StickyPie: A Gaze-Based, Scale-Invariant Marking Menu Optimized for AR/VR

SUNGGEUN AHN, Chatham Labs, Toronto, Ontario, Canada and HCI Lab, School of Computing, KAIST, Daejeon, Korea, Republic of
STEPHANIE SANTOSA, Chatham Labs, Toronto, Ontario, Canada
MARK PARENT, Chatham Labs, Toronto, Ontario, Canada
DANIEL WIGDOR, Chatham Labs, Toronto, Ontario, Canada and University of Toronto, Toronto, Ontario, Canada
TOVI GROSSMAN, Department of Computer Science, University of Toronto, Toronto, Ontario, Canada
MARCELLO GIORDANO, Chatham Labs, Toronto, Ontario, Canada

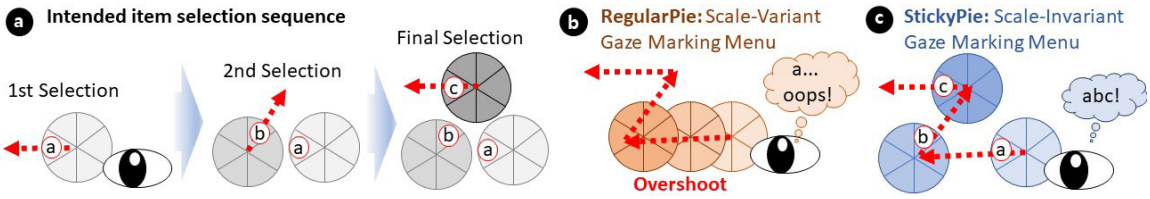

Fig. 1. StickyPie mitigates overshoot errors when performing a border crossing selection on a gaze-based marking menu: (a) Example item selection sequence with a gaze-based marking menu. (b) Example of overshoot error which frequently occurs with a regular gaze-based marking menu (RegularPie) because of the scale-variance of gaze marking input. (c) We present StickyPie, adopting the concept of scale-invariant marking input toward better support of gaze gestures in marking menus.

This work explores the design of marking menus for gaze-based AR/VR menu selection by expert and novice users. It first identifies and explains the challenges inherent in ocular motor control and current eye tracking hardware, including overshooting, incorrect selections, and false activations. Through three empirical studies, we optimized and validated design parameters to mitigate these errors while reducing completion time, task load, and eye fatigue. Based on the findings from these studies, we derived a set of design guidelines to support gaze-based marking menus in AR/VR. To overcome the overshoot errors found with eye-based expert marking menu behaviour, we developed StickyPie, a marking menu technique that enables scale-independent marking input by estimating saccade landing positions. An evaluation of StickyPie revealed that StickyPie was easier to learn than the traditional technique (i.e., RegularPie) and was 10% more efficient after 3 sessions.

CCS Concepts: • **Human-centered computing** → **Interaction techniques**; *Empirical studies in interaction design.*

Additional Key Words and Phrases: eye gaze input, marking menu, AR/VR, head-worn display

ACM Reference Format:
Sunggeun Ahn, Stephanie Santosa, Mark Parent, Daniel Wigdor, Tovi Grossman, and Marcello Giordano. 2021. StickyPie: A Gaze-Based, Scale-Invariant Marking Menu Optimized for AR/VR. In *CHI Conference on Human Factors in Computing Systems (CHI'21), May 8–13, 2021, Yokohama, Japan.* ACM, New York, NY, USA, 23 pages. https://doi.org/10.1145/3411764.3445297

Manuscript submitted to ACM

1 INTRODUCTION

With the integration of eye tracking within many off-the-shelf augmented reality (AR) and virtual reality (VR) head-mounted displays (HMDs) [11], such as Vive Pro Eye, FOVE VR, Hololens2, and Magic Leap One, there is an opportunity to leverage gaze as an input modality for hands-free interaction. Current AR/VR user interface designs typically rely on controller input to navigate graphical user interface (GUI) elements using a WIMP-like interaction style, however, best practices for gaze-based AR/VR interfaces have yet to be established.

Navigating a GUI menu and inputting hotkey commands are two of the most common methods users employ to interact with applications on computing devices [4]. GUI menu interfaces offer help to novice users because they enable users to input a command or select an item by navigating through the GUI interface and don't require the user to have prior knowledge about the structure of the menu. However, for expert users, navigating these GUIs can be cumbersome and slow [4]. Thus, many applications also provide hotkey functionality to provide experts with greater input efficiency.

Marking menus offer support for novice (GUI menu) and expert (gestures) users and facilitate gradual transitions towards expertise [4]. As with traditional input modalities, varying levels of expertise need to be supported within eye-based interfaces. To this end, Huckauf and Urbina adopted a marking menu interface for the eye input modality [15]. With eye-based marking menus, a novice user can make a selection by exploring the available options in a pie menu and then select an item using a border-crossing gesture ([15], Figure 2a). This is well matched to the ballistic eye movements (i.e., saccades), and experts can make the same selection by moving their eyes in a gaze gesture - a trajectory that mimics the stroke they would input if using a mouse or stylus [8].

Despite the potential benefits of gaze-based marking menus, there is limited work exploring their use. Huckauf and Urbina presented studies exploring eye-based marking menu interfaces [14, 15, 43], however, these studies did not consider a number of important design considerations; (1) how large to render pie menus; (2) how large a margin (see figure 8a) must be from content to border to avoid accidental activations of border crossing; (3) how to design marking menu gestures considering possible limits of depth and granularity.

Our first experiment was aimed at determining the optimal size of menu items by comparing user preference and input performance across different pie sizes using a baseline menu layout, then layouts with increased depth and granularity. In addition to insights on pie size, the first experiment also revealed that overshooting, i.e., the unintentional selection of multiple items, was a significant source of error.

To address this, we present StickyPie (Figure 1): a marking menu that treats every saccade as a single border crossing action. In contrast, with the regular pie version (RegularPie), a long distance saccade could yield multiple border crossing actions. By detecting the next landing position, the StickyPie system can match the next menu location to it and avoid multiple selections of adjacent menu items. Thus, StickyPie is not sensitive to different saccade lengths, and enables a "scale-invariant" gaze marking input.

Subsequently, we conducted a second experiment to investigate the usability of StickyPie with respect to menu depth and granularity given the optimal size of menu items determined from first experiment. Lastly, we evaluated StickyPie against RegularPie using a multi-day experimental methodology with a marking menu design for a universal command set for VR applications derived from the design guidelines. Our contribution is three-fold:

- **Design Guidelines**: we introduce design guidelines for gaze-based marking menus derived the findings from experiments 1 and 2.
- **Practical Design Example of the Gaze-based Marking Menu for VR application**: we create a practical marking menu with 32 universal commands for VR applications following our design guidelines.

- **StickyPie**: we present StickyPie as a optimized gaze marking menu that supports scale invariance. The results of Experiment 3 demonstrated StickyPie had better input performance and better learnability than RegularPie.

2 RELATED WORK

This work applies and builds on prior research on gaze interaction in AR/VR, marking menu techniques, gaze input methods, and gaze gesture guidance method.

2.1 Eyes-only- and Head-Combined Gaze Interactions in AR/VR HMD

Previous studies have explored challenges and unique design considerations with gaze interactions in AR/VR HMD. One key consideration is whether or not to combine eye tracking with head rotation for gaze input. Natural gaze shifts are not only performed with eye movement but also coupled with head rotation [37].

Recent work has explored new pointing and selection techniques in VR by leveraging eye-only and eye+head input [24, 34, 38] as well as tradeoffs between eye and head interactions, with eye gaze being potentially faster but less accurate. Qian and Teather [33] showed head-only conditions performed better than eye+head, followed closely by eye-only in a reciprocal selection task . However, Blattgerste *et al.* [3] reported the opposite result: the eye-only condition outperformed the head-only condition. They attributed the difference in performance to the effect of the eye tracker's accuracy. Similarly, Qian and Teather [33] also indicated that this was largely attributable to lower precision and accuracy in the eye tracking and calibration techniques, and noted that users complained about neck fatigue and nausea with head-only interactions. Another study by Sidenmark and Gallerson [37] revealed different contributions of head rotation to gaze behavior in different gaze zones. They found head rotation improves user comfort for gaze shifts over 25°, however eyes-only gaze affords user comfort for gaze shifts up to 25°.

Overall, we found that there was no clear evidence in the literature to support eye-only or head-combined interaction. Head-combined interaction can provide better usability for large interaction areas (over 25°). Otherwise, eye-only interaction can be a suitable option for a smaller interaction area. Both have their respective advantages and disadvantages and are worth investigating. In this paper we decided to limit our scope to eye-only interactions.

2.2 Gaze Input Menu and Gaze Gesture

The majority of eye gaze input techniques use three types of eye movements [29]: fixations, saccades, and smooth pursuits. Fixations, or dwell-based, methods are often used to select an item from within a GUI menu interface [20, 42]. Dwell methods are easy to use for the novice user, but the required dwell time introduces an overhead cost, specifically for a hierarchical menu. Many studies have investigated possible ways to reduce this overhead [27, 42], however, shorter dwell times often lead to more unintended selection errors, referred as the Midas touch problem [19].

Using smooth pursuit eye movements is a possible solution to increasing menu capacity without sacrificing input efficiency. With Orbits, Esteves *et al.* demonstrated that smooth pursuits, when combined with overlapping, moving GUIs, can be used for gaze-based smartwatch display input [9]. In addition, Drewes *et al.* [7] found that an extensive number of targets (160 targets) can be presented at a time on the 17-inch monitor. While this is easy to use for a novice, finding an item within a collection of bunch of moving items could be time consuming. Moreover, because items are not placed in a fixed position, the user is not able to memorize where a target is placed. Thus, the visual search time would create overhead for an expert user.

Gaze gestures [8], which use sequential ballistic eye movements (i.e. saccades), are somewhat free of these problems. Unlike in the dwell menu, graphical interfaces are not mandatory for gaze gestures [17] so menu capacities are not

limited by the size of a display. In addition, if a user knows the required gestures beforehand, they do not need to spend time searching through every item. Gaze gestures have been shown to have better input efficiency than dwell-based methods [16]. These features will benefit expert users, however, because human memory resources are limited, the number of memorable items is also limited and will require more effort for novices to learn.

Given these advantages for gaze gestures, we believe they are well-suited to command entry for experts. Thus, we explore the application of these gestures for interactions with marking menus in the gaze input modality.

2.3 Marking Menu Technique

Traditional marking menus offer users the option of using pie/radial-based menu input as a novice input mode and gesture- or stroke-based input as an expert input mode, while ensuring that both modalities can be explicitly distinguished [22]. In these marking menus, the user is initially in an expert mode without a GUI menu, but they can easily pop up GUI menu as a novice input mode. If the user places a stylus on a input space, a radial GUI menu appears and the user can select an item from the menu by stroking in the corresponding direction (i.e., novice mode). It therefore supports natural transitions from novice to expert [4]. To this end, Kurtenbach *et al.* highlighted three design principles for supporting this natural transition [23]: revelation (i.e. the interface should reveal which items are in the menu and how select them), guidance (i.e. the information should be seamlessly given to not interrupt the specific process of a user's task), and rehearsal (i.e. the guidance provided for the novice mode input should provide rehearsal for the use of expert mode input). The importance of rehearsal was also highlighted by Scarr *et al.* [36], who demonstrated that the rehearsal property can prevent performance decreases when users transition from novice to expert. We apply the principles and techniques from this body of work and adapt them to support both novices and experts for Gaze-Based menu input.

2.4 Supporting Novice Users in Gaze Gestures

Many techniques have been developed to support the transition from novice to expert (i.e., gaze gesture input). These can largely be classified into five categories: (1) memory aids, (2) cheat sheets (3) static feedforward guidance, (4) dynamic feedforward guidance, and (5) sequential feedforward guidance.

2.4.1 Memory aids. Generally, gesture strokes are expected to provide cognitive aids while learning gestures [1]. As eye input techniques, EdgeWrite [45] and Eye-S [32] used letter-like gestures for text-entry. Similarly, G3 [6] assigned the shapes of the first letter of a command to the corresponding command, such "C" for "Copy" [35]. As user-defined gesture sets have been shown to be more memorable than pre-designed gesture sets [30], Hou *et al.* presented a user defined gesture set for eye input. Memorable gestures could be use to boost expertise, however, a novice user can not use such gestures before memorizing them.

2.4.2 Cheat sheets. Providing a cheat sheet provides users with a list of gestures for them to reference [16, 45]. However, the glancing behavior that is required to look away from one's task towards the cheat sheet requires explicit context switching which can negatively influence novice to expert transitions thus violating the guidance principle of marking menu design [23].

2.4.3 Static feedforward guidance. This form of guidance is similar to a cheat sheet but does not require explicit context switching. Majaranta *et al.*, for example, presented a method for static illustration using gaze gestures [28]. They provided a gesture-shaped illustration to a user with the corresponding command-based information. One concern

with this type of guidance is that the limited screen space that is often available can restrict the number of illustrations that can be shown at a time.

2.4.4 Dynamic feedforward guidance. These methods guide the production of a gesture through the use of moving stimuli to encourage smooth-pursuit eye movements. G3 [6], for example, is an adaptation of the OctoPocus menu [2], and encourages semaphoric gesture input using smooth pursuits rather than the dynamic target selection, which is the dominant method of using smooth-pursuit information [7, 9, 21]. OctoPocus itself is also a well-known novice to expert transitioning technique, but as the authors pointed out, smooth-pursuit based methods requires a passive following eye movement to perform gestures, which differs from usual expert mode input.

2.4.5 Sequential Feedforward Guidance: Marking Menu. This marking menu-based technique is a powerful tool to support the development of expertise. However, surprisingly, only a few studies have tried to adapt marking menus for eye gaze input [14, 15, 18, 43]. Work by Isomoto *et al.* [18] presented a marking input detection technique as the first adaptation of a marking menu for eye input, however, they did not support a novice input mode.

Huckauf and Urbina presented the first adaptation of the marking menu, pEyeWrite [15]. However, the techniques in pEye and traditional marking menus [22, 46] have several differences. There is no differentiation in interaction with pEye menus between novice and expert users. The GUI menu is always rendered but a user can stroke sequentially to give (i.e. novice) or not give (i.e. expert) attention to the GUI. In contrast, traditional marking menu GUIs fade from use when expertise builds, no longer rendering for marking input. In addition, the scale of the gesture stroke is fixed (i.e. scale-dependent) for the pie menu, because each pie is always rendered at the same fixed position after crossing the border. In contrast, a gesture stroke in the marking menu is ideally scale-independent if all marks have inflection [46]. However, Henderson at al. [12] presented a no-delay marking menu recently. They found that explicit mode switching is not a necessity for marking menus and their marking menu implementation rendered a GUI menu once a selection was made so their menu could be considered to be scale-dependent.

Therefore, the pEye technique [43] presented by Huckauf and Urbina provides the best basis for our gaze marking menu technique. We explore the implications of eye tracking sensing accuracy and ocular motor capabilities on design and use empirical evaluations to determine optimal menu item size, granularity, and depth for HMD environments. We also identify the main errors in this modality, and present a scale-invariant gaze marking menu, StickyPie, as a technique for mitigating Overshoot, one of the more prevalent errors we observed.

3 EXPERIMENT 1: DETERMINING THE EFFECTS OF PIE SIZE

With the pEye technique [15] a user can select an item by performing a saccade stroke crossing the border of the item's wedge (Figure 2 a). If the pie is not a leaf node, the next pie will appear at the center of the selected wedge's arc after the selection is made. From this basis input mechanism, we defined four different type of errors as shown in figure 2 (b) overshooting, (c) false activation, (d) incorrect selection, and (e) false activation by blinking.

We designed an experiment to determine the effect of pie size and three different menu hierarchies on the error occurrences. We first chose 4x4 layout as a baseline, then chose 2 possible variations; increasing depth (4x4x4 layout) and increasing granularity (8x8 layout). Here, *depth* indicates the number of levels of the menu hierarchy and *granularity* refers to the number of items on each level of menu.

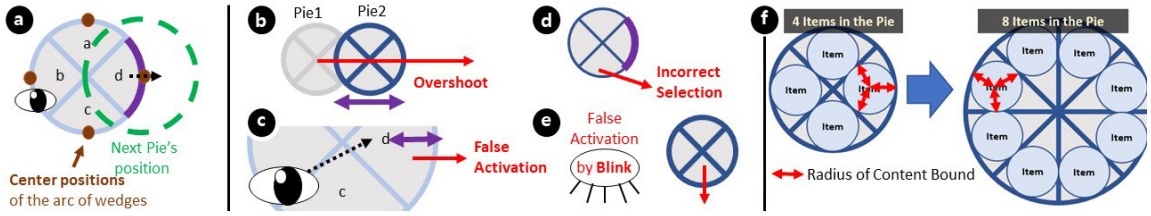

Fig. 2. (a) Gaze pie menu (re-illustrated from pEye [15]). When a user moves their eyes out to the border, the crossed wedge is selected then the next pie appears. Illustrations of how (b) overshoot errors, (c) false activations, (d) incorrect selections, and (e) blink errors occur. (f) An illustration of the content bound rule with examples of 4 and 8 pies where a virtual content bound fits into the wedges.

3.1 Type of Errors

The motivation of our first study is to assess how the size of the pie could affect the error rate. Thus, we first clarified how size affects overshooting, false activations, and incorrect selections. *Overshoot* error refers to unintended selections caused when a user's eye crosses the border of the next pie unintentionally. *False activation* refers to unintentional activations while a user is navigating a menu, occurring when the offset error in the eye tracking data is larger than the distance between the pie border and the menu content. *Incorrect selection* refers to a failure to select a desired item when a user intentionally crosses a border. It occurs when the offset is larger than the half length of the arc of the wedge intended to be selected. Finally, *Blink error* refers to unintentional selections caused by the user's eye moving down when they blink.

3.2 Content Bound Rule

Occurrence of the three types of error will be affected by the different dimensions of the pie menu: the radius of pie (overshooting), the length of arc (incorrect selections), and the distance of the content from the border (false activations). Because the length of arc (i.e., $2\pi \times r/g$) is determined by the radius (r) and the granularity (g) of pie, a pie with more granularity needs to have a larger radius than a pie with less granularity. However, there is no method for determining how large to set the radius for a pie with different granularity. Therefore, we propose that the virtual boundary of the content space (i.e., Content Bound) determine the radius which allows all wedges to touch the border (Figure 2f). Based on this rule, if the content bounds of pies are the same, more (or less) granularity makes more (or less) incorrect selection errors but less (or more) overshoot errors, thus both type of errors were roughly balanced whether increasing or decreasing granularity. Finally, we set the diameter of pies according to this content bound rule.

3.3 Task

Participants were asked to memorize a given *target* (which refers to the sequence to be entered of menu item selection on the hierarchical menu e.g. "CTS" in Figure 3b and f) before starting a trial. After memorizing the target, they could start a trial by pressing the trigger button on the controller (Figure 3a). After starting a trial, participants could invoke a pie menu by gazing at the initiating button, placed 7° below the center position of the viewing angle, for 0.2 seconds (Figure 3c), which would then cause the button to disappear and first layer of the pie menu to appear at the button's position (Figure 3d). Next, the participant selected an item sequentially, 2 or 3 times per trial. If the depth of the menu hierarchy was 3 (i.e., a 4x4x4 layout) or 2 (i.e., a 4x4 and 8x8 layout), a target would consist of 3 or 2 letters, respectively.

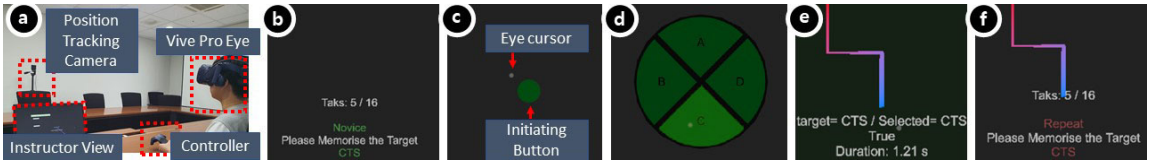

Fig. 3. (a) Experimental setup. (b) An example of a given target, "CTS", for novice trials (Experiment 1 and 2). (c) The eye cursor and menu initiation button. (d) The implemented pie menu. An item is highlighted with a brighter shade when a user gazes at it. (e) The feedback was presented after a trial is completed. (f) An example of a given target for experienced trials (Experiment 1 and 2).

For instance, figure 3 shows a 3 depth target ("CTS"). In this case, a participant was required to select "C", "T", and "S", subsequently. The target was randomly presented without redundancy from a pre-defined subset.

Participants were asked to select each target as quickly and accurately as possible. After the final selection was complete, they were provided with feedback about the result of the selection for 500 milliseconds (Figure 3e). If the trial as a success (or failure), the background turned green (or red), and the path of the sequential selection was visualized with a gradient line that transitioned from blue to red.

3.3.1 Novice trials and Experienced trials. As the level of expertise could affect the input performance, the level of expertise was explicitly controlled using two trial types: novice trials and experienced trials.

To ensure the participants knew nothing about the contents of the menu, letters were randomly assigned on the pie at the start of every novice trial. After a successful trial, participants performed an experienced trial using the same target on the same menu. Thus, it could be assumed that they knew where the given target was placed. To ensure this, a repeat-until success task methodology was used, wherein if they failed a trial, they would have to repeat it until they succeeded. At the start of each experiment, participants were told what kind of errors could possibly occur. Participants were asked to describe how they failed whenever the system indicated that they failed. In addition, if they thought the error did not fall into one of the pre-defined categories, they asked to describe the error in detail.

3.4 Apparatus

The pie menu interface was implemented in a VR environment using Unity and a Vive Pro Eye - an eye-tracking capable VR HMD. Each participant wore the HMD and held a controller while performing the experiments (Figure 3a).

The pie menu interface was rendered 1.5 meters away from the rendering camera. Gaze positions for each eye were computed using 3D gaze vectors and 3D eye positions, which were generated by the Vive Pro Eye at a sampling rate of 90 Hz. The gaze vectors and positions were smoothed using an averaging filter with a window of six. The 2D gaze positions were then computed on the rendered interface as ray-casted points using the 3D gaze vectors from each 3D eye position. The combined gaze position was also visualized as the average of the 2D gaze positions from both of the eyes using a gray-colored circle (i.e., eye cursor; 0.5° diameter).

3.5 Participants

We recruited twelve participants to participate in the study (i.e., 8 males, 4 females, mean age: 24 years, range: 18 to 28 years). Five of twelve participants wore glasses. Nine participants had experience using VR and one participant had experience of using eye-gaze input, however, none of the participants were regular users of such technology. For

this experiment as well as those proceeding, participants received a gift voucher amount of $18 USD per hours as compensation.

3.6 Design and Procedure

The study used a two-factor (i.e., Layout, Size) within-subject design with 3 layouts (i.e., granularity 4 - depth 2 (4x4), granularity 4 - depth 3 (4x4x4), and granularity 8 - depth 2 (8x8)), and 3 radius content bound sizes (i.e., 2° (small), 3° (medium), and 4.5° (large)). The diameters of the pies for each size were: granularity 4 (9.66°, 14.49°, and 21.73°) and granularity 8 (14.45°, 21.68°, and 32.52°) for 2°, 3°, and 4.5°, respectively.

The study was divided over three days, with each day consisting of three sessions for each radius content bound size. The day and session order were counterbalanced within participants using a balanced Latin square. Before each session, the eye tracker was calibrated using the default calibration method from the SteamVR software package and the participant performed a few trials for practice. This process lasted about a minute if the calibration result was satisfactory. The eye tracker was recalibrated if the participant ever felt that the eye cursor was inaccurate. Each session consisted of 3 blocks with 16 tasks per pre-defined gesture for each layout and lasted approximately 15 minutes.

At the end of each block of trials, participants were required to close their eyes for 2 seconds to minimize eye fatigue. The eye closure duration was also checked using the eye tracker. After 2 seconds of eye closure, a click sound was played to let the participant know that the 2 seconds was over. In addition, they were required to close their eyes for 10 seconds at the end of each block. At the end of the last session of each day, participants were asked to rate which size (i.e., small, medium, and large) was most and worst preferred to use for a given layout. A total of 10,368 successful novice and 10,368 experienced trials were recorded (i.e., 12 participant x 3 layouts × 3 size x 3 blocks × 16 tasks).

3.7 Results

We computed three types of input performance evaluation metrics, i.e., Completion Time, Error Rate, and Information Transfer Rate (ITR). Completion Time was measured from the time the pie menu interface was initiated (Figure 3c) until the final selection was completed (Figure 3e). Error Rate (0 to 100%) was computed as the number of failed tasks divided by the number of total tasks for each of the novice and experienced trials. Participants repeated trials until they were successful, but we only considered a task as a success if it was success at the first time. The number of failed tasks was also counted as a single failure even if a participant failed the same task multiple times. In addition, trials where the participant forgot a target were ignored. Finally, ITR (bit/s) was computed as [41]: $log2(N) + P \times log2(P) + (1 - P) \times log2(\frac{1-P}{N-1})/CompletionTime$, where P was the success rate (1 - error rate; 0 to 1) and N was the menu capacity (e.g., if a menu has 4 granularity and 3 depth, N= $4^3 = 64$). ITR represented the overall input efficiency because it takes into account completion time, accuracy, and menu capacity.

We ran the statistical analysis for novice trials and experienced trials, separately. We used repeated measure ANOVA (RM-ANOVA) and ran a number of paired-sample t-tests as post hoc tests using Bonferroni correction. If an interaction effect was significant, we compared each level of the layout to each level of the size as post hoc analysis.

All measurements meet the normality assumption and we used Greenhouse-Geisser adjustment for the violation of sphericity assumption.

3.7.1 Novice trials. In terms of **completion time**, the layout ($F_{(1.3,9.9)} = 53.193$, $p < 0.01$) and the size ($F_{(2,22)} = 18.122$, $p < 0.01$) had a significant effect, and their interaction was also significant; $F_{(4,44)} = 7.835$, $p < 0.01$. Because we found a significant interaction effect, we ran 18 post hoc tests for each level of layouts and for each level of sizes.

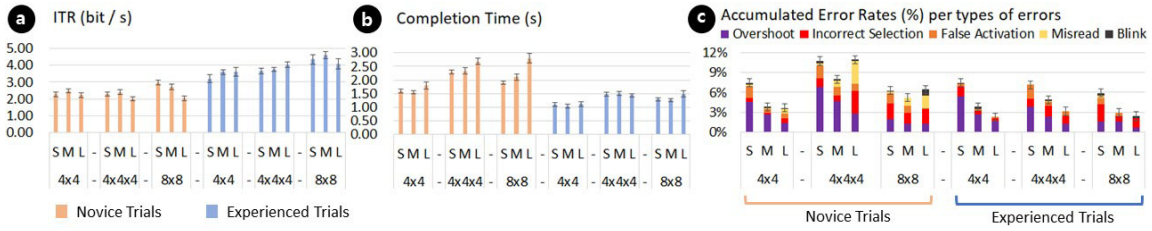

Fig. 4. Results of the experiment for (a) Information Transfer Rate, (b) Completion time, and (c) Accumulated error rates per type of error. Error bars depict standard errors. The X-axis indicates the size and layout; Size: S (Small; 2° of content bound), M (Medium; 3°), and L (Large; 4.5°).

Fig. 5. Reported errors versus the content bound size for novice and experienced trials.

When considering the layout differences, the 4x4 layout was significantly faster than the 4x4x4 layout (t_{11} = 7.589, 6.369, and 9.904, $p < 0.01$, 0.01, and 0.01) and the 8x8 layout (t_{11} = 5.722, 8.378, and 8.318, $p < 0.01$, 0.01, and 0.01) for small, medium, and large size condition, respectively. However, the difference between 4x4x4 and 8x8 was significantly different only for the small size; $t_{11} = 4.752$, $p < 0.05$. Regarding the size difference, completion time for the large size was significantly slower than small ($t_{11} = 6.272$, $p < 0.01$) and medium size ($t_{11} = 4.775$, $p < 0.05$) for the 8x8 layout, however, no significant effect was found for the other layouts.

In terms of **error rate**, only the layout had a significant effect; $F_{(2,22)} = 5.885$, $p < 0.01$. The effect of size was not significant and we found no interaction between the two factors. We also found the 4x4 layout to be significantly more accurate than the 4x4x4 layout ($t_{11} = 3.215$, $p < 0.05$) and the 8x8 layout ($t_{11} = 2.556$, $p < 0.05$), but the difference between the 4x4x4 and the 8x8 was not significant as a result of three post hoc tests with one-tailed assumption of paired-sample t-test.

In terms of **ITR**, the layout ($F_{(2,22)} = 4.985$, $p < 0.05$) and the size ($F_{(2,22)} = 11.010$, $p < 0.01$) had a significant effect, and the interaction effect was also significant; $F_{(4,44)} = 7.579$, $p < 0.01$. We then ran 18 post hoc tests. Regarding the layout difference, the 8x8 layout was significantly better than 4x4 ($t_{11} = 6.234$, $p < 0.01$) and 4x4x4 ($t_{11} = 3.890$, $p < 0.05$) layouts for the small size condition. For the size difference, the large size was significantly worse than small ($t_{11} = 6.746$, $p < 0.01$) and medium ($t_{11} = 4.897$, $p < 0.05$) sizes for the 8x8 layout condition. The other tests showed no significant results.

3.7.2 Experienced trials. In terms of **completion time**, only the layout factor had a significant effect ($F_{(2,22)} = 18.758$, $p < 0.01$). The size and size-layout interaction were not significant. We also found that the 4x4 layout was significantly faster than the 4x4x4 layout ($t_{11} = 6.253$, $p < 0.01$), and the 8x8 layout ($t_{11} = 4.842$, $p < 0.01$), but the difference between the 4x4x4 and the 8x8 was not significant as a result of paired-sample t-test.

Nov.	Size	Layout	Interaction	Remarks
CT	p< 0.01	p< 0.01	p< 0.01	The 8x8 layout tends to be faster than 4x4x4 but a significant difference was only found for the small size.
ER	-	p< 0.01	-	Size did not affect error rate significantly, due to the side effect of increasing size; poor visual acuity (Misreading) and poor eye tracking accuracy (Incorrect Selection).
ITR	p< 0.01	p< 0.05	p< 0.01	The large size tends to decrease ITR compare with the medium size.
Exp.	Size	Layout	interaction	Remarks
CT	-	p< 0.01	-	Size did not affect completion time for experienced trials
ER	p< 0.01	-	-	Layout did not affect error rate and the side effect decreased compare with novice trials; Less Misreading and Less Incorrect Selection
ITR	-	p< 0.01	p< 0.05	Size did not affect ITR for experienced trials

Table 1. Summary of the results of experiment 1. RM-ANOVA's results (p-values) remarks from the results; Two main effects (Size and Layout) and interaction effect (Interaction). Nov. refers to the results for the novice trials while Exp. refers to the results for experienced trials. CT, ER stand for Completion Time and Error Rate respectively.

In terms of **error rate**, only the size had a significant effect ($F_{(1.3,13.8)} = 9.753$, $p < 0.01$). We also found that the small size was significantly less accurate than the large size ($t_{11} = 5.326$, $p < 0.01$), but the difference between the small and the medium and between the medium and the large were not significant.

In terms of **ITR**, the layout was found to be significant ($F_{(2,22)} = 10.161$, $p < 0.01$) as well as size-layout interaction ($F_{(4,44)} = 3.314$, $p < 0.05$), but the size effect was not significant. We then ran 18 post hoc tests. For the layout difference, 8x8 were significantly better than 4x4 ($t_{11} = 4.230$ and 4.507, $p < 0.05$ and $p < 0.01$) for the small and medium size condition, respectively, and only for the medium size condition, 8x8 was better than 4x4x4 ($t_{11} = 4.830$, $p < 0.01$) layout. All the other tests showed no significant effect of size.

3.8 Summary

We summarize the results and the main findings from the experiment in Table 1. The experiment revealed that, for novice users, completion time increased as the size of pies increased. Some participants explicitly commented that they needed more time to find targets when the pie size increased. However, size did not influence completion time if a user knew where the items would be placed (i.e., experienced trials), even if the minimum eye movement distance was longer due to the increased pie size.

Second, as shown in figure 5, the initial assumption about the effect of size on different types of errors was confirmed. Most errors were confirmed to be incorrect selections and overshoot errors. For the granularity of 4, overshoots were the most frequent and, for the granularity of 8, there were more incorrect selections than overshoot errors.

Third, we found two side effects of larger-sized pies on the error rate. (1) For the novice trials, the *misread* errors increases as the size increases. *Misread* indicates an incorrect selection error when a participant misreads a letter in the pie menu. The poor visual acuity at the periphery region of the VR display increased the misread error. (2) For the novice trials, incorrect selection errors occur more frequently for the large size condition. This is caused by the inaccuracy of eye tracking in the periphery region, which has been reported by Alexandra *et al* [39] as well.

Last, we found that the medium size (3° of content bound) was best for all tested layouts (4x4, 4x4x4, and 8x8) when considering all measurements comprehensively. It was fast enough to compare with the small size and accurate enough to compare with the large size. Moreover, participants most preferred the medium-sized menu for all the layouts; nine, seven, and eight of twelve participants most preferred the medium size for 4x4, 4x4x4, and 8x8 respectively. Otherwise,

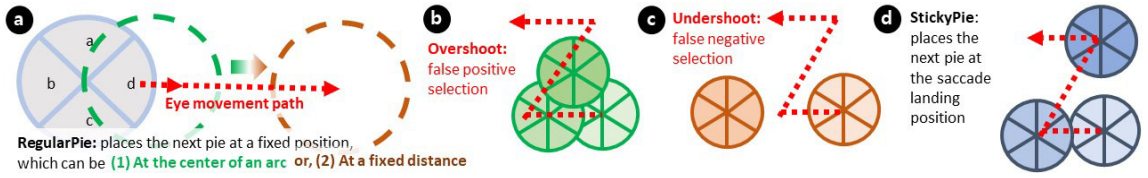

Fig. 6. Comparison illustration of RegularPie designs and StickyPie design. (a) Two options for placing the next pie for RegularPie (b) An example of overshoot error if the next pie is placed at the center position (c) An example of undershoot error if the next pie is placed at the distanced position. (d) The basic concept of StickyPie. The next pie appears at the saccade landing position.

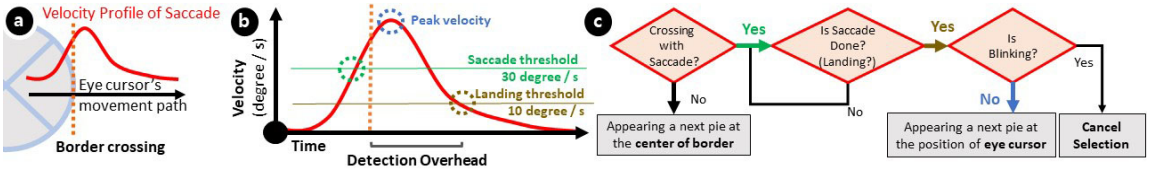

Fig. 7. Illustration of how StickyPie works. (a) Eye movement path and velocity profile of a user performing a border crossing gesture. (b) Example velocity profile of a saccade eye movement with the saccade and landing detection criteria. (C) Diagram of StickyPie algorithm.

one, zero, and one participants least preferred the medium size, respectively. From this result, the diameter of pies can be computed based on the content bound rule and menu granularity.

4 STICKYPIE: SUPPORTING SCALE-INVARIANT MARKING INPUT

The results from Experiment 1 suggested that overshoots and incorrect selections were the cause of the majority of errors, and we found that overshoot errors could impact the transition to expert usage. Because regular pie menus (i.e., RegularPie) appear at fixed positions, the user has to make their gesture stroke at a fixed scale (as Figure 6a and b). This creates a barrier to using gesture strokes, due to potential overshoot and/or the undershoot errors. Shendong and Ravin [46] similarly pointed this out for pen-based marking menus when the marking input occurs without the menu being displayed. Our findings show that this effect occurs for gaze marking menus as well, even if the menu is displayed. To address this, we designed StickyPie: a scale-invariant marking menu to support better novice to expert transitioning into using gesture input (Figure 6d). StickyPie treats every saccade as a single border crossing action, independent from the scale of the pie menu (scale-invariant), whereas in RegularPie, a long distance saccade could invoke multiple border crossing actions.

With StickyPie, the next pie is rendered at a saccade's landing position (Figure 7). To implement StickyPie, we implemented a saccade detection method using a velocity-based threshold algorithm [40] and we applied a peak detection method. If the velocity of an eye movement is over a certain threshold, we check that a saccade is starting (Figure 7b). After this, whenever the velocity is lower than the threshold, a check was performed to determine if the saccade was completed and if the eye cursor has landed in the same position.

It is important to set the landing threshold appropriately because, if it is too low, the pie will appear too late (Figure 7 b), otherwise, if it is too high, the landing position detection accuracy will be lower because the user's eyes will keep moving until the velocity reaches zero. To set the saccade threshold, we used the recommendations from Olsen et al.'s study [31], where a velocity threshold between 20°/s to 40°/s detected the beginning of a saccade. Results from our

Fig. 8. StickyPie implementation for experiment 2. (a) Content bound size and content margin for experiment 2. Diameters of (b) 4 Granularity pie, (c) 6 Granularity pie, and (d) 8 Granularity pie.

informal pilot studies concluded that for the StickyPie method, a saccade detection threshold of 30°/s and a landing detection threshold of 10°/s were the most appropriate.

In addition, we introduced a time limit for detecting the saccade landing position, to limit the maximum detection overhead (figure 7b) and limit the use of the periphery region. From the data of Experiment 1, the duration of 95% of the saccades for a border crossing action was under 200 milliseconds so this data was used as the time limit.

Whenever a border crossing event occurs, there is a check to determine if a saccadic eye movement has occurred. If the movement is not a saccade then the next pie will appear at the center position of the border. Otherwise, if a saccade is made, once it slows to the landing threshold, the next pie appears at the current eye cursor position, provided the user is not blinking. Two criteria are used to detect blinking: the peak velocity (i.e. threshold: > 800°/s; observed in 0.2% of the success trials from Experiment 1) and the amplitude of the saccade (i.e. > 45°; observed in 0.5% of the success trials). After the velocity slows down below the landing threshold, a check is made to determine if the saccade is a blink or not. If at least one of the above blinking criteria is over the threshold, the border crossing event is ignored.

Finally, some participants indicated that it was hard to learn how to cross a pie border without a visual target or guide. Therefore, we revised our design so that when a user looks at a pie item, a line on the item will be highlighted in white (Figure 8). The line is placed across the border of the item so the user can easily select the item by following the line across the pie border.

5 EXPERIMENT 2: EFFECTS OF DEPTH AND GRANULARITY ON STICKYPIE

Both depth and granularity are the main design factors within hierarchical menu designs. In Experiment 1, we investigated the effect of these factors on the occurrence of the most common errors while interacting with a gaze marking menu in VR, both in novice and expert mode. In the latter mode, these factors determine the shape of the gestures that can be used as input: the depth corresponds to the number of strokes within a gaze gesture, while the granularity determines the maximum possible angular discrimination at inflection points. In Experiment 2, we investigate further the effect of the depth and the granularity given the optimal size of a pie menu, derived from Experiment 1. Our aim is to confirm the effects of different depth and granularity for gaze marking menu design.

5.1 Participants, Task, and Apparatus

We recruited twelve participants from (redacted) University to participate in the study (i.e., 9 male, 3 female, mean age: 23 years; range: 19 to 28 years). Eight of the twelve participants wore glasses. Eleven participants had experience using VR and five participants had experience using an eye gaze input method. None of the participants were regular users of these technologies. Task and Apparatus were the same as that used in Experiment 1. The letters were placed at a distance of 3.5° from the border and the content bound was set to 2.5°.

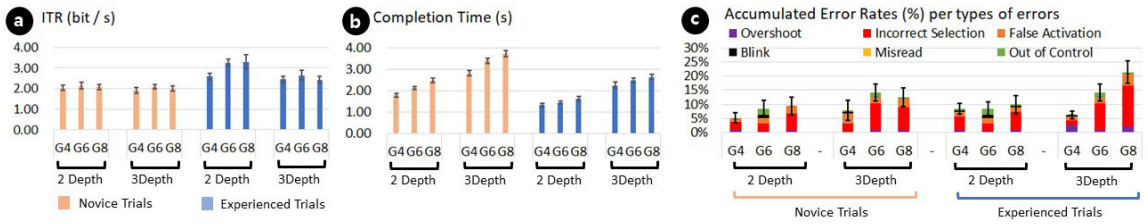

Fig. 9. The results of Experiment 2. (a) Information Transfer Rate (ITR), (b) Completion time, and (c) Accumulated error rates per type of errors. The Out of Control error indicates an error occurred because a pie appeared at a too distanced position to control. Error bars depict mean standard errors. G4, G6, and G8 in the X-axis indicate the granularity of 4, 6, and 8, respectively.

5.2 Design and Procedure

A two factor (Depth, Granularity) within-subject design was used, where Depth had 2 levels (i.e., 2, 3) and Granularity had 3 levels (i.e., 4, 6, and 8). The study was split into six sessions, one for each Granularity and Depth combination. The six sessions were grouped into three for each Granularity level. Grouped Sessions for the same level of Granularity were conducted sequentially. The order of the grouped granularity conditions was counterbalanced using a balanced Latin square and Depth order was counterbalanced within subject.

The procedure was similar to Experiment 1. The eye tracker was calibrated before each session then participants performed a few trials for practice. Each session consisted of a block with 16 tasks per predefined gesture, for each layout and lasted approximately 10 minutes. At the end of the experiment, participants were asked if they would choose increasing the depth or granularity. In total, 2,304 successful trials for each novice- and experienced trials were recorded (i.e., 12 participants × 2 depths × 3 granularities × 16 tasks).

5.3 Results

Similar to Experiment 1, we computed completion time, error rate, and information transfer rate (ITR) and used the same statistical methods. For both the novice- and experienced trials, error rate did not meet normality assumption, thus we ran an Aligned Rank Transform (ART) [44] for the error rate before conducing the RM-ANOVA. We summarize the results of RM-ANOVA and the main findings from the experiment in Table 2.

5.3.1 Novice trials. In terms of **completion time**, the RM-ANOVA shows that the depth ($F_{(1,11)} = 179.100$, $p < 0.01$) and the granularity ($F_{(2,22)} = 42.705$, $p < 0.01$) had a significant effect, but their interaction was not significant. We then ran 6 post hoc tests to compare the granularity effect for each depth. The results show that only the difference between the 6 vs 8 for 3-depth condition was not significant; 2-depth condition, 4 vs 6: $t_{11} = 4.220, p < 0.05$), 4 vs 8: $t_{11} = 10.506$, $p < 0.01$, and 6 vs 8: $t_{11} = 3.795, p < 0.05$, 3-depth condition, 4 vs 6: $t_{11} = 4.700, p < 0.01$, 4 vs 8: $t_{11} = 4.631, p < 0.01$.

In terms of **error rate** with ART and **ITR**, the RM-ANOVA showed no significant results.

5.3.2 Experienced trials. In terms of **completion time**, RM-ANOVA show that the depth ($F_{(1,11)} = 225.129$, $p < 0.01$) and the granularity ($F_{(2,22)} = 8.928$, $p < 0.01$) had a significant effect, but their interaction was not significant. We then ran 6 post hoc tests to compare the granularity effect for each depth, but we found nothing significant.

In terms of **error rate**, the RM-ANOVA with ART shows that the depth ($F_{(1,11)} = 14.197$, $p < 0.01$) and the granularity ($F_{(2,22)} = 3.962$, $p < 0.05$) had a significant effect, and their interaction ($F_{(2,22)} = 4.523$, $p < 0.05$) was also significant. We then ran 9 post hoc Wilcoxon signed rank tests, but we found nothing significant. In terms of **ITR**, the

Nov.	Granularity	Depth	interaction	Remarks
CT	p< 0.01	p< 0.01	-	CT increased as increasing Depth and/or Granularity
ER	-	-	-	Both Granularity and Depth did not significantly affect ER
ITR	-	-	-	Both Granularity and Depth did not affect ITR
Exp.	Granularity	Depth	interaction	Remarks
CT	p< 0.01	p< 0.01	-	CT seemed to be more affected by Depth than Granularity
ER	p< 0.01	p< 0.05	p< 0.05	ER increased compared with novice trials and was significantly affected by the factors.
ITR	-	p< 0.01	-	ITR decreased when increasing depth to 3 for Granularity of 6 and 8

Table 2. Summary of the results of experiment 2. RM-ANOVA's results (p-values) remarks from the results; Two main effects (Granularity and Depth) and interaction effect (Interaction). Nov. refers to the results for the novice trials while Exp. refers to the results for experienced trials. CT, ER stand for Completion Time and Error Rate, respectively

RM-ANOVA shows that only the Depth ($F_{(1,11)} = 23.568$, $p < 0.01$) had a significant effect. We then ran 3 post hoc tests to compare the depth effect for each Granularity; only the 6 granularity significantly different ($t_{11} = 6.362$, $p < 0.01$).

5.3.3 Preference. In terms of preference of increasing the depth or the granularity, participant's choices were split.

Five of the twelve participants preferred to increase the depth because it was easier to use, more accurate, less fatiguing, and a better input method overall. Interestingly, the other seven participants chose to increase the granularity due to the same reasons. This may be due to personal differences in ocular muscle controllability, however, a common opinion was that the Granularity of 8 was hard to use with a Depth of 3.

6 DESIGN GUIDELINES FOR MARKING MENUS USING EYE INPUT

In this section, we summarize the key findings from experiments 1 and 2. In experiment 1, we identified an optimal sizing for the pie menu. In experiment 2, we assessed the maximal usable depth and granularity for gaze marking menu. The following four design guidelines were derived from the results:

G1. Increase granularity rather than depth for better input efficiency. If a larger menu capacity is required, increasing granularity will result in lower completion times and increased accuracy.

As shown in figure 9 and table 2, granularity and depth did not affect input efficiency (ITR) for novice user. However, the ITR decreased as the depth increased for experienced user. This is likely due to the great increase of incorrect selection errors while increasing the depth, particularly for granularity levels of 6 and 8 (see second guideline below).

G2. Keep granularity at 6 or less for depths of over 2, as the depth of the menu hierarchy determines the number of inflection points in a gesture. If a gesture requires more than 2 inflection points (depth of 3), the granularity should be less than 6. Of note, the granularity of the menu determines the possible angular discrimination of the inflection points. If a gesture set needs multiple angles, the pie menu should be rotated to achieve the desired inflection angle for a particular gesture, without increasing the granularity.

G3. Place the content of a wedge at least 3° away from the border. This guideline was derived from experiment 1. As shown in figure 5, we found the false activation errors decreased by almost half with a 3° content margin compared with a 2° margin, but the error rate did not change when compared to 4.5° content margin. Thus, the content margin should be at least a 3° to mitigate the false activation errors.

G4. Set the pie size from 2.5° to 3° of content bound size. In experiment 1, we found that the content bound size was a better indicator of user satisfaction than the diameter size of the pie for the pie menus with varying granularities. Therefore, we recommend setting the size of the pie according to the content bound rule. As shown in figure 4, the

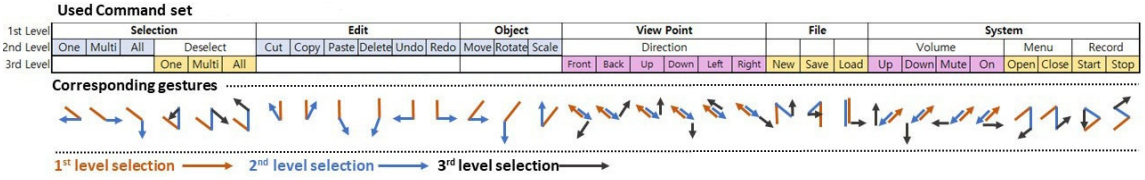

Fig. 10. Command set used in experiment 3 and the corresponding gestures for each command.

medium size (3°) provides better input speed than large size and better accuracy than small size, and participants mostly preferred the medium sized menu for all the layouts.

7 EXPERIMENT 3: COMPARING STICKYPIE WITH REGULARPIE

We expected that the scale-independent nature of StickyPie has the advantage of less overshoot errors and better support of the transition from novice user (GUI menu input) to expert user (gesture input). Experiment 3 evaluates whether there is an advantage over RegularPie with a study conducted over three days.

7.1 Menu and Gesture Design

The purpose of this experiment was to observe the actual development of expertise in terms of knowledge and control. Considering the knowledge expertise aspect, we designed a practical menu structure for a general application. To this end, a common command set that uses universal operating tasks in VR was adapted from Hou et al. [13]. Finally, we identified 32 commands under 6 categories (Figure 10) and designed a pie menu to take into account the semantic hierarchy of the commands and applied our proposed design guidelines from Section 6 (Figure 11) and general gesture design guidelines.

G1: Twelve frequently used commands used a depth-2 hierarchy (i.e., **Select**: *one, multi, all,* and **Edit**: *cut, copy, paste, delete, redo, undo*, and **Object**: *rotate, move, scale*).

G2 and **Provide cognitive aid while learning gestures [1]**: Seven commands were implemented using complex gestures with a depth-3 hierarchy, which were expected to be less frequently used and could provide cognitive aid by the shape of gesture. (i.e., 3 initial letter shape: **File**: *New, Load*, and **System** - Record: *Stop*, 4 iconic shape: **File**: *save* (shape of flag), **System** - menu: *open* (shape like 3D menu), *close* (shape of canceling action), **System** - record: *start* (shape of play button). Another seven commands provided semantic meaning based on their direction, where the gesture's final stroke corresponded to a directional selection (i.e., 6 **View point commands**, and **System** - Volume: *up, down, mute,* and *on*). For the remaining three commands that could not be associated with semantic gestures, we just applied our second guideline: **Select** - Deselect: *one, multi,* and *all*).

G3: The content was arranged such that the ends of the letters were 3.5° away from the border.

G4: Both RegularPie and StickyPie were then implemented using a 2.5° content bound size (Diameter size for 4 granularity pie is 12.07° and 15.00° for a 6 granularity pie).

We set RegularPie's menu size to be ±15° (Horizontal) × ±14° (Vertical). This size is within the comfort gaze zone guidelines (under 25° of gaze shift) reported in [37]. Figure 10 shows the designed gesture set and Figure 11 (a) shows the corresponding arrangement of pie menus.

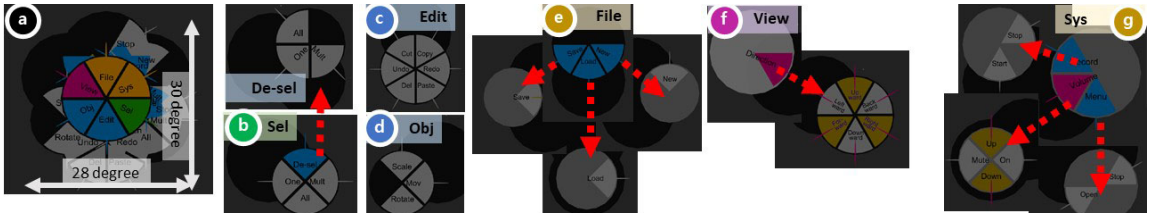

Fig. 11. Designed pie menu for experiment 3. (a) Required minimum space of the designed pie menu. Visualization with all pies overlapped at each appearing position for RegularPie. The top-most and center-positioned pie is the first layer of the menu hierarchy. (b) Pies for **Selection** command. (c) **Edit** commands. (d) **Object manipulation** commands. (e) **File** commands. (f) **View Point** Commands. (g) **System** commands. Red arrows indicate the next pies of each pie.

7.2 Participants, Task, and Apparatus

The apparatus was the same as Experiment 1. The task was also similar, but each task was not split between novice and experienced. We also did not shuffle the contents on the menu at the start of task, thus the participants could build their knowledge through repeated use, which is similar to the methods used with real-world menu-based interfaces.

We recruited four participants from each prior experiment (1 and 2), in the order of application first (5 male, 3 female, mean age: 25 years, range: 18 to 28 years). Four of the eight participants wore glasses. All of the participants were experienced in VR and three participants had experienced eye gaze input methods before.

7.3 Design and Procedure

We used a two-factor (2 Techniques x 9 blocks) within-subject design; StickyPie vs. RegularPie. The study was split into three days and each day consisted of two sessions for each technique. The session order was counterbalanced within participants.

At the start of the experiment, we briefly explained the menu structure. Before each session, we calibrated the eye tracker. On the first day, we gave 3 minutes to explore the menu freely at the start of each session. On the other two days, we only briefly checked that the calibration of the eye tracker was successful. Each session consisted of three blocks with 32 tasks. A total of 32 targets was given once per block in a random order. Sessions lasted approximately 12 to 15 minutes. As in experiment 1, participants were asked to close their eyes at the end of each trial and block. At the end of each session, we surveyed the eye fatigue as in the previous experiments, and surveyed perceived expertise using an adapted Borg10 scale (Figure 14 b). In addition, on the last day only, we surveyed the task load with a NASA-TLX questionnaire. We also surveyed preference of use, ease of use, and ease of learn using a five-point Likert scale. Finally, we conducted a recall test at the end of each day. All 32 commands were randomly placed on the recall test sheet. We obfuscated their hierarchical structure in the menu, i.e. if a target command is "edit-cut", only "cut" is shown on the sheet. Finally, we collected a total of 2,304 success trials (eight participants × three days × two techniques × three blocks × sixteen tasks) of success trials and 477 failed trials.

7.4 Results

Completion time, error rate, and information transfer rate (ITR) were computed and the same statistical methods as in Experiment 1 and 2 were used. All the measurements meet the normality assumption.

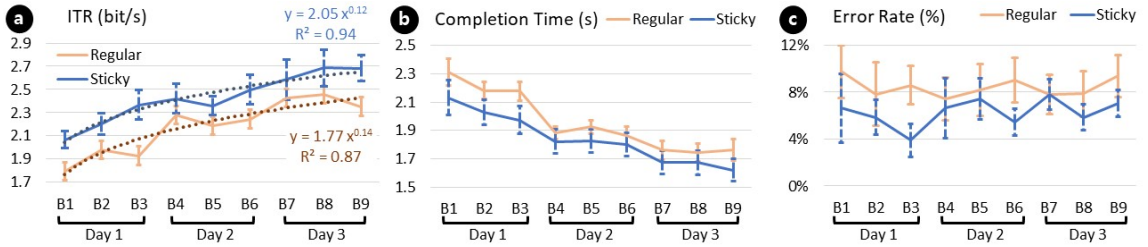

Fig. 12. Results of experiment 3. (a) Information Transfer Rate (ITR), the dotted lines indicate the corresponding learning curve. (b) Completion time, and (c) Error rates. Error bars mean standard errors. X-axis indicated blocks.

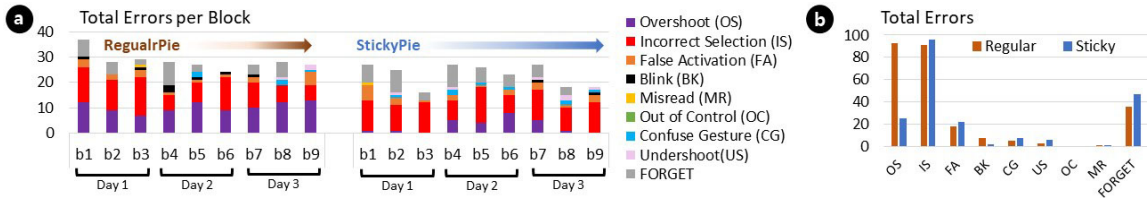

Fig. 13. The total number of errors per type of errors in experiment 3 (a) per block and (b) the sum of the errors from all blocks of trials. Unlike in the previous experiments, we counted errors in duplicate if a participant made multiple errors in a single task.

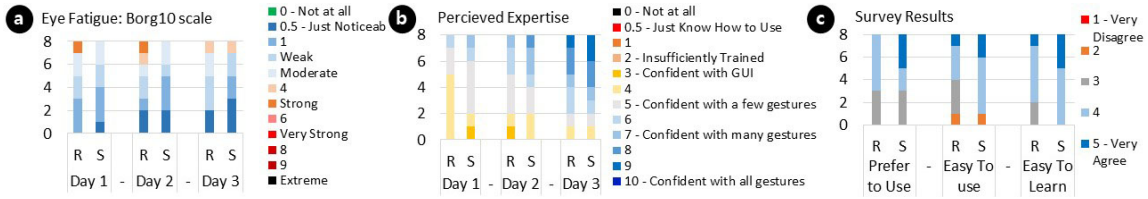

Fig. 14. Survey results from Experiment 3. (a) Eye fatigue, (b) Perceived expertise, and (c) Prefer to Use, Easy to Use, and Easy to Learn. The Y-axis indicates the number of responses.

7.4.1 Input Performances. In terms of **completion time**, RM-ANOVA showed only the block effect was significant; $F_{(8,56)} = 31.392$, $p < 0.01$. In terms of **error rate**, we found nothing significant. However, as shown in figure 13, the overshoot errors on StickyPie were greatly reduced compared with RegularPie. In terms of **ITR**, we found that the technique- ($F_{(1,7)} = 13.232$, $p < 0.01$) and the block effect ($F_{(8,56)} = 8.426$, $p < 0.01$) were significant, and interaction was not significant. At the last block, StickyPie (2.684 bit/s) was shown to have 14% better input efficiency than RegularPie (2.352 bit/s). Subsequently, we computed the learning curve on the ITR following the power law. A learning curve analysis found that StickyPie (learning curve: y= $2.05x^{0.12}$, $R^2 = 0.94$) was faster to learn than RegularPie (learning curve: y= $1.77x^{0.14}$, $R^2 = 0.87$).

7.4.2 Eye Fatigue and Perceived Expertise. We ran two-way (Technique x Day) RM-ANOVA with ART for the eye fatigue and the perceived expertise. In terms of the eye fatigue, we found nothing significant. In terms of the perceived expertise, only the day effect was significant ($F_{(2,14)} = 11.152$, $p < 0.01$).

7.4.3 Task Load, Easy to Use, and Easy to Learn. Mean task load of RegularPie (Overall: 37.75) and of StickyPie (35.48) were very similarly rated. The difference was not significant as determined by the Wilcoxon signed rank test. As shown in figure 14, StickyPie rated somewhat better than RegularPie for the all questionnaire results, however, only the question of easy to learn ($Z = -2.000$, $p < 0.05$) was significantly different based on the Wilcoxon signed rank test.

7.5 Qualitative Feedback

In terms of preference, six (one) of 8 participants gave a better score for SickyPie (RegularPie) and the other participant gave the same score for both. Six Participants who preferred StickyPie mentioned that, thanks to the saccade landing prediction, they were less concerned about the position of the eye cursor after a border crossing. They found that it was easier to input "gestures" using StickyPie and that they were more confident than with RegularPie. Three of eight participants liked the visual guidance line because it made their eye movement easier, while two participants preferred RegularPie. One participant pointed out that the visual guidance line was of great help initially, but by the last day, they could perform well without it. One participant preferred RegularPie because of the increased eye fatigue when a pie of StickyPie appeared in the periphery region.

In terms of eye fatigue, four (one) of 8 participants gave a better score for StickyPie (RegularPie) and the other three participants gave the same score for both on the last day. Participants indicated that RegularPie induced more eye fatigue because they had to be more conscious of their eye movements than with StickyPie, to avoid overshoot error. Otherwise, as mentioned above, StickyPie induced more eye fatigue when a pie appeared in the periphery region. Six participants specifically pointed out that the "System - Record - Stop" commands induced more eye fatigue using StickyPie. This is due to the fact that this gesture required using more of the peripheral region than other gestures. Additionally, participants felt more eye fatigue when performing upwards movements with their gaze. These effects were more pronounced while using StickyPie, because menus in StickyPie can appear further into the periphery of the user's FOV than RegualrPie. Thus, the comfort gaze zone needs to be carefully considered when designing gestures for StickyPie.

7.6 Novice - Expert Transition

In terms of knowledge expertise, we found that our marking menu design has good memorability. On the first day, the overall recall rate was 59%, but the recall rate rapidly increased: 85% on the second day and 96% on the third.

We also found that ITR for RegularPie and StickyPie continuously increased over the three days, and analysis of the learning curve revealed that StickyPie was faster to learn than RegularPie. We also found that overshooting errors greatly decreased (RegularPie: 93 times, StickyPie: 25 times) and seven of eight participants agreed that StickyPie was better to use for gesture input. These results indicate that our gaze-based marking menu works well in facilitating the novice-expert transition, improving RegularPie by supporting a scale-invariant marking input.

7.7 Micro Analysis of the Saccade Landing Estimation Algorithm

We used two criteria for our saccade landing detection method; (1) velocity threshold for the saccade landing (10°) and (2) time limit of detection (200 ms). We set the velocity threshold loosely and set the time limit to reduce the detection overhead (figure 7 b). Our method, thus, has an offset compare with the exact saccade landing position.

To validate our criteria and thresholds, we computed the landing position more accurately by using local minimum point of saccade velocity after next pie is appearing. We then computed the distance between the next pie's position and the eye cursor's position at the time of the local minimum point. We analyzed the data only from success trials and only

if the saccade made a selection, except last selection. As a result, mean distance was 0.288° (5 percentile: 0.014°, Medium: 0.053°, 95 percentile: 0.312°). Considering the radius of pies (6.0° for 4 Granularity and 7.5° for 6), we concluded that this is sufficiently accurate. Subsequently, we computed the distances for RegularPie, we then found that 5% (95 percentile: 5.856°) of border crossing gestures very closely landed on the border of 4 granularity pies (radius of 6.0°); mean: 2.61°.

8 GENERAL DISCUSSION AND FUTURE WORK

8.1 Adapting Design Guidelines for other AR/VR HMDs

Gaze marking menus are suitable for most applications that require a menu system, and are particularly valuable for offering quick, hands-free input. The guidelines we introduced in Section 6 are meant to be a practical aid to menu design, as shown in experiment 3 where we implemented an example menu for a VR application.

Our recommendations in terms of pie size (**G4**) and content margin (**G3**) are more susceptible to hardware limitations such as eye-tracking accuracy, and might have to be adapted to individual devices. In our studies, we used a Vive Pro Eye whose accuracy is reported to be between 0.5 and 1.1° [5]. Other eye-tracking-enabled VR HMDs report similar values; for instance, the FOVE VR has an accuracy of 1.15° [10], while the Pupil Labs Add-on is rated with an accuracy of 1.0° [25]. AR devices such as the Hololens 2 and Magic Leap One do not provide specifications of their eye-tracking accuracy, but considering other wearable eye tracker's accuracy [26], AR HMDs could provide similar eye-tracking fidelity. For instance, the SMI - ETG 2.6 glasses report an accuracy of 1.21°, and Tobii Pro Glasses 2 of 1.42°. We therefore believe that our parameters would be applicable for most HMDs. Given the above, we believe that our guidelines would apply for most HMDs, and they will inform better design parameters for gaze marking menus.

In addition, the size of a menu should be set such that one is mindful of the eye comfort. Sidenmark and Gallerson recommended taking into account the preferred eye-in-head motion range for the gaze interaction design [37], and they found that a user felt eye comfort for gaze shifts up to 25° using only eyes. Similarly, we found that the participants usually felt more eye fatigue for pies appearing in the peripheral region, further than ±20° from the central region. Thus, we also recommend that the menu placement should be up to ±20° from the central region of the user's field of view in consideration of eye comfort.

8.2 Investigating the use of Head-Combined Gaze

Head-combined gaze input is common in a VR HMD environment but in this paper we focused on the eye-only gaze modality. Adding head-combined gaze input to our current implementation could help mitigate eye fatigue in cases where a pie is appears in the peripheral region. It also could help mitigate the effect of inaccurate eye tracking [37, 39] and poor visual acuity (Misread errors) in the peripheral region, observed in experiment 1.

Head-combined gaze could be more comfortable [37] and efficient [38] for regions exceeding 25°. On the other hand, if the required menu size is under 25°, eye-only input could potentially provide better input efficiency [38] and has the advantage of less neck fatigue [33].

Additionally, VR interaction techniques combining eye and head input such as those presented by Sidenmark Gellerson [24, 38] may be leveraged to offer novel and efficient input methods for gaze-based marking menus. Both these areas will be further explored in our future work.

8.3 Reducing the majority of errors: Overshoot and Incorrect Selection

With StickyPie, we reduced the major source of error, overshoot, but we did not entirely eliminate it, as users still generated a number of overshoot errors. These errors are caused by the time threshold we introduced to reduce detection overhead (figure 7b). As mentioned, the 200 ms threshold was derived from our data, but it could be optimized in the future.

Our results also show that incorrect selection remains a large source of error. We reduced this error type by selecting the optimal size of pie based on experiment 1 results, which show that the 3° content bound enables accurate selections. However, in light of our experiment results, we expect that using saccade direction could further reduce the incorrect selection error, because it could mitigate the offset error of eye-tracking. This point will be addressed in our future work.

8.4 Supporting Error Recovery

Given the noisy signal in eye-tracking, it is important for the system to provide an easy means for error recovery. As indicated above, incorrect selection is the main type of error with StickyPie. Both incorrect selection and accidental invocation of menu items, and even the menu itself, should be easily reversed. The design of these interactions was beyond the scope of this work but are important next steps to extend it.

9 CONCLUSION

This paper presented StickyPie, a scale-invariant gaze-based marking menu optimized for AR and VR HMDs. We conducted three experiments in total. In the first two experiments, we derived design guidelines around optimal menu properties. Our first experiment explored what the ideal menu item size would be for efficient and accurate selection. We first formulated the content bound rule for the pie size determination in pies with various granularities. The results showed trade-offs between different types of error across the different menu item sizes, and identified Overshoot as a major source of error. To mitigate this error type, we then designed StickyPie, a technique that leverages saccade landing position estimation for determining the render location of the subsequent pie menu. In our second experiment, we investigated the effects of depth and granularity on StickyPie's usability.

Based on the results of experiments 1 and 2, we proposed the following design guidelines: **(G1)** Increasing granularity is better than increasing depth if input efficiency is most important. **(G2)** The 8 granularity is equally effective to the 6 granularity if the depth is 2, however, if a marking menu design requires more than 2 depth, one should set the granularity up to 6. **(G3)** Place the content of a wedge at least 3° away from the border. **(G4)** Set the size of the pie, using the content bound rule with a menu item size from 2.5 to 3° is optimal.

Finally, we designed a practical marking menu for VR applications following the above design guidelines and we evaluated RegularPie and StickyPie using the designed menu. The results of experiment 3 confirmed that StickyPie greatly reduced the overshoot errors and it was shown to be superior to RegularPie in terms of information transfer rate. StickyPie was also found to have better learnability than RegularPie.

This work demonstrated that our design guidelines obtained from empirical results and the StickyPie technique for scale-invariance could be applied to create a comfortable and efficient gaze-based marking menu interface.

REFERENCES

[1] Caroline Appert and Shumin Zhai. 2009. Using Strokes as Command Shortcuts: Cognitive Benefits and Toolkit Support. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Boston, MA, USA) *(CHI '09)*. Association for Computing Machinery, New York, NY, USA,

2289–2298. https://doi.org/10.1145/1518701.1519052

[2] 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* (Monterey, CA, USA) *(UIST '08)*. Association for Computing Machinery, New York, NY, USA, 37–46. https://doi.org/10.1145/1449715.1449724

[3] Jonas Blattgerste, Patrick Renner, and Thies Pfeiffer. 2018. Advantages of Eye-Gaze over Head-Gaze-Based Selection in Virtual and Augmented Reality under Varying Field of Views. In *Proceedings of the Workshop on Communication by Gaze Interaction* (Warsaw, Poland) *(COGAIN '18)*. Association for Computing Machinery, New York, NY, USA, Article 1, 9 pages. https://doi.org/10.1145/3206343.3206349

[4] Andy Cockburn, Carl Gutwin, Joey Scarr, and Sylvain Malacria. 2014. Supporting Novice to Expert Transitions in User Interfaces. *ACM Comput. Surv.* 47, 2, Article 31 (Nov. 2014), 36 pages. https://doi.org/10.1145/2659796

[5] HTC Corporation. 2020. VIVE Pro Eye Specs User Guide. https://developer.vive.com/resources/vive-sense/hardware-guide/vive-pro-eye-specs-user-guide/, last visited Dec. 2020.

[6] William Delamare, Teng Han, and Pourang Irani. 2017. Designing a Gaze Gesture Guiding System. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services* (Vienna, Austria) *(MobileHCI '17)*. Association for Computing Machinery, New York, NY, USA, Article 26, 13 pages. https://doi.org/10.1145/3098279.3098561

[7] Heiko Drewes, Mohamed Khamis, and Florian Alt. 2019. DialPlates: Enabling Pursuits-Based User Interfaces with Large Target Numbers. In *Proceedings of the 18th International Conference on Mobile and Ubiquitous Multimedia* (Pisa, Italy) *(MUM '19)*. Association for Computing Machinery, New York, NY, USA, Article 10, 10 pages. https://doi.org/10.1145/3365610.3365626

[8] Heiko Drewes and Albrecht Schmidt. 2007. Interacting with the computer using gaze gestures. In *IFIP Conference on Human-Computer Interaction*. Springer, 475–488.

[9] Augusto Esteves, Eduardo Velloso, Andreas Bulling, and Hans Gellersen. 2015. Orbits: Gaze Interaction for Smart Watches Using Smooth Pursuit Eye Movements. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology* (Charlotte, NC, USA) *(UIST '15)*. ACM, New York, NY, USA, 457–466. https://doi.org/10.1145/2807442.2807499

[10] inc. FOVE. 2020. FOVE0 Headset Specification. https://fove-inc.com/product/, last visited Dec. 2020.

[11] Aaron L. Gardony, Robert W. Lindeman, and Tad T. Brunyé. 2020. Eye-tracking for human-centered mixed reality: promises and challenges. In *Optical Architectures for Displays and Sensing in Augmented, Virtual, and Mixed Reality (AR, VR, MR)*, Bernard C. Kress and Christophe Peroz (Eds.), Vol. 11310. International Society for Optics and Photonics, SPIE, 230 – 247. https://doi.org/10.1117/12.2542699

[12] Jay Henderson, Sylvain Malacria, Mathieu Nancel, and Edward Lank. 2020. Investigating the Necessity of Delay in Marking Menu Invocation. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376296

[13] Wen-jun Hou, Kai-xiang Chen, Hao Li, and Hu Zhou. 2018. User Defined Eye Movement-Based Interaction for Virtual Reality. In *Cross-Cultural Design. Methods, Tools, and Users*, Pei-Luen Patrick Rau (Ed.). Springer International Publishing, Cham, 18–30.

[14] Anke Huckauf and Mario Urbina. 2007. Gazing with PEYE: New Concepts in Eye Typing. In *Proceedings of the 4th Symposium on Applied Perception in Graphics and Visualization* (Tubingen, Germany) *(APGV '07)*. Association for Computing Machinery, New York, NY, USA, 141. https://doi.org/10.1145/1272582.1272618

[15] Anke Huckauf and Mario H. Urbina. 2008. Gazing with PEYEs: Towards a Universal Input for Various Applications. In *Proceedings of the 2008 Symposium on Eye Tracking Research Applications* (Savannah, Georgia) *(ETRA '08)*. Association for Computing Machinery, New York, NY, USA, 51–54. https://doi.org/10.1145/1344471.1344483

[16] 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* (Santa Barbara, California) *(ETRA '12)*. Association for Computing Machinery, New York, NY, USA, 229–232. https://doi.org/10.1145/2168556.2168602

[17] Poika Isokoski. 2000. Text Input Methods for Eye Trackers Using Off-Screen Targets. In *Proceedings of the 2000 Symposium on Eye Tracking Research Applications* (Palm Beach Gardens, Florida, USA) *(ETRA '00)*. Association for Computing Machinery, New York, NY, USA, 15–21. https://doi.org/10.1145/355017.355020

[18] Toshiya Isomoto, Shota Yamanaka, and Buntarou Shizuki. 2020. Gaze-based Command Activation Technique Robust Against Unintentional Activation using Dwell-then-Gesture. In *Proceedings of Graphics Interface 2020* (University of Toronto) *(GI 2020)*. Canadian Human-Computer Communications Society / Societe canadienne du dialogue humain-machine, 256 – 266. https://doi.org/10.20380/GI2020.26

[19] Robert J. K. Jacob. 1991. The Use of Eye Movements in Human-computer Interaction Techniques: What You Look at is What You Get. *ACM Trans. Inf. Syst.* 9, 2 (April 1991), 152–169. https://doi.org/10.1145/123078.128728

[20] Yvonne Kammerer, Katharina Scheiter, and Wolfgang Beinhauer. 2008. Looking My Way through the Menu: The Impact of Menu Design and Multimodal Input on Gaze-Based Menu Selection. In *Proceedings of the 2008 Symposium on Eye Tracking Research Applications* (Savannah, Georgia) *(ETRA '08)*. Association for Computing Machinery, New York, NY, USA, 213–220. https://doi.org/10.1145/1344471.1344522

[21] Mohamed Khamis, Carl Oechsner, Florian Alt, and Andreas Bulling. 2018. VRpursuits: Interaction in Virtual Reality Using Smooth Pursuit Eye Movements. In *Proceedings of the 2018 International Conference on Advanced Visual Interfaces* (Castiglione della Pescaia, Grosseto, Italy) *(AVI '18)*. Association for Computing Machinery, New York, NY, USA, Article 18, 8 pages. https://doi.org/10.1145/3206505.3206522

[22] Gordon Kurtenbach and William Buxton. 1994. User learning and performance with marking menus. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 258–264.

[23] Gordon Kurtenbach, Thomas P. Moran, and William Buxton. 1994. Contextual animation of gestural commands. In *Computer Graphics Forum*, Vol. 13. Wiley Online Library, 305–314.

[24] Mikko Kytö, Barrett Ens, Thammathip Piumsomboon, Gun A. Lee, and Mark Billinghurst. 2018. Pinpointing: Precise Head- and Eye-Based Target Selection for Augmented Reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. ACM, New York, NY, USA, Article 81, 14 pages. https://doi.org/10.1145/3173574.3173655

[25] Pupil Labs. 2020. FOVE0 Headset Specification. https://pupil-labs.com/products/vr-ar/tech-specs/, last visited Dec. 2020.

[26] Jeff J Macinnes, Shariq Iqbal, John Pearson, and Elizabeth N Johnson. 2018. Wearable Eye-tracking for Research: Automated dynamic gaze mapping and accuracy/precision comparisons across devices. *bioRxiv* (2018), 299925.

[27] Päivi Majaranta, Ulla-Kaija Ahola, and Oleg Špakov. 2009. Fast Gaze Typing with an Adjustable Dwell Time. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Boston, MA, USA) *(CHI '09)*. Association for Computing Machinery, New York, NY, USA, 357–360. https://doi.org/10.1145/1518701.1518758

[28] Päivi Majaranta, Jari Laitinen, Jari Kangas, and Poika Isokoski. 2019. Inducing Gaze Gestures by Static Illustrations. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research Applications* (Denver, Colorado) *(ETRA '19)*. Association for Computing Machinery, New York, NY, USA, Article 75, 5 pages. https://doi.org/10.1145/3317956.3318151

[29] Päivi Majaranta, Kari-Jouko Räihä, Aulikki Hyrskykari, and Oleg Špakov. 2019. *Eye Movements and Human-Computer Interaction*. Springer International Publishing, Cham, 971–1015.

[30] Miguel A. Nacenta, Yemliha Kamber, Yizhou Qiang, and Per Ola Kristensson. 2013. Memorability of Pre-Designed and User-Defined Gesture Sets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France) *(CHI '13)*. Association for Computing Machinery, New York, NY, USA, 1099–1108. https://doi.org/10.1145/2470654.2466142

[31] Anneli Olsen and Ricardo Matos. 2012. Identifying parameter values for an I-VT fixation filter suitable for handling data sampled with various sampling frequencies. In *proceedings of the symposium on Eye tracking research and applications*. 317–320.

[32] Marco Porta and Matteo Turina. 2008. Eye-S: A Full-Screen Input Modality for Pure Eye-Based Communication. In *Proceedings of the 2008 Symposium on Eye Tracking Research Applications* (Savannah, Georgia) *(ETRA '08)*. Association for Computing Machinery, New York, NY, USA, 27–34. https://doi.org/10.1145/1344471.1344477

[33] Yuan Yuan Qian and Robert J. Teather. 2017. The Eyes Don't Have It: An Empirical Comparison of Head-Based and Eye-Based Selection in Virtual Reality. In *Proceedings of the 5th Symposium on Spatial User Interaction* (Brighton, United Kingdom) *(SUI '17)*. Association for Computing Machinery, New York, NY, USA, 91–98. https://doi.org/10.1145/3131277.3132182

[34] Vijay Rajanna and John Paulin Hansen. 2018. Gaze Typing in Virtual Reality: Impact of Keyboard Design, Selection Method, and Motion. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications* (Warsaw, Poland) *(ETRA '18)*. ACM, New York, NY, USA, Article 15, 10 pages. https://doi.org/10.1145/3204493.3204541

[35] 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* (Paris, France) *(CHI '13)*. Association for Computing Machinery, New York, NY, USA, 2325–2328. https://doi.org/10.1145/2470654.2481321

[36] Joey Scarr, Andy Cockburn, Carl Gutwin, and Philip Quinn. 2011. Dips and Ceilings: Understanding and Supporting Transitions to Expertise in User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC, Canada) *(CHI '11)*. Association for Computing Machinery, New York, NY, USA, 2741–2750. https://doi.org/10.1145/1978942.1979348

[37] Ludwig Sidenmark and Hans Gellersen. 2019. Eye, Head and Torso Coordination During Gaze Shifts in Virtual Reality. *ACM Trans. Comput.-Hum. Interact.* 27, 1, Article 4 (Dec. 2019), 40 pages. https://doi.org/10.1145/3361218

[38] Ludwig Sidenmark and Hans Gellersen. 2019. EyeHead: Synergetic Eye and Head Movement for Gaze Pointing and Selection. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 1161–1174. https://doi.org/10.1145/3332165.3347921

[39] Alexandra Sipatchin, Siegfried Wahl, and Katharina Rifai. 2020. Eye-tracking for low vision with virtual reality (VR): testing status quo usability of the HTC Vive Pro Eye. *bioRxiv* (2020).

[40] Samuel Stuart, Aodhan Hickey, Rodrigo Vitorio, Karen Welman, Stacy Foo, David Keen, and Alan Godfrey. 2019. Eye-tracker algorithms to detect saccades during static and dynamic tasks: a structured review. *Physiological Measurement* 40, 2 (feb 2019), 02TR01. https://doi.org/10.1088/1361-6579/ab02ab

[41] David E Thompson, Stefanie Blain-Moraes, and Jane E Huggins. 2013. Performance assessment in brain-computer interface-based augmentative and alternative communication. *Biomedical engineering online* 12, 1 (2013), 43.

[42] Geoffrey Tien and M. Stella Atkins. 2008. Improving Hands-Free Menu Selection Using Eyegaze Glances and Fixations. In *Proceedings of the 2008 Symposium on Eye Tracking Research Applications* (Savannah, Georgia) *(ETRA '08)*. Association for Computing Machinery, New York, NY, USA, 47–50. https://doi.org/10.1145/1344471.1344482

[43] Mario H. Urbina, Maike Lorenz, and Anke Huckauf. 2010. Pies with EYEs: The Limits of Hierarchical Pie Menus in Gaze Control. In *Proceedings of the 2010 Symposium on Eye-Tracking Research Applications* (Austin, Texas) *(ETRA '10)*. Association for Computing Machinery, New York, NY, USA, 93–96. https://doi.org/10.1145/1743666.1743689

[44] Jacob O. Wobbrock, Leah Findlater, Darren Gergle, and James J. Higgins. 2011. The Aligned Rank Transform for Nonparametric Factorial Analyses Using Only Anova Procedures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC, Canada) *(CHI '11)*.

ACM, New York, NY, USA, 143–146. https://doi.org/10.1145/1978942.1978963

[45] Jacob O. Wobbrock, James Rubinstein, Michael W. Sawyer, and Andrew T. Duchowski. 2008. Longitudinal Evaluation of Discrete Consecutive Gaze Gestures for Text Entry. In *Proceedings of the 2008 Symposium on Eye Tracking Research Applications* (Savannah, Georgia) *(ETRA '08)*. Association for Computing Machinery, New York, NY, USA, 11–18. https://doi.org/10.1145/1344471.1344475

[46] Shengdong Zhao and Ravin Balakrishnan. 2004. Simple vs. Compound Mark Hierarchical Marking Menus. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology* (Santa Fe, NM, USA) *(UIST '04)*. Association for Computing Machinery, New York, NY, USA, 33–42. https://doi.org/10.1145/1029632.1029639