Level-of-Detail AR: Dynamically Adjusting Augmented Reality Level of Detail Based on Visual Angle

Abby Wysopal* Vivian Ross† Joyce Passananti‡ Kangyou Yu§ Brandon Huynh¶ Tobias Höllerer‖

Computer Science Department
University of California, Santa Barbara

Figure 1: Images of the Level-of-Detail AR mechanism applied to a compass application. The level of detail, such as the points on the compass, adjusts as the visual angle changes due to a decreasing distance between the user and the object. The image on the left shows the compass at the farthest distance from the user, resulting in the cardinal directions being included in the display and the non-cardinal directions being excluded from the display. The image in the center, which shows the compass at a smaller distance from the user, includes both the cardinal and non-cardinal directions in the display but excludes the degrees and their associated markings. The image on the right, in which the compass is the closest to the user, shows all levels of detail for the compass application including the display of cardinal directions, non-cardinal directions, degrees, and degree markings.

Abstract

Dynamically adjusting the content of augmented reality (AR) applications to efficiently display information best fitting the available screen estate may be important for user performance and satisfaction. Currently, there is not a common practice for dynamically adjusting the content of AR applications based on their apparent size in the user's view of the surround environment. We present a Level-of-Detail AR mechanism to improve the usability of AR applications at any relative size. Our mechanism dynamically renders textual and interactable content based on its legibility, interactability, and viewability respectively. When tested, Level-of-Detail AR functioned as intended out-of-the-box on 44 of the 45 standard user interface Unity prefabs in Microsoft's Mixed Reality Tool Kit. We additionally evaluated impact on task performance, user distance, and subjective satisfaction through a mixed-design user study with 45 participants. Statistical analysis of our results revealed significant task-dependent differences in user performance between the modes. User satisfaction was consistently higher for the Level-of-Detail AR condition.

Index Terms: Human-centered computing—Human computer interaction (HCI)—Interaction paradigms—Mixed / augmented reality; Human-centered computing—Interaction design—Interaction design process and methods—User interface design

*e-mail: abbywysopal@ucsb.edu
†e-mail:vivianross@ucsb.edu
‡e-mail:joycepassananti@ucsb.edu
§e-mail:kangyouyu@ucsb.edu
¶e-mail:bhuynh@ucsb.edu
‖e-mail:holl@cs.ucsb.edu

1 Introduction

In 3D environments, the visual angle that an application subtends is constantly changing with respect to the user's pose and the application's location and size. Visual angles indicate the size of the retinal image of an application. As the user moves closer to and further from an application, its relative size will increase and decrease, respectively. Additionally, a common user interaction with applications in 3D environments is scaling, i.e. resizing an application to the user's desired size. The continuous changing of the visual angle of an application is a common occurrence in AR, making the usability of an application at all visual angles critical for the user experience and efficient use of virtual content. In addition to usability, user satisfaction with the application's appearance and interaction, for all viewing angles and distances, is crucial.

A similar challenge has been presented in 2D environments, which largely influenced the design of web applications. *Responsive design* is a development philosophy used as an approach to application development, primarily used in web applications, that aims to render applications suitably for a variety of devices and window or screen sizes. Due to the design methodology prioritizing usability and satisfaction for all display sizes, web pages generally have a desirable user experience across desktop, mobile, and tablet devices. With the success of responsive design in 2D environments and the apparent need for a dynamic approach to application design in 3D environments, we developed a Level-of-Detail AR mechanism to dynamically adjust application content for improved usability at any visual angle.

The adaptation of an application's content to all visual angles might be beneficial to user enjoyment and efficient use of augmented space and rendering resources. This work presents and evaluates an approach for AR applications dynamically adjusting displayed content to changes in relative sizes of the application depending on its visual angle. It addresses four primary research questions:

- How can we design and deploy a Level-of-Detail AR mecha-

2642-5254/23/$31.00 ©2023 IEEE
DOI 10.1109/VR55154.2023.00022

nism so that it is generally applicable and easy/automatic for AR user interface designers and engineers to adopt?

- How does dynamically changing an application's content based on its visual angle affect task performance?
- How does it affect users' motion behavior?
- How does it affect user satisfaction?

To explore these research questions, we evaluated our implementation of Level-of-Detail AR on typical interaction tasks in AR applications. Through a user evaluation and applying our mechanism to numerous types of user interface elements, we investigated the effect of responsive design in AR. Primarily, we were interested in exploring our research questions by understanding the effect responsive design has on application usability and its level of distraction to users.

This paper is structured in seven main sections. Section 2 presents a brief overview of the related work, discussing the current techniques used in web and AR design to adapt content based on context and user activity. Section 3 presents and explains our implementation of Level-of-Detail AR, and Section 4 provides an explanation of how to use the Level-of-Detail mechanism in AR development. The next two sections outline the design of the user study experiment and its associated evaluation. Section 7 presents results and analysis on our experiment data. The paper ends with a discussion of the results, our final conclusions and future work.

The primary contributions of this paper are as follows: We developed the Level-of-Detail AR mechanism to dynamically adjust application content based on subtended visual angle and demonstrate its general applicability by applying it to 45 standard user interface examples from the Microsoft Mixed Reality Toolkit. We then conducted a user study (n=45) to evaluate the Level-of-Detail AR mechanism on 9 tasks within 3 example applications, with respect to a) task performance, b) induced motion behavior, and c) subjective satisfaction. The results shed light on the benefits, overall usability, and potential pitfalls of responsive UI design for AR applications.

2 Background and Related Work

In order to successfully apply responsive design to AR we should first understand the key benefits of responsive design in web and mobile applications. Next, it is important to evaluate the current state of activity centered AR design. A thorough review of both research areas will fundamentally assist us in successfully designing a mechanism for responsive design in AR.

2.1 Responsive Design

Responsive web design allows web applications to dynamically adjust to a device's screen size, orientation, and proportion [1]. From a smartwatch to a desktop computer, this approach is intended to adapt a web page for a diverse range of device screens. The development of responsive design came as users began accessing web pages more often on their handheld device rather than their desktop computer [2]. A static webpage was no longer suitable for the variety of devices available. Therefore, developers could either make multiple designs at fixed display sizes or make a single design to adjust to all screen sizes. For instance, Jiang et al. proposed a novel approach, which enables designers to safely create flexible layouts for different screen sizes and orientations, for constraint-based graphical user interface (GUI) [12]. Gajos et al. also designed and applied an interactive tool for eliciting user preferences and using this feedback to automatically learn a factored cost function of optimization-based system [7]. The success of responsive web design is noteworthy in 2D environments with its ability to improve the usability of a web page for such a variety of devices.

2.2 Dynamic Content Management

Dynamically adjusting the content of applications to efficiently display relevant information is an important field of research within AR. The following is an overview of a few related areas.

2.2.1 Information Filtering

Information filtering adjusts the applications content based on the user's current state (location and intent) to display understandable information. Here information is displayed efficiently through the organization of content. A filtering algorithm designed by Julier et al. dynamically adapts content based on the changes in the environment [13]. Similarily, Tatzgern et al. used filtering to balance potential clutter using hierarchical clustering to create a Level of Detail (LOD) structure [22]. Information filtering effectively reduces clutter through managing the display of large amounts of data. We will expand on this idea to reduce clutter on a per application basis.

2.2.2 Context-Aware Optimizations

Context aware optimizations resourcefully manage virtual space by moving applications to useful locations. Grubert et al. outlined the need for context aware applications as an important step towards the future of Pervasive AR [8]. Gajos et al. designed an interactive tool to elicit user preferences to automatically learn the parameters to optimize the layouts of user interfaces [7]. Sarcar et al. purposed a ability-based optimization approach which aims to improve interface designs for users with sensorimotor and cognitive impairments [20]. Cheng et al. used context aware optimization to increase the usability of applications by automatically placing virtual elements in an environment [3]. Lindlbauer et al. controlled both when and where an application is displayed through context aware optimization [15]. Lages and Bowman [14] further optimize application placement dynamically while walking around. These designs prioritize virtual space by the arrangement of applications. Context-aware optimization increase the usability of applications by placing them in beneficial locations. Our work gives the user the flexibility to place applications in AR while also increasing the usability of the application at any position.

2.2.3 Level of Detail Interfaces

The parameterizing of an application's display information with respect to the distance from the camera was previously accomplished by LOD Interfaces. LOD interfaces were first presented by DiVerdi et al. with the goal of optimizing screen space in 3D environments solely based on the applications distance from the camera [5]. This idea was then applied to mobile application by Gòmez et al. who used LOD dependent on both distance and orientation [9] and to camera tracking from image targets by Sung et al. [21]. Like Diverdi et al., Lindlbauer et al. [15] utilize discrete handcrafted levels of detail content presentations as a basis for adaptive AR. In the same vein, Daskalogrigorakis et al. present an LOD glanceable interface that provides compact information at a glance [4]. In these works, the display size cannot be adjusted by the user limiting their approaches to only consider distance in its implementation.

Head-based and gaze-based inputs have also been used to trigger LOD interfaces. Pfeuffer et al. explored a design space for adapting AR applications based on eye gaze [19] and used it to implement various sample applications, one of which used information levels of varying detail. Proposed by Lu et al., Glanceable AR provides users information at a glance [16], but can trigger additional levels of detail by gaze and dwell. While Glanceable AR efficiently relays content, users have limited flexibility in the amount of information presented to them.

A distinguishing feature of our work is that the generation of Level-of-Detail content is automated. The AR designer/programmer only needs to correctly identify default, text, interactable and non-interactive components, and the levels of detail will automatically apply. Our work also provides one of the first controlled user studies focused specifically on Level-of-Detail AR interfaces.

3 Implementation

Our goal in developing the Level-of-Detail AR mechanism is to improve the usability of an application at any visual angle. This section illustrates how Level-of-Detail AR helps to achieve this goal by dynamically adapting application content based on its visual angle. Our mechanism was implemented for the HoloLens-2 in C# as a behavioral component to be applied to objects in Unity [23]. We developed and tested Level-of-Detail AR with Unity 2020.3.20f1 LTS to be compatible with MRTK-Unity. Figure 2 illustrates the mechanism's logical flow, where Start is the function called at the application startup and Update is the function called every frame. The Level-of-Detail AR mechanism has four main functions. Upon application startup, it performs object classification and determines each object's usability threshold. For every frame, it compares the application's current scale to distance ratio to each object's threshold and manages application content by rendering only the objects with a threshold less than the current ratio.

The Level-of-Detail AR mechanism examines two key concepts - determining a threshold value for when an object is to be considered usable and transitioning between an object's usability based on its threshold value with limited disruption to the user. We have divided the explanation of our mechanism into three parts: UI object classification, threshold determination, and transitioning strategy. For this implementation, we are assuming the device technical specifications of the HoloLens-2 device, which has a resolution of 2k, holographic density greater than 2.5k, and a target frame rate of 60 FPS.

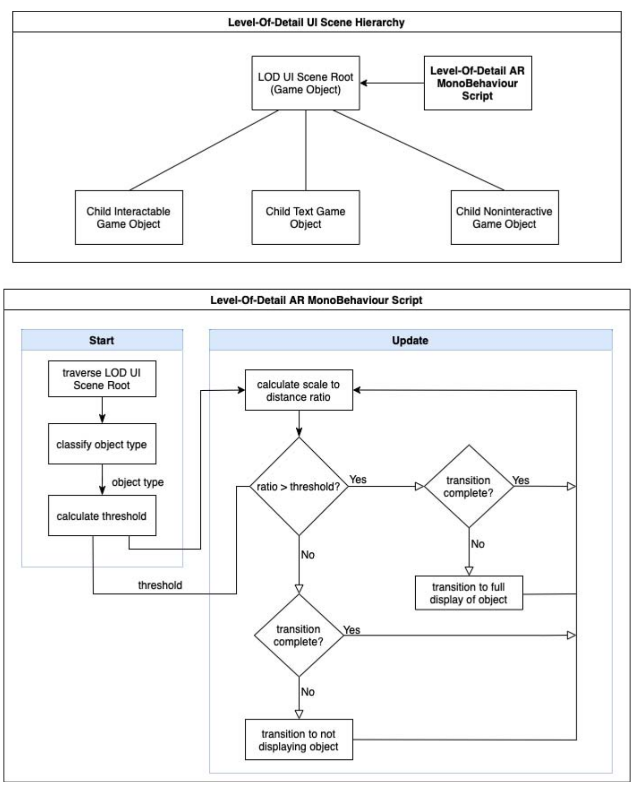

Figure 2: Top: Hierarchical structure of a Unity scene with Level-Of-Detail AR MonoBehaviour script attached as a behavioral component. Bottom: Flow chart illustrating the behavior of the Level-Of-Detail AR MonoBehaviour script.

3.1 Object Classification

In order to correctly determine the threshold for usability of an object, we must first classify an object as one of three groups – text, interactable, or noninteractive object. This grouping was decided because each type of object implies a different type of content, and each type of content has different rules for determining its usability. The key idea to this mechanism is to only display usable content and remove all content that is not considered usable. The three types of classifications and their rules for determining usability are:

Text Object A text object is to be considered usable when it is legible. Therefore, its usability is based on its readability. Any text object that is not considered readable should not be displayed in the application content.

Interactable Object An interactable object is to be considered usable when it is large enough for interaction. Therefore, its usability is based on its ability to be successfully interacted with. For example, a button should only be rendered if it is large enough to be selected. Therefore, any interactable object that is not considered interactable should not be displayed in the application content.

Noninteractive Object A noninteractive object is to be considered usable when it is large enough to be viewed. Therefore, its usability is based on its viewability. A noninteractive object that is not considered viewable should not be displayed in the application content.

3.2 Threshold Determination

Thresholds are determined for an object based on its calculated scale to distance ratio and its classification. After an object is classified and its rules for determining usability are outlined, a threshold value will be calculated representing the value at which an object will be rendered. The threshold value is the mathematical representation of an object's usability. The threshold values for the three types of classifications are calculated as follows: .

Text Object The readability of text in AR is based on the text height and its distance from the user, in which distance from the user is the distance between a user's head and the center of the object. The threshold value is determined by a height to distance ratio. Microsoft typography [18] recommends the minimum legible font height at a distance of 45 cm to be in the range of 3.14mm to 3.9mm. In our mechanism, we calculate the minimum threshold for a text object by dividing .36cm by 45cm and therefore determining the threshold to be .008.

Interactable Object The usability of an interactable object in AR is based on its volume, which is calculated as $width * length * height$, and distance from the user. The threshold value is determined by a volume to distance ratio. Microsoft interactable [17] recommends the target size for interaction as 3.5cm by 3.5cm for a distance of 200 cm. Based on the target size, we calculated the minimum threshold for an interactable object by dividing (3cmx3cmx1cm) by 200cm and therefore determining the threshold to be .045.

Noninteractive Object The viewability of a noninteractive object in AR is based on its volume and distance from the user. The threshold value is determined by a volume to distance ratio. Microsoft has not established a target size for when an object is viewable in HoloLens2. Because of this, we found (through trial and error) that a good threshold of viewability to be 10 times smaller than the threshold for usability of an interactable object. Therefore, our threshold for a noninteractive object is set to .0045.

3.3 Transitioning Strategy

After determining the threshold value for each object, we developed a strategy for the transitioning between threshold values. If an object's minimum threshold is met, such that the current scale to distance ratio is larger than their predetermined threshold, then that object is

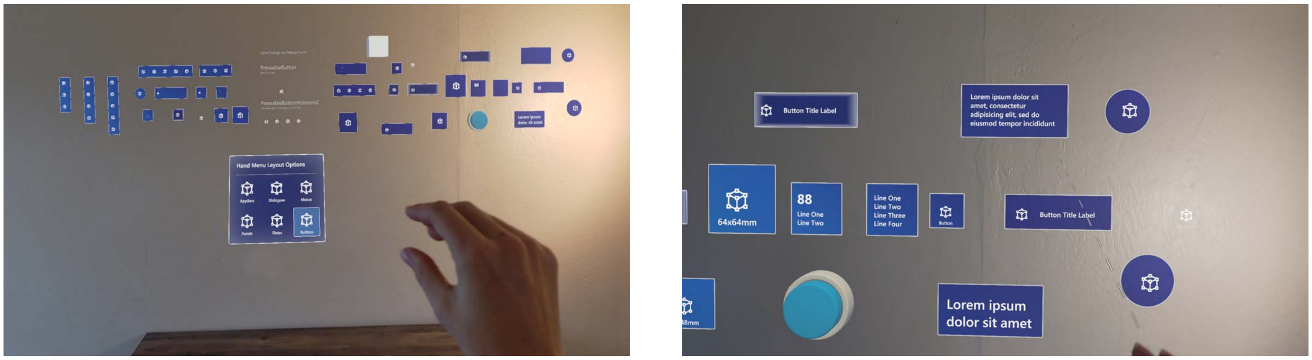

Figure 3: Level-of-Detail AR applied to MRTK Example prefabs. As shown, the viewing distance affects text visibility.

displayed. However, there is a stark transition between rendering and not rendering an object, which can be startling to a user because objects do not appear and disappear in the real world. In order to minimize the starkness of this transition we implemented a transition strategy based on opacity. When an object's threshold is reached, it goes through a transition stage where it incrementally increases opacity until it is fully visible. In our implementation, we increase opacity by incrementing 8-bit alpha by one every frame (using the target framerate of 60 FPS) until alpha is 255 or the threshold is no longer reached. A similar process occurs when the current scale to distance ratio falls below the threshold value. The object goes through a transition stage where it incrementally decreases opacity until it is invisible. We decrease opacity by decrementing alpha by one every frame until alpha is 0 or the threshold has been reached. If the threshold changes during the transition stage, we simply switch the type of transition method.

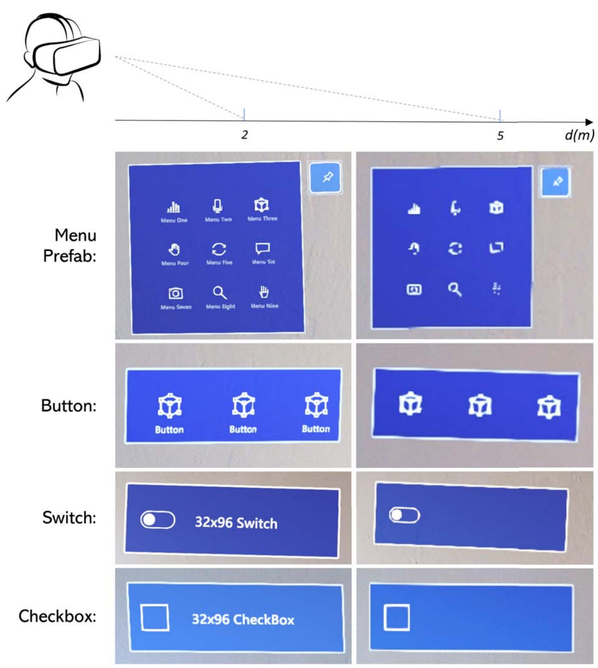

Figure 4: Level-of-Detail AR applied to MRTK prefabs. Comparison of the view from 2m vs. 5m using a HoloLens-2. Widget sizes do not reflect actual visual angle here, but resolution does.

4 Generality of Approach / Ease of Authoring

As mentioned in the section 3, Level-of-Detail AR can be applied to user interface elements as a behavioral component in Unity [23]. The Level-of-Detail AR mechanism can be added to any Unity GameObject element in two simple steps. First, add the script as a behavioral component to the desired object, then insert the object as the "Parent Object" parameter. Once completed, Level-of-Detail AR will automatically apply the Level-of-Detail mechanism to the object and all of it's child objects. The automatic functionality of the mechanism requires little effort from the programmer beyond a basic understanding of Unity concepts.

We applied our Level-of-Detail AR mechanism to a wide variety of user interface elements in Microsoft's Mixed Reality Tool Kit (MRTK) such as menus, dialogues, and slates (containing buttons, sliders, and check boxes). The user interface elements were collected through an extensive search for all prefab assets within MRTK. Some examples of the mechanism are shown in Figure 3. This allowed us to investigate the mechanism's behavior on arbitrary elements and understand its ability to integrate into many types of elements. We found that 44 of the 45 tested elements showed the expected behavior. This helped demonstrate the robustness of our created mechanism as it resulted in expected behavior for over 97% of the elements it was applied to.

Although one element resulted in unexpected behavior, this behavior highlights an important design choice we made in our mechanism. The unexpected behavior shown by this element was that the text objects appeared when the element's app bar button was pressed, regardless of the text objects' readability threshold. This is because the element enabled and disabled its objects within another behavioral component, and the Level-of-Detail AR mechanism did not override those actions. We made this design choice to give developers the flexibility to include other behavioral components in their objects without Level-of-Detail AR overriding other desired behaviors.

While applying the mechanism to user interface elements, we analyzed the usefulness of Level-of-Detail AR for various element types. The *Follow Me* behavior component ensures an element is always within a specified visual angle. Because of this, any element with the follow-me script applied should not also apply the Level-of-Detail AR script. This is because the Level-of-Detail AR mechanism manages the content to be usable at any visual angle. When an element has a constant visual angle, the key benefits of the Level-of-Detail AR would not be used. We can see a close-up comparison of a variety of MRTK prefabs viewed from different distances in Figure 4. When elements become too small and are determined illegible, they disappear, minimizing the space taken up and simplifying the view. An interesting observation made during this evaluation, was the adjusting of content on the slate element. In this user interface, when an object is disabled, all other objects are

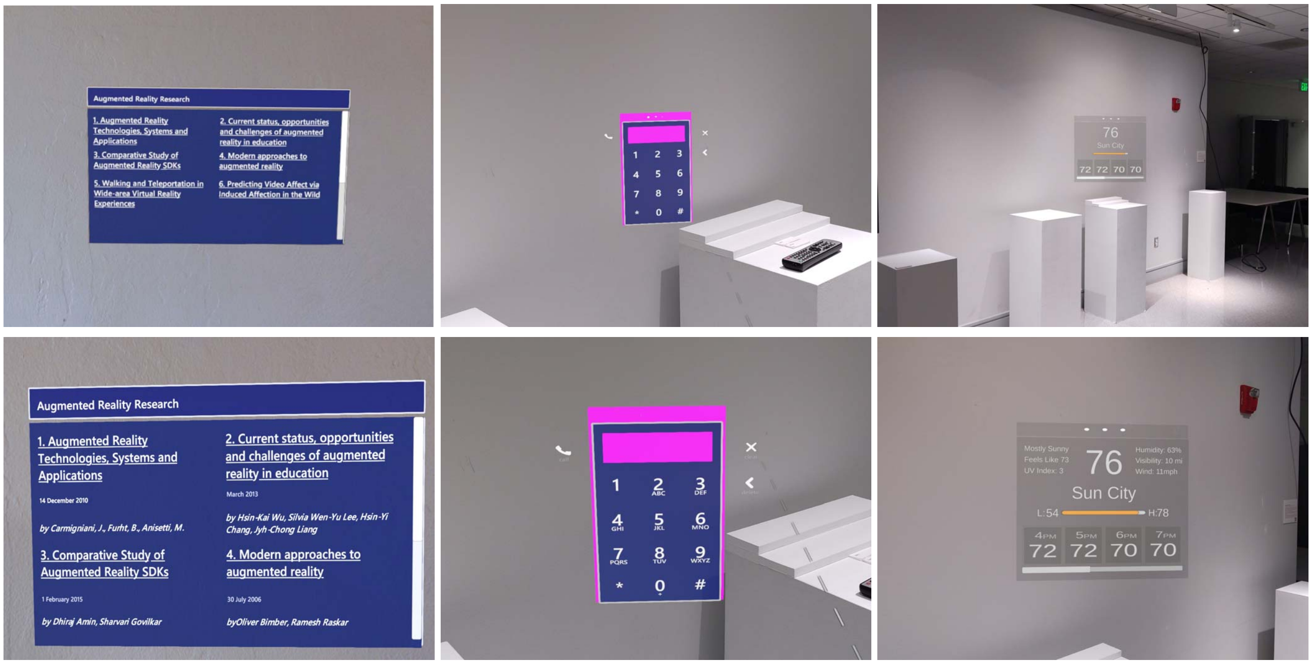

Figure 5: Experiment applications (from left to right) - Article Display (Tasks 1-4), Touchpad (Tasks 5-6), and Weather Display (Tasks 7-9)

shifted to fill the gap of the missing objects. This observed behavior is very beneficial for rearranging current content that is managed by the Level-of-Detail AR mechanism. Essentially, the slate adjusts all useful information to the visible portion of the element and avoids unnecessary gaps within the displayed content.

5 Experiment

The main purpose of our user study experiment was to compare the usability of our Level-of-Detail AR mechanism with conventional AR on a variety of interaction tasks.

5.1 Experiment Design

The user study followed a 2×9 mixed design with a between-subject variable **Mode** (Conventional vs. Level-of-Detail AR) and a within-subject variable **Task** (9 tasks within three types of applications). The main purpose of the experiment was to investigate and evaluate the usability of the Level-of-Detail mechanism in AR. To eliminate learning and fatigue effects, we opted for **Mode** as a between-subject variable.

In order to explore the general effect and efficiency of Level-of-Detail on users' performance (in terms of completion time, average distance and average head rotation), we had users perform a variety of different tasks across different applications, leading to **Task** as a secondary within-subject independent variable (nine different tasks in total needed to be completed by participants, which will be detailed in 5.4). Participants were randomly assigned into one of two groups (group A using conventional mode, and group B using Level-of-Detail mode) and required to perform the nine tasks. Participants were not informed which type of application they would be using. A post-experimental questionnaire assessed user satisfaction and impression of the mode they had used.

5.2 Objective Measurement: Task Performance

The following dependent-variable measurements were collected and analyzed:

- **Completion Time:** The time for a user to complete a task.
- **Head Rotation:** The average amount of user head rotation recorded during a task.
- **Average Distance:** The average distance between the user's and application's positions throughout a task.

5.3 Subjective Measurement: User Experience

All participants were required to answer eight seven-point (i.e., 1 = strongly disagree, 7 = strongly agree) Likert questions in a post-study survey questionnaire, all of which were specifically asking for comments about the mode (Level of Detail or Conventional) that the user experienced (i.e. ease of readability, ease of interaction, enjoyment of interaction, ease of task comprehension, ease of task completion, ease of content comprehension, and overall enjoyment, and also amount of fatigue it caused).

5.4 Tasks

The experiment consisted of nine tasks, which took part in three applications (Article Display, Touchpad and Weather Display). The applications were designed to represent mobile applications that a user would frequently use in their daily lives. Each task asked the user to look at an application's content and correctly complete an interaction. The task was complete when the participant performed the correct interaction. The application was displayed at the same distance from the user for all of its tasks. However, the size of the application differed among tasks in order to allow users in group B to experience the effects of Level-of-Detail Mode. The following paragraphs describe each application and the intended mobile application it represents. The tasks are summarized in Table 1.

Article Display This application is intended to represent a news or blog application. Tasks within the Article application were intended to simulate how a user would select an article to read from a news application. These tasks consisted of asking the participant to click on various article components on the application screen. As shown on the left column in Figure 5, this application consists of a display of six research articles on a sliding slate background. Each article has a title, list of authors, and date of publication. A user interacts with this application by scrolling and clicking on text. In Conventional Mode, all information is displayed, regardless of application size. In Level-of-Detail mode, the application data is optimized to fit the space given by limiting the display to article titles, and additional information (such as author and date) is displayed as a user hovers over an article.

Touchpad This application is intended to represent a cell phone dial pad. Tasks within the Touchpad application were meant to simulate how a user would interact with a standard phone alphanumerical keypad. These tasks consisted of asking a user to dial a number (including such with mnemonic letter sequences) and then press the call button. As shown in the middle column of Figure 5, the digital Touchpad consists of a set of buttons with the typical arrangement of digits, symbols and alphabetical letters needed in dialing. In Conventional Mode, the Touchpad always displays both numbers and their corresponding letters, as found on a standard telephone keypad. In Level-of-Detail Mode, the Touchpad omits letters on the keypad when space or visibility is limited.

Weather Forecast This application was intended to represent a weather application for the user's current location. Tasks within the Weather Forecast application were meant to simulate how a user would check the weather. These tasks consisted of asking a user to find information on the Weather Forecast display and enter it via a standard AR keyboard. As shown in the right column in Figure 5, this application shows the weather for the user's current location (here anonymized to a fictitious location). Specifically, the application displays the location, current temperature in Fahrenheit, the forecasted high and low temperatures of the day, hourly temperatures, humidity, visibility, wind speed, UV index, adjusted temperature, and a written description of weather conditions. In Conventional Mode, the Weather Forecast always displays all information. In Level-of-Detail Mode, information is omitted as size or visibility decreases, leaving only current temperature, location, and hourly temperatures at the smallest setting.

Task	App name	Instruction
1	Article Display	The task is to click on the 2nd article title.
2	Article Display	The task is to click on the 5th article title.
3	Article Display	The task is to click on the Author of the 4th article.
4	Article Display	The task is to click on the Date of the 1st article.
5	Touchpad	The task is to type the phone number 949-927-4580 on the touchpad, then click call.
6	Touchpad	The task is to type phone number (such as) 567-CALL-COM on the touchpad then click Call.
7	Weather Display	The task is to find the current temperature on the weather application and type it into the keyboard.
8	Weather Display	The task is to find the current location on the weather application and type it into the keyboard.
9	Weather Display	The task is to find the low temperature on the weather application and type it into the keyboard.

Table 1: All experiment tasks with their number, application name, and instructions.

6 User Study

Our university's local human subjects committee approved the user study prior to running the experiment on participants. Before beginning of the user study, we hypothesized that the Level-of-Detail mode (group B) would outperform the Conventional mode (group A) by completing the tasks in less time. This hypothesis was driven by the idea that less content on an application would allow for increased usability of the available content. For example, a user would be able to find and select information such as article titles more quickly with less relevant information (such as authors and dates) removed.

The user study was conducted with a Microsoft HoloLens 2 headset in an experiment room at the university.

6.1 Participants

45 individuals participated in the user study (26 self-identified as female, 16 as male, 2 as non-binary, and 1 as genderfluid) with ages ranging from 18 to 42 years old. All participants volunteered to participate in the experiment with nominal financial compensation. The criteria for participating in the user study was that participants have a 6th grade English reading level and not have a health history containing seizures or epilepsy. In terms of experience using AR headsets, 21 participants had used an AR headset prior to the study and 24 had not. Among the 21 participants experienced with using an AR headset, 5 were self-assigned experts having used such a device for more than 10 times.

6.2 Procedure

Prior to arriving at the experiment space, participants signed an informed consent form explaining the experiment procedure, purpose, risks, and other necessary experiment information. Additionally, they filled out a pre-study questionnaire to gather their demographic data. Each participant was made aware that the study was evaluating the Level-of-Detail design in AR and that we were recording their performance for data collection. The experiment was divided into three phases:

Training Each participant was given a set of training tasks and a pre-experiment exercise. There were a total of 8 training tasks, which took approximately 5 minutes to go through, consisting of clicking buttons, sliding a slate and typing on a keyboard. Additionally, each participant previously partook in a separate pre-experiment exercise evaluating multitasking in AR for approximately 40 minutes. With the combination of the 5-minute training tasks and the 40-minute separate pre-experiment exercise, the participants were given around 45 minutes of experience using the HoloLens2 before beginning the Level-of-Detail AR formal experiment. The purpose of the training was to increase each participant's comfort with the device and reduce the negative impact of incorrect manipulation on evaluating the usability of our applications.

Formal Experiment After completing the training phase, participants undertook the formal experiment and completed all 9 tasks, knowing the time to complete a task was being measured and that the only way to complete a task was to perform the correct interaction described in the task description.

Post Survey After completing the training and formal experiment, participants were asked to complete a post-study questionnaire. The questionnaire was used to evaluate their opinion on the usability of the applications throughout the tasks. Our main focus was on the difficulty interacting and reading the applications. We also inquired about the participants' enjoyment using the applications throughout the tasks.

6.3 Data Collection

Overall, 405 trials were recorded: 9 tasks $\times$ 45 participants. For each trial, performance was measured by time to complete the tasks. In addition, the user's head rotation and head position were also

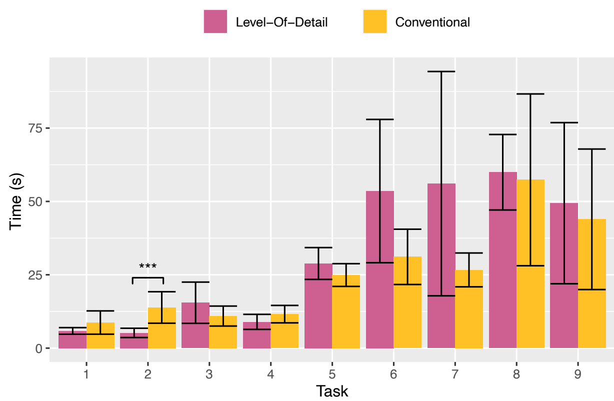

Figure 6: Effect of mode (Level-of-Detail/Conventional) on time to complete each task. Error bars show 95% confidence interval and *** indicates $p < 0.001$.

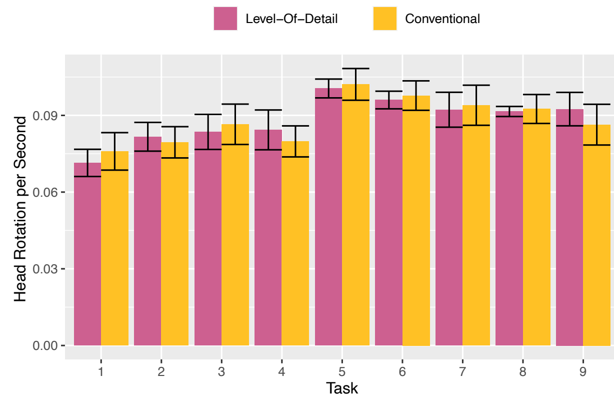

Figure 7: Effect of mode (Level-of-Detail/Conventional) on time-averaged head rotation for each task. Error bars show 95% confidence interval.

collected throughout each task. Head position data was used to determine the distance between a user and an application.

7 Results

The dependent variables we were interested in included the time taken to complete each task, the amount of head rotation during each task, and the distance between a user and an application during each task.

A Shapiro-Wilk test found that our collected data was not normally distributed. We performed an Aligned Rank Transform mixed model ANOVA [24] to evaluate the significance of the relationship between mode and task performance. For our post-hoc analysis, for which there is no established aligned rank transform methodology yet given our study design [6], we used a Mann-Whitney U test with Bonferroni correction. For all significance tests, a p-value of 0.05 was used.

7.1 Task Performance

Completion Time: First, we evaluated the effect of mode on time for both groups on a per task basis. We measured the total time (in seconds) per user to complete each task and analyzed the difference in performance between the two groups. We removed two participants (one from each group) from the analysis of tasks 7, 8, and 9 (which involved the Weather Display) due to their difficulties using the AR keyboard, leading to experimenter intervention. An ART ANOVA ($F_{1,8} = 7.805, p = 0.008$) revealed a significant task-dependent difference between modes, as well as an interaction between task and mode ($F_{1,8} = 3.77, p = 0.0003$). In addition, ART ANOVA revealed a difference in time between tasks ($F_{1,8} = 55.2, p < 2.22 * 10^{-16}$), which is unsurprising given the purposeful difference of tasks. A Bonferroni-corrected Mann Whitney U test ($p < 0.0001$, *large* effect size, $r = 0.586$) revealed that the Level-of-Detail group was significantly faster than the conventional group for task 2, which was to click on the article title of the fifth article (out of six) on display. Other tasks did not show a statistically significant difference in time in the Mann Whitney U test.

Head Rotation: Next, we evaluated the head rotation throughout each task. We recorded head rotation quaternions frame-by-frame throughout each task, and we calculated average head rotation as the summed norm of the difference of quaternions between subsequent frames [11] divided by the time for each task. An ART ANOVA revealed significant differences in head rotation between tasks ($F_{1,8} = 61.02, p < 2 * 10^{-16}$) but a lack of a significant effect of mode ($F_{1,8} = 0.0039, p = 0.95$) on mean head rotation. However, we found an interaction between task and mode ($F_{1,8} = 2.06, p = 0.039$), indicating that the impact of mode on head rotation is task-dependent.

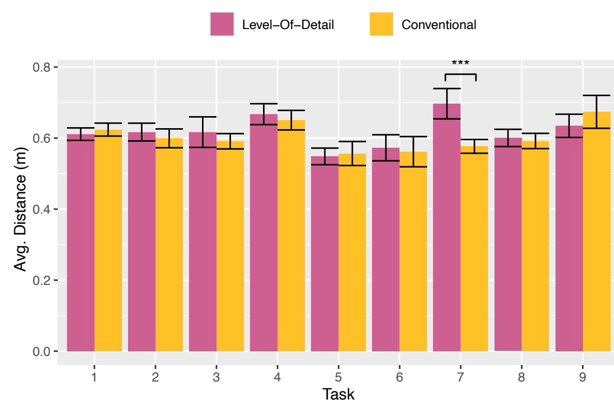

Figure 8: Effect of mode (Level-of-Detail/Conventional) on average distance between user and application for each task. Error bars show 95% confidence interval and *** indicates $p < 0.001$.

7.2 Motion Behavior

Finally, we evaluated the average distance (in meters) from the application the user maintained during each task. All participants began each task at the same distance from the application. Participants were given no guidelines on how far from the application they should remain throughout each task. An ART ANOVA revealed a significant task-dependent difference between modes ($F_{1,8} = 4.65, p = 0.037$), as well as an interaction between mode and task ($F_{1,8} = 4.29, p = 6.18 * 106{-}5$). A Mann Whitney U test ($p < 0.001$, *large* effect size, $r = 0.593$) revealed a significantly higher average distance for task 7, which was to find the current temperature on the Weather Display. Other tasks did not have statistically significant differences in the Mann Whitney U test. Additionally, the ART ANOVA revealed a difference in average distance between tasks ($F_{1,8} = 16.36, p < 2.22 * 10^{-16}$) and this can be attributed to the different requirements of the tasks.

7.3 User Satisfaction

All participants answered 10 seven-point Likert questions in a post study questionnaire, 8 of which were specifically about the mode (Level-of-Detail or Conventional) that the user experienced. The responses to these 8 Likert scores by mode are shown in Figure 9. Seven out of the eight questions asked the user about their agreement with positive attitudes regarding the usability of the mode (i.e. ease of readability, ease of interaction, enjoyment of interaction, ease of task comprehension, ease of task completion, ease of content comprehension, and overall enjoyment). The eighth question inquired about their experience of, or lack of, fatigue (the only question with an inverted scale regarding usability). Across criteria, users (who did each just experience one of the modes) were consistently more satisfied with the Level-of-Detail mode. When asked about their experiences with the level of detail method, users stated "It was immediately clear to me how the interface worked (the closer I got, the more information was displayed to me)" and "Determining the current temperature on the weather application [was easy]. The information was immediately available to me with no clutter". Another user noted the "Layout [changed], which was expected since viewing a website on your phone versus on your laptop will show a different layout to fit the screen", supporting one of the motivations behind our design, which borrowed from Responsive Web Design.

Two questions in our post-study questionnaire showed the largest difference in self-reported effects: When asked about the ease of interaction, the Level-of-Detail group responded more positively than that conventional group. Interestingly, when asked about fatigue, the Level-of-Detail group reported higher levels of experiencing fatigue than the Conventional group ($p = 0.068$), the only outlier from a numerically higher usability rating for Level-of-Detail AR.

8 Discussion

We hypothesized that the Level-of-Detail AR mechanism would outperform the conventional mode in user performance with regards to time. Our results provide limited support for this hypothesis, in that users in Level-of-Detail mode completed one task significantly faster than users in conventional mode.

Task 2, in which users selected the 5th article title from the Article Display application, was the only task in which users in Level-of-Detail mode completed the task significantly faster than users in conventional mode. The Level-of-Detail mechanism may have given participants a potential way to get around scrolling the slate containing the article displays. From a particular viewing distance on, sufficient amounts of information would be omitted from the article display to fit the fifth entry into the visible portion of the canvas without scrolling being necessitated. A sufficient number of participants may have been adjusting their distance to the article display in this way to create a statistically significant advantage. An additional potential explanation is found in Hick's law, which states that having more options available to a person will increase the amount of time they take to make a decision [10]. The Conventional group was given more options to choose from in selecting information on the Article Display, therefore increasing the amount of time per task. The Level-of-Detail design limited the amount of content rendered in the Article Display to usable content based on the visual angle, which reduced the number of choices available and therefore increased usability through efficient use. This explanation is further supported by the fact that every task involving the Article Display (tasks 1-4) had a lower average time for users in Level-of-Detail mode than conventional mode as shown in Figure 6, although only task 2 had a statistically significant difference. This may also explain why there were no statistically significant differences in time for tasks 5 and 6. These tasks involved entering a phone number rather than finding information on a display and would have been unaffected by Hick's law. Tasks 7, 8, and 9 did involve finding information from the Weather Display, but users

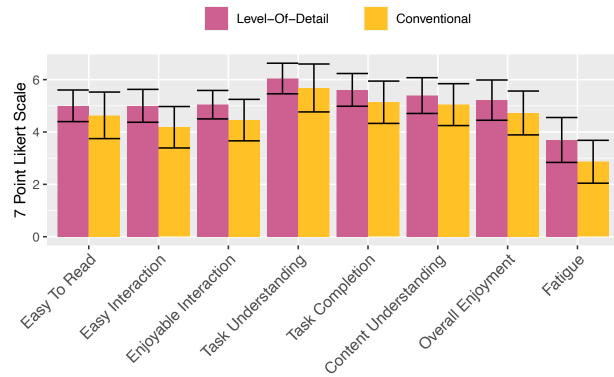

Figure 9: Effect of mode (Level-of-Detail/Conventional) on post-questionnaire Likert scores.

in Level-of-Detail mode did not complete the tasks significantly faster than those in conventional mode. We believe a large part of the reason behind this non-result is the use of the HoloLens' virtual keyboard to input the answers to these tasks. Throughout the experiment, we received feedback on the difficulty of using the backspace on the keyboard and its impact on their ability to complete the task. Time spent on text input (including large variances) may simply drive the task timings here.

Our head rotation analysis gives us some additional information for interpreting the statistical differences in task time for task 2. We speculated above that Level-of-Detail mode allowed users to find relevant information in shorter amounts of time due to the more limited information presented on the display or due to affording a convenient distance from the article display that avoided the need for scrolling. Avoiding scrolling the slate should have coincided with somewhat decreased head motion since the scrolling action necessitates hand-eye coordination. However, we witnessed a small amount of increased head motion per time interval for Level-of-Detail mode in task 2 (cf. Figure 7), albeit not rising to a significant difference. Based on this, while both possibilities remain possible, the scrolling explanation is a little less plausible.

Our final metric was the average distance between a user and the application for each task, as measured in meters. Task 7, in which users found the current temperature on the Weather Display, resulted in users in Level-of-Detail mode having a significantly higher average distance than users in conventional mode. We speculate that Level-of-Detail mode allowed users to more clearly see the relevant information by removing unusable content from the display according to visual angle. This may have allowed users to find and parse the temperature without having to move closer to the display to search for the information. In contrast, users in conventional mode may have needed to move closer to the display in order to find the current temperature among all of the other information presented. This may indicate a greater difficulty in finding the information at the original distance for those in conventional mode.

In terms of user evaluation, there was a trend of higher positive responses from the Level-of-Detail group. It is noticeable as a consistent numerical difference in favor of positive attitudes in the questionnaire result plot (Figure 9). The gained ease of interaction may come at the price of higher fatigue, however, possibly caused by the dynamically changing interface, in spite of our attempts to make the transitions as smooth as possible.

Altogether, our quantitative and qualitative findings help support our belief that Level-of-Detail design can increase aspects of usability, which is important for both user interaction and satisfaction.

9 LIMITATIONS AND FUTURE WORK

This work only considers single-user mixed reality environments. The methodology for determining object visibility when multiple users are interacting with the object, as in a multiple-user mixed reality environment, was not a focused on or tested in our research.

One notable potential drawback to the use of the Level-of-Detail mode from our user study is the trend of an increase in fatigue. It was up to the participants to interpret the term 'fatigue' in answering the question. It is possible that some users find the transitions between different levels of information disorienting, increasing self-reported fatigue. In future work, we hope to further study the effect of our transition strategy on users to better understand its correlation with different dimensions of fatigue.

10 CONCLUSION

In this work we developed Level-of-Detail AR, which dynamically adapts application content based on the visual angle subtended by an application in AR space. We created our mechanism to be generalized, robust, and easy to adopt, bypassing the need to design discrete Level-of-Detail interface variants. AR UI designers and engineers simply need to correctly classify three types of UI components within the AR user interfaces they create, and the levels of details are automatically generated and deployed based on visual subtended angle of the resulting interface.

Through user evaluation and analysis, we found that Level-of-Detail AR results in some limited improvements in task performance. Users completed one task significantly more quickly with the Level-of-Detail mechanism, which suggests that the mechanism can increased an application's usability. In addition, the mechanism allowed and prompted users to experience and understand displayed information from a greater distance on one task.

Finally, user satisfaction responses indicated an overall trend towards higher enjoyability and usability for the Level-of-Detail group, possibly at the cost of increased fatigue. In the future, we would like to further study the user's response to Level-of-Detail AR. The key metrics evaluated in this experiment were performance based on time. Expanding upon this research, future user studies will prioritize the reactions and responses to our mechanism.

ACKNOWLEDGMENTS

This work was supported in part by the Office of Naval Research, under grants N00014-19-1-2553, N00174-19-1-0024, and N00014-20-1-2719, as well as NSF awards IIS-2211784 and IIS-1911230.

REFERENCES

[1] F. Almeida and J. Monteiro. The role of responsive design in web development. *Webology*, 14(2), 2017.

[2] J. Bernacki, I. Błażejczyk, A. Indyka-Piasecka, M. Kopel, E. Kukla, and B. Trawiński. Responsive web design: testing usability of mobile web applications. In *Asian Conference on Intelligent Information and Database Systems*, pp. 257–269. Springer, 2016.

[3] Y. Cheng, Y. Yan, X. Yi, Y. Shi, and D. Lindlbauer. Semanticadapt: Optimization-based adaptation of mixed reality layouts leveraging virtual-physical semantic connections. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, pp. 282–297, 2021.

[4] G. Daskalogrigorakis, A. McNamara, and K. Mania. Holo-box: Level-of-detail glanceable interfaces for augmented reality. In *ACM SIGGRAPH 2021 Posters*, pp. 1–2. 2021.

[5] S. DiVerdi, T. Höllerer, and R. Schreyer. Level of detail interfaces. In *Third IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 300–301. IEEE, 2004.

[6] L. A. Elkin, M. Kay, J. J. Higgins, and J. O. Wobbrock. An aligned rank transform procedure for multifactor contrast tests. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, pp. 754–768, 2021.

[7] K. Gajos and D. S. Weld. Preference elicitation for interface optimization. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*, UIST '05, p. 173–182. Association for Computing Machinery, New York, NY, USA, 2005. doi: 10.1145/1095034.1095063

[8] J. Grubert, T. Langlotz, S. Zollmann, and H. Regenbrecht. Towards pervasive augmented reality: Context-awareness in augmented reality. *IEEE transactions on visualization and computer graphics*, 23(6):1706–1724, 2016.

[9] D. Gómez, A. M. Bernardos, and J. R. Casar. A system to enable level-of-detail mobile interaction with augmented media objects. In *2014 Eighth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, pp. 346–351, 2014. doi: 10.1109/IMIS.2014.47

[10] W. E. Hick. On the rate of gain of information. *Quarterly Journal of experimental psychology*, 4(1):11–26, 1952.

[11] D. Q. Huynh. Metrics for 3d rotations: Comparison and analysis. *Journal of Mathematical Imaging and Vision*, 35(2):155–164, 2009.

[12] Y. Jiang, R. Du, C. Lutteroth, and W. Stuerzlinger. Orc layout: Adaptive gui layout with or-constraints. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, p. 1–12. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3290605.3300643

[13] S. Julier, M. Lanzagorta, Y. Baillot, L. Rosenblum, S. Feiner, T. Höllerer, and S. Sestito. Information filtering for mobile augmented reality. In *Proceedings IEEE and ACM International Symposium on Augmented Reality (ISAR 2000)*, pp. 3–11, 2000. doi: 10.1109/ISAR.2000.880917

[14] W. Lages and D. Bowman. Adjustable adaptation for spatial augmented reality workspaces. In *Symposium on Spatial User Interaction*, pp. 1–2, 2019.

[15] D. Lindlbauer, A. M. Feit, and O. Hilliges. Context-aware online adaptation of mixed reality interfaces. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, UIST '19, p. 147–160. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3332165.3347945

[16] F. Lu, S. Davari, L. Lisle, Y. Li, and D. A. Bowman. Glanceable ar: Evaluating information access methods for head-worn augmented reality. In *2020 IEEE conference on virtual reality and 3D user interfaces (VR)*, pp. 930–939. IEEE, 2020.

[17] Microsoft. Interactable object. `https://docs.microsoft.com/en-us/windows/mixed-reality/design/interactable-object`, 2021. Accessed: 2022-03-06.

[18] Microsoft. Typography. `https://docs.microsoft.com/en-us/windows/mixed-reality/design/typography`, 2021. Accessed: 2022-03-06.

[19] K. Pfeuffer, Y. Abdrabou, A. Esteves, R. Rivu, Y. Abdelrahman, S. Meitner, A. Saadi, and F. Alt. Artention: A design space for gaze-adaptive user interfaces in augmented reality. *Computers & Graphics*, 95:1–12, 2021.

[20] S. Sarcar, J. P. Jokinen, A. Oulasvirta, Z. Wang, C. Silpasuwanchai, and X. Ren. Ability-based optimization of touchscreen interactions. *IEEE Pervasive Computing*, 17(1):15–26, Jan 2018. doi: 10.1109/MPRV.2018.011591058

[21] M.-H. Sung, Y. Choi, H. Ko, and J.-I. Hwang. Level-of-detail ar: Managing points of interest for attentive augmented reality. In *2014 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 351–352. IEEE, 2014.

[22] M. Tatzgern, V. Orso, D. Kalkofen, G. Jacucci, L. Gamberini, and D. Schmalstieg. Adaptive information density for augmented reality displays. In *2016 IEEE Virtual Reality (VR)*, pp. 83–92, 2016. doi: 10.1109/VR.2016.7504691

[23] Unity. Unity documentation: Behavior class, 2021. `https://docs.unity3d.com/ScriptReference/Behaviour.html`, last accessed June 2022.

[24] J. O. Wobbrock, L. Findlater, D. Gergle, and J. J. Higgins. The aligned rank transform for nonparametric factorial analyses using only anova procedures. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pp. 143–146, 2011.