
Orbits: Enabling Gaze Interaction in Smart Watches Using Moving Targets

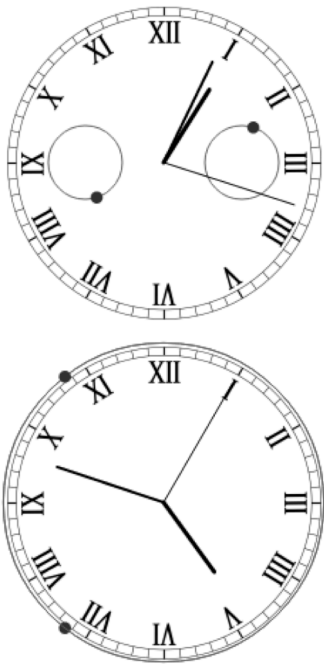


Figure 1: Two watch faces with Orbits controls. Top: two Orbits with single targets, moving in a 1cm trajectory. Bottom: one Orbit with two targets moving in a 2.8cm trajectory (the edge of the watch's face).

Augusto Esteves

Lancaster University
Lancaster, United Kingdom
a.esteves@lancaster.ac.uk

Eduardo Velloso

Lancaster University
Lancaster, United Kingdom
e.velloso@lancaster.ac.uk

Andreas Bulling

Max Planck Institute for
Informatics
Saarbrücken, Germany
bulling@mpi-inf.mpg.de

Hans Gellersen

Lancaster University
Lancaster, United Kingdom
hwg@comp.lancs.ac.uk

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s). Copyright is held by the author/owner(s). UbiComp/ISWC '15 Adjunct, September 7–10, 2015, Osaka, Japan. ACM 978-1-4503-3575-1/15/09.
<http://dx.doi.org/xx.xxxx/xxxxxxx.xxxxxx>

Abstract

In this paper we demonstrate Orbits, a novel gaze interaction technique that accounts for both the reduced size of smart watch displays and the hands-free nature of conventional watches. Orbits combines graphical controls that display one or multiple targets moving on a circular path, with input that is provided by users as they follow any of the targets briefly with their eyes. This gaze input triggers the functionality associated with the followed target – be it answering a call, playing a song or managing multiple notifications.

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

Smart watches are wrist-worn, wearable devices that enable users to discreetly consume information at a glance. While touch and speech are the de facto interaction modalities for smart watches, in this paper we introduce a novel interaction technique that relies, for the first time in such devices, on gaze input. Named



Figure 2: Two examples of the music player application. In this interface, three Orbits enable users to play and pause a song (center Orbit), increase or decrease its volume (left Orbit) and skip back or ahead in the playlist (right Orbit). Both left and right Orbits have two targets moving in opposing directions (clockwise: increases the volume, skips ahead; counterclockwise: decreases the volume, skips back). A red target represents a user selection.

Orbits, our technique not only overcomes the fat finger problem (by effectively enabling hands-free interaction) but also presents three immediate advantages over traditional eye-tracking solutions. First, it works with most off-the-shelf eye-trackers. Second, it does not require any calibration between the eye tracker and the display. And third, it works independent of target size.

In the remainder of this paper we will introduce Orbits in detail, and present three example applications that illustrate how gaze input can be used to accomplish common, everyday tasks: a music player, a notifications face plate and missed call menu.

Related Work

One of the biggest challenges of designing smart watch applications is ensuring that users are able to interact through the device's very small display, while at the same time being able to consume the content that is displayed. So far, the most common approach to this challenge is to decouple a watch's input and output space. Examples include moving the input space to the edge of the watch's face [e.g., 4, 8], to its strap [5] or directly above its display through a combination of the watch's magnetometer and a magnet strapped to the user's finger [2].

Most successful smart watches on the market also rely on this approach. Interaction with the Apple Watch¹ is achieved through a combination of direct touch and input through a physical wheel on the body of the watch; while the Pebble Time² solely supports input through three buttons on its side.

1 <https://www.apple.com/watch/technology/>

2 https://getpebble.com/pebble_time

While these examples successfully illustrate how to enable rich interaction in the physically limited space of a smart watch, they also share a common limitation that is especially aggravating in wearable devices: they constrain both of the user's hands during interaction. As such, in this paper we argue for gaze, not touch, as the ideal modality for this domain, as it naturally enables hands-free interaction.

But the use of gaze as input is not without its challenges. Gaze pointing – the user's fixation on different targets in the interface – can be highly efficient [6], but is affected by accuracy issues and requires users to go through a tedious calibration process. Such limitations are aggravated in smart watch devices, where targets are especially small and the unconstrained nature of the interaction makes mapping the user's gaze to the screen unfeasible.

Orbits

To overcome these challenges we have developed a gaze interaction technique for smart watches based on smooth pursuits: the movement the eyes perform when following a moving target [1]. By correlating the movement of the eyes with the movement of the targets being displayed, we are able to robustly identify which of the targets the user is looking at. This idea is not new. Vidal *et al.* [7] was arguably the first to use it as input method for large, public displays, and successfully demonstrated the benefits of the technique: no calibration is needed as we rely on the relative movement of the eyes in relation to the targets; and it works independently of target size, as these are identified by their movement, not their position.

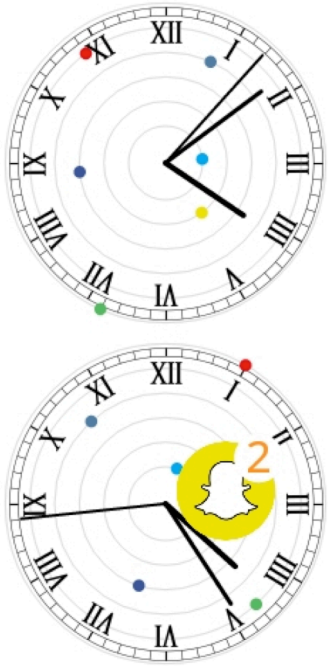


Figure 3: The notifications face plate interface. Six Orbits representing six applications are positioned in the center of watch's face. Each has a unique trajectory size (1~2.8cm) and a colored target matching the application it represents (e.g., light blue for Skype). The size of an Orbits' trajectory conveys the number of unattended notifications from an application (bigger = more). Lastly, following any the targets transform these into iconic representations for explicit information (bottom).

Inspired by mechanical watches, Orbits populates an interface with targets moving in circular paths (see Figure 1). These provide both discrete and continuous controls, allowing users to, e.g., turn the sound of their smart watches on and off by following the corresponding target until the control is activated; or raising and decreasing the volume by following the corresponding target for the desired amount of time.

Each Orbit comprises of one or more targets moving along a trajectory, and an iconic representation that can be displayed in the center of the Orbit (so users can read it without fear of activating the control) or as a moving target (so each target behavior is explicitly represented). Different Orbits have different sized trajectories and targets of different angular speeds and directions. Finally, the technique detects which of the targets a user is looking at by performing a Pearson's correlation between gaze data points and each of the targets' positions (x, y) within a certain window of time. If the correlation value is superior to the pre-defined correlation threshold the Orbit control is activated.

Applications

We developed three example applications that illustrate the use of Orbits in smart watch devices. These were developed with the *Processing* programming language and deployed on a Callisto 300³ smart watch. Gaze input was captured through a Pupil Pro head-mounted eye-tracker [3].

Music player

The music player application consists of five Orbits that provide users with gaze controls to play and pause a

song (one target moving in a circular trajectory of 1.6cm), skip back or ahead and adjust the playback volume (both Orbits with two targets moving in opposing directions in a 1cm trajectory) – see Figure 2.

This interface illustrates how gaze input can provide access to music content in challenging scenarios such as riding the subway while holding on to the rail, or when using both hands to type a document.

Notifications face plate

The notifications face plate example uses six Orbits to represent six applications and their unattended notifications (see Figure 3). Each Orbit has a single colored target, moving in a unique trajectory size that matches the number of unattended notifications from the application it represents (more = bigger). Finally, when users follow a target this transforms into an iconic representation of the application (its logo) and the explicit number of notifications to attend.

The goal of this example is to demonstrate some of the unique qualities of Orbits as interaction technique. These include targets that are no more than 0.1cm in size (very hard to acquire with touch); how these expand and contract on a need-to-know-basis to manage the limited screen space; and how by contrasting the movement of a target against all others it is possible to convey extra information such as the latest unattended notification.

Missed call: menu

The last example application illustrates how Orbits can be used in a more traditional options menu. In this example, users can act upon a missed call notification by following the targets around several iconic

³ <http://www.iconbit.com/products/watches/>



Figure 4: The missed call menu interface. Top: an Orbit control with an iconic representation informing the user of a missed call. Upon following the target the user is shown four other Orbits, each with an iconic representation of an action: call back, text back, add to contacts and clear notification.

representations related to this event, e.g.: call back, store the number, clear notification (see Figure 4).

The goal of this example is to demonstrate how Orbits can enable users to inconspicuously address common smartphone events, even in sensible contexts such as meetings or when talking to others.

Conclusion

In this paper we introduced Orbits, a novel interaction technique for smart watches that relies not on touch, but on gaze input. We also presented three example applications that illustrate how users can benefit from interfaces that combine movement and gaze controls.

While the focus of this paper was on smart watches, due to the calibration-free nature of the technique, and because it works independent of target size, we believe a wide range of small display and wearable devices can benefit from it. We envision Orbits interfaces in car dashboards, exercise equipment (e.g., rowing machines, treadmills), ubiquitous environments (e.g., lighting and entertainment systems) and in novel, more robust assistive interfaces. Additionally, the technique can also be used to interface with the real world, by detecting and reacting to naturally occurring movements in the environment (e.g., the London Eye, digital billboards).

In sum, Orbits provides a glimpse into how we will be able to interact with the devices of the future, a future where touch is not core to the experience, but complimentary to other modalities such as a gaze.

References

1. Barnes, G. R. 2012. Rapid learning of pursuit target motion trajectories revealed by responses to randomized transient sinusoids. *Journal of Eye Movement Research*, 5(3).
2. Harrison, C., & Hudson, S. E. 2009. Abracadabra: wireless, high-precision, and unpowered finger input for very small mobile devices. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*, 121-124.
3. Kassner, M., Patera, W., & Bulling, A. 2014. Pupil: an open source platform for pervasive eye tracking and mobile gaze-based interaction. In *Proceedings of the International Joint Conference on Pervasive and Ubiquitous Computing*, 1151-1160.
4. Oakley, I., & Lee, D. 2014. Interaction on the edge: offset sensing for small devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 169-178.
5. Perrault, S. T., Lecolinet, E., Eagan, J., & Guiard, Y. 2013. Watchit: simple gestures and eyes-free interaction for wristwatches and bracelets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1451-1460.
6. Sibert, L. E., & Jacob, R. J. 2000. Evaluation of eye gaze interaction. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, 281-288.
7. Vidal, M., Bulling, A., & Gellersen, H. (2013). Pursuits: spontaneous interaction with displays based on smooth pursuit eye movement and moving targets. In *Proceedings of the International Joint Conference on Pervasive and Ubiquitous Computing*, 439-448.
8. Xiao, R., Laput, G., & Harrison, C. (2014). Expanding the input expressivity of smartwatches with mechanical pan, twist, tilt and click. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 193-196.