A Rotary Dial for Gaze-based PIN Entry

Darrell S. Best and Andrew T. Duchowski*
School of Computing, Clemson University

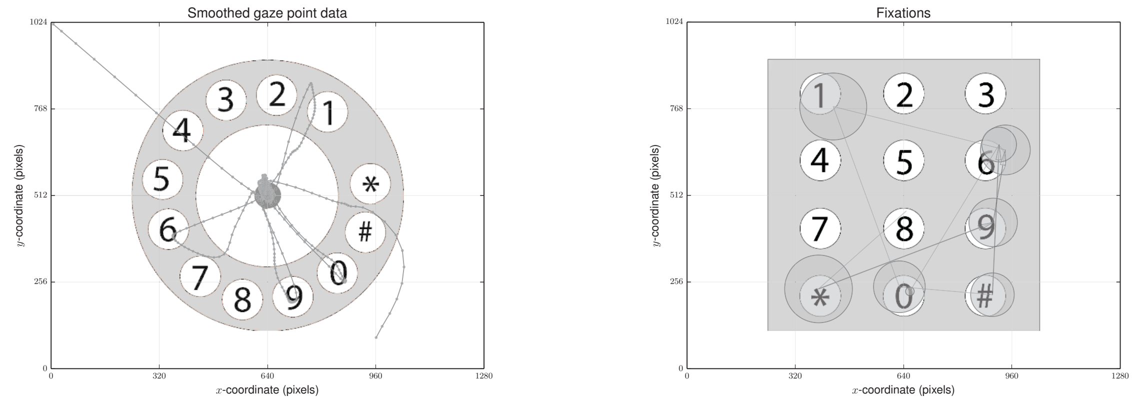

Figure 1: *Entering 9610: a rotary design for gaze-based PIN entry affords faster input than a traditional keypad grid design.*

Abstract

We detail the design and evaluation of a rotary interface for gaze-based PIN code entry. Interface design promotes equal distance between PIN numerals, leading to a circular layout resulting in the choice of a rotary telephone dial metaphor. The rotary's speed advantage over the traditional grid-based (e.g., keypad) design is derived from its elimination of dwell time for gaze-based numeral selection, relying instead on a weighted voting scheme of numerals whose boundaries are crossed by the streaming (smoothed) gaze points. Screen center-bias is exploited by requiring users to transition to and from the center to the rotary numerals, in order to enter in the PIN code sequence. Compared with the keypad layout, empirical results show that PIN digit entry errors do not differ significantly between interfaces, although the rotary incurs fewer errors overall. Expressing preference for the rotary, users appeared to quickly grasp its operation.

Keywords: eye tracking, security, authentication

Concepts: •**Security and privacy** → *Usability in security and privacy;* •**Human-centered computing** → *Pointing;*

1 Introduction

Calibration-free eye tracking will afford immediate gaze input for a myriad of unmodified devices such as tablet computers [Wood and Bulling 2014]. A particularly tantalizing application of gaze input is touchless unlocking of these devices via entry of a Personal Identification Number, or PIN. Current touch-based user authentication is prone to hacking by shoulder-surfing and smudge attacks [Lashkari et al. 2009; Biddle et al. 2012]. Using gaze to enter one's PIN would prevent both types of attacks.

Hoanca and Mock [2006] implemented gaze-based PIN entry with *PassFaces*, requiring users to look at a grid of face images, e.g., a 3×3 grid of American presidential portraits, and then entering their PIN by fixating a memorized sequence of presidents (e.g., Lincoln, Van Buren, etc.). Although theirs and similar approaches since (reviewed below) showed promise, they relied on dwell time as a means of selecting particular image face tiles by gaze.

In this paper we introduce a gesture-like approach wherein the need for dwell time is obviated by virtue of the graphical design chosen for the interface, which affords selection through analysis of gaze boundary crossings (see Figure 1). Not having to rely on dwell time frees the system from the dependency on the selection of a fixation detection algorithm. Instead, the boundary-crossing approach requires only minimal processing of the streaming eye movement signal which ultimately leads to faster data entry. We show a clear speed advantage over an approach based on dwell-time without any penalty in accuracy (i.e., due to errors incurred during PIN entry).

2 Background

The present work falls under the broad category of graphical passwords [Biddle et al. 2012], e.g., perhaps partially resembling *PassFaces*, but it is specific to gaze-based entry using an eye tracker. As such, our interface overlaps and draws from prior work utilizing gaze gestures as well as circular menus for item (e.g., character) selection. Our rotary interface may overlap with these prior efforts, but it differs in subtle and important ways.

Graphical Passwords. Graphical passwords connote the act of drawing a password [Van Oorschot and Thorpe 2008; Sreeramareddy et al. 2012], but when entering a password with one's eyes, this is generally not the metaphor one typically construes. Rather, the notion of selection by gaze is the more common connotation.

*[bbest|duchowski]@clemson.edu

ETRA '16,, March 14-17, 2016, Charleston, SC, USA
ISBN: 978-1-4503-4125-7/16/03
DOI: http://dx.doi.org/10.1145/2857491.2857527

Hoanca and Mock [2006] introduced a gaze-based variant of *PassFaces* password entry. Their design relied on dwell time-based fixation detection to select the individual image tiles that would form the user's unique password sequence. Kumar et al. [2007] evaluated this type of selection-based interface and, not surprisingly, found it slower than regular keyboard entry. More recent efforts, while testing salience-masked images to cue password locations, still relied on fixation detection to match against previously fixated password image regions [Bulling et al. 2012]. Our keypad interface essentially replicates these designs which we compare to the rotary alternative, whose central innovation is the elimination of the need for fixation detection. Instead, the rotary interface relies on stream-based processing of smoothed gaze data.

Drawing passwords, in contrast, falls under the perhaps somewhat narrower category of gaze gestural input. De Luca et al.'s [2007; 2009] *EyePIN* and *EyePassShapes* relied on a gesture alphabet, based on Wobbrock et al.'s *EyeWrite* [2008]. Although our rotary design is to a limited extent reminiscent of gaze gestures, it does not require the user to remember a shape. Rather, the user must remember a 4-digit PIN code and then glance to each of the digits to enter in the PIN. Since no fixation detection is made, one can consider eye movements gestures, but our system does not recognize them as such. Moreover, De Luca et al. [2007] present an approach called look and shoot that uses a button press instead of dwell time. Although they argue that dwell time is not appropriate for this kind of input, substitution of dwell time by a button press amounts to a similar keystroke-level delay. Our system obviates the need for either dwell time or button press. De Luca et al. [2009] noted that non-dwell time approaches are faster, but we are not aware of prior work evaluating a rotary-like PIN entry system.

Gaze Gestures. Beyond PINs and passwords, the general application of entering symbols by gaze falls under the category of eye typing [Majaranta and Räihä 2002]. Various efforts have been proposed for speeding up input, not all necessarily restricted to gaze, e.g., Perlin's *Quickwriting* [1998] or *ShapeWriter* [2009] from which we drew some inspiration. For gaze-based eye-typing, Isokoski's gaze gestures to off-screen targets [2000] showed some potential early on, however, the speed benefits for our rotary interface were largely expected due to *Dasher*'s boundary-crossing approach [Tuisku et al. 2008]. *Dasher* uses boundary-crossings to select characters and words outpacing traditional dwell-time based character entry during eye-typing, but only after some user training. Apart from *ShapeWriter*, none of the gaze-based systems were exploited for password entry that we are aware of.

Not all gaze gestural input techniques are necessarily expected to outpace dwell-time based input. Wobbrock et al.'s *EyeWriter* [2008] showed utility of gaze gestures for eye typing, but its performance was slower than dwell-time based eye typing because of its average stroke per character requirement. In the case of eye typing, dwell time outpaced gestures because it was faster, on average, to select one character key with dwell time than say 4 gaze gestures per character. In the specific case of PIN entry, where the alphabet is small, dwell time is a hindrance to speed.

Dasher's speedup is partially due to its predictive capability, which offers the most statistically likely characters for selection. Our rotary interface does not use such prediction, however, it does implement a weighted voting scheme which aids target selection based on boundary crossings. The voting scheme resembles somewhat Weaver et al.'s [2011] gaze point clustering for gaze-based password authentication, but their displays included only keypad and keyboard images from which to select password (or PIN) symbols. We test our rotary interface against this kind of spatial layout.

Circular Menus and Border Crossings. Perhaps the most similar gaze-gesture based display to our rotary interface was the *pEYE* menu interface designed by Huckauf et al. [2007; 2008]. These circular menus were meant for eye typing, although it was acknowledged that the design could be used for other applications. PIN entry was not one of them, however. Moreover, the *pEYE* menus lacked a central fixation point designed to exploit center-bias [Holmqvist et al. 2011; Peterson and Eckstein 2012].

pEYE menus were designed to be hierarchical in nature. Urbina et al. [2010] tested various combinations of pie segmentation (e.g., 4, 6, 8, 12 slices) and menu depths (e.g., 2, 3, 4) along with selection via either dwell time or border crossing. Urbina et al. suggest that up to six slices can be effectively and efficiently selected with eye trackers. This observation is likely to be confounded by the multi-level nature of the tasks explored, e.g., selection of consecutive slices *at different depths*. Our rotary design is effectively a mono-level pie menu but with a large center plateau region instead of an apex. Our experiment corroborates findings reported by Urbina et al. in terms of dwell time and border crossings: the former is considered more intuitive but slower.

Our implementation of the rotary dial is similar to Patidar et al.'s [2013] *Quickpie*, a *pEYE* menu with one depth and border crossing activation instead of dwell time. Unlike *Quickpie*, we handle gaze jitter and PIN selection by a combination of smoothing filter (to reduce jitter) and a weighted voting scheme, allowing the user to correct an unintended numeral selection by moving to the intended numeral prior to exiting the numeral ring back to the center region. Both *Quickpie* and *pEYE* user studies report on learnability rates, given multiple trials. We do so as well.

In our approach we first started with a tiled interface similar to *PassFaces* [Hoanca and Mock 2006]. We then designed the rotary input interface, resembling a mono-level *pEYE* menu activated by gaze border crossings, similar to *Quickpie*. As with Bee and André's *Quickwriting* adapted to eye gaze [2008], our rotary dial also relies on gaze to re-enter a large central region to finalize numeral input. Our rotary dial is limited to numeral input (not text characters) and does not and can not provide visual feedback to the user (as that would defeat the purpose of it being meant for secure PIN entry). Our user study differs from Bee and André's by evaluating data from 30 participants instead of only 3.

Refinements to our interface depend on signal smoothing along with a weighting scheme allowing the user to back out of an unintended selection. The resultant rotary interface design draws its entry speed advantage from these improvements as well as from holding the distance to PIN tiles constant.

3 Interface Design

A gaze-based PIN entry method free from the constraints of dwell time is, by design, expected to offer faster input than one carrying the built-in delay incurred by the dwell time timeout on which it is based. Ignoring dwell time, however, requires a clever design of the interface which precludes the potential for errors due to gaze traversing successive PIN keypad numerals (or tiles which represent them). We chose the classic rotary dial metaphor both for its speed advantages (highlighted below) as well as for its expected familiarity. Although the rectangular numeric keypad is commonly used on keyboards and smartphones (where the order of numerals can be reversed, e.g., the ①②③ row is either at top or bottom), the rotary dial is still likely to be remembered by most users.

Consider the tiled image face grid introduced by Hoanca and Mock [2006] for gaze-based password entry. For numeric PIN entry, we can replicate this design by a traditional numeric keypad, as shown

```
4444cccccc999999999ccccccc77776666cccccc11111111ccccc000000000cccccccccccc##
```

Figure 2: *Example smoothed gaze data stream produced by the rotary interface.*

in Figure 1. For PIN entry with this interface, the user is required to fixate, defined by dwell time, successive numerals (technically tiles on which they are represented, as per Hoanca and Mock, or in our design, referred to as *pads* since they are circular). Numeral pads should be of sufficient width, W, to promote selection by gaze in the presence of eye tracker error (usually about 1° visual angle, or about 30-35 pixels diameter at a $22''$ viewing distance on a $19''$ screen set to 1280×1024 pixel resolution).

Because of well-known "center-bias" [Holmqvist et al. 2011; Peterson and Eckstein 2012], where the first fixation is often made to the screen center, using dwell time the system could accidentally recognize the numeral 5 or 8 as the initial PIN numeral. To prevent this, the dwell time-based PIN entry interface must provide start and end pads to force the user to indicate to the system when PIN entry starts and ends, effectively start-of-text (SOT) and end-of-text (EOT) delimiters. In our implementation, we use the * and # pads as the SOT and EOT symbols, respectively.

To enter PIN 9610 on the numeric keypad, the user first fixates pad *, then each of the numeric pads in turn, and finally fixates # to complete the sequence, thus requiring 6 dwells to enter a 4-digit PIN. An actual entry example shown in Figure 1 indicates gaze started on 8 before proceeding to *, and then re-fixating 6 after completing the sequence by fixating #.

In contrast, to enter PIN 9610 on the rotary interface, the user is required to transition gaze to the center pad ● between successive numerals, i.e., ●9●6●1●0●, as shown in Figure 1, *without stopping to fixate* targets transitioned to in succession. The rotary's chief speed advantage lies in this aspect of its design. The rotary design also exploits center bias instead of requiring a workaround (i.e., artificial and potentially distant starting and ending points).

For gaze-based entry, note that the keypad design *must* use fixation detection to prevent accidental triggering of pad selections located midway along a trajectory, an instance of the classic Midas Touch problem [Jacob 1990]. For example, when entering PIN 4071, because eye movements are ballistic and stereotyped [Carpenter 1977], it is highly probable that pad 4 would be intersected by the trajectory between pads 7 and 1, resulting in the system's interpretation of the input sequence of numerals 40741.

4 Predicting Entry Time as a Fitts' Law Task

Modeling gaze-based PIN entry as a Fitts' Law task, the total expected entry time for the entire PIN sequence is given by the sum of gaze transitions between numeral pads,

$$\sum_{i}^{|\mathrm{PIN}|} \mathrm{MT}_i \quad \text{with} \quad \mathrm{MT}_i = a + b\log_2(2A/W) \qquad (1)$$

where MT_i (for movement time) denotes each gaze transition, a is a constant describing the response time per selection, b is the relationship between movement time and the Index of Difficulty (ID) of the transition, A is the (saccadic) amplitude or distance of gaze movement between successive numeric pad targets (modeled as Euclidean distance between pad target centers), and W is their width (diameter in this case). According to Jastrzembski's [?] keystroke-level task analysis of fixations during mobile phone usage and her use of Card et al.'s [?] Model Human Processor (MHP), care should be taken when considering the user's age. Accordingly, based on the MHP cycle time of the motor processor for younger adults, we

Table 1: *Modeled mean time (ms) to PIN entry per interface.*

	PIN: 3010		PIN: 0000	
user	keypad	rotary	keypad	rotary
young adult	4,195	1,722	2,793	1,725
older adult	5,771	3,202	3,248	3,207

set the selection response time parameter a to 70 ms, and the ID parameter b to 100 bits/s. For older adults, we set $a = 146$ ms and $b = 175$ bits/s. Saccade amplitude and pad distances are measured in pixels allowing direct comparison of both keypad and rotary interfaces, assuming comparable dimensions of the screen, interface, and size of pads. Figure 1 shows the screen size of both interfaces drawn to scale, centered on the screen.

There are two critical differences between the interface designs and hence their Fitts' Law models. First, because the rotary design obviates the need for dwell time, constant a is used unaltered, whereas it is increased by the minimum expected dwell time when modeling the numeric keypad. Majaranta et al. [?] suggest 450 ms as a usable minimum dwell time for eye typing interfaces. We therefore add in this constant to parameter a in (**??**) for the keypad interface. Second, due to the spatial layout of the numeric keypad, distances between numerals is not constant, e.g., the longest distances between numeric pads are between 3 (or 1) and 0. In contrast, the rotary design benefits from a constant distance between the interface center and any of its numeric pads. Moreover, the central target is much larger than any of the numeric pads and thus faster to transition to from any of the pads.

Numeric pads are each modeled with $W = 120$ pixels, except for the central ● pad of the rotary interface, which is modeled by $W = 420$ pixels. To highlight the potential differences in expected mean entry times for both interfaces, we considered the best and worst case PINs in terms of minimum and maximum pad distances on the numeric keypad. The PIN evoking the fastest expected time would thus be 0000, requiring four refixations of 0, while the one evoking the slowest would be 3010.

Results of Fitts' Law modeling (see Table **??**) clearly show the expected speed advantage of the rotary interface. Fitts' Law modeling of the two interfaces exposes the advantages of the rotary design:

- constant distance between successive targets,
- elimination of the need for target fixation detection, and
- exploitation of the screen's center bias obviating the need for SOT and EOT symbol delimiters.

For these reasons, we hypothesized that a speed advantage would emerge when tested empirically with 4-digit PINs.

5 Implementation

Gaze-based input generally relies on signal processing of the raw eye movement data $g_i = (x_i, y_i, t_i)$, including signal de-noising, filtering, and possibly classification into fixations $f_i = (\hat{x}_i, \hat{y}_i, t_i, d_i)$, where $(\hat{x}_i, \hat{y}_i)$ coordinates indicate the position of the centroid of the fixation, with t_i recording the timestamp of the fixation and d_i the fixation's duration [Duchowski et al. 2014a].

Classification of raw data into fixations is often based on signal filtering, usually dispersion- or velocity-based. Most commercial eye tracking software packages provide only a limited choice of fixation detection algorithms (i.e., filters), sometimes relying on proprietary filter parameters that are not disclosed to the user. Some systems rely on dispersion-based "fixation pickers" [Karn 2000] which have been shown to be less than reliable [Ouzts and Duchowski 2012]. Velocity-based filters, or "saccade pickers", while more difficult to tune, are generally more reliable and readily available.

In the case of PIN entry, we chose velocity-based filtering and smoothing of the raw gaze data stream with a Butterworth filter. Treating x_i and y_i independently, smoothing or differentiating (to order s) is achieved by convolving $2p+1$ inputs with filter $h_i^{t,s}$ and $2q+1$ (previous) outputs $\bar{x}_i$ or $\bar{y}_i$ with filter $g_i^{t,s}$ at midpoint i [Hollos and Hollos 2014]: $\bar{x}_n^s(t) = 1/(\Delta t^s)\left(\sum_{i=-p}^{p} h_i^{t,s} x_{n-i} - \sum_{i=-q}^{q} g_i^{t,s} \bar{x}_{n-i}\right)$ and similarly for y_i and $\bar{y}_i$, where n and s denote the polynomial fit to the data and its derivative order, respectively [Gorry 1990]. Based on evaluation of calibration data, we chose a 2^{nd} order Butterworth filter to smooth the raw gaze data with sampling and cutoff frequencies of 60 and 2.35 Hz, respectively.

For the rotary interface, only smoothing with the Butterworth filter is required. The PIN detection algorithm operates directly on streamed data output by the Butterworth filter. For the keypad interface, streaming data output by the Butterworth filter is further convolved with a 2^{nd} degree Savitzky-Golay differential filter acting as a velocity-based saccade detector [Nyström and Holmqvist 2010]. We use a 7-tap (width) filter with velocity threshold set to $\pm 20°/\text{s}$. Resultant fixations with a duration less than 250 ms are culled from the data. The PIN detection algorithm operates on the fixation stream output by the Savitzky-Golay filter.

Note that in the present empirical analysis all data is processed offline. However, although the filtering operations do incur a slight processing delay, they are usable in real-time settings as well, as demonstrated by Duchowski et al. [2014b]. It is important to emphasize that although our current implementation is offline, it is readily applicable to online, real-time scenarios. One might be gaze-based PIN entry while at an ATM (Automatic Teller Machine).

Rotary PIN Entry. The goal of the PIN detection algorithm is to assemble the PIN string from numeric pads visited by successive (smoothed) gaze points. The algorithm for determining the PIN from the streaming smoothed gaze point data is based on a state machine with states indicating presence of the gaze point in either the center (the home) pad (see Figure 1 noting the larger diameter of pad ◉), a valid numeric pad, or in an unknown state (e.g., gaze points that are outside of the interface rotary dial, as is possible with the first set of points in the data stream). The state changes when a boundary is crossed between a numeric or the home pad.

For each smoothed gaze point $\bar{g}_i = (\bar{x}_i, \bar{y}_i, t_i)$ as input, the algorithm finds the closest pad with its center to coordinates $(\bar{x}_i, \bar{y}_i)$ defined by the smallest Euclidean distance to the pad center (with tolerance $r = 30$ pixels). This produces an intermediate stream of characters that needs to be analyzed on-the-fly as the characters are streamed through. An example of such a stream produced by the user using the rotary interface in Figure 1 is shown in Figure 2 where the character `c` denotes the central (home) pad ◉.

Repeated symbols are produced by the stream of smoothed gaze points traversing through the pads. Notice, for example, the stream of points that pass close to pad ⑦ then continue on through ⑥. That segment of points produces the string segment `ccc77776666ccc` in the stream shown in Figure 2.

Given the detected transition between the home and a numeric pad, the algorithm relies on a voting scheme cast by the streamed gaze points as they encounter the numeric pads. The voting scheme is represented by an array of length 10, acting as a histogram where each bin collects votes for numeric pads encountered. The voting scheme is weighted such that the last vote cast is weighted by 1.25, 25% greater than the unit weight given to every earlier vote.

The weighted voting scheme effectively places greater importance on the numeric pad from which gaze *exits* on its return to the home pad. This allows the user to correct their input should they inadvertently start on an incorrect pad. In the case of input `ccc77776666ccc`, 4 votes are cast of numeral `7`, and 4 votes are cast for numeral `6` with last vote weighted by 1.25, revising the vote to 4 for `7`, and 5 for `6`. In this instance pad ⑥ is therefore selected.

When processing the stream of smoothed gaze points, a count is kept of the number of transitions to the home region (the `c` characters in the stream shown in Figure 2). PIN entry is terminated when the 5^{th} transition to the center (home) pad is detected.

Timing for the rotary interface is started when the first transition from the center pad to a numeric pad is detected. The ending timestamp is reset whenever a transition to the center from a numeric pad is detected. This makes possible entry of fewer than 4 digits, but no more than 4 since stream processing is halted upon detection of the 5^{th} transition into the center (which also terminates the timing).

Keypad PIN Entry. The algorithm for keypad PIN entry is somewhat simpler because the input stream of fixations is to a large extent reduced from the continuous smoothed gaze point stream received by the rotary interface in comparison.

The algorithm proceeds for each fixation $f_i = (\hat{x}_i, \hat{y}_i, t_i, d_i)$ as input, by finding the closest pad with its center to coordinates $(\hat{x}_i, \hat{y}_i)$ defined by the smallest Euclidean distance to the pad center (with tolerance $r = 90$ pixels). When ⊛ is detected, find the next sequential pad to which the subsequent fixation f_i is closest. Continue appending these pad symbols until the next fixation f_i is found to be within ⓘ#. Return the assembled PIN string. Note that it is possible to enter PIN codes that are longer than 5 digits long (this was evident prior to culling fixations shorter than 250 ms). An example of a PIN entered correctly is the stream of characters `9610`, as was produced by the user with the keypad interface shown in Figure 1. Timing for PIN entry is started when the first ⊛ is detected. Timing ends when # is detected. The difference between start and end timestamps gives the total time of PIN entry.

6 User Study

To test our rotary design for gaze-based PIN entry, we ran a user study. Users were asked to enter in 4-digit PIN codes while looking at either the keypad or rotary PIN entry designs. It is important to note that, in this experiment, analysis is done offline, making the PIN entry system not much different from looking at a still image, from the user's perspective. No real-time feedback on gaze position could be given, as doing so would render PIN entry vulnerable to shoulder surfing attack.

Experimental Design. The experiment followed a repeated-measures factorial design with the user interface as the sole independent variable at two levels (rotary vs. keypad). Dependent measures included password entry time, PIN entry error, and percentage of correct entries completed by the user. Analysis included trial as an additional fixed factor [Baron and Li 2007], using R, the language for statistical computing.

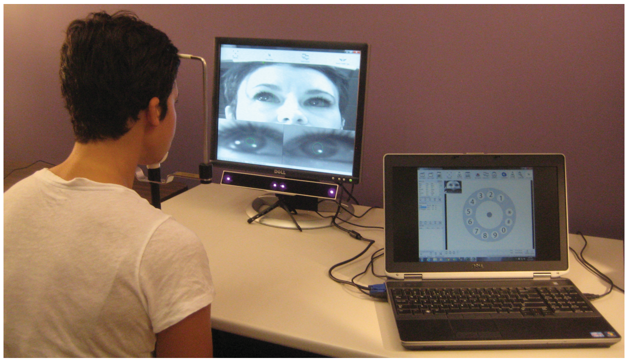

Figure 3: *Experimental setup: initial view of eye tracker's camera view helps the user position themselves; experimenter's laptop shows the stimulus display that would be next shown to the user.*

Stimulus. PIN entry targets, see Figure 1, were 1280×800 images rendered by Adobe Illustrator. Thirty (30) 4-digit PIN codes were generated randomly. The same PIN was used for each interface, with 30 trials conducted by each participant for each of them.

Participants. Thirty participants volunteered for the study (24 M, 6 F), with ages ranging from 18 to 44 (M=22, SD=5). Participants were either university faculty, staff, or students. Of the thirty, only one self-reported red-green-yellow color blindness, all others did not report any problems with their vision; 14 wore glasses, 3 wore contact lenses, and the rest reported no corrective lenses. Data from three participants (2 F, 1 M) was excluded from the analysis due to poor calibration or other distractions (one of the excluded female participants was distracted by a loud conversation next door).

Apparatus. The visual stimuli (image of a target PIN code and keypad or rotary PIN entry interfaces) were displayed on a 19″ Dell monitor (model 1905FP, 60 Hz display refresh rate), with screen resolution set to 1280×1024 pixels. Gaze data was collected by a Gazepoint GP3 eye tracker. The tracker operates at a 60 Hz sampling rate, with 0.5°–1.0° visual angle accuracy, according to the manufacturer. The eye tracker was connected to a Dell Latitude laptop (model E6530 with 4 G RAM, running the 32-bit Win7 operating system). The tracker was controlled by the Gazepoint Analysis software (v2.2.0). Gazepoint's Control software, used to calibrate the tracker, was configured to perform calibration on the Dell 1950FP display, connected to the laptop as an external monitor. This allowed the experimenter to monitor eye movements as the participant performed their task (see Figure 3). Each participant rested their chin on a height-adjustable chin rest.

Procedure. The experiment was conducted in a lab (see Figure 3). Upon entry, each participant was greeted and given an informational letter as part of the experimental protocol approved by our Institutional Review Board. The participant was then given a pretest questionnaire containing demographic information. The participant was then seated in a sturdy (wooden) chair that did not allow rolling or elevation adjustment, and was asked to place their chin on a chin-rest which was adjusted to a comfortable height.

The experimental procedure consisted of two main trial sequences, one for each of the rotary or keypad user interfaces. Each trial sequence started with a 9-point calibration sequence performed by the eye tracker software. The calibration was redone if considered inadequate by the experimenter. Each trial sequence then began with a 30 second training trial, where an image of the given user interface was shown for 30 seconds, accompanied by verbal instructions on how to enter pin ①②③④.

The Gazepoint Analysis software used did not allow stimulus advancement by any other means other than timing. Hence, instead of advancing by mouse button press, we were forced to display each trial for a finite amount of time. Note that this hampers data analysis somewhat since code used to determine the PIN code entered by the user must search for the end of PIN code entry, i.e., for the rotary interface, we did this by using only the first four pad entries detected. Our timing results therefore reflect the time needed by the user to enter in the first four numerals. On the keypad interface, end of input is indicated by the # symbol.

For each of the rotary interface trials, following training, an image showing the target PIN was displayed for 10 seconds, followed by an image of the rotary interface, also displayed for 10 seconds. The keypad interface sequence proceeded similarly, except that each of the PIN images and keypad interface images was shown for 15 seconds instead of 10 seconds. The timing rationale was based on anecdotal observations made during a short pilot study conducted with 4 individuals, wherein it was noted that entering the PIN took longer with the keypad interface. Therefore, a slightly longer stimulus viewing time was set for each of the keypad trials.

Each participant completed 30 trials with each interface (yielding a total of 60 trials). The order of trial sequences (rotary or keypad) was balanced. A grand total of 1,620 trials was collected for 27 participants (3 data sets were removed due to poor calibration).

Following the experiment, each participant filled out a post-experiment preference questionnaire with space for responses to open-ended questions.

7 Results

On average, users entered in PIN codes at a rate of correctness of 71.16% using the rotary interface and 64.20% using the keypad interface. This aggregate rate of correctness depends on computation of individual PIN entry error. PIN entry error for specific instance of PIN code is computed via the normalized Levenshtein distance [Waterman 1989]. The Levenshtein distance (or similarity) is computed by an optimization algorithm that builds an $n \times m$ array (where n and m are the lengths of the two PIN strings) and finds the minimum cost to transform one string into the other via insertions, deletions, or substitutions. For example, the PIN ④③③⑥ entered as ④③③⑧ incurs a one-character substitution cost, giving an error of 0.25 when normalized to the length of the PIN. Normalization ensures that no error is larger than 1.0.

Target PINs are inherently 4 digits long and, for the rotary interface, the first 4 digits entered by the user are taken as PIN entries. In general, however, if no such PIN length checking is performed on user entries, due to potential errors, user-entered PINs could be shorter or longer. In such cases, normalization would be performed to the longer of the two PINs. Analyses of mean errors for each of the user interfaces, along with mean entry speed and effect of trial are given below.

Note that out of 1,620 total trials, 18 resulted in no PIN entry, with 0 time recorded. This occurred 3 times for the rotary interface and 15 times for the keypad. For both interfaces, a possible observed reason for not entering a PIN was forgetting of the code to enter. When this was observed for the rotary interface, users tended to simply stare at the center. Other observed problems with the keypad interface was users forgetting to dwell on the * start or # end pads, or not dwelling long enough (minimum of 250 ms) on them.

Table 2: *Modal responses to subjective post-experiment questions, marked on a 7-point Likert scale with 1 indicating strong disagreement.*

#	Question	mode
1.	The eye tracking device distracted me and hindered my ability to perform my tasks.	1
2.	I preferred the keypad interface to the rotary style interface.	2
3.	I understood what was expected of me in each task.	7

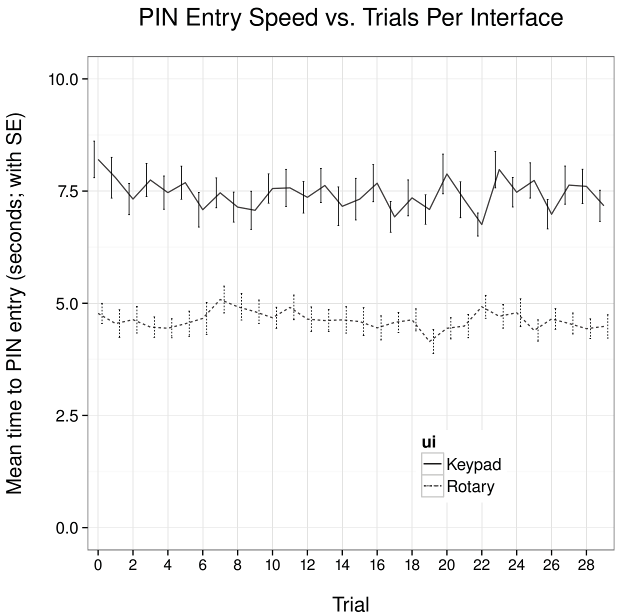

Figure 4: *Effect of trial on entry speed.*

These 18 0-time trials are included in the analysis below (favoring the keypad in terms of speed).

Speed. To determine the effect of interface on PIN entry speed, a two-way repeated-measures ANOVA (analysis of variance) was performed with interface and trial as fixed factors. Both interface and trial were considered as within-subjects factors. Analysis reveals the effect of user interface on speed as significant ($F(1,26) = 61.20, p < 0.01$), with the rotary interface affording faster mean PIN entry (M = 4.62 seconds, SD = 1.37) than the keypad (M = 7.44 seconds, SD = 1.97). The effect of trial on speed is not significant ($F(29,754) = 1.46, p = 0.06$, n.s.). Interaction between interface and trial, however, is significant ($F(29,754) = 1.80, p < 0.01$).

Accuracy. To determine the effect of user interface on PIN entry error (accuracy), a two-way repeated-measures ANOVA was performed with user interface and trial as the fixed factors. Both user interface and trial were considered as within-subjects factors. Analysis fails to show a significant effect of interface on error ($F(1,26) = 1.16, p = 0.29$, n.s.). Mean PIN entry errors are similar, with the rotary incurring error (M = 0.12, SD = 0.21) that is slightly smaller than the error incurred by the keypad (M = 0.14, SD = 0.22) but not significantly so. The effect of trial on error is significant, however ($F(29,754) = 2.00, p < 0.01$). Interaction between interface and trial is marginally significant ($F(29,754) = 1.61, p < 0.05$).

Trials. Examining the effect of trials more closely, a one-way ANOVA was conducted separately for the timing data collected per interface. This time, only the trial was considered as the fixed factor, with trial again considered as a within-subjects factor, see Figure 4. Analysis shows that the effect of trial on PIN entry time was significant, but only for the keypad interface ($F(29,754) = 1.96, p < 0.01$). The analysis failed to show a significant effect of trial on entry time for the rotary interface ($F(29,754) = 1.09, p = 0.34$, n.s.).

Table 3: *Sample responses to open-ended questions regarding differences between the keypad and rotary interfaces.*

1:	*The rotary is quicker and slightly more intuitive.*
2:	*This was amazing!*
3:	*It was lots of fun, and I highly enjoyed it.*
4:	*The rotary felt faster to use because I did not have to dwell on the numbers like the keypad.*
5:	*My eyes were set to center of the screen making it easier on the rotary whereas the keypad required me to look down-left first.*
6:	*The keypad felt more natural—it was easier and faster.*
7:	*Having to go back to the circle on the rotary seemed like it made me go slower but it's more efficient in case you mess up—you can change the number before you enter it.*
8:	*The rotary was easier to navigate with my eyes to the different numbers.*
9:	*This is certainly a cool idea.*

Of particular interest is the potential of a learning effect, i.e., differences in entry time between the first and subsequent trials, in particular the last. For the keypad, a pairwise t-test of trial entry times (with no correction) shows significant differences between the 1^{st} trial and the 7^{th}, the 15^{th}, and the 23^{nd} ($p < 0.01$), with an almost steadily decreasing trend entry time (8.21, 7.08, 7.16, 6.76 seconds, respectively). A similar pairwise t-test of trial entry times for the rotary interface shows no significant differences (at the $p < 0.01$ confidence level) between the 1^{st} and any subsequent trials.

Subjective Responses. Modal responses to the subjective post-experiment questionnaire show strong preference for the rotary interface, while indicating understanding of the task, and marking strong unobtrusivity of the eye tracking device (see Table 2). Additional open-ended questions asked about any other factors, other than visual, that differed between the interfaces or influenced perceived performance, and also solicited any additional feedback users may have had. With the exception of one user who seemed to find the keypad more natural, most users appeared to prefer the rotary interface, acknowledging its advantages in exploiting the center-bias and not having to dwell. One respondent also appreciated the consequence of the rotary's weighted voting scheme by noting that it was possible to change the numeric pad before returning to the center (home) pad (see Table 3).

8 Discussion

Subjective responses show a strong preference for the rotary interface. Users appeared to readily understand its apparent advantages regarding center-bias and equidistant targets. Users also noted its intuitive design. Other circular numeric pad arrangements may also be recognizable, e.g., the clock face comes to mind, which would allow users to potentially remember PIN codes as some memorable hour configurations, e.g., `11:30` (time for lunch!).

Error rates support the notion of operational equivalence: on average, users were able to enter PIN codes that differed no more than by 1 character symbol. Therefore, neither interface inherently promoted entry of PIN codes with greater errors (e.g., two or more characters that were misaligned, incorrect, or extraneous). Unexpectedly, the keypad allowed these small errors at a slightly larger frequency (35.80% vs. the rotary interface's 28.84%). The rotary interface's weighted voting scheme allowed corrections, which was duly noted by at least one participant and likely utilized by others as well. One could argue that a real-time implementation of the keypad interface that graphically provides dwell time expiration feedback (as is commonly done in eye typing keyboard character selection [Majaranta and Räihä 2002]) may level the playing field in terms of total errors committed, but any type of visual feedback such as this would allow shoulder-surfing attacks, defeating the purpose of gaze-based PIN entry. It is more likely that the circular arrangement and emphasis on selection of the numeric pad that is glanced upon exit of the numeric disk was intuitively understood by users allowing them to correct pad selection on-the-fly.

The observed speed advantage of the rotary interface corroborates previous work. Similarly, no learning effect was observed for the rotary interface. Initially we had thought that a fair amount of training would be needed for users to understand its usage. Apparently 30 seconds worth of training is sufficient to prevent large variation of entry time over the course of subsequent trials. In contrast, the same cannot be said of the keypad interface, as users also benefited from a 30 second training period prior to its use. The significant effect of learning on PIN entry with the keypad interface is perhaps likely due to the requirement of starting * and ending # symbols.

9 Limitations

Because data was processed offline and the eye tracking software did not allow for notification of PIN entry completion (e.g., by mouse or keyboard), there was no way to determine (online) when PIN entry was completed. We therefore had to treat the experiment as machine-paced where trials were limited to 10 (rotary) or 15 seconds (keypad). However, in the final analysis this decision had no impact since PIN entry time was well below threshold in each case and we found no aborted PIN entry trials due to timer expiration.

Clearly, future validation of the interface should include real-time PIN entry, with some form of feedback provided to the user upon detected completion of (correct or incorrect) PIN code. We should point out that lacking any sort of feedback our entry system presented users with a basic timing task and admittedly a rather poor form of interface from an interactive standpoint. However, we point out that even under these crude conditions, users were still able to grasp the intended interaction (that a real-time system would afford) and they were able to complete the PIN entry tasks quite well (erring less than 1 character per PIN code on average, with an overall success rate of 71% for the rotary dial—we attribute a large proportion of the error to memory issues and not the interface itself). We could only expect that giving users feedback with better task (PIN code) memorability (e.g., fewer trials) would improve the error rate further. We are therefore confident that the novelty of our interaction, namely one based on boundary crossing and not dwell time, is an improvement over a dwell-based grid layout.

10 Conclusion

We have presented the design and evaluation of a novel rotary-style interface for PIN code entry with gaze. The interface eliminates the need for eye movement fixation classification along with dwell time that is often employed in response to the Midas Touch problem. The rotary interface operates by detecting gaze boundary crossings between its central (home) region and surrounding numeric pads displayed on a disk encircling the central region. The numeric pads' equidistant placement relative to the center affords faster PIN entry than a spatial grid adopted by the common keypad layout. The rotary's speed advantage was shown empirically through a user study.

Empirical results show that individual PIN code entry error did not, on average, differ significantly between the rotary and keypad interfaces, although the rotary's overall error rate was lower (28.84% vs. 35.80% for the keypad). Users appeared to grasp the operational requirements of the rotary interface immediately, with no evident learning effect on PIN entry speed. The keypad interface, on the other hand, showed a significant decrease in entry speed over time suggesting a steeper learning curve.

References

BARON, J., AND LI, Y., 2007. Notes on the use of R for psychology experiments and questionnaires. Online Notes, 09 November. URL: http://www.psych.upenn.edu/~baron/rpsych/rpsych.html (last accessed Sep. 2014).

BEE, N., AND ANDRÉ, E. 2008. Writing with Your Eye: A Dwell Time Free Writing System Adapted to the Nature of Human Eye Gaze. In *Proceedings of the 4th IEEE Tutorial and Research Workshop on Perception and Interactive Technologies for Speech-Based Systems: Perception in Multimodal Dialogue Systems*, Springer-Verlag, Berlin, Heidelberg, PIT '08, 111–122.

BIDDLE, R., CHIASSON, S., AND VAN OORSCHOT, P. C. 2012. Graphical Passwords: Learning from the First Twelve Years. *ACM Computing Surveys 44*, 4 (Aug.), 1–41.

BULLING, A., ALT, F., AND SCHMIDT, A. 2012. Increasing the Security of Gaze-based Cued-recall Graphical Passwords Using Saliency Masks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, CHI '12, 3011–3020.

CARPENTER, R. H. S. 1977. *Movements of the Eyes*. Pion Limited, London.

DE LUCA, A., WEISS, R., AND DREWES, H. 2007. Evaluation of Eye-gaze Interaction Methods for Security Enhanced PIN-entry. In *Proceedings of the 19th Australasian Conference on Computer-Human Interaction: Entertaining User Interfaces*, ACM, New York, NY, OZCHI '07, 199–202.

DE LUCA, A., DENZEL, M., AND HUSSMANN, H. 2009. Look into my Eyes! Can you guess my Password? In *Symposium on Usable Privacy and Security (SOUPS)*.

DUCHOWSKI, A. T., BABU, S. V., BERTRAND, J., AND KREJTZ, K. 2014. Gaze Analytics Pipeline for Unity 3D Integration: Signal Filtering and Analysis. In *Proceedings of the 2nd International Workshop on Eye Tracking for Spatial Research (ET4S)*.

DUCHOWSKI, A. T., HOUSE, D. H., GESTRING, J., WANG, R. I., KREJTZ, K., KREJTZ, I., MANTIUK, R., AND BAZYLUK, B. 2014. Reducing Visual Discomfort of 3D Stereoscopic Displays with Gaze-contingent Depth-of-field. In *Proceedings of the ACM Symposium on Applied Perception*, ACM, New York, NY, SAP '14, 39–46.

GORRY, P. A. 1990. General Least-Squares Smoothing and Differentiation by the Convolution (Savitzky-Golay) Method. *Analytical Chemistry 62*, 6, 570–573.

HOANCA, B., AND MOCK, K. 2006. Secure graphical password system for high traffic public areas. In *Proceedings of the 2006 Symposium on Eye Tracking Research & Applications*, ACM, New York, NY, ETRA '06, 35–35.

HOLLOS, S., AND HOLLOS, J. R. 2014. *Recursive Digital Filters: A Concise Guide*. Exstrom Laboratories, LLC, Longmont, CO, April. ISBN: 9781887187244 (ebook), URL: http://www.abrazol.com/books/filter1/ (last accessed Sep. 2014).

HOLMQVIST, K., NYSTRÖM, M., ANDERSSON, R., DEWHURST, R., JARODZKA, H., AND VAN DE WEIJER, J. 2011. *Eye Tracking: A Comprehensive Guide to Methods and Measures*. Oxford University Press, Oxford, UK.

HUCKAUF, A., AND URBINA, M. 2007. Gazing with pEYE: New Concepts in Eye Typing. In *Proceedings of the 4th Symposium on Applied Perception in Graphics and Visualization*, ACM, New York, NY, APGV '07, 141–141.

HUCKAUF, A., AND URBINA, M. H. 2008. Gazing with pEYEs: Towards a Universal Input for Various Applications. In *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications*, ACM, New York, NY, ETRA '08, 51–54.

ISOKOSKI, P. 2000. Text input methods for eye trackers using off-screen targets. In *Proceedings of the 2000 Symposium on Eye Tracking Research & Applications*, ACM, New York, NY, ETRA '00, 15–21.

JACOB, R. J. 1990. What You Look at is What You Get: Eye Movement-Based Interaction Techniques. In *Human Factors in Computing Systems: CHI '90 Conference Proceedings*, ACM Press, 11–18.

KARN, K. S. 2000. "Saccade Pickers" vs. "Fixation Pickers": The Effect of Eye Tracker Choice on Research Findings (Panel Discussion). In *ETRA '00: Proceedings of the 2004 Symposium on Eye Tracking Research & Applications*, ACM, 87–88.

KUMAR, M., GARFINKEL, T., BONEH, D., AND WINOGRAD, T. 2007. Reducing Shoulder-surfing by Using Gaze-based Password Entry. In *Symposium on Usable Privacy and Security (SOUPS)*.

LASHKARI, A. H., ZAKARIA, O. B., FARMAND, S., AND SALEH, R. 2009. Shoulder Surfing Attack in Graphical Password Authentication. *International Journal of Computer Science and Information Security 6*, 2.

MADATHIL, K. C., AND GREENSTEIN, J. S. 2011. Synchronous Remote Usability Testing – A New Approach Facilitated by Virtual Worlds. In *CHI '11: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM Press, New York.

MAJARANTA, P., AND RÄIHÄ, K.-J. 2002. Twenty Years of Eye Typing: Systems and Design Issues. In *Eye Tracking Research & Applications (ETRA) Symposium*, ACM.

NYSTRÖM, M., AND HOLMQVIST, K. 2010. An adaptive algorithm for fixation, saccade, and glissade detection in eyetracking data. *Behaviour Research Methods 42*, 1, 188–204.

OUZTS, A. D., AND DUCHOWSKI, A. T. 2012. Comparison of Eye Movement Metrics Recorded at Different Sampling Rates. In *Proceedings of the 2012 Symposium on Eye-Tracking Research and Applications*, ACM, New York, NY, ETRA '12.

PATIDAR, P., RAGHUVANSHI, H., AND SARCAR, S. 2013. Quickpie: An interface for Fast and Accurate Eye Gaze based Text Entry. In *Proceedings of 11th Asia Pacific Conference on Computer-Human Interaction (APCHI): Student Design Consortium*.

PERLIN, K. 1998. Quikwriting: Continuous Stylus-based Text Entry. In *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology*, ACM, New York, NY, UIST '98, 215–216.

PETERSON, M. F., AND ECKSTEIN, M. P. 2012. Looking just below the eyes is optimal across face recognition tasks. In *Proceedings of the National Academy of Sciences of the United States of America*, vol. 109, E3314–23.

SREERAMAREDDY, L., FENG, J., AND SEARS, A. 2012. Preliminary Investigation of Gesture-Based Password: Integrating Additional User Behavioral Features. In *Symposium On Usable Privacy and Security*, 4–5.

TUISKU, O., MAJARANTA, P., ISOKOSKI, P., AND RÄIHÄ, K.-J. 2008. Now Dasher! Dash Away!: Longitudinal Study of Fast Text Entry by Eye Gaze. In *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications*, ACM, New York, NY, ETRA '08, 19–26.

URBINA, M. H., LORENZ, M., AND HUCKAUF, A. 2010. Pies with EYEs: The Limits of Hierarchical Pie Menus in Gaze Control. In *Proceedings of the 2010 Symposium on Eye-Tracking Research &; Applications*, ACM, New York, NY, ETRA '10, 93–96.

VAN OORSCHOT, P. C., AND THORPE, J. 2008. On Predictive Models and User-Drawn Graphical Passwords. *ACM Transactions on Information and System Security 10*, 4 (Jan.), 1–33.

WATERMAN, M. S. 1989. Sequence Alignments. In *Mathematical Methods for DNA Sequences*, M. S. Waterman, Ed. CRC Press, Inc., Boca Raton, FL, 53–92.

WEAVER, J., MOCK, K., AND HOANCA, B. 2011. Gaze-based password authentication through automatic clustering of gaze points. In *2011 IEEE International Conference on Systems, Man, and Cybernetics*, IEEE, 2749–2754.

WITMER, B. G., AND SINGER, M. J. 1998. Measuring Presence in Virtual Environments: A Presence Questionnaire. *Presence 7*, 3 (June), 225–240.

WOBBROCK, J. O., RUBINSTEIN, J., SAWYER, M. W., AND DUCHOWSKI, A. T. 2008. Longitudinal Evaluation of Discrete Consecutive Gaze Gestures for Text Entry. In *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications*, ACM, New York, NY, ETRA '08, 11–18.

WOOD, E., AND BULLING, A. 2014. Eyetab: Model-based gaze estimation on unmodified tablet computers. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ACM, New York, NY, ETRA '14, 207–210.

ZHAI, S., KRISTENSSON, P. O., GREINER, M., PENG, S., LIU, L., AND DUNNIGAN, A. 2009. ShapeWriter on the iPhone—From the Laboratory to the Real World. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07 Extended Abstracts)*, 2667–2670.