Mobile Head Tracking for eCommerce and Beyond

Muratcan Cicek; University of California, Santa Cruz; Santa Cruz, California
Jinrong Xie, Qiaosong Wang, Robinson Piramuthu; eBay Inc; San Jose, California

Abstract

Shopping is difficult for people with motor impairments. This includes online shopping. Proprietary software can emulate mouse and keyboard via head tracking. However, such a solution is not common for smartphones. Unlike desktop and laptop computers, they are also much easier to carry indoors and outdoors.To address this, we implement and open source button that is sensitive to head movements tracked from the front camera of iPhone X. This allows developers to integrate in eCommerce applications easily without requiring specialized knowledge. Other applications include gaming and use in hands-free situations such as during cooking, auto-repair. We built a sample online shopping application that allows users to easily browse between items from various categories and take relevant action just by head movements. We present results of user studies on this sample application and also include sensitivity studies based on two independent tests performed at 3 different distances to the screen.

Introduction

According to Pew Research Center in 2016 [54] 80% of Americans shop online. Shopping at stores can be cumbersome as consumers do not prefer to be present at stores physically for several reasons [25]. In addition, there are also specific barriers [56] that make in-store shopping harder for people with motor impairments. We believe that online shopping would address this problem since it requires considerably less effort to interact with a digital devices to buy an item than to travel to the nearest store. But even this task is hard to complete for people with motor impairments. They are required to operate a pointer with a mouse-like device or tapping a certain point on the touch screen [15, 40, 50, 62]. According to The Centers for Disease Control and Prevention, there are 39.5 million adult Americans with some sort of physical functioning difficulty [17] and they potentially have difficulties during shopping, both in-store and online.

Proprietary software can emulate mouse and keyboard via head tracking. However, such solution is not common on smartphones. Modern smartphones such as iPhone X are packed with various sensors and abundant computing power. Unlike desktop and laptop computers, they are also much easier to carry around. Therefore, it is natural to make head tracking based interaction available on smartphones. For example, a button, which is typically sensitive to touch, could capture head tracking motion and activate when tracking region falls within its extent. Currently, it is difficult for developers to achieve this since it requires special expertise. Our hope is to make it easy for application developers to use buttons that capture head tracking events. This will make basic activities such as shopping easily accessible to people with motor impairments. Also, this could be extended to other situations demanding hand-free browsing.

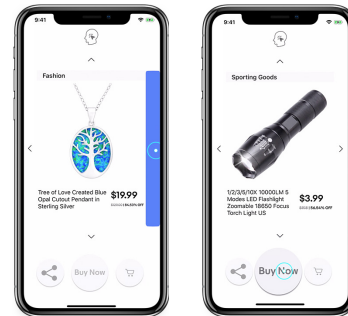We create such a button and plan to open source the project

Figure 1. *Illustration of a sample eCommerce application that we designed and open sourced. Cursor is shown as a cyan circle with dot at its center. Navigation buttons are filled with blue when they are fully activated (left image). Action buttons expand fully for activation (right image).*

so that developers could easily integrate to their own projects. Using our software, users could interact with the application only by head movement. This requires significantly less effort compared to using touchscreen. We build our application on top of the state-of-the-art ARKit 2 [21] toolkit. Our work shows that the Augmented Reality tools can be used for assistive technology.

In this paper, we focus on its application as an assistive tool mainly for people with motor impairments. To illustrate the use of such modality, we provide an online shopping experience for everyone through a mobile application. Although our initial motivation was developing an accessible online shopping application, the scope of possible use cases includes other applications such as gaming and hands-free situations such as during cooking, auto-repair. A great variety of mobile applications can be empowered with a touch-free design by employing our ready-to-use interface module.

KEY CONTRIBUTIONS

The contributions of this work are:

- An open-source solution for mobile user interfaces that provides hands-free interaction with smartphones. The code is completely open-sourced and will be available soon. It features an accurate mapping from 3D head pose to a virtual pointer, a dwelling function to generate hands-free touch events, a sample practice interface and a demo application for online shopping experience. Anyone should be able to extend our work and/or develop hands-free applications by integrating their design with our UI components.
- Design of a mobile user interface for online shopping that can benefit everyone including the users with motor impairments. To the best of our knowledge, this research is the first one that highlights the defined problem and proposes a practical solution to developing a working product.

IS&T International Symposium on Electronic Imaging 2020
Mobile Devices and Multimedia: Enabling Technologies, Algorithms, and Applications

303-1

Related Works

Our goal is to make basic activities such as shopping to be accessible to everyone via smartphone, including people with motor impairments. We first revisit most of the related work on online shopping and walk through several well-known eCommerce platforms prior to our study. In addition, we will also review interface requirements that are employed in online shopping and existing methods that try to meet these requirements.

Assistive Technology

WebAIM [61] lists several *selection* and *pointing* methods as the mainstream assistive technologies.

Selection Methods:

Adaptive switches with any kind replace a necessary button pressing for mouse clicking, tapping or for a more specific function like in use of a screen scanning solution. By design, adaptive switches are larger buttons that can be attached anywhere from floor to wheelchair headrests and can be hit by appropriate body parts. Tecla [57] lists 7 most common switches as adaptive switch, including several types of buttons, joysticks, Sip-and-Puff tools and blink recognizers. The mobile platform, iOS by Apple [23] where we develop our application also has built-in features like touch screen tapping as a switch and a front camera-based switch that detects very simple head movements as trigger in addition to the support of external switches.

Dwelling function is a selection mechanism rather than an assistive tool by itself. Most of solutions tend to have dwelling function as a replacement for switches we mentioned above because these solutions are designed to provide a completely hands-free interaction and greater mobility by not relying on any external switch. Instead, they require the user to hover over a target in graphical user interfaces for an activation time to select the target. As several comparative studies [66, 33] show, dwelling method is quite slower than clicking method for selection. It also has the Midas Touch problem, unintentional selection due to 'pointing-hold' on a random target and this issue limits the design graphical user interface since bigger targets are necessary [46].

Pointing Methods:

Adaptive Pointing Devices are several adaptive mouse solutions with different shapes and sizes to provide easier use in the market in addition to larger touchpad and touchscreens that recognize customized gestures like Mott et al. [41] suggest. While these solutions provide greater accessibility than standard input devices, they still require fine physical ability that majority of people with motor impairments do not have. Also, having an external device or an extra large touchscreen counteracts the logic of complete mobility that our approach brings.

Screen Scanning mechanisms scan in a specific order along horizontal and vertical axes of the screen in a loop until the user selects the desired target by a switch. Although this is the only practical interaction technique when the user has no physical ability, it functions as a single switch only in the worst cases. This is the least efficient pointing method in terms of time since the system needs to scan two dimensional targets in one dimension and the cost of missing item selection during the scan is the twice of looping time which would be very long when the number of items is considerably large. Like most of today's popular operating systems have, iOS also has a built-in scanning mechanism, Switch Control, [23] that supports both item and point scanning.

Gaze-tracking: The gaze-based pointing methods calculate an approximate gaze point by tracking the movement of eye components where the user has no other physical ability. n these unfortunate cases, gaze-based techniques [34, 27, 64] connect the user to the outside world by building a direct communication. There are also other gaze-based solutions with more sophisticated interaction [14, 23, 39, 42] that allows users to play games and browse the web. But these solutions generally require to use a well-calibrated external device under a fixed lighting condition. Also, comparisons between head-based and gaze-based interactions [4, 28] conclude that head-based techniques are more voluntary, stable and have greater accuracy while gaze-based techniques would be faster for some specific tasks like typing [18]. Despite experimental studies [20, 26, 49] that have potential to increase its accuracy on mobile platforms, the practical mobile usage of gaze-based interactions are still limited.

Head-based Pointing: Head-mounted stylus has a long history. It is used as a writing tool by attaching a regular pencil at the edge of the stylus. Today, we still have physical head-mounted styluses [48] for touchscreens and sophisticated products like Quha Zono [45] and Glassouse [19]. Besides physical devices, an important number of today's alternative pointing methods employ visual-based interactions [35] that detect and track a voluntary movement of a body part [53] for two-dimensional pointing. Betke et al. [5] show that visual tracking of body features, especially facial [e.g. face, nose, eyebrow], can be a successful pointing tool for people with motor impairments. Mauri et al. [38] also review assistive technologies for the same user group and conclude that visual-based systems would be the only way of computer interaction for some users. Advantages include flexibility and lower cost over other traditional assistive technologies. Today, there are many successful applications of head-based pointing in assistive technology (e.g. Camera Mouse [44], Smyle Mouse [30], Enable Viacam [36], HeadMouse Nano [13], EVA Facial Mouse [37]). In addition to assistive technology, we would also like to highlight the huge potential of head-based tracking in other areas, including desktop GUIs [6], wearable computing [9], and VR 3D user interface [8, 12, 28].

Evaluation of head-tracking on mobile: Today, there are also ready-to-use mobile head-tracking solutions such as EVA Facial Mouse [37] and Essential Accessibility [2] available on mobile platforms that rely on head-tracking and provide considerably free control of the mobile environment without any external devices, nor requiring a sensitive calibration. But these kind of products are only available on Android. On the other hand, we find that recent studies [52, 53, 51, 1] that evaluate head-tracking on mobile devices tend to use iOS as their experimental environments. In their first study, Roig-Maimó et al. [53] propose two similar tasks to evaluate head-tracking, a picture-revealing puzzle game for pointing and a item selection task for different sized items. In their most recent work, Roig-Maimó et al. [52] applies an user performance evaluation through Fitts' law for a mobile head-tracking interface by following the multidirectional tapping test described in the ISO standard [24] after his non-ISO study [51]. To the best of our knowledge, these are the first and still only Fitts' law studies on user performance of head-tracking interfaces with mobile devices.

303-2

IS&T International Symposium on Electronic Imaging 2020
Mobile Devices and Multimedia: Enabling Technologies, Algorithms, and Applications

Augmented Reality

Augmented Reality (AR) is another exciting field and there are several recent developments [3, 7] that potentially transform the way people play games [31, 59] or experience online shopping [55, 63]. The most popular mobile platforms (e.g. iOS and Android) encourage their developers to build more integrated applications with AR by providing ready-to-use native AR development kits (e.g. ARKit [21], ARCore [29]). Smart phone manufacturers like Apple upgraded the features (i.e. TrueDepth Camera [22]) of their products to support AR applications. While these exciting advancements are happening right now, we have discovered a smart way to benefit from these recent improvements for everyone. The augmented facial expressions/masks are one of the most common applications of AR and rely on the sophisticated Computer Vision tasks such as precise head pose tracking and facial gesture detection [10, 11] including gaze. Recently, Apple released iPhone X with its front TrueDepth Camera [22] to support further advanced AR features available in the ARKit 2 [21]. This toolkit serves as black-box on complicated head pose and face tracking and provides precise localization in 3D space. AR developers can easily access these measurements in few lines of code. In our work, we utilize ARKit 2 to develop an Assistive Technology that relies on the same measurements and project from real world coordinates to screen coordinates.

Online Shopping Interfaces

Online shopping has been growing incredibly fast [58] and keep expanding its tentacles globally ever since the dot com boom in the 1990's [60]. Nowadays, more and more consumers choose to go shopping online instead of in-store as it allows them to shop 24/7 from anywhere and to compare the products more easily among a greater varieties [25]. Monsuwe et al. [47] also points out additional factors that encourage consumers to shop online. These include situational factors, product characteristics, previous online shopping experiences, and trust in online shopping besides ease of its use.

For both online and in-store shopping, consumers need to search for a specific product among many other choices, choose the desired ones and compare them until final checkout. Any online shopping service needs the functionality of showing multiple products and allowing the user to select desired choices. Since the number of listing products cannot fit the screen size of devices in most cases, almost any Graphical User Interface (GUI) design for online shopping relies on scrolling to allow the user browse enough number of products.

Design Process
Initial Observations

These were our observations prior to start of this work. Firstly, people with motor impairments have several difficulties during shopping. We overview the literature and discuss our findings in the *Introduction* section.Current interface of online shopping application has its own barriers for people with motor impairments since it heavily relies on standard interaction methods. We highlighted this limitation in the related work subsection. Besides, standard interaction methods (i.e. selection, pointing and typing) need to be carefully adapted for people with motor impairments. Although sophisticated interactions introduced to Virtual and Augmented Reality inspired our work, they are nothing short

of limitation in the mobile environment. We surveyed relevant interaction methods individually in corresponding subsections under *Related Work* and try to choose the most optimal solution. Among alternative interaction methods for people with motor impairments, there are only a few that provide true mobility. This is especially true on iOS platform where open-source tool for hands-free pointing is sparse. Although Apple ARKit is mainly intended for entertainment e.g. Pokemon Go and Animoji, we saw the potential of taking advantage of its head/face tracking capability to empower mobile interface with hands-free option.

Design Requirements

The aforementioned observations and the urgent need to enable motor-impaired person using mobile device with minimum effort inspired us to create a mobile hands-free control without any auxiliary hardware that can be easily to integrated into existing applications. To be specific:

- *Hands-free* is a must-have as we want to empower the conventional mobile applications to be accessible to everyone including motor-impaired persons while reducing the friction in target-pointing and -selection.
- *Auxiliary-free* Any external device or sensor brings additional cost and undermines mobility in most cases.
- *Mobile-friendly* Nowadays, smartphones are ubiquitous and has become the main source of information we consume everyday. Being mobile-friendly means we can maximize the benefits of our heads-free control on the most common platform and be influential in numerous existing and future mobile applications.

Design Decisions

Next, we make the following design decisions to meet each requirement in the most compact way.

Interaction Methods

We propose a *Completely mobile* solution to the defined problem. After comprehensive literature search, we concluded that people with motor impairments have serious difficulty to use regular mobile devices and that the solutions rely on external devices, thus limiting mobility scientifically. To make our solution *hands-free* and *auxiliary-free*, we evaluate all possible methods for human-computer interaction in the *Related Work* section. These comparisons led us to pick head-based *pointing* with dwelling function for *selection*. With this combination, we make sure our solution relies only on the embedded features of the device - specifically the front camera, since the methods we prefer are visual-based interaction methods.

Application for Online Shopping

We would like to enable people with motor impairments to be able to shop online on smartphone and feel independent. We design an online shopping application that requires minimum effort to interact with. To keep our design even simpler, we build only browse-based shopping application that allows users to browse products across categories. This also eliminates other complex tasks such as searching and ranking which are required for a regular online shopping application. While a browse-only

IS&T International Symposium on Electronic Imaging 2020
Mobile Devices and Multimedia: Enabling Technologies, Algorithms, and Applications

303-3

Figure 2. Mobile head tracking architecture to control button using head "pointing". We created a set of modules to extend native UIKit to handle head tracking, pointer visualization, head gaze event detection, triggering and forwarding.

shopping application does not provide a complete shopping experience, it is still important since our approach will be one of the very first assistive eCommerce applications. Also, since we open source this project, anyone could extend its functionality by adding searching logic on top of our work.

Development Environment

With the intention of developing a mobile online shopping that only depends on visual-based interaction, head-tracking specifically, we searched for the most appropriated environment that has the necessary functionalities. Instant head-tracking is still a challenging task on mobile devices since it relies on power-hungry video processing and ability to capture high quality video through front-facing camera. This solution is not practical on many mobile device with limited resource and lead us to focus on the most recent product releases only. iPhone X has TrueDepth Camera [22], an impressive feature. It was originally designed to support advanced AR applications on iOS 11 and accessible through Apple's ARKit 2 [21], an AR development SDK. ARKit 2 implements efficient on-device head-tracking and exposes precise measurements of several facial landmarks to the developer. The biggest advantage of this setup is that it only relies on the device's built-in components without requiring extra accessories. This is power efficient and portable. Our open-source head-based interface components on iOS were built on top of ARKit 2. Even though the tools we built exist only on the most recent iPhone model, which may limit its usability, according to the report of Strategy Analytics [43], iPhone X is the world's best-selling smart phone model, shipping an impressive 16 million units during the first quarter of 2018. These numbers together with the promising future of AR applications lead us to believe that most of the smartphones on the market would intend to adopt similar features in the near future. Overall, a smart phone based solution is more affordable, portable, and more beneficial than many of the common external solutions that heavily rely on separate gaze or head-tracking gadget.

System Design and Implementation
Head-based Pointing

To meet our design requirements, we first implement a *hands-free*, *auxiliary-free*, *mobile-friendly* interaction system. Our system consists of two main modules. A module for head-based pointing, and a customized UI module for the interaction with on-screen pointer and feedback visualization.

Head-tracking

ARKit 2 comes with stable head-tracking capability where the 3D location, orientation of the tracked head can be queried at each AR session through *ARFaceAnchor* object. Abbaszadegan et al. [1] used the same technique to highlight the potential of using the TrueDepth front-facing camera of an iPhone X for tracking.

Head-to-pointer Mapping

However, the limitation of ARKit head-tracking is that it does not provide additional information regarding the location on the screen at which the head or nose is pointing. To bridge this gap between 3D head posture and pointing location on the phone screen, we designed an intuitive virtual stylus model to give user stable control of their pointing direction and instant feedback of the pointing location on screen. Figure 2 illustrates how the model works. To aid calculation of pointing location, we model the phone screen as a 2D plane perpendicular to the z-axis at position z=0. A proxy ray is spawned from the head center and passes through the nose. The ray-plane intersection is then calculated and is considered as the pointing location. Another possible solution is to project nose position directly on the screen and take its screen space coordinates as pointing location. We found that in practice the virtual stylus model takes into consideration the distance between head and phone, which allows distance-based sensitivity adjustment. Such feature is adorable as it adapts to UI interface that features widgets/controls that vary in size, shape, and gap. To compute the intersection, let $\mathbf{p}_0 \in \mathbb{R}^4$ denote the homogeneous coordinates of the original head position in the object space, and $\mathbf{W} \in \mathbb{R}^{4\times4}$ denote the world transformation matrix from *ARFaceAnchor* object updated at each frame. The initial virtual stylus direction is denoted by $\mathbf{d}_0 \in \mathbb{R}^3$ and points to the negative *z* direction perpendicular to the screen. At each frame, the head center is updated to $\mathbf{p} = \mathbf{W} \cdot \mathbf{p}_0$ and the pointing direction is updated to $\mathbf{d} = \mathbf{W} \cdot \mathbf{d}_0$. The resulting ray-plane intersection $\mathbf{b}$ in Normalized Device Coordinate (NDC) space $[0,1] \times [0,1]$ is then calculated by $\mathbf{b} = \mathbf{p} + \mathbf{d} \times t$, where $t = -\mathbf{p.z}/\mathbf{d.z}$. We further apply viewport transformation to map the NDC coordinates to the screen space that can be used for UI widget intersection test.

Pointer Visualization

To give user instant visual feedback on head-pointing, we visualize the location as an on-screen cross-hair using Apple SpriteKit API. Specifically, to cover the full screen and to provide precise tracking feedback, we add to our application a *UI-*

HeadGazeView layer that inherits from *SKView* to make *UI-HeadGazeView* a Spritekit Scene object that serves as a canvas on rendering 2D geometry on the screen to represent pointing location.

UI Module
Selection Method

Native UI widgets implemented by the Apple UIKit package only react to user physical touch event such as touch down, touch up, pinch, etc. Unfortunately, there is nothing to support interacting with the UI from mid-air. This requires a new definition of interaction for head-based pointer and native UI components. To bridge this gap between the pointing location and interaction with existing UI, we defined several interactions and implemented them through a set of customized UI widgets on top of Apple UIKit so that they can be seamlessly integrated into new iOS applications or extend the existing one with minimum effort. For example, in Swift, we created a *UIHoverableButton* class by extending iOS' default *UIButton* class to respond to the pointer interaction. Such customized button senses the pointer's location change and triggers hovering events whenever the pointer enters the button, and will further trigger selection event if the pointer has been hovering beyond a user specified time interval, hence dwelling. During pointer hovering, a user customized animation effect can be added to the *UIHoverableButton* to serve as visual cue on the elapsed hovering duration. For instance, by gradually increasing the button size or gradually filling the background with different colors. See Figure 1 for illustration.

Besides *UIHoverableButton*, there are other customized classes implemented to extend native UIKit to support head tracking, pointer visualization, head gaze event detection, triggering and forwarding. The inter-module connection is illustrated in Figure 2. At high level, *UIHeadGazeViewController* keeps track of head motion and passes the head's world transformation matrix to *UIHeadGazeView*. *UIHeadGazeView* then computes the head-to-pointer mapping, updates current pointer location, and notifies the registered event recognizer *UIHeadGazeRecognizer* about the updates. These updates are encapsulated in a *UIHeadGazeEvent* object. The design follows iOS UIKit protocol and blends into regular event handling scheme. Such design requires minimum effort to modify existing code base to support our head-based pointing feature.

User can select different types of *UIHeadGazeEvent* depending on how long the pointer needs to hover over a button before triggering the corresponding type of *UIHeadGazeEvent*. We currently have predefined two types based on the length of the dwelling duration: glance (1 sec) and gaze (2 secs). Trade-off has to be made on setting the duration, as longer duration reduces the chance of accidental clicking but increases latency. By default the duration is 1 second (glance). This configuration reduces the amount of accidental clicking of unintended buttons while maintaining the interaction at high throughput. Furthermore, to avoid unintentional repeated clicks on the same buttons, the button is configured not to emit another *UIHeadGazeEvent* until the pointer reenters that button. With this limitation, the user has to move the pointer away from the button to emulate touch up before they can re-click the button if needed. While this is an extra effort for intentional re-clicking, the effort is actually negligible compared to the one incurred by undoing the unintentional re-clicking, which normally requires clicking different buttons to reverse it.

Simple Online Shopping Interface Design

Once we completed the implementation of our head-based interaction system, we integrated this system into a simple shopping application to allow motor impaired person to shop online.

Product Browsing Page

Our shopping application starts with a product browsing page which takes up the main screen of the application (figure 1). We present one product at a time on the screen with a reasonably large photo along with title and price information. At the bottom of the screen are the three action buttons which allow users to share item on social media, checkout item, or add item to shopping cart. Each of the three action buttons brings up a separate hands-free page for the subsequent process. For the interest of space, we focus our discussion on the main product view page. Items are organized in a 2D grid and grouped in each row by their category. The item view is surrounded by four direction buttons that allow users to browse through items (horizontal swipe) and categories (vertical swipe). Users are presented with one cell of the grid at a time while they can swipe the item via "clicking" the direction buttons.

Open Source

We open-sourced the head-based pointing solution with advanced mobile UI components compatible with head-based pointing. We believe such generic library can benefit both researchers and developers in many different use cases. Through a sample online shopping application, we addressed several design concerns and showed practicality of our solution.

Experiments

Beyond the online shopping application we implement, we believe that the primary contribution of this work is the open-source tools for head-based pointing on iOS which is one of the most common mobile platforms all around the World. To see the first user experience and understand the relative performance, usability, and user preference of these tools, we conducted a field experiment with 75 able-bodied participants during an internal showcase of our organization. Despite we state our target user group as people with motor impairments, anyone would benefit from a hands-free interaction with smart phones. Notice that the physical abilities of people with motor impairments varies a lot from person to person and the assistive technology solutions they currently use would also vary. In this case, there is high possibility of unfair evaluation of the proposed system by only limited number of people with motor impairments. We believe that providing a baseline evaluation of this system with able-bodied participants would be an important contribution since the results would inspire the community for the future works. In addition to the field study, we also recruited 27 able-bodied participants for a lab-based user study to test our head-based interface for possible future work not limited by online shopping. Finally, We conducted a simpler experiment by hiring a person (24, male) with motor impairments to try the same interface and provide his feedback for comparison.
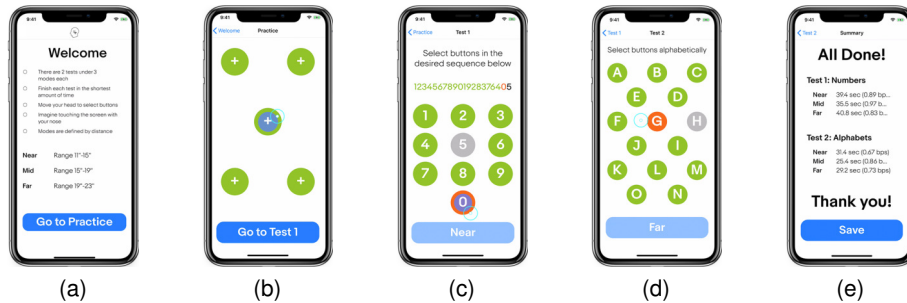
IS&T International Symposium on Electronic Imaging 2020
Mobile Devices and Multimedia: Enabling Technologies, Algorithms, and Applications

303-5

Figure 3. UI design for user study via 2 tests. Each test is performed at 3 different distances between face and screen (near, mid, far). A test is complete when the desired sequence is complete. Next button in the sequence to be selected is shown in orange. To reduce response time, we highlight the following button in the sequence in gray. Buttons fill up blue during dwell time after which the next button is highlighted in orange and the following button in gray. (a) welcome screen with instructions (b) unlimited practice session for warm up in different distance modes (c) first test for densely packed button around the center of screen (d) second test with buttons uniformly distributed across the screen (e) summary is shown in the final screen.

Participants

We conducted 2 separate experiments. In the field study, we recruited 75 able-bodied unpaid participants (35 female, 40 male) to experience our online shopping app. We randomly picked the participants during a crowded event and solicit their initial feedback on our approach right after they played with the app. In the lab based study, we recruited 27 able-bodied unpaid participants from our organization and involved them in an instructed user study. Among them, 16 were male and 11 were female. In both experiments, the participants had no prior experience with head-based pointing methods, nor dwelling function. They were from different ages with various races. Some of them wore glasses during the experiment. With this diverse group of participants, we intended to get a realistic insight into the learnability and usability of the system from novice adopters of the technology since this methodology is uncommon, especially on mobile. These studies are insightful on the practicality of our approach to a real-world application such as an online shopping solution.

Design and Procedure

For the lab-based study, We implemented a separate UI design as a single mobile application and conducted the experiment by running this App on an Apple iPhone X running iOS 11.2 with a resolution of 1125×2436 px and a pixel density of 458 ppi. This corresponds to a resolution of 375×812 Apple points (pt) which is an abstract unit that covers two pixels on retina devices. The camera was also the iPhone X's embedded 7-megapixel front-facing TrueDepth Camera [22]. We fixed the position of the phone by placing it on a holder in portrait mode. The participants were required to sit against the phone at 3 different distances for each test. We placed the phone on a holder which was attached to an adjustable table. Before the experiment, participants were also informed of the purpose and instructed to complete the tasks. The experiment lasted about 15 minutes per participant.

The user study App for this experiment had 5 different screens (figure 3) to show in order during each session. Our user study App opens with a welcome screen with the instructions of the overall experiment. Then, we have an unlimited practice session that allows the participants to get familiar with head-based pointing. We have two consecutive tests with different layouts on separate pages to get different feedbacks on the performance of participants. During the experiment, we also collected the times-

tamps of the participant's each action and the position of the cursor for analysis. At the end of this experiment, we also asked each participant to fill a questionnaire for recording their user experience. The experiment is carried out on two layouts with three distances: Near (11 to 15 inches), Mid (15 to 19 inches), Far (19 to 23 inches).

Test 1 (Numbers) was designed for densely packed targets around the center of screen (figure 3c). The first test screen contains 10 targets that were labeled from zero to nine by digits similar to the numeric keypads. In addition to highlighting the next target in orange by the predefined order, we also displayed the whole desired sequence on the upper part of screen and colored the sequence accordingly. The desired order was [12345678901928376405], we especially selected this sequence to catch participants' performance on vertical, horizontal and diagonal trials with different lengths.

Test 2 (Alphabets) was with targets uniformly distributed across the screen (figure 3d). We aimed to see the reachability of targets which positioned on different parts of the screen. Test 2 had 15 same-sized targets which were labeled by the letters, from A to N. The targets had almost the same distances to their neighbors and were required to be selected in the alphabetic order. Here, we expected that the participants could select the targets around the center of the screen easily while it could be hard to select the targets far from the center, especially the targets at the bottom.

In total, there were 27 participants $\times$ 3 distances $\times$ (20 + 15 targets) = 2835 trials in our user study.

Results

Our lab-based study tended to precisely evaluate the practicality of the proposed interaction. Therefore, we applied a quite similar study with Fitts' Law [16] since it is the standard way to evaluate this kind of approaches and derive the dependent measure throughput (Fitts' *index of performance*) as part of the comparison and evaluation [51].

Objective Evaluation

We calculated throughput (TP) as follows:

$$TP = \frac{Effective\ index\ of\ difficulty}{Movement\ time} = \frac{ID_e}{MT}, \quad (1)$$

303-6

IS&T International Symposium on Electronic Imaging 2020
Mobile Devices and Multimedia: Enabling Technologies, Algorithms, and Applications

where ID_e is derived from the movement amplitude A and effective target width W_e and MT is averaged movement time per trial over a sequence. They have units "bits" and "seconds" respectively, the units of TP are "bits per second (bps)". The effective index of difficulty is also a measurement of the difficulty and user precision in completing a task:

$$ID_e = \log_2(\frac{A}{W_e} + 1), \tag{2}$$

where A is the movement amplitude, the distance between the centers of two consecutive targets and W_e is the effective target width, calculated from the width distribution of selection coordinates made by a participant over a sequence of trials which is calculated as below:

$$W_e = 4.133 \cdot S_x, \tag{3}$$

Note that we used the standard-deviation method to calculate throughput [32]. This is because our specific design utilizes dwelling function and has no *error rate* [65]. Also, we followed Roig-Maimó's [51] equations and applied our own Non-ISO tests. A detailed description of the calculation of throughput can be found in [51, 65]. We show the summary result of dwell time and throughput in Fig. 5. The top row shows average dwell time of all users at the 3 distances. The box and whisker plot goes from the lower to upper quartile values of the dwell time data collected from all users at all 3 distances, with a line at the median. The whiskers extends from the box to indicate the data range. The median value of the data range are plotted in black, while mean values per distance are shown in the legend. We also follow the same fashion to show throughput per motion sequence in the middle row. Finally, at the bottom row we show throughput per user averaged over all motion sequences.

For the *Numbers* test, it is worth noting that it is easier for the participants to navigate in the *Mid* range compared to the *Near* range, and most participants completed the test in the shortest time under the *Far* range setting. This is because the spacing between the buttons are relatively large in this test. If the users are operating at a closer distance to the screen, it requires more head movements to control the cursor to move across larger distances. This is especially true when the participants perform large diagonal movements from number 0 to 1 or from 1 to 9. On the contrary, for the *Alphabets* test, participants complete under the *Near* setting in shortest amount of time with highest throughput. As the button size and spacing become smaller (See Fig. 3), it is easier to navigate in the *Near* range as it offers more precise control. The above observations can also be verified at the bottom plot from Fig. 5. For the *Numbers* experiment, throughputs are generally higher at far distances, while for the *Alphabets* experiment, throughputs are usually higher at near distances. In addition to throughput calculation, we also studied the direction of selection points with respect to buttons at different locations (see Fig. 4). We calculated the eigen-vectors of selection point coordinates on the covariance matrix to see if there is a pattern. We found that for the first column of buttons, statistics show that most users point to the left side. This may indicate that the participants are trying to position the cursor more towards the left side of the screen and most of the cursors overshoot the center position of the buttons. For the center column of the buttons, the eigen-vectors show a

pattern of a path placed more towards the right side. For the third and final column, most participants tend to reverse the path more towards the left side in order to hit the buttons sequentially. On the third row, the pattern changed compared to the previous two rows, indicating participants getting more familiar with the app and thus exhibit better control.

Subjective Evaluation:

We applied a questionnaire to the participants of our lab-based study right after their individual sessions. 96.3% of participants considered moving cursor by head was easy and normal. Majority of them also reported that the selection method was normal. 48.1% reported there was no scientific difference between the difficulty of selection based on the target regions (upper, lower and middle portions of the screen on portrait mode). Only 18.5% thought the lower region was hardest to reach while 29.6% disagreed and said the targets at the upper region was hardest to select. Majority of the participants claimed practicing our tests by keeping the distance at mid from the phone was easier to use our system while 29.6% said far distance was easier.

Our single participant with motor impairments also completed both tests and experienced the shopping App. He completed the first test slower than other participants since he had difficulty to keep his head stable for dwelling. He shows average performance on the second test as he gets more familiar with dwelling and the software interface. At the end, he found our system quite practical for several use-cases including typing. He also reported that our system's pointing accuracy is quite better than the software he uses on a daily basis for communication which runs on Windows via a web-cam.

We also applied a questionnaire to the participants who experienced our online shopping interface during the field study. We sought their initial thoughts about the proposed hands-free online shopping App. Among 75 participants, 51 of them thought this is a useful idea. 41.3% reported our App was easy to use even though they were not familiar with this interaction method. 31 participants also found it enjoyable while 5 of the participants could not find a use case and said this solution is not for them. Only 9.3% claimed moving the cursor by head was hard for them while 52% thought it was easy and 38.7% thought it was normal to practice. Furthermore, 50 participants reported that this solution would fit somebody's needs they know.
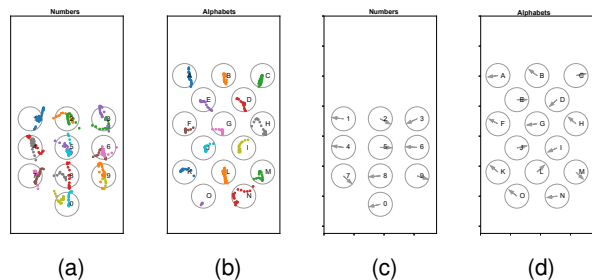

Figure 4. *Analysis of dwell points. (a-b) Randomly selected dwell points from a participant. (c-e) Directions of eigenvectors derived from dwell point coordinates.*
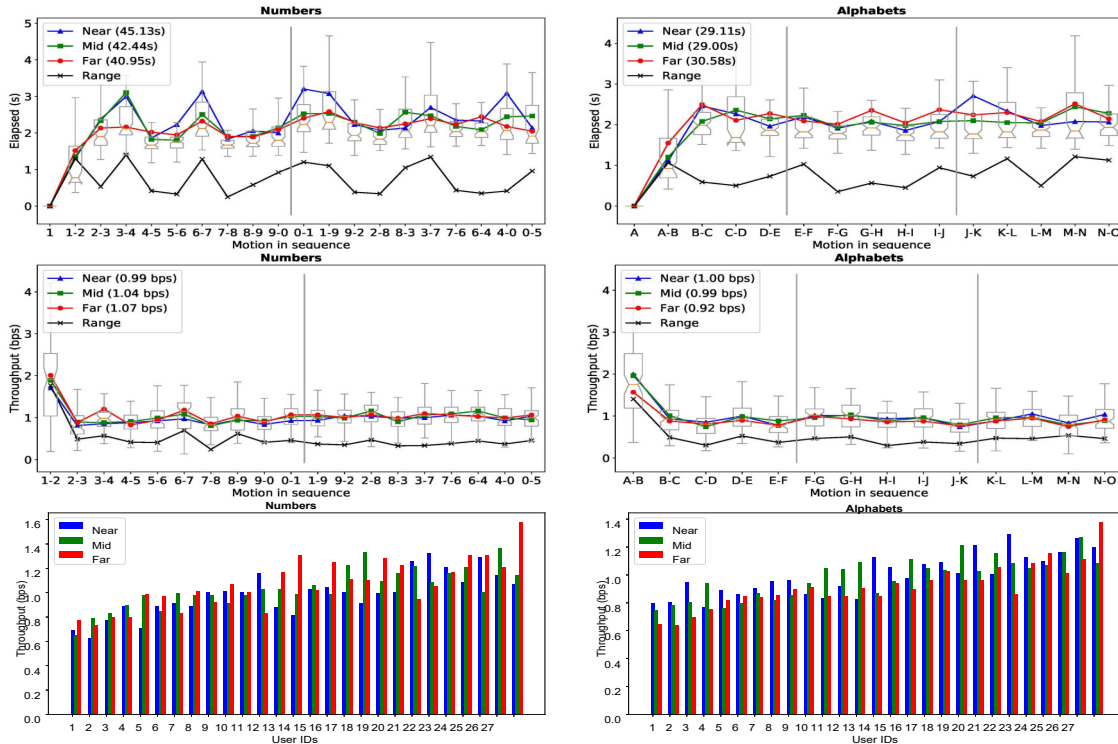
IS&T International Symposium on Electronic Imaging 2020
Mobile Devices and Multimedia: Enabling Technologies, Algorithms, and Applications

303-7

Figure 5. *Analysis of dwell time and throughput. Top: elapsed time per motion sequence averaged over all participants. Middle: throughput per motion sequence averaged over all participants. We also show box and whisker plot for data collected from all participants at each sequence. The range plot shows median of the range of a set of data collected from all participants across all 3 distances. Bottom: throughput per participant averaged over all motion sequences, sorted by throughput averaged across all distances. Note that for the numbers experiment, the throughput is generally higher at far distances, while for the alphabets experiment, throughputs are usually higher at near distances. See Sec. 6 for details*

Discussion

We saw that online shopping has several barriers for people with motor impairments. Our design led us to address this problem with a *hands-free*, *auxiliary-free*, *completely mobile online shopping sample application*. Our application allows users to browse several deals from different categories and buy/share the desired products. Users do not need any external device, nor a fine calibration step for using our approach. This eliminates the need of touch screen which requires fine motor control. The initial user feedback also shows that this interaction with an online shopping application is practical and enjoyable. Beyond this application, we actually proposed our *hands-free* interface components as an open-source solution so that developers may incorporate head tracking into their mobile applications, not limited to eCommerce. Our open-source solution can be used in several applications from communication tools for people with motor impairments to a hands-free recipe or DYI application.

Our lab-based study suggests keeping the phone at mid-distance (15 to 19 inches) from the face increases the interaction quality and is more comfortable since it provides an accurate and robust interaction. There is a trade-off between cursor speed and stableness. To be specific, small phone-to-head distance requires bigger head movements that can easily cause fatigue but delivers fine control with higher stableness, while larger distance requires minor head movements but with higher precision and stable head movements which is potentially stressful as well. Future work on head-based interaction may consider this fact and design its inter-face accordingly. Another fact is that, while majority of the participants reported that there was no scientific difference between the difficulty of selection based on the target regions, we concluded that the targets around the center of the screen are more reachable than the targets on the lower portion of the screen since there are higher throughput values for the center sequence of *Alphabets* test (see Fig. 5). This is natural since pitching the head up or down is relatively more uncomfortable task than yawing the head left or right. Placing the frequently visited targets around the screen center while reserving the lower portion for information display or for less commonly used targets like the ones in our design (Fig. 1) would provide a more ergonomic interaction via head-based pointing on mobile and keeps the amount of uncomfortable pitching tasks to the minimum.

We highlight the potential of the proposed approach as an important number of people who experienced our approach considered it useful for them in several use cases such as baby feeding or fixing automobile with dirty hands. In addition to able-bodied user group, our single participant with motor impairments was also able to practice precise target selection on such a small screen for the very first time. He also reported that a hands-free interaction with smart phones is especially a serious need for any wheelchair rider even if they have fine hand control since they also would like to keep interacting with their phones while moving. In so many cases like this, our open-sourced hands-free UI components would be really helpful for developers to build new applications in consideration or these needs. Therefore, future

303-8

IS&T International Symposium on Electronic Imaging 2020
Mobile Devices and Multimedia: Enabling Technologies, Algorithms, and Applications

work would be developing any kind of mobile applications on top of our framework or adding new head-pointing sensitive UI components to our open-sourced tool set.

Conclusion

In this work, we proposed an open-source solution for mobile phones to enable an easy-to-use interface for buttons that are sensitive to head pointing.

We hope that this open source will make it easier for developers to incorporate head tracking into applications not limited to eCommerce. This includes socially impactful applications for users with motor control disabilities. It also allows hands-free applications such as following recipe while cooking or fixing car while following an instructional manual. Fun applications like gaming can also consume this.

Author Biography

Muratcan Cicek received his BS in Computer Science from Ozyegin University (2017) and is still pursuing his Ph.D. in Computer Engineering at University of California, Santa Cruz. As a person with motor impairments the author has been benefiting from head-tracking mechanisms that provide pointing functionality on stationary settings for years, his recent work focuses on bringing such techniques on mobile as this study within eBay's Computer Vision team happened during his summer 2018 internship.

IS&T International Symposium on Electronic Imaging 2020
Mobile Devices and Multimedia: Enabling Technologies, Algorithms, and Applications

303-9

References

[1] Mahdieh Abbaszadegan, Sohrab Yaghoubi, and I Scott MacKenzie. Trackmaze: A comparison of head-tracking, eye-tracking, and tilt as input methods for mobile games. In *International Conference on Human-Computer Interaction*, pages 393–405. Springer, 2018.

[2] Essential Accessibility. Essential accessibility, 2018. Retrieved August 6, 2018 from `https://www.essentialaccessibility.com/assistive-technology-for-android/`.

[3] Woodrow Barfield. *Fundamentals of wearable computers and augmented reality*. CRC Press, 2015.

[4] Richard Bates and Howell O Istance. Why are eye mice unpopular? a detailed comparison of head and eye controlled assistive technology pointing devices. *Universal Access in the Information Society*, 2(3):280–290, 2003.

[5] Margrit Betke, James Gips, and Peter Fleming. The camera mouse: Visual tracking of body features to provide computer access for people with severe disabilities. *IEEE Transactions on neural systems and Rehabilitation Engineering*, 10(1):1–10, 2002.

[6] Martin Bichsel and Alex Pentland. Automatic interpretation of human head movements. 1993.

[7] Mark Billinghurst, Adrian Clark, Gun Lee, et al. A survey of augmented reality. *Foundations and Trends® in Human–Computer Interaction*, 8(2-3):73–272, 2015.

[8] Doug Bowman, Ernst Kruijff, Joseph J LaViola Jr, and Ivan P Poupyrev. *3D User interfaces: theory and practice*. Addison Wesley, 2004.

[9] Stephen Brewster, Joanna Lumsden, Marek Bell, Malcolm Hall, and Stuart Tasker. Multimodal 'eyes-free' interaction techniques for wearable devices. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 473–480. ACM, 2003.

[10] Andrea Canessa, Manuela Chessa, Agostino Gibaldi, Silvio P Sabatini, and Fabio Solari. Calibrated depth and color cameras for accurate 3d interaction in a stereoscopic augmented reality environment. *Journal of Visual Communication and Image Representation*, 25(1):227–237, 2014.

[11] Chien-Hsu Chen, I-Jui Lee, and Ling-Yi Lin. Augmented reality-based self-facial modeling to promote the emotional expression and social skills of adolescents with autism spectrum disorders. *Research in developmental disabilities*, 36:396–403, 2015.

[12] Rory MS Clifford, Nikita Mae B Tuanquin, and Robert W Lindeman. Jedi forceextension: Telekinesis as a virtual reality interaction metaphor. In *3D User Interfaces (3DUI), 2017 IEEE Symposium on*, pages 239–240. IEEE, 2017.

[13] Origin Instruments Corporation. Headmouse nano, 2017. Retrieved July 17, 2018 from `http://www.orin.com/access/headmouse/`.

[14] Tobii Dynavox. Microsoft & tobii dynavox, 2018. Retrieved July 16, 2018 from `https://www.tobiidynavox.com/en-US/landing-pages/td_and_microsoft/`.

[15] Leah Findlater, Karyn Moffatt, Jon E Froehlich, Meethu Malu, and Joan Zhang. Comparing touchscreen and mouse input performance by people with and without upper body motor impairments. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 6056–6061. ACM, 2017.

[16] Paul M Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology*, 47(6):381, 1954.

[17] National Center for Health Statistics. Disability and functioning (noninstitutionalized adults aged 18 and over), 2017.

[18] Yulia Gizatdinova, Oleg Špakov, and Veikko Surakka. Comparison of video-based pointing and selection techniques for hands-free text entry. In *Proceedings of the international working conference on advanced visual interfaces*, pages 132–139. ACM, 2012.

[19] Glassouse. Glassouse assistive device, 2018. Retrieved July 17, 2018 from `http://glassouse.com/`.

[20] Qiong Huang, Ashok Veeraraghavan, and Ashutosh Sabharwal. Tabletgaze: unconstrained appearance-based gaze estimation in mobile tablets. *arXiv preprint arXiv:1508.01244*, 2015.

[21] Apple Inc. Get ready for arkit 2, 2018. Retrieved July 17, 2018 from `https://developer.apple.com/arkit/`.

[22] Apple Inc. Truedepth camera, 2018. Retrieved July 17, 2018 from `https://www.apple.com/iphone-x/#truedepth-camera`.

[23] Apple Inc. Use switch control to navigate your iphone, ipad, or ipod touch, 2018. Retrieved July 15, 2018 from `https://support.apple.com/en-us/ht201370`.

[24] ISO ISO. 9241-9 ergonomic requirements for office work with visual display terminals (vdts)-part 9: Requirements for non-keyboard input devices (fdis-final draft international standard), 2000. *International Organization for Standardization*.

[25] KPMG. The truth about online consumers. 2017 global online consumer report, Klynveld Peat Marwick Goerdeler, 2017.

[26] Kyle Krafka, Aditya Khosla, Petr Kellnhofer, Harini Kannan, Suchendra Bhandarkar, Wojciech Matusik, and Antonio Torralba. Eye tracking for everyone. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2176–2184, 2016.

[27] Andrew Kurauchi, Wenxin Feng, Ajjen Joshi, Carlos Morimoto, and Margrit Betke. Eyeswipe: Dwell-free text entry using gaze paths. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 1952–1956. ACM, 2016.

[28] Mikko Kytö, Barrett Ens, Thammathip Piumsomboon, Gun A Lee, and Mark Billinghurst. Pinpointing: Precise head-and eye-based target selection for augmented reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 81. ACM, 2018.

[29] Google LLC. Arcore overview, 2018. Retrieved July 17, 2018 from `https://developers.google.com/ar/discover/`.

[30] Perceptive Devices LLC. Smylemouse, 2016. Retrieved July 15, 2018 from `https://smylemouse.com/`.

[31] Zhihan Lv, Alaa Halawani, Shengzhong Feng, Shafiq Ur Réhman, and Haibo Li. Touch-less interactive augmented reality game on vision-based wearable device. *Personal and Ubiquitous Computing*, 19(3-4):551–567, 2015.

[32] I Scott MacKenzie. Fitts' law. *The Wiley Handbook of Human Computer Interaction*, 1:347–370, 2018.

[33] John Magee, Torsten Felzer, and I Scott MacKenzie. Camera mouse+ clickeraid: Dwell vs. single-muscle click actuation in mouse-replacement interfaces. In *International Conference on Universal Access in Human-Computer Interaction*, pages 74–84. Springer, 2015.

[34] Päivi Majaranta. *Gaze Interaction and Applications of Eye Tracking: Advances in Assistive Technologies: Advances in Assistive Technologies*. IGI Global, 2011.

[35] Cristina Manresa-Yee, Pere Ponsa, Javier Varona, and Francisco J Perales. User experience to improve the usability of a vision-based interface. *Interacting with Computers*, 22(6):594–605, 2010.

Retrieved July 9, 2018 from `https://www.cdc.gov/nchs/fastats/disability.htm`.

303-10

IS&T International Symposium on Electronic Imaging 2020
Mobile Devices and Multimedia: Enabling Technologies, Algorithms, and Applications

[36] Cesar Mauri. Enable viacam, 2017. Retrieved July 15, 2018 from `http://eviacam.crea-si.com/index.php`.

[37] Cesar Mauri. Eva facial mouse, 2018. Retrieved July 16, 2018 from `https://github.com/cmauri/eva_facial_mouse#user-content-eva-facial-mouse`.

[38] César Mauri, Toni Granollers i Saltiveri, Jesús Lorés Vidal, and Mabel García. Computer vision interaction for people with severe movement restrictions. *Human Technology: An Interdisciplinary Journal on Humans in ICT Environments, vol. 2, núm. 1, p. 38-54*, 2006.

[39] Microsoft. Eye control for windows 10, 2018. Retrieved July 16, 2018 from `https://www.microsoft.com/en-us/garage/wall-of-fame/eye-control-windows-10/`.

[40] Kyle Montague, Hugo Nicolau, and Vicki L Hanson. Motor-impaired touchscreen interactions in the wild. In *Proceedings of the 16th international ACM SIGACCESS conference on Computers & accessibility*, pages 123–130. ACM, 2014.

[41] Martez E Mott, Radu-Daniel Vatavu, Shaun K Kane, and Jacob O Wobbrock. Smart touch: Improving touch accuracy for people with motor impairments with template matching. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 1934–1946. ACM, 2016.

[42] MyGaze. Mygaze assistive, 2018. Retrieved July 16, 2018 from `http://www.mygaze.com/products/mygaze-assistive/`.

[43] Neil Mawston. Apple iphone x becomes world's no.1 smartphone model in q1 2018. Technical report, Strategy Analytics, 2018. Retrieved August 13, 2018 from `https://www.strategyanalytics.com/access-services/devices/mobile-phones/smartphone/smartphone-model-tracker/reports/report-detail/apple-iphone-x-becomes-world-s-best-selling-smartphone-model-in-q1-2018`.

[44] Trustees of Boston College. Cameramouse, 2018. Retrieved July 15, 2018 from `http://www.cameramouse.org/`.

[45] Quha oy. Quha zono, 2018. Retrieved July 15, 2018 from `http://www.quha.com/products-2/zono/`.

[46] Abdul Moiz Penkar, Christof Lutteroth, and Gerald Weber. Designing for the eye: design parameters for dwell in gaze interaction. In *Proceedings of the 24th Australian Computer-Human Interaction Conference*, pages 479–488. ACM, 2012.

[47] Toñita Perea y Monsuwé, Benedict GC Dellaert, and Ko De Ruyter. What drives consumers to shop online? a literature review. *International journal of service industry management*, 15(1):102–121, 2004.

[48] Ondrej Polacek, Thomas Grill, and Manfred Tscheligi. Nosetapping: what else can you do with your nose? In *Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia*, page 32. ACM, 2013.

[49] Rajeev Ranjan, Shalini De Mello, and Jan Kautz. Light-weight head pose invariant gaze tracking. *arXiv preprint arXiv:1804.08572*, 2018.

[50] Cameron N Riviere and Nitish V Thakor. Effects of age and disability on tracking tasks with a computer mouse: Accuracy and linearity. 1996. PMID: 8868412.

[51] Maria Francesca Roig-Maimó, I Scott MacKenzie, Cristina Manresa-Yee, and Javier Varona. Evaluating fitts' law performance with a non-iso task. In *Proceedings of the XVIII International Conference on Human Computer Interaction*, page 5. ACM, 2017.

[52] Maria Francesca Roig-Maimó, I Scott MacKenzie, Cristina Manresa-Yee, and Javier Varona. Head-tracking interfaces on mobile devices: Evaluation using fitts' law and a new multi-directional corner task for small displays. *International Journal of Human-Computer Studies*, 112:1–15, 2018.

[53] Maria Francesca Roig-Maimó, Cristina Manresa-Yee, Javier Varona, and I Scott MacKenzie. Evaluation of a mobile head-tracker interface for accessibility. In *International Conference on Computers Helping People with Special Needs*, pages 449–456. Springer, 2016.

[54] Aaron Smith and Monica Anderson. Online shopping and purchasing preferences. Technical report, Pew Research Center, 2016.

[55] Marco Speicher, Philip Hell, Florian Daiber, Adalberto Simeone, and Antonio Krüger. A virtual reality shopping experience using the apartment metaphor. In *Proceedings of the 2018 International Conference on Advanced Visual Interfaces*, page 17. ACM, 2018.

[56] Bonnie Swaine, Delphine Labbé, Tiiu Poldma, Maria Barile, Catherine Fichten, Alice Havel, Eva Kehayia, Barbara Mazer, Patricia McKinley, and Annie Rochette. Exploring the facilitators and barriers to shopping mall use by persons with disabilities and strategies for improvements: Perspectives from persons with disabilities, rehabilitation professionals and shopkeepers. *ALTER-European Journal of Disability Research/Revue Européenne de Recherche sur le Handicap*, 8(3):217–229, 2014.

[57] Tecla. Introduction to 7 common adaptive switches, 2018. Retrieved July 16, 2018 from `https://gettecla.com/blogs/news/introduction-to-assistive-switches`.

[58] the Census Bureau. Quarterly retail e-commerce sales 1st quarter 2018. Technical report, U.S. Department of Commerce, 2018.

[59] Bruce H Thomas. A survey of visual, mixed, and augmented reality gaming. *Computers in Entertainment (CIE)*, 10(1):3, 2012.

[60] Efraim Turban, Jon Outland, David King, Jae Kyu Lee, Ting-Peng Liang, and Deborrah C Turban. *Electronic commerce 2018: a managerial and social networks perspective*. Springer, 2017.

[61] WebAIM. Assistive technologies, 2018. Retrieved July 12, 2018 from `https://webaim.org/articles/motor/assistive`.

[62] Jacob O Wobbrock. Improving pointing in graphical user interfaces for people with motor impairments through ability-based design. In *Assistive Technologies and Computer Access for Motor Disabilities*, pages 206–253. IGI Global, 2014.

[63] Mark Yi-Cheon Yim, Shu-Chuan Chu, and Paul L Sauer. Is augmented reality technology an effective tool for e-commerce? an interactivity and vividness perspective. *Journal of Interactive Marketing*, 39:89–103, 2017.

[64] Xiaoyi Zhang, Harish Kulkarni, and Meredith Ringel Morris. Smartphone-based gaze gesture communication for people with motor disabilities. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 2878–2889. ACM, 2017.

[65] Xuan Zhang and I Scott MacKenzie. Evaluating eye tracking with iso 9241-part 9. In *International Conference on Human-Computer Intrction*, pages 779–788. Springer, 2007.

[66] Rafael Zuniga and John Magee. Camera mouse: Dwell vs. computer vision-based intentional click activation. In *International Conference on Universal Access in Human-Computer Interaction*, pages 455–464. Springer, 2017.
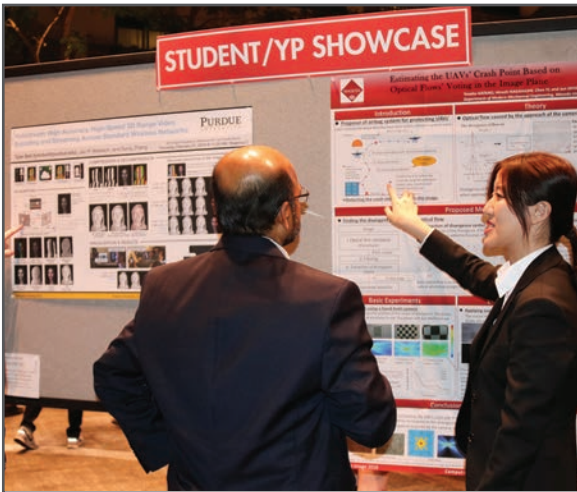
IS&T International Symposium on Electronic Imaging 2020
Mobile Devices and Multimedia: Enabling Technologies, Algorithms, and Applications

303-11