Designing a gaze gesture guiding system

William Delamare, Teng Han, Pourang Irani

Designing a Gaze Gesture Guiding System

William Delamare, Teng Han, Pourang Irani
University of Manitoba
Winnipeg, Canada
{delamarw, hanteng, irani}@cs.umanitoba.ca

ABSTRACT
We propose the concept of a guiding system specifically designed for semaphoric gaze gestures, *i.e.* gestures defining a vocabulary to trigger commands via the gaze modality. Our design exploration considers fundamental gaze gesture phases: *Exploration*, *Guidance*, and *Return*. A first experiment reveals that *Guidance* with dynamic elements moving along 2D paths is efficient and resistant to visual complexity. A second experiment reveals that a Rapid Serial Visual Presentation of command names during *Exploration* allows for more than 30% faster command retrievals than a standard visual search. To resume the task where the guide was triggered, labels moving from the outward extremity of 2D paths toward the guide center leads to efficient and accurate origin retrieval during the *Return* phase. We evaluate our resulting Gaze Gesture Guiding system, G3, for interacting with distant objects in an office environment using a head-mounted display. Users report positively on their experience with both semaphoric gaze gestures and G3.

Author Keywords
Gaze; Gesture; Guidance; Design; Eye input; RSVP; Guide.

ACM Classification Keywords
H.5.2. Information interfaces and presentation (e.g., HCI): User Interfaces – Interaction styles (e.g., commands, menus, forms, direct manipulation), training, help and documentation.

INTRODUCTION
Head-Mounted Displays (HMD) enable users to interact with both digital and augmented physical objects in mobile contexts. Researchers have proposed several hand-based interactions for HMD input, such as by detecting hand gestures via a front camera embedded on the device [9,32]. Although such direct manipulation is generally favorable, this approach can burden users in situations where hands are busy, or simply by causing fatigue after some use [21]. As an alternative, we explore inputs based on the gaze modality with HMDs [36]. However, instead of relying on the gaze as a pointing mechanism, we focus on gaze gestures.

MobileHCI '17, September 04-07, 2017, Vienna, Austria

ACM ISBN 978-1-4503-5075-4/17/09…$15.00
http://dx.doi.org/10.1145/3098279.3098561

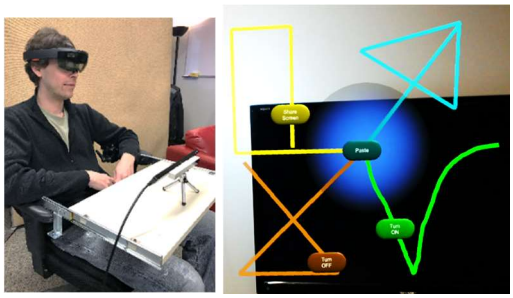


Figure 1: Example scenario using G3. The user is looking at a distant physical object (TV screen) through a see-through Head-Mounted Display. G3 allows the user to discover available commands and their associated gestures. Users can perform a gaze gesture by following the desired label moving along its corresponding 2D path.

Gaze gestures, the ability to trigger a command via eye movement patterns [24], do not require additional dwell time that is common with gaze input to avoid unwanted Midas 'touch' events [26]. Gaze gestures are also conceptually resistant to accuracy limitations since only position variations are used in the gesture recognition [14], which allows usage without a per-user calibration process [48].

The most widely used type of gestures are semaphoric (or symbolic) gestures, which define a vocabulary to map gestures to system commands (*e.g.*, drawing a circle to start recording, tracing a 'C' to close a menu, etc.) [1,29]. However, despite the practical advantages of semaphoric gestures [29] (*e.g.*, explicit command/gesture mapping), novice users still need to discover the available commands and their associated gestures. Thus, researchers and practitioners have identified the need to carefully design a guiding process [30]. Over three dozen gesture guiding systems exist [12] for 2D mouse gestures [5], touch gestures [17], 3D hand gestures [13], or whole body gestures [3]. None, however, are currently designed for gaze gestures, i.e. by considering the dual nature of the eyes during gaze interaction – capturing and transmitting information.

In this paper, we introduce a novel Gaze Gesture Guiding system, G3, and illustrate its use *via* a HMD to interact with distant digital and augmented physical objects (Figure 1). G3 allows users to efficiently trigger commands by performing gestures with gaze. In a step-wise manner, we explore design factors by considering the meaningful events unique to gaze. Through an informal evaluation, users report on the ease of using G3.

Figure 2: G3 walkthrough. (a) The first (dark) gesture appears with the corresponding command label at the center of the guide. Arrow '1' shows the path the label will use to shift toward the end of the gesture. Arrow '2' shows the path the label will follow to guide the gaze toward the guide origin. (b) The second (yellow) gesture appears. The first label is moving toward the end of the corresponding 2D path (arrow '1'). (c) The fourth (blue) gesture appears. The two first gestures (dark and yellow) are moving back along their 2D path. The third (red) label is moving toward the end of the red 2D path. (Arrows are for illustration purpose only.)

Our contributions are threefold. First, we propose the concept and the design of a gaze gesture guiding system. Second, we experimentally explore design options based on the fundamental gaze gesture phases (see below). Lastly, we evaluate our resulting design, in an office environment scenario to illustrate the complete integration of G3 with mobile tasks.

PHASES IN GAZE GESTURE INPUT

We use the scenario depicted by Figure 1 to illustrate the structure of the guiding system into phases. A user wants to share pictures from a HMD onto a TV screen. The user explores available commands and executes a desired gesture with gaze. Afterwards, users would aim to return their gaze onto the TV screen to focus on the shared pictures. This typical scenario reveals the following three phases:

- With a gesture guiding system, there is an initial *Exploration* phase during which users must look for a target command and its corresponding gesture. With other modalities such as hand motion, the visual search does not interfere with the gesture execution. In contrast, with gaze gestures, eye motion for visual search overlaps with eye motion used for gesture execution. Any design for a gaze gesture guiding system needs to carefully consider this phase to avoid unwanted or spurious input.
- It is not clear how to efficiently guide eye movement during the *Guidance* phase to facilitate gesture execution. We are not aware of any previous work stating the benefits and drawbacks of candidate visual elements while guiding the gaze along gesture shapes for interaction purpose. For instance, is it better to represent the complete gesture shape with a 2D path and/or provide dynamic visual elements to help the eyes draw a gesture? As a result, attention is needed to design the *Guidance* phase.
- After performing the gaze gesture, the user's eyes easily lost their initial focus point. This can be inconvenient if users need to view any feedback taking place at the origin of the gesture [44] or want to resume their previous task. The gaze may need to *Return* to the original location. While this may not be necessary in every application, we consider the *Return* phase in our design exploration for comprehensive design.

Our resulting design, G3, functions following the three phases: *Exploration*, *Guidance*, and *Return* phases. Command names are revealed using a Rapid Serial Visual Presentation (RSVP) [27,42] in the center of a radial guide (Figure 2, a), initiating the *Exploration* phase. Command names are then quickly shifted to the outward extremity of the 2D path representing the corresponding gesture (Figure 2, b-c). The command name labels play the role of a dynamic target for users to follow and smoothly move back to the center for guiding users during the execution of the gesture (Figure 2, c). This was designed to support the *Guidance* phase. Once the gesture is completed, users' gaze lands on the original point of interest, thus supporting the *Return* phase.

RELATED WORK

Our work builds on prior research in gaze gestures and gesture guiding systems.

Gaze Gestures

We adopt the distinction between eye *movements* and gaze *gestures* defined by Møllenbach *et al.* [33]. Eye movements refer to the actual pupils' motion, while gaze gestures refer to the path produced by this motion. Two types of eye movements have been identified to interpret gaze gestures: saccadic and smooth pursuit eye movements.

Gestures Using Saccadic Eye Movements

Saccades are fast (between 30ms and 120ms) ballistic eye movements (between 1° and 40°) [26]. Intentional saccade movements have been used to execute (multi)stroke gestures, which can be represented as sequences of straight lines [41] separated by fixation points [25]. Fixations happen when the gaze is essentially stationary. Stroke gestures have been used for desktop applications. such as [14] video games [22,24], selecting targets [34] and have even allowed for text-entry [6,23,49]. Researchers have shown the value of stroke gestures on a wide variety of mobile devices, including smartphones [15,28,41] and smartwatches [19,20] or while interacting with augmented physical objects [4]. Users are able to select color targets with one stroke gestures [15], as well as more complex three-stroke gestures [41]. On smartwatches, researchers have used iconic and textual prompts to indicate the direction to look at for triggering additional actions [20].

Gestures Using Smooth Pursuit Eye Movements

Pursuits are smooth eye movements that can only happen when looking at moving objects [48]. Pursuit movements have been introduced to interact with on-screen visual content without any calibration [48]. Users were able to

select a target by visually following its motion along straight and/or circular paths. AmbiGaze allows users to interact with augmented physical objects via pursuit selection of digital or physical targets [47]. Orbits [16] is an example where pursuit movements can be used with smaller smartwatch displays [16], by attaching a visual target to a graphical widget. As with other modalities such as the hand [8], users follow the target orbiting the widget instead of directly selecting a widget to trigger a command.

Previous work dealing with pursuit movements did not focus on semaphoric gaze gestures, but on dynamic target selection. Selection via smooth pursuit is based on the correlation between targets' and gaze's motion, ignoring the semantic value of the executed shapes. In our work, we focus on the end-result shapes drawn during the gesture execution. Indeed, pursuit-based selection would require the system to always be displayed, without possibility for executing a gaze gesture without moving targets. Thus, we simply consider pursuit movements as a potential candidate for drawing 2D shapes while being guided, not as a dynamic target selection mechanism. This also allows us to extend the concept of gaze gestures using "typically strokes" [14] to any shape, including both straight and curved lines.

Gesture Guiding Systems

Gaze gesture studies often use a crib-sheet as a guiding system, either displayed on the screen [22,24] or via a piece of paper attached to the device [34,41]. But a crib-sheet is often used as a limited baseline during guidance studies [2,5,7,18,30,39].

It is beyond the scope of this work to describe all existing guiding systems proposed for other modalities in the literature [12]. Instead, we focus our description on design considerations related to the meaningful phases of *Exploration* and *Guidance*. The *Return* phase has not yet been explored as no guiding system is specifically designed for gaze gestures.

Exploration Phase

The *Exploration* phase is the necessary step for users to find a command label. A guiding system can reveal all gestures, a subset of gestures, or only one gesture [5]. For instance, radial guides such as Marking Menus [30] and OctoPocus [5] reveal all gestures in the gesture set: command names and respectively directions and 2D paths. Augmented Letters [40] reveals only the subset of gestures corresponding to commands starting with the traced letter. Finally, GestureBar [7] only shows the gesture representing the desired command selected in a menu bar. Our design exploration draws inspiration from these earlier non-gaze approaches.

Guidance Phase

The *Guidance* phase is the step during which the guiding system is actually helping the gaze perform the gesture shape. We consider two features related to the *Guidance* phase: the portion of gesture displayed [5] and the pace tolerance for executing the gestures [12].

Portion of Gesture: The guiding system can display the direction of the gesture, a portion of the gesture or the complete path [5]. For instance, the Arrow condition of LightGuide projects only the direction of the path onto the hand [43]. A Hierarchical Marking Menu displays only portions of the gestures to execute, corresponding to the current menu-level in the hierarchy [31]. On the contrary, a Marking Menu displays the complete portion of the stroke to execute [30]. Commonly used crib-sheets also display the complete gesture path [2,5,7,18,30,39].

Pace Tolerance: The speed for executing a gesture can be user-imposed or system-imposed [12]. For instance, the 3D Follow Arrow condition of LightGuide projects a 3D arrow onto the hand [43]. The projection of the arrow moves to let users follow the direction indicated by the arrow. However, several guiding systems let the user chose the pace for the execution. These include OctoPocus [5], which lets users draw gestures by following 2D paths; ShadowGuides [17], which lets users perform whole-hand gestures on a tabletop; and Arpège [18], which allows users to choose the speed for performing finger chord gestures.

To our knowledge, there are no experimental studies using the gaze modality to perform semaphoric gaze gestures – with both straight and curved lines. In addition, no previous gesture guiding system considered peculiarities of the gaze modality. We explore thee, and in particular address concerns arising from the unavoidable *Exploration*, *Guidance* and *Return* phases of the gaze gesture interaction.

EXPERIMENT 1: GUIDANCE PHASE

In our first experiment, we aim to define the visual elements that work best during the *Guidance* phase, which in turn influence the design of the *Exploration* phase. We explore the benefits and drawbacks of combining two basic visual alternatives: static 2D paths and dynamic moving targets.

Guidance Phase Designs

We do not restrict our guiding system to the use of one particular eye movement type, i.e. saccade or pursuit. Instead, we aim to provide a guiding system for gaze gestures using both straight and curved lines, to not constrain the gesture set design. We focus our design exploration on combinations of dynamic and static visual elements and portion of gesture displayed.

- *Moving Targets*: As in prior work using pursuit movements (e.g. Orbits [16]), the guiding system only shows moving targets (Figure 3, a). Targets are colored circles moving along invisible paths.
- *2D Paths*: As in prior work on gesture guidance with non-gaze modalities, the guide can display gestures as colored 2D paths [2,5,17] (Figure 3, b).
- *Combined Moving Target and 2D Path*: The guide can show gestures via targets moving along displayed 2D paths (Figure 3, c).
- *Segments*: The guide can display a combination of moving targets and moving path segments (Figure 3, d).

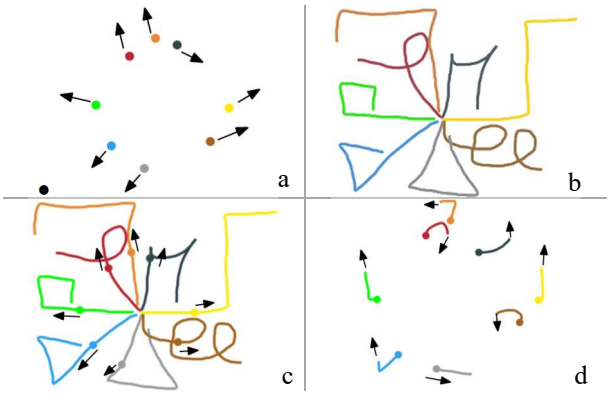


Figure 3: Guidance phase designs considered for Experiment 1. a) Moving targets only. b) 2D paths only. c) Combination of 2D paths and moving targets. d) Moving targets and moving portions. Arrows are for motion illustration only.

Dynamic visual elements will lead to pursuit eye movements. This might allow users to perform curved shapes better and more comfortably than using saccades [16]. However, because of their dynamicity, these visual elements can (1) bring confusion [16,48], (2) overwhelm users [16], and (3) cause problems in cases of paths crossing each other [10,16]. Displaying the complete gesture might lead to better anticipating eye movements than simple circle targets [16,38,48], but can also increase the overall visual complexity of the scene [12]. We conducted the first experiment to evaluate the four design solutions.

Participants and Apparatus

16 participants (7 females), ages 19 to 35 (M=25, SD=4.13), volunteered for this experiment. Six participants had glasses and one participant had contact lenses. None of them had previous experience with gaze input.

We carried out our experiments on a desktop screen. This setup allowed us to focus on the target designs without considering the context and hardware limitation. Participants were sitting ~70cm in front of a 24" desktop screen (1920×1080 resolution). We used the Eye Tribe eye tracker [11], with 60Hz sampling rate and gaze estimation error between 0.5° and 1° according to the manufacturer. The eye tracker was placed between the participant and the screen (Figure 4, left). The four guiding systems were implemented in C# with the Unity3D 5.3 game engine and ran on a 3.6 GHz Intel Core i7 computer. To calculate the recognition rate, we used a modified $1 recognizer [50]: we removed the rotation invariance to consider gesture orientation.

Each gesture class included only one instance of a unique gesture (i.e., without repetition). The combination of a first-entry market eye tracker and the $1 recognizer might not lead to the best context for gaze gestures [11,50]. However, (i) we are not evaluating the tracking device or the recognition algorithm, and (ii), this setup is sufficient for comparing our designs.

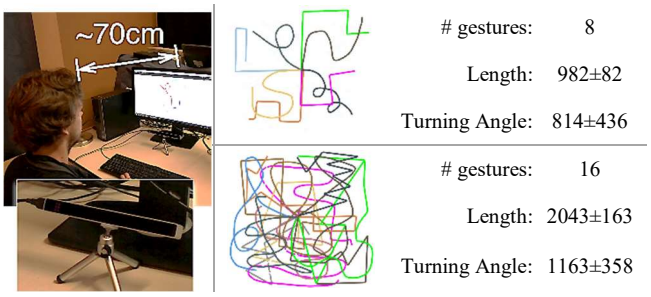


Figure 4: Left: Experimental setup. Right: Simple (top) and complex (bottom) gesture sets (illustrated with 2D paths). For each gesture set, we report the number of gestures, the length (px) and the total turning angle (°) (M ± SD) [45]**.**

Procedure

Even though not necessary for actual use of the guidance designs, we performed a 9-points per-user calibration. This calibration step allowed us (i) to explain the concept of gaze tracking, and (ii) to check that the hardware was able to track participants' eyes.

The experiment lasted around 1.5 hours per participant, followed by a questionnaire for qualitative data. Participants could have a break between trials to avoid fatigue, and were instructed to always execute the green-colored gesture (Figure 4, right). This removed the need to visually search for command labels (focus of the second study), yet added visual complexity with multiple paths.

Each trial began with pressing the 'space' bar on a keyboard. Once the gestures showed up, dynamic elements started to move. Participants could get familiar with the scene as long as they desired to (i.e., acquaintance time), but were instructed to complete the task as quickly and accurately as possible. When ready to execute the gesture, the participants pressed the 'space' bar again to re-initialize the position of the moving elements. Once the gesture execution was completed, a last press on the 'space' bar ended the recording and triggered the recognition process. We provided visual feedback regarding the gesture recognition result. We did not provide feedback regarding the gaze position in order to avoid any confusion due to the potential offset between the gaze position and a cursor [26].

2D paths were 27.5mm wide lines. Moving targets were circles with a 77mm diameter. Moving targets moved at a speed of 11cm.s^{-1} and returned at the origin of the gesture once they reached the end. Segments had a 3.74cm length.

Experimental Design

We consider the following experimental factors:

- Guidance phase ***Design***: We compare the execution of gaze gestures guided by *moving targets*, *2D paths*, *combined* moving targets and 2D paths, and moving *segments*.
- Gesture Set ***Complexity***: We created two gesture sets of different complexity: a *simple* gesture set and a *complex* one (Figure 4, right). To increase the gesture set

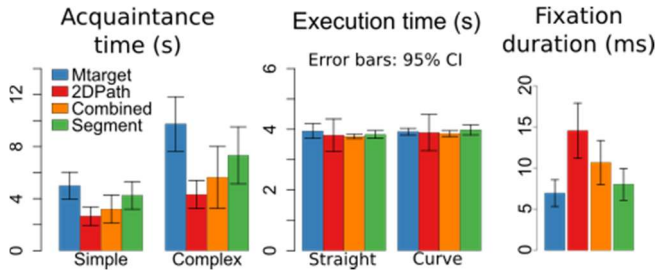


Figure 5: Acquaintance time (left), execution time (center), and fixation duration (right) for all designs.

complexity, we considered: (i) the number, (ii) the length [46], and (iii) the total turning angle of the gestures [45].

- Gesture ***Type***: Since some designs will lead to saccade movements and other to pursuit movements, we want to compare their performance regarding the type of gesture to execute: *straight* or *curved* lines. We included half of each gesture type in both gesture sets.

We used a repeated-measure within-participants design. The independent variables were the *Guidance* phase design (Mtarget, 2DPath, Combined, Portion), the gesture set complexity (simple, difficult) and the gesture type (straight, curve). The ordering of design and complexity was counter-balanced across participants using a Latin-square design. The gesture type presentation was random.

The experiment was divided into four sections: one for each *Guidance* phase design. Each section was divided into three sequences of trials: one sequence using a simple gesture set for practice, and two sequences using our two gesture sets with varying complexity. Participants had to execute 8 gestures randomly chosen, and then repeat the operation three more times. This design resulted in 4 design × 2 gesture set complexity × 8 gestures × 4 repetitions = 256 trials per participants, for a total of 4096 trials.

Results

The main dependent measures were the acquaintance time, the execution time, and the recognition rate (Table 1). A trial was successful if the score returned by the $1 Recognizer corresponding to the executed gesture was the highest score among the scores of all gestures in the current gesture set. We discarded 42 trials (~1.03%) due to tracking problems. Our data did not satisfy both the normality and the homogeneity of variances assumptions. We performed our analysis with Friedman and Wilcoxon non-parametric tests.

Acquaintance Time

We found a significant main effect of design [$\chi^2(3)=14.7$, $p<0.01$] and complexity [W=136, Z=3.52, $p<0.001$, r=0.62] on the acquaintance time, with the simple gesture set being faster than the complex gesture set. There was no significant difference between gesture types [W=75, Z=0.36, $p>0.05$].

Post-hoc tests revealed that 2DPath leads to significantly faster acquaintance time than Mtarget [$p<0.01$, r=0.61] for both complexities (Figure 5, left) and both gesture types:

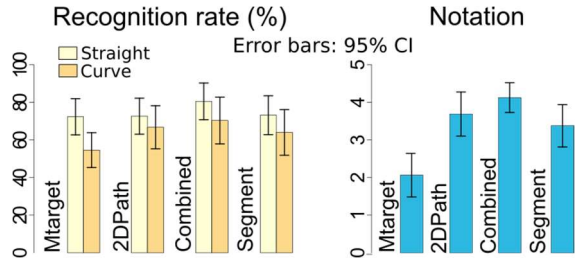


Figure 6: Left: Effect of gesture type and design on recognition rate. Right: Qualitative notation of each design by participants. (1: not preferred; 5 preferred).

Participants waited to detect the end of the loop with Mtarget in order to see the complete gesture shape.

Execution Time

We found a significant effect of complexity on execution time [W=136, Z=3.52, $p<0.001$, r=0.62]: complex gestures took longer than simple gestures. We did not find any significant effect of design [$\chi^2(3)=1.13$, $p>0.05$] and gesture type [W=93, Z=1.29, $p>0.05$]. Participants took the same amount of time to execute gestures with a user- or system-imposed pace, i.e. with dynamic or static visual elements only. This result holds for straight [$\chi^2(3)=3.68$, $p>0.05$] and curved gestures [$\chi^2(3)=0.90$, $p>0.05$] (Figure 5, right).

To understand why self-paced and system-imposed designs lead to the same execution time, we first looked at a possible side effect of our within-subject design: participants could have use 2DPath at the pace they used preceding designs. However, we did not find any significant effect of ordering [$\chi^2(3)=2.1$, $p>0.05$]. Second, we performed a statistical analysis on the number of fixations and their duration. 2DPath leads to more and longer fixations than the other designs [$\chi^2(3)=25.95$, $p<0.001$, r=0.3], slowing down the self-paced execution (Figure 5, right).

Recognition Rate

We found a main effect of design [$\chi^2(3)=19.28$, $p<0.001$], with Combined leading to a better recognition rate than Mtarget [$p<0.001$, r=0.33]. Gesture set complexity has no significant effect on the recognition rate [W=84, Z=0.83, $p>0.05$] for all designs. Gestures with straight lines had a higher recognition rate than curved gestures for all designs [W=0, Z=-3.52, $p<0.01$, r=0.62] (Figure 6, left). This result

		Acquaintance time (s)	Execution time (s)	Recognition rate (%)
Simple	Mtarget	4.99 [4.46, 5.52]	2.64 [2.56, 2.73]	62.65 [57.78, 67.51]
	2DPath	**2.66 [2.30, 3.02]**	2.84 [2.63, 3.04]	70.31 [65.15, 75.48]
	Combined	3.19 [2.46, 3.75]	**2.48 [2.46, 2.50]**	**75.03 [68.80, 81.25]**
	Segment	4.25 [3.71, 4.79]	2.56 [2.51, 2.62]	65.15 [59.21, 71.09]
Complex	Mtarget	9.74 [8.67, 10.82]	5.25 [5.15, 5.36]	64.17 [58.74, 69.61]
	2DPath	**4.31 [3.76, 4.86]**	**4.89 [4.49, 5.29]**	68.57 [62.86, 74.28]
	Combined	5.65 [4.42, 6.87]	5.16 [5.13, 5.19]	**76.56 [71.53, 81.59]**
	Segment	7.32 [6.20, 8.44]	5.27 [5.19, 5.35]	71.46 [65.98, 76.93]

Table 1: Values of the dependent measures of Experiment 1 for all design and complexity factors represented by: Average value [95% confidence interval]. Bold indicates best values.

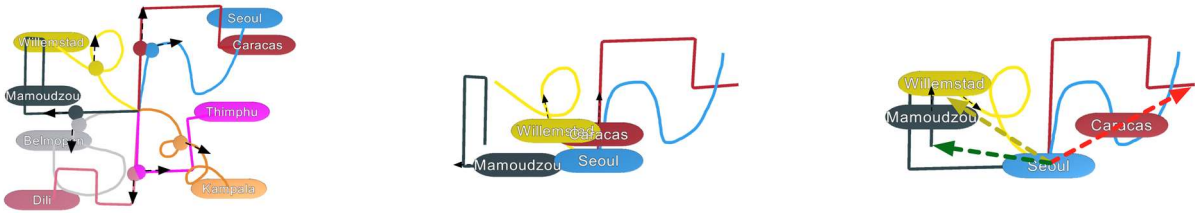


Figure 7: Designs considered in Experiment 2. The baseline displays all labels fixed at the end the paths (left). After apparition at the center, labels start moving along the 2D paths (black arrows) with G2 (center) or shift toward the end of the gesture (colored arrows) to move back toward the origin along the 2D paths (black arrows) with G3 (right). Arrows are for motion illustration only.

seems intuitive for 2DPath since users perform saccade movements, thus creating sequences of straight lines. However, this result even holds for designs with dynamic targets leading to pursuit eye movements.

Discussion

The visual complexity (controlled via the number, the length [46], and the total turning angle of the gestures [45]) has an effect on acquaintance and execution times. Longer gestures require longer execution time. However, the visual complexity has no effect on the recognition rate. Participants followed the target gesture with little concern for distractors.

Participants preferred Combined to Mtarget (Figure 6, right) [$\chi^2(3)=16.74$, $p<0.01$, $r=0.71$]: gesture shape anticipation overcomes visual complexity. Note that there is no particular preference between the designs regarding the eye movement induced, i.e. saccades with static 2D paths (2DPath), and pursuits with dynamic targets (Combined and Segment). The type of eye movement was hence not the main factor impacting the user experience.

EXPERIMENT 2: EXPLORATION AND RETURN PHASES

Concerning the *Exploration* phase, we aim at providing a solution that prevents overlap between visual search and gesture executed eye movements. For the *Return* phase, the aim is to provide a solution to help users find the initial area of interest after a gesture execution.

Exploration and Return Designs

We consider two enhancements to the Combined baseline using 2D paths and moving targets from experiment 1.

Baseline

The baseline displays colored 2D paths and corresponding dynamic colored targets moving along the path (Figure 7, a). Command names are displayed at the end of the 2D paths.
Exploration: No enhancement. Labels, 2D paths and moving targets show up at the same time. Users need to move their eyes to search for the desired commands.
Return: No enhancement. The guide disappears after gesture execution. Users finds the origin without visual support.

G2

For the first step toward G3, we use a Rapid Serial Visual Presentation (RSVP) to help exploring available commands, which consists in sequentially presenting words at the same location [27,42]. RSVP has been used with Personal Device Assistants [35] and smartwatches [19,20] as it minimizes the output space required for presenting information.

Exploration: Gesture paths along with command labels show up sequentially, with the label positioned at the center of the guide, i.e. where the user is supposedly looking at when the guide is triggered. The label represents the moving targets and starts moving along the corresponding 2D path after a short delay upon appearance (Figure 7, b).
Return: No enhancement. The guide disappears after the execution of a gesture.

G3

Users can end gestures at the location where they started the execution with closed gesture paths (e.g., circle or triangle). However, this imposes a specific type of shapes. Otherwise, users will have to move their eyes back to the origin of the guide after gesture execution. We invert the order of these eye movements, to make users perform the gesture from the exterior toward the interior, i.e. in an inward motion. Although the extra gaze movement is still present, it should eliminate the visual search for the origin of the guide.
Exploration: Similar to G2, each 2D path and label combination shows up one-by-one, with the label positioned at the center of the guide. After a short delay, the label quickly moves towards the end of the path along a straight line. Thus, labels quickly move away from the center, allowing a quick clarification of the reading area.
Return: Labels move from the end point to the first point of the gesture path, i.e. the center of the guide (Figure 7, c).

Apparatus and Participants

We used the same apparatus as in the first study. Twelve new participants (5 females), ages 21 to 27 (M=24.08, SD=2.17), volunteered for this experiment. One participant wore corrective lenses and four used contact lenses. None had previous experience with gaze tracking.

Procedure

The setup is similar to the first experiment. The experiment lasted around 1h per participant after which they filled a questionnaire for qualitative data. We presented each design (baseline, G2, G3) before starting the experiment.

A trial began by pressing the 'space' bar. Participants could see (i) the target command label to find and execute, and (ii) a grid of 20×9 circles displayed on the screen. Circles were grey except for a randomly placed green one indicating where the gestures will show up. Participants pressed 'space', which displayed the guide and entered into the *Exploration* phase. We asked participants to press 'space' again as soon as they found the desired gesture. Upon gesture

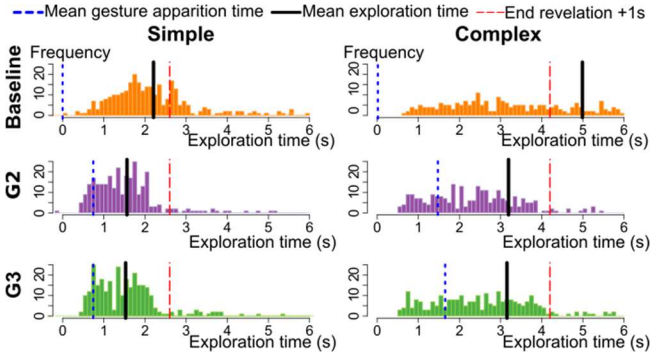


Figure 8: Frequency of exploration times for all designs across complexities. 0 is the trigger of the guide. The blue line shows the average gesture apparition time. The red line shows 1s after the last gesture is revealed. The black line shows the average exploration time of participants.

completion, participants could see the corresponding moving target blink. They pressed 'space' again during the blinking interval. This ensured that participants did not move their gaze toward the origin while the guide was still displayed. Once the 'space' bar was pressed, the guide disappeared, the green circle became grey and arrows appeared in each one of the 180 circles. Arrows indicated the top, bottom, left or right direction. Participants needed to press the arrow on the keyboard corresponding to the arrow in the origin circle to ensure they found the correct location. Participants finally received feedback regarding the match between the arrow pressed and the arrow to retrieve and could press the 'space' bar again to begin the next trial. The potential additional cognitive load due to the arrow key press action was the same for all three designs.

We made sure that the guide always showed up in a random circle avoiding out-of-screen gestures. This resulted in 27 different potential origins centered in the grid of circles. We used the same gesture sets as in the first study and used a scaling factor to avoid out-of-screen gestures. Moving targets completed gesture with the same time, hence a slightly slower speed during the *Guidance* phase (not the focus of this second study):

Visual characteristics: 2D paths had a thickness of 27.5mm. Moving targets were 1.24cm-diameter circles. Labels used a font 82.5mm. The grid of circles consisted of 1.65cm-diameter circles with 0.55mm space between them.

Timing characteristics: Moving elements had a speed of 9.9cm.s^{-1}. For RSVP, we defined a 0.2s delay between each label apparition. A label stayed 0.1s in the center before starting to move. For G3, labels moved toward the end of the gesture at a speed of 12.4cm.s^{-1}.

Experimental Design

We used a repeated-measure within-participant design. The independent variables were the guiding system design (Baseline, G2, G3) and the gesture set complexity (simple, difficult). We counter-balanced the design and complexity presentation across participants with a Latin-square design.

The experiment was divided into three sections: one for each design. Each section consisted of two blocks: one for each gesture set. Participants had a simple gesture set for practice before using the actual two experimental gesture sets. In each gesture sets, participants needed to perform a random sequence of 8 gestures three times. This design resulted in 3 designs × 2 gesture set complexities × 8 gestures × 3 repetitions = 144 trials per participants, for a total of 1728 trials. To avoid learning effects, gestures got new random label and color at each trial. Labels were chosen among a set of 100 worldwide capital cities.

Results

We considered the following dependent measures (Table 2):

- *Exploration time*: the elapsed time between the revelation of the guide and the moment participants found the desired label and pressed 'space'.
- *Return time*: the elapsed time between the disappearance of the guide after completion of a gesture and the moment participants pressed an arrow on the keyboard.
- *Origin retrieval success rate*: the success rate of the origin circle found after the disappearance of the guide.

We did not consider the recognition rate for this experiment, as we focused on the *Exploration* and *Return* phases, not the *Guidance* phase. Gesture recording segmentation mechanisms would have been too different between our designs and lead to different side-effects on our data.

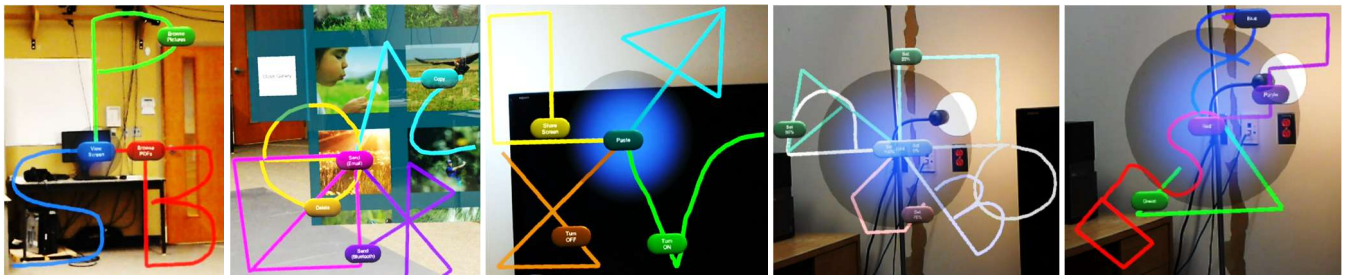
Exploration Time

We found a significant main effect of design [$\chi^2(2)=8.17$, $p<0.05$], and complexity [$W=78$, $Z=3.06$, $p<0.001$, $r=0.62$] on the exploration time. The Baseline leads to longer exploration times than G2 [$\chi^2(2)=8.00$, $p<0.05$, $r=0.61$] and G3 [$\chi^2(2)=8.00$, $p<0.05$, $r=0.71$] in both complexities.

Interestingly, we notice that more than 92% (resp. 86%) of the exploration phases end within the time between (i) the average apparition time of the simple (resp. complex) gestures (Figure 8, blue lines) and, (ii) 1s after the last gesture showed up (Figure 8, red lines). This time interval contains only 73% (resp. 59%) of the exploration phases for the simple (resp. complex) gesture set with the Baseline. Thus, average exploration times (Figure 8, black lines) are ~30% (resp. ~37%) faster with G2 and G3 than with Baseline for the simple (resp. complex) gesture set.

		Exploration time (s)	Return time (s)	Origin retrieval success rate (%)
Simple	Baseline	2.21 [1.98, 2.43]	1.21 [1.06, 1.37]	71.18 [66.97, 75.39]
	G2	1.55 [1.47, 1.63]	1.18 [0.99, 1.37]	85.76 [82.27, 89.26]
	G3	**1.52 [1.48, 1.56]**	**0.81 [0.72, 0.89]**	**97.57 [96.49, 98.65]**
Complex	Baseline	4.99 [4.51, 5.47]	1.49 [1.30, 1.68]	61.11 [58.28, 63.94]
	G2	3.13 [2.77, 3.48]	1.54 [1.30, 1.79]	64.93 [61.32, 68.54]
	G3	**3.11 [2.67, 3.55]**	**0.81 [0.92, 1.19]**	**92.01 [89.48, 94.55]**

Table 2: Values of the dependent measures of Experiment 2 for all design and complexity factors represented by: Average value [95% confidence interval]. Bold indicates best values.



a: Desktop b: Gallery c: TV screen d: Light e: Light

Figure 9: Gesture sets used in the office scenario based on literal representations (a: Desktop), icons (b: Gallery, c:TV screen), and arbitrary geometrical shapes (d and e: Light). Note that the light commands cannot be seen all at once. Blue spheres are holograms to represent interactive physical objects within the HoloLens viewpoint.

Return Time

We found a significant main effect of design [$\chi^2(2)=8.67$, $p<0.05$] and complexity [$W=70$, $Z=2.43$, $p<0.05$, $r=0.5$] on the return time. The complexity affects the return time with all designs: the Baseline [$W=72$, $Z=2.59$, $p<0.01$, $r=0.53$], G2 [$W=68$, $Z=2.27$, $p<0.05$, $r=0.46$], and unexpectedly G3 [$W=71$, $Z=2.51$, $p<0.01$, $r=0.51$]. G3 allows for ~30% faster return time than the other designs with the simple [$\chi^2(2)=12.17$, $p<0.01$, $r>0.4$] and the complex [$\chi^2(2)=10.5$, $p<0.01$, $r>0.48$] gesture sets (Figure 10, left).

Origin Retrieval Success Rate

We found a significant main effect of design [$\chi^2(2)=22.17$, $p<0.001$] and complexity [$W=0$, $Z=-3.07$, $p<0.001$, $r=0.63$]: participants were able to find the correct arrow more often with the simple gesture set than with the complex gesture set.

The complexity affected the ability to find the correct arrow with all designs, even G3 [$W=1.5$, $Z=2.29$, $p<0.05$, $r=0.47$]. However, G3 allows for more than 40% better results than Baseline and G2 with the complex gesture set (Figure 10, right). Although the difference between G2 and G3 is less important with the simple gesture set (14%), the difference between G3 and Baseline remains the same with the simple gesture set (40%). This result can be explained by the fact that with Baseline, even with the simple gesture set, participants had to explore the scene and hence move their eyes, losing the origin of the guide as soon as it appears.

There is no difference between G2 and G3 regarding the *Exploration* phase and the qualitative preferences of participants. Only the preference of G2 over Baseline was significant [$F_{2,22}=4.80$, $p<0.05$]. The final guide G3 can hence be used with or without the *Return* option without impact on performances or preferences.

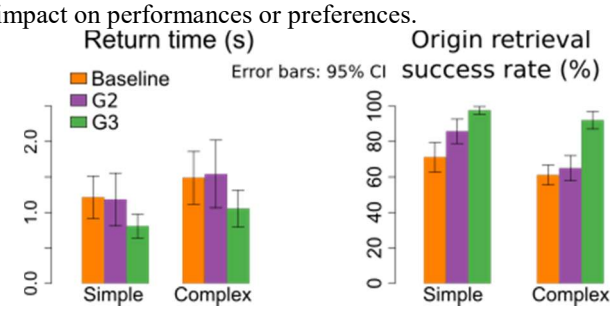


Figure 10: Return time (s) (left), and origin retrieval success rate (%) (right) across complexities.

Discussion

We provided two enhancements regarding the guiding system defined by 2D paths and moving targets from study 1. For the *Exploration* phase, we used a RSVP to display command labels in the center of the guide. This option proved to be more efficient than an explicit visual search of the desired label. For the *Return* phase, we used inward label motions, i.e. labels moving from the end point to the first point of the gesture path. This option proved to be more efficient than letting the participants find the origin of the guide on their own. The *Return* option can be removed without impact on preferences.

G3 EVALUATION: OFFICE ENVIRONMENT SCENARIO

Our summative evaluation is motivated by illustrating G3 in an 'office environment' scenario where users can interact with augmented objects via a see-through HMD. Our aim was to obtain qualitative feedback from participants using G3 in this scenario.

Scenario

In the context of an office environment, we envision augmented objects offering multiple services to users.

Interactive Objects and Gesture Sets

We use three augmented physical objects: a desktop computer, a TV screen and a LED light. We also use digital interactive pictures (Figure 11). We define:

- Three commands for the desktop computer – Browse Pictures, Browse PDFs and View Screen. This gesture set illustrates how the shapes can be based on the literal representation of the commands (e.g., a 'P' shape for "Browse Pictures", or 'S' for "View Screen") (Figure 9, a).
- Four commands for pictures – Copy, Delete, Send (Email) and Send (Bluetooth). This gesture set illustrates how the shapes can be icon-based (e.g., the Bluetooth icon or ^*C* for copy) (Figure 9, b).
- Four commands for the TV screen – Turn On, Turn OFF, Share Screen, and Paste. This gesture set also illustrates how gestures shapes can be icon-based (e.g., a check mark to turn on the screen) (Figure 9, c).
- Thirteen commands for the light to control intensity, temperature or color. This gesture set illustrates the use of arbitrary geometric shapes to accommodate the high number of gestures (Figure 9, d and e).

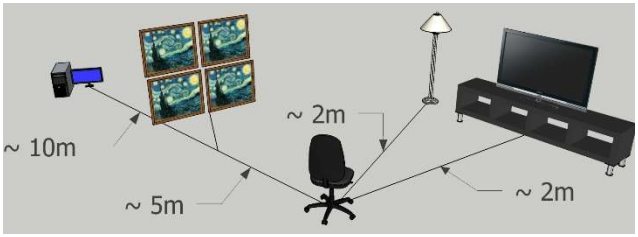


Figure 11: Positions of interactive objects used in our scenario with the user sitting on a distant chair.

Task

First, participants had to select the desktop computer to trigger G3. They then had to select the 'Browse Pictures' command using G3 to display a picture gallery. We asked them to select and copy pictures of their choice. They could scroll the gallery up and down by moving their head toward the bottom and the top of the picture gallery. They then had to select the TV screen and trigger the 'Paste' command. Lastly, we asked them to select the light and trigger a command of their choice. We asked participants to repeat the scenario 5 times. The experiment lasted around 30 minutes, including informal discussions during trials.

G3 Integration

We explored two trigger mechanisms for G3: finger tap and gaze dwell selections. A cursor displayed the head-orientation and was used for tap and dwell selections. Only gaze gestures with G3 used the eye-tracking system. For physical object selection, we chose a mid-air tap finger gesture (as provided on the HoloLens [32]). While this still requires users to perform a mid-air gesture, (1) it is a one-time input, and (2) it allows participants to perform an explicit action to trigger the guide and initiate the entire interaction process. For picture selection, we chose a dwell action. A circular progress bar appeared after looking at the same picture for 2s. Participants could cancel the trigger of the guide during the progress bar animation (1s) by looking away from the picture.

The system recorded an instance of gaze positions for each gesture, i.e. from the moment a label was displayed to the moment the label reached the center of the guide. This allowed the system to do the gesture registration and termination without any extra action from participants..

Participants and Apparatus

Twelve participants (3 females), ages 21 to 37 (M=25.5, SD=5.1), volunteered for this experiment. None of them participated in our previous experiments or had experience with gaze input. Three participants had glasses.

We used the HoloLens [32] as a head-mounted see-through display. Since the EyeTribe could not correctly track the eyes through the HoloLens display, we used the Tobii eye-tracker [51] with a 60Hz sampling rate. The custom eye-tracker server ran on a 3.6 GHz Intel Core i7 computer. The TV gallery listener ran on a 1.6GHz Intel Core i5 laptop. All three programs (HoloLens, TV gallery listener, and eye-tracker server) were implemented in C# with the Unity3D 5.3 game engine and communicated via a local Wi-Fi network.

The HoloLens has no built-in eye-tracking capabilities. Our prototype used a table attached to the arms of an office chair to fix the eye-tracker/user's head relative positions. Although preventing participants from freely moving, this final prototype still allowed orientation mobility.

Results

G3 was overall well received by all participants. Some participants said that "[G3] is really cool", "works well" (P1, P4, P5, P7, P8, P10), "is easy to use" (P2, P10) and "pretty intuitive" (P8). Although this validates the use of G3 as a viable input method, we also asked participants to think out loud to get more insights.

Some participants noted some **confusion** at first, either because of the number of gestures moving at the same time (P0, P1, P6) or from the overwhelming feeling brought by all the colors (P10). However, all of them added that this confusion disappeared with the second exposure and that the system had a "fast learning curve" (P10).

Participants shared different viewpoints on the **gesture sets** we used. Some would have preferred simpler gesture shapes (P3, P9, P11) while others found that the gesture shapes "made sense and are easy to do" (P10) or "easy to recognize" (P4). Some participants also mentioned the wish to do their own gestures (P5), potentially with default ones at first (P8).

Controlling the light was left open to each participant's will. When free to execute any commands, some participants reported a **discoverability** problem (P3, P4), as they don't know what commands are available on the first try. Although this problem concerns only the first use of G3 with a particular object, this could be prevented by adding a "quick map overview of the commands to show what you will have" (P4) or a simple looping mechanism.

Some participants expressed their wish for **additional feedback** regarding the recognition process (P0, P3, P5, P9). This feedback could highlight the most likely recognized gesture (P3), textually tells users what action to perform (P9), or as OctoPocus [5], reduces the number of displayed gestures by removing unlikely recognized gestures (P0, P5).

There is no definitive solution regarding the **trigger** of G3. Some participants preferred the finger-tap gesture (P3, P8, P10) as it was an easy and fast action, while others preferred the dwell action (P5, P6, P11) to perform "hands-free" and no "cumbersome and tiresome" hand movements. We believe that the trigger should be user-defined according to users' preferences. The same applies to G3's **dynamics**. For instance, while some participants found the speed of the RSVP comfortable (P5, P8, P9), some others found it "too fast" (P7, P10) and others too slow (P2, P3). The same applies for the speed of the label during the *Guidance* phase. This shows that the fine-tuning of G3 is user-dependent. Note that these feedbacks concerned mostly the first

exposure and that with even more practice, all participants might want to accelerate the RSVP and label motions.

Finally, we also report feedback regarding the **learning** aspect. Some participants memorized the order of the gestures (P6), tried to execute the gesture before the command appears (P7) or ahead of the label motion (P8). Performing semaphoric gaze gestures without a guiding system such as G3 is further discussed in the next section.

LIMITATION AND FUTURE WORK

In this work, we introduce a gesture guiding system, namely G3, allowing users to perform any type of semaphoric gaze gestures. Our design exploration results in gaze gestures performed with pursuit eye movements.

Pursuit eye movements are not feasible without a moving target [26]. This means that G3 needs to be displayed to allow this type of eye movements. But the end-goal of any guiding system is to ultimately let users perform gestures on their own once they are experts. The novice-to-expert transition is the next step to tackle regarding the use of semaphoric gaze gestures. From a cognitive point-of-view, it is not clear if the reverse execution used by G3 will impact the learning of the mapping between gestures and commands. G3 could then use shapes centered on their last point instead of their first point, so that the gesture execution follows the semantic value of the gesture shape. From a motor point-of-view, the first challenge concerns the gesture segmentation without G3, i.e. without dynamic labels recording the registration and the termination of the gesture. The second challenge concerns the actual gesture execution without the guiding system. We explore this in future work.

The Guidance Hypothesis states that users better manage to reproduce motor movements if they learn without relying too much on guidance [37]. The next step consists in reducing and then removing these visual elements according to the user's level of expertise [2]. We consider two G3's main limitations for the novice-to-expert transition: the **pursuit** movements induced by the dynamic elements, and the anticipation cues provided by the static **2D paths**.

The first step consists in making users perform saccades instead of pursuit movements, as they would do in expert mode, i.e. without G3. A solution is to provide periodic accelerations of the moving targets so that the movement consists of sequences of saccades and fixations. An interesting challenge with this option is the actual segmentation of the gesture shapes: Should fixation points be on strategic geometric landmarks of the shape? While corners seem appropriate landmarks for sequences of straight lines, what about curved lines? The second step is the removal of dynamic targets in order to keep only 2D paths, a viable guiding solution as shown in study 1.

The third step consists in making users perform gestures without 2D paths. A solution we are aiming for is to reduce the opacity of the 2D paths. Indeed, we should help users perform the gesture based on what is in front of them in the physical world - hence reducing the opacity. This will finally lead to two more research questions: (1) can users perform any semaphoric gestures without guidance, i.e. without dynamic and/or static visual elements? (2) Is the background impacting the learning process, i.e. are users considering physical world elements as strategic reference points? If so, can users perform gaze gestures with a different viewpoint than during practice? Also, if physical landmarks are used during learning, an interesting approach consists in generating the gesture shapes based on these landmarks. The guiding system could use image analysis to display a particular shape according to landmarks in front of the user.

CONCLUSION

Current Head-Mounted Display (HMD) input often requires large hand movements. This can be cumbersome in hands-busy situations or can cause fatigue. As an alternative, we propose the use of semaphoric gaze gesture interaction. However, any application using semaphoric gestures needs to provide a solution for users to know the gesture vocabulary, i.e. the available commands and their associated gestures. This work addresses the design of a gesture guiding system especially adapted to the gaze modality.

We presented two user studies to design a gaze gesture guiding system based on three key phases: the *Exploration*, the *Guidance,* and the *Return* phases. The first experiment compared design options for the *Guidance* phase. Results show that users prefer following a dynamic target moving along a displayed 2D path. This solution also proved to be resistant to visual complexity. The second experiment compared enhancement of the previous design with respect to the *Exploration* and the *Return* phases. Results show that a Rapid Serial Visual Presentation (RSVP) of gesture paths with their labels at the center of the guide allows for more than 30% faster command retrieval than a standard visual exploration of on-screen labels. To accommodate the *Return* phase and help users resume the task they were doing before triggering the guide, labels (1) quickly shift toward the outer side, and (2) guide users by moving back toward the center along their corresponding 2D paths. This solution allows for 40% better origin retrieval than when users are guided with labels moving outward. Lastly, we illustrated how to integrate and use our resulting Gaze Gesture Guiding System, G3, on a HMD for interacting with distant digital and augmented physical objects. This summative evaluation identifies strengths and limitations with G3 for semaphoric gaze gestures. The next step is the design exploration of adaptive behaviors of G3 to help novices transition to expert behavior to execute gaze gestures without guidance.

ACKNOWLEDGMENTS

We thank NSERC and Mitacs for the financial support of this project. We also thank our participants and reviewers.

REFERENCES

1. Roland Aigner, Daniel Wigdor, Hrvoje Benko, et al. 2012. *Understanding Mid-Air Hand Gestures : A Study of Human Preferences in Usage of Gesture*

Types for HCI. Retrieved March 18, 2014 from http://131.107.65.14/en-us/um/people/benko/publications/2012/Understanding MidAir Gestures - MSR-TR-2012-111.pdf

2. Fraser Anderson and Walter F Bischof. 2013. Learning and performance with gesture guides. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13*, 1109–1118. http://doi.org/10.1145/2470654.2466143
3. Fraser Anderson, Tovi Grossman, Justin Matejka, and George Fitzmaurice. 2013. YouMove: Enhancing Movement Training with an Augmented Reality Mirror. In *Proceedings of the 26th annual ACM symposium on User interface software and technology - UIST '13*, 311–320. http://doi.org/10.1145/2501988.2502045
4. Mihai Bâce, Teemu Leppänen, David Gil de Gomez, and Argenis Ramirez Gomez. 2016. ubiGaze: ubiquitous augmented reality messaging using gaze gestures. In *SIGGRAPH ASIA 2016 Mobile Graphics and Interactive Applications - SA '16*, 1–5. http://doi.org/10.1145/2999508.2999530
5. Olivier Bau and Wendy E. Mackay. 2008. OctoPocus: A Dynamic Guide for Learning Gesture-Based Command Sets. In *Proceedings of the 21st annual ACM symposium on User interface software and technology - UIST '08*, 37–46. http://doi.org/10.1145/1449715.1449724
6. Nikolaus Bee and Elisabeth André. 2008. Writing with your eye: A dwell time free writing system adapted to the nature of human eye gaze. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 5078 LNCS: 111–122. http://doi.org/10.1007/978-3-540-69369-7_13
7. Andrew Bragdon, Robert Zeleznik, Brian Williamson, Timothy Miller, and Joseph J. LaViola. 2009. GestureBar: Improving the Approachability of Gesture-based Interfaces. In *Proceedings of the 27th international conference on Human factors in computing systems - CHI '09*, 2269–2278. http://doi.org/10.1145/1518701.1519050
8. Marcus Carter, Eduardo Velloso, John Downs, Abigail Sellen, Kenton O'Hara, and Frank Vetere. 2016. PathSync: Multi-User Gestural Interaction with Touchless Rhythmic Path Mimicry. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16*, 3415–3427. http://doi.org/10.1145/2858036.2858284
9. Andrea Colaço, Ahmed Kirmani, Hye Soo Yang, Nan-Wei Gong, Chris Schmandt, and Vivek K. Goyal. 2013. Mime: Compact, Low-Power 3D Gesture Sensing for Interaction with Head-Mounted Displays. In *Proceedings of the 26th annual ACM symposium on User interface software and technology - UIST '13*, 227–236. http://doi.org/10.1145/2501988.2502042
10. Dh Cymek, Ac Venjakob, Stefan Ruff, Otto Hans-Martin Lutz, Simon Hofmann, and Matthias Rötting. 2014. Entering PIN Codes by Smooth Pursuit Eye Movements. *Journal of Eye Movement Research* 7, 4: 1–11. http://doi.org/10.16910/jemr.7.4.1
11. Edwin S. Dalmaijer. 2014. Is the low-cost EyeTribe eye tracker any good for research? *PeerJ PrePrints* 4, 606901: 1–35. http://doi.org/10.7287/peerj.preprints.141v2
12. William Delamare, Céline Coutrix, and Laurence Nigay. 2015. Designing guiding systems for gesture-based interaction. In *Proceedings of the 7th ACM SIGCHI Symposium on Engineering Interactive Computing Systems - EICS '15*, 44–53. http://doi.org/10.1145/2774225.2774847
13. William Delamare, Thomas Janssoone, Céline Coutrix, and Laurence Nigay. 2016. Designing 3D Gesture Guidance : Visual Feedback and Feedforward Design Options. In *Proceedings of the International Working Conference on Advanced Visual Interfaces - AVI '16*, 152–159. http://doi.org/10.1145/2909132.2909260
14. Heiko Drewes and Albrecht Schmidt. 2007. Interacting with the Computer Using Gaze Gestures. In *Proc. of the Int. Conf. on Human-computer interaction'07*. 475–488. http://doi.org/10.1007/978-3-540-74800-7_43
15. Morten Lund Dybdal, Javier San Agustin, and John Paulin Hansen. 2012. Gaze input for mobile devices by dwell and gestures. In *Proceedings of the Symposium on Eye Tracking Research and Applications - ETRA '12*, 225–228. http://doi.org/10.1145/2168556.2168601
16. Augusto Esteves, Eduardo Velloso, Andreas Bulling, and Hans Gellersen. 2015. Orbits: Gaze Interaction for Smart Watches using Smooth Pursuit Eye Movements. *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology - UIST '15*, 1: 457–466. http://doi.org/10.1145/2807442.2807499
17. Dustin Freeman, Hrvoje Benko, Meredith Ringel Morris, and Daniel Wigdor. 2009. ShadowGuides: Visualizations for In-Situ Learning of Multi-Touch and Whole-Hand Gestures. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces - ITS '09*, 165–172. http://doi.org/10.1145/1731903.1731935
18. Emilien Ghomi, Stéphane Huot, Olivier Bau, Michel Beaudouin-Lafon, and Wendy E. Mackay. 2013. Arpège: Learning Multitouch Chord Gestures Vocabularies. In *Proceedings of the 2013 ACM*

international conference on Interactive tabletops and surfaces - ITS '13, 209–218. http://doi.org/10.1145/2512349.2512795

19. John Paulin Hansen, Florian Biermann, Janus Askø Madsen, et al. 2015. A gaze interactive textual smartwatch interface. *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers - UbiComp '15*: 839–847. http://doi.org/10.1145/2800835.2804332
20. John Paulin Hansen, Haakon Lund, Florian Biermann, Emillie Møllenbach, Sebastian Sztuk, and Javier San Agustin. 2016. Wrist-worn pervasive gaze interaction. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications - ETRA '16*, 57–64. http://doi.org/10.1145/2857491.2857514
21. Juan David Hincapié-Ramos, Xiang Guo, Paymahn Moghadasian, and Pourang Irani. 2014. Consumed Endurance: A Metric to Quantify Arm Fatigue of Mid-Air Interactions. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14*, 1063–1072. http://doi.org/10.1145/2556288.2557130
22. Aulikki Hyrskykari, Howell Istance, and Stephen Vickers. 2012. Gaze gestures or dwell-based interaction? In *Proceedings of the Symposium on Eye Tracking Research and Applications - ETRA '12*, 229–232. http://doi.org/10.1145/2168556.2168602
23. Poika Isokoski. 2000. Text input methods for eye trackers using off-screen targets. *Proceedings of the symposium on Eye tracking research applications ETRA 00*: 15–21. http://doi.org/10.1145/355017.355020
24. Howell Istance, Aulikki Hyrskykari, Lauri Immonen, Santtu Mansikkamaa, and Stephen Vickers. 2010. Designing Gaze Gestures for Gaming: An Investigation of Performance. In *Proceedings of the 2010 Symposium on Eye-Tracking Research and Applications - ETRA '10*, 323–330. http://doi.org/10.1145/1743666.1743740
25. RJK Jacob and Keith S Karn. 2003. Eye tracking in human-computer interaction and usability research: Ready to deliver the promises. In *The Mind's Eye: Cognitive and Applied Aspects of Eye Movement Research*. 4.
26. Robert J K Jacob. 1993. Eye Movement-Based Human-Computer Interaction Techniques: Toward Non-Command Interfaces. *Advances in human-computer interaction* 4: 151–190.
27. James F Juola, Nicklas J Ward, and Timothy McNamara. 1982. Visual Search and Reading of Rapid Serial Presentations of Letter Strings, Words, and Text. *Journal of Experimental Psychology: General* 111, 2: 208–227. http://doi.org/10.1037//0096-3445.111.2.208
28. Jari Kangas, Deepak Akkil, Jussi Rantala, Poika Isokoski, Päivi Majaranta, and Roope Raisamo. 2014. Gaze gestures and haptic feedback in mobile devices. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14*, 435–438. http://doi.org/10.1145/2556288.2557040
29. Maria Karam and m. c. Schraefel. 2005. A Taxonomy of Gestures in Human Computer Interactions. 1–45. http://doi.org/10.1.1.97.5474
30. G. Kurtenbach, T. P. Moran, and W. Buxton. 1994. Contextual Animation of Gestural Commands. *Computer Graphics Forum* 13, 5: 305–314. http://doi.org/10.1111/1467-8659.1350305
31. Gordon Kurtenbach, Abigail Sellen, and William Buxton. 1993. An Empirical Evaluation of Some Articulatory and Cognitive Aspects of Marking Menus. *Human-Computer Interaction 8*, 1–23. http://doi.org/10.1207/s15327051hci0801_1
32. Microsoft. Hololens. Retrieved December 8, 2016 from https://www.microsoft.com/microsoft-hololens/fr-ca
33. Emilie Møllenbach, John Paulin Hansen, and Martin Lillholm. 2013. Eye Movements in Gaze Interaction. *Journal of Eye Movement Research* 6: 1–15. http://doi.org/10.16910/jemr.6.2.1
34. Emilie Møllenbach, Martin Lillholm, Alastair Gail, and John Paulin Hansen. 2010. Single gaze gestures. *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications - ETRA '10*, 2: 177. http://doi.org/10.1145/1743666.1743710
35. Gustav Öquist, Anna Sågvall Hein, Jan Ygge, and Mikael Goldstein. 2004. Eye Movement Study of Reading on a Mobile Device Using the Page and RSVP Text Presentation Formats. In *Lecture Notes in Computer Science*. 108–119. http://doi.org/10.1007/978-3-540-28637-0_10
36. Hyung Min Park, Seok Han Lee, and Jong Soo Choi. 2008. Wearable augmented reality system using gaze interaction. *Proceedings - 7th IEEE International Symposium on Mixed and Augmented Reality 2008, ISMAR 2008*: 175–176. http://doi.org/10.1109/ISMAR.2008.4637353
37. J H Park, C H Shea, and D L Wright. 2000. Reduced-frequency concurrent and terminal feedback: a test of the guidance hypothesis. *Journal of motor behavior* 32, 3: 287–296. http://doi.org/10.1080/00222890009601379
38. Nicholas M. Ross and Elio M. Santos. 2014. The relative contributions of internal motor cues and

external semantic cues to anticipatory smooth pursuit. *Proceedings of the Symposium on Eye Tracking Research and Applications - ETRA '14*: 183–186. http://doi.org/10.1145/2578153.2578179

39. Gustavo Rovelo, Donald Degraen, Davy Vanacken, Kris Luyten, and Karin Coninx. 2015. Gestu-Wan - An Intelligible Mid-Air Gesture Guidance System for Walk-up-and-Use Displays. In *Proc. Interact '15*, 368–386. http://doi.org/10.1007/978-3-319-22668-2_28

40. Quentin Roy, Sylvain Malacria, Yves Guiard, Eric Lecolinet, and James Eagan. 2013. Augmented Letters: Mnemonic Gesture-based Shortcuts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '13), 2325–2328. http://doi.org/10.1145/2470654.2481321

41. D Rozado, T Moreno, J San Agustin, F B Rodriguez, and P Varona. 2015. Controlling a Smartphone Using Gaze Gestures as the Input Mechanism. *Human–Computer Interaction* 30, 1: 34–63. http://doi.org/10.1080/07370024.2013.870385

42. Gary S Rubin and Kathleen Turano. 1992. Reading without saccadic eye movements. *Vision Research* 32, 5: 895–902. http://doi.org/10.1016/0042-6989(92)90032-E

43. Rajinder Sodhi, Hrvoje Benko, and Andrew Wilson. 2012. LightGuide: Projected Visualizations for Hand Movement Guidance. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI '12*, 179–188. http://doi.org/10.1145/2207676.2207702

44. Oleg Špakov, Poika Isokoski, Jari Kangas, Deepak Akkil, and Päivi Majaranta. 2016. PursuitAdjuster : An Exploration into the Design Space of Smooth Pursuit – based Widgets. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications - ETRA '16*, 287–290. http://doi.org/10.1145/2857491.2857526

45. Radu-Daniel Vatavu, Lisa Anthony, and Jacob O. Wobbrock. 2013. Relative accuracy measures for stroke gestures. *Proceedings of the 15th ACM on International conference on multimodal interaction - ICMI '13*: 279–286. http://doi.org/10.1145/2522848.2522875

46. Radu-Daniel Vatavu, Daniel Vogel, Géry Casiez, and Laurent Grisoni. 2011. Estimating the Perceived Difficulty of Pen Gestures. In *Proceedings of the 13th IFIP TC 13 international conference on Human-computer interaction - INTERACT'11*. Springer-Verlag Berlin, Heidelberg, 89–106. http://doi.org/10.1007/978-3-642-23771-3_9

47. Eduardo Velloso, Markus Wirth, Christian Weichel, Augusto Esteves, and Hans Gellersen. 2016. AmbiGaze: Direct Control of Ambient Devices by Gaze. *Proceedings of the 2016 ACM Conference on Designing Interactive Systems - DIS '16*, July: 812–817. http://doi.org/10.1145/2901790.2901867

48. Mélodie Vidal, Andreas Bulling, and Hans Gellersen. 2013. Pursuits: Spontaneous interaction with displays based on smooth pursuit eye movement and moving targets. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing - UbiComp '13*, 439. http://doi.org/10.1145/2493432.2493477

49. Jacob O Wobbrock, James Rubinstein, Michael Sawyer, and Andrew T. Duchowski. 2007. Not Typing but Writing: Eye-based Text Entry Using Letter-like Gestures. *Proceedings of The Conference on Communications by Gaze Interaction (COGAIN)*, Figure 1: 61–64.

50. Jacob O Wobbrock, Andrew D Wilson, and Yang Li. 2007. Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes. In *Proceedings of the 20th annual ACM symposium on User interface software and technology - UIST '07*, 159. http://doi.org/10.1145/1294211.1294238

51. Tobii. Retrieved December 8, 2016 from http://tobiigaming.com/product /