Lattice Menu: A Low-Error Gaze-Based Marking Menu Utilizing Target-Assisted Gaze Gestures on a Lattice of Visual Anchors

Taejun Kim
HCI Lab, KAIST
Daejeon, South Korea
taejun.kim@kaist.ac.kr

Auejin Ham
HCI Lab, KAIST
Daejeon, South Korea
hamaj@kaist.ac.kr

Sunggeun Ahn
HCI Lab, KAIST
Daejeon, South Korea
topmaze@kaist.ac.kr

Geehyuk Lee
HCI Lab, KAIST
Daejeon, South Korea
geehyuk@gmail.com

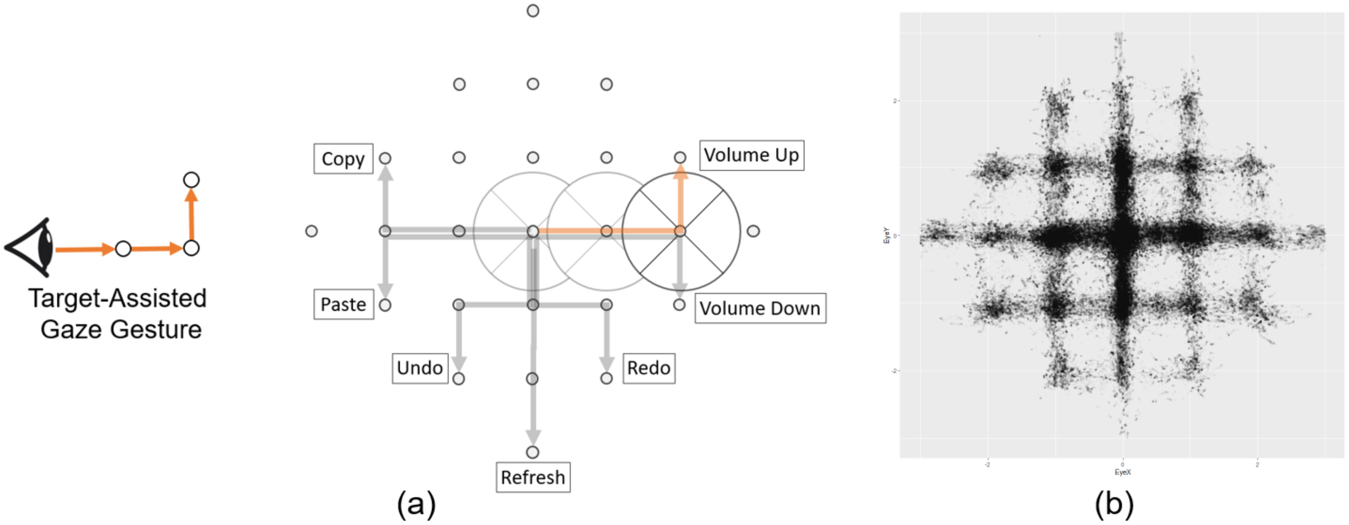


Figure 1: (a) Lattice Menu allows users to perform stable target-assisted gaze gestures by providing a lattice of visual anchors. (b) Distribution of eye saccade landing positions on Lattice Menu collected from User Study 2 (n ≈ 300,000).

ABSTRACT

We present Lattice Menu, a gaze-based marking menu utilizing a lattice of visual anchors that helps perform accurate gaze pointing for menu item selection. Users who know the location of the desired item can leverage target-assisted gaze gestures for multilevel item selection by looking at visual anchors over the gaze trajectories. Our evaluation showed that Lattice Menu exhibits a considerably low error rate (~1%) and a quick menu selection time (1.3-1.6 s) for expert usage across various menu structures (4 × 4 × 4 and 6 × 6 × 6) and sizes (8, 10 and 12°). In comparison with a traditional gaze-based marking menu that does not utilize visual targets, Lattice Menu showed remarkably (~5 times) fewer menu selection errors for expert usage. In a post-interview, all 12 subjects preferred Lattice Menu, and most subjects (8 out of 12) commented that the provisioning of visual targets facilitated more stable menu selections with reduced eye fatigue.

CCS CONCEPTS

• **Human-centered computing → Interaction techniques**.

KEYWORDS

Eye Tracking, Gaze-Based Interaction, Marking Menu, AR/VR

ACM Reference Format:
Taejun Kim, Auejin Ham, Sunggeun Ahn, and Geehyuk Lee. 2022. Lattice Menu: A Low-Error Gaze-Based Marking Menu Utilizing Target-Assisted Gaze Gestures on a Lattice of Visual Anchors. In *CHI Conference on Human Factors in Computing Systems (CHI '22), April 29-May 5, 2022, New Orleans, LA, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3491102.3501977

1 INTRODUCTION

Hands-free system control with a gaze-based menu [1, 16, 35] has become more viable with the advancement of eye tracking technology. In particular, the integration of eye trackers into major augmented reality (AR) and virtual reality (VR) headsets, such as Hololens 2, Vive Pro Eye, and FOVE VR, makes a gaze-based menu more feasible for AR/VR applications.

A marking menu [17, 19] offers fast selection skills for expert users by drawing a mark in the direction of the desired menu item. Users who already know the location of the desired item in the

CHI '22, April 29-May 5, 2022, New Orleans, LA, USA

ACM ISBN 978-1-4503-9157-3/22/04…$15.00
https://doi.org/10.1145/3491102.3501977

menu do not have to navigate a hierarchical menu [4, 27] step by step. The gradual transition from novice to expert is supported as physical actions made by a novice (i.e., navigating the GUI menu for selecting the desired item) become rehearsal of expert gesture skills [18]. Transferring these benefits of a marking menu into the gaze input domain, i.e., gaze-based marking menus, is a valuable research goal.

Border-crossing selection [35] (Figure 15a) is a first step for realizing gaze-based marking menus. It allows users to select a menu item by crossing the border of a pie slice with their eye movements. Expert users can perform a fast selection skill with successive border-crossing eye movements (i.e., zig-zag gaze gestures) that mimic the hand-drawn mark of the original marking menu. A gradual transition from novice to expert is also supported because users can memorize the required gaze gesture path up to the target command over time. Urbina et al. [35] reported that the border-crossing selection method produced a better performance than a dwell-based method.

However, using gaze gestures "in the air" (i.e., border-crossing [35]) for a marking menu can be error-prone because there are several unwanted options located adjacent to a single wanted item in a menu structure. In previous studies on eye gaze gestures [6, 8, 10, 22, 25, 37], researchers have proposed the use of visual targets to anchor the gaze for accurate eye movements. EyeDraw [8] uses a grid of dots and was reported to be helpful for eye-drawing tasks. In the EyeWrite interface [37], small visual anchors are placed at the four corners of the input box to support precise eye-typing movements. Inspired by previous studies [8, 10, 22, 37], we attempted to integrate visual anchors into the gaze-based marking menu technique to facilitate stable eye control.

Herein, we present Lattice Menu, a gaze-based marking menu with a lattice of visual anchors for supporting accurate gaze pointing. As depicted in Figure 4, users can select a menu item by looking at the corresponding visual anchor outside the border of the pie slice. Experienced users, who already know the location of the desired item in the menu, can perform target-assisted gaze gestures over the lattice of visual anchors. In this study, we implemented and evaluated Lattice Menu technique in a VR environment.

In the remainder of this paper, we review related works, describe the design of Lattice Menu, and explain the overall structure of the user study. Preliminary studies are first described to demonstrate the process of optimizing the Lattice Menu technique. Subsequently, we evaluate the performance of the designed Lattice Menu on various menu layouts (tested structures, 4 × 4 × 4 and 6 × 6 × 6; menu radius, 8, 10, and 12°) in User Study 1. Finally, we compare the performance of Lattice Menu with the traditional border-crossing marking menu [35] that does not utilize visual targets in User Study 2.

The main contributions of this study are as follows:

- We introduce Lattice Menu, a gaze-based marking menu technique that supports target-assisted gaze gestures on a lattice of visual anchors.
- Through empirical evaluation, we show that Lattice Menu provides a stable user control with a remarkably low error rate (~1%).

2 RELATED WORK

2.1 Marking Menu Technique

A marking menu [19], initially proposed in the hand-based input domain (e.g., stylus and mouse), is a radial menu that allows users to make a selection by drawing a mark to the desired item. The original marking menu offered two different modes for selection; novice users wait for a certain amount of time after pressing the input device to trigger the display of the GUI menu (*menu mode*), whereas expert users simply draw a mark to the desired item without waiting (*mark mode*). The marking input is scale-invariant through the post-processing of gesture decoding.

Meanwhile, Henderson et al. [7] recently questioned the necessity of the delayed display and mode separation. They failed to reveal the obvious performance advantage compared to the immediately displayed design without a mode separation. We are now curious about the essential aspects of a marking menu.

We believe that there are two fundamental benefits in a marking menu; (1) support for faster menu selection through a gestural input, i.e., a mark, and (2) the principle of rehearsal [19], i.e., a gradual transition from a novice to an expert. We believe that the separate modes or the scale-invariance may be just a means to achieve these goals.

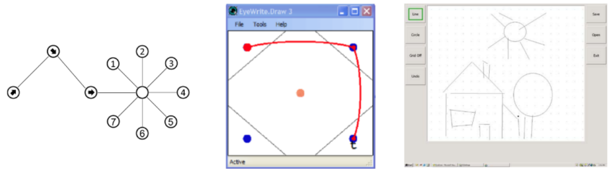
Lattice Menu does not separate modes for selection, nor does it have scale-invariant features. However, we call Lattice Menu a marking menu because it supports (1) expert skills through a gestural input (i.e., target-assisted gaze gesture) and (2) the principle of rehearsal; eye movements for novice users' selection become rehearsals for expert skill. Lattice Menu is a new marking menu designed for the eye gaze input domain.

2.2 Gaze-Based Marking Menu Techniques

Despite their potential benefits, only a few studies [1, 9, 35] have explored gaze-based marking menu designs. Border-crossing selection [35] was the first step for a gaze-based marking menu technique. The border-crossing marking menu allows users to select a menu item by crossing the border of the pie slice with eye movements (Figure 15a). Expert skill with successive border-crossing eye movements, i.e., gaze gestures, can be supported. This approach was also reported to be effective for preventing an unintentional selection compared to dwell-based menu selection [21, 34].

Huckauf et al. [9] introduced the *pEYE* depicted in Figure 6b, which employs a pizza-crust-like area for gaze pointing guidance. Their study on *pEYE* did not focus on an investigation of the marking menu technique (e.g., expert-level hierarchy); however, the idea of visual guidance for stable eye control is valuable. We compare the performance of the *pEYE* design with our approach in Section 3.4.

StickyPie [1] was recently proposed to overcome the overshooting problem (i.e., unwanted menu selection owing to the overshot border-crossing eye movement) of the traditional technique by locating the sub-menu on the estimated saccade landing position. An evaluation of StickyPie showed a faster menu selection time and a lower error rate than the traditional technique [35]. We considered performance comparison of ours with StickyPie, but we had a practical difficulty. Careful setting of parameters is crucial for the optimal performance of StickyPie, but the parameters used in



(a) Majaranta et al.

(b) Wobbrock et al. (EyeWrite)

(c) Hornof et al. (EyeDraw)

Figure 2: Previous studies that adopted visual targets for stable gaze gestures. (a) [22], (b) [37], and (c) [8]

the published paper could not be adopted because the interaction context of the StickyPie differs from that of ours (e.g., eyes-only input vs. head-combined gaze input). As our focus in the current study is to validate the effectiveness of the provisioning of visual targets, we decided to compare ours with the traditional border-crossing marking menu [35] in this study. Instead, we compared the reported performances of StickyPie with ours in Section 8.4.

2.3 Target-Assisted Gaze Gesture

Majaranta et al. [22] noted that *"Having something to look at not only guides the gaze but also avoids the need to perform unnatural, potentially uncomfortable gaze gestures"*. Previous studies [6, 8, 10, 25, 37] have employed visual targets for achieving stable gaze gestures, i.e., target-assisted gaze gestures. Drewes et al. [6] compared three gaze gestures with and without visual targets and reported that many subjects failed to perform complex gaze gestures without the help of visual targets. As shown in Figure 2b, EyeWrite [37] utilized small point targets in the four corners to support stable gaze gestures for text entry. EyeDraw [8] shown in Figure 2c utilized a grid of dots for eye-drawing tasks and reported that it was helpful to EyeDraw users.

Inspired by previous studies, Lattice Menu was designed to allow users to perform target-assisted gaze gestures over the lattice of visual anchors. Compared to gaze gestures "in the air" without any target to look at [13], users can achieve more stable menu selections. In the preliminary studies of this paper, we further explore several design dimensions of visual anchors to better support the target-assisted gaze gestures.

3 LATTICE MENU

Lattice Menu is a gaze-based marking menu with a lattice of visual anchors that helps accurate gaze pointing. Users can select a menu item by looking at the corresponding visual target (i.e., visual anchor) outside the border of the pie slice. When the eye gaze enters the *Item Selection Zone* (Figure 3a) around the visual anchor, a corresponding menu item is immediately selected. Experienced users, who already know the required trajectory to the desired item, can perform target-assisted gaze gestures over the lattice of visual anchors as depicted in Figure 4b. In this study, we implemented and evaluated the Lattice Menu technique in a VR environment, under a three-level hierarchical menu structure, covering 64 commands for a 4 × 4 × 4 menu structure, and 216 commands for a 6 × 6 × 6 menu structure.

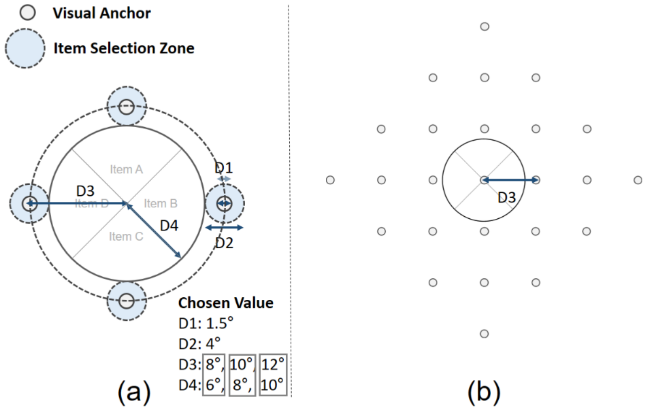


Figure 3: (a) Controllable parameters of Lattice Menu. (b) 4 × 4 × 4 Lattice Menu. D1, width of visual anchor; D2, width of *Item Selection Zone*; D3, effective radius of Lattice Menu, and D4, radius of visual pie.

3.1 Head-Fixed Menu vs. World-Fixed Menu in AR/VR

There can be two types of menus in the AR/VR environment: (1) a menu that is fixed at the headset display, *Head-Fixed Menu* (i.e., a menu that follows users' head movement) or (2) a menu located in a fixed coordinate of the AR/VR world, *World-Fixed Menu*. An eyes-only input is used for the *Head-Fixed Menu*, and a head-combined gaze input is used for the *World-Fixed Menu*.

Validation in the *World-Fixed Menu* setup has the advantage of being compatible with an external, stationary eye tracker (e.g., for smart TV) environment, because a head-combined gaze input is utilized in both cases. By contrast, validation in the eyes-only input (*Head-Fixed Menu*) setup also has its advantage, being the only possible option when 3D sensing for the surrounding space is incapable of (e.g., smart glasses without 3D sensing capability). In this study, we decided to validate our Lattice Menu technique in the *World-Fixed Menu* setup.

3.2 Design Rationale

The controllable parameters of Lattice Menu and our chosen values are shown in Figure 3a. In this section, we describe the rationale of our design. First, we decided to use a circular-type visual anchor, which is commonly applied in gaze pointing tasks [8, 10, 22, 31, 37]. We decided to set *D1* (the width of the visual anchor) to 1.5° referring to the fixation target size adopted in previous vision studies [31]. In addition, *D2* (the width of the *Item Selection Zone*) was determined to be 4° considering the advertised tracking accuracies of major VR headsets [11, 12, 26], which are better than 2°; the advertised tracking accuracy of the FOVE [11] is 1.15°, and that of HTC Vive Pro Eye [12] is 0.5-1.1°.

For the menu size (*D3*: the effective radius of Lattice Menu), we used 10° as the base size, allowing a single-level menu selection to be performed with a comfortable eye-only movement (i.e., within a range of ± 15° [24]), and multi-level target-assisted gaze gestures to be performed with a comfortable head-combined eye movement

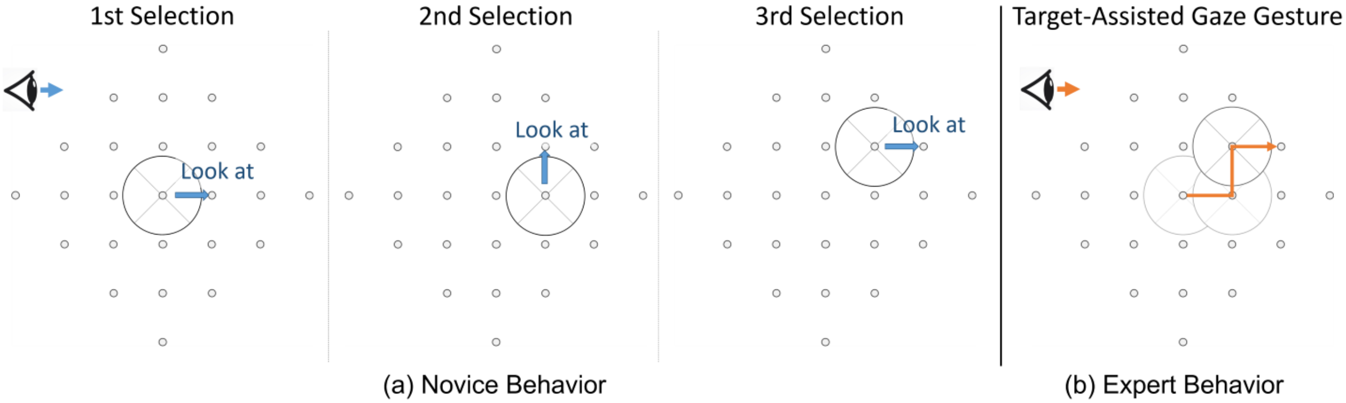


Figure 4: Illustration of (a) novice user behavior and (b) experienced user behavior of Lattice Menu.

[29]. Preliminary studies were conducted with the base size (10°). In User Studies 1 and 2, three menu sizes, i.e., 8° (small), 10° (base), and 12° (large), were tested for a broader evaluation.

Finally, for *D4* (the radius of a visual pie), the need for this parameter has to be explained first. The reason we do not simply set *D4* same with D3 (i.e., the reason we make visual pie smaller than the effective radius of Lattice Menu) is to separate the area of the visual pie from the *Item Selection Zone* as shown in Figure 3a. In an informal pilot study, unintentional selection errors frequently occurred while reading item labels when D4 was the same as D3. However, after we separated the visual pie from the *Item Selection Area*, the errors were effectively reduced. This observation implies that the visual pie can "capture" eye gaze of the user inside the boundary, which is consistent with the previous finding that the "eyes prefer to stay within the same object" [33].

3.3 Distance from Content to Item Selection Zone

If the menu item contents (e.g., label texts) are located too close to the *Item Selection Zone*, unintentional selection errors can easily occur during a visual search of the items. For the contents placement guidelines, we conducted a pilot study (n = 10) to examine the effect of the distance from the menu item content to the *Item Selection Zone* on the error rates. We collected 540 *Novice Trials* (described in Section 4.3) for each 1° to 5° *Distance* conditions under 4 × 4 × 4 menu structure with the base menu size (10°). The order of the conditions across subjects was counterbalanced using a Balanced Latin Square. The results are shown in Figure 5 and a Friedman test revealed that the effect of the *Distance* on the error rates was significant ($\chi^2(4) = 15.653$, $p < .005$). The results imply that a *Distance* of 3° or more is required to prevent unintentional errors during the visual search of the menu contents, which is consistent with the previous gaze-based menu guidelines [1]. From this observation, we fixed the *Distance* to 3° in the remaining studies.

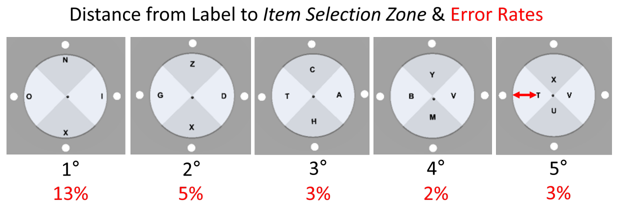


Figure 5: Distance from item label to *Item Selection Zone* and the observed error rates for each condition.

3.4 Comparison with pEYE

Our design was inspired by *pEYE* [9] shown in Figure 6b. The *pEYE* employed a pizza-crust-like area outside the border for gaze pointing guidance. We decided to compare the performance of the two designs through a pilot study. We expected that a more localized point target would better support a stable gaze control than the pizza-crust-like target of *pEYE*.

To validate our expectations, we conducted a 2-day in-lab pilot study (n = 6). We collected 120 *Novice Trials* and 240 *Experienced Trials* (described in Section 4.3) of the menu selection task for each technique under the menu layout depicted in Figure 6. The order of the conditions was counterbalanced across the subjects. As a result, we observed generally higher error rates in *pEYE*. Particularly for *Experienced Trials*, *pEYE* showed error rates six-times higher than Lattice Menu (1.1 vs. 6.1%), which validates our initial expectation and the superiority of the Lattice Menu design.

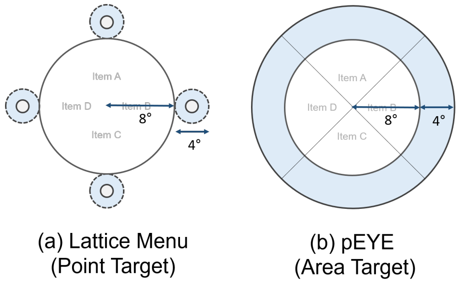


Figure 6: Areas for item selection in (a) Lattice Menu and (b) *pEYE* [9]

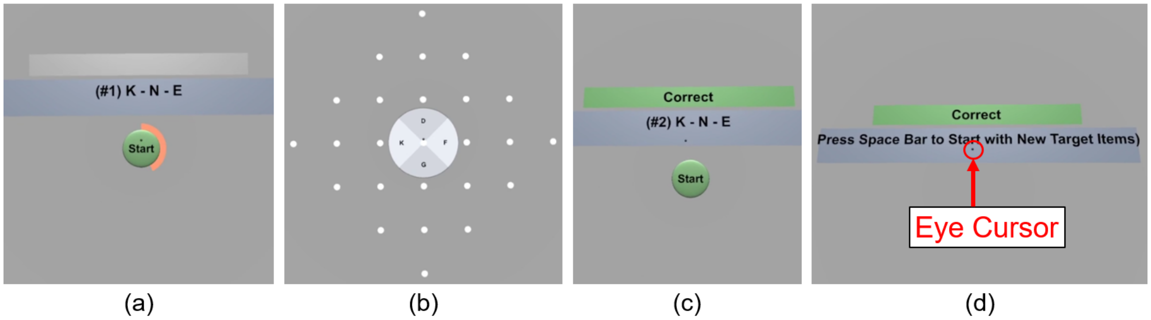


Figure 7: Sequence of user tasks in VR testing program. (a) After memorizing given item sequence (i.e., K-N-E), subjects dwell over start button, and (b) select items using Lattice Menu. (c) Results are shown after each trial (i.e., correct). (d) After repeating each sequence four times, subjects pressed space bar to obtain new target items.

4 USER STUDY OVERVIEW

In preliminary studies, we explored several design dimensions of visual anchors to optimize and finalize our Lattice Menu design. In User Study 1, we evaluated the designed Lattice Menu across various menu layouts (tested structures of 4 × 4 × 4 and 6 × 6 × 6, and tested radii of 8°, 10°, and 12°). In User Study 2, we compare the performance of Lattice Menu with the traditional border-crossing marking menu [35] that does not utilize visual targets.

4.1 Participants and Participant Screening Process

A group of 12 subjects (4 females, mean age of 24.0 and SD of 5.3) participated in preliminary study 1 (days 1 and 2), preliminary study 2 (day 3), and User Study 1 (day 4) on consecutive days. Afterward, for a fair comparative study, a new group of 12 subjects (three females, mean age of 22.0 and SD of 2.0) with no prior experience of Lattice Menu was recruited for User Study 2. Each subject was paid 15,000 KRW (≈ 13 USD) per hour as a reward.

The width of the *Item Selection Zone*, as shown in Figure 3, was chosen at 4°. This decision was based on the advertised tracking accuracies of major VR headsets, which are better than 2°; the advertised tracking accuracy of the FOVE [11] is 1.15°, and that of HTC Vive Pro Eye [12] is 0.5-1.1°. However, FOVE that we used in this study exhibited tracking errors larger than 2° for some users. Therefore, we had to screen two and four applicants from the first and second groups, respectively, who repeatedly showed tracking errors larger than 2° in a 9-point accuracy test we developed.

4.2 Common Apparatus

Lattice Menu was implemented in a VR environment using Unity. FOVE [11], a VR headset with an integrated eye tracker was used. FOVE has a display resolution of 2560 × 1440 pixels and a field of view of 100° with a frame rate of 70fps. The built-in binocular eye tracking system of FOVE has a sampling rate of up to 120 Hz.

The ray casting of the eye gaze was computed using the gaze origin and gaze vector collected from each eye. The 2D gaze positions hit on the rendered menu plane were computed using the gaze rays from both eyes. A combined gaze position was calculated by averaging left/right 2D gaze positions. The eye cursor indicating combined gaze position was visualized with a dark-gray circle (diameter of 0.5°) as shown in Figure 7d.

4.3 Common Task and VR environment

Figure 7 shows how subjects performed the task using the VR testing program. First, three menu items (e.g., K-N-E) to be selected in each menu level are shown. After memorizing the target items, subjects can stare at the start button for 1 s to start a task trial. After the menu selection is made, the result of the trial is provided, as shown in Figure 7c (i.e., correct or incorrect)

The same target items (e.g., K-N-E) were repeated four times to simulate an expert user performance. In the first trial of repetition, subjects should navigate the GUI menu to find the target items. In the latter trials of repetition, subjects were able to follow the known trajectory up to the target item they found before. They were informed about the strategy of the target-assisted gaze gesture, consecutively looking at visual anchors along the way. After four repetitions with the same target items, subjects could obtain a new target item by pressing the space bar (Figure 7d). When the space bar was pressed, new randomized target items were provided to subjects. The item (i.e., letters) arrangement of the menu was also randomized.

In this study, we denote the first (#1) trials of repetition as *Novice Trials*, and the third and fourth (#3 and #4) trials as *Experienced Trials*. To test a range of menu positions, we randomly chose an appropriate number of target item paths from *0-bent*, *1-bent*, and *2-bent*, depicted in Figure 8, for testing.

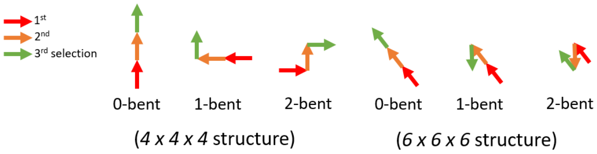


Figure 8: Different types of target item paths used in our studies.

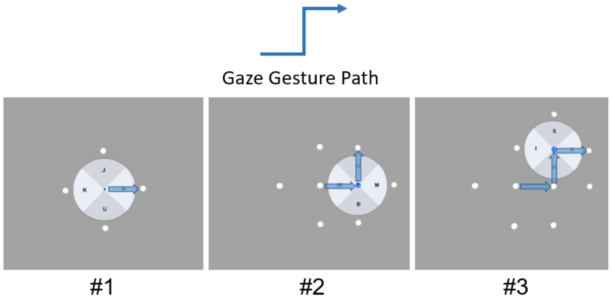


Figure 9: Lattice Menu with *Progressive Unfolding* effect; *Progressive Lattice Menu*

4.4 Common Procedures

Subjects sat on a chair and were instructed on how to conduct the task through a demonstration video before starting the experiment. After putting the headset, the eye tracker was calibrated using the default calibration method of FOVE, and subjects could request a recalibration when they felt a mismatch with the position of the eye cursor. They were instructed to perform the task as accurately and quickly as possible and were informed that they could use eye and head movements together. Subjects took a short break between each block or experimental condition to relieve eye fatigue.

5 PRELIMINARY STUDY

Preliminary user studies were conducted to finalize our Lattice Menu design. Here, we explore several design possibilities that can significantly affect the usability of Lattice Menu. Experiments were conducted under the baseline menu layout (menu structure of 4 × 4 × 4, effective radius (*D3*) of 10°, and visual pie radius (*D4*) of 8°).

5.1 Preliminary Study 1: Progressive Unfolding Effect

Visual distraction owing to the lattice of visual anchors on the screen is a critical usability problem of Lattice Menu. As a solution, we devised the *Progressive Unfolding* effect. When the effect is activated, Lattice Menu does not show all lattices but progressively unfolds the visual anchors used for the item selection of the current menu level. For instance, as shown in Figure 9, four visual anchors at menu level 1 are visible from the beginning, and, when a user looks at one visual anchor to make a selection, another four visual anchors immediately appear.

In preliminary study 1, we examined the effect of the *Progressive Unfolding Effect* on the user performance. We compare the performance of Lattice Menu with and without the *Progressive Unfolding Effect* (*Progressive Lattice Menu* vs. *Full Lattice Menu*). The experiment was a one-way within-subject design with the following independent variable and levels. The order of the conditions was counterbalanced.

- MenuType: *Progressive Lattice Menu*, *Full Lattice Menu*

5.1.1 Procedure. For each *MenuType*, the same six target paths (two each from *0-bent*, *1-bent*, and *2-bent*, depicted in Figure 8 chosen at random) were tested. One block consisted of 48 trials (2 *MenuType* × 6 paths × 4 repetitions) of the menu selection task and a subject applied three blocks on the first day for the interface training, and five blocks on the second day for testing. It took less than 1 h to complete the experiment each day.

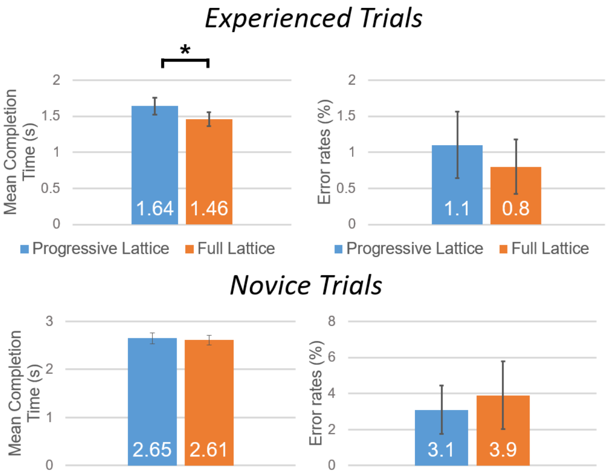


Figure 10: Mean of CT (s) and ER (%) from preliminary study 1. Error bars show standard errors, and asterisks indicate significant differences (p < .005).

5.1.2 Analysis. We collected 720 *Novice Trials* (2 *MenuType* × 6 paths × 5 blocks × 12 subjects) and 1440 *Experienced Trials*, and calculated the task completion time (CT) and error rate (ER). CT was measured from the moment the menu was opened, by staring at the start button for 1 s, until the leaf menu item was selected. A trial was counted as correct only when all three target items were correctly selected. Otherwise, it was counted as an error. For the analysis, we performed a one-way ANOVA on CT and a Friedman test for ER. Because the CT in *Experienced Trials* violated the normality assumption, we performed a Friedman test in this case.

5.1.3 Result. For the *Experienced Trials*, the effect of *MenuType* was significant on CT ($\chi^2(1) = 8.333$, $p < .005$) but not on ER. The *Full Lattice Menu* showed a significantly faster CT than *Progressive Lattice Menu*. For the *Novice Trials*, the effects of *MenuType* on both CT and ER were not significant.

5.1.4 Post-interview on user experience. Despite the significantly longer CT observed in *Progressive Lattice Menu* for *Experienced Trials*, more subjects (four) preferred *Progressive Lattice Menu* than *Full Lattice Menu* (two). Subjects who preferred *Progressive Lattice Menu* mainly commented on the discomfort from the visual distraction experienced in *Full Lattice Menu*. Subject p2 said *"Showing only the necessary anchors makes it easier to maintain my concentration"*. However, subjects who preferred *Full Lattice Menu* mainly commented on the ease of eye movement planning with visual anchors fully shown for up to the last level. However, importantly, most subjects (six) had no particular preference. They commented that they did not feel much difference in terms of the selection performance once they became familiar with both *MenuTypes*.

We had a long discussion on the tradeoff between the advantage of reducing visual distraction and the disadvantage of increasing

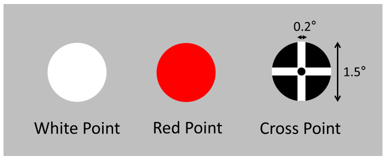


Figure 11: Three types of visual anchors used in preliminary study 2.

the menu selection time (0.18s in average) for expert usage that *Progressive Lattice Menu* brings. Finally, we chose to use *Progressive Lattice Menu* for our final design as we judged that the visual distraction of *Full Lattice Menu* can have a more crucial impact on usability, particularly in a realistic environment with different visual contents in the background.

5.2 Preliminary Study 2: Visual Anchor Design - Visual Saliency

A salient visual target encourages an effortless shift of attention [20, 32, 38], and can help with a rapid and accurate eye gaze pointing [2, 5, 15]. The visual saliency of the anchors can affect the user experience and performance of Lattice Menu. Herein, we compare three different types of visual anchors: *White Point* (baseline), *Red Point*, and *Cross Point* depicted in Figure 11. *Red Point* is a target with a higher visual saliency than *White Point* in terms of the *Color Opponency* [14]. *Cross Point* is another salient target with a high spatial frequency, and has often been used in previous eye-fixation studies [3, 28, 31].

In preliminary study 2, we investigated the effect of *VisualAnchorType* on user experience and performance of Lattice Menu. The experiment was a one-way within-subject design with the following independent variable and levels. The order of the conditions was counterbalanced.

- VisualAnchorType: *White Point*, *Red Point*, *Cross Point*

5.2.1 Procedure. For each *VisualAnchorType*, the same six target paths (two each from *0-bent*, *1-bent*, and *2-bent* chosen at random) were tested. One block consisted of 72 trials (3 *VisualAnchorType* × 6 paths × 4 repetitions) of the menu selection task, and the subject performed five blocks without training because they were already trained after 2 days of experience. It took approximately 1 h to complete the experiment.

5.2.2 Analysis. We collected 1080 *Novice Trials* (3 *VisualAnchorType* × 6 paths × 5 blocks × 12 subjects) and 2160 *Experienced Trials*, and calculated the CT and ER. For the analysis, because CT in both *Experienced Trials* and *Novice Trials* violated the normality assumption, we performed Friedman tests on both CT and ER.

5.2.3 Result. For both *Experienced Trials* and *Novice Trials*, the effects of *VisualAnchorType* on both CT and ER were not significant.

5.2.4 Post-interview on user experience. In the post-interview, most of the subjects (eight) chose the *White Point* as their preferred condition. For the most disliked condition, a majority of the subjects (nine) chose the *Red Point*. They mainly commented on the drawbacks of "over-saliency," stating that *"It causes unintended gaze shifts"* (p1), *"it was too bright"* (p2 and p6), and *"I felt higher eye fatigue."* (p5, p7, p8, p11, and p12). Similarly for the *Cross Point*, the subjects commented on their eye fatigue (p3 and p12). In addition, p6 and p11 described that the multiple *Cross Points* on the screen made them feel dizzy.

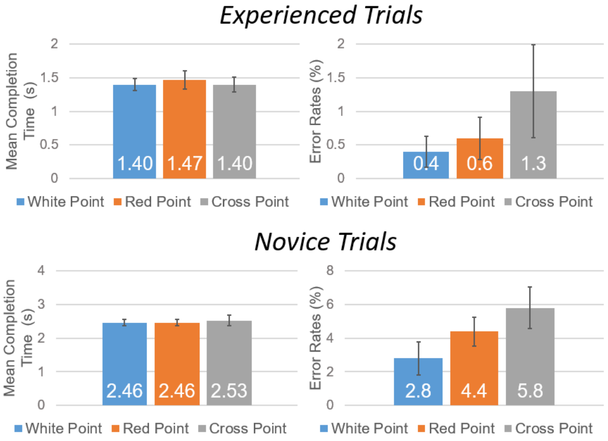


Figure 12: Mean of CT (s) and ER (%) from preliminary study 2. Error bars indicate standard errors.

We decided to use the *White Point* for our final design. Even considering that the *White Point* might be favorable because the subjects were already familiarized with it from preliminary study 1, negative user experiences regarding the *Red Point* and the *Cross Point* were crucial. Moreover, we also considered that the *White Point* showed the lowest error rates for both *Experienced Trials* and *Novice Trials* though the differences were not statistically significant.

6 USER STUDY 1: PERFORMANCE EVALUATION

In User Study 1, we formally evaluated the performance of the designed Lattice Menu on various menu layouts with different menu sizes and structures. *Structure* implies the number of items in each menu level. The experiment was a two-way within-subject

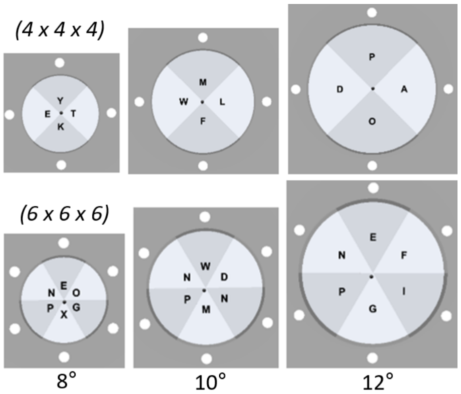


Figure 13: Various menu layouts tested in User Study 1.

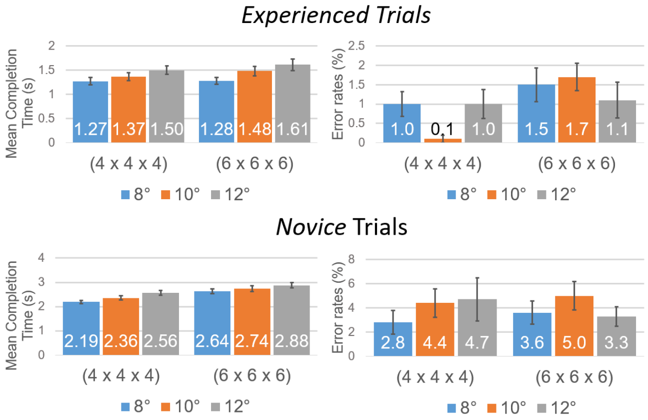


Figure 14: Mean of CT (s) and ER (%) from User Study 1. Error bars indicate standard errors.

design with the following independent variables and levels. The order of the conditions was counterbalanced using Balanced Latin Square.

- Structure: *4* × *4* × *4* and *6* × *6* × *6*
- Size (*D3*, Effective Radius): *8*, *10*, and *12*°

6.1 Procedure

For each *Structure* × *Size* condition, the same six target paths (two each from *0-bent*, *1-bent*, and *2-bent* chosen at random) were tested. One block consisted of 144 trials (2 *Structures* × 3 *Size* × 6 paths × 4 repetitions) of the menu selection task and the subject performed five blocks without prior training because they were already trained after 3 days of experience on the interfaces. It took approximately 1.5 h to complete the experiment.

6.2 Analysis

We collected 2160 *Novice Trials* (2 *Structures* × 3 *Sizes* × 6 paths × 5 blocks × 12 subjects) and 4320 *Experienced Trials*, and calculated the CT and ER. For the analysis, we performed a two-way ANOVA on CT and ER. We applied an aligned rank transform (ART) [36] on ER before conducting RM-ANOVA. For a post hoc comparison, a paired sample t-test on CT and a Wilcoxon signed-rank test on ER with a Bonferroni correction was used.

6.3 Results

6.3.1 *Experienced Trials.*

In the case of CT, the effects of both *Structure* (F(1,11) = 10.964, p < .01) and *Size* (F(2,22) = 19.070, p < .001) were significant. No significant interaction effects were observed. The post hoc comparison under each factor revealed that there were significant differences between *Size* conditions (8° vs. 10°, t = -3.530, p < .05; 8° vs. 12°, t = -5.396, p < .005; 10° vs. 12°, t = -3.181, p < .05) and *Structure* conditions (4 × 4 × 4 vs. 6 × 6 × 6, t = -3.311, p < .05). In the case of ER, the effect of *Structure* was significant (F(1,11) = 4.468, p < .05) whereas the effect of *Size* was not. There was a significant interaction effect (F(2,22) = 3.538, p < .05).

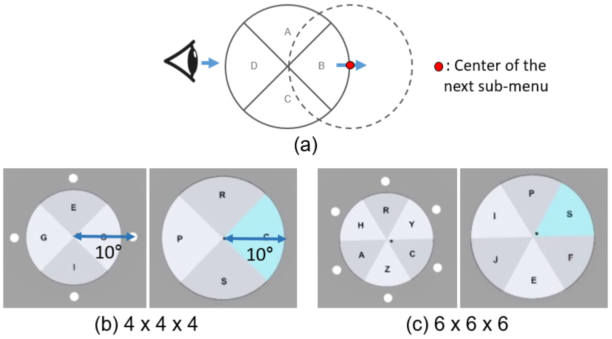


Figure 15: (a) Border-crossing marking menu (*BorderPie*, re-illustrated from [35]). Example layouts of *Lattice Menu* and *BorderPie* of 10° *Size* with (b) 4 × 4 × 4 and (c) 6 × 6 × 6 *Structure.*

6.3.2 *Novice Trials.*

In the case of CT, the effects of both *Structure* (F(1,11) = 86.973, p < .001) and *Size* (F(2,22) = 16.893, p < .001) were significant. No significant interaction effects were observed. The post hoc comparison under each factor revealed that there were significant differences between *Size* conditions (8° vs. 12°, t = -8.361, p < .005; 10° vs. 12°, t = -3.228, p < .05) and *Structure* conditions (4 × 4 × 4 vs. 6 × 6 × 6, t = -8.261, p < .005). In the case of ER, the effects of both *Structure* and *Size* were not significant.

6.4 Discussion

For the menu selection time in expert usage (*Experienced Trials*), we observed similar performance between 4 × 4 × 4 and 6 × 6 × 6 (1.38 vs. 1.46 s on average) which implies that the target-assisted gaze gesture with a diagonal eye movement can be easily performed as well. We can observe general increase of menu selection time with an increase in the number of items (*Structure*) and menu sizes (*Size*). For *Experienced Trials*, the effect of *Size* was dominant, whereas for *Novice Trials*, the effect of the number of items (*Structure*) was dominant. For the selection errors, 6 × 6 × 6 showed generally higher error rates than the 4 × 4 × 4 *Structure* in both novice and expert usage.

For overall performance, Lattice Menu showed considerably low error rates (0.7% for *4* × *4* × *4* and 1.4% for *6* × *6* × *6 Structures*) and short menu selection time (1.3-1.6 s) for expert usage (i.e., *Experienced Trials*). For the novice usage (i.e., *Novice Trials*), the menu item selection took 2.2-2.9 s with an overall error rate of ~4%.

7 USER STUDY 2: COMPARING LATTICE MENU WITH BORDERPIE

In User Study 2, we compared the performance of Lattice Menu with the traditional border-crossing marking menu design [35] (*BorderPie*) that does not utilize visual targets. The purpose of this study is to investigate the performance improvement of Lattice Menu over traditional technique.

The experiment was a two-way within-subject design with the following independent variables and levels. The order of the *Technique* × *Structure* conditions was counterbalanced using Balanced Latin Square.

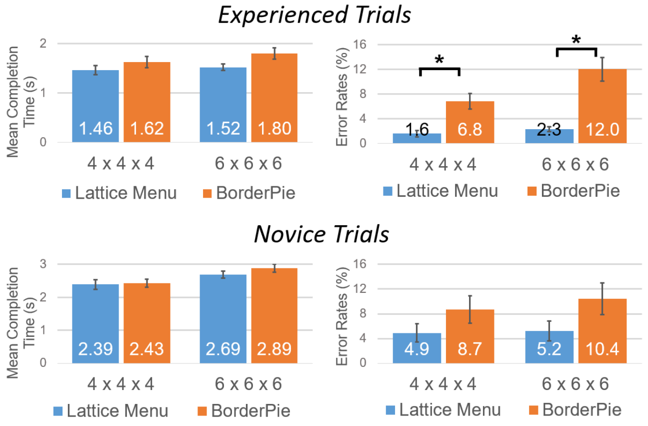


Figure 16: Mean of CT (s) and ER (%) from User Study 2. Error bars show standard errors, and asterisks indicate significant differences (p < .05).

- Technique: *Lattice Menu*, Border-crossing Marking Menu (*BorderPie*)
- Structure: 4 × 4 × 4, 6 × 6 × 6

7.1 Procedure

The experiment was conducted over 4 consecutive days. Subjects experienced different *Technique* × *Structure* conditions each day. Three menu sizes (*D3*: 8, 10, and 12°) were tested. For each *Technique* × *Structure* × *Size* condition, the same 16 target paths were provided; 1, 6, and 9 paths from *0-bent*, *1-bent*, and *2-bent* were randomly picked for 4 × 4 × 4 *Structure*, and 1, 4, and 11 paths from *0-bent*, *1-bent*, and *2-bent* were randomly picked for 6 × 6 × 6 *Structure* considering the ratio of paths for each *Structure*) tested. The experiment each day consisted of 192 trials (3 *Sizes* × 16 paths × 4 repetitions) for the training and the same number of trials for the testing. It took less than 1 h for a participant to complete the experiment each day.

7.2 Analysis

We collected 2304 *Novice Trials* (2 *Techniques* × 2 *Structures* × 3 menu sizes × 16 paths × 12 subjects) and 4608 *Experienced Trials*, and calculated the CT and ER. For the analysis, we performed a two-way ANOVA on CT and ER. We applied an aligned rank transform (ART) to ER before conducting the RM-ANOVA. For a post hoc comparison, a paired sample t-test on CT and a Wilcoxon signed-rank test on ER with a Bonferroni correction were conducted.

7.3 Results

7.3.1 Experienced Trials. In the case of CT, the effects of both *Technique* and *Structure* were not significant. In the case of ER, the effects of both *Technique* (F(1,11) = 53.400, p < .001) and *Structure* (F(1,11) = 14.458, p < .001) were significant. There was a significant interaction effect (F(1,11) = 7.446, p < .05). The post hoc comparison revealed that there were significant difference between pairs (4 × 4 × 4 *Lattice Menu* vs. 4 × 4 × 4 *BorderPie*, Z = -2.703, p <.05; 6 × 6 × 6 *Lattice Menu* vs. 6 × 6 × 6 *BorderPie*, Z = -2.934, p <.05).

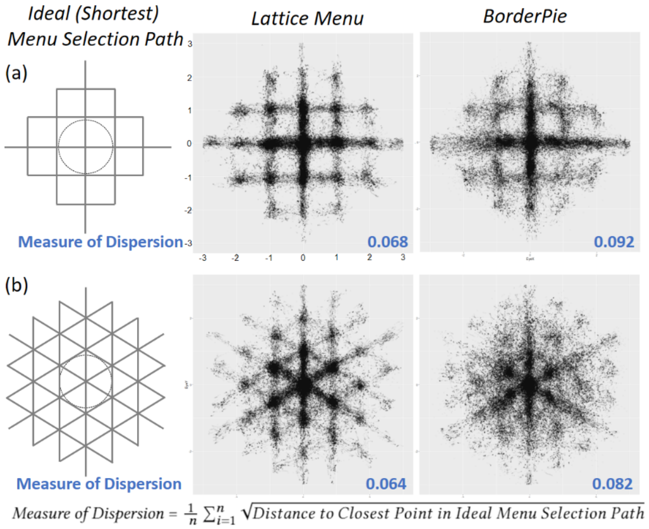


Figure 17: Distribution of saccade landing position for (a) 4 × 4 × 4 and (b) 6 × 6 × 6 *Structures* (normalized scale, n ≈ 300,000 for each plot) from User Study 2 and calculated *Measure of Dispersion* with the illustration of ideal menu selection paths.

7.3.2 Novice Trials. In the case of CT, the effect of *Technique* was not significant; however, the effect of *Structure* (F(1,11) = 56.855, p < .001) was significant. No significant interaction effects were observed. In the case of ER, the effect of *Technique* was significant (F(1,11) = 7.060, p < .05), whereas the effect of *Structure* was not significant. No significant interaction effects were observed.

7.4 Discussion

The comparative evaluation showed the remarkably (~5 times) lower error rates of *Lattice Menu* in comparison to *BorderPie* for expert usage (i.e., *Experienced Trials*). *BorderPie* in particular showed a steep increase in error rate in a 6 × 6 × 6 *Structure* (12%). This observation implies that visual targets can be more beneficial when the number of items in each menu level increases. In general, there were no significant differences in the menu selection time; however, there was a remarkable difference in the error rates. We observed approximately 2 (*Novice Trials*) to 5 (*Experienced Trials*) times higher error rates in *BorderPie*. The scatter plot of the saccade landing positions shown in Figure 17 also indicates the effectiveness of the *Lattice Menu* technique. The lattice of visual anchors allowed the users to better follow the ideal paths for selection.

In the post-interview on user experience, all 12 subjects preferred to use *Lattice Menu* over *BorderPie*. Most of the subjects (eight) noted that they felt more tired with increased eye fatigue when using *BorderPie*. The main comments on *Lattice Menu* are as follows: (1) Having a visual target to focus helps accurate gaze pointing (*p1, p4, p5, p7, p9*). (2) There was not much difference in difficulty between 4 × 4 × 4 vs. 6 × 6 × 6 *Lattice Menu* (*p5, p6*), and (3) *Lattice Menu* was more sensitive to the tracking offsets because of the smaller area used for selection (*p4, p5*). The main comments on *BorderPie* are as follows: (1) Errors often occur by moving the eyes more than intended (*p1, p10*). (2) As the menu size becomes smaller, unintended selections occurs more frequently (*p2, p8, p12*). These observed types of errors in *BorderPie* are consistent with the observations in StickyPie [1].

8 GENERAL DISCUSSION AND FUTURE WORK

8.1 Design Insight for Gaze-Based Menu

We share several insights for gaze-based menu design based on the observations in this study.

(1) Visual anchors effectively guide the eye gaze for safe menu selection. Several unwanted options are always located adjacent to a single wanted item. Because of the ballistic nature of human eye movement [23], designing a safe gaze-based menu selection is challenging. In this study, we confirmed that the provisioning of visual anchors can safely guide eye movement and prevent unintentional selection errors in the gaze-based menu.

(2) Over-saliency for the visual guidance target results in poor user experiences. The visual target for eye guidance should be sufficiently salient to encourage effortless [20] and accurate [5] gaze pointing. However, as observed in preliminary study 2, excessive saliency can induce increased eye fatigue and impact the overall user experience at the interface. The target saliency must be carefully designed for comfortable eye guidance.

(3) The space for the menu contents (e.g., label texts or icons) needs to be separated from the space for the menu item selection. Separation of the two spaces allows users to navigate the menu items safely. When two spaces are not separated, unintentional selection errors easily occur as observed in the pilot study noted in Section 3.2. We suggest making an explicit boundary to capture the eye [32] inside the safe region during user menu navigation.

8.2 Applying Lattice Menu in Real Application

8.2.1 External Validity: Surrounding Environment. We used a plain gray background for the controlled experimental setup in this study. However, in a real application, the surrounding AR/VR environment and UI contents can be dynamic, which should be considered when adopting Lattice Menu. We recommend showing Lattice Menu under an opaque background if the application context allows the background to be occluded. Giving a border to the visual anchor circle [22] may also be helpful for distinguishing the anchor from the background.

8.2.2 External Validity: Size and Depth. In this study, we used a menu size (*D3* in Figure 3) of between 8° and 12° with a three-level menu depth. However, depending on the target situation, designers may need to limit the size or depth of Lattice Menu. Reducing the menu depth to two-level can be effective if the required menu capacity of the target application is small (e.g., using 4 × 4 Lattice Menu when required number of commands is less than 16). *D3* and *D4* may be reduced until the space limit for the menu contents (e.g., icon or text). Using symbolic icons instead of long texts can be a good option. Designers can also adjust *D2* considering the tracking offset of the eye tracker.

Table 1: The comparison between StickyPie [1] and Lattice Menu

	Lattice Menu	StickyPie
Tested Input	Head-Combined Gaze Input	Eyes-only Input
Reported Menu Selection Time	1.4-2.6s	2.5-3.4s
Reported Menu Selection Error Rate	2-6%	~15%

8.2.3 Design Opportunity: Personalizing Lattice Menu. Designers can change the parameter values (depicted in Figure 3) to personalize Lattice Menu. Because there are large individual differences in ocular motor control skills and eye tracking offsets, personalizing Lattice Menu is expected to be beneficial. Decreasing *D4* (i.e., making the visual pie smaller) will be helpful for users who suffer from unintentional selection errors during menu navigation because it will provide the distance between the visual pie and *Item Selection Zone*. In addition, increasing *D2* to greater than 2° will be helpful for users who have trouble selecting items owing to the large eye tracking offsets.

8.3 Head-Fixed Lattice Menu

As we described in Section 3.1, there can be two types of menus in the AR/VR environment; *Head-Fixed Menu* that uses eyes-only input and *World-Fixed Menu* which uses a head-combined gaze input for control. The head-combined gaze input can be comfortable [29] for gaze shifts of over 25°, whereas eyes-only input has a more limited range of comfortable shifts (~20° [30]). Designers can properly utilize each type of menu considering the target use case in the AR/VR. (e.g., using *Head-Fixed Menu* for a global-level menu and *World-Fixed Menu* for a context menu with a certain target object)

In this study, Lattice Menu was implemented and evaluated using a head-combined gaze input, i.e., a *World-Fixed Menu* setup. Validation in this setup has the advantage of being compatible with an external, stationary eye tracker (e.g., for smart TV) environment because head-combined gaze input is utilized in both cases. Meanwhile, validation in the eyes-only input (*Head-Fixed Menu*) setup also has its advantage, being the only possible option when 3D sensing for the surrounding space is incapable of (e.g., smart glasses without 3D sensing capability).

To design Head-Fixed Lattice Menu with eyes-only control, further investigation (e.g., a usable menu size and hierarchy) is necessary considering the differences in motor skills between eyes-only and head-combined eye movements. Eyes-only, Head-Fixed Lattice Menu will be further explored in our future study.

8.4 Comparison with StickyPie

Similar to this work, StickyPie [1] proposes a way to make the gaze-based marking menu less error-prone. StickyPie predicts the saccade landing position to prevent overshooting errors, whereas Lattice Menu provides visual anchors for stable eye control. Because StickyPie and Lattice Menu pursue different ideas for the same goal, a comparison between the two should be made. However, as described in Section 2.2, a direct comparison was not conducted in this study because of practical difficulties.

Instead, we attempted to compare the reported performances of each technique. The 6 × 6 × 6 Lattice Menu with an 8° radius condition tested in User Study 2 of this paper has a similar menu layout with the *G6* condition (6 × 6 × 6 menu structure with a 7.5° radius) in Experiment 2 of the StickyPie study. Both conditions underwent a similar level of training (32-64 repetitions). For *Experienced Trial*, menu selection on *Lattice Menu* (1.4s, 2%) was 1.8-times faster and showed almost 8-times lower error rates than *StickyPie* (2.5 s, 15%). For *Novice Trial*, menu selection on *Lattice Menu* (2.6s, 6%) was approximately 1.3 times faster and showed 3 times less error rates than *StickyPie* (3.4s, 15%). The results of the comparisons are summarized in Table 1. Though the difference in tested input need to be considered, we were able to observed remarkably lower error rates in Lattice Menu from the reported performances.

8.5 Cancel and Back Menu Operation

Although not explored in this study, *Cancel* (i.e., closing a menu without selection) and *Back* (i.e., a roll-back to the previous menu-level) are important functionalities for the menu. For *Cancel*, we considered utilizing an eye behavior that does not naturally occur during the menu selection process (e.g., long eye winks/blinks). For *Back*, we considered reserving one menu item in each submenu for the roll-back operation. With this approach, the 4 × 4 × 4 layout can cover 36 commands (4 × 3 × 3), and the 6 × 6 × 6 layout can cover 150 commands (6 × 5 × 5).

9 CONCLUSION

This study introduced Lattice menu, a gaze-based marking menu utilizing target-assisted gaze gestures on a lattice of visual anchors. From our empirical evaluation, Lattice Menu showed a considerably low error rate (~1%) and quick menu selection time (1.3-1.6 s) on various menu layouts. The distribution of the saccade landing position demonstrated the effectiveness of the visual anchors on stable gaze guidance. We also shared several design insights for the gaze-based menu technique from the observed user behaviors. We hope this study can be a meaningful step on realizing practical hands-free menu interaction.

ACKNOWLEDGMENTS

This work was supported by Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2020-0-00537, Development of 5G based low latency device – edge cloud interaction technology).

REFERENCES

[1] 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 *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–16.

[2] AH Bell, MA Meredith, AJ Van Opstal, and DougP Munoz. 2006. Stimulus intensity modifies saccadic reaction time and visual response latency in the superior colliculus. *Experimental Brain Research* 174, 1 (2006), 53–59.

[3] Dipesh Bhattarai, Marwan Suheimat, Andrew J Lambert, and David A Atchison. 2019. Fixation stability with Bessel beams. *Optometry and Vision Science* 96, 2 (2019), 95–102.

[4] Andy Cockburn, Carl Gutwin, Joey Scarr, and Sylvain Malacria. 2014. Supporting novice to expert transitions in user interfaces. *ACM Computing Surveys (CSUR)* 47, 2 (2014), 1–36.

[5] H Deuble, W Wolf, and G Hauske. 1984. The evaluation of the oculomotor error signal. In *Advances in Psychology*. Vol. 22. Elsevier, 55–62.

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

[7] 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*. 1–13.

[8] Anthony Hornof, Anna Cavender, and Rob Hoselton. 2003. Eyedraw: a system for drawing pictures with eye movements. *ACM SIGACCESS Accessibility and Computing* 77-78 (2003), 86–93.

[9] 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*. 51–54.

[10] 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*. 229–232.

[11] FOVE inc. 2021. FOVE0 Headset Specification. https://fove-inc.com/product/, last visited Dec. 2021.

[12] HTC inc. 2021. HTC Vive Pro Eye Headset Specification. https://www.vive.com/kr/product/vive-pro-eye/specs/, last visited Dec. 2021.

[13] Toshiya Isomoto, Shota Yamanaka, and Buntarou Shizuki. 2020. Gaze-based Command Activation Technique Robust Against Unintentional Activation using Dwell-then-Gesture. (2020).

[14] Laurent Itti and Christof Koch. 2001. Computational modelling of visual attention. *Nature reviews neuroscience* 2, 3 (2001), 194–203.

[15] RP Kalesnykas and PE Hallett. 1994. Retinal eccentricity and the latency of eye saccades. *Vision research* 34, 4 (1994), 517–531.

[16] 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*. 213–220.

[17] Gordon Kurtenbach and William Buxton. 1991. Issues in combining marking and direct manipulation techniques. In *Proceedings of the 4th annual ACM symposium on User interface software and technology*. 137–144.

[18] 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.

[19] Gordon Paul Kurtenbach. 1993. *The design and evaluation of marking menus*. University of Toronto Toronto.

[20] Olivier Le Meur and Zhi Liu. 2015. Saccadic model of eye movements for free-viewing condition. *Vision research* 116 (2015), 152–164.

[21] 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*. 357–360.

[22] 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*. 1–5.

[23] Päivi Majaranta, Kari-Jouko Räihä, Aulikki Hyrskykari, and Oleg Špakov. 2019. Eye movements and human-computer interaction. In *Eye Movement Research*. Springer, 971–1015.

[24] M v Menozzi, A v Buol, H Krueger, and Ch Miège. 1994. Direction of gaze and comfort: discovering the relation for the ergonomic optimization of visual tasks. *Ophthalmic and Physiological Optics* 14, 4 (1994), 393–399.

[25] 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*. 27–34.

[26] 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*. 1–10.

[27] 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*. 2741–2750.

[28] Immo Schuetz, T Scott Murdison, Kevin J MacKenzie, and Marina Zannoli. 2019. An Explanation of Fitts' Law-like Performance in Gaze-Based Selection Tasks Using a Psychophysics Approach. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–13.

[29] Ludwig Sidenmark and Hans Gellersen. 2019. Eye, head and torso coordination during gaze shifts in virtual reality. *ACM Transactions on Computer-Human Interaction (TOCHI)* 27, 1 (2019), 1–40.

[30] Ludwig Sidenmark and Hans Gellersen. 2019. Eye&head: Synergetic eye and head movement for gaze pointing and selection. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. 1161–1174.

[31] Lore Thaler, Alexander C Schütz, Melvyn A Goodale, and Karl R Gegenfurtner. 2013. What is the best fixation target? The effect of target shape on stability of fixational eye movements. *Vision research* 76 (2013), 31–42.

[32] Jan Theeuwes, Arthur F Kramer, Sowon Hahn, and David E Irwin. 1998. Our eyes do not always go where we want them to go: Capture of the eyes by new objects. *Psychological Science* 9, 5 (1998), 379–385.

[33] Jan Theeuwes, Sebastiaan Mathôt, and Alan Kingstone. 2010. Object-based eye movements: The eyes prefer to stay within the same object. *Attention, Perception, & Psychophysics* 72, 3 (2010), 597–601.

[34] 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*. 47–50.

[35] 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*. 93–96.

[36] 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*. 143–146.

[37] 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*. 11–18.

[38] Steven Yantis and Anne P Hillstrom. 1994. Stimulus-driven attentional capture: evidence from equiluminant visual objects. *Journal of experimental psychology: Human perception and performance* 20, 1 (1994), 95.