the correlation threshold for matching eye movement and target movement.
- Third, an experimental evaluation of the robustness of Orbits for target selection using different numbers of targets, different target speeds and different orbit sizes: a question left open by prior work of Vidal *et al.* [30].
- Fourth, the description of three smart watch applications implemented to demonstrate Orbits and to solicit qualitative feedback from users.

RELATED WORK

One of the main challenges when designing interfaces for smart watches is the small size of the display. This has been an issue with most touch-based mobile devices, including smart phones and tablet computers. Various interaction techniques address this challenge: techniques that support multi-touch gestures (e.g., pinching [8]), clip-on physical controls [33] or even touch-input on the back of the device [2]. While these and other solutions have proven very useful in phones and tablets, they do not scale down appropriately to the much smaller displays of smart watches [27]. The wearable form factor of these devices not only leaves little room for spatial gestures, but also makes clip-on controls infeasible and interactions on the back of the device impossible to perform.

As a result, most new interaction techniques specific to small watches tend to focus on decoupling the input and output space of these devices. Xiao *et al.* used the face of the watch as a mechanical interface that users can physically twist, tilt and pan [32]; Blaskó and Feiner and Oakley and Lee added touch sensors to the face [3] and edge of a smart watch [22]; Perrault et al. added touch sensors to the strap of a watch for tap and stroke input [23]; Kratz and Rohs used proximity sensors on the face of a watch to detect fast but coarse gestures [17]; and Harrison and Hudson used a magnetometer to support pointing input through a magnet strapped to the user's finger [13]. All of these approaches require some form of touch input or hand control, and thus constrain both hands during interaction. In contrast, we suggest a novel approach to smart watch interaction that relies on gaze instead of touch, thereby enabling smart watches to stay true to Cartier's original motivation of hands-free interaction for wristwatches.

Gaze Input

Gaze is long established as alternative to manual input for applications and users that require hands-free controls for interaction [14, 19, 21]. The prevailing gaze input technique is gaze pointing, which involves the user fixating on statically presented targets for selection, using either dwell time or a complementary input method (e.g., mouse click) to avoid unintended activations (the 'Midas touch') [15]. Gaze pointing can be highly efficient [28], but depends on accurate estimation of the gaze direction, requires a calibration to map gaze direction to the target display and has inherent limitations for selection of small targets due to the natural jitter of the eyes during fixation [34]. In Orbits, we avoid these limitations by using smooth pursuits instead of fixations for input.

Smooth pursuits are relatively slow and consistent eye movements that occur only when the eyes follow a moving object [1]. Initially, the eye is accelerated to catch up with the stimulus but converges with the stimulus motion in less than 300ms [5]. Vidal *et al.* were arguably the first to propose smooth pursuits as input method for selection of objects displayed as moving targets. Their technique uses Pearson's product moment correlation method to match the tracked eye movement with the trajectories of displayed targets [30], an algorithm that we also adopt for target matching in Orbits. Cymek *et al.* used smooth pursuit for PIN entry in an interface design with moving tiles [7] and Lutz *et al.* presented a similar design for text entry [18]. Pfeuffer *et al.* used smooth pursuits for calibration with a moving target, demonstrating how this enables a smarter calibration process where the system knows when the user is attending the target and when not [24]. In Orbits, we also specifically exploit how the user's eyes will only produce smooth pursuit movement when the user is actually following a target – in our case to avoid unintended input.

Both our own work, as well as prior work based on smooth pursuits, involves the display of moving targets and matching with motion produced by the user. The principle of selecting targets by matching motion has previously been explored in Fekete *et al.*'s motion-pointing technique [11]. As with motion-pointing, Orbits controls display animated targets, however the user is not required to explicitly mimic

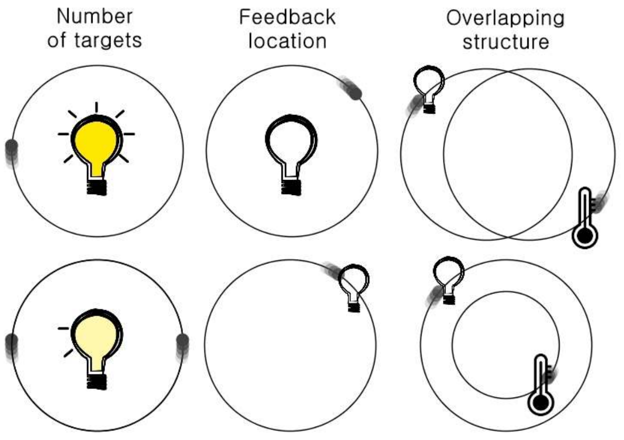

Figure 2. Different Orbits concepts for light and temperature switches. Left: each orbit can have one (top) or multiple targets (bottom). Middle: feedback can be presented at the centre (top) or on an Orbits' target (bottom). Right: Orbits controls can be overlaid (top) or nested (bottom).

the motion of the target but is able to perform selection by natural smooth pursuit of an intended target.

A body of work has relied on other forms of relative input from the eyes for interaction. This includes gaze gestures that require users to perform saccadic movement in a pre-learned single- or multi-stroke pattern [9, 20], however such gestures are not ideal as they require users to perform unnatural eye movements. Zhang *et al.* used 'sideways glances' for discrete input [35] and continuous scrolling [36], but these were specifically designed for interaction with larger displays. Recent work has begun to consider gaze for smaller-scale devices, e.g., comparing dwell time and gaze gestures for object selection on smart phones [10] and proposing gaze gestures with haptic feedback on eyeglasses [26]. We follow this trend by demonstrating for the first time gaze-only input on smart watches.

ORBITS

Classic mechanical watches already contain several moving targets, in the form of the watch's hands themselves or the small dials and cogs in timers and chronometers. Inspired by this design, we named our gaze interaction technique *Orbits,* as it relies on interface controls that contain targets that move continuously in circular trajectories. Each target performs a distinct function and can be activated by following it with the eyes for a certain amount of time. They can be used for both discrete control (by treating each Orbits activation as a command, see Figure 2 – top left) and continuous control (by using the time following the target to modify the value of the controlled parameter, see Figure 2 – bottom left). Each Orbits widget comprises a trajectory, one or multiple target, and feedback elements. In this section we discuss the design decisions for such interface.

Orbits Design: We use circular Orbits, as this fits well with the shape of most watches and the shape of the dials on watches' faces. Multiple Orbits can be differentiated in several ways. First, we can vary the *phase* of their targets. This can be achieved by adding an offset to the initial position of each target (e.g., in Figure 2 – bottom left – there is a 180° offset between them). Second, we can vary their *angular speeds*. Whereas different Orbits can have different speeds, there is a certain range of speeds that is optimal for smooth pursuits, which we investigate in our studies. Third, we can vary the *direction* of movement, having some targets move clockwise and others counter-clockwise (see Figure 2 – bottom left). This can also help convey information about the corresponding functionality of the Orbits control (e.g. one direction increases the controlled parameter and the other decreases it). Fourth, we can vary the Orbits *size,* i.e. the diameter of the trajectory. Even though having Orbits with different sizes can help users visually follow a target, the phase, direction or the speed must also be modified in order to make the selection possible. This is because the correlation algorithm normalises the absolute values, making different sized targets equivalent. In our two studies we investigate how these parameters, as well as the number Orbits on-screen affect the performance and usability of our technique.

Feedback Design: In gaze interaction the eyes perform the dual function of capturing visual information and providing input to the system. This requires careful consideration on how to provide feedback to the user. We propose two ways of doing so. The first is to use abstract targets and provide graphical feedback at the centre of the Orbits control (Figure 2 – middle top). This has the advantage of making the screen less cluttered and is appropriate for situations where different targets refer to the same object (Figure 2 – bottom left). Moreover, it offers a neutral object to look at when the user does not want to acquire any target. However, the functionality of each control becomes less clear as they all look the same. The second is to have the feedback on the target itself (Figure 2 – middle bottom). This requires more screen space but makes the functionality of each target explicit. We will describe prototype applications that use both feedback designs.

Algorithm Design: To recognise which target the user is looking at (if any), and as proposed by Vidal *et al.* [30], we compute the Pearson's correlation between the corresponding x- and y-coordinates of the gaze point and each target's positions within a certain time window – storing the smallest of the two. If this minimum correlation exceeds a certain threshold we activate the Orbits control. Therefore, in terms of algorithm design, we must choose a window size for the correlation calculation (i.e. how long the user must follow a target to activate it) and a correlation threshold to trigger the target activation. Increasing the window size improves the recognition performance, but decreases the responsiveness of the system due to the added lag. By increasing the correlation threshold, we can discard more false positives, at the risk of discarding more true positives in the process. In our first study we explore these trade-offs so as to decide on both these parameters.

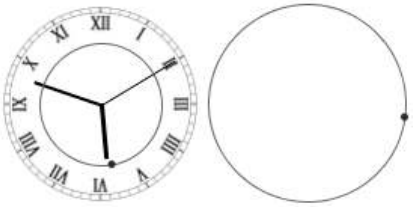

Figure 3. Two of the trials in the first study. Left: an Active Target Avoidance (ATA) trial, displaying a medium sized orbit. Right: an Active Target Pursuit (ATP) trial, displaying a large sized orbit. In an ATA trial participants try to read the time without looking at the orbit displayed. In the ATP trial participants actively follow the orbit displayed.

1ST STUDY: EVALUATION OF ORBIT RECOGNITION

In our first study we wanted to validate the concept of using Orbits in a small display under a controlled setup. The study had three goals. First, we tested the effects of the window size and correlation threshold on the true and false positive rates in order to find suitable algorithm parameters. Second, given these parameters, we tested the effects of the targets' speed and trajectory size on the true and false positive rates in order to better understand how different Orbits designs affect the recognition performance. Third, we tested the algorithm against users' natural eye movements in situations where they were reading the time and text, watching a video and playing a video game, in order to test how robust our technique is to false positives.

Participants

We recruited twelve participants (8M/4F), aged between 20 and 36 years (M = 27.8). With the exception of one, all were full-time undergraduate and graduate students at the local institution. Participants rated their experience with eye tracking at 4.36 (*SD* = 2.41) on a 1 to 7 scale (no experience to very experienced) and eight wore vision aids during the study (five wore glasses and three contact lenses).

Experimental Setup and Design

We conducted the experiment in a quiet laboratory space, with participants sitting comfortably at a desk at a distance of 63cm to a 17" laptop (1920×1080 resolution screen). We recorded participants' gaze with a 30Hz Tobii EyeX eye tracker mounted below the screen (manufacturer-reported average gaze estimation error of 0.4° of visual angle). The experiment was designed to capture controlled, calibrated gaze data (x and y) in three scenarios:

- **Active Target Pursuit (ATP)**, where participants are actively trying to follow a target in order to activate it.
- **Active Target Avoidance (ATA)**, where participants read the time while Orbits controls are being displayed.
- **No Target (NT)**, where no controls are displayed and the users' natural eye behaviour performing other tasks is recorded.

We implemented five tasks in *Processing*[1] to collect data in these three scenarios. In the ATP scenario, users followed a target for six seconds, nine times. Each of these targets varied in its diameter (Large: 2.6cm/2.36° of visual angle, Medium: 1.6cm/1.46° and Small: 0.6cm/0.55°) and angular speed (Slow: 60°/sec, Medium: 120°/sec and Fast: 240°/sec). In the ATA scenario we instructed users to read and write down a random time presented on a 2.6cm analogue watch, nine times. Each time, one of the nine Orbits from the ATP scenario was visible on the watch's face – see Figure 3. Finally, we collected data for the NT scenario with users performing three tasks: reading text, watching a video and playing a game. In the reading task, participants read a 900-word news article[2]. In the video task, participants watched a 3.5 minutes TED talk[3]. In the game task, participants played a 2D platformer game[4] for approximately four minutes. We selected these three tasks as they provide a wide range of different gaze input, and thus are suitable to represent different everyday tasks (e.g., reading a billboard, attending a class).

Procedure

Upon arrival, participants signed a consent form and completed a demographics questionnaire. We calibrated the eye tracker with the manufacturer's default 9-point procedure. Participants were then given a short introduction to each of the five tasks. At the beginning of each task we displayed written instructions on the screen, and the task started by a space bar press. In the ATP and ATA tasks users were presented with nine trials (diameter × angular speed), each ending automatically after six seconds and starting only after a key press (ensuring participants could rest in-between trials). In the ATA task, participants completed each trial once they correctly checked the time displayed, tapped on the screen and wrote the time down in a sheet of paper. Finally, the NT tasks ended with a key press after participants finished reading the text; automatically after the video finished playing; and by instruction of the experimenter, who kept track of how long participants were playing for.

The ATP task was repeated three times throughout the study (9 trials × 3 = 27 trials) and the ATA task was repeated twice (9 trials × 2 = 18 trials). This ensured that a comparable number of data points were captured across the three scenarios (ATP, ATA and NT) and to observe any practice effects in the ATA task. All participants performed these tasks in the same order: ATA (Practice), ATP, NT (Text), ATP, NT (Video), ATP, NT (Game), ATP and ATA. All participants performed each of the nine trials in the ATP and ATA tasks in the same order. Each session lasted up to 30 minutes.

---

[1] www.processing.org

[2] www.theverge.com/2014/11/25/7276157/nanogenmo-robot-author-novel

[3] www.ted.com/talks/matt_cutts_try_something_new_for_30_days

[4] riskofraingame.com

Analysis and Results

In the ATP and ATA scenarios we computed the correlation between gaze and target coordinates in overlapping windows of 15, 20, 30 and 40 samples. We considered each trial (combination of task, speed, size and participant) to be activated if there was at least one window in which the correlation exceeded the threshold. To compute the Receiver Operating Characteristic curve in Figure 4, we computed the true and false positive rates using correlation thresholds ranging from 0 to 1 in 0.1 increments.

We found a window size of 30 samples (1 second) to represent a good balance for the trade-off between the true positive rate, the false positive rate and the lag introduced by larger window sizes. Using these parameters, we tested the effects of the size and speed on the true positive rates (in the ATP tasks) and false positive rates (in the ATA and NT tasks) with a between-subjects factorial ANOVA. We report Greenhouse-Geisser-corrected degrees of freedom in cases where Mauchly's test showed a violation of sphericity and Bonferroni-corrected post-hoc tests.

In the ATP scenario we tested for effects on the true positive rates. We found a large significant effect of the trajectory size ($F_{2,22} = 56.0$, $p < .001$, $\eta^2 = .51$) and a medium effect of the speed ($F_{2,22} = 20.1$, $p < .001$, $\eta^2 = .09$), but no significant effect of the interaction between the two ($F_{2.08,22.9} = 2.15$, $p = .14$). The differences were significant for all levels of Size at $p < .05$, but not between Fast and Medium speeds. The mean true positive rate with large Orbits (.90) was 25% higher than with medium-sized Orbits (.72) and almost four times higher than the small Orbits (.23). The mean true-positive rate in the Medium and Fast speeds (.68) was 38% higher than in the Slow condition (.49). The best true-positive rate was with Large Orbits with Medium speed (.96).

In the ATA scenario, we tested for effects on the false positive rates. We found a medium significant effect of Size ($F_{2,22} = 4.01$, $p = .03$, $\eta^2 = .07$), but no significant effect of Speed ($F_{2,22} = .51$, $p = .61$) nor interaction between the two ($F_{4,44} = 1.57$, $p = .20$). The effect of Size was significant only between the Medium (.06) and Small (.21) sizes, but moderately ($p = .042$). The mean false-positive rate across all conditions was 0.12 and the best condition was Medium-sized Orbits with Medium speeds: zero false-positives.

In the NT scenario, because no targets were displayed, we computed the correlation between the gaze data and Orbits simulated with the same characteristics as in the ATP and ATA tasks (with random offsets). As each trial of these conditions were substantially longer, we computed the false positive rate as the number of 1-second windows in which the target would be activated divided by the total number of windows. We tested for effects of Size, Speed and Task. We found a large significant effect of Speed ($F_{1.34,14.75} = 33.44$, $p < .001$, $\eta^2 = .20$) with significant differences between all levels. The false positive rate for Slow Orbits (.0098) was 37% higher than for the Medium speed (.0072) and 3.4 times higher than for Fast (.0029) Orbits. We also found a moderate significant effect of Task ($F_{2,22} = 9.90$, $p < .001$, $\eta^2 = .09$) due to a significant higher false positive rate in the Video task (.0091), but no significant difference between the Text (.0047) and the Game (.0061) tasks ($p = .16$). We found no significant effect of Size ($F_{2,22} = 0.74$, $p = 0.48$, $\eta^2 = .002$) and its interactions.

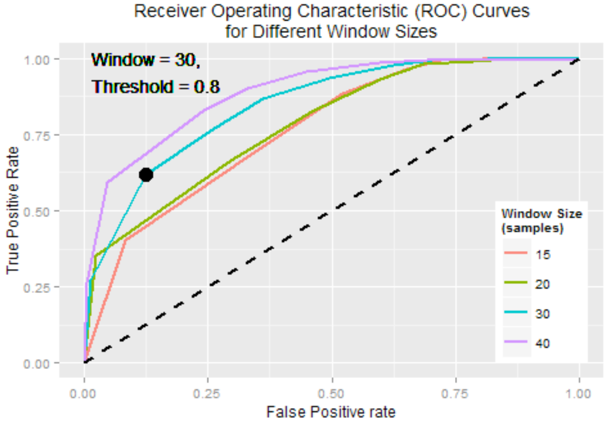

Figure 4. ROC curves showing the detection performances for different window sizes. The following studies use a window size of 30 samples (1s) and a correlation threshold of 0.8.

Discussion

Large Orbits with Medium speed achieved the best recognition rates. In this design, we activated the correct Orbits in 96% of the trials, and only incorrectly activated them in 12.5% of cases where the user was reading the time and in 0.73% of the windows in which the user was performing tasks unrelated to the watch. The overall recognition performance can be improved with subsequent filtering or using larger window sizes (at the cost of activation lag), but these results demonstrate that Pearson's correlation alone already achieves high performance.

We found that the larger the Orbits size the higher is the true positive rate, with little impact in the false positive rate. This is because the correlation algorithm is scale-invariant, as it normalises both the eye tracking data and target coordinates. When the user is actively pursuing a target the tracking error strongly affects the recorded gaze path – the size of our smallest target trajectory (0.55°) was almost as small as the nominal error of the tracker (0.4°). However, when the user is engaged in different task the trajectory size makes no difference. In our setup, however, the distance between the user and the display (63cm) was much larger than the distance between the eyes and the watch, which means that, in terms of visual angle, the trajectory of the targets used were much smaller than they would appear in an actual watch usage scenario.

Whereas we found that the bigger the trajectory the better the performance, the same did not apply for speed: a sweet spot could be found. This is because smooth pursuits

Figure 5. The experimental setup for the second study.

operate in a certain speed range: if it is too slow it becomes a fixation; if it is too fast it turns into repeated saccades. Our Medium speed outperformed the others, achieving better rates for true and false positives.

2ND STUDY: EVALUATION OF ORBIT ROBUSTNESS

The goal of the second study was to evaluate the performance of Orbits in a more realistic setup, with a head-worn eye tracker and a smart watch. We tested the effects of the number of targets, trajectory size and target speed on the true and false positive rates of an abstract task.

Experimental Setup and Design

We conducted the experiment in a quiet laboratory setting where participants sat at a desk wearing a *Callisto 300*, a 1.54-inch multi-touch smart watch (see Figure 5). The device used Android 4.2.2 and a 240×240 resolution screen. Participants' eyes were tracked by a 30 Hz *Pupil Pro* head-mounted eye tracker with an average gaze estimation accuracy of 0.6° of visual angle [16]. The eye tracker was connected to a laptop that communicated with the watch through a wireless UDP connection. To reduce the fatigue of holding the arm up for an extended time and to ensure the same configuration between the watch and participants' eyes, these rested their arm on a support stand. The average distance between the eyes and the watch was of 35cm. The eye tracker was not calibrated to the watch display.

To test for true and false positives we used two conditions from the previous study: active target pursuit (ATP) and active target avoidance (ATA). Our independent variables were the number of targets on the screen (2, 4, 8 and 16), the trajectory diameter (Large: 2.6cm/4.25° of visual angle, Medium: 1.6cm/2.62° and Small: 0.6cm/0.98°) and the target speed (Slow: 120°/sec, Medium: 180°/sec, and Fast: 240°/sec). We recorded which target was activated (if any), and the time until a selection was made. To minimize acquisition errors we maximized the distance between the targets displayed by separating their initial positions by 720/n (with *n* equal to the number of targets displayed on-screen) and by having half of the targets move in opposing directions (clockwise and counter clockwise). As identified in the first study, we used a window size of one second and a correlation threshold of 0.8.

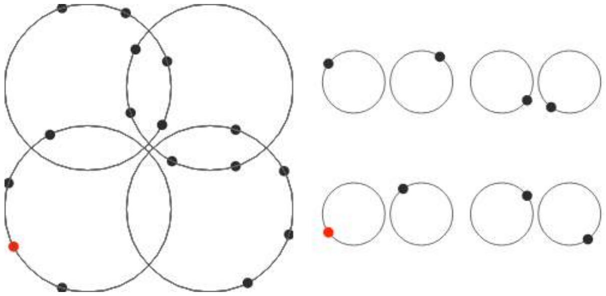

Figure 6. Two ATP trials as used in the second study. Left: 16 medium sized Orbits. Right: eight small sized Orbits. The read target indicates which one the users should actively follow with their eyes.

Participants

We recruited twelve participants (8M/4F), aged between 20 and 36 years (M = 27.3). On a 1 to 7 scale (no experience to very experienced), participants rated their experience with eye tracking at 3.6, with wearable devices at 1.9, with smart watches at 1.2, and with analogue watches at 4.4. Two participants wore contact lenses and five needed glasses, but were not wearing them during the study.

Procedure

Upon arrival, participants signed a consent form and completed a demographics questionnaire. Participant then strapped the smart watch to their non-dominant hand and wore the eye tracker. We ensured that the devices were correctly worn and that the eye tracking camera could see the user's eyes.

In each task participants were first presented with a screen with instructions. Upon tapping the screen of the smart watch for two seconds the task started. In the ATP task, users were presented with 36 trials (3 sizes × 3 speeds × 4 numbers of targets), each ending automatically after an eight seconds timeout or when participants acquired any of the targets displayed. A trial in the ATP task always started automatically two seconds after the previous trial was completed (see Figure 6). In these trials, the target participants had to follow was coloured in red. In the ATA task participants were presented with the same 36 trials, each ending and starting after a tap on the screen of the watch. Participants ended each trial once they correctly checked the time presented on-screen, wrote it down on a sheet of paper and started the next trial when ready. Both the ATP and the ATA blocks were repeated twice (36 trials × 4 blocks = 144 trials). All participants performed the blocks in the same order: ATA, ATP, ATP and ATA, but the trial order within each block was randomised to reduce practice and fatigue effects.

Results

We compared the true positive rate – the ratio of trials in which the system selected the target intended by the users – in the ATP trials (see Figure 7) with a factorial repeated-

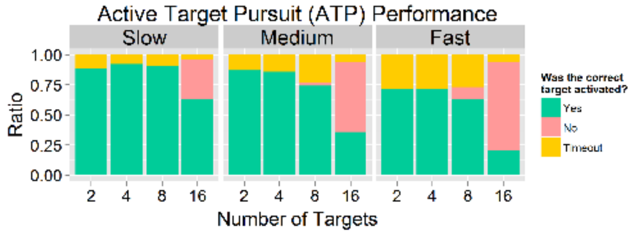

Figure 7. The ATP performance results from the second study. The slowest speed showed the overall best performance. Interfaces with up to eight orbits performed similarly.

measures ANOVA, Greenhouse-Geisser-corrected in cases where Mauchly's test indicated a violation of sphericity and with Bonferroni-corrected post-hoc tests. We found no significant effect neither of target Size ($F_{2,24} = 1.85$, $p = 0.18$, $\eta^2 = 0.02$) nor of interactions between independent variables at $p < .05$.

The true positive rate in the Slow speed (0.84) was 17% larger than the Medium speed (0.71) and 45% larger than the Fast speed (0.57). This effect was significant and large ($F_{2,24} = 38.0$, $p < .001$, $\eta^2 = 0.11$), with significant differences between all combinations of speeds ($p < .001$).

We also found a large significant effect of the Number of Targets on the true positive rate ($F_{3,36} = 41.8$, $p < .001$, $\eta^2 = 0.26$), but only the condition with 16 targets (0.39) was significantly different than the others (0.81) at $p < .001$.

We used the data from the ATA blocks to evaluate the false negative rate – the ratio of trials in which the system detected an activation while the user was reading the time. We found a moderate significant effect of the Number of Targets ($F_{1.47,17.6} = 5.22$, $p = .024$, $\eta^2 = 0.07$), but only the condition with 16 targets (0.10) was significantly different from the others (0.021) at $p < .05$.

ORBITS APPLICATIONS AND QUALITATIVE STUDY

Encouraged by these results, we developed three example applications that help illustrate Orbits in practice. We deployed them on a Callisto 300 smart watch and tracked gaze input with a Pupil Pro head-mounted eye tracker [16].

Music Player

The first example application was a music player (see Figure 1). The interface consisted of five different Orbits that allowed users to perform several discrete actions such as play/pause (one 1.6cm target), skip to the previous or next song (two 1cm targets with opposing directions) and adjust the playback volume (two 1cm targets with opposing directions). All targets shared the same angular speed of 180°/sec. The goal of this interface was to provide fast and hands-free access to music content, enabling interaction in previously challenging (e.g., biking to work) or cumbersome scenarios (e.g. using both hands to type a document).

Figure 8. The notifications face plate interface. In this example, each colored target represents an application's notifications. Because the size of each Orbit represents the number of unattended notifications (larger – more), users can have an overall understanding of what needs attending to (left). By following a target, users effectively change its visual feedback from an abstract (a colored dot) to iconic representation (the application's logo and exact number of unattended notifications), managing the limited screen space by presenting detailed information on a need-to-know basis (middle). Lastly, the Orbits control with a counter-clockwise moving target informs the user of which application holds the latest notification (right).

Notifications Face Plate

The second example was a notification panel that presented six coloured targets on a watch's face (see Figure 8). Each target and individual colour represented an application (e.g., Facebook, Snapchat) and the size of their trajectory represented the number of unaddressed notifications (the bigger the diameter the more notifications it represented). The trajectories of these targets ranged from 2.6 to 0.6cm and all shared an angular speed of 180°/sec. The goal of the application was to highlight some of the unique qualities of Orbits interfaces. This included how the selection area of these targets was no more than 0.1cm in size (making it very challenging to acquire with touch input) and how it would expand to represent additional information such as the number of notifications and the logo of the application when users would follow it with their eyes (effectively managing the limited screen space by presenting information only on a need-to-know basis). Furthermore, Orbits also supported novel applications of established design principles such as Gestalt theory on focal points [31]: newer notifications would alter the direction of their target making them more noticeable to the user through contrast of movement.

Missed Call: Menu

The third and last example application provided a quick access menu to a contextual event: a missed call (see Figure 9). The interface consisted of a 2.6cm main Orbits that informed users of the event, and upon acquisition would display four other controls of 1cm diameter. These four, smaller Orbits allowed users to call-back, reply-to, store the number or clear the event. All these shared targets with the same angular speed of 180°/sec. Finally, these four Orbits would disappear after four seconds of inactivity. The goal of this interface was to enable users to inconspicuously

Figure 9. The missed call menu. In this interface, users interact with iconic orbits that represent common communications controls. Upon being notified of a missed call (left), users can acquire this Orbits control to quickly access a small four item menu which allows them to either call or text back, to store the number or the clear the notification (right).

address common communication events that can occur in sensible or inappropriate situations (e.g., meetings).

Evaluation

We evaluated all three applications immediately after the second study and with the same participants. We first demonstrated how to use each application, and asked participants to comment on the usefulness and ease of interaction of each one, similarly to how Chen *et al.* evaluated Duet [6]. Participants then interacted with the applications in a spacious environment, where they were free to walk around and even use a stationary bike. Participants rated each application on their ease of use and usefulness on a 7-point scale. We recorded audio and video of the sessions, which were later transcribed for analysis.

All three interfaces were generally well received. As expected, the initial reaction was of slight bewilderment and confusion, as participants tried to make sense of all Orbits displayed – "*it looks a bit chaotic*" (P1). Furthermore, several expected the interfaces to respond to traditional gaze input, such as fixations on the icons or anywhere inside the orbits (P8, P9); or eye gestures for specific actions such as rising or decreasing the volume of the music player (P1, P7, P11) – "*I did not think the (moving) dots were part of the interaction*" (P9). After a small introduction to each interface, participants also expressed concern with the technique's learning curve, as they still felt "*overwhelmed*" (P12) by the "*many moving dots*" (P11). Despite the lukewarm initial reception, after a couple of minutes with each interface most participants understood and felt comfortable with Orbits – "*I get it now, now that I use it*" (P8). The following is participant feedback for each individual interface.

Music Player – of all interfaces, the music player was the one participants were more eager to try. After a brief introduction, several participants suggested the targets to be colour coded (P1, P2, P4), so it would be easier to "*understand the relation between (an Orbits') dot and icon*" (P2). Others suggested a smaller size for the play Orbits, as it is confusing to have the dot move "so far away from the (play) icon" (P2), and because it overlaps with the volume and skip Orbits. This latest concern was echoed amongst four participants (P1, P2, P3, P7), who felt it "*it was hard to keep following an Orbits (' dot) once it crosses paths with another*" (P1). Despite these initial concerns, the consensus amongst participants was that even though "*it is not a familiar interface*" (P6), and it might "*take some time to get used to" (P1), it is "easy to learn and understand*" (P5) – "*if you really had this device you would know exactly what each orbit does*" (P2). Additionally, participants envisioned using the system at the gym (P3) and while jogging (P1, P2, P8), or for more general purposes such as cooking (P6), riding the bus (P5), driving a car (P9) or even shopping (P10). Finally, participants rated the ease of use of the music player interface at 4.58 (*SD* = 1.83) and its usefulness at 4.75 (*SD* = 1.71).

Missed Call Menu – the simpler of the interfaces, the missed call menu was subject to very few comments from the participants. Most reported the interface was easy to understand and use, and that it would be useful when you want to quickly (P2) and privately address a call (P2, P7, P9, P12) without bothering others – such as in the bus, the cinema (P4) or in the classroom (P12). As with the music player, one participant also reported that the interface would be useful when one's hands are occupied (e.g., cooking, P6). Participants rated the ease of use of the missed call menu at 5.25 (*SD* = 2.01) and its usefulness at 4.50 (*SD* = 1.62).

Notifications Face Plate – whereas it was by far the most popular of the three interfaces at the end of the study, the notifications face plate initially received mixed feedback from the participants. Some felt there were "*too many Orbits*" on-screen (P8), causing the eyes to "*not know where to look*" (P1). As such, several participants did not notice the opposing, clockwise movement of the newest notification (P1, P6, P7, P8, P12) – "*because (of how) the (Orbits') dots are distributed, it is hard to compare their movements*" (P10). While most of these participants reported they could identify this movement after learning about it, they suggested graphical cues, not movement, to represent the newest notification. Their ideas included a target ("dot") that would blink (P4, P6) or pulse (P6); that would be larger (P4) than the others; or simply replacing the target with an icon that would convey novelty (P1).

The metaphor of bigger and faster Orbits for applications with more notifications also divided participants. While several agreed that faster and bigger Orbits (closer to the edge) evoked urgency and importance (P2, P3, P4, P5), and were generally easier to spot (P12), others felt applications with more notifications should be at the centre of the screen (and thus smaller and slower). Their preference for the centre was justified by: how the centre is far "away from the distractions of the clock face" (P1, P6, P8, P10); how slower and smaller Orbits are "easier" to follow (P1, P3, P7, P9, P10, P11); and how the centre is normally associated with "*important information*" (P9). Despite this initial reaction to our design decisions, most participants recognized the benefits of this interface after some minutes of interaction, stating that "*you do not need to check your phone to know what is happening*" (P3) or that "*information only pops when you want (it)*" (P4), enabling "*a lot more options on-screen*" (P7). The most enthusiastic feedback on this interface included: "*this one is awesome*" (P6), "*I love this one!*" (P7), "*that is quite cool, actually*" (P9). Finally, participants rated the ease of use of this application at 4.08 (*SD* = 1.44) and its usefulness at 5.42 (*SD* = 1.38).

LIMITATIONS AND CONCLUSIONS

In summary, we demonstrated the usability of Orbits as a robust interaction technique for smart watches. We demonstrated that with a speed of 120°/sec up to eight moving targets could be reliably detected with an average accuracy of 83% and zero false activations. The approach also has a low false positive rate, triggering a false selection in just 2.1% of trials when users read the time on the watch.

Our studies demonstrate that Orbits does not depend on any particular tracking technology. In fact, both of our studies used affordable, consumer-grade eye trackers, both remote and head-mounted. However, to be able to use Orbits in commercial systems, these must provide some form of eye tracking capability. We envision two ways for how this can be achieved. The first is adding remote eye tracking to the watch itself: companies such as *EyeTribe* are already exploring this possibility[5]. The second is adding eye tracking capabilities to head-worn devices. *Google Glass* already has an infra-red proximity sensor that recognises blinks and winks, and Google has patents on using eye tracking to unlock the Glass screen [25].

The main motivation of our work was to enable hands-free interaction on smart watches, but we also foresee the combination of Orbits with other modalities. For example, while manual or other hand-based techniques could remain as the primary input modality due to their input speed, Orbits could be added as a complementary modality when the hands are otherwise engaged (e.g., cooking). Additionally, Orbits can be used in devices other than smart watches, particularly where hand interaction is difficult (e.g., using a smartphone while on a treadmill) or impossible (e.g., assistive interfaces).

Because Orbits uses the relative movement of the eyes, it does not need any registration between the user's gaze and the watch's coordinate systems. This removes the necessity of the scene cameras in head-worn eye trackers, which often introduce privacy concerns. This means that Orbits can potentially be used with EOG (electrooculography) based trackers, which monitor eye movements through the electrical signal they generate [4, 29].

Lastly, we identify three main limitations in our studies. First, our recognition results reflect the system's performance in a controlled setting. Participants had their arms resting on a stand, which reduces the ecological validity of the experiment. However, users were still allowed to use the system freely when evaluating the applications (e.g., standing). While we did not collect any quantitative performance data for these tasks, their qualitative responses show that the system still works in a less controlled setting. Second, we do not compare Orbits with other smart watch techniques. As smart watch technology is still in its infancy, no single technique can be considered a universal baseline for us to compare against. Third, the performance of the technique depends on the context of use. While touch can be quicker for few targets on the screen, Orbits is more adequate for when the user's hands are otherwise engaged. Finally, for the second study, we chose algorithm parameters based on the data from our first study. However, these parameters are by no means the absolute best. We emphasized low false positives, somewhat penalising our true positive rate. Depending on the application, other trade-offs might be more reasonable: e.g., for more responsive systems we could reduce the window size at the cost of accuracy.

REFERENCES

1. Barnes, G. 2012. Rapid learning of pursuit target motion trajectories revealed by responses to randomized transient sinusoids. *Journal of Eye Movement Research 5*, 3.
2. Baudisch, P. and Chu, G. 2009. Back-of-device interaction allows creating very small touch devices. In *Procs of CHI '09*, 1923–1932. ACM.
3. Blasko, G. and Feiner, S. 2004. An interaction system for watch computers using tactile guidance and bidirectional segmented strokes. *Eighth International Symposium on Wearable Computers*, 120–123. IEEE.
4. Bulling, A., Roggen, D., and Tröster, G. 2009. Wearable EOG goggles: Seamless sensing and context-awareness in everyday environments. *Journal of Ambient Intelligence and Smart Environments 1*, 2: 157–171.
5. Burke, M.R. and Barnes, G.R. 2006. Quantitative differences in smooth pursuit and saccadic eye movements. *Experimental Brain Research 175*, 4, 596–608.
6. Chen, X., Grossman, T., Wigdor, D.J., and Fitzmaurice, G. 2014. Duet: exploring joint interactions on a smart phone and a smart watch. In *Procs of CHI '14*, 159–168. ACM.
7. Cymek, D.H., Venjakob, A.C., Ruff, S., Lutz, O.H.-M., Hofmann, S., and Roetting, M. 2014. Entering PIN Codes by Smooth Pursuit Eye Movements. *Journal of Eye Movement Research 7*, 4.

---

[5] http://www.cnet.com/uk/news/eye-tracking-on-a-smartwatch-theeyetribe-prototype/

8. Dietz, P. and Leigh, D. 2001. DiamondTouch: a multi-user touch technology. In *Procs of UIST '01,* 219–226.
9. Drewes, H. and Schmidt, A. 2007. Interacting with the Computer Using Gaze Gestures. In *Procs of INTERACT '07*, 475–488. Springer.
10. Dybdal, M.L., Agustin, J.S., and Hansen, J.P. 2012. Gaze Input for Mobile Devices by Dwell and Gestures. In *Procs of the Symposium on Eye Tracking Research and Applications*, 225–228. ACM.
11. Fekete, J.-D., Elmqvist, N., and Guiard, Y. 2009. Motion-pointing: target selection using elliptical motions. In *Procs of CHI '09*, 289–298. ACM.
12. Haines, R. *Vintage Watches*. Krause Publications, 2010.
13. Harrison, C. and Hudson, S.E. 2009. Abracadabra: wireless, high-precision, and unpowered finger input for very small mobile devices. In *Procs of UIST '09*, 121–124. ACM.
14. Hutchinson, T.E., White, J. K.P., Martin, W.N., Reichert, K.C., and Frey, L.A. 1989. Human-computer interaction using eye-gaze input. *IEEE Transactions on Systems, Man and Cybernetics, 19*, 6: 1527–1534.
15. Jacob, R.J. 1990. What you look at is what you get: eye movement-based interaction techniques. In *Procs of CHI '90*, 11–18. ACM.
16. Kassner, M., Patera, W., and Bulling, A. 2014. Pupil: an open source platform for pervasive eye tracking and mobile gaze-based interaction. In *Procs of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, 1151–1160. ACM.
17. Kratz, S. and Rohs, M. 2009. HoverFlow: expanding the design space of around-device interaction. In *Procs of MobileHCI '09*, 4. ACM.
18. Lutz, O., Venjakob, A., and Ruff, S. 2015. SMOOVS: Towards calibration-free text entry by gaze using smooth pursuit movements. *Journal of Eye Movement Research 8*, 1: 1–11.
19. Majaranta, P. and Räihä, K.-J. 2002. Twenty Years of Eye Typing: Systems and Design Issues. In *Procs of the 2002 Symposium on Eye Tracking Research & Applications*, 15–22. ACM.
20. Møllenbach, E., Hansen, J.P., Lillholm, M., and Gale, A.G. 2009. Single stroke gaze gestures. In *Procs of the Extended Abstracts of CHI '09,* 4555–4560.
21. Morimoto, C.H. and Mimica, M.R.M. 2005. Eye gaze tracking techniques for interactive applications. *Comput. Vis. Image Underst. 98*, 1: 4–24.
22. Oakley, I. and Lee, D. 2014. Interaction on the edge: offset sensing for small devices. In *Procs of CHI ' 14*, 169–178. ACM.
23. Perrault, S.T., Lecolinet, E., Eagan, J., and Guiard, Y. 2013. Watchit: simple gestures and eyes-free interaction for wristwatches and bracelets. In *Procs of CHI '13*, 1451–1460. ACM.
24. Pfeuffer, K., Vidal, M., Turner, J., Bulling, A., and Gellersen, H. 2013. Pursuit calibration: Making gaze calibration less tedious and more flexible. In *Procs of UIST '13*, 261–270. ACM.
25. Raffle, H.S., Wong, A., and Geiss, R. 2012. Unlocking a screen using eye tracking information.
26. Rantala, J., Kangas, J., Akkil, D., Isokoski, P., and Raisamo, R. 2014. Glasses with Haptic Feedback of Gaze Gestures. In *Procs of the Extended Abstracts of CHI '14*, 1597–1602. ACM.
27. Roudaut, A., Huot, S., and Lecolinet, E. 2008. TapTap and MagStick: improving one-handed target acquisition on small touch-screens. In *Procs of the working conference on Advanced visual interfaces*, 146–153. ACM.
28. Sibert, L.E. and Jacob, R.J.K. 2000. Evaluation of Eye Gaze Interaction. In *Procs of CHI '00*, 281–288. ACM.
29. Vidal, M., Bulling, A., and Gellersen, H. 2011. Analysing EOG signal features for the discrimination of eye movements with wearable devices. In *Procs of the 1st international workshop on pervasive eye tracking & mobile eye-based interaction*, 15–20. ACM.
30. Vidal, M., Bulling, A., and Gellersen, H. 2013. Pursuits: spontaneous interaction with displays based on smooth pursuit eye movement and moving targets. In *Procss of Ubicomp '13*, 439–448. ACM.
31. Wertheimer, M. and Riezler, K. 1944. Gestalt theory. *Social Research*, 78–99.
32. Xiao, R., Laput, G., and Harrison, C. 2014. Expanding the input expressivity of smartwatches with mechanical pan, twist, tilt and click. In *Procs of CHI '14*, 193–196.
33. Yu, N.-H., Tsai, S.-S., Hsiao, I.-C., et al. 2011. Clip-on gadgets: expanding multi-touch interaction area with unpowered tactile controls. In *Procs of UIST '11*, 367–372. ACM.
34. Zhai, S., Morimoto, C., and Ihde, S. 1999. Manual and Gaze Input Cascaded (MAGIC) Pointing. In *Procs of CHI '99*, 246–253. ACM.
35. Zhang, Y., Bulling, A., and Gellersen, H. 2013. Sideways: A gaze interface for spontaneous interaction with situated displays. In *Procs CHI '13*, 851–860.
36. Zhang, Y., Müller, J., Chong, M.K., Bulling, A., and Gellersen, H. 2014. GazeHorizon: enabling passers-by to interact with public displays by gaze. In *Procs of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 559–563. ACM.