MA-EDGE: MID-AIR GESTURE INTERACTIONS WITH EDGE CROSSING

by

Ritesh Udhani

A Thesis submitted to the Faculty of Graduate Studies of

The University of Manitoba

in partial fulfillment of the requirements of the degree of

Master of Science

Department of Computer Science

University of Manitoba

Winnipeg

ABSTRACT

This research work presents "MA-Edge", a novel multi-level mid-air hand gesture(s)-based target selection and interaction technique for large display screens utilizing the screen edges. MA-Edge works with crossing-targets placed along the screen boundaries where users can select them by simply crossing them near the screen edge. We present results from two controlled studies. In the first study, we investigate first-level MA-Edge interaction design parameters, such as input region configurations, with target selection performance and arm fatigue measurement, for optimizing the interaction performance and output. The results show that some edges, input region configurations and target locations perform better than others and also that MA-Edge can support a large number of targets at the first level of the MA-Edge menu. In the follow-up study, we do a comparative evaluation of second-level target selection between MA-Edge with linear menu and Point-and-Select target selection techniques with linear and grid menu configurations, through target selection performance and consumed endurance measurements. The results show that MA-Edge alone at both levels outperforms the combination method for target selection, quantitatively and qualitatively. This work also provides information on other potential interactions using only MA-Edge and MA-Edge in combination with other interaction techniques.

OTHER PUBLICATIONS

[1] Rafael Veras, Gaganpreet Singh, Farzin Farhadi-Niaki, Ritesh Udhani, Parth Pradeep Patekar, Wei Zhou, Pourang Irani, and Wei Li. "Elbow-Anchored Interaction: Designing Restful Mid-Air Input." In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems.* 2021, pp. 1–15.

[2] Michael Gammon, Ritesh Udhani, Parth Patekar, Yumiko Sakamoto, Carlo Menon, and Pourang Irani. "Limb-O: real-time comparison and visualiza- tion of lower limb motions." In: *Proceedings of the 11th Augmented Human International Conference.* 2020, pp. 1–8

ACKNOWLEDGMENTS

First and foremost, I extend my deepest appreciation to Dr. Pourang Irani, whose unwavering guidance and mentorship have been the cornerstone of this transformative journey. Dr. Irani's steadfast support and encouragement have been invaluable.

I extend sincere gratitude to Dr. Rasit Eskicioglu and Dr. Ahmed Ashraf for being on my advisory committee. Additionally, I am deeply thankful to Dr. Yumiko Sakamoto for her invaluable guidance, encouragement and consistent support throughout this journey.

I am appreciative of the Visual and Disease Analytics program, which not only provided an exceptional educational experience but also contributed to funding a portion of my research. Acknowledgment is also due to Mitacs for their pivotal funding support, crucial in exploring the commercial potential of my ideas. I extend special gratitude to Huawei Technologies for their collaboration on this research endeavor. A special note of thanks to Wei Zhou from Huawei Technologies Inc. for offering me a significant internship opportunity project within this project.

My gratitude extends to my colleagues in the HCI lab, whose support and camaraderie enhanced this journey. And finally, my deepest appreciation goes to my family. Their enduring support has been my anchor through every high and low. To my wife, my mother, and my father—your unwavering encouragement made reaching this milestone possible.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

1

INTRODUCTION

With considerable improvements in display technologies and their increasing affordability, large displays such as SmartTVs, projected screens and/or video walls have become very popular. Consequentially, large displays are now ubiquitous and appear in many environments, including homes, shopping malls, airports and museums. However, interaction with such displays may be inconvenient and challenging due to a number of factors, such as the size of the display [10, 56], the distance of the display from the user [22, 63], the types of interactions supported and types of interactions required [9, 47]. Expectantly, researchers have been exploring ways to simplify interaction with such large displays and provide end-users with an enhanced interaction experience. Apart from the popular button-based remote controllers and the traditional input methods using mice and keyboards, additional means for interacting with large displays have been explored, namely, laser pointers [55, 65], personal mobile devices [5, 22, 26, 37], virtual pointing and selection techniques [10, 56, 63], and wearable devices, such as smartwatches [13, 18, 62]. Even though some of these approaches work well, they rely on a user-held or user-worn input device, bulky tracking systems, require a user to interact in proximity with the display or require touching it, which has been a concern in recent times with shared screens [9, 34, 47].

An emerging and promising approach for interaction with such large displays is the mid-air hand gesture-based (interchangeably referred to as hand gestures or simply gestures) interaction. This approach appears to have tremendous potential as it allows users to interact with displays by simply using their hands and bodies without requiring any input device manipulation for control. Although hand gestures may appear convenient in this regard, they come with their own set of challenges. Some of these challenges may include problems in accurately detecting hand gestures [45] (e.g., in low-light conditions, from subject background limitations, occlusion issues, distance of the user and in multi-gesture hand-gesture-based interactions), accidental activation of a gesture by the user (Midas Touch [9, 14, 64] issue), and arm fatigue from prolonged usage of gesture-based interactions (Gorilla Arm effect) [29, 62]. Due to these limitations, many mid-air-based gesture interaction works in the past have assumed freehand interaction [42, 63] and have proposed interaction techniques that are restricted to research labs due to technology constraints, and thus, their real-world usage is either limited or less optimal. For instance, while the Mid-air-Pan-and-Zoom [40] uses a large VICON setup, Gunslinger [29], which attempts to reduce the fatigue experience in interaction, needs a Leap Motion Controller mounted on the user's body. Nevertheless, despite their current shortcomings, mid-air hand gesture-based interactions still appear as one of the most promising ways of interacting with large displays, and many works, such as this research, are directed toward mitigating these associated challenges and exploring ways to enhance their performance.

An essential requirement for mid-air gesture interaction is quick and accurate target selection. Point-and-Select technique with mid-air gestures, similar to the desktop mouse-based Point-and-Select, is a popular approach for target selection [14]; however, the selection efficiency of this technique can be affected by hand tremors and target size as it requires a user to bring the pointer to the target with precision for selection [46, 63, 69]. Proxemic Cursor [67], Air Tap and Thumb trigger gestures [63] have attempted to mitigate this issue through nearest target detection from the pointer and by reducing hand movement for selection, respectively. Other approaches such as Summon-and-Select [14], semaphoric gesture, [23] and Wrist-Tilt-and-Pinch [42] have attempted to mitigate this issue by avoiding pointing and using a large number of gestures, but that may be difficult to detect and differentiate from each other from a relatively longer distance. Another target selection approach that does not rely on precise pointing is the crossing-based approach [1, 3, 13, 31, 61, 68] that has been shown to be as effective as the popular point-and-select technique for other input modalities, such as pen, touch, AR and VR ray casting [1, 30, 60, 61], while also offering performance benefits through multiple consecutive interactions [2, 11]. This crossing-based selection approach has also been used in interaction with remote displays [39] and VR/AR devices [19, 60, 61, 71, 72]. By virtue of its design, crossing-based selection can also be well suited for the purpose of target selection in the development of the new mid-air hand gesture-based interaction technique for interaction with large display screens.

Another important design criterion is the placement of targets for selection on the screen. For instance, multiple crossing targets placed close to each other [1, 30, 39, 59] in the central areas of the screen can interfere with one another during user target selection or may simply get selected with freehand pointer browsing at the center of the screen [39]. Target menu placement in the center regions also causes occlusion of content behind the menus, negatively affecting the user experience. Considering these issues, a new mid-air hand gesture-based interaction technique can benefit to an extent from some lessons learnt from the touch input-based interaction techniques. Touch input is embedded in almost all non-button devices (smartphones, touchpads, remote controls) and very commonly utilizes bezels, also known as edges, for ease of interaction. Bezels offer many impressive benefits, such as eyes-free interaction with the device [24], lower content occlusion on the touch screen [41], and quicker access to apps and menus [54]. Some of such advantages of bezel-based interactions may translate well for mid-air gesture-based interactions as well, but they still remain largely unexplored. Target placement along the edges will also reduce the inadvertent target selection during pointer navigation and also ease the interference between any two crossing-targets due to their strategic positioning along the edges.

Inspired by the benefits of bezel-based interactions on touchscreens and intrigued by their potential in combination with crossing-based target selection for large displays through mid-air hand gesture-based interactions, we introduce MA-Edge. MA-Edge is short for mid-air hand gesture-based edge interaction and is a novel interaction technique that utilizes the edges of a regular display screen to anchor multiple selectable targets, reducing content occlusion on the central areas of the screen and moving interaction to the typically unused areas, i.e., the edges. The targets could be anchored to any of the four edges of the screen, and each edge could have multiple targets. MA-Edge permits freehand pointer control using only a camera placed on the display, i.e., without any complex hardware, and through simple and fewer mid-air hand gestures. This simplified pointer control executes target selection by simply crossing the targets over with the pointer. Each target selection either completes the interaction or opens another set of edge-anchored targets, providing a multilevel menu support, that could be selected by reverse edge crossing. The presence of targets on the edges and a simple target selection method allow quick access to the content by single or consecutive edge crossing for target selection. A detailed description of the technique is provided in the later sections of this work.

As with any new interaction system, an essential objective of MA-Edge is to allow for efficient interaction with large display screens. To achieve the same, the first user study (hereinafter "first study") focuses on identifying and empirically testing the impact of design parameters, such as different input region configurations in the user's motor space, on selection time, throughput and fatigue during target selection at the first level of MA-Edge menu. The first study also aims to identify the number of targets that can be optimally accommodated on edges for the first level of selection without performance degradation and if any target zones have advantages over the others. The first study also tests if MA-Edge follows the Fitts' paradigm of target selection or if any other relationship can be observed between target selection time and the target's index of difficulty. Using the results obtained from the first study, in the second study, we explore the second level of the multilevel MA-Edge interaction and compare it with the different configurations of the popular point-and-select [1, 3, 13, 31, 61, 68] technique to identify any performance advantages that MA-Edge has to offer over the other. The description of the two studies is preceded by a section on the gesture recognition system required for MA-Edge and a brief description of the MA-Edge design and setup. Following the user studies sections, to demonstrate the wide scope of MA-Edge, we show the other possible interactions using MA-Edge, such as scrolling, and how MA-Edge can also be combined with various popular menu styles, such as pie menu and grid menu.

The contributions of this research are twofold: (1) to introduce the novel multi-level MA-Edge interaction technique and also provide information on the design parameters to be considered for optimizing the MA-Edge interaction performance and output, including the edges, input regions and target locations that work better than the others, affecting target selection performance and also the number of targets that can be supported at the first menu level, through systematic exploration, such as analysis of throughput, selection time, and arm fatigue; (2) to provide a comparison of the second-level target selection with MA-Edge and the popular point-and-select interaction technique, with linear and grid menu configurations, showing the target selection performance and consumed endurance advantages with MA-Edge as opposed to the Point-and-Select interaction configurations. Based on these contributions, we conclude with a discussion of the potential of MA-Edge interactions and future beneficial research directions to enable our vision of simplified and efficient interactions for large-screen displays through mid-air hand gestures.

2

RELATED LITERATURE

We begin this section with the mid-air hand gesture-based target selection literature, including the associated issues such as Midas Touch and fatigue problems. This is followed by the literature on the crossing-based target selection technique and the bezel-based interactions with touch-based hand gestures.

2.1 MID-AIR HAND GESTURE INTERACTION

Interaction with large displays has been studied well through the use of interaction devices, such as generic controllers, laser pointers [4, 55, 65], personal mobile devices [5, 22, 26, 37] and smartwatches [13, 18, 62]. This list also includes the virtual point-and-select techniques [10, 56, 63, 66, 67] based on proximal gesture detectors [58], complex and bulky tracking systems, and touchscreens. Using an additional interaction device could pose many practical challenges. For instance, the device may not always be in a reachable space, run out of battery, encounter pairing issues with the display, or even be broken. Some require a user to interact in proximity with the display and/or touch it, which has been a concern lately with shared screens [9, 34, 47] in the post-COVID era. In our work, we intend to eliminate the requirement for any additional interaction device to be operated by the user and avoid using a device constraining the user location to device proximity and focus solely on mid-air freehand interactions.

2.1.1 *Gesture detection issues*

Hand gesture detection from a distant camera, say placed on a Smart TV display, offers the advantage of not requiring a user-operated input device or a detection system in proximity. However, gesture detection from this relatively farther distance can be prone to errors. This is due to various reasons, such as the distance of the subject from the camera, environment lighting, subject background limitations [73], camera resolution, object occlusion [28], self-occlusion [28], algorithm accuracy and datasets used for training [45]. On the same lines, complex hand gestures are even more difficult to detect as compared to simple ones due to some of the issues listed above. Given such constraints, a novel mid-air hand gesture-based technique could rely on using simpler and fewer gestures.

2.1.2 *Interaction through hand gestures*

Several works have proposed different techniques for interacting with on-screen interface elements, such as pie menus, target pointing, selection, text entry, drag, zoom, pan, and 2D and 3D object manipulation [27, 36, 40, 57, 63]. For target selection, the point-and-select technique (from the desktop-based point-and-select technique) has been a popular approach [14, 32, 67]. In this approach, a user points to the target through approaches such as ray casting or relative pointing, rather precisely and accurately, before making the selection [63]. The selection of target may be made by the same hand used for pointing [4, 12, 63, 66] or through other means such as the other hand [62]. Using the same hand for both pointing and selecting the target can be error-prone [6, 69]. Mainly, errors can occur from the involuntary movement of the hand when selecting smaller targets [46, 63]. Some techniques proposed by researchers to reduce such errors in target selection include Air Tap and Thumb Trigger gestures [63] and Squeeze and Trigger gestures [4]. All these techniques rely on the subtle movement of fingers to reduce the hand movements when making a selection. Another approach is Breach (similar to AirPush [58]) [4]. However, all these listed techniques utilize either a large high-precision motion tracking system or a gesture detector, such as a Leap Motion controller, in proximity to measure the subtle movement of fingers, causing them to be primarily employed as experimental in-lab platforms or for limited purposes.

Some researchers attempted to remove the pointer precision problem from the equation by introducing new techniques, such as utilizing multiple gestures or associating the nearest widget to the focus of the pointer [67]. Wrist-tilt-and-Pinch gestures [42] and Summon-and-Select [14] interaction techniques avoid the use of a conventional pointer and invoke the menu through the use of a gesture. For making further interactions, they use even more gestures. Using such multiple gestures, as mentioned earlier, can be subject to gesture detection issues and will be avoided for our requirement. Moreover, another problem with using multiple gestures is that if the gestures are too complicated, straining or tiring, then they may act as a deterrent, and the users might not be comfortable using them [49]. Alternative techniques rely on many semaphoric gestures[23]; however, learning several gestures for interaction may not be feasible [38] for a more significant acceptance of the technique.

Hence, a mid-air freehand input technique that can minimize the number of hand gestures and rely only on a few simple ones for easy detection from a camera located on the display will help increase the acceptance and usability of mid-air gestures with a larger user base.

2.1.3 *Arm fatigue and Midas Touch issues*

Since gestures meant for interaction could also be used by the user for other purposes, they may lead to inadvertent activation of commands. This issue is termed as the Midas Touch effect and is a known issue with mid-air hand gestures [9, 14, 64]. One reason for using multiple gestures in the works of Gupta et. al. and others is to prevent this accidental activation of commands. However, as discussed, using multiple hand gestures requires to be avoided for this work.

When performed for a prolonged time, mid-air gestures may also tire a user's arm and cause what is known as the Gorilla Arm effect. Therefore, one aim of the design of a novel technique should be to mitigate this issue and measuring arm fatigue when creating any mid-air gesture-based systems, such as this research, is of particular significance to study its efficacy [8, 16, 21]. Studies have considered this factor in their designs. Gunslinger [29] and Anchored-Elbow [62] allow users to perform hand gesture interactions without raising their arms and in restful postures. However, they both rely on additional devices, such as the leap-motion sensor or a smartwatch on the user's body, to identify the gestures. Wrist-tilt and pinch [42] and Summon-and-select[14] selection techniques also reduce arm fatigue by avoiding direct pointing but have other issues as well as discussed before.

2.2 CROSSING-BASED TARGET SELECTION

Another popular target selection technique is the trajectory-based crossing target selection, also called goal-crossing, [1, 3, 13, 30, 31, 60, 61, 68], wherein a user moves the pointer on the screen across a target to cross it once and select the target. Such crossing-based target selection has been shown to be equally or more efficient as compared to the point-and-select technique for various input modalities such as pen, touch, AR and VR ray casting [1, 30, 60, 61]. With a mouse input device, it has been shown to perform better for people with motor impairments [68]. This technique also offers a more fluid selection performance in selecting multiple consecutive targets [2, 11] due to its trajectory-based continuous selection method than Point-and-Select, which is a discrete selection method.

Since target selection happens only by crossing and does not require another selection method, for mid-air gestures, the system needs to disambiguate pointer movement from actual target selection. This is particularly a concern for crossing-targets positioned in the middle of the screen when target selection happens with a single crossing. Hence, Nakamura et al. [39] proposed double crossing of the target, i.e., pointer movement in a particular direction across the target followed by reverse crossing in the opposite direction to separate selection from movement. Even though Nakamura et al. [39] implemented this using an LED worn by a user on their finger, with the latest Computer Vision techniques, it is possible to implement such a selection technique utilizing a single gesture.

Another known issue with crossing-based interactions are the distractions. Distractions are selectable crossing targets which if placed close to the desired target, could affect the performance of target selection [59]. Therefore, it becomes necessary to take this limitation into account and attempt to overcome it in the design of the new interaction technique for better performance. This can be done by improving the placement of crossing-targets to prevent interference from each other. We discuss this in the design of MA-Edge in Chapter 3

A popular trajectory-based target selection technique is marking menu and has been shown to be very effective in target selection [25]. For orientation-based marking menus, a user can optimally interact with a menu width and depth of 8x2 by using a single mid-air hand gesture [27]. Therefore, marking menus provide a good reference in comparison to evaluate the menu width supported by MA-Edge design.

2.3 EDGE AND BEZEL INTERACTION

As mentioned earlier, the crossing-targets can be distractions to each other, and the target placement needs to be arranged such that the target selection performance is not affected. With this in mind, we look at Bezel-based menus for inspiration in target placement for MA-Edge.

Bezel and edge-based interactions have been widely used on touchscreen devices to either enhance an existing interaction functionality or provide a new one. For example, Roth et al. [50] proposed Bezel Swipe for multiple target selection and various other operations, such as cutting, copying, and pasting, where a user drags the finger across the screen edge, starting from outside. Hinckley et al. [17] demonstrate the use of bezel gestures for multi-finger edge crossing and tap gestures on the bezel, thereby increasing the number of touch gestures available at disposal. For tablets, Bezel-Tap gestures [54] allow quick activation of an application on a device in sleep mode through either a quick succession of tap gestures starting from bezel followed by a tap gesture on-screen or tap on bezel followed by a tap-slide on screen.

Bragdon et al. [7] suggested that bezel gestures require less attention as compared to the precise selection of soft buttons on a touchscreen phone. In devices with even smaller touch screens, such as smartwatches, the selection area is limited [43], and finger occlusion [51] from interaction impacts target selection. Here, B2B-Swipe [24], and BIS [70] provide new bezel-initiated swipe gestures for selection on rectangular and circular smartwatches, respectively, allowing eyes-free interaction. Bezel Menu Layouts [20] extended these results by placing eight menu activation areas near the bezel of a smartphone and showed that users could transition from novice to expert mode of interaction within one hour of usage. BezelGlide [41], which utilizes the edge, allows efficient interaction with graphs on a circular smartwatch screen. All these studies show the potential of bezel or edge-based interaction through touch gestures and motivate us to explore edge-based interaction for mid-air gestures to enhance the interaction.

Relative pointing is one way of achieving bezel or edge-based interaction with displays where an input region that maps with the output screen is defined for interaction hand movement. However, Vogel et al. [63] illustrate that using relative pointing causes slower performance due to the requirement of readjustment of the input region every time the user moves. This issue could easily be addressed by attaching the input region to the user's body virtually, such that the region moves along with the user, thus requiring no readjustment. Fixing the location of the input region about the user's body may also allow a user to utilize spatial memory for effective interactions, similar to eyes-free interactions with touch devices. Since relative pointing will not physically restrict a user, off-screen interactions could also happen. Off-Limits [35] utilizes the off-screen region for content selection to reduce the number of steps to complete the operations, implying that off-screen interaction has the potential to enhance the mid-air hand-gesture interactions. Another study, Imaginary interfaces [15], shows that a user can also interact reasonably well without any feedback.

One technique that may be considered closest to our work is suggested by Shizuki et al. [55], where they use a laser pointer in order to interact with a projected screen by using edge crossing as a command. Each edge could be crossed in three ways, in-out, out-in, and in-out-in, allowing three different commands per edge. However, the system requires additional devices, such as a laser pointer and an external camera, to capture the pointer movement and process it for the pointer position. Further, this system does not provide interface design guidelines for menu design utilizing edge crossing.

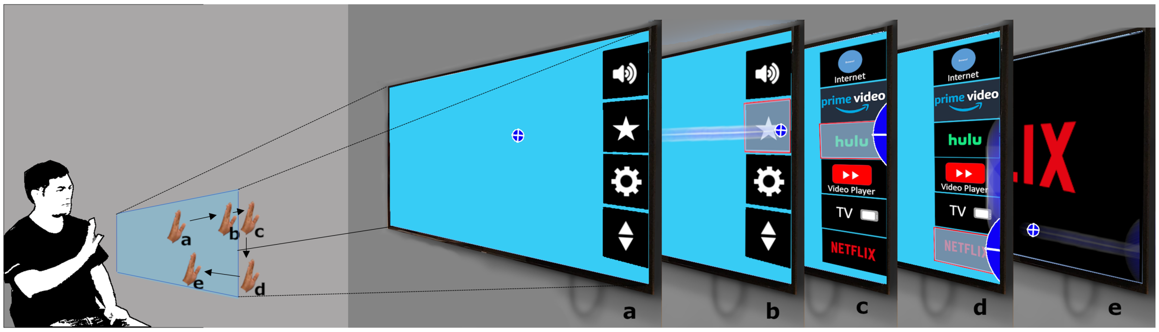
From the above review, we observe that even though bezel or edge-based interactions have been quite popular with touch-based input, there has been no past work evaluating these bezel or edge-based interactions for use with mid-air hand gestures. We next explore the ability to devise an interaction for mid-air input using MA-Edge, which primarily relies on edge-based crossing for input.

3

MA-EDGE INTERACTION

MA-Edge is short for mid-air hand gesture-based edge interaction. MA-Edge is an interaction technique that allows a user to interact with a large screen display, such as a SmartTV, from a comfortable viewing distance through the use of mid-air hand gestures. These hand gestures are captured through a camera mounted on the display and are detected through a gesture recognizer module (hereinafter "gesture recognizer") that runs on this display. The gesture recognizer tracks the hand movement to control a pointer on the display screen, thus permitting freehand cursor movement. The essence of MA-Edge lies in the placement of targets (or menu) along the edges of the screen and interaction with them in the offscreen region as well. These targets can be selected using the pointer by simply crossing them over. The targets can be crossed by moving the cursor out of the screen across the edge and also by moving the cursor back into the screen for multilevel MA-Edge menus. A sample target selection interaction with MA-Edge is shown in Figure3.1. The other possible interactions with MA-Edge are also discussed in later sections.

As explained earlier, the design of MA-Edge evolves from the problems it attempts to address and the requirement to achieve efficient interaction with large



A user initiates an interaction by raising their hand in a simple palm gesture, thereby activating a cursor tracked by hand position and a menu with selectable items along the edge of the screen; (b) to select a menu item, the user moves the cursor towards the menu. Once the cursor is over the item, it becomes highlighted; (c) once the cursor crosses the edge of the screen, the item in the menu gets selected and, in turn, launches another menu corresponding to the selection; (d) next, to navigate to the desired menu item for selection, they move their hand vertically; (e) finally, to select the highlighted target they cross the edge back onto the screen through a controlled hand movement.

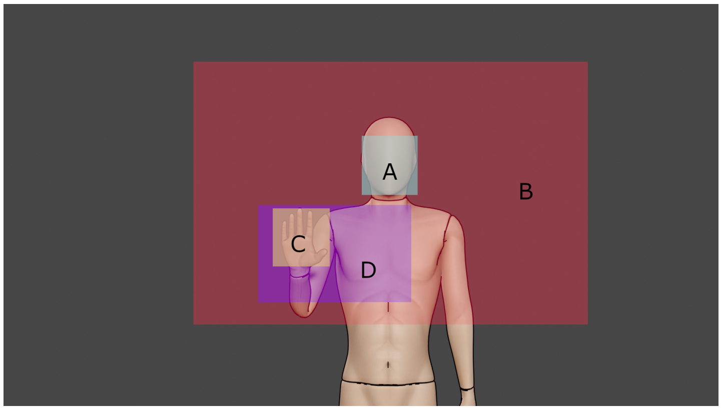
Figure 3.1: A user interacts with a large screen display using MA-Edge on the right edge

displays. MA-Edge can be implemented simply through the use of a simple gesture recognizer module that detects just a few simple gestures combined with a mechanism for target selection utilizing the edges. Therefore, MA-Edge eliminates the need for a complex or bulky gesture detection system or a system that is required to be in the user's proximity to function and instead only relies on a single gesture for target selection purposes, which can be detected with just a simple display-mounted camera. A single gesture activates the hand-tracking pointer for interaction and is distinguished from any other gestures that may not be intended for interaction. In this work, we use a palm gesture and a fist gesture to conduct two separate user studies with MA-Edge. Using such simpler gestures prevents the strain and/or pain caused by complex hand-contorting gestures, while a fewer number of gestures prevents the burden on the user to remember multiple gestures and their associated purposes.

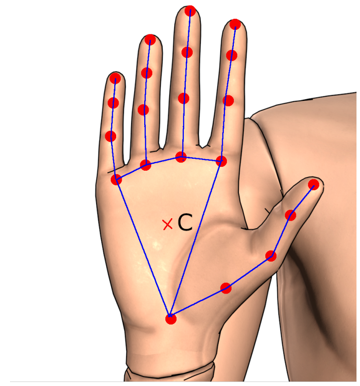
A gesture recognizer needs to be first configured to develop MA-Edge interaction and carry out its empirical evaluation. For this, we utilized a proprietary gesture recognition library and customized it as per the MA-Edge requirements. The gesture library we utilized runs on Android devices (such as smart TVs or smartphones) and takes a continuous image feed from the device's camera (at 25 fps processing speed) to identify the hand gestures performed by the user at run time through gesture recognition algorithms.

The gesture recognition process begins with detecting a user's face in an input image through a face detection algorithm, such as Haar Cascades. The detected face area is marked as a square-shaped facebox (Figure 3.2a, A). Using this facebox, the gesture recognition algorithm defines a big enough rectangular human region-of-interest (ROI) (Figure 3.2a, B) around the facebox such that the ROI can contain the user's hands in general interaction positions. Thus, this human ROI's location and dimensions are at a fixed predefined offset from the user's facebox. The bottom edge of the human ROI is set to about 2.1 times the height of the facebox from the lower edge of the facebox to minimize unintended gesture activation [14] from the resting hand positions in sitting or standing postures. These dimensions were roughly chosen through pilot experiments such that any unintended gesture input is not generated from the user's resting hand position while still being able to reach a low enough height with the hand for gesture detection comfortably.

Once the human ROI is defined, it is fed as an input to a hand detection algorithm, which identifies the user's hand within this input human ROI and creates a square-shaped hand box (Figure 3.2a, C) around the identified hand. This hand box is then fed to a classifier algorithm, which further determines the landmarks Figure 3.2b and the gesture pose of the hand. Next, to detect the hand's position, the gesture recognition algorithm utilizes the landmark coordinates



(a) Camera feed overlaid with regions recognized by gesture recognizer. Regions: A. Face box, B. Human ROI, C. Hand box, D. Activation Area.



(b) Openhand gesture showing centroid location of the hand 'C' and landmarks as red dots

Figure 3.2: Gesture recognition steps

provided by the classifier. We chose the landmarks only in the user's palm to calculate the centroid location in the image coordinates of the camera feed and use it as the hand position (Figure 3.2b). This method of finding the hand position in the camera image feed through centroid calculation minimizes the issue of cursor jump that may occur when the user moves their fingers during hand movement or gesture pose switching. For performance optimization, in subsequent frames, the hand tracker algorithm locates the new position of the hand box. It feeds this information to the classifier algorithm to fetch the new hand coordinates and the gesture pose according to the updated hand position. This detected hand position, along with the detected gesture pose, is then reported by the gesture recognition library to the application implementing the MA-Edge interface for user interaction.

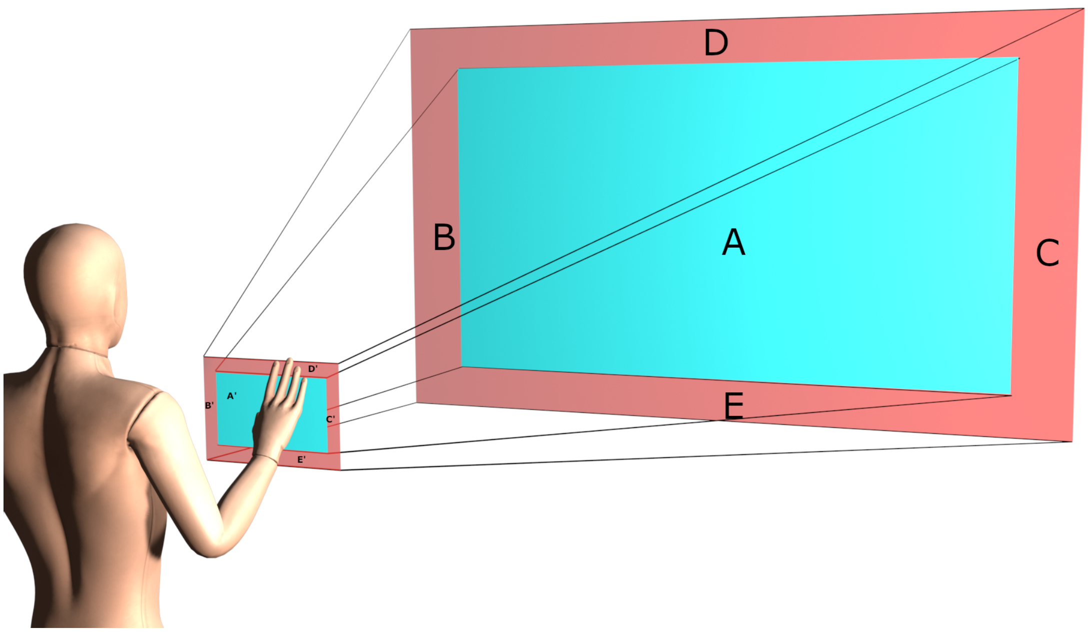
To report the hand location data to the application for cursor control, we chose a relative pointing design [63] over absolute pointing with ray casting. One of the reasons is that the former can be done with a simple single camera setup, as discussed before, while the latter may require dependency on the detection of finer hand movements, in turn requiring a non-simple gesture detection setup. Another reason is the problem of accuracy from hand jitters which makes absolute pointing less appealing [63]. Further, owing to its inherent interaction design requiring no transformation in the input and output space, relative pointing, like absolute pointing, is also a natural relationship instead of a learned relationship [33], allowing a user to control the cursor without learning it. Since relative pointing requires a 1:1 mapping between the input region and the output space, it is necessary to position the rectangular input region (Figure 3.2a, D) appropriately within the human ROI. This input region is also termed as activation area because once the user creates a hand gesture for interaction within this input region, it activates a pointer on the screen for interaction.

The vertical position of this activation area needs to be kept sufficiently low so that the hand does not overlap with the face. Positioning the activation area in such a manner prevents unintentional invocation of any alternate gesture for systems that may intend to use multiple gestures (for example, a finger over mouth can also be interpreted as a mute gesture). Such positioning of the activation area also provides the benefit of preventing content occlusion by the user's hands. Another reason for the lower positioning of the activation area is the fact that a higher activation area will cause more fatigue to the user's arm from extended interactions [16]. At the same time, the activation area should not be too low in the human ROI to cause unintended hand gesture interaction from resting hand positions. Furthermore, an additional buffer space of approximately 0.5 times the facebox height between the lower edge of the facebox and the centre of the handbox is provided to accommodate the fingers to prevent overlap with the face. This is because the centroid for the hand position lies in the lower half of the hand box (Figure 3.2b), which is about the same size as the facebox. Further, since the lower edge of the activation area cannot be below the lower edge of the human ROI, it could, at max be located at a distance of 2.1 times (from pilot tests) the height of the facebox from the lower edge of the facebox.

The horizontal positioning of the activation area can be estimated from the observations on the range of arm movement reported by Veras et al. [62] for anchored elbow, which explains that the proportion of arm movement towards the body is relatively more than that away from it. Therefore, considering the biomechanics of arm movement, it may be appropriate to center this activation area about the inner shoulder joint horizontally so that the left and right edges of this input region are reachable for the user with equal ease. Thus, by conducting pilot runs with some users, we identified an approximate distance of 0.75 times the width of the face from the body center and used this distance as the horizontal center of this activation area.

From the above constraints, the activation area is set at a particular offset in the x and y directions from the facebox. It is worth noting that a specific location of the activation area about the user's body allows the gesture recognizer to get the hand coordinates in the user's frame of reference and makes them independent of the user's distance from the camera, except when standing very close to display, which will impact the proportions. Additionally, it also allows a user to interact freely with the display without the need to adjust/calibrate the activation area before every interaction [63].

As mentioned above, the input region, i.e., the activation area, maps 1:1 with the output region (Figure 3.3). The aspect ratio of the activation area rectangle in MA-Edge is kept the same as the aspect ratio of the output screen to keep the CD gain the same in both the x and y directions for a simplified interaction. As a result, MA-Edge can support a variety of aspect ratios of the output screen, such as HD, UHD, SD and so on. Additionally, since the MA-Edge target selection approach requires a pointer to cross the screen edge, the technique also requires an off-screen area to enable this interaction behaviour. Accordingly, these onscreen and off-screen output regions map with the activation area in the input motor space as shown in Figure 3.3.



'A' is the output screen and 'B', 'C', 'D', and 'E' are the off-screen region for cursor movement (not visible to the user). All these regions have corresponding mapping in the input region at A', B', C', D' and E'.

Figure 3.3: Input output mapping.

MA-Edge allows target selection by placing the targets along the four edges of the screen. The targets may have associated graphical or text icons as shown in Figure 3.1, if required as per the user interface requirement or for a beginner user interaction. Alternatively, for advanced users, the targets may be placed along the edges without any graphical icons as shown in the second user study described later in section 5. This target placement along the edges clears up the central screen area from being occluded, in turn preventing the content from being hidden during user interactions.

Further, as noted earlier, MA-Edge also permits multi-level menus. For this, the targets at the first level can be selected by crossing the edge at the particular target location with the direction of pointer movement outwardly from the onscreen region to the off-screen region. Once a particular target is selected at the first-level menu, it opens another second-level menu (Figure 3.1, (b)-(c)). Menu items (or targets) of this second level menu can be selected by moving the pointer inwardly from the off-screen region back to the onscreen region, crossing over the target at the edge at which the target is located (Figure 3.1, (d)-(e)). Similarly, theoretically, MA-Edge can support menus with more depth (i.e. levels), but we study target selection with two levels empirically in this research. The presence of targets on the edges in MA-Edge combined with the simple crossing-based target selection method, reducing distractions [59], is expected to allow quick access to the content by single and/or consecutive edge crossings [2, 11] for target selection.

4

STUDY1: FIRST-LEVEL MA-EDGE INTERACTION DESIGN PARAMETERS

Having discussed the foundational set-up of MA-Edge interaction technique in the previous section, we set out next to analyze the MA-Edge design parameters that can affect MA-Edge performance through a user study. The study will also determine the number of targets that can fit in the first level of the MA-Edge menu. It will also seek to identify if a mathematical model, such as the Fitts' Law, can be applied to the target selection time and the target's level of difficulty.

4.1 EXPERIMENT SET-UP AND APPARATUS

As our gesture detection library is Android based (Chapter 3), to carry out this user study, we utilized a Huawei P40 Android smartphone, which has inbuilt cameras to provide a camera feed to the gesture library. To simulate the MA-Edge interaction experience with large screen displays, such as smart TVs, we placed the smartphone in front of a user with the front camera capturing the user's hand gestures. These hand gestures and position information would be sent to a graphical interaction application running on the same device, i.e., the smartphone, for interaction purposes. The smartphone's screen was connected through a USB connection to a computer and mirrored on the computer screen using the Scrcpy application. This computer screen was in turn projected through a projector on a large whiteboard right behind the smartphone (Figure 4.1a). To measure user fatigue, a Kinect device was also placed right next to the smartphone device (not shown) and was connected to the computer system for consumed endurance (CE) [16] measurement.

With the gesture library and smartphone combination, using the front camera, the optimum range of distance between the user and the camera is found to be between one and three meters as per our pilot testing. Additionally, the user could be positioned anywhere within the viewing range of the camera and gesture recognition could still be done for them as long as the user is within the optimum distance range, as mentioned above. For the study, we chose a distance of 1.5 meters between the user and the camera, while the user sat straight in front of the camera (Figure 4.1a).

Although any gesture can be used with MA-Edge, we wanted to use a simple and convenient hand gesture for MA-Edge in this user study. Since only a single gesture is required for target selection, we chose fist gesture as it is the closest to the shape of the hand in the resting position. The user need not make a tight fist for gesture detection and a loose fist hand position is also detected as a fist gesture by our gesture recognizer. This advantage permits longer interactions without straining the hand.

We used the Kinect device to measure the consumed endurance (CE) for fatigue estimation from each interaction (as discussed later) as it detects the user's hand and body posture. The kinect is engaged/disengaged using the accompanying CE measurement software running on the Windows system. Since a single interaction was expected to last for only a few seconds, the CE desktop application was modified to sync with the Android smartphone interaction (as discussed later) application to trigger the start and stop of CE measurement accurately. This syncing was achieved through a TCP socket connection between the two devices on the same wireless network, where the interaction application on the Android device sent start and stop events to the CE measurement desktop application to start and stop the CE measurements for every interaction.

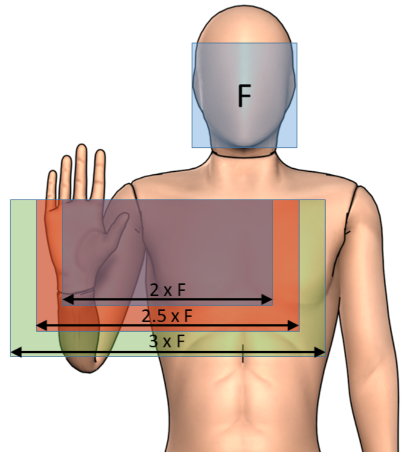
4.1.1 *Activation area*

The graphical interaction application (as will be described below) taking input from the gesture library was a hybrid application (i.e., native + web). The application contained the graphical user interface created using webview in the web part, while the Android framework and gesture library were in the native part. The screen area on the smartphone (without the Android Navigation bar) for the graphical interaction application had a width and height of 714 and 361, respectively, in webview coordinates, yielding an aspect ratio of 1.97:1. As discussed before, since we are constrained vertically between the regions of 0.5 and 2.1 facewidths, but not horizontally, we chose to utilize the complete width of the webview instead of the popular 16:9 (1.77:1) screen ratio used in the UHD, HD, FHD smart TVs to demonstrate MA-Edge usability beyond standard screen sizes.

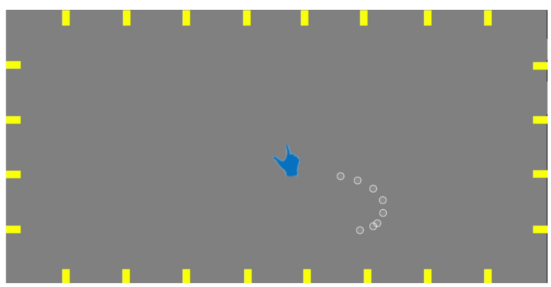
Since MA-Edge requires pointer movement in both onscreen and off-screen areas for interactions, we kept off-screen buffers just wide enough to allow the pointer to move to the off-screen area. For this, we add the off-screen buffer such that the aspect ratio is kept the same. However, to keep calculations simple (useful for test conditions below), a buffer of a marginally larger size of 106 units was added to the left and right of the onscreen area, while only 51 units were added to the top and bottom off-screen areas. This resulted in the output region of size 926x463, in webview coordinates, with a total aspect ratio of 2:1.

As discussed in the previous Chapter 3, there is 1:1 mapping of the input region (activation area) and the output region and CD-gain needs to be kept constant in both the x and y direction for simplicity. Since the aspect ratio of the output region is 2:1, the activation area needs to be in the same aspect ratio. As per the other constraints, the horizontal center of this activation area needs to be about 0.75 times facewidth distant from the center of the face towards the dominant hand (right hand used in this study). The upper edge of the activation area rect is limited to 0.5 times faceheight below the lower edge of the facebox, while the lower edge of this activation area rect is limited to 2.1 times faceheight from the lower edge of the facebox. This leaves us a maximum height of activation area to 2.1 minus 0.5, which is 1.6 times the faceheight, which gives 3.2 times facewidth as the maximum size of activation area in the x direction, through the aspect ratio of 2:1. The lowest width of activation area that could be handled without too much sensitivity was identified through pilot experiments and was found to be 2 times facewidth. Since location and dimensions of input region can impact MA-Edge performance, we chose activation area as a factor for MA-Edge evaluation with 2, 2.5 and 3.0 times facewidth as widths of activation areas as test conditions (Figure 4.1b).

(a) Experiment Set up: Chair for user, smartphone running gesture recognizer and interaction application, and white board for projection



(b) Activation area test conditions: 2 x facewidth, 2.5 x facewidth, 3 x facewidth.



(c) Total targets 24: 8 on each horizontal side and 4 on each vertical side; reference target; pointer with a trail

Figure 4.1: User study #1: Setup and activation areas

4.1.2 *Interface*

To determine the first level menu width, i.e., the number of targets that can fit along the screen edges, we created an Android application using webview, which gave us a screen size of 714 x 361 webview units. Taking the optimal menu width of marking menus, i.e., 8 targets, as reference [27], i.e., 8 targets, we placed three times the number of targets, i.e., 24 targets, symmetrically and in an equidistant fashion across the four edges, for target selection (Figure 4.1c). This large number of targets is expected to create overlapping target selections which will in turn help to obtain the maximum allowable targets that can fit in the first level of MA-Edge. The targets, meant to be point sized, were sized to 10x20 units as a stub so that the user can easily view all of them on the screen from the interaction distance. A reference target, shown as a hand, to mark the start of a target selection trial was placed at the center of the screen. Since for each target there are 3 other targets with the exact same distance from the center, we placed the reference target with a slight offset of 22.5 webview units in the lower and right direction to create asymmetry between such target combinations and have a different theoretical Index of difficulty (ID). A circular transparent pointer of 9 units in diameter was provided for hand tracking. The transparent center would allow one to see the pointer hotspot well under the pointer. The pointer also included a trail to easily and quickly locate it on the screen after starting interaction from an idle position.

4.2 EXPERIMENT DESIGN

We use a single factor, within-subjects design for this study, with activation area being the independent variable with three test conditions of 2, 2.5 and 3 times facewidth as the width of the activation areas. The three test conditions will be identified as Block 1 (activation area of 2 times facewidth), Block 2 (activation area of 2.5 times facewidth) and Block 3 (activation area of 3 times facewidth). The order of blocks was counterbalanced using the Balance Latin Square method. Each block has 24 targets, and at a time, only one target is displayed on the screen for the user to select, with each target presented a total of 5 times (not consecutively) to each participant in a completely randomized order for target selection. The dependent variables measured for each trial of target selection are time of selection, consumed endurance (CE), throughput and intercept size (described below).

Since each target within a block is presented in a completely random order, a 4x3 factor within-subject comparison can also be performed using block edges as a factor with the four sides as four test conditions along with activation area as the other factor with three blocks as test conditions. Another single-factor comparison could be performed for each of the 24 targets individually with the activation area as the factor to identify target level differences between the three blocks of activation areas. For each of these edge-based and target-based analyses, time, CE, throughput and intercept sizes are the dependent variables.

As we want to count the number of targets that can fit in each block, as mentioned earlier, we overfilled the block with 24 targets, evenly placed along the edges. Now, after target selection is performed, neighbouring targets' intercepts, i.e., distance of selection from the target, are expected to overlap. The number of non-overlapping target selection intervals will provide us with the number of targets that can fit in a block

4.3 PARTICIPANTS

We recruited 12 participants (3 females, 9 males) with ages ranging between 20 and 38 years (M = 28.4 years, SD = 5.6). All but one participant were university students and were right-handed, except two. One was ambidextrous from the remaining two, and the other left-handed participant claimed enough skill for interaction with their right hand.

4.4 TASK

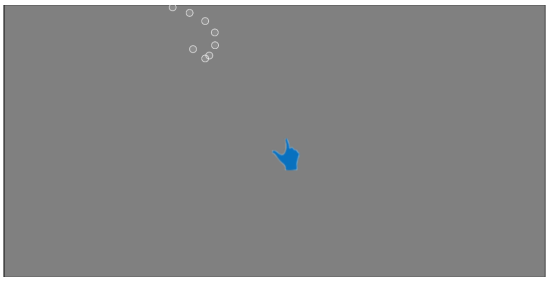
We chose a Fitts' Law style target selection task, also called goal crossing [1, 30], in this user study. A user would first be presented with one of the three blocks as per the order in the experiment design. They would use fist gesture to move the pointer on the display screen and cross the targets to make a target selection. To initiate a trial in a block, the user would first be required to select the reference target displayed at the centre of the screen (Figure 4.2a) by simply crossing it with the pointer. As soon as this reference target is selected (Figure 4.2b), only one MA-Edge target that needs to be selected from the 24 MA-Edge targets in the block would appear on one of the edges of the screen, for selection through edge crossing (Figure 4.2b). As mentioned earlier, the order in which the individual MA-Edge targets are presented is completely randomized. The selection of the reference target also starts a timer which will measure the total time taken by the user to select the actual MA-Edge target after selecting the reference target. It also sends a trigger event to the consumed endurance measurement application, which starts to measure the consumed endurance of the user's arm throughout the trial.

To select the MA-Edge target, a user moves the pointer in the direction of the target. This pointer movement trajectory is also drawn on the screen for participant's reference(Figure 4.2c). A user will have to cross the edge (Figure 4.2c) on which the target appears from a location that is as close as possible to the target's location. The selection had to be performed as fast and as accurately as possible as per task instructions. A correct edge selection was shown by a green bar across the edge (Figure 4.2d), whereas an incorrect edge selection is shown by a red bar across the edge (Figure 4.2e). For a user to check their accuracy, the trajectory drawn allows them to check how far from the target they made the selection. Since the MA-Edge target is just a point on the screen, highlighted by a small stub (size 10x20) along the edge, the user will cross the edge at a certain distance from the target. We refer to this distance as the intercept on that edge for that target. Once a target has been selected (Figure 4.2d) we trigger a trial end event for the CE measurement application to finish the consumed endurance measurement of a trial, and we note the intercept size, selection time and consumed endurance for this trial.

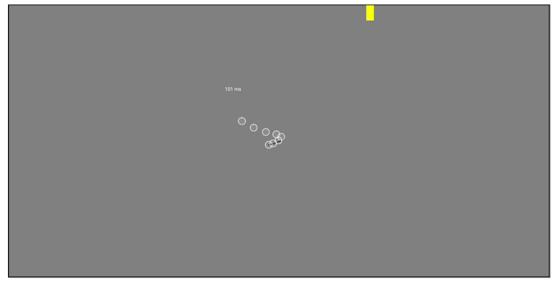
Once a user completes this trail, they perform similar trials for all the 24 MA-Edge targets in a block, with each target selected a total of 5 times, non-consecutively. Once a block is completed, a similar target selection trial happens for the remaining two blocks as per the experiment design order.

4.5 PROCEDURE

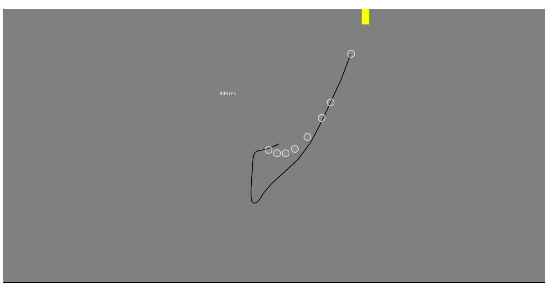
Firstly, participants were asked to complete a demographic questionnaire right after they arrived for the user study. This was followed by introducing them to the experiment setup and interaction technique. This included information on the three blocks of trials, each using a different activation area, the target selection task and the number of selections that they will have to perform for each block. For each block, all participants were required to practice target selections for the particular trial block until they were confident to start with the actual trials. Each participant had practiced with the same randomized target selection sequence for that particular block before they started the trials to avoid any difference in learning between participants. Since mid-air gesture interactions can be fatiguing



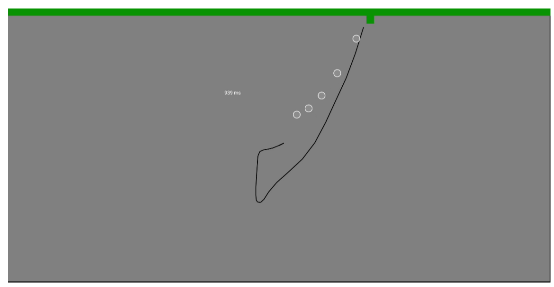
(a) Step 1: Pointer with trail for easy visibility, shown with the reference target



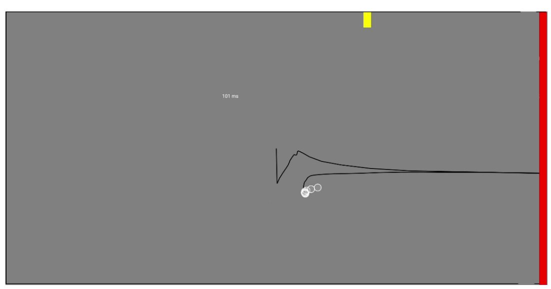
(b) Step 2: Reference target selected, marking start of a trail and also showing the real target to be selected



(c) Step 3: pointer movement towards the edge for target selection along with the trajectory of the movement



(d) Step 4: correct target selection indicated by green edge



(e) Alternate scenario: Incorrect target selection indicated by red edge

Figure 4.2: User study #1: Target selection task through application screenshots

when performed for a prolonged duration, the participants were informed that they could rest anytime and for any duration to avoid/minimize any hand fatigue. Further, the study conductor reminded participants to rest after every 15 trials, if required. Participants were informed and reminded to perform the selections as fast and accurately as possible before each block so that the target selection occurs at a nominal speed. After completing each block, the participants were presented with a NASA-TLX form to provide their ratings for that particular block of trials. At the end of all three blocks, a descriptive questionnaire was given to them to express their preferred block of trial. The questionnaire also requested other suggestions or comments about the interaction technique. Each participant also performed the Ishihara color blindness test at this stage.

4.6 RESULTS

In this section, we present the results of our experiment. We ran the experiment with 12 participants and obtained 5 trial data per target. Since there were 24 targets in each block, the total number of observations per block was 12 x 24 x 5 = 1440 and in 3 blocks was 1440 x 3 = 4320 trials.

After pre-processing the ratio scale data, as discussed below, we used repeated measures factorial ANOVA and followed it up with post-hoc tests. We use pairwise t-tests with Holm-Bonferroni correction to prevent Type I errors. Mauchly's test of sphericity was used to test homogeneity of variance and degrees of freedom were corrected when necessary. All violations of sphericity were corrected through Greenhouse-Geisser correction. The graphs for the mean values of the statistics also show 95% confidence intervals for parameter estimation.

For ordinal data, such as NASA-TLX and participant preferences, we performed the Friedman test and utilized Wilcoxon signed-rank tests for post-hoc pairwise comparisons when required.

The colorblindness results were negative for all participants.

4.6.1 *Data Preprocessing*

The raw data was preprocessed before performing any analysis on it. We checked the data for all kinds of errors at this stage and applied possible solutions to either fix them or remove them from the analysis.

The consumed endurance (CE) measurement unit at times did not correctly measure the CE values of a trial, due to Kinect device's posture detection issues occurring completely randomly in the course of the trial. A total of 112 trails were recorded incorrectly. All CE values lower than 0.1 were marked as erroneous values and were 107 in count, while a boxplot analysis of the remaining values showed 4 upper bound and 1 lower bound CE values as erroneous measurements. Since the missing CE data was of the type Missing Completely At Random (MCAR), MICE Imputation, a standard practice to fill missing data, was used to fill up those values [44] to prevent the loss of data.

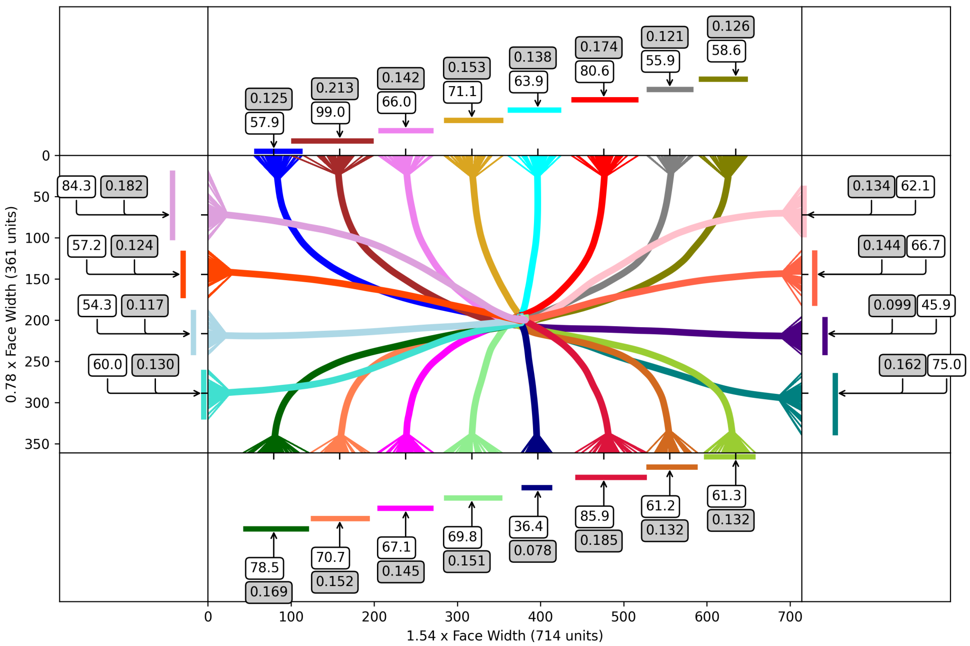
For each block, trials were marked as erroneous for file writing errors from time measurement, gesture recognizer errors, incorrect edge selection errors from participants, and trajectory errors where participants tried being very accurate by doing a lot of zig-zag movements before target selection. Block1, 2 and 3 had 12, 11 and 19 such trials, respectively, totalling 42 trials. These trials were removed from the analysis.

Even though the participants were instructed to make target selection at nominal speed, at times some still chose to be very accurate and attempted to select the point target accurately, thereby significantly increasing target selection time and impacting the purpose of the trial. A boxplot analysis of target selection time after the removal of the above errors showed some upper-bound outliers for time. Such data also needed to be removed from the analysis as participants were not given any performance feedback during the trial and were only instructed to perform as fast and as accurately as possible. With this another 43 trials were removed from analysis. In all, a total of 85 (1.97%) trials were marked as erroneous and the remaining 4235 trials were used for analysis.

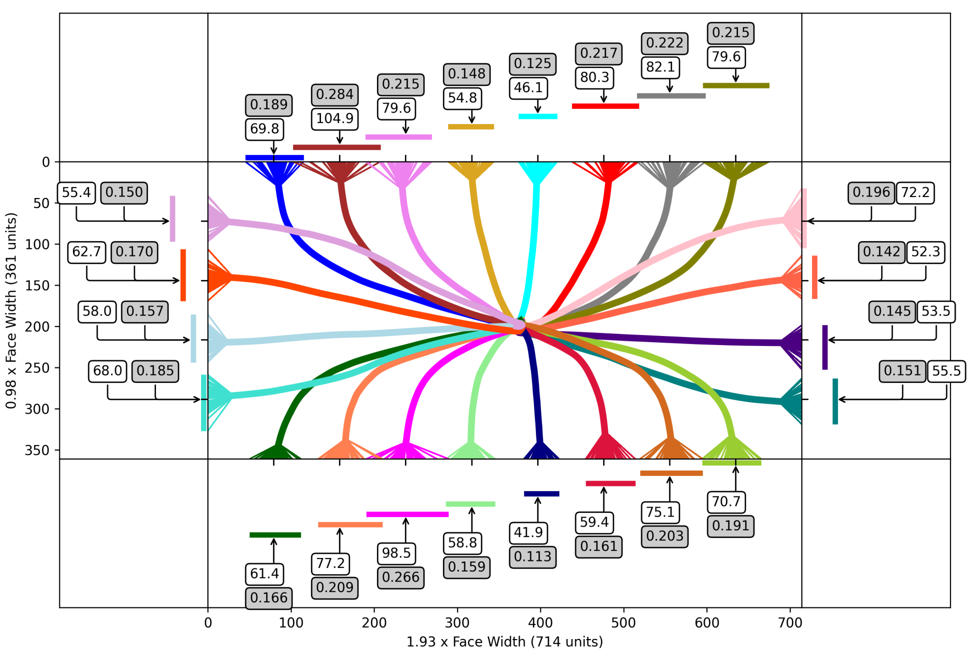
4.6.2 *Effective Target Width and Trajectory*

We next calculate the effective target width for each target of each block. With reference to the effective width calculation method [32], we select the middle 96% of the target intercept data. However, instead of symmetrically removing up to 4% values, we removed the farthest values from the median values as the data was not symmetrically distributed. The result of the process is visualized in Figures 4.3a, 4.3b and 4.4 and shown in both webview coordinates and the standardized facewidth coordinates.

The three figures also show Edge Bundling visualization in the central rectangular region that corresponds to the visible screen area of the interaction application. The edge bundles are the mean paths taken by all the participants from the reference target at the center to the actual targets along the four edges. A comparison of the Edge Bundling visualization for the three blocks is shown in Figure 4.12.

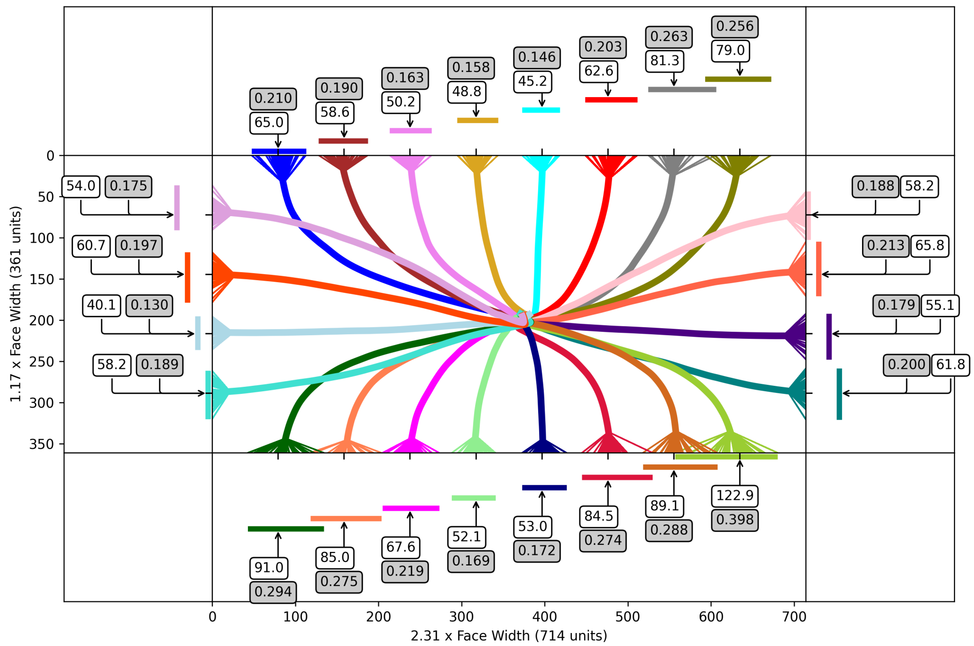


(a) Block1



(b) Block2

Figure 4.3: Edge Bundling visualization and effective target widths in facewidth units and webview units of the application



(c) Block3

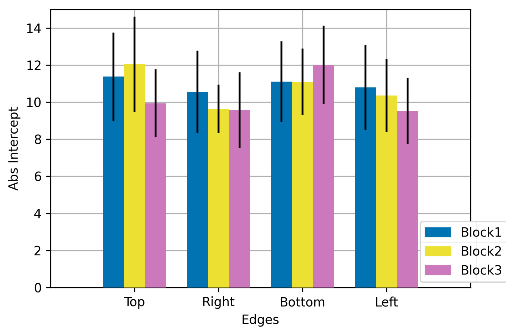
Figure 4.4: User study #1: Edge Bundling visualization and effective target widths in facewidth and webview units

It can be seen that the mean trajectory is not straight from the origin to the destination. This is because of the variety of pathways taken by the participants to select the targets. While some participants chose the shortest path trajectory straight to the targets (Figure 4.10), some others chose an L-shape trajectory (Figure 4.8) among other trajectory shapes. In the L-shaped trajectory, some participants used the metaphorical approach of 'point and shoot', where they would first use a straighter path in the horizontal direction with a relatively moderate speed to position the pointer right above/below the targets for the top and bottom edge targets and once they positioned it well, they would dart the pointer towards the target. This selection method can be identified where the trajectory is smooth, without any jaggedness.

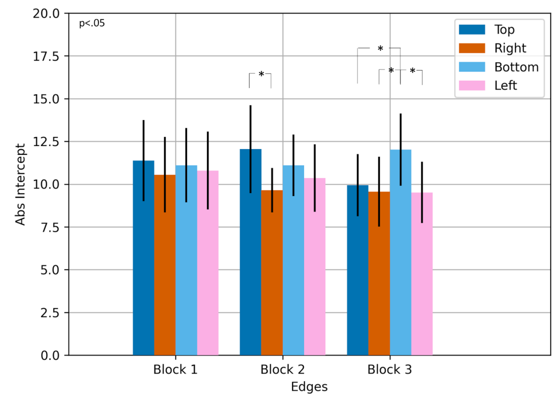
4.6.3 *Absolute Intercept*

While the intercept could be created on either side of the target, the absolute intercept is the absolute value for that intercept in webview coordinates irrespective of the side of the target on which it is created.

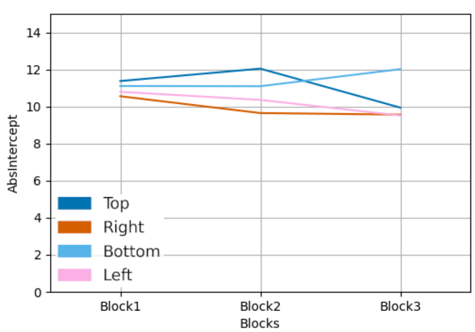
In this section, we show results from mean absolute intercept sizes when compared for each of the edges between the blocks, edges within each block and overall at the block level. Our results reveal main effect of edges ($F_{(3,33)}= 4.85$, $p<.05$, $\eta^2 = 0.0375$) with top and bottom edges forming larger intercepts than left and right edges and the right edge being significantly different than the bottom edge (Figure 4.5c). Block level comparison Figure 4.5d of absolute intercept shows that the mean absolute intercept size of Block3 was marginally smaller than the other two blocks but not statistically significant ($F_{(2,22)}= 0.64$, *n.s.*, $\eta^2 = 0.009$) signifying that overall there was not much difference in intercept sizes in webview absolute intercepts values between the three blocks. Edge level pairwise comparison shows significant differences between the bottom and all other edges of Block3, signifying that the bottom edge of Block3 formed significantly larger intercepts Figure 4.5b. Similarly, we found significant differences between the top and right edge of Block2 Figure 4.5b. We found no significant differences when each edge was compared with the corresponding edge of other blocks Figure 4.5a. The interaction effect observed visually in Figure 4.5c between the top and bottom edges of Block2 and Block3 was not found to be statistically significant.



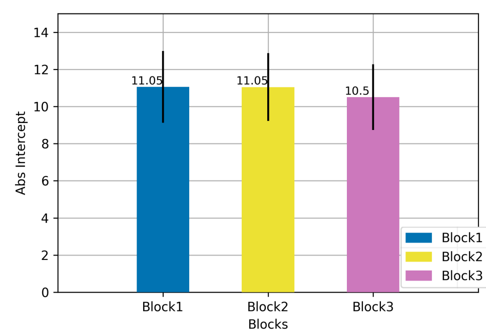
(a) Mean absolute intercept comparison of each of the four edges between blocks



(b) Mean absolute intercept comparison of the four edges within each block



(c) Main and interaction effects of blocks and edges on mean absolute intercept



(d) Mean absolute intercept comparison between blocks

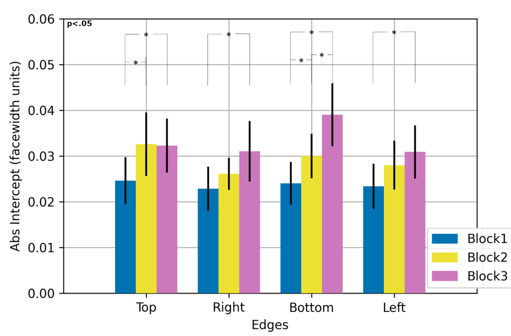
Figure 4.5: User study #1: Edge and Block level mean absolute intercepts with 95% confidence intervals

4.6.4 *Standardized Absolute Intercept*

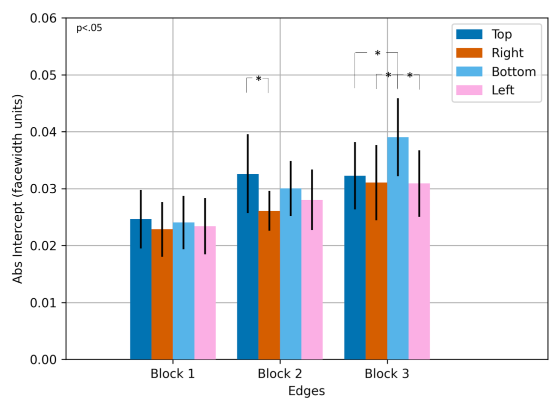
The earlier section compared intercept values through webview coordinates. These coordinates, when converted to interaction space, are termed here as standardized absolute intercepts. This is done by multiplying the webview coordinates with factors of (2/926), (2.5/926) and (3/926) for Block1, Block2 and Block 3, respectively.

In this section, we show results from mean standardized absolute intercept sizes when compared for each of the corresponding edges between the blocks, edges within each block and overall at the block level. Our results reveal main effect of edges ($F_{(1.7,18.79)}= 5.89$, $p<.05$, $\eta^2 = 0.04$). As can be seen from Figure 4.6c, top and bottom edges make larger intercepts than left and right edges. This difference was statistically significant between bottom and right; bottom and left; and top and right edges. We also see main effect of blocks ($F_{(2,22)}= 14.79$, $p<.05$, $\eta^2 = 0.18$) in Figure 4.6c. A pairwise contrast through emmeans (Figure 4.6d) of standardized absolute intercept showed significant differences between all three blocks, with Block3 forming the largest mean standardized absolute intercepts, followed by Block2 and Block1. The results also show an interaction effect between blocks and edges ($F_{(6,66)}= 3.03$, $p<.05$, $\eta^2 = 0.037$). Pairwise comparisons of corresponding edges between blocks show statistically significant differences between all corresponding edges of Block1 and Block3; the corresponding top and bottom edges of Block1 and Block2; and corresponding bottom edge of Block2 and Block 3 Figure 4.6a. Similarly, for within block comparison, significant differences were observed between the top and left edge of Block2; and between the bottom and all other edges of Block 3 Figure 4.6b. The interaction effect could be observed visually in Figure 4.6c between the bottom and top edges of Block3 and Block2.

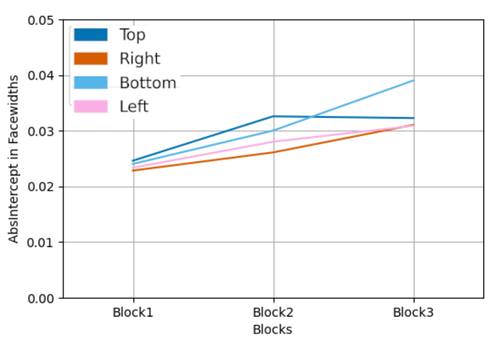
The results above imply that with the increasing size of the activation area of each block, the participants tended to form larger intercepts. This behaviour is reflected significantly at the block level and also for each of the edges of Block1 and Block3. Results from standardized absolute intercept data similar to absolute intercept results above also show that Block3 bottom edge formed significantly larger intercepts than the top, left and right edges of the same block and significantly larger intercepts for the same edge of the other two blocks.



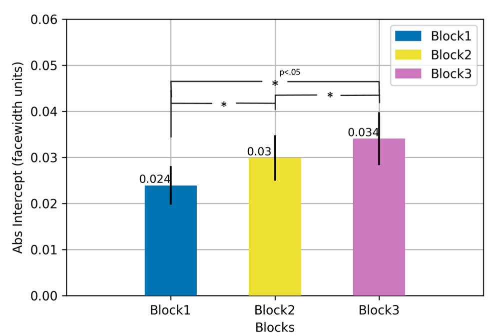
(a) Mean standardized absolute intercept comparison of each of the four edges between blocks



(b) Mean standardized absolute intercept comparison of the four edges within each block



(c) Main and interaction effects of blocks and edges on mean standardized absolute intercept



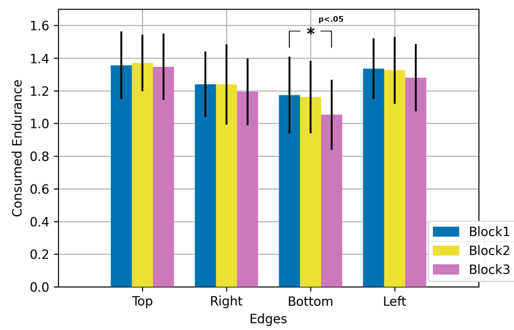
(d) Mean standardized absolute intercept comparison between blocks

Figure 4.6: User study #1: Edge and Block level mean standardized absolute intercepts with 95% confidence intervals

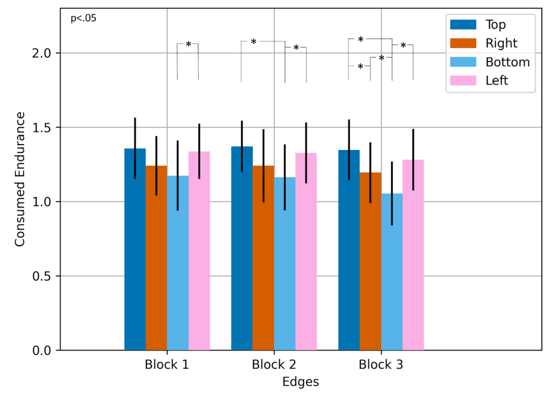
4.6.5 *Consumed Endurance*

In this section, we show results from consumed endurance measurement when compared for each of the edges between the blocks, edges within each block, overall at the block level and for each target between the blocks.

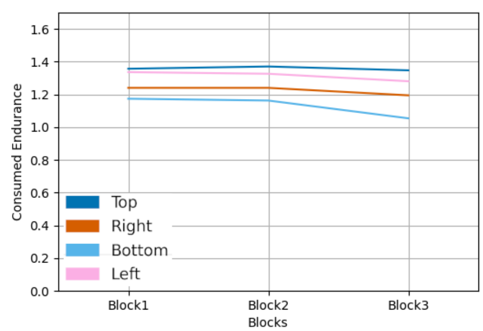
A factorial ANOVA analysis results reveal main effect of edges ($F_{(1.66,18.33)}$= *17.53*, *p<.05*, $\eta^2 = 0.0.07$), where top and left edges tended to utilize more consumed endurance than the bottom and right edges three blocks Figure 4.7c. This difference is statistically significant between top and bottom edges; top and right edges; and left and bottom edges. The results do not show that the main effect of blocks ($F_{(2,22)}$= *0.48, n.s.*) was significant and a block-level comparison (Figure 4.7d) shows a slightly lower mean consumed endurance value of Block3 compared to the other two blocks. This can be attributed to the many targets placed lower in the input space for Block3 compared to the other two blocks, and hence, requiring lesser effort. While there is a small interaction effect seen from Figure 4.7c, statistical analysis results also show no significant interaction ($F_{(6,66)}$= *1.23, n.s.*) effect between blocks and edges. However, pairwise t-test comparisons show significant differences amongst all edges of Block3; significant differences between the top and bottom, and bottom and left edge of Block2; and bottom and left edge of Block1 Figure 4.7b. Pairwise t-test comparison also shows significant differences between the bottom edge of Block1 and Block3 Figure 4.7a. Even though pairwise t-test comparison for top and bottom; top and right; left and bottom; and left and right were not significantly different for the remaining combinations, their p values were all under 0.06, i.e., bordering significance. Hence, it is safe to infer that the right and bottom edges tended to be lower in consumed endurance than the top and left edges for all three blocks.



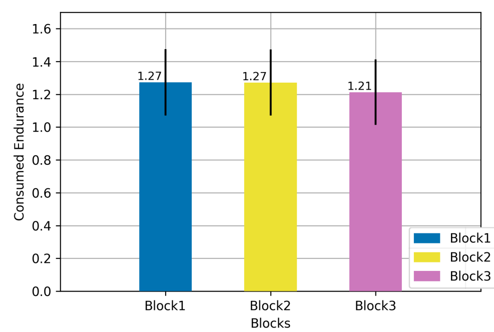
(a) Mean consumed endurance comparison of each of the four edges between blocks



(b) Mean consumed endurance comparison of the four edges within each block



(c) Main and interaction effects of blocks and edges on mean consumed endurance



(d) Mean consumed endurance comparison between blocks

Figure 4.7: User study #1: Edge and Block level mean consumed endurance values with 95% confidence intervals

Target level comparison results reveal main effect of targets ($F_{(23,253)}$= *20.75*, *p<.05*, $\eta^2 = 0.0.07$) with no interaction ($F_{(46,506)}$= *0.93*, *n.s.*) or block effect ($F_{(2,22)}$= *0.56*, *n.s.*). Pairwise t-test comparisons show significant differences amongst target numbers 13, 15 and 18 between Block1 and Block3 Figure 4.8. As can be seen from Figure 4.8, we can observe through the mean values shown with confidence intervals that a target at the shortest distance from the central reference target had the lowest CE values and the CE values increased with increasing distance from the reference target on the same edge. This behaviour can be seen for all the edges. The above-mentioned behaviour can be attributed to two reasons, (i) the central locations of the screen with respect to top and bottom targets required less upper arm extension compared to the corners; (ii) the central targets required less time (as shown later) for selection due to target distance, thus reducing the consumed endurance measurements.

Next, we check within edge behaviour for all blocks individually. For the top edge of Block1, when we compare target #1 with target #8, and similarly target #2 with target#7, and lastly #3 with target #6, we find that targets on the left side, on an average, had higher consumed endurance as compared to the targets on the right. The same behaviour can be seen for the remaining two blocks as well. We see the same behaviour for the bottom edge as well for all three blocks.

For the left and right edges, for all three blocks, we see that the higher the target position, the higher the consumed endurance values.

4.6.6 *Effective Throughput*

We first calculated the Index of Difficulty (ID) for each of the targets, with the effective width calculated in section 4.6.2, using the Fitts' Law formula [32] below:

$$ID_e = \log_2(\frac{D}{W_e} + 1) \tag{4.1}$$

where D is the distance from the center of the reference target to the midpoint of the effective target widths, W_e, calculated earlier. Using the effective Index of

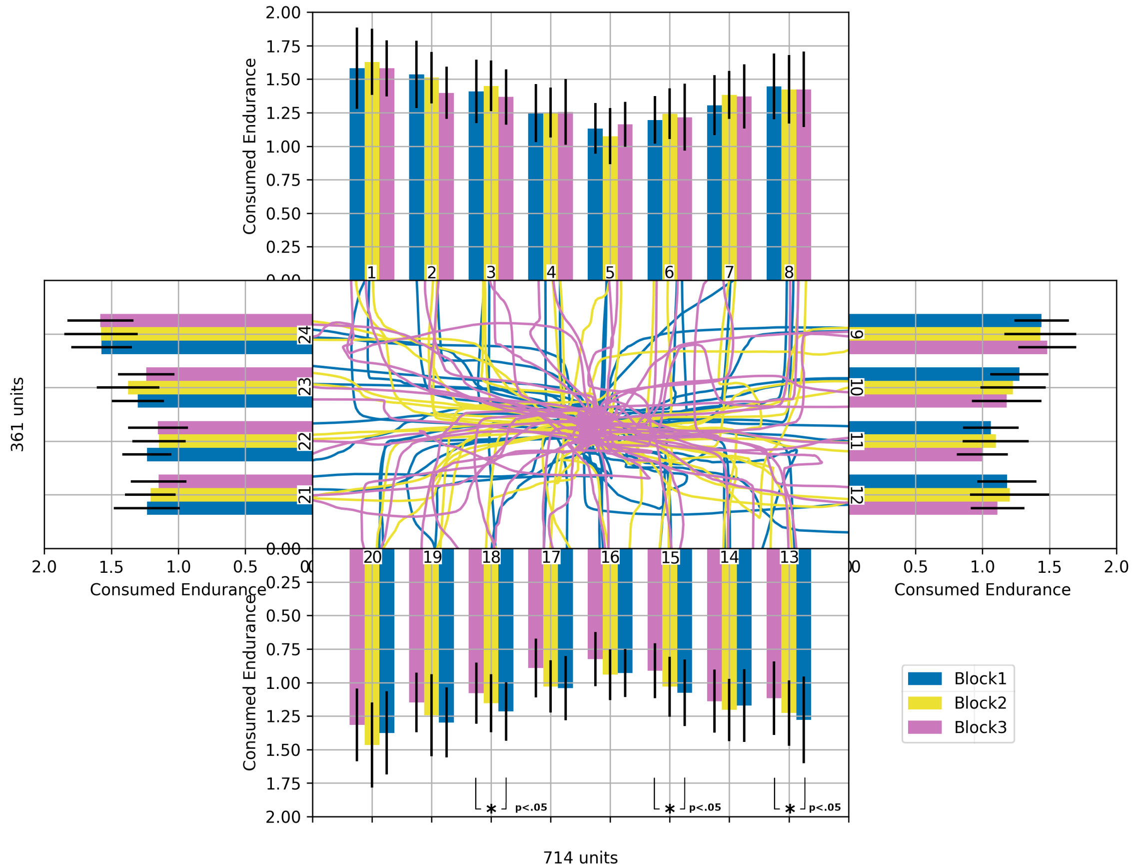


Figure 4.8: User study #1: Mean consumed endurance comparison of individual targets with 95% condidence intervals and L-shaped trial trajectories

Difficulty, effective throughput (TP_e) was calculated using the following equation for each of the targets.

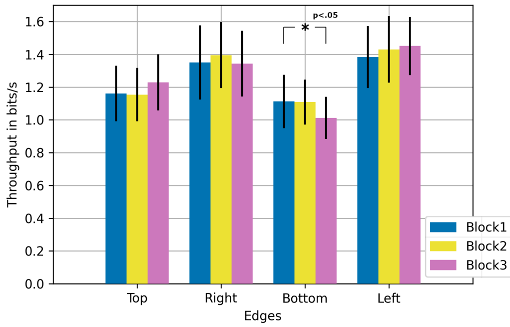
$$TP_e = \frac{ID_e}{MT} \tag{4.2}$$

where ID_e is the index of difficulty from the previous equation and MT is the mean movement time for each of the participants.

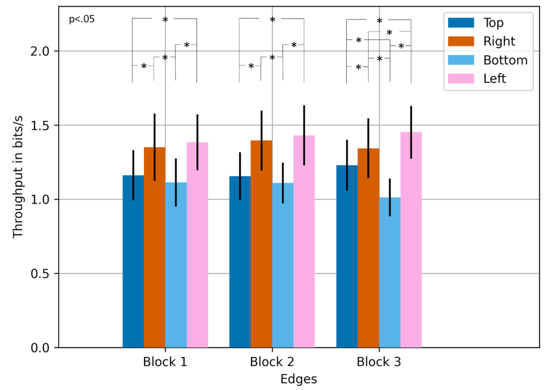
In this section, we show results from effective throughput measurement when compared for each of the edges between the blocks and within each block, at the block level and for each target between the blocks.

As can be seen in Figure 4.9c, a main effect of edges can be seen. The order of throughputs from highest to lowest is left, right, top and bottom as per the graph. A factorial ANOVA analysis confirms the same, i.e., main effect of edges ($F_{(3,33)}$= *59.74, p<.05*, $\eta^2 = 0.20$). Pairwise contrasts show a significant difference between top and left; top and right; bottom and left; bottom and right; and top and bottom edges. While no effect of blocks is seen as per the statistical analysis ($F_{(1.35,14.95)}$= *0.06, n.s.*), a block-level comparison (Figure 4.9d) showed Block2's mean effective throughput slightly larger than the other two blocks, without any statistically significant differences. We also find from Figure 4.9c an interaction effect between the edges and blocks. This interaction effect is also confirmed by statistical analysis ($F_{(6,66)}$= *4.83, p<.05*, $\eta^2 = 0.014$). Pairwise t-tests for each block show significant differences between the top edge and right edge; top edge and left edge; bottom edge and left edge; and bottom edge and right edge Figure 4.9b. However, no significant difference was found between the top and bottom edges; and also between left and right edges for Block1 and Block2. They also show significant differences between the bottom edge of Block3 and all the other edges of the same block. A significant difference was also observed between the bottom edge of Block1 and Block3 Figure 4.9a.

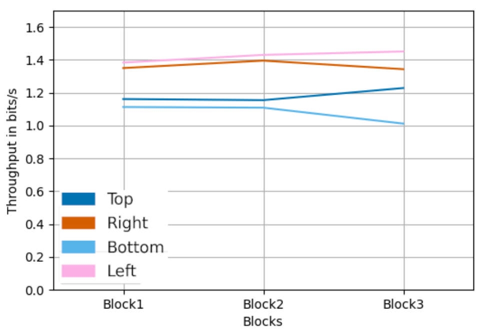
The results imply that the right and left edges for all blocks had significantly higher throughput than the top and bottom edges for all blocks. Also, the bottom edge of Block3 had the worst throughput in Block3, also it was worse than the bottom edge of Block1.



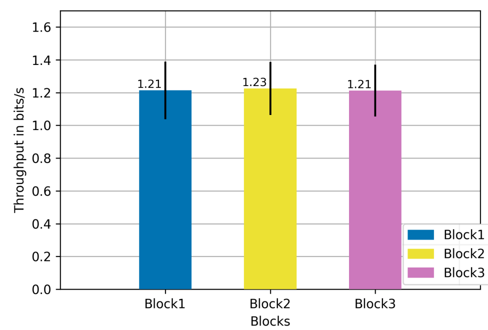
(a) Mean effective throughput comparison of each of the four edges between blocks



(b) Mean effective throughput comparison of the four edges within each block



(c) Impact of interaction effect and main effects of blocks and edges on effective throughput



(d) Mean effective throughput comparison between blocks

Figure 4.9: User study #1: Edge and Block level mean effective throughputs with 95% confidence intervals

Target level comparison results reveals main effect of targets ($F_{(23,253)}$= *35.22*, *p<.05*, $\eta^2 = 0.51$) and interaction effect ($F_{(46,506)}$= *9.28*, *p<.05*, $\eta^2 = 0.21$) between blocks and targets, while showing no main effect for blocks ($F_{(2,22)}$= *0.03*, *n.s.*). Pairwise t-test comparisons show some significant differences between the three blocks for some of the targets taken individually in top and bottom edges Figure4.10. As can be seen from Figure 4.10, we can observe through the mean values shown with 95 %confidence intervals that a target that required a relatively straighter movement along the horizontal or vertical direction, for instance, target #5, #11,

#16, and #22, had a higher effective throughput than the targets that were present at higher angles. The throughput was less for targets at the corners of the same edge as opposed to the central locations.

4.6.7 *Target Selection Time*

In this section, we show results from target selection time measurement when compared for each of the edges between the blocks, edges within each block, overall at the block level and for each target between the blocks.

While the Figure 4.11c shows a main effect of the edges, where the time taken by the right and left edges of all blocks is more than the top and bottom edges of the same blocks, the statistical analysis did not capture this effect. A factorial ANOVA analysis does not show significant main effect of edges ($F_{(3,33)}= 2.74$, $p<.05$, $\eta^2 = 0.07$), main effect of blocks ($F_{(2,22)}= 0.237$, *n.s.*) or a significant interaction effect ($F_{(6,66)}= 0.327$, *n.s.*) between edges and blocks . This could possibly be attributed to the reason that the selection time not only includes the time to move the pointer to the target to make the selection, but it also includes the time to identify the position of the target after the reference target is selected.

The edge level comparison between blocks for each edge is shown in Figure 4.11a, while the comparison within each block for each edge is shown in Figure 4.11b. The mean target selection time of Block2 was lower than Block1 and Block3, at 2069 ms, however, the results were not statistically significant (Figure 4.11d).

Target level target selection time comparison results reveal main effect of targets ($F_{(23,253)}= 20.42$, $p<.05$, $\eta^2 = 0.39$) with no interaction ($F_{(46,506)}= 0.76$, *n.s.*) or block effect ($F_{(2,22)}= 0.22$, *n.s.*). Pairwise t-test comparisons show no significant

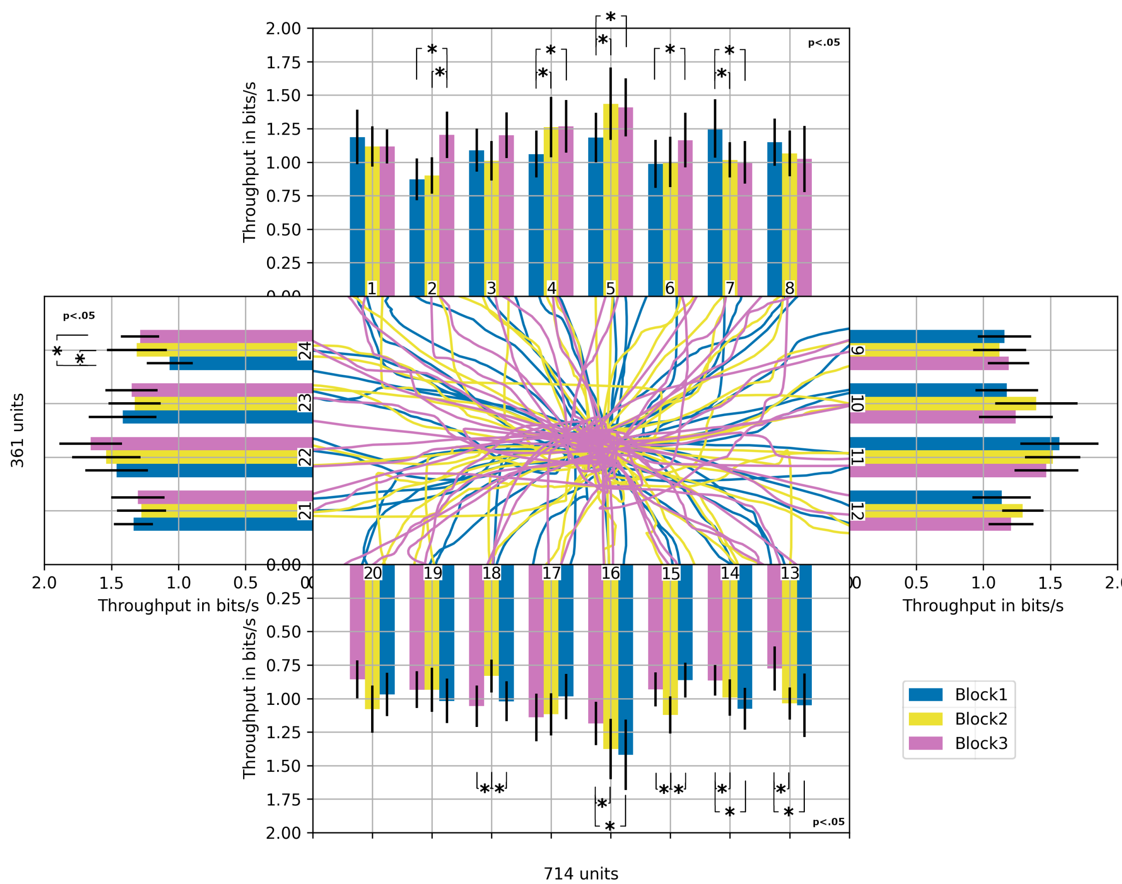
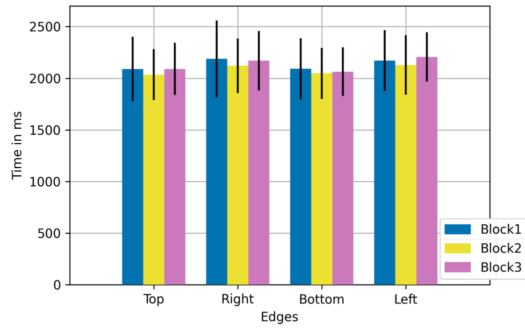
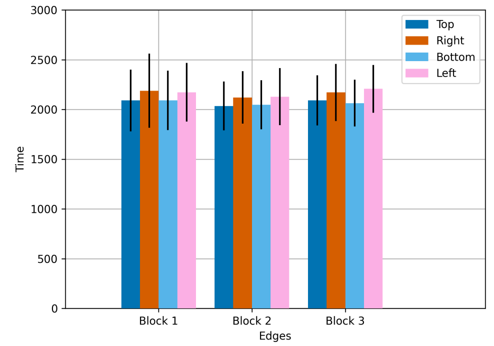


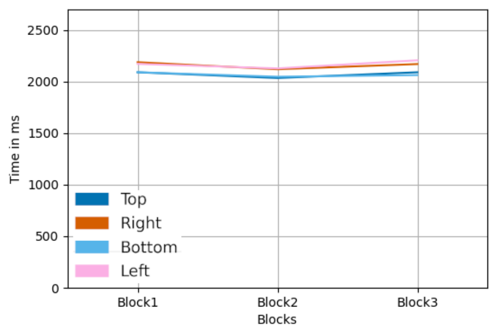
Figure 4.10: User study #1: Mean effective throughput comparison of individual targets with 95% confidence intervals and shortest-path trial trajectories



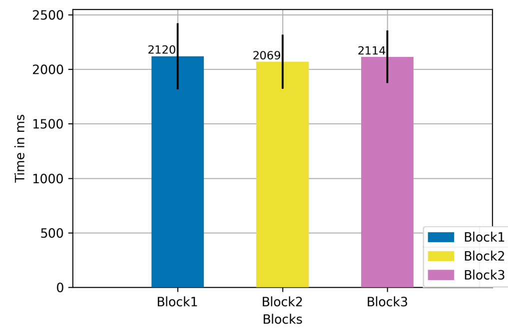
(a) Mean target selection time comparison of each of the four edges between blocks



(b) Mean target selection time comparison of the four edges within each block



(c) Impact of main effects of edges on time



(d) Mean target selection time comparison between blocks

Figure 4.11: User study #1: Edge and Block level mean target selection times with 95% confidence intervals

differences between the three blocks for each of the targets taken individually. However, from Figure 4.12, we can observe through the mean values shown with 95 %confidence intervals that a target at the shortest distance from the central reference target had the lowest target selection time values and the time values increased with increasing distance from the reference target on the same edge. This behaviour can be seen for all the edges of all the blocks. However, this behaviour may be attributed not just to the distance but the angle of the target and the effective target width as shown below.

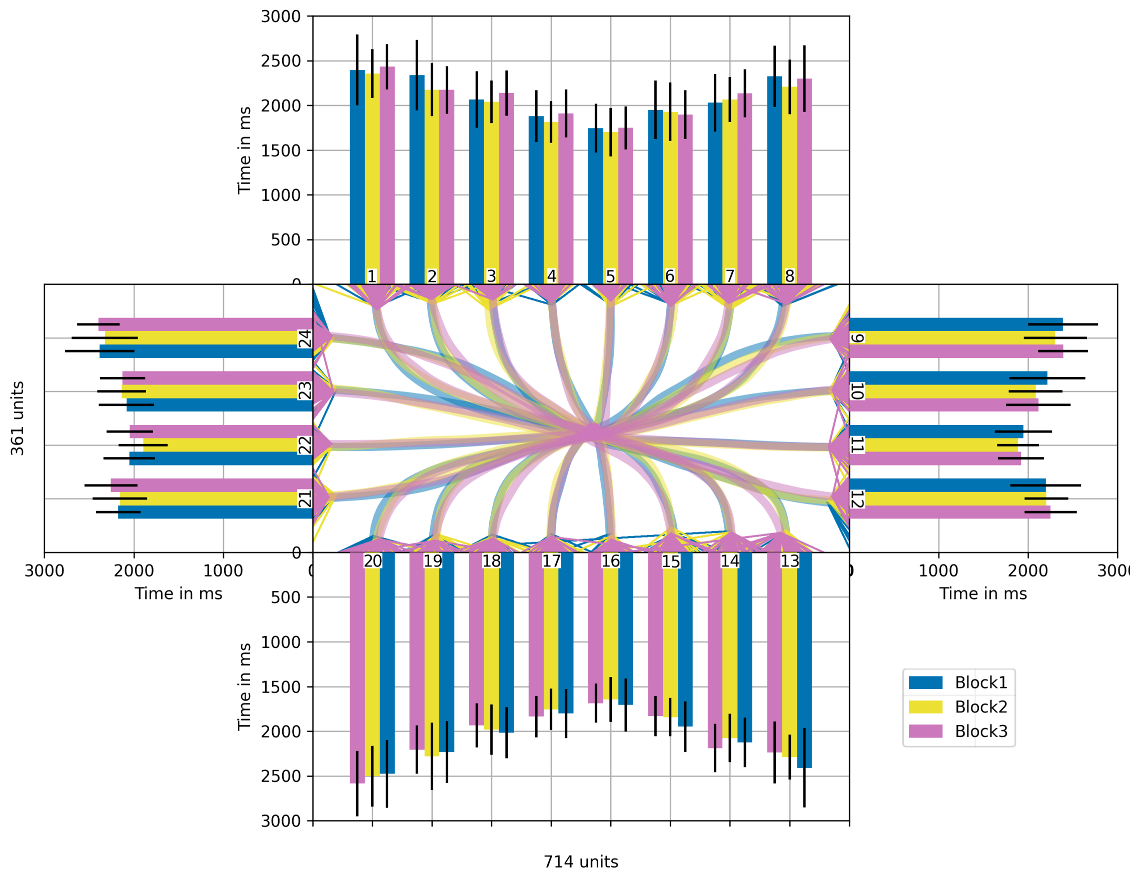


Figure 4.12: User study #1: Mean target selection time comparison of individual targets with 95% confidence intervals and Edge Bundling trajectory visualization for block comparison

4.6.8 *Linear Model fitting*

We fit a multiple linear regression model for each of the targets in a block with target selection time (MT) as the dependent variable and effective Index of Difficulty (ID_e) and angle of the target as predictor variables using Microsoft Excel. The angle of a target for the right and left edge is calculated as the angle made by the center of the effective target widths from the horizontal x-axis. Both up and down directions are marked as positive or negative, but not opposite. Similarly, for the top and bottom edges, the vertical y-axis is the 0-degree angle and all angles to the left and right of it are either both positive or negative.

We calculate a separate equation for each of the 4 edges of all three blocks. The results for the linear regression are shown in the tables Table 4.1, Table 4.2 and Table 4.3. It can be seen that there is a significant effect of angle for the top, right and bottom edges of Block3, while no significant effect of (ID_e) for this block. For Block2, for the same three edges, there is a significant effect of angle and (ID_e), both. And for Block1, there is a significant effect only for top and bottom edges for angle, while there is no effect for (ID_e). For all blocks, there is a stronger effect of angle as opposed to ID_e values implying that target selection time was impacted more by the angle of the target.

For all blocks, there is a strong linear relationship between the predictors and dependent variable time as the R^2_{adj} is a minimum of 0.85 for any edge of any block. The strongest relationship was observed for Block2 with all R^2_{adj} values over 0.95.

These results are similar to those reported by Apitz et. al. [3], where the angle of the target from the starting location was a strong predictor of the time of selection.

However, in our case inclusion of ID_e as a predictor improved the R^2_{adj} values as well.

Table 4.1: Block1: Multiple Linear Regression model fit with ID_e and angle as predictor variables for time

Edge	R^2_{adj}	Coefficients	Estimate	p value	Lower 95%	Upper 95%
Top	0.88	Intercept	1573.12	0.00	869.78	2276.47
		Angle	10.40	0.00	4.19	16.61
		ID_e	23.97	0.87	-347.27	395.21
Right	0.85	Intercept	2177.70	0.27	-10754.08	15109.49
		Angle	20.85	0.31	-120.98	162.69
		ID_e	-81.48	0.85	-4470.86	4307.90
Bottom	0.95	Intercept	1098.25	0.00	581.46	1615.03
		Angle	11.61	0.00	7.81	15.41
		ID_e	215.93	0.07	-33.05	464.91
Left	0.98	Intercept	3509.71	0.12	-5438.15	12457.57
		Angle	6.02	0.50	-70.75	82.79
		ID_e	-493.20	0.2734	-3363.72	2377.31

Table 4.2: Block2: Multiple Linear Regression model fit with ID_e and angle as predictor variables for time

Edge	R^2_{adj}	Coefficients	Estimate	p value	Lower 95%	Upper 95%
Top	0.98	Intercept	1202.23	0.00	871.32	1533.13
		Angle	10.30	0.00	8.57	12.02
		ID_e	165.28	0.03	22.67	307.89
Right	0.99	Intercept	739.72	0.24	-3028.79	4508.24
		Angle	32.45	0.04	5.00	59.90
		ID_e	361.39	0.16	-893.38	1616.18
Bottom	0.96	Intercept	886.08	0.00	406.95	1365.22
		Angle	12.96	0.00	9.27	16.64
		ID_e	267.87	0.03	34.40	501.35
Left	0.97	Intercept	1801.94	0.22	-6731.43	10335.31
		Angle	23.46	0.10	-27.52	74.45
		ID_e	23.12	0.93	-2981.21	3027.46

Table 4.3: Block3: Multiple Linear Regression model fit with ID_e and angle as predictor variables for time

Edge	R^2_{adj}	Coefficients	Estimate	p value	Lower 95%	Upper 95%
Top	0.93	Intercept	1038.15	0.02	217.30	1858.99
		Angle	9.84	0.00	6.53	13.15
		ID_e	250.69	0.11	-92.10	593.49
Right	0.99	Intercept	1806.45	0.19	-5304.37	8917.29
		Angle	29.39	0.06	-8.58	67.36
		ID_e	14.61	0.95	-2577.66	2606.88
Bottom	0.86	Intercept	549.71	0.35	-832.39	1931.82
		Angle	13.33	0.00	7.12	19.54
		ID_e	461.92	0.14	-222.55	1146.40
Left	0.99	Intercept	1729.86	0.06	-428.66	3888.40
		Angle	20.86	0.04	1.14	40.58
		ID_e	81.58	0.36	-585.09	748.27

4.6.9 *Target Fitting*

In this section, we measure the first level MA-Edge menu width for the three blocks. For this, we first remove all overlapping targets. This gives us the count of targets that fit each block as per their current varied sizes. However, most interfaces utilize uniform-sized targets. Hence, we determine the number of targets that fit with uniform sizes by picking the largest target from the remaining targets and placing it repeatedly on the edge to fill up all the empty space. This method of target counting prevents any performance degradation from the results as discussed earlier. It should be noted that with a slower selection rate, it would be possible to accommodate an even higher number of targets.

With reference to figures Figure 4.3a, Figure 4.3b, and Figure 4.4 and above-mentioned procedure, when we remove the overlapping targets, we can see that

Block1 can fit 22 targets (7 top edge + 4 right edge + 7 bottom edge + 4 left edge), Block2 can fit 20 targets (6 top edge + 4 right edge + 6 bottom edge + 4 left edge) and Block3 can fit 21 targets (7 top edge + 4 right edge + 6 bottom edge + 4 left edge). From this result, we see that at least 20 targets fit in all three blocks. However, with the existing arrangement it leaves a lot of space between the targets and a rearrangement can do a better utilization of edge real estate. We next calculate the number of uniform targets that can fit on each edge.

For getting fit of uniformly sized targets, we get the largest target of each edge of each block and see the whole number of targets that can be accommodated within the range of each edge. The largest targets for Block1, Block2 and Block3 are (#6, #12, #20, #24), (#6, #9, #19, #21) and (#8, #10, #13, #23), respectively. With these new target sizes, it can be seen that the three blocks Block1, Block2 and Block3, fit 25, 27 and 24 targets, respectively. The number of targets that fit each edge are as follows: (i) Block1 - 8 top edge, 4 right edge, 9 bottom edge, 4 left edge; (ii) Block2 - 8 top edge, 5 right edge, 9 bottom edge, 5 left edge; and (iii) Block3 - 9 top edge, 5 right edge, 5 bottom edge, 5 left edge.

From the above results, we can see that MA-Edge allows a large menu width at the first level with at least 24 targets that could fit with each activation area.

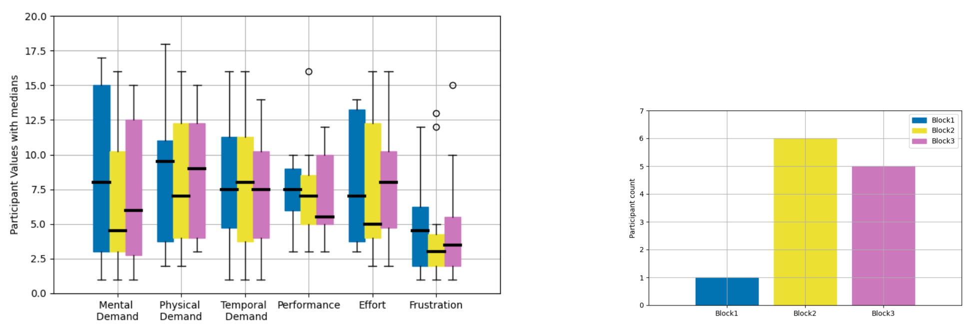
4.6.10 *Questionnaires and qualitative data*

We asked participants to rate the three blocks from 1 to 3 and also interviewed them for their opinions about the technique at the end of the user study. Most participants preferred Block2 or Block3 as their first choice, while Block1 was the least preferred $\chi^2_{(2,N=12)} = 4.67, n.s.$. The reason provided for the low preference of Block1 was the high sensitivity and low control, which in turn were due to the high CD-gain of this block from having the smallest input region from the three.

Next, we discuss the results from the NASA-TLX questionnaire. It is essential to note that the following discussion is only regarding the median performance values and no statistically significant results were observed with Friedman analysis, possibly due to the smaller size of the data. Blockwise NASA-TLX results are provided in the Figure 4.13a. The median mental demand for Block1 was the highest, $\chi^2_{(2,N=12)} = 0.4186, n.s.$, while Block2 and Block3 were lower in mental demand. For physical demand, Block2 median value was the lowest, while that of Block1 and Block3 were in the same region $\chi^2_{(2,N=12)} = 0.56, n.s.$. There is no difference observed in temporal demand between the three blocks $\chi^2_{(2,N=12)} = 3.25, n.s.$. The median performance for Block3 was the lowest, while that of Block1 and Block2 was about the same $\chi^2_{(2,N=12)} = 0.18, n.s.$. Regarding effort, the median value of Block2 are the lowest, while both Block1 and Block3 are higher $\chi^2_{(2,N=12)} = 0.18, n.s.$. Lowest median frustration was recorded with Block2, while higher median frustration was observed with Block1 and 3.

We observe a few outliers in the graph. An important result from Block3 outlier is that the participant who marked the highest frustration with this block also marked the physical demand and effort values as 8 and 10, respectively. In their interview, it was difficult for them to reach the bottom edge to make target selection as keeping the fist gesture at such a low height caused them pain in their hand. We did not find any other participant reporting this issue, although one other participant found the bottom edge selection of Block3 more demanding than the other two blocks.

Another outlier from Block2 shows a value of 16 for performance, but data from this participant shows a value of 10 for Block1 and 12 for Block2 for the same parameter. So, they tended to rate all blocks higher in general. However, they rated Block2 as the best performer. Analysis of another outlier from Block2 for frustration shows that the participant tended to use higher values of frustration for all blocks, i.e., 13, 12 and 10, for Block1, Block2 and Block3, respectively. In their reference, Block2 fared better than Block1 but worse than Block3. The last outlier for frustration shows one participant rating the three blocks, i.e., Block1, Block2, and Block at 8, 12, 7. The higher frustration could be attributed to the initial experiment setup issues, given they first performed trials from Block2. Hence, this outlier could also be disregarded.



(a) NASA-TLX

(b) Block Preference

Figure 4.13: User study #1: NASA-TLX and participants block preference

Some participants chose to use L-shape trajectories instead of shortest-path trajectories for target selection. We enquired about their reason for doing so after the completion of the study. One participant mentioned the shape of the target being a rectangle for doing so. They felt if the target was a semicircle, they probably would have chosen the shortest path trajectory. Another participant mentioned that they found the L-shape selection faster and more convenient than the shortest-path approach.

We asked participants for feedback on the technique and got the following data. One participant mentioned two reasons that slowed them a little, (i) they found the pointer trail to be distracting; (ii) the size of the target being very small. The trail distraction issue was concurred by two other participants. A third participant mentioned that the requirement of two motions for top and bottom targets was tougher than selecting targets on left and right edges which needed a single motion. A fourth participant mentioned that reaching the right edge of Block3 was more difficult due to the larger input region size. However, overall the participants tended to like the novel interaction technique.

4.7 DISCUSSION

The results from our study provide information on the design parameters that when taken into account while implementing MA-Edge interaction for remote displays can get an enhanced performance from the technique along with less effort. These include positioning the frequently used targets on edges taking lower effort and higher throughput, the number of targets that can be fit on each edge and block and selection of the appropriate activation region.

Our results show that the left and right edges always have a higher throughput than the top and the bottom edges for all three blocks which differ in their activation area sizes and positions. Similarly, targets that require a non-diagonal movement were easier to select and had higher throughput. We also showed that the effort to select a target increases when a target is placed higher up on the left and right edges or as the target sits wider on the top and bottom edges. Overall, we showed that the right and bottom edges could be chosen for low fatigue interaction as opposed to the top and left edges. We also showed that at least 24 equal-sized targets could easily be fit for all three blocks. It should be noted that the higher number of targets may be also placed with smaller target sizes, however, that will lead to lower performance and increased target selection times. Although not studied in this experiment, the performance of edges is expected to flip horizontally for left-handed users. For instance, with left-handed users, the left and bottom edges will take the least effort.

Our results do not capture any target selection time difference between the edges or between the blocks. One reason that can be attributed to this is that the real target was visible to the participant only after the reference target was selected. Hence, the total time included the time to find the target location as well. In future works, the target could be shown beforehand so that this time for locating the target is not part of the time measured. It should be noted that MA-Edge interactions will not always start from the middle of the screen, but our results provide a reference from the center for comparison as it is the most likely location for a user to start the interaction, given its strategic selection discussed earlier.

Regarding the choice of activation area, our results show that overall the participants preferred Block2 and Block3. However, some participants struggled to select targets on the bottom edge of Block3 due to the inability to maintain the fist gesture at lower angles. While most other participants did not report this issue, looking at the feedback for the bottom edge of Block3 from a few participants in combination with the results from standardized absolute intercept and absolute intercepts, which show that participants tended to form much larger intercepts on this edge, we can safely say that the Block3 may not be a good choice to be used with MA-Edge.

Similarly, Block1 received the least favourable feedback from the participants due to its high sensitivity. Out of 12 participants, only 1 preferred Block1. While on the other hand, Block2 was the most preferred, had a marginally higher throughput and the lowest mean target selection time than the other two blocks, had better sensitivity and control than Block 1 and was not reported for any adverse issues like the bottom edge of Block3. The block also fits the most number of equal-sized targets and the regression model also provides a better fit than the other two Blocks with significant effects from angle and ID_e, both. Due to the above reasons Block2, with an activation area side of 2.5 x facewidth is our preferred choice for MA-Edge.

This activation area can also be used for screens with other aspect ratios, such as the popular 16:9 ratio. This can be done by mapping the width of Block2 activation area, i.e., 1.93 facewidths, to the screen width. The placement of the top and bottom edges of the onscreen region can be adjusted as per the aspect ratio by keeping the onscreen area vertically symmetric within the action area. Even though the performance parameters will change a little with this activation area and aspect ratio combination, it is expected to remain close to our reported results. Further, it may be noted that we only studied three activation areas, which serves the purpose of the novel MA-Edge interaction; however, it may be possible to refine this input region location and dimensions even more with a larger study of more activation area variations.

Since the activation area is located at the fixed position about the user's body, MA-Edge could even allow a user to transition from novice mode to expert mode for the first level selection due to spatial consistency [15], although it may need further evaluation. The expert mode could be of two types. In the first type, the first-level targets on the edges of the screen may not be visible, and only the pointer would be seen, and the user could make a selection using their spatial memory for the onscreen target in this expert mode [52, 53]. This could even save some screen real estate from first-level menu removal. In an even higher expert-level mode, the user may not even be shown the pointer. The association of the activation area to the body will enable the users to use spatial memory to remember target locations and be able to select targets without there being any icons or text for reference with use [15].

In the next section, we explore the second-level of the multilevel MA-Edge interaction.

5

STUDY2: MA-EDGE VS POINT-AND-SELECT AT SECOND LEVEL

Now that we have identified the optimal MA-Edge design parameters for the first level of interaction, we next evaluate MA-Edge at the second level of interaction. This evaluation is done by comparing MA-Edge with the popular Point-and-Select method of target selection. For the same, we use MA-Edge with a linear menu (MA-Edge interaction configuration) and the Point-and-Select technique with both linear (1D-Point-and-Select interaction configuration) as well as grid menus (2D-Point-and-Select interaction configuration). We compare these interaction configurations in terms of performance through target selection time and error rate; and fatigue through consumed endurance, with the aim to determine if MA-Edge offers any advantages over the Point-and-Select technique.

5.1 APPARATUS

The apparatus (Android smartphone for interaction application and gesture recognition, gesture recognizer library, Kinect device for consumed endurance measurement, computer, projector for simulating large screen interaction by projecting application interface) used was the same as that used in the first study. However, we developed a new hybrid Android application for interaction in this study, as will be discussed later.

5.1.1 *Hand gestures*

While MA-Edge needs a single gesture for both navigation and selection, Point-and-Select needs two gestures, one for pointer navigation and another for making a target selection. To implement the Point-and-Select interaction configurations, we wanted to use a set of hand gestures that resemble that behaviour naturally. Therefore, we chose an open-hand palm gesture with the right hand to move the pointer and a fist gesture to select targets using the Point-and-Select technique. Further, to keep pointer navigation in the MA-Edge interaction configuration consistent with that of the Point-and-Select interaction configurations, we used open-hand palm gesture to control the pointer in MA-Edge as well.

5.1.2 *Activation area*

The new hybrid Android application we created contained the interface in the web part, while the native part included the same gesture recognizer module and the Android component of the application, as used in the first user study. From the results of the first user study, we chose the width of the region corresponding to the onscreen area in the activation area same as that of Block2, i.e., 1.93 facewidth and 0.98 facewidth. This is mapped to the onscreen coordinates (Figure 3.3) in the same way as that of study one, i.e., 714 x 361 units in webview coordinates.

Similarly, we also kept the off-screen width in the vertical direction for this study about the same as that of the first study, i.e., 50 units ((D) and (E) regions in Figure 3.3) in webview coordinates, giving a total of 1.25 facewidth units in the vertical direction for the activation area (A′ + D′ + E′ regions height in Figure 3.3). However, a modification was necessary to the off-screen width in the horizontal direction (B and C regions in Figure 3.3), as in this study, the interactions would utilize the off-screen areas as well at the second level. To ensure consistency on both edges, the horizontal offscreen areas had to be reduced from the earlier 100 webview units to 50 webview units in order to keep the horizontal and vertical offscreen areas of the same dimensions. Hence, the B and C regions to the left and right in (Figure 3.3) are now kept at 50 units each. This required a corresponding reduction in the offscreen width of the activation area, i.e., the regions B′ and C′ in (Figure 3.3), as well.

5.1.3 *Interface*

As mentioned earlier, we investigate two types of interaction techniques at the second level: 1) Point-and-Select target selection technique with linear and grid menus and 2) MA-Edge, giving us a total of three interaction configurations or test conditions. These interaction configurations are shown as three blocks in Figure 5.1, with each block comprising four menu configurations. Each block has the menus on two edges, the bottom and the right edge. The reason we used the right and bottom edges for the menu configurations is because they required lower effort than the other two edges. Another reason was that the two edges were categorized under high and low throughput categories, respectively, as per the first study results, covering all the categories of throughput variations for a comprehensive analysis. It may be noted that since MA-Edge menus offer the advantage of reducing the content occlusion in the central screen region by moving the interaction to the edge of the screen, in our comparison of MA-Edge and Point-and-Select interaction configurations, we positioned the two Point-and-Select menus along the edges of the screen as well. For each edge, we have two menu widths to compare: i) menus with 6 cells and ii) menus with 9 cells. The cell size of each of the menus of all three interaction configurations, i.e. 1D-Point-and-Select, 2D-Point-and-Select and MA-Edge, was kept the same, i.e., 40 webview units, chosen as per pilot experiments.

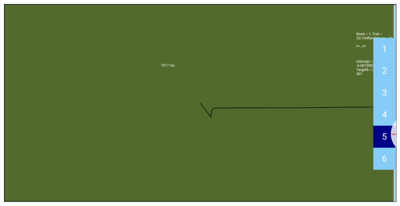
To distinguish between the pointer that is within the screen area and the pointer that is in the offscreen area, we modified some of the pointer parameters when compared to the first study. If the pointer is within the screen area, it appears as a small white circle with a trail of circles hollow at the center (Figure 5.2a). On the other hand, if the pointer is outside of the screen, only a part of it is visible while it changes its size and increasingly grows bigger as it moves farther away (Figures 5.1a and 5.1c) from any screen edge and reduces back gradually in size as it returns near the edge. Additionally, when the pointer is in the off-screen area, it is also overlaid with a crosshair. However, since only a part of the pointer is visible when it moves outside, we only see a part of the crosshair, shown as a red segment (Figure5.1c). This partly visible crosshair helps accurately identify the hotspot along the edge.

To enable the first-level menus, we placed the first-level targets at the shortest distance from the central reference target, i.e., the hand icon (Figure 5.2a), as they offered the fastest target selection time with the highest throughput as per the results of the first user study. The size of these targets is also chosen as per the results of the first study, which permits fitting 5 targets on the right edge and 9 targets on the bottom edge of the screen. Accordingly, we chose the targets located at the center of these edges for this study. To reduce the error rate in the selection of these first-level targets, we also included a trajectory of the pointer movement to provide the participant with feedback on the location of target selection along the edge of the screen.

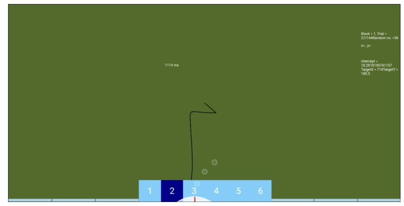
5.2 EXPERIMENT DESIGN

We chose a one-factor within-subject design for this experiment with three interaction configurations at the second level, i.e., 1D linear menu with Point-and-Select (1D-Point-and-Select), 2D grid menu with Point-and-Select (2D-Point-and-Select) and MA-Edge configuration, as the three test conditions. The order of these test conditions was counterbalanced using the Balanced Latin Square method. We employed a target selection task to compare performance and fatigue parameters between the three conditions. Each test condition used a block design where the order of sides, i.e., right and bottom edge, and menu widths, i.e., 6 and 9 cells, were completely randomized in trials performed by the participants. The blocks were named Block1, Block2 and Block3 for 1D-Point-and-Select, 2D-Point-and-Select and MA-Edge interaction configurations, respectively.

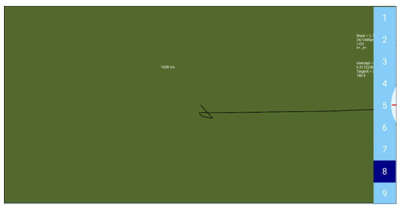
The target selection task, also called goal crossing [1, 30], performed in each trial, discussed later, required a participant to select targets which were displayed one at a time as individual cells of the menu. The order of these targets within a menu was completely randomized as well. Due to the randomization of target selection within each block, a 4x3 factor within-subject comparison can also be performed using



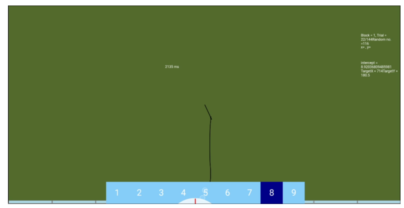
(a) Block1: 1D-Point-and-Select: 6-cell right



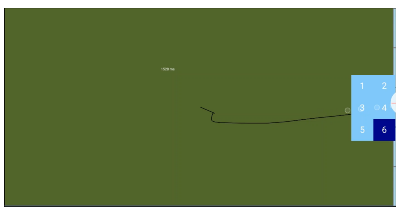
(b) Block1: 1D-Point-and-Select: 6-cell bottom



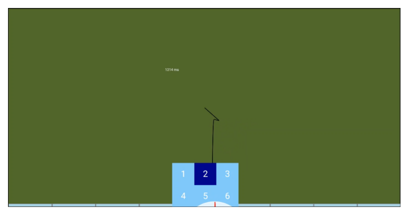
(c) Block1: 1D-Point-and-Select: 9-cell right



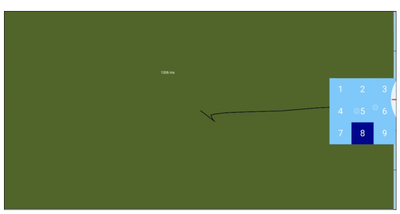
(d) Block1: 1D-Point-and-Select: 9-cell bottom



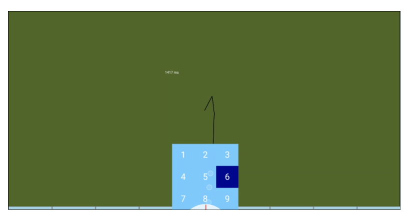
(e) Block2: 2D-Point-and-Select: 6-cell right



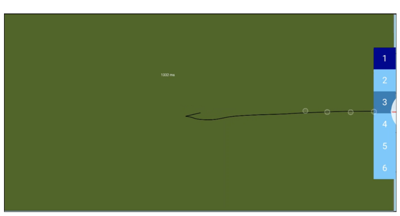
(f) Block2: 2D-Point-and-Select: 6-cell bottom



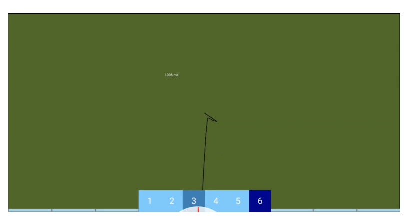
(g) Block2: 2D-Point-and-Select: 9-cell right



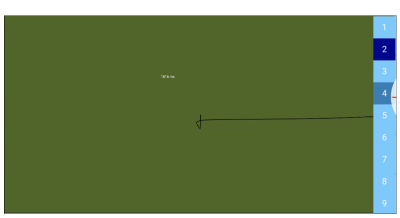
(h) Block2: 2D-Point-and-Select: 9-cell bottom



(i) Block3: MA-Edge: 6-cell right



(j) Block3: MA-Edge: 6-cell bottom



(k) Block3: MA-Edge: 9-cell right

(l) Block3: MA-Edge: 9-cell bottom

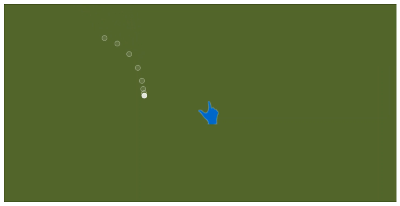
Figure 5.1: User study #2: Interaction configurations with their menu configurations

the menu configurations as another factor with the four test conditions, i.e., right edge 6-cell menu; bottom edge 6-cell menu; right edge 9-cell menu; and bottom edge 9-cell menu; along with the three interaction configurations discussed above, i.e., 1D-Point-and-Select, 2D-Point-and-Select, and MA-Edge, as the three test conditions. This allows us to compare the performances and consumed endurance not only between MA-Edge and Point-and-Select configurations, but also helps us to evaluate the impact of menu configurations, with differing locations and menu widths, within each interaction configuration.

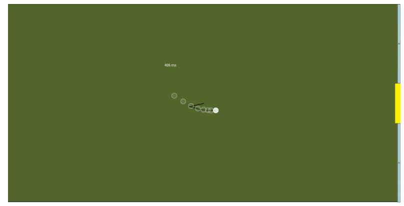
As a total of 4 configurations were available for each block Figure 5.1 and for each configuration, 36 target selections were performed by each participant, generating a total of 144 observations for each block per participant. Since there were 12 participants, we had 144 x 12, i.e., 1728 trial data for each block. The total data from the experiment for all three blocks was 1728 x 3 = 5184.

5.3 TASK

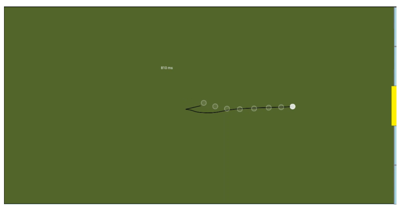
To initiate a trial, the user will first be required to select a reference target, i.e., the hand icon, displayed at the center of the screen (Figure 5.2a), by crossing it, i.e., the same as the previous study. As soon as the reference target is selected, a first-level target appears on one of the two edges, i.e., right or bottom edge, for selection through edge crossing (Figures 5.2b and 5.2g). At this moment, two separate timers will also start simultaneously to measure the total time taken by the user to select the final target at the second level and to measure the consumed endurance of the trail. The first-level target can only be selected if the user crosses the target from within its boundaries (Figure 5.2c). However, if it is crossed from anywhere else at



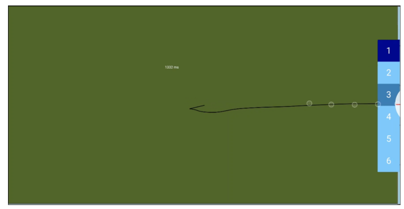
(a) Initial pointer position (any) to select the reference target (hand icon)



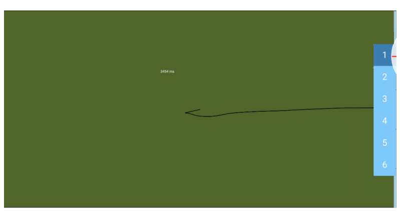
(b) Post reference target selection, first-level target (yellow bar) exposed and the trial begins



(c) Participant moves the pointer towards the first-level target to select it by crossing it (and the edge as a result) outwardly

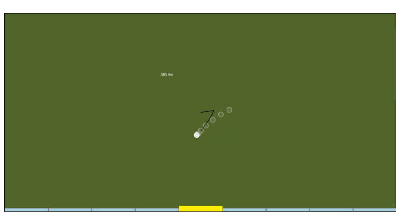


(d) Upon crossing the target a new menu appears with the second-level target in dark blue, current highlighted cell in blue based on the pointer position and the remaining cells in light blue

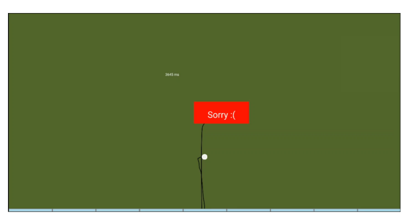


(e) Moving the pointer vertically in the offscreen region highlights the desired target for selection

(f) Crossing the edge inwardly selects the second-level target with feedback for correct selection



(g) Bottom location of the first-level target used in the study



(h) Feedback for incorrect selection of another second-level target located at bottom edge

Figure 5.2: User study #2: Target selection task through application screenshots

the edge, the target will not be selected. For the two Point-and-Select interaction configurations, if the user does not cross the edge through the target, they will have to return inside the screen boundary and make the selection again. This part was enabled primarily to verify the results from study 1 and check if incorrect selections stay within the reasonable error percentage range of the total selections made by the user. In contrast, for MA-Edge, the users had to re-select the reference target beginning from the first step.

A correct selection of the first-level target opens one of the menu configurations from the four possibilities based upon the block (Figure 5.1) corresponding to the interaction configuration currently under study. For instance, Figure 5.2d shows the MA-Edge second-level menu after the first-level target is selected on the right edge. For all three interaction configurations, the target meant for the next target selection appears in dark-blue color, whereas the other menu cells appear in light-blue color. As the user moves the pointer through the menu, while being in the offscreen region, the cell that is highlighted by the pointer changes its color to an intermediate shade of blue. Figure 5.2d shows all colors described above for the MA-Edge second-level menu.

At this step, the second-level target selection is dependent upon the interaction configuration currently under study. Let's look at each one of them. For MA-Edge (Figure 5.1i - 5.1l), once the second-level menu appears, the user pushes the pointer virtually against the outer offscreen boundary while sliding and scrolling to the relevant target to be selected. This allows them to slide and scroll through the menu options stably and highlight them (Figure 5.2e), without worrying about an inadvertent selection. Once the user reaches the intended target for selection, they bring the pointer back into the screen, i.e., by crossing the edge inwardly, to make the target selection (Figure 5.2f).

In the Point-and-Select interaction configurations, showed in Figures 5.1e - 5.1h for 2D menu and in Figure 5.1a - 5.1d for 1D menu, once the menu appears, the user needs to move the pointer back to the onscreen area so that they could first highlight the second-level menu cells (initially in dark blue and light blue). The highlighted cell will be turned to a blue color of intermediate shade. Once the correct cell is highlighted the user can then select the target by transitioning to the fist gesture from the current palm gesture. As a side note, since it is possible that while selecting the target or while transitioning to the fist gesture, a user might move their hand inadvertently and thus, cause the pointer to shift from the initial target pointing position, we deactivated the pointer movement as soon as fist gesture was detected in an attempt to increase the accuracy of target selection. However, while this deactivation did resolve the issue of pointer shift during selection to some extent, it did not resolve this issue completely. One drawback of this approach is that after the incorrect selection, once the movement is resumed, the pointer movement does not appear smooth and continuous and it potentially increases the total time of selection for subsequent attempts at the same target selection and therefore is not recommended for use.

In the end, once the second-level target is selected, the user is given visual feedback on whether their selection was correct or not. For correct second-level selections, “Perfect” text is displayed at the centre of the screen (Figure 5.2f) whereas for incorrect second-level selections, “Sorry” text is displayed (Figure 5.2h).

5.4 PARTICIPANTS AND PROCEDURE

Twelve (12) (8 males, 4 females) people participated in this experiment, with ages between 20 and 35 years old (M = 26.08, SD = 4). One participant did not report their age. All participants except two were right-handed. From the remaining two, one claimed ambidexterity and the other one claimed enough skill for interaction with their right hand but did not call themselves ambidextrous.

Firstly, the participants were asked to complete a demographic questionnaire right after they arrived for the user study. This was followed by introducing them to the set-up along with the three interaction configurations in the three blocks of trials along with their four menu configurations. All the participants were asked to do practice selections for each of the three blocks of trials until they were confident to start the actual blocks of trials. Each participant practised with the same randomized target sequence before the actual trials to reduce any difference in learning during the practice. We reminded the participants before each block to perform the selections as fast and accurately as possible. Since mid-air interactions performed continuously at length can be fatiguing, the participants were informed that they could rest anytime and for any duration to avoid any hand fatigue. Additionally, the study conductor reminded the participants to rest after every 15 trials.

After completing each block, the participants were presented with a short questionnaire with the NASA-TLX form and a few brief questions about the interaction configuration. After completion of all three blocks of trials, the participants were also presented with a questionnaire enquiring about their preferred interaction configuration on various parameters. At this stage, the participants were also administered the Ishihara Color Blindness Test.

5.5 RESULTS

In this section, we present the results of our experiment. After obtaining the raw data, it was first preprocessed for removal of invalid data, discussed next. After preprocessing the data, we performed statistical analyses to check the significance of the observation in accordance with the data type.

For ratio-scale data, we used repeated measures 3x4 factorial ANOVA and followed it up with post-hoc tests. To check the main effect of factors, the emmeans library was used in R. For comparing specific groups, we use pairwise t-tests with Holm-Bonferroni correction to prevent Type I errors. Mauchly's test of sphericity was used to test homogeneity of variance and degrees of freedom were corrected when necessary. All violations of sphericity were corrected through Greenhouse-Geisser correction.

For ordinal data like NASA-TLX, we performed the Friedman test and followed it up with Wilcoxon signed-rank post-hoc tests when the omnibus test was significant. To adjust for Type I errors, Holm-Bonferroni correction was used.

For categorical data, such as user preferences, due to the small sample size, we used xmulti omnibus test and followed it up with pairwise binomial post-hoc tests. To adjust for Type I errors, Holm-Bonferroni correction was used.

For error analysis, since a factorial comparison is needed on discrete error data, not fitting normality, we use the Aligned Rank Transform procedure and use emmeans and phia libraries in R for pairwise comparison of main effect and group effects, respectively.

The results are also shown with graphs for the mean values of the statistics along with their 95% confidence intervals for easy interpretation.

The colorblindness results were negative for all participants.

5.5.0.1 *Data preprocessing*

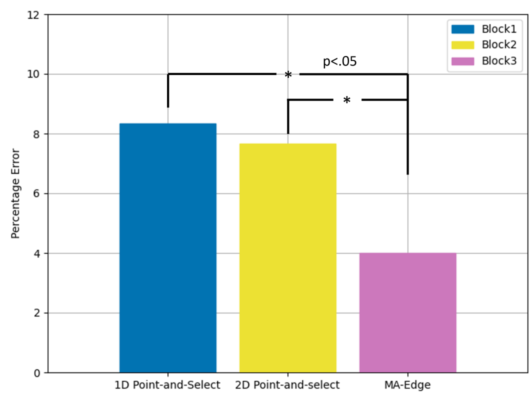
The raw data was preprocessed before performing any analysis on it. The data was checked for all kinds of possible errors at this stage to remove the erroneous trials from the analysis.

The errors which occurred completely at random were system errors and included file writing errors by the interaction application and consumed endurance application, gesture recognition errors, and application errors. After the removal of all these errors from a total of 1728 trial data from each block, Block1, Block2 and Block3 had 1713, 1709, and 1725, trial data, respectively, i.e., a total of 5147 trial data.

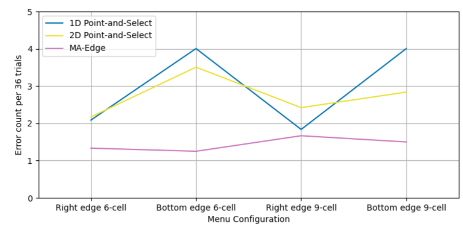
5.5.1 *Error Analysis*

In this section we show the target selection performance of users through error rate analysis at the first level and the second level of target selection.

As mentioned earlier, Block 1 and Block 2 were also configured to detect the incorrect first-level target selections using MA-Edge, i.e., if the user did not select the first-level target (Figures 5.2g and 5.2b) correctly, they could try to select it in multiple attempts. Out of the 1713 and 1709 trials in Block1 and Block2,



(a) Percentage of incorrect trials for each type of interaction configuration



(b) Main and interaction effects of interaction configurations and menu configurations on error rate

Figure 5.3: User study #2: Error analysis

respectively, the number of incorrect first-level target selections are found to be only 15 (0.8%) and 27(1.56%), respectively, with a mean error rate of 1.22%. This low error rate in target selection validates our results from the previous study on the first level, i.e., it ensures that the error rate for the first level of target selection is under 4 percent at nominal speed for the given target size. For further analysis, we remove this erroneous data from the total dataset, as multiple selections of the edge target would increase the total target selection time and consumed endurance affecting the analysis and results.

After removing the erroneous first-level target selections, we are left with 1698, 1682, and 1723 trial data for the three blocks, respectively, i.e., a total of 5103 trial data. Next, we analyze target selection error at the second level in these three blocks, i.e., after the first level of the target selection. The total number of trails that are marked as incorrect selections are 143, 131 and 69, i.e., (8.42 %, 7.80%, 4.00%), respectively, for blocks 1, 2 and 3 (Figure 5.3a). An Aligned Rank Transform (ART) 3x4 multi-factor analysis with the interaction configurations and menu configurations as factors show a significant main effect of the interaction configurations (Figure 5.3b). A statistically significant difference ($F_{(2,121)}$= *10.92*, *p<.05*, $\eta_p^2 = 0.15$) is observed between MA-Edge and 1D-Point-and-Select; and MA-Edge and 2D-Point-and-Select (Figure 5.3a). These results show that, for the given target size, i.e., the cell size of the menu, used for the study, MA-Edge for target selection is significantly less error-prone when compared to the Point-and-Select linear and grid configurations.

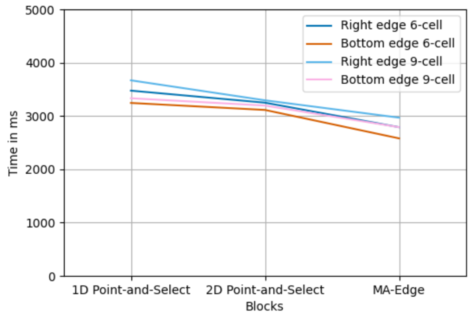
Referring to the (Figure 5.3b), we see a main effect of menu configurations as well for 1D-Point-and-Select and 2D-Point-and-Select configurations ($F_{(3,121)}$= *3.13*, *p<.05*, $\eta_p^2 = 0.07$). However, pairwise contrasts through emmeans does not show any significant differences between the four menu configurations. From the same figure, we see an interaction effect between the two factors as well ($F_{(6,121)}$= *1.99*, *p =.073.*, $\eta_p^2 = 0.08$). Results from the statistical analysis show the difference between 1D-Point-and-Select and MA-Edge in the bottom edge 6-cell menu configuration is significantly different from the right edge 9-cell menu configuration. Although from (Figure 5.3b), we can see a similar difference, not significant statistically, in error rates exists for the bottom edge 9-cell menu configurations as well, where 1D-Point-and-Select had a much higher error rate.

The results from Figure 5.3b signify that there is not much difference in the target selection error rate between any of the four menu configurations with MA-Edge and that both right and bottom edge will have a similar error rate with MA-Edge, irrespective of the 6-cell or 9-cell size. A slight increase in the mean error rate is seen for MA-Edge at 1.66 counts per 36 trials (CP36T) for the right edge 9-cell configuration in comparison to the other menu configurations, i.e., right edge 6-cell, bottom edge 6-cell, and bottom edge 9-cell, with mean error counts of 1.33, 1.25 and 1.5 CP36T, respectively. The reason for this is discussed further in section5.5.4 later.

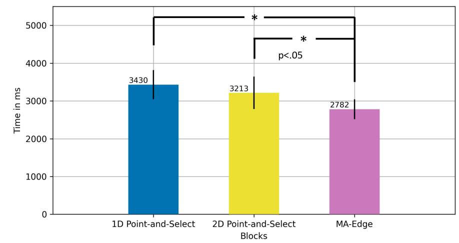
In contrast to MA-Edge, a difference between the error performances of the two edges exists for Point-and-Select interaction configurations. The Figure 5.3b shows that the error rate of 1D-Point-and-Select is higher for the bottom edge as opposed to the right edge. A similar difference in error rates between the right and bottom edge exists in 2D-Point-and-Select but is less pronounced. Finally, we also see that for the right edge, the difference in error rates between the three interaction configurations is lower as opposed to the bottom edge. From the above observations, we can see that it was relatively more error-prone to select second-level targets at the bottom edge with the Point-and-Select technique. We attribute this result to the observation that at lower positions it was relatively more difficult to hold the open-hand palm gesture for a prolonged time due to the strain caused in the hand and lower arm. Consequently, posture switching from open-hand palm gesture to fist gesture would be more difficult and error-prone. We revisit this in the section5.5.4 with qualitative analysis of data from participants. Even though the error rate is lower for right-edge menu configurations with Point-and-Select, it is still more than MA-Edge as per the Figure 5.3b.

5.5.2 *Target Selection Time*

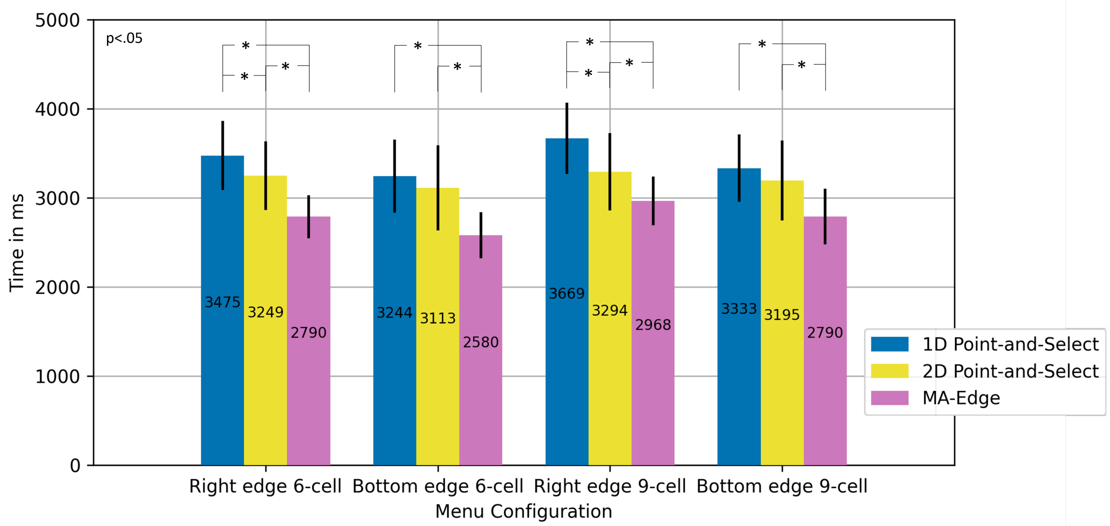
In this section, we show results from target selection time measurement. First, we report the mean target selection time comparison when compared at the block level, i.e., between the interaction configurations. Next, we also show the mean target selection time comparisons of corresponding menu configurations, i.e., right edge 6-cell, bottom edge 6-cell, right edge 9-cell, and bottom edge 9-cell, between each of the blocks and also when compared with each other within each block.



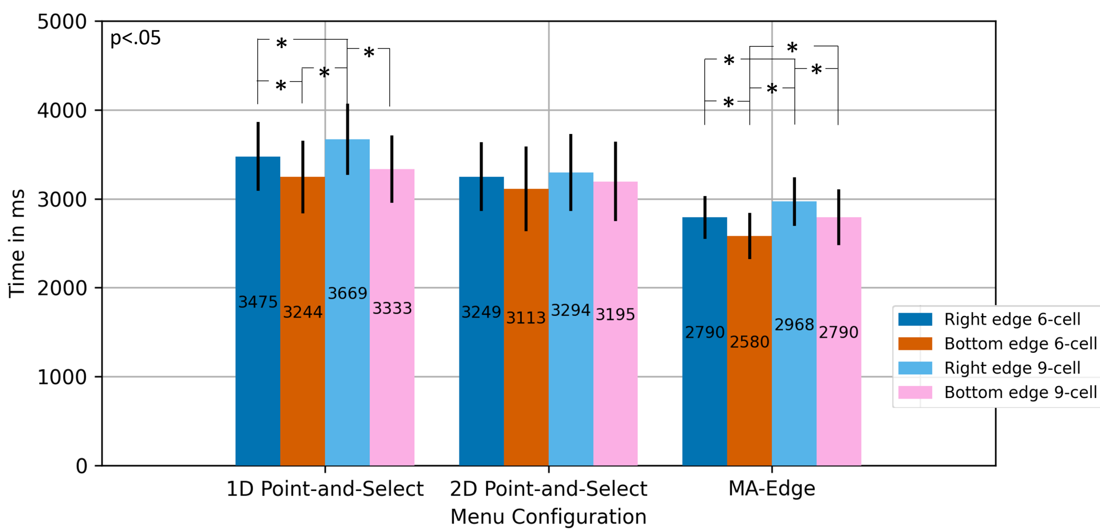
(a) Main and interaction effects of interaction configurations and menu configurations on mean selection time



(b) Mean selection time comparison between interaction configurations



(c) Mean selection time comparison for each corresponding menu configuration between the blocks



(d) Mean selection time comparison for each menu configuration within each block

Figure 5.4: User study #2: Mean selection times comparison with 95% confidence intervals

With reference to the Figure 5.4a, we can see that there is a main effect of the interaction configurations, with the target selection time decreasing as we move from 1D-Point-and-Select to 2D-Point-and-Select and to MA-Edge at the end. A 3x4 factorial repeated measures ANOVA analysis performed with interaction configurations and menu configurations as factors shows that the main effect of the interaction configurations is statistically significant ($F_{(2,22)}$= *19.77, p <.05.*, $\eta^2 = 0.18$) with a large effect size. In Figure 5.4b, we show results from pairwise contrasts of marginal means, using emmeans library in R. We find that MA-Edge was significantly faster than 1D-Point-and-Select and 2D-Point-and-Select at the overall block level.

The Figure 5.4a, also shows that there is the main effect of the menu configurations, with the right edge 9-cell menu configuration taking the most amount of time for all three interaction configurations, i.e., 1D-Point-and-Select, 2D-Point-and-Select and MA-Edge. This was followed by the right edge 6-cell menu configuration, then by the bottom edge 9-cell configuration, with the bottom edge 6-cell menu configuration being the fastest. The factorial ANOVA shows a small effect size for these differences ($F_{(1.56,17.18)}$= *22.41, p <.05.*, $\eta^2 = 0.04$), same as visually observed in Figure 5.4a. The pairwise contrasts of marginal means, using emmeans library in R, reports significant differences between all menu configurations with each other, except between the right 6-cell and bottom 9-cell menu configurations. We can see the same result from the graph (Figure 5.4a) as well, where we see some overlap between the two towards MA-Edge.

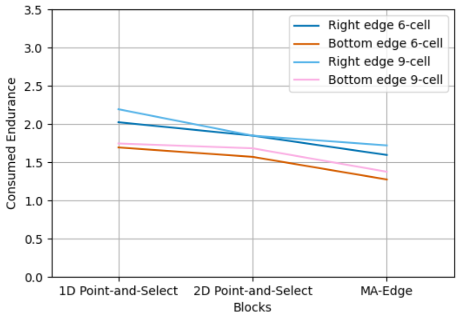
Next, we analyze the mean target selection times for each of the menu configurations and compare the corresponding menu configurations between the blocks (Figure 5.4c), i.e., interaction configurations. As can be seen from Figure 5.4a, there is a very small interaction effect between the three interaction configurations and the three menu configurations. This is also confirmed statistically ($F_{(6,66)} = 3.18$, $p < .05$., $\eta^2 = 0.007$). Pairwise t-tests for between block analysis (Figure 5.4c), show that for each menu configuration, i.e., right edge 6-cell, bottom edge 6-cell, right edge 9-cell and bottom edge 9-cell, there is a significant difference between 1D-Point-and-Select and MA-Edge; and 2D-Point-and-Select and MA-Edge; implying MA-Edge offers the fastest target selection rate when compared to 1D-Point-and-Select and 2D-Point-and-Select irrespective of the menu configuration. We also find that for right edge menu configurations of 6-cell and 9-cell, 2D-Point-and-Select offers significantly faster selection time than 1D-Point-and-Select. We do not find such a significant difference between menu configurations of the bottom edge. This could be attributed to the higher difficulty in transitioning quickly from open-hand palm gesture to fist gesture and maintaining cursor position simultaneously for accurate target selections at the bottom edge due to the slight strain felt in the hand and lower arm by participants for target selections at this edge.

We now analyze the mean target selection times for each menu configuration within a block (Figure 5.4d) and compare them to each other using pairwise t-tests. For 1D-Point-and-Select and MA-Edge, both with linear menu designs, we observe a significant difference in the mean selection time of right edge 6-cell and 9-cell menu configurations. One reason that can be attributed is that menus with larger widths (9-cell in our case) require a larger hand movement for selection as compared to the shorter menus (6-cell) on the same edge. The same occurs for the bottom edge as well. However, the difference is statistically significant for MA-Edge but not for 1D-Point-and-Select. For the same interaction configurations, i.e., 1D-Point-and-Select and MA-Edge, we also find the bottom edge to be faster than the right edge for the menu configurations with the same cell size. This behaviour was statistically significant as well for both cell sizes, i.e., 6 and 9. The possible reason for this appears to be the time taken for the first-level target selection, which is lower for the bottom edge as compared to the right edge due to the distance between the first-level targets. For the 2D-Point-and-Select menu configuration, the mean selection times of each of the menu configuration follows the same order from fast to slow as the other interaction configurations, i.e., bottom 6-cell fastest, followed by bottom 9-cell, then right 6-cell and then right 9-cell. However, the difference is not statistically significant. This implies that the 2D grid menu performs similarly in terms of target selection time, irrespective of the cell size, i.e., 6 or 9 cells, or the edge, i.e., right or bottom. The reason that could be attributed is the relatively shorter distance the pointer additionally needs to move for selecting targets in menus with larger widths in 2D grid menus as compared to linear menus.

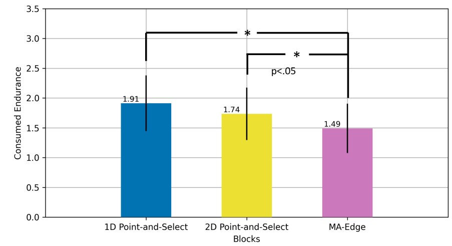
Overall, target selection times were faster in MA-Edge at the block level and also for all menu configurations in comparison to the Point-and-Select interaction configurations.

5.5.3 *Consumed Endurance*

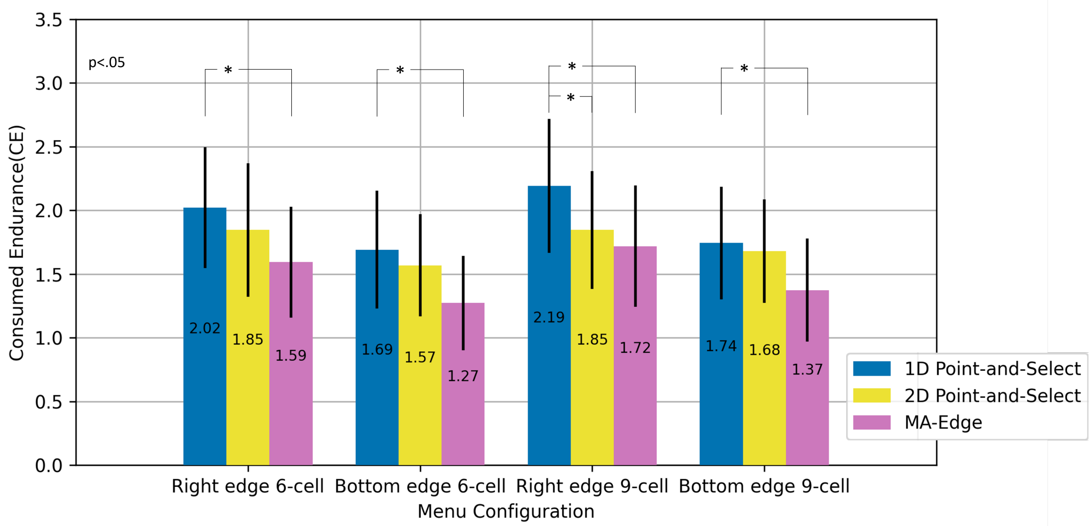
In this section we show results from consumed endurance (CE) measurement. First, we report mean CE comparison when compared at block level, i.e., between the interaction configurations, 1D-Point-and-Select, 2D-Point-and-Select and MA-Edge. Next, we also show the mean target selection CE value comparisons of corresponding menu configurations, i.e., right edge 6-cell, bottom edge 6-cell, right edge 9-cell, and bottom edge 9-cell, between each of the blocks and also when compared with each other within each block.



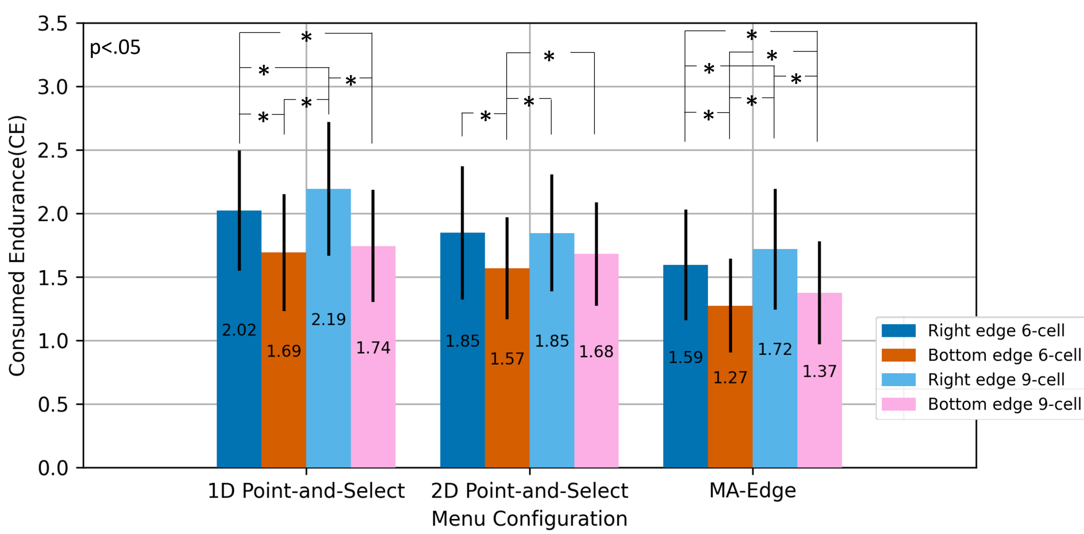
(a) Main and interaction effects of interaction configurations and menu configurations on mean consumed endurance (CE)



(b) Mean selection CE comparison between interaction configurations



(c) Mean selection CE comparison for each corresponding menu configuration between the blocks



(d) Mean selection CE comparison for each menu configuration within each block

Figure 5.5: User study #2: Mean selection consumed endurance comparisons with 95% confidence intervals

With reference to the Figure 5.5a, similar to mean selection times, we can see that there is a main effect of the interaction configurations, with the mean target selection CE values decreasing as we move from 1D-Point-and-Select to 2D-Point-and-Select and to MA-Edge at the end. A 3x4 factorial repeated measures ANOVA analysis performed with interaction configurations and menu configurations as factors show that the main effect of the interaction configurations is statistically significant ($F_{(2,22)}$= *9.69, p <.05.*, $\eta^2 = 0.06$) with medium effect size. In Figure 5.5b, we show results from pairwise contrasts of marginal means, using the emmeans library in R. We find that MA-Edge required significantly less consumed endurance than 1D-Point-and-Select and 2D-Point-and-Select at the overall block level.

The Figure 5.5a, also shows that there is a main effect of the menu configurations, with right edge 9-cell menu configuration being the most fatiguing for all three blocks, i.e., 1D-Point-and-Select, 2D-Point-and-Select and MA-Edge. This was followed by the right edge 6-cell menu configuration, then by the bottom edge 9-cell configuration, with the bottom edge 6-cell menu configuration being the fastest. The factorial ANOVA shows a medium effect size for these differences ($F_{(1.19,13.17)}$= *28.12, p <.05.*, $\eta^2 = 0.06$), same as visually observed in Figure 5.5a. The pairwise contrasts of marginal means, using the emmeans library in R, reports significant differences between all menu configurations with each other, except between the right 6-cell and right 9-cell; and bottom 6-cell and bottom 9-cell menu configurations. We can see the same result from the graph (Figure 5.5a) as well, where we see some overlap between the right edge 6-cell and right edge 9-cell lines. Also, the lines for the bottom edge 6-cell and bottom edge 9-cell are very close towards 1D-Point-and-Select interaction configuration. This result is expected as the bottom edge requires lower hand positions in comparison to the right edge menu configurations.

Next, we check the mean target selection CE values for each of the menu configurations and compare the corresponding menu configurations between the blocks, i.e., interaction configurations (Figure 5.5c). As can be seen from Figure 5.5a, there is a very small interaction effect between the three interaction configurations and the three menu configurations. This is also confirmed statistically ($F_{(2.96,32.60)}= 2.48$, $p <.05.$, $\eta^2 = 0.004$). Pairwise t-tests for between block analysis (Figure 5.5c), show that for each menu configuration, i.e., right edge 6-cell, bottom edge 6-cell, right edge 9-cell and bottom edge 9-cell, there is a significant difference between 1D-Point-and-Select and MA-Edge; implying MA-Edge causes the least amount of fatigue when compared to 1D-Point-and-Select irrespective of the menu configuration. We note that even though the mean CE values of each of the menu configurations are lower for MA-Edge as compared to 2D-Point-and-Select, there are no statistically significant differences between the two. We also find that for right edge menu configurations of 9-cell, 1D-Point-and-Select takes the most amount of effort and has statistically significant differences with 2D-Point-and-Select and MA-Edge.

We now analyze the mean target selection CE values for each menu configuration within a block (Figure 5.5d) using pairwise t-tests. For both linear menu configurations, i.e., 1D-Point-and-Select and MA-Edge, we observe that the mean CE values of the right edge 9-cell menu configuration was always the highest irrespective of the interaction configuration. This was followed by right edge 6-cell, bottom edge 9-cell and bottom edge 6-cell menu configurations. The following differences were statistically significant for these interaction configurations, right edge 6-cell and right edge 9-cell; right edge 6-cell and bottom edge 6-cell; right edge 9-cell and bottom edge 9-cell; right edge 9-cell and bottom edge 6-cell; and right edge 6-cell and bottom edge 9-cell menu configurations. In MA-Edge, the bottom edge 6-cell and 9-cell differences were also statistically significant. From these results, we can infer that the choice of edge and menu size will affect the amount of fatigue experienced by a user for a given interaction technique.

For 2D-Point-and-Select, the right edge 6-cell and right edge 9-cell menu configurations caused the same amount of fatigue. This could be attributed to the fact that both 6-cell and 9-cell menus were located at the same height and hence, their CE values are the same. The bottom edge menu configurations were always having the lowest CE values as compared to the right edge configurations. The bottom edge 9-cell configuration also had higher CE values than the 6-cell configuration as the bottom edge 9-cell configuration added another row of 3-cells over the 6-cell menu, increasing the height of the top row cells and thus, increasing the CE values also significantly.

5.5.4 *Questionnaires and qualitative results*

The participants were administered the NASA-TLX survey after completing trials from each block corresponding to the respective interaction configurations. The results from the survey are shown in the Figure 5.6. As can be seen, the median values of mental demand, physical demand and temporal demand, and effort were the least with the MA-Edge interaction configuration, with statistically significant differences observed for mental demand between MA-Edge and the other two interaction configurations. The median frustration and mental demand were the highest with 1D-Point-and-Select.

The participants were also administered another survey which asked them a number of questions that required to compare the three interaction configurations

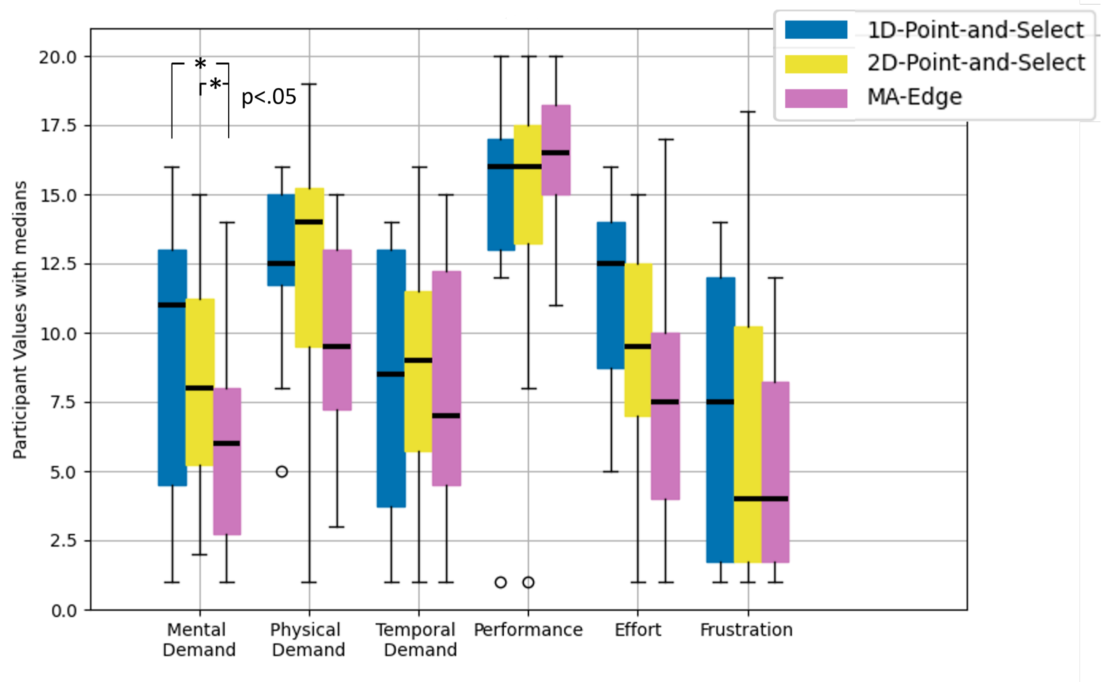


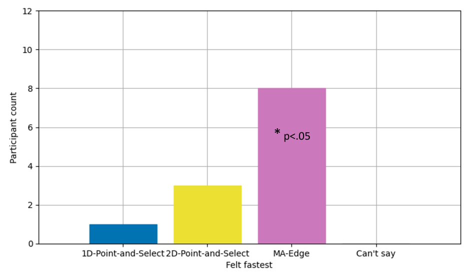
Figure 5.6: User study #2: NASA-TLX

with each other after all blocks were experienced by them, i.e., towards the end of the session. They were asked questions such as, which interaction configuration from the three appeared the fastest, slowest, most accurate, least accurate, most effort taking, least effort taking and best overall. The choices given to them included 1D-Point-and-Select, 2D-Point-and-Select, MA-Edge and Can't say. The results from these questions are analyzed using statistical analysis and it was found that a statistically significant number of participants found MA-Edge to be the fastest, most accurate, most comfortable, least effort-taking and best overall amongst the three interaction configurations. We also found that 1D-Point-and-Select felt the slowest for interaction, least comfortable and most effort-taking, however, the results were not statistically significant. The results from these questionnaires are shown in Figure 5.7.

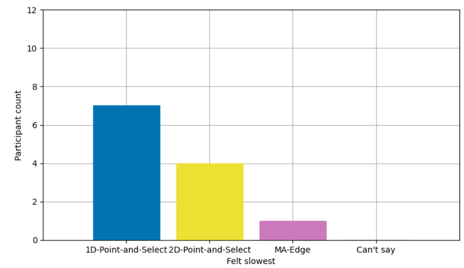
The qualitative survey from the participants helped us understand some aspects of the interaction configurations as well. We found that with MA-Edge, the error rate and time of selection with right edge 9-cell menu style was slightly higher than the other menu configuration with MA-Edge. The users expressed difficulty in selecting targets at the upper and lower extremes and also found them to be more error-prone as well. We find this to be an issue with the right edge but not the bottom edge because once the users reach the menu cells at the extremities a slight further movement along that direction would cause the pointer to go offscreen resulting in target selection errors or increased selection times.

One participant provided the reason for more difficulty, in their experience, in target selection with 1D-Point-and-Select and their preference for 2D-Point-and-Select between these two interaction configurations. "I would probably prefer the 2D menu if there was a gap between the grid of targets and the edge. The fact that they are more condensed around the area where you cross the edge is nice. However, targets that are right against the edge are more difficult to select as you have to be careful with the cursor coming back on screen. A gap would provide the ability to confidently come back on screen and then select without overshooting." As per this comment, we find that Point-and-Select can give better performance if the menus are placed at some distance from the edges. However, the drawback to this is that the occlusion of content below the menus, which MA-Edge aims to mitigate. Another way to resolve this issue could be by having better pointer graphics, which would allow a user to more easily perceive when the cursor is about to cross the edge inwards, which could preserve the Point-and-Select menu locations along the edges. It may also be possible that with longer usage of the technique, users will be able to learn to control the pointer better with offscreen navigation. Alternatively, the cell size of these menus may also be increased to address this concern. Finally, the performance can also be improved by improving the pointing time. This can be done by using the same strategy as MA-Edge linear menu to highlight the target, i.e., the linear menu would be highlighted even when the cursor is in the offscreen region and the user could then make a target selection without being required to bring the pointer over the menu cells.

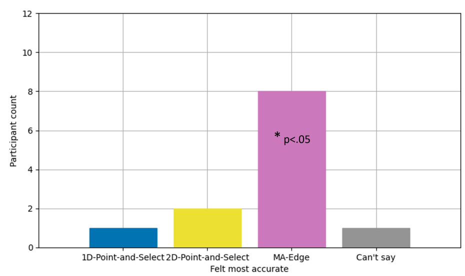
Our participants expressed some discomfort from the strain in the hand and forearm region from the open-hand palm interaction style chosen for this study when selecting targets at the bottom edge with Point-and-Select interaction configurations. We attribute this issue to the reason that open-hand palm gesture may be more straining at lower positions of the hand as opposed to the higher positions because for gesture detection to happen correctly, the palm plane must be roughly perpendicular to the camera view. Lower hand positions naturally bend the top of the palm towards the ground. Hence, to ensure detection and to keep the palm plane perpendicular to the ground, it caused more strain. Further, as we see Point-and-Select required longer interaction times for bottom edge menu configurations as opposed to MA-Edge and also required a user to hold the position while making a selection. All these factors appear to add to the strain experienced with this technique. These results concur with our identified challenges with some mid-air hand gesture-based interaction techniques, where prolonged usage of hand contorting gestures may not be feasible. Hence, simpler hand gestures are essential for mid-air interactions as we use in MA-Edge.



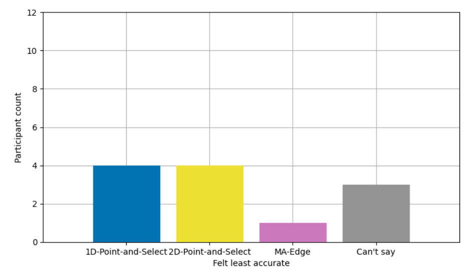
(a) Felt fastest



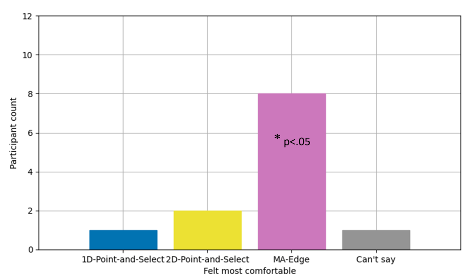
(b) Felt slowest



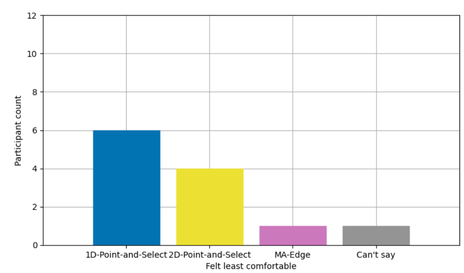
(c) Felt most accurate



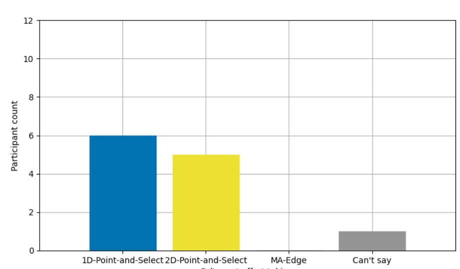
(d) Felt least accurate



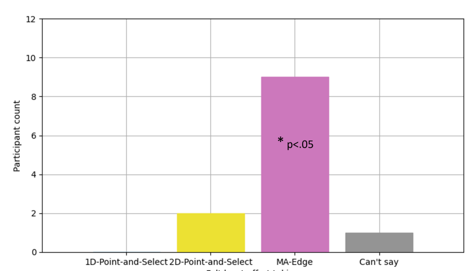
(e) Felt most comfortable



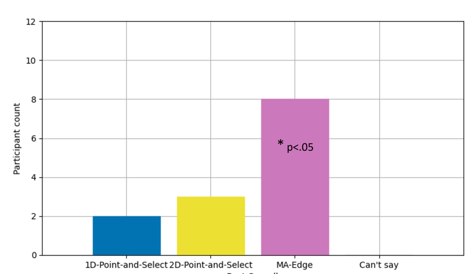
(f) Felt least comfortable



(g) Felt most effort taking



(h) Felt least effort taking



(i) Best overall

Figure 5.7: User study #2: Participant preferences

5.6 DISCUSSION

From Study 2 results, we can conclude that the MA-Edge interaction configuration has significant advantages over the other two Point-and-Select interaction configurations in terms of performance. MA-Edge offers faster target selection speed and lower error rates in target selection at the block level and for all four menu configurations individually as well. MA-Edge also demands lower fatigue in comparison to the Point-and-Select menu configurations. Our quantitative findings also match with survey results from the users regarding their preference for MA-Edge being their choice of interaction. Two-thirds of the participants found MA-Edge as the fastest, most comfortable, most accurate and the best overall.

Our results also show that a 6-cell MA-Edge menu performed better than a 9-cell MA-Edge menu for both bottom and right edges in terms of the time of selection and fatigue from interaction, while the bottom edge is faster and better for reducing fatigue in interaction as opposed to the right edge. A broadly similar pattern is also seen for the Point-and-Select interaction configurations as well, although with lesser difference in the intensities for the 2D grid menu.

One observation for MA-Edge was that the right edge 9-cell menu configuration performance was slightly lower as compared to the other configurations in terms of selection time, error rate and consumed endurance. However, this performance issue could possibly be fixed by either of the two following methods: i) By changing the 9-cell menu to a multi-level menu of 3x3 size, possible to achieve through MA-Edge or ii) by virtually restricting pointer movement beyond the screen boundary along the direction of the linear menu so that the extreme cells in the menu could be selected without much effort or time taken from the user. Although further work is required to check the efficacy of the proposed solutions; however, given that consecutive target selection in crossing-based menus performed fluidly with other input modalities [11], the multi-layer menu approach and the pointer restriction approaches look promising in fixing this minor performance issue.

Another important aspect of this experiment was the fatigue and hand/arm strain experienced by some users during target selection with the bottom edge menu configurations of Point-and-Select interaction. Since the bottom edge was straining to the users, the left and top edges could additionally be used in addition to the right edge, although further work is necessary to evaluate the performance of these two new edges. An alternate but non-natural behavioural approach may also be utilized to mitigate this issue. The approach could use fist gesture for pointer navigation to reduce hand strain and use open-hand palm gesture for making target selections, even for the bottom edge. However, more work needs to be done to test its efficacy.

With respect to fatigue with gesture interactions, an interesting observation from our results and observations is that despite the input region designed to be reached with lower fatigue by being reachable through the relatively relaxed bent arm posture (Figure 3.3), few users chose to use the extended arm configuration [16] leading to a larger increase in consumed endurance values for their measurements. In long-term usage of this technique, we believe the users will adjust their interaction technique to reduce the fatigue from their choice of arm posture.

As a result of comparative analysis, in this experiment we also showed that MA-Edge can be combined with Point-and-Select interaction techniques. We used first-level target selection with MA-Edge, while 1D-Point-and-Select and 2D-Point-and-Select were performed at the second level of interaction. Although we see lower performance of Point-and-Select methods in our results, a combination of MA-Edge and Point-and-Select can still be used. This could be done by increasing the cell size of menus to reduce the error rate and target selection time, and therefore, the fatigue as well. On these lines, in the next section, we see other possible interactions with MA-Edge alone and in combination with other interaction methods.

6

OTHER MA-EDGE INTERACTIONS AND MENU COMBINATIONS

In the second study, linear and grid menu designs using the Point-and-Select target selection technique were combined with MA-Edge. However, more such menu styles and interaction combinations are possible with MA-Edge. Therefore, in this section, we provide some sample interactions possible with MA-Edge alone and also some more interactions and/or menu styles that can be combined with MA-Edge to show the versatility of the interaction technique.

6.1 SLIDER MENU INTERACTION

Interaction with slider menus could be useful for various operations, such as volume control, channel navigation, zooming, and panning. Manipulation of multiple sliders has been explored before [48]. While some techniques allow manipulation of individual sliders through multiple semaphoric gestures [14] and specialized hand tracking systems from proximity [66, 67], MA-Edge can allow the manipulation of individual sliders from a distance and through the use of a single gesture. Figure 6.1 below demonstrates slider interaction with MA-Edge.

The first level target selection with MA-Edge, by crossing the right edge, will select a slider menu in the first step. In the next steps, the user can move their hand up and down to move the slider. The menu could then be closed by moving the hand back into the screen boundary.

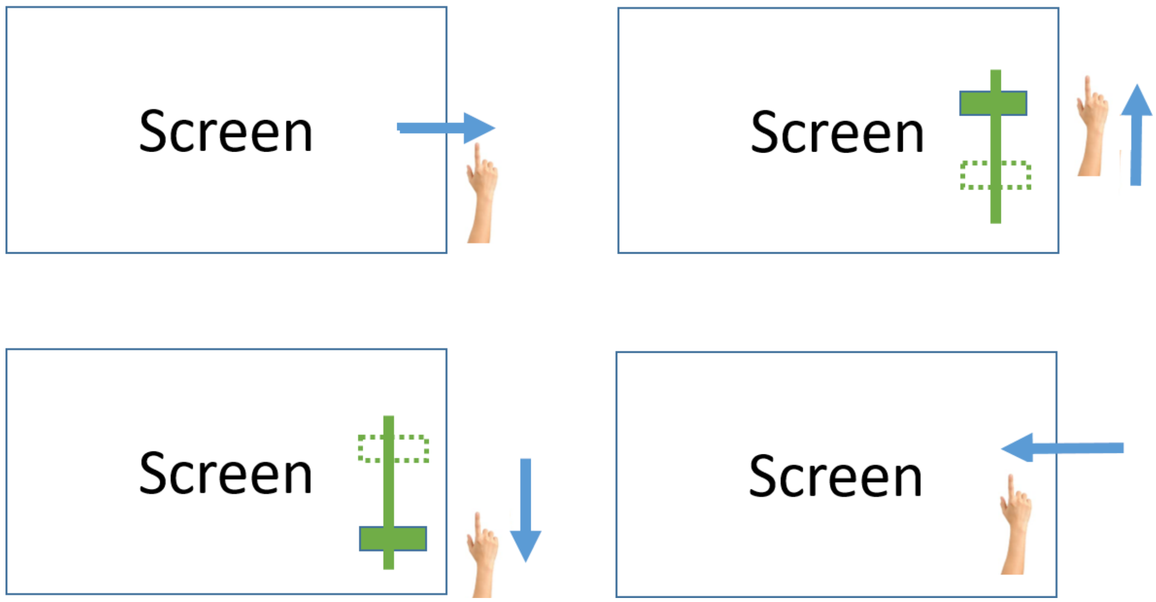


Figure 6.1: Slider menu interaction using MA-Edge

6.2 MARKING MENU COMBINATION WITH MA-EDGE

MA-Edge could also be combined with marking menus. Figure 6.2 below demonstrates marking menu interaction combined with MA-Edge. The first level target selection with MA-Edge, by crossing the right edge, will select a marking menu in the first step. In the next step, the user could move in any of the three directions through the edge. The angle of movement will select the marking menu option corresponding to the angle in which the pointer is moved back in.

Although we list design layouts of MA-Edge for continuous sliders and its combination with marking menus, it may well be contemplated that the MA-Edge interaction technique could be extended to other possible menu styles as well.

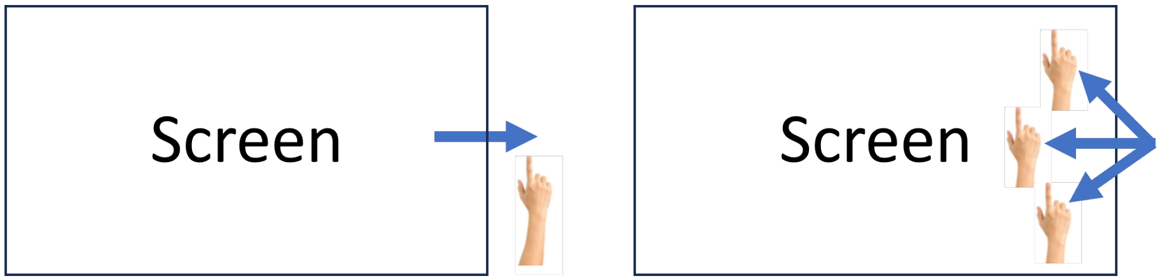


Figure 6.2: Marking menu and MA-Edge combination interaction

7

GENERAL DISCUSSION

In this research work, we have presented MA-Edge, which is a novel mid-air hand gesture-based interaction and target selection technique for large displays. From the results of the two studies, we show that MA-Edge offers considerable advantages over the other interaction techniques. One advantage is that this technique allows 24 targets to choose from at the first level of menu depth while allowing smooth multi-level interactions as well. Through linear menus, as used in this research, it may be possible to have MA-Edge menus with a depth of more than 2 levels. However, further work may be required to study the efficacy and issues associated with increasing the depth. This large menu width in combination with the multilevel interaction could be used to design new interfaces, such as MA-Edge-based input keyboards.

From the results of the second study, we also show that MA-Edge outperforms the Point-and-Select technique in terms of the time of selection and error rate of selection while showing that it is also less fatiguing. This performance advantage of MA-Edge was also reflected in the survey results from the participants. The amount of fatigue in a target selection trial is determined by the time duration of the interaction and also the location of the target as per our results. Since MA-Edge reduces the target selection time, it reduces the fatigue factor as a consequence.

Since MA-Edge alone can work with a single user gesture, it can allow significant savings on mid-air hand gesture development costs, efforts, and time. That is to say that by using just a single comfortable gesture, such as the fist gesture (the default resting hand posture), we can complete many types of interactions with a large display screen. It can be appreciated that although our studies were done with large, distant screens in this work, MA-Edge may be extendible to smaller screens, such as laptops, tablets, and even smartphones. However, more work on identifying the optimal design parameters for such smaller screens may be needed.

Additionally, the technique, by virtue of its design, allows users to interact with menus along the edges, i.e., away from the centre of the screen where the content is mainly displayed, thus minimizing content occlusion. As mentioned earlier, this could be very useful for multi-user viewer scenarios, for example, in multi-screen video games, where one user does not have to distract others for their interactions with the device.

Since the location of the activation area in MA-Edge is always connected to the user's body and is always at a fixed location with respect to the user's body, spatial memory could be utilized for MA-Edge interactions. Accordingly, MA-Edge menus could have expert modes for interaction, such as those in Marking menus, to improve target selection performance even further. However, more work is necessary to evaluate this.

As we have seen, MA-Edge offers the advantage of combining with other target selection techniques, such as Point-and-Select and Marking menus, and interactions other than just target selection, it is possible to make the user experience more efficient with MA-Edge. For instance, multilevel MA-Edge could utilize Marking menus at the second level of target selection instead of the linear menu, we used in the second study.

Through the host of advantages listed above, MA-Edge can allow an easy way to interact with smart public displays, where it may not be feasible to provide input devices to so many users. If many of the interactions could be done by MA-Edge, additional user-controlled input devices may not be necessary, thus making interacting with such displays extremely simple and convenient.

7.1 LIMITATIONS AND FUTURE WORK

As our studies were conducted mostly with the right-handed participants additional work is required to show that the findings will be mirrored for the left-handed participants. The optimal activation area for MA-Edge, found in the first study, assumes that the user is either standing or sitting up straight and is positioned in reference to those postures. However, for restful postures or for scenarios such as boardroom meetings where a user may be sitting behind a desk, the activation area location and dimensions will need to be readjusted. Further, the optimal activation area found in the first study could serve well for the screen aspect ratios chosen for the study and may as well extend to some aspect ratios, such as the popular 16:9 ratio; however, it may not be applicable to all aspect ratios, such as 9:16, which could occur when Smart Tvs are rotated by 90 degrees. For creating such a list of optimal activation areas for various screen aspect ratios, further work may be needed.

Our second study results show performance comparisons for the right and the bottom edges only between MA-Edge and Point-and-Select. Since these did not include results for the top and left edges, a user needs to exercise caution to extrapolate the results to these edges. Additionally, more work can be done to study the performance and fatigue parameters for the two edges.

7.2 CONCLUSION

In this work, we presented a new mid-air freehand gesture-based interaction technique called MA-Edge, which uses the screen edges to anchor targets and allows selecting them through edge crossing. We also reported the optimal design parameters for first-level interaction with MA-Edge, such as the activation area, the screen edges and target locations within those edges that are optimal from throughput, fatigue and time of selection perspectives. We also analyzed the results and found that a total of 24 equally-sized targets could fit the optimal activation area chosen for MA-Edge, which is much higher than Marking menus. We also showed that a linear relationship exists between the angle of the target for the first level with its effective Index of Difficulty and the target selection time. In the second study, we compared MA-Edge at the second level with the popular Point-and-Select technique and found through experimental study that MA-Edge outperforms Point-and-Select in terms of selection time, consumed endurance and selection error rate. These findings were echoed in the participant survey results as well, where most found MA-Edge to be the fastest, most comfortable, most accurate, least effort taking and the best overall. Finally, we showed that MA-Edge is a versatile design, and it could be extended to perform additional interactions other than just target selection, such as slider operations, and it could also be paired easily with some other menu styles, such as marking menu to create seamless multi-level interactions.

BIBLIOGRAPHY

[1] Johnny Accot and Shumin Zhai. "More than dotting the i's—foundations for crossing-based interfaces." In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. 2002, pp. 73–80.

[2] Georg Apitz and François Guimbretière. "CrossY: a crossing-based drawing application." In: *Proceedings of the 17th annual ACM symposium on User interface software and technology*. 2004, pp. 3–12.

[3] Georg Apitz, François Guimbretière, and Shumin Zhai. "Foundations for designing and evaluating user interfaces based on the crossing paradigm." In: *ACM Transactions on Computer-Human Interaction (TOCHI)* 17.2 (2008), pp. 1–42.

[4] Amartya Banerjee, Jesse Burstyn, Audrey Girouard, and Roel Vertegaal. "MultiPoint: Comparing laser and manual pointing as remote input in large display interactions." In: *International Journal of Human-Computer Studies* 70.10 (2012), pp. 690–702.

[5] Sebastian Boring, Dominikus Baur, Andreas Butz, Sean Gustafson, and Patrick Baudisch. "Touch projector: mobile interaction through video." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2010, pp. 2287–2296.

[6] Doug Bowman, Chadwick Wingrave, Joshua Campbell, and Vinh Ly. "Using pinch gloves (tm) for both natural and abstract interaction techniques in virtual environments." In: (2001).

[7] Andrew Bragdon, Eugene Nelson, Yang Li, and Ken Hinckley. "Experimental analysis of touch-screen gesture designs in mobile environments." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2011, pp. 403–412.

[8] Noshaba Cheema, Laura A Frey-Law, Kourosh Naderi, Jaakko Lehtinen, Philipp Slusallek, and Perttu Hämäläinen. "Predicting mid-air interaction movements and fatigue using deep reinforcement learning." In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 2020, pp. 1–13.

[9] Sean Cronin, Euan Freeman, and Gavin Doherty. "Investigating clutching interactions for touchless medical imaging systems." In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 2022, pp. 1–14.

[10] Terence Dickson, Rina R Wehbe, Fabrice Matulic, and Daniel Vogel. "HybridPointing for Touch: Switching Between Absolute and Relative Pointing on Large Touch Screens." In: *Proceedings of the ACM on Human-Computer Interaction* 5.ISS (2021), pp. 1–22.

[11] Morgan Dixon, François Guimbretière, and Nicholas Chen. "Optimal parameters for efficient crossing-based dialog boxes." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2008, pp. 1623–1632.

[12] Tafadzwa Joseph Dube, Yuan Ren, Hannah Limerick, I Scott Mackenzie, and Ahmed Sabbir Arif. "Push, Tap, Dwell, and Pinch: Evaluation of Four Mid-Air Selection Methods Augmented with Ultrasonic Haptic Feedback." In:

Proceedings of the ACM on Human-Computer Interaction 6.ISS (2022), pp. 207–225.

[13] Bruno Fruchard, Eric Lecolinet, and Olivier Chapuis. "Side-crossing menus: enabling large sets of gestures for small surfaces." In: *Proceedings of the ACM on Human-Computer Interaction* 4.ISS (2020), pp. 1–19.

[14] Aakar Gupta, Thomas Pietrzak, Cleon Yau, Nicolas Roussel, and Ravin Balakrishnan. "Summon and select: Rapid interaction with interface controls in mid-air." In: *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces*. 2017, pp. 52–61.

[15] Sean Gustafson, Daniel Bierwirth, and Patrick Baudisch. "Imaginary interfaces: spatial interaction with empty hands and without visual feedback." In: *Proceedings of the 23nd annual ACM symposium on User interface software and technology*. 2010, pp. 3–12.

[16] Juan David Hincapié-Ramos, Xiang Guo, Paymahn Moghadasian, and Pourang Irani. "Consumed endurance: a metric to quantify arm fatigue of mid-air interactions." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2014, pp. 1063–1072.

[17] Kenneth P Hinckley and Koji Yatani. *Multi-finger gestures*. US Patent 9,965,165. 2018.

[18] Tom Horak, Sriram Karthik Badam, Niklas Elmqvist, and Raimund Dachselt. "When david meets goliath: Combining smartwatches with a large vertical display for visual data exploration." In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 2018, pp. 1–13.

[19] Susu Huang, Qi Daqing, Yuan Jiabin, and Tu Huawei. "Review of studies on target acquisition in virtual reality based on the crossing paradigm." In: *Virtual Reality & Intelligent Hardware* 1.3 (2019), pp. 251–264.

[20] Mohit Jain and Ravin Balakrishnan. "User learning and performance with bezel menus." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2012, pp. 2221–2230.

[21] Sujin Jang, Wolfgang Stuerzlinger, Satyajit Ambike, and Karthik Ramani. "Modeling cumulative arm fatigue in mid-air interaction based on perceived exertion and kinetics of arm motion." In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 2017, pp. 3328–3339.

[22] Yvonne Jansen, Pierre Dragicevic, and Jean-Daniel Fekete. "Tangible remote controllers for wall-size displays." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2012, pp. 2865–2874.

[23] Maria Karam et al. "A taxonomy of gestures in human computer interactions." In: (2005).

[24] Yuki Kubo, Buntarou Shizuki, and Jiro Tanaka. "B2B-Swipe: Swipe gesture for rectangular smartwatches from a bezel to a bezel." In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 2016, pp. 3852–3856.

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

[26] Linlin Li, Zhiyi Fang, Weiqin Jia, and Fengjun Zhang. "Hierarchical Pointing on Distant Displays with Smart Devices." In: *International Journal of Human–Computer Interaction* (2022), pp. 1–16.

[27] Wenmin Li, Xueyi Wan, Yanwei Shi, Nailang Yao, Ci Wang, and Zaifeng Gao. "Depth and Breadth of Pie Menus for Mid-air Gesture Interaction." In: *International Journal of Human–Computer Interaction* 37.2 (2021), pp. 131–140.

[28] Shangchun Liao, Gongfa Li, Hao Wu, Du Jiang, Ying Liu, Juntong Yun, Yibo Liu, and Dalin Zhou. "Occlusion gesture recognition based on improved SSD." In: *Concurrency and Computation: Practice and Experience* 33.6 (2021), e6063.

[29] Mingyu Liu, Mathieu Nancel, and Daniel Vogel. "Gunslinger: Subtle arms-down mid-air interaction." In: *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. 2015, pp. 63–71.

[30] Yuexing Luo and Daniel Vogel. "Crossing-based selection with direct touch input." In: *Proceedings of the SIGCHI conference on human factors in computing systems*. 2014, pp. 2627–2636.

[31] Yuexing Luo and Daniel Vogel. "Pin-and-cross: A unimanual multitouch technique combining static touches with crossing selection." In: *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. 2015, pp. 323–332.

[32] I Scott MacKenzie. "Fitts' law as a research and design tool in human-computer interaction." In: *Human-computer interaction* 7.1 (1992), pp. 91–139.

[33] I Scott MacKenzie. "Human-computer interaction: An empirical research perspective." In: (2012), pp. 86–88.

[34] Ville Mäkelä, Jonas Winter, Jasmin Schwab, Michael Koch, and Florian Alt. "Pandemic Displays: Considering Hygiene on Public Touchscreens in the Post-Pandemic Era." In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 2022, pp. 1–12.

[35] Anders Markussen, Sebastian Boring, Mikkel R Jakobsen, and Kasper Hornbæk. "Off-Limits: Interacting Beyond the Boundaries of Large Displays." In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 2016, pp. 5862–5873.

[36] Anders Markussen, Mikkel Rønne Jakobsen, and Kasper Hornbæk. "Vulture: a mid-air word-gesture keyboard." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2014, pp. 1073–1082.

[37] David C McCallum and Pourang Irani. "ARC-Pad: absolute+ relative cursor positioning for large displays with a mobile touchscreen." In: *Proceedings of the 22nd annual ACM symposium on User interface software and technology*. 2009, pp. 153–156.

[38] Miguel A Nacenta, Yemliha Kamber, Yizhou Qiang, and Per Ola Kristensson. "Memorability of pre-designed and user-defined gesture sets." In: *Proceedings of the SIGCHI conference on human factors in computing systems*. 2013, pp. 1099–1108.

[39] Takashi Nakamura, Shin Takahashi, and Jiro Tanaka. "Double-crossing: A new interaction technique for hand gesture interfaces." In: *Asia-Pacific Conference on Computer Human Interaction*. Springer. 2008, pp. 292–300.

[40] Mathieu Nancel, Julie Wagner, Emmanuel Pietriga, Olivier Chapuis, and Wendy Mackay. "Mid-air pan-and-zoom on wall-sized displays." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2011, pp. 177–186.

[41] Ali Neshati, Bradley Rey, Ahmed Shariff Mohommed Faleel, Sandra Bardot, Celine Latulipe, and Pourang Irani. "BezelGlide: Interacting with Graphs on Smartwatches with Minimal Screen Occlusion." In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 2021, pp. 1–13.

[42] Tao Ni, Doug A Bowman, Chris North, and Ryan P McMahan. "Design and evaluation of freehand menu selection interfaces using tilt and pinch gestures." In: *International Journal of Human-Computer Studies* 69.9 (2011), pp. 551–562.

[43] Ian Oakley, Carina Lindahl, Khanh Le, DoYoung Lee, and MD Rasel Islam. "The flat finger: Exploring area touches on smartwatches." In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 2016, pp. 4238–4249.

[44] D Oberski, J Robertson, and M Kaptein. "Modern statistical methods for HCI." In: *Cham: Springer* (2016), pp. 69–71.

[45] Munir Oudah, Ali Al-Naji, and Javaan Chahl. "Hand gesture recognition based on computer vision: a review of techniques." In: *journal of Imaging* 6.8 (2020), p. 73.

[46] Andriy Pavlovych and Wolfgang Stuerzlinger. "The tradeoff between spatial jitter and latency in pointing tasks." In: *Proceedings of the 1st ACM SIGCHI symposium on Engineering interactive computing systems*. 2009, pp. 187–196.

[47] Jennifer Pearson, Gavin Bailey, Simon Robinson, Matt Jones, Tom Owen, Chi Zhang, Thomas Reitmaier, Cameron Steer, Anna Carter, Deepak Ranjan Sahoo, et al. "Can't Touch This: Rethinking Public Technology in a COVID-19 Era." In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 2022, pp. 1–14.

[48] Charles Perin and Pierre Dragicevic. "Manipulating multiple sliders by crossing." In: *Proceedings of the 26th Conference on l'Interaction Homme-Machine*. 2014, pp. 48–54.

[49] David Rempel, Matt J Camilleri, and David L Lee. "The design of hand gestures for human–computer interaction: Lessons from sign language interpreters." In: *International journal of human-computer studies* 72.10-11 (2014), pp. 728–735.

[50] Volker Roth and Thea Turner. "Bezel swipe: conflict-free scrolling and multiple selection on mobile touch screen devices." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2009, pp. 1523–1526.

[51] Léa Saviot, Frederik Brudy, and Steven Houben. "WRISTBAND. IO: expanding input and output spaces of a Smartwatch." In: *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. 2017, pp. 2025–2033.

[52] Joey Scarr, Andy Cockburn, Carl Gutwin, and Andrea Bunt. "Improving command selection with CommandMaps." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2012, pp. 257–266.

[53] Joey Scarr, Andy Cockburn, Carl Gutwin, and Sylvain Malacria. "Testing the robustness and performance of spatially consistent interfaces." In: *Proceed-*

ings of the SIGCHI Conference on Human Factors in Computing Systems. 2013, pp. 3139–3148.

[54] Marcos Serrano, Eric Lecolinet, and Yves Guiard. "Bezel-Tap gestures: quick activation of commands from sleep mode on tablets." In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. 2013, pp. 3027–3036.

[55] Buntarou Shizuki, Takaomi Hisamatsu, Shin Takahashi, and Jiro Tanaka. "Laser pointer interaction techniques using peripheral areas of screens." In: *Proceedings of the working conference on Advanced visual interfaces*. 2006, pp. 95–98.

[56] Garth Shoemaker, Anthony Tang, and Kellogg S Booth. "Shadow reaching: a new perspective on interaction for large displays." In: *Proceedings of the 20th annual ACM symposium on User interface software and technology*. 2007, pp. 53–56.

[57] Peng Song, Wooi Boon Goh, William Hutama, Chi-Wing Fu, and Xiaopei Liu. "A handle bar metaphor for virtual object manipulation with mid-air interaction." In: *Proceedings of the SIGCHI conference on human factors in computing systems*. 2012, pp. 1297–1306.

[58] *TouchFree solution: Turn any screen touchless | Ultraleap — ultraleap.com*. https://www.ultraleap.com/enterprise/touchless-experiences/touchfree-solution/. [Accessed 02-12-2023].

[59] Huawei Tu, Jin Huang, Hai-Ning Liang, Richard Skarbez, Feng Tian, and Henry Been-Lirn Duh. "Distractor Effects on Crossing-Based Interaction." In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 2021, pp. 1–13.

[60] Huawei Tu, Susu Huang, Jiabin Yuan, Xiangshi Ren, and Feng Tian. "Crossing-based selection with virtual reality head-mounted displays." In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 2019, pp. 1–14.

[61] Stephen Uzor and Per Ola Kristensson. "An Exploration of Freehand Crossing Selection in Head-Mounted Augmented Reality." In: *ACM Transactions on Computer-Human Interaction (TOCHI)* 28.5 (2021), pp. 1–27.

[62] Rafael Veras, Gaganpreet Singh, Farzin Farhadi-Niaki, Ritesh Udhani, Parth Pradeep Patekar, Wei Zhou, Pourang Irani, and Wei Li. "Elbow-Anchored Interaction: Designing Restful Mid-Air Input." In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 2021, pp. 1–15.

[63] Daniel Vogel and Ravin Balakrishnan. "Distant freehand pointing and clicking on very large, high resolution displays." In: *Proceedings of the 18th annual ACM symposium on User interface software and technology*. 2005, pp. 33–42.

[64] Panagiotis Vogiatzidakis and Panayiotis Koutsabasis. "Mid-air gestures for manipulation of multiple targets in the physical space: Comparing the usability of two interaction models." In: *CHI Greece 2021: 1st International Conference of the ACM Greek SIGCHI Chapter*. 2021, pp. 1–9.

[65] Toshiharu Wada, Masanobu Takahashi, Keiichiro Kagawa, and Jun Ohta. "Laser pointer as a mouse." In: *SICE Annual Conference 2007*. IEEE. 2007, pp. 369–372.

[66] Kieran Waugh, Mark McGill, and Euan Freeman. "Push or Pinch? Exploring Slider Control Gestures for Touchless User Interfaces." In: *Nordic Human-Computer Interaction Conference*. 2022, pp. 1–10.

[67] Kieran Waugh, Mark McGill, and Euan Freeman. "Proxemic Cursor Interactions for Touchless Widget Control." In: *Proceedings of the 2023 ACM Symposium on Spatial User Interaction*. 2023, pp. 1–12.

[68] Jacob O Wobbrock and Krzysztof Z Gajos. "Goal crossing with mice and trackballs for people with motor impairments: Performance, submovements, and design directions." In: *ACM Transactions on Accessible Computing (TACCESS)* 1.1 (2008), pp. 1–37.

[69] Dennis Wolf, Jan Gugenheimer, Marco Combosch, and Enrico Rukzio. "Understanding the heisenberg effect of spatial interaction: A selection induced error for spatially tracked input devices." In: *Proceedings of the 2020 CHI conference on human factors in computing systems*. 2020, pp. 1–10.

[70] Pui Chung Wong, Kening Zhu, Xing-Dong Yang, and Hongbo Fu. "Exploring eyes-free bezel-initiated swipe on round smartwatches." In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 2020, pp. 1–11.

[71] Yukang Yan, Yingtian Shi, Chun Yu, and Yuanchun Shi. "Headcross: Exploring head-based crossing selection on head-mounted displays." In: *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies* 4.1 (2020), pp. 1–22.

[72] Chuang-Wen You, Yung-Huan Hsieh, Wen-Huang Cheng, and Yi-Hsuan Hsieh. "AttachedShock: Design of a crossing-based target selection technique on augmented reality devices and its implications." In: *International journal of human-computer studies* 72.7 (2014), pp. 606–626.

[73] Weina Zhou and Kun Chen. "A lightweight hand gesture recognition in complex backgrounds." In: *Displays* 74 (2022), p. 102226.